Social Investment Agency
Toi Hau Tāngata

Analyst's Foundation

Developing skills in integrated data research

July 2026

New Zealand Government
Te Kāwanatanga o Aotearoa

Author

Simon Anastasiadis

Acknowledgements

Andrew Stephenson, Stats NZ Integrated Data teams

Creative Commons Licence

Liability

Citation

Social Investment Agency 2026. Analyst's Foundation. Wellington, New Zealand.

ISBN 978-1-997329-02-2 (online)

Published in July 2026 by
Social Investment Agency
Wellington, New Zealand

Contents

Developing skill with integrated data

Integrated data can be a powerful tool for conducting research and creating new knowledge. In New Zealand, the Integrated Data Infrastructure (IDI) and Longitudinal Business Database (LBD) are two collections of integrated data protected and maintained by Stats NZ.

While both the IDI and LBD are significant resources, their complexity often makes them intimidating environments to begin working in. A range of materials have already been published to aid researchers working with integrated data. These include training videos, standardised tools, and guidance on project and process design. However, effective use of integrated data requires not just technical skill – how a researcher thinks about their work also plays a significant role.

This document sets out some of the thinking and approaches that we have observed make researchers more effective when working with the IDI. This advice is published so that more researchers might use integrated data with skill and confidence. It includes an appendix with three guided analyses to support researchers looking to practise these skills.

Integrated data differs from non-integrated

Integrated data collections contain microdata from many different sources. This enables insights that are not possible from a single source, but it also requires analytic processes that are not necessary when working with data from a single source.

Key aspects of integrated data

Integrated data is created by constructing links between data drawn from different sources. These links mean that connections can be drawn where the same person, organisation, or location appears in different collections. This has consequences for the use and protection of the data:

- **More extensive data has greater sensitivity and merits greater protection.**
  Access to integrated data is controlled by Stats NZ, under their Five Safes and Ngā Tikanga Paihere frameworks. These ensure that only approved people can see the unit record data.
- **Data in the IDI and LBD have been de-identified.**
  This means that personally identifying information (such as name, day of birth, address) has been removed. In some cases, these are encrypted and replaced by numeric codes.
- **Data integration does not merge data into a single source.**
  It only enables data from different sources to be combined. This means that analyses use many different tables and require multiple joins between them.
- **Data integration does not include restructuring the data for ease of analysis.**
  Data in the IDI is often presented in the same format that is used by the source organisation. This is seldom the format of most use for conducting research across different sources. As a result, the IDI contains over 400 tables each with their own structure.

What results can be released

Access to the IDI and LBD is only permitted within a secure data lab environment. All outputs from the secure environment are checked for confidentiality before being made available outside the environment. Until results have been released, they cannot be shared with people who are not approved to access the underlying data.

Stats NZ publishes the Microdata Output Guide that specifies the rules for releasing results. Researchers are responsible for preparing their output according to these rules. The rules do not prevent detailed and robust research from being conducted but they do require researchers to consider what level of detail they need for publication. The most common rules to consider are:

- Minimum counts of observations are required for outputs.
- Rounding includes randomness to determine whether values are rounded up or down.
- Values related to a single organisation cannot be released even if they relate to many people.
- Non-data material, such as code or notes, must go through the checking process to be shared outside the data lab.

Together these rules ensure that no individual person or organisation can be re-identified and that no information about an individual person or organisation can be recovered with high confidence.

Understanding what is fast and slow

While an analytic summary from an internal database might be delivered within hours, working with integrated data is much slower. This is due to a mix of factors:

- **Datasets that have been integrated vary in format, structure, limitations, and quality.** Combining data from multiple sources often requires custom handling of each source. The more data sources used; the more effort is required to prepare them for that analysis.
- **Integrated data researchers are not experts in all the data they work with.** They often work without knowing the source organisation's standard processing rules. As a result, researchers must spend a significant effort checking data quality and ensuring their data use is robust.
- **People who have not been approved to view the unit record data can only review results that have been released.** This creates delays in feedback loops and iteration between researchers and their stakeholders.

For integrated data projects, the key factor determining the duration of the work is the number of data sources used. More sources mean more complexity and slower delivery. As Table 1 and

Table 2 demonstrate, this is due to increased time in data preparation and has little relation to the amount of output that is requested.

