Prepared for: Muka Tangata

# Skills forecasting model

## Technical report

October 2024

Scarlatti.co.nz

# Contents

# List of tables

# List of figures

# Overview

## A skills forecasting model for the food and fibre sector

The skills forecasting model is a micro-simulation model that aims to provide depth to workforce demand forecasts beyond simple headcounts or FTE requirements. 'Virtual workers' are tracked through their employment journeys in the target industries, allowing the user to evaluate the effect of various scenarios on experience levels, qualification uptake, and more. For a more detailed summary of the model, see page 29.

## Purpose of this manual

This manual is intended to convey the technical information underlying the skills forecasting model for the food and fibre sector. This information should enable technical analysts to interact with the model's back end, including:

- Updating data and refreshing the model
- Defining, constructing, and running custom scenarios within the model simulation
- Updating the user interface with new outputs (e.g., scenarios, 'knobs and sliders').

To that end, the following information is documented in this manual:

- **Data:** An overview of input and calibration data, including descriptions, definitions and sources.
- **Code**: An overview of the Python environment, including packages and files used within the simulation.
- **Model logic:** A description of the logic we have utilised across simulation processes, such as factors influencing workforce exit, identification of seasonal workers, and how capability factors into decisions for recruitment and promotion.
- **Model**: A description of how to run the model, an overview of simulation calculations contained within the model, the process for developing custom scenarios, and how data and simulation outputs are managed.
- **Visualisation and user interface:** An overview of the dashboards, including their structure and content, and how to update existing visualisations or add custom visualisations.

Included in the appendices are:

- **Appendix A**: A full data dictionary

# Glossary of terms

Below is a list of terms used throughout the document and within the model. Full detailed definitions, see Appendix A: Data dictionary.

| Term | Description |
|---|---|
| **Age** | Age group of the individual worker |
| **Attributes** | A quantifiable characteristic or feature of an entity (e.g., age, qualification, role level) |
| **Capability model** | A model quantifying a worker’s capability in a role that is used to inform who is hired and/or promoted to fill a vacancy |
| **Employee** | A wage or salary earner |
| **Employer** | A business owner |
| **Employer size** | The number of employees in a company |
| **Entry point** | The origin of the individual as they enter the food and fibre sector |
| **Ethnicity** | Mutually exclusive ethnic groups (ie., an individual is only counted for one ethnicity) |
| **Food and fibre tenure** | The number of months since an individual entered employment |
| **Gender** | Gender of the individual |
| **Industries** | ANZSIC06 names of industries relevant to Muka Tangata |
| **Industry group tenure** | The number of months since an individual entered employment within a specific industry group |
| **Industry groups** | Muka Tangata industry groups |
| **Intrinsic ability** | Part of the capability model, this is a random normally distributed number that is assigned to each worker acting as a proxy of motivation and ability |
| **IPFP** | A statistical technique known as Iterative Proportional Fitting Procedure |
| **Job levels** | The job level of the individual, which is determined by a combination of their role level, employment status and work status |
| **MPI sectors** | Food and fibre industry divisions used by MPI |
| **New entrant** | An individual with less than one month of tenure in the food and fibre workforce |
| **NZQF level** | The New Zealand Qualifications Framework is a comprehensive, nationally endorsed framework that lists all quality-assured qualifications in New Zealand, split into 10 levels reflecting degree of complexity and time spent on studying |
| **Role level** | Workforce classifications representing role hierarchy based on individual capability and experience |

| | |
|---|---|
| **Self-employed** | An individual who earns shareholder income, company director income, sole trader income, or partner income in their tax return, but does not employ staff |
| **Time step** | A discrete interval of time within the simulation model, specifically one month in duration, starting January 2000 to December 2050 |
| **Training uptake model** | |
| **Work status** | A classification of an individual's participation in the workforce |

# Section one: Input and calibration data

## Overview

This section delves into the structure and content of the input and calibration spreadsheets. These spreadsheets serve multiple purposes:

- Store data tables that feed into the model simulation
- Provide an interface for users to easily adjust and combine these tables for various scenarios
- Contain a calibration data database. This database is essential for comparing model outputs with existing, relevant data.