As a rough indication of time frames, we suggest allowing 1 to 2 days of data preparation for each concept or data source that will be required for the analysis. This means that a medium sized project can require 20 days' worth of data wrangling. Reusing existing resources will decrease this time. A researcher's very first integrated data project will likely take twice as long.

Table 1: Example of easy analysis that might appear slow

Task	For people who interact with NZ Police, compare life satisfaction against every other measure in the General Social Survey (GSS).
Non-technical perspective	Concern that the large number of comparisons will be time consuming to create.
Analytic approach	Quick and straightforward. Only two input tables. GSS table already structured in ideal format. Connecting GSS and NZ Police data only requires a simple join. Repetitive processing done by computer not by the researcher.

Table 2: Example of slow analysis that might appear fast

Task	Count the number of benefit recipients with children who have diabetes.
Non-technical perspective	May expect request to be straightforward as only a single number to output. Benefit receipt, children, and diabetes are all unambiguous concepts.
Analytic approach	Very challenging. No single data source for diabetes information. Could construct by combining pharmaceuticals, lab tests, hospital in-patients, and hospital out-patients tables. Multiple ways that parenting status can be defined in the data: birth records, child care records, and households. May need to test and compare approaches.

Plan and structure your approach

Although it can be tempting to focus on writing code, an effective data project requires deliberate planning and documentation. Together these will save time and effort for yourself and others.

Plan an analysis in common phases

The better you plan your project, the smoother it will run and the higher the quality of the end result. We have observed five common phases that data and analytics projects go through, which are a good basis for planning:

Phase I: Research question – setting the aim and scope of the work.
This includes clarifying what needs to be done, why it is required, and how the expected output will be used. It may also include discussions with stakeholders and negotiation of trade-offs that need to be made.

Phase II: Concept definition – extracting from source data the information or concepts of interest. Based on the requirements of the research question, this phase ensures that each measure or fact required for analysis is available and fit for purpose.

Phase III: Dataset assembly – combining separate definitions together ready for analysis.
All the individual concept definitions from the previous phase are merged ready for the next phase. Often this phase produces a single table – the 'master table' or 'analysis ready table'.

Phase IV: Analysis – the core focus of the project.
Applies relevant technical methods to the assembled data to produce analytic results that will answer the research question.

Phase V: Output – the creation of results for dissemination.
This phase includes the application of confidentiality rules to the analytic results so that they are safe for circulation.

We recommend using these phases to plan any integrated data project. This provides several advantages:

- Having clear inputs and outputs from each stage enhances the robustness of the work. Correctness can be validated at the end of each stage, if an error is identified it is simpler to locate and correct its cause within a phase.
- Dividing the project into separate phases reduces the complexity of each phase. This lowers the mental effort required by the researcher. It also makes revisions and peer review easier.
- Using the same set of phases for plannings creates consistency across projects. This simplifies the effort of running a project and makes it easier to reuse material between projects.
- Having an outline of the project phases at the beginning helps ensure that time is allocated for every phase of the work. This allows for more accurate estimation of project timeframes and reduces stress when deadlines approach.

Outline how source data becomes ready for analysis

Data preparation (Phase II and Phase III, above) merits deliberate thought for integrated data projects: it can be three quarters of the effort for an analysis. Poor data preparation can cause projects to fail before any analysis occurs.

A method we have found useful for streamlining data preparation is demonstrated in Table 3. We start in the right most column, listing out all the measures that will be needed for the analysis. For each measure we complete the remaining two columns with information on what source data will be used and what concepts will be extracted from the source data and assembled together.

Table 3: Example of planning process from source data to analysis ready

Source for definition		Column at assembly		Measure for analysis
Residential population	→	Resident	→	Residence indicator
Personal details	→	Year of birth	→	Age
Border movements	→	Arrival into NZ	→	Returning expatriate indicator
Birth records	→	Born in NZ	+	

Inland Revenue	→	Total annual income	→	Average monthly income
		Months employed	+	

In some cases, no single data source will have the required measure. So instead of preparing the required measure directly, we instead prepare data from which we can calculate the required measure. For example, Table 3 shows that a returning expatriate indicator is required as a measure for analysis. This measure is not available in source data. Instead, we initially assemble native born from birth records and arrivals to the country from border movements. During our analysis, these are combined to indicate whether a person is a returning expatriate.

Limit scope to control complexity

For any analytic project, as the complexity of the project increases the project becomes slower to complete and more vulnerable to unexpected challenges. In general, integrated data projects grow in complexity as:

- The number of data sources being combined increases.
- The number of measures being created increases.
- The unit of analysis changes (for example, converting personal income to household income).
- The more time periods that are analysed.
- The number of collaborators increases.

Of these, the number of data sources a project uses tends to have the greatest effect on the complexity of a project. Reducing scope to measures that can be constructed from a small number of data sources is a simple yet effective option to minimise a project's complexity.

For researchers who are new to working with integrated data, we recommend starting with a task that uses at most three sources as inputs. This is a size that it is practical to work through in a few weeks full time. Even if you intend to conduct a much larger project, a small end-to-end analysis will provide you with valuable experience and is also short enough that you can reflect and learn from the entire process. If you are planning a large analysis, consider starting with a simplified version of your full project, then you can build upon the initial work, following a familiar pattern but adapting it for the broader scope.

Limiting the scope of an integrated data project is not just a technique for new researchers. Even experienced integrated data researchers will choose to restrict the scope of their analyses to ensure that it can be completed within deadlines.

Communicate results with meaningful comparisons

The goal of research is seldom just the production of analytic results. Much of the impact of research comes from communicating these results in a way that leads to new understanding.

When communicating your results, it is often effective to use a small number of key tables and charts that support your main points. Even if you have conducted much more analysis, focusing your communication on a few key points helps ensure your audience concentrates on the details that are most important.

What makes a table or chart insightful? Most insights arise from comparison. Hence when producing results, you should also plan to include comparison values. For example:

- In a time-series analysis, we might compare individual points against the trend line. This shows whether individual points are consistent with the trend, or whether the latest observations have varied from the norm.
- When studying a specific sub-population, we might compare their results against equivalent numbers for the entire population to judge if the sub-population is doing better or worse.
- Performance reporting often compares current status against an expectation. This comparison assists decision makers to know whether activity is on target or requires corrective action.

In some cases, comparison may be implicit. For example, reporting the average income for people with a degree may be meaningful if your audience already knowns the average income in New Zealand and can compare the value you report against their existing knowledge. But it is often better to make the comparison explicit. This guides how your audience understands your results. Otherwise, you risk your audiences' conclusions being determined more by their personal opinion and experience rather than the facts of your research.

Document your structure

It is rare that any piece of analytics or research is done once and then discarded. It is much more common that work needs to be revised in response to feedback or repeated as data is updated. Even bespoke research projects will have components that are useful to reuse in future projects.

Documentation is a major factor for determining the ease with which a project or a piece of code can be reused. This is true regardless of whether the reuse is:

- The original analyst returning to the piece of work after six months.
- A colleague who steps into the project when the original analyst is sick or on leave.
- An analyst who takes over the work when the author moves organisation or role.

There can be a perspective that documenting a project will take significant time and effort. However, attempting to document every detail is not required. The goal of documentation is to capture the key points to give anyone reviewing the project a good foundation from which to learn. This only requires three things:

1. **Document the overall structure of the work.**
   This includes the purpose of the project, its agreed outputs, any reference points, and the sequence for running different files. A good length for this kind of documentation is two pages.
2. **Start code files with a header containing some key information.**
   This could include the date the file was created, the author, and the purpose of the file. Having

this information up front makes it much easier to identify which files are relevant and how files fit into a larger project.

3. **Divide code into sections and give each section a title.**
   Section titles require a trivial amount of effort to write but make it much easier to plan and revise code files.

Creating this kind of documentation requires minimal effort. Section titles and file headers take at most five minutes per file but can save hours of effort navigating within and across files. Writing overall documentation can be done in an hour – faster if you scan handwritten notes or diagrams.