### Input spreadsheets

Four input spreadsheets were developed to feed data into the model simulation, with each tailored to fulfil a specific role. Below is a breakdown of these spreadsheets and how they contribute to shaping the model's output:

- **Initial population:** contains data detailing the characteristics of the population at the start of the simulation period (year 2000).
- **New entrants:** provides data and scenarios to describe the sources and attributes of new entrants to the workforce.
- **Overall parameters:** acts as the control centre for the simulation, storing settings including scale, timeframe, attributes and processes, and the probability of exiting the workforce.
- **People needed:** provides data to describe different scenarios for workforce needs over time.

### Calibration spreadsheet

The separate calibration spreadsheet is distinct from the input spreadsheets in its purpose and use. While input data are directly read by the model to generate outputs, calibration data serve as a reference point for testing, tuning, and validating the model's outputs.

- Calibration: stores a range of calibration data tables broken down by various groupings to test accuracy and consistency of model outputs

## Spreadsheet structure

While each spreadsheet is uniquely tailored with specific tabs, the functionality of these tabs is standardised through a color-coded system:

Orange tabs **typically include the disclaimer and navigation pages.** These tabs are pivotal for setting the foundational understanding and navigation within the spreadsheet.

Green tabs **are for engaging with and processing data.** These tabs allow users to customise the simulation inputs. For example, select specific age groups, gender, or ethnic groups to define a population.

Blue tabs **include input and/or calibration data.** These tabs are dedicated to housing the raw data for simulations and model calibrations.

## Data details

The details of the data each spreadsheet contains, including the name of the tabs, description and definition of the data, and data sources can be found in the tables below:

- Initial population
  - File name: model_input_population
  - Last update: Oct 2024
  - Data details on page 11 in Table 1.
- New entrants
  - File name: model_input_new_entrants
  - Last update: Oct 2024
  - Data details on page 12 in Table 2.
- Overall parameters
  - File name: model_input_overall_parameters
  - Last update: Oct 2024
  - Data details on page 14 in Table 3.
- People needed
  - File name: model_input_people_needed
  - Last update: Oct 2024
  - Data details on page 18 in Table 4.
- Calibration
  - File name: model_input_calibration_data
  - Last update: Oct 2024
  - Data details on page 19 in Table 5.

Table 1: Initial population

| Tab | Description and data breakdown | Source |
|---|---|---|
| disclaimer | The relevant Stats NZ IDI disclaimer | Stats NZ |
| navigation | A list of the tabs contained in the spreadsheet, along with a description and hyperlink for each | Scarlatti |
| ig x nzsced x nzqf | Highest qualifications of the initial workforce, by industry group. Derived from equilibrium simulation outputs. | Scarlatti |
| ig x emp | De-seasonalised one-dimensional demographic breakdowns of the initial workforce, by industry group and employment status. | Stats NZ IDI, Scarlatti |
| ig x emp x age | | |
| ig x emp x gen | | |
| ig x emp x eth | | |
| ig x emp x ff_ten | | |
| ig x emp x visa | | |
| age x ff_ten x ig_ten | Correlation of age, whole simulation tenure, and industry group tenure, for the initial workforce. | IDI, Scarlatti |
| ig x emp x work x age | Correlation between various demographics, employment status, and full/part time. Based on 2018 census data. | Stats NZ IDI, Scarlatti |
| ig x emp x work x gen | | |
| ig x emp x work | | |

Table 2: New entrants

| Tab | Description and data breakdown | Source |
|---|---|---|
| disclaimer | The relevant Stats NZ IDI disclaimer | Stats NZ |
| navigation | A list of the tabs contained in the spreadsheet, along with a description and hyperlink for each | Scarlatti |
| controls | Controls for talent pool splits by industry group for nonseasonal and seasonal workers and managing the border status by year and month | Stats NZ IDI, Scarlatti |
| ig x border x tp x time non-seas | Proportion of non-seasonal new entrants x industry group x border status x talent pool x year x month | Stats NZ IDI, Scarlatti |
| ig x border x tp x time seasona | Proportion of seasonal new entrants x industry group x border status x talent pool x year x month | Stats NZ IDI, Scarlatti |
| qual of new entrants | Counts of qualifications for new entrants x industry group x talent pools x NZSCED x NZQF level | Stats NZ IDI, Scarlatti |
| nzsced x nzqf x time | Counts of graduates by industry group x talent pool x NZSCED x NZQF level x year x month | Stats NZ IDI, Scarlatti |
| New qual graduates output | Graduates each month from a new qualification, as specified in the Qualifications tab of the overall parameters spreadsheet. This is used for scenarios where a new pre-employment qualification is introduced. When these scenarios are not being ran, the *n* columns should be all zeros. | Scarlatti |
| snz year x age | Annual population projections x age group from Stats NZ | Stats NZ IDI, Scarlatti |
| tp x ig x emp x age perc | Proportion of new entrants x talent pool x industry group x employment status x age group x year | Stats NZ IDI, Scarlatti |
| tp x ig x emp x gen perc | Proportion of new entrants x talent pool x industry group x employment status x gender x year | Stats NZ IDI, Scarlatti |
| tp x ig x emp x visa perc | Proportion of new entrants x talent pool x industry group x employment status x visa x year | Stats NZ IDI, Scarlatti |
| tp x ig x visa seasonal | Proportion of new entrants x talent pool x industry group x visa x year for seasonal workers | Stats NZ IDI, Scarlatti |
| visa x eth | Proportion of new entrants x visa x ethnicity | Stats NZ IDI, Scarlatti |
| tp x ig x emp x age | Counts of new entrants x industry group x talent pool x employment status x age group | Stats NZ IDI, Scarlatti |

| Tab | | Description and data breakdown | Source |
|---|---|---|---|
| | tp x ig x emp x gen | Counts of new entrants x industry group x talent pool x employment status x gender | Stats NZ IDI, Scarlatti |
| | tp x ig x emp x eth | Counts of new entrants x industry group x talent pool x employment status x ethnicity | Stats NZ IDI, Scarlatti |
| | ind x year x eth x ig seasonal | Counts of seasonal workers x industry x ethnicity x industry group x year (2021 to 2022) | Stats NZ IDI, Scarlatti |
| | ind x year x age x ig seasonal | Counts of seasonal workers x industry x age group x industry group x year (2021 to 2022) | Stats NZ IDI, Scarlatti |
| | ind x year x gen x ig seasonal | Counts of seasonal workers x industry x gender x industry group x year (2021 to 2022) | Stats NZ IDI, Scarlatti |
| | ig x anzsic06 | Muka Tangata industry groups mapped to ANZSIC06 codes | Muka Tangata, MPI, Scarlatti |
| | emp lookup | A lookup table mapping old to new employment status categories | Scarlatti |

Table 3: Overall parameters

| Tab | | Description and data breakdown |
|---|---|---|
| | disclaimer | The relevant Stats NZ IDI disclaimer |
| | navigation | A list of the tabs contained in the spreadsheet, along with a description and hyperlink for each |
| | running options | User controls for how to run and save the simulation. These enables users to:<br>• Select the output mode (none, SQL, point-in-time file, or annually aggregated file)<br>• Specify the output filepath<br>• Specify the scenario ID (only required for SQL output)<br>• Run tests<br>• Adjust the tenure threshold for annual aggregation. |
| | sim scale | User controls for scale of simulation. This enables users to control the scale of the simulation (i.e., simulate every '1' out of 'X' workers) |
| | timeframe | User controls for timeframe and time steps. This enables users to:<br>• Define the initial year x interval (e.g., 2000, q1) and final year x interval<br>• Select time step, either quarter or year. |