Figure 1: Example project structure documentation

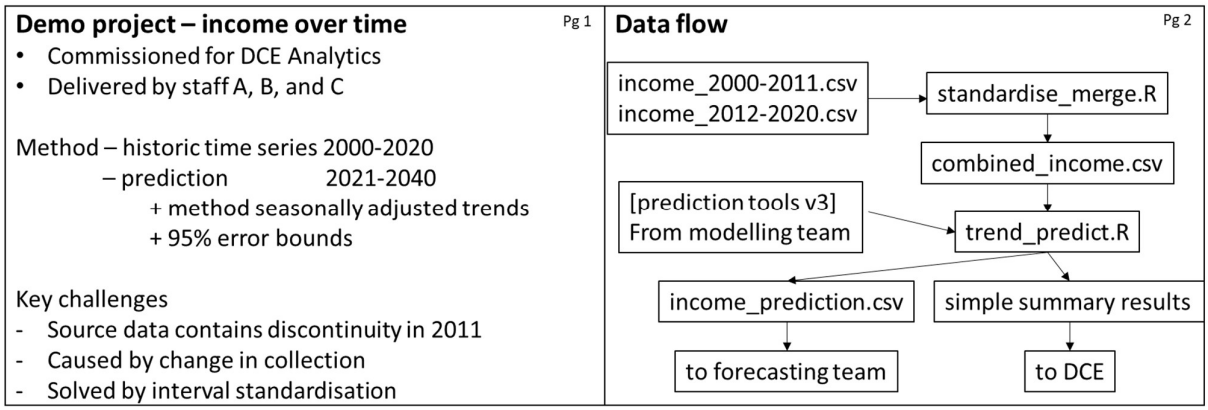

Figure 1 provides a simplified example of documenting a project structure. The key information is easy to see from this example, as are the links between the different project files. While the information in Figure 1 might seem trivial to someone familiar with the work, it is often very difficult information to deduce if starting from just a folder of code files.

Understand your tools and materials

Part of the skill in any profession is familiarity with the tools and materials of the job: knowing their intended purpose, where they best perform, and how to use them safely.

Choose programming language to suit the task

Programming languages are designed for specific purposes. While it is possible to do most tasks in most languages, the better your programming language is aligned with the needs of your project, the easier it will be to complete your project.

When selecting a programming language for a task, we recommend considering the amount of memory and processing power you will require. Complex calculations are best done by statistical languages on limited data loaded into memory. Larger amounts of data are often best handled via database languages where you are limited to less complex calculations. Even larger amounts of data will require deliberate “Big Data” solutions.

Another factor to consider is ease of collaboration. Working in a programming language that is shared with colleagues increases the project’s reach and makes it easier to get assistance.

There is also no need to limit yourself to a single programming language. Using two or more together allows you to select the best one for each subtask. The main thing this requires is a way to export data from one language so that it can be imported into the next (and almost every programming language can read and write data from CSV files).

Data structures

A key feature of data is how it is structured. Different structures are suitable for specific purposes. We highlight three structures that are common when working with integrated data.

1. **Rectangular data** is structured with one row per unit and one column per measure. Such tables are often named according to the unit. For example, a rectangular hospitalisation event table would have one row per hospitalisation (with each column containing a fact about the hospitalisation – date, hospital, age of patient, etc.).
2. **Long-thin structure** is most suited for submitting files for output checking. This is where all values of the same type, and subject to the same rules, are located in the same column. This allows for checking of entire columns in one go, making for a faster process.
3. **Display structures** are whatever makes the most sense for a user to view or plot results. This will vary by application and analyst, but a common example is two-dimensional tables where comparisons can be made across both the rows and columns.

Most data sources have a rectangular structure but differ in the type of unit they describe. A big part of data preparation is combining data together in a rectangular format with consistent units.

Table 4: Example of long-thin and display structures

Long-thin structure

Age	Island	Count
Younger	North	1111
Younger	South	2222
Older	North	3333
Older	South	4444

Display structure

Age \ Island	North	South
Younger	1111	2222
Older	3333	4444

Table 4 shows the same information in long-thin and display structures. Pivot tables in Excel allow easy transformation from long-thin format to other formats. Hence, we encourage researchers to produce their outputs in long-thin format, and then rearrange to any preferred structure after results have been approved for release from the data lab.

Build familiarity with the data you depend upon

Confident and correct use of data depends on researchers understanding the data they work with. Data dictionaries provide a foundation for this understanding but are not a complete solution. There are two further approaches we recommend researchers undertake:

- **Background reading helps ensure researchers know the context of the data.**
  This is more important when researchers have not experienced that context themselves. Consider reading both descriptions of the system that generated the data, and reviewing examples where other researchers use the same data. For example: if a researcher was looking at court data but had never interacted with the justice system themselves, then reading about the administrative process of a court case would give them more understanding of how the data reflects real world events.
- **Hypothesis testing provides a way to verify understanding of the data.**
  As a researcher forms their understanding of new data, it is common for to go through a stage whether there are things they suspect are true but do not know for certain. We refer to these things as hypotheses. To ensure a thorough understanding of new data it is recommended to check the data for evidence to support or disprove each hypothesis. For example: suppose we used data where each record has a referral and an appointment date. It seems reasonable to expect that the referral date is always before the appointment date – this is a hypothesis. Rather than assuming the dates act as we expect, we would check this by counting all records where the dates are in the reverse order. If the hypothesis is correct, then the count should be zero.

The breadth of data that can be integrated means that becoming familiar with new data is a reoccurring activity for researchers. Becoming comfortable with these two approaches will be useful throughout a career.

Practice before embarking on a full project

Like learning any new skill, it is often wise to start simple and expand as your confidence increases. To support this kind of learning, this document includes an appendix with three basic analyses that can be conducted within the IDI. Along with the IDI Exemplar Project, these analyses are a good size for new researchers to explore as a starting point when beginning to work in the IDI.

Appendix: Simple IDI learning analyses

This appendix contains three simple IDI learning analyses. They have been designed to be completed using only SQL. For compactness, these are written as notes rather than as detailed instructions.

Analysis 1: Age group mobility

The purpose of this analysis is to understand how many people move between regions during their transition to adulthood.

The learning goals for this analysis are:

1. Defining people and time periods as part of the analysis process.
2. Become familiar with core IDI tables.
3. Become familiar with spell-based data structures.
4. Preparing results for output checking.

Study population

A study population is a list of unique snz_uid values that represent the people we are interested in studying. As we are studying mobility, this requires location information at two points in time. Hence defining our study population requires both *who* (the list of snz_uid values) and *when* (the dates).

Review the personal_details table in the data schema.

- How might this table assist?
- Only year of birth and month of birth is available. Why is day of birth not provided?
  Because this is identifying information, and the IDI has had identifying information removed.
- It would be useful to have age in our analysis, but this table does not have age. Why?
  Age depends on what time you are conducting your analysis. Once you choose a time for your analysis then you can calculate age.

We will use the personal_details table to define our study population. Using the year of birth and month of birth information, calculate an approximate date that each person turns 15 and 25 years old. We will use this pair of dates as the transition to adulthood and will compare region at age 15 against region at age 25.

Location

Measuring mobility requires location information at two points in time. For address information look at address_notification table.

- How is table arranged? Look at the data for a single snz_uid to understand better. Try several different snz_uid until you understand. One row per person, per address spell.
- What is a spell? A time period with a start and end date.

To join this table to our study population we need to join on two conditions:

1. Records in both tables have the same snz_uid (the records correspond to the same person).
2. The date that each person turns 15 or 25 is between the start (notification) and end (replacement) date of the address.

Assemble together

We are now ready to combine the previous steps to make a table for analysis. Starting from our study population, we can join location information once based on date at age 15, and then a second time based on date at age 25. Save the result into the Sandpit for subsequent analysis. The end result should have columns like this:

- snz_uid
- approximate date of 15th birthday
- approximate date of 25th birthday
- region at date 1
- region at date 2

This data structure is called a cross-section. One row per person. It should be complete – every row should have a pair of birthday dates and regions. Check this now.

This is a good point to save all your code and delete any temporary Sandpit tables. Have you added notes or explanations to your code files? If you reopen them in a month, how will you know what they were for?

Analysis

Now that we have a table prepared, we can undertake our analysis. Mobility occurs when people start in one region and end up in another region.

- Between ages 15 and 25, how many people move region?
- How many people remain in the same region?
- Which regions are the most common to move out of? Or to move into?