| Tab | | Description and data breakdown |
|---|---|---|
| | attributes and processes | User controls for including and/or excluding attributes and processes in the simulation.<br><br>**Attributes** include:<br>• Ethnicity<br>• Gender<br>• Age<br>• Tenure<br>• Part-time<br>• Industry group<br>• Role<br>• Employment status.<br>•<br>**Processes** include:<br>• Withdrawals<br>• Recruitment<br>• Increment age and tenure<br>• Maximum working age<br>• Inter-industry group movement<br>• Promotions.<br><br>Both attributes and processes are included in the simulation by selecting ‘1’ or excluded by selecting ‘0’. |

| Tab | Description and data breakdown |
|---|---|
| exit prob | User controls for exit probability by age group and tenure group, and employment status. Parameters in this tab are fitted using a logistic regression.<br>• **Base:** Determines the default exit probability<br>• **Adjustments:** The exit probability can be further modified to differ according to age group, tenure group, and employment status<br>• **Exit probability:** The probability of a worker leaving the workforce. |
| ig movement | User controls for adjusting movement probability by attribute and between industry groups. Parameters in this tab are fitted using a logistic regression.<br><br>**Movement probability x attribute include:**<br>• Tenure groups<br>• Age<br>• Employment status<br>• Industry groups.<br><br>**Interindustry group movement: Origins and destinations includes:**<br>• Industry groups. |
| emp size | Distributions used to assign employee size. Each table contains the distribution of roles x employer size x industry group. These tables are used in the simulation to assign employer size to employees based on their role and employment status. |
| capability model | Parameters for the utility function used to control promotions. These determine a worker's capability in a role and are used to inform who is hired and/or promoted to fill a vacancy. These parameters can be interpreted as the return that a worker brings a company, measured in dollars per hour. |
| quals | Parameters affecting qualifications. These enables users to adjust the relevance of NZSCED qualifications to each industry group, and the industry group destinations for tertiary graduates of NZSCED qualifications. |
| quals over time | How the uptake of qualifications changes over time. |

| Tab | | Description and data breakdown |
|---|---|---|
| | recruiting parameters | User controls for the percentage of each role being recruited from each source for each industry group. Also includes an under-recruitment adjustment for keeping a proportion of positions unfilled. |
| | recruitment to job levels | Parameters used to control the distribution of recruitment across job levels. |
| | promotions | Parameters used to control the likelihood of promotions. |
| | seasonal workforce | Parameters used to control seasonal workers for each industry group. Parameters include:<br>• Mean start month<br>• Start month standard deviation<br>• Mean contract duration<br>• Contract duration standard deviation<br>• Size relative to permanent workforce. |
| | visas | Parameters on visa categories, durations and transitions. |
| | learner roi | User controls for the return of training in learner salary. |
| | categories | Lookup lists of industry group and region. |

Table 4: People needed

| Tab | Description and data breakdown | Source |
|---|---|---|
| disclaimer | The relevant Stats NZ IDI disclaimer | Stats NZ |
| navigation | A list of the tabs contained in the spreadsheet, along with a description and hyperlink for each | Scarlatti |
| controls | Contains controls used to define the people needed scenario, and a graph to see the resulting scenario | Scarlatti |
| chart | Chart of the people needed for the current scenario | Scarlatti |
| export | Contains the export table read by the simulation engine | Scarlatti |
| projection | Contains a constructed people needed scenario – generated through power query | Scarlatti |
| ig x role x emp x hours x reg | Contains recent regional splits – source table for the power query. **"monthly deseasonalised counts"** | Scarlatti |
| ig x anzsic06 | Muka Tangata industry groups mapped to ANZSIC06 codes | Muka Tangata, MPI, Scarlatti |
| categorical variables | List of the values various categorical variables can take, used as source tables for the power query | Scarlatti |
| business as usual | The scenario uses past performance to describe the food and fibre sector in 2032 | NZIER, MPI |
| increased technology | This scenario envisions a food and fibre sector that takes full advantage of existing and emerging technologies | NZIER, MPI |
| transformed sector | This scenario builds on the scenario of increasing technologies and adds an increased focus on sustainability and high-value products and markets | NZIER, MPI |

Table 5: Calibration

| Tab | | Description and data breakdown | Source |
|---|---|---|---|
| | disclaimer | The relevant Stats NZ IDI disclaimer | Stats NZ |
| | navigation | A list of the tabs contained in the spreadsheet, along with a description and hyperlink for each | Scarlatti |
| | dem x month | Counts of the workforce x industry group x employment status x attribute x month x year. Attributes include:<br>• Age<br>• Ethnicity<br>• Gender<br>• Food and fibre tenure<br>• Industry group tenure. | Stats NZ IDI, Scarlatti |
| | ne demo | Counts of new entrants x industry group x origin x employment status x attribute x month x year group.<br><br>**Attributes** include:<br>• Age<br>• Ethnicity<br>• Gender.<br><br>**Year group** include:<br>• 2005-2009<br>• 2010-2014<br>• 2015-2019<br>• 2020-2024. | Stats NZ IDI, Scarlatti |
| | age x ff ten x ig ten | Counts of the workforce by age x food and fibre tenure x industry group tenure x month x year | Stats NZ IDI, Scarlatti |

| Tab | Description and data breakdown | Source |
|---|---|---|
| ig x emp x emp size | Counts of industry group x employment status x employer size x year | Stats NZ IDI, Scarlatti |
| ig x anzsic06 | Mapping of ANZSIC06 industry to MPI industry groups to Muka Tangata industry groups | Muka Tangata, MPI, Scarlatti |

# Section two: Python code

## Overview

This section provides a detailed overview of the Python environment **(Python 3.12.4)** used to develop the simulation engine. It includes:

- A list of packages used
- The function of all Python files.

## Packages

Table 6 below lists the required packages used by the simulation engine, along with each package's version number and its role or how it is utilised within the engine.

Table 6: Packages used in the simulation engine

| Package | Version | Purpose |
|---|---|---|
| openpyxl | 3.0.10 | Reads Excel spreadsheets |
| pandas | 2.1.1 | Manages dataframes and performs data analysis |
| ipfp | 1.4.4 | Executes the IPFP |
| numpy | 1.26.0 | Handles arrays |
| matplotlib | 3.8.0 | Creates plots for efficient testing |
| statsmodels | 0.14.0 | Fits sub-models using regression |
| emcee | 3.1.4 | Fits sub-models using Bayesian methods |
| SQLAlchemy | 2.0.31 | Uploads simulation results to a database |

## Files

Table 7 below lists all files used by the simulation engine, detailing their file name, descriptions, class information, and purposes (i.e., how the simulation engine is using the file).

Table 7: Scripts used by the simulation engine, and their important components

| File | Description | Class or function | Purpose |
|---|---|---|---|
| data.py | This file loads model parameters and calibration data from the input spreadsheets. Additionally, it contains constants used by the simulation. | Inputs | Dataclass that stores model inputs from the input spreadsheets |
| | | CalibrationMetadat a | Dataclass that stores metadata related to calibration data, such as age range lookups |
| | | CalibrationData | Dataclass that store the calibration data (and the relevant CalibrationMetadata object) |
| | | inputs_from_spread sheet | Function that reads the Overall Parameters and People Needed Input Spreadsheets, creating an Inputs object |
| | | calibration_data_fro m_spreadsheet | Function that reads the New Entrants and Initial Population Input Spreadsheets, creating a CalibrationData object |
| | | running_options | Function that reads the running options from the Overall Parameters Input Spreadsheet (e.g., output mode, whether to perform tests) |
| ipfp_fitting.py | This file executes IPFP to fit the distributions used in this model, as described in the Generating new workers with IPFP section on page 34 | IpfpResult | Dataclass that contains the fitted distribution as a Pandas DataFrame, and a relevant CalibrationMetadata object |
| | | fitted_array | Function that returns a NumPy array of the fitted distribution for a given CalibrationData object |
| | | initial_workforce_ip fp_result | Function that returns an IpfpResult object for the initial workforce |
| | | new_entrants_ipfp_ result | Function that returns an IpfpResult object for new entrants to the workforce |