As SQL only produces tables of numbers, copying these into MS Excel is an effective way to visualise them. The easiest way to transfer data: right click on results > select all > copy with headers > paste in Excel. Pivot tables can be a useful way to rearrange summarised data from SQL once it has been copied into Excel.

Don't copy all the raw data into Excel. Copying in large amounts of data can cause Excel to freeze or crash. It is more efficient to get SQL to prepare summarised data and then copy the summary into Excel for visualisation.

Write notes as you go. What insights did you gain? What code and what visualisations did you use to get those insights?

Output

Look back at your analysis, what were the results that produced the *most useful* insights? These are the results you should submit for output checking. All results that are removed from the data lab must go through output checking by Stats NZ to confirm they are safe for release.

The easiest results to output are numeric tables containing counts and totals. All the rules for outputting results from the data lab are covered in the Microdata Output Guide. There are several key rules that are most important:

- When counting people, groups of size less than 6 must be suppressed.
- When counting people, counts must be randomly rounded to base 3 (RR3) *after* suppression.
- When giving a total for a group of people (e.g. total income), there must be at least 20 people whose individual values combine to make the total.

Once you have prepared your output submit it to Stats NZ for checking. If Stats NZ staff cannot confirm that it meets all the rules, then you will have to fix your submission and resubmit.

- There is an output checking tool that you can use to pre-check your own submission. This assesses your submission against the most common rules.
- Another way to practice is to have a colleague review your prepared file. Can they understand what you have done without any explanation? The easier you can make it for Stats NZ staff to understand your file, the faster it is for them to check it and give it back to you.

Extension

How would you adapt your analysis to handle each of the following:

- Suppose we wanted to change from regional council areas to territorial authority areas. How can you adapt your code to quickly reproduce your results for this different choice of areas?
- What about replacing age 15 and age 25 with age 18 and age 22?
- What about limiting your analysis to a single sex/gender or a single ethnicity?

Analysis 2: Income over time

The purpose of this analysis is to understand how income patterns change over the life cycle.

The learning goals for this analysis are:

1. Good analysis process
2. Checking a dataset prior to analysis
3. Become comfortable working with time series data
4. Preparing results for output checking

Study population

A study population is a list of unique snz_uid values that represent the people we are interested in studying. Looking at patterns in people's incomes requires us to observe their income in multiple years. As it is difficult to observe income for people who do not live in New Zealand, the simplest option is to limit our analysis to people who are resident in the country for every year of our analysis.

- Examine the snz_res_pop table. This provides a list of snz_uid and the dates at which those people are resident.
- Summarise the table by year. What date range does the table cover? How many people are resident each year? Is this consistent with what you know of the population?
- Decide on a year range that you will analyse. For example, the three years from 2021 to 2023.
- Calculate how many people are resident in **all** the years you have chosen. This number must be smaller than counting the number in each year. If you extend the year range, then the number of people should reduce further because you are adding an additional condition.

Write and save a list of snz_uid for people who are resident in every year for the range you have chosen. This list will define our study population.

Income

There are several sources of income information in the IDI that we could use. Inland Revenue is the key source for income data. But Census and surveys often collect self-reported income.

- In the data schema there are some pre-prepared income tables by calendar and tax year.
- Open and inspect all these tables. How do they compare? To understand these tables you may find it easiest to pick 1-3 snz_uid and compare their records between the tables.
- Let's use the calendar-year summary table. This has more columns than we require so we only need to query some of the columns. At a minimum we need snz_uid, year, and total income.
- When defining our resident population we chose a specific range of years. We need to make sure that we are only looking at income from these same years.

Demographic characteristics

To understand changes in income over the life cycle we need to know about people's ages when they earned their income. Review the personal_details table.

- How might this table assist?
- Only year and month of birth is available. Why is day not provided?
  Because this is identifying information, and identifying information has been removed.
- It would be useful to have age in our analysis, but this table does not have age. Why? Because age changes each year, most databases do not store age and instead store date of birth.

Assemble together

We are now ready to combine the previous steps to make a table for analysis. Starting from our resident population, we can join income and demographic data on to this, and save the result into the Sandpit for subsequent analysis. The end result should have columns like this:

- snz_uid
- calendar year
- total income in calendar year
- year of birth
- age in calendar year

This data structure is called panel data. One row per person per time period. It should be a complete panel – every person appears in every time period and every time period appears for every person. Check this now.

What happens in your assembly if a person does not earn any income in a calendar year? Is there no row in the data? Is there a row with NULL/missing income? Is there a row with zero income? For this analysis the ideal data has a row with zero income this ensures we have a complete panel and simplifies our analysis.

This is a good point to save all your code and delete any temporary Sandpit tables. Have you added notes or explanations to your code files? If you reopen them in a month, how will you know what they were for?

Analysis

Now that we have a table prepared, we begin by checking the quality of the data in the table.

- Does everyone have a year of birth / age? How many people do not?
- What does the income distribution look like in each year? In it similar across years? If a year looks different, what might have affected income in that year? (e.g. pandemic lockdowns).
- Check income for extreme values. Are there negative incomes? What are the largest incomes? Do these values make sense, or should they be excluded from analysis?
- What is the mean or median income in each year. Is this consistent with official statistics of average incomes?

Changes in income can occur for many reasons. For this analysis we are interested in how income changes as people age.

- What does average income look like at each age?

- Graduation and retirement are significant times of change for many people. Can you observe significant changes in income around the ages where graduation and retirement tend to occur?

The focus of this analysis is changes in income between years. Calculate changes in income between consecutive years. Classify these changes by the size of the increase or decrease.

- What proportion of people have increases or decreases in income each year?
- How does this proportion change with age?

As SQL only produces tables of numbers, copying these into Excel is an effective way to visualise them. The easiest way to transfer data: right click on results > select all > copy with headers > paste in Excel. Pivot tables can be a useful way to rearrange summarised data from SQL once it has been copied into Excel.

Don't copy all the raw data into Excel. Copying in large amounts of data can cause Excel to freeze or crash. It is more efficient to get SQL to prepare summarised data and then copy the summary into Excel for visualisation.

Write notes as you go. What insights did you gain? What code and what visualisations did you use to get those insights?

Output

Look back at your analysis, what were the results that produced the most useful insights? These are the results you will need to submit for output checking.

All the rules for outputting results from the data lab are covered in the Microdata Output Guide. There are several key rules that are most important:

- When counting people, groups of size less than 6 must be suppressed.
- When counting people, counts must be randomly rounded to base 3 (RR3) after suppression.
- When giving a total for a group of people (e.g. total income), there must be at least 20 people whose individual values combine to make the total.

When preparing output, you may need to create age groups or income bands so that when you produce a summary table of counts there are enough people in each group to output.

Extension

How would you adapt your analysis to handle each of the following:

- Analyse income from only wages and salaries instead of total income?
- Change the time period in your analysis to add or remove a year?
- Consider changes in income by sex/gender, ethnicity, or industry of employment?

Analysis 3: Consistent cultural affiliation

The purpose of this analysis is to understand how often children report the same ethnicity as their parents.

The learning goals for this analysis are:

1. Comparing data sources and ensuring consistency
2. Conducting analysis at relationship level
3. Data cleaning
4. Preparing results for output checking

Defining relationships

To compare the ethnicities of children and parents we need links between them. Take a look at the personal_details table in the data schema.

- Observe how this table contains both ethnicity information and parent snz_uid values.
- Ethnicity information is spread across multiple columns. Why is this? Because people can have more than one ethnicity. Check how often people have multiple ethnicities.
- Ethnicity information can exist at three different levels of detail. What level of detail is available in this table? If more than one level of detail is available, which level will you use?
- If this table contains all the information we need, why is further data preparation needed? Because the data is not structured in the best format for our analysis. We need to compare parent and child ethnicity – this needs both ethnicity values as part of the same record.

Begin by defining relationships. Write and save a list of pairs of child_snz_uid and parent_snz_uid. Where a record has two parent_snz_uid values, then one record in the personal_details should become two records in our relationship table.

Every parent-child pair should be unique, but the same snz_uid may appear multiple times in the child_snz_uid column and multiple times in the parent_snz_uid column. Some snz_uid values may appear in both columns.

- Why are individual snz_uid values not unique? Because the unit for our analysis is not the individual but the parent-child pair.
- How would you check how many parents appear more than once?