| File | Description | Class or function | Purpose |
|---|---|---|---|
| | | save_results | Function saving a provided IpfpResult object to a file |
| worker.py | This file contains the worker classes used in the simulation | Worker | Dataclass that represents a permanent worker and its attributes |
| | | SeasonalWorker | Dataclass that represents a seasonal worker (inherited from Worker) |
| | | Workforce | Dataclass that represents the workforce as a whole. Notable attributes are lists of workers and seasonal workers, and a NumPy array storing the size of the workforce. When updating worker attributes or updating workers, we use the methods of this class rather than deal with Worker objects directly |
| saver.py | This file manages saving the simulation results, whether it be to the database, to a file, or not at all | OutputSaver | Function that saves/uploads outputs from the simulation. Save the current workforce by running the asynchronous method OutputSaver.save() |
| | | create_saver | Function that creates an OutputSaver object with a specified output mode |
| scenarios.py | This file contains scenarios used by the simulation | BaselineScenario | File that contains methods calculating probabilities, worker capabilities, and other mathematical functions that form the basis of the model |
| simulation_engine.py | This file runs the agent-based simulation to evaluate the scenarios of interest, as described in the Simulation section | SimulationEngine | Dataclass that contains methods to run the simulation for a provided scenario |
| | | SimulationEngine.run | Function that runs the simulation and saves results (if desired) |

| File | Description | Class or function | Purpose |
| --- | --- | --- | --- |
| | | setup_model | Function that sets up the model and creates a SimulationEngine object |
| | | SimulationEngine.advance_workforce | Advances the workforce one time step, stepping through the main cycle pictured in Figure 1 |
| logger.py | This short script sets up the logging used by the program and is called before running the simulation.<br><br>Log files have a file path with the following format: “YYYY-mm-dd_HH-MM-SS.log”.<br><br>To use logging, run setup_logger at the at the top of the program, then use Python’s built-in logging module to create logs. | setup_logger | Function that initialises the logger, creating the log file and setting up formatting |
| | | error | Function that logs and prints an error message, raising an exception |
| | | warning | Function that logs and prints a warning message |
| main.py | This script is intended to be the one run from the terminal, and provides several command line options for convenience | main | Script that processes command line arguments and running options from the input spreadsheets, then runs the simulation or tests as desired |
| secret.py | This file holds the connection parameters used to connect to a SQL database which can be used to store the outputs | - | - |
| job_levels.py | This file describes the various possible job levels | JobLevel | Dataclass that describes job levels, including its role level, employment status, and whether it is full or part time |
| | | VALID_JOB_LEVELS | Immutable tuple that contains all valid job levels that can be assigned to workers |
| industry_groups.py | Contains classes that describe the industry groups, and track the capability required for the job levels | IndustryGroup | Class that describes an industry group, contains methods to represent the industry group, and track capability requirements |