Assemble together

With our parent-child relationship table and demographic information in the personal_details table, we are now ready to assemble a table for analysis. Starting from parent-child relationships, we can join ethnicity for parents, and then for children, on to the table. Save the result into the Sandpit. The end result should have columns like this:

- parent_snz_uid
- child_snz_uid
- parent's ethnicities

- child's ethnicities

This is a good point to save all your code and delete any temporary Sandpit tables. Have you added notes or explanations to your code files? If you reopen them in a month, how will you know what they were for?

Analysis

We want to know whether children share their parent's ethnicity.

- Add an indicator to your table for whether the parent and child have at least one ethnicity the same. Add another indicator for having all ethnicities the same.
- Using these indicators, how many children share ethnicities with their parents?
- How many children share ethnicity with one parent but not with the other?
- Are certain ethnicities more likely to be shared by the parent and their child?

We are most interested in the proportion of parent-child relationships that are consistent or inconsistent. However, it is easiest to produce counts in SQL. Try coping these counts from SQL into Excel to calculate proportions or visualise them. The easiest way to transfer data: right click on results > select all > copy with headers > paste in Excel.

Make sure you write notes as you go. What insights did you gain? What code and what visualisations did you use to get those insights?

Output

Look back at your analysis, what were the results that produced the most useful insights? These are the results you should submit for output checking. All results that are removed from the data lab must go through output checking by Stats NZ to confirm they are safe for release.

The easiest results to output are numeric tables containing counts and totals. All the rules for outputting results from the data lab are covered in the microdata output guide. There are several key rules that are most important:

- When counting people, groups of size less than 6 must be suppressed.
- When counting people, counts must be randomly rounded to base 3 (RR3) after suppression.
- When giving a total for a group of people (e.g. total income), there must be at least 20 people whose individual values combine to make the total.

Once you have prepared your output submit it to Stats NZ for checking. If Stats NZ staff cannot confirm that it meets all the rules, then you will have to fix your submission and resubmit.

- There is an output checking tool that you can use to pre-check your own submission. This assesses your submission against the most common rules.
- Another way to practice is to have a colleague review your prepared file. Can they understand what you have done without any explanation? The easier you can make it for Stats NZ staff to understand your file, the faster it is for them to check it and give it back to you.

Extension

- Differences in ethnicity could vary with the quality of the data source. Census is often considered the best source for self-report of demographic information.
  How would you adapt the analysis to use only ethnicity information from a census?
- Suppose you wanted to change from parent-child relationships to grandparent-child relationships.
  How would you adapt your method to do this?
- Where children are young, parents may fill out forms for them. This means that consistent answers may reflect only the parent's perspective, not consistency between the parent's and child's perspectives.
  How would you filter your data to include only children who are old enough to decide on their own ethnicity separate from their parents?

Alternative option: iwi affiliation

The original design of this practice analysis was written for iwi analysts and used iwi affiliation instead of ethnicity to consider consistent whakapapa. However, use of iwi affiliation data is subject to additional constraints and considerations, so we do not recommend it as a practice data source for researchers in general. Below we provide some guidance on adapting this analysis to consider iwi affiliation where it is appropriate for a researcher to do so.

Census data provides the best source of iwi affiliation data in the IDI. Creating the most complete measure requires combining data across more than one census.

- Review the individual census tables and their available columns. Locate columns that contain iwi affiliation information.
- People can affiliate to more than one iwi. How is this handled in the data? There are three likely options: multiple rows per person, multiple iwi columns, or a single delimited column.
- Iwi information may be stored as numeric codes. You will need to locate metadata that will translate numeric codes into recognisable names.

When reviewing metadata to understand numeric codes it is important to consider that the classification used for iwi may have changed between censuses. Comparing the classifications:

- Is there a common classification that you can use for both sources?
- With iwi-specific knowledge, can you determine which codes should be combined when working with more than one source.

Write and save your prepared iwi affiliation information. This should be a list of snz_uid values and the iwi they affiliate to.

- Where the same person has responded to more than one census there could be duplicate information when combining data sources. How would you check for the presence of duplicates? How will you ensure that your prepared data does not include duplicate info?