| File | Description | Class or function | Purpose |
|---|---|---|---|
| | | init_industry_groups | Creates all the IndustryGroup objects |
| | | capability_range_for_promotion | Calculates the acceptable capability range required for an externally recruited worker to enter a job level (determined from historic promotions) |
| | | update_job_level_capabilities | Updates the job level capability ranges using the capability function and job level from a successful promotion |
| recruiter.py | This file handles filling vacancies through promotion and recruitment. Also handles changes of employer. | Recruiter | Dataclass that contains methods to fill vacancies and change employers |
| | | Recruiter.fill_vacancies | Function that runs the entire recruitment and promotion process to fill all vacancies |
| | | Recruiter.perform_promotions | Function that promotes workers of the highest capability to fill vacancies passed to it |
| | | Recruiter.perform_external_recruitment | Function that performs external recruitment to fill vacancies passed to it, generating workers from appropriate talent pools with appropriate capabilities |
| seasonal_worker_model.py | This file models seasonal workers in a parallel, but linked, model | SeasonalWorkerModel | Function that generates, advances, and removes seasonal workers in the workforce |
| | | SeasonalWorkerModel.advance | Function that advances the seasonal worker model one time step |
| talent_pools.py | This file contains code describing and implementing talent pools | NonSkillsBasedTalentPool and its children | File that describes talent pools, containing methods to generate workers |

| File | Description | Class or function | Purpose |
|---|---|---|---|
| | | talent_pools | Function that constructs and returns the various talent pools |
| time_class.py | This file manages time | Time | Function that stores as a pair of integers, the year and the month index. Contains methods to advance time by one time step, and calculate the time as a float |
| attribute_types.py | Contains various classes used to represent attributes of workers | Ethnicity | Enum, contains the different possible values for ethnicity |
| | | Gender | Enum that contains the different possible values for gender |
| | | VisaCategory | Class that describes a visa category |
| | | Visa | Class that describes a single visa given to a worker. Contains methods to assist in management and representation. |
| input_spreadsheet_config.py | This file stores locations of data in the input spreadsheets | N/A | |
| qualifications.py | This file initialises qualifications, and contains classes to store and manage qualifications | Qualification | Class that describes a single qualification, or qualification category. Contains methods to assist in management and representation in the output files. |
| | | qualifications | Initialises and returns the list of qualification objects, taking parameters from the overall parameters input spreadsheet as inputs |
| visas.py | This file models visas, generating visas and managing transitions between them | VisaManager | Manages and models visas |
| | | VisaManager.advance_visas | Advances the visas of all workers in the workforce |

| File | Description | Class or function | Purpose |
| --- | --- | --- | --- |
| | | VisaManager.transition_visas | Transitions a worker's visa to a new category once the old visa expires |
| | | VisaManager.new_visa | Generates a new visa of the passed category |
| training_uptake_model.py | This file models on-the-job training | TrainingUptakeModeller | Models on-the-job training |
| | | TrainingUptakeModeller.run_training_uptake_model | Executes the training uptake model for one time step |
| | | TrainingUptakeModeller.award_qualifications | Awards qualifications to workers when they complete the qualification |
| movements.py | Models changes in industry group and region of employment | MovementModeller | Models changes in industry group and region of employment |
| | | MovementModeller.run | Runs the movement modelling for every worker in the workforce and one time step |
| | | MovementModeller.model_movement_for | Runs the movement modelling for one worker and one time step |
| redundancy.py | Models redundancies caused by shocks in demand | RedundancyModel | Models redundancies, calculating which workers are made redundant and removing them from the workforce |
| | | RedundancyModel.model_redundancy | Runs the redundancy model for the current time step |

| File | Description | Class or function | Purpose |
|---|---|---|---|
| | | RedundancyModel.number_to_make_redundant | Calculates the number of people that need to be made redundant |
| | | RedundancyModel.workers_to_remove | Chooses which workers are made redundant |
| | | RedundancyModel.remove_workers_from_workforce | Removes workers from the workforce |

# Section three: Model logic

## Overview

This section explores the core logic of the simulation model and covers:

- An overview of the micro-simulation model
- An overview of the simulation sequence
- A detailed exploration of each step in the simulation sequence.

## Overview of micro-simulation model

The skills forecasting model is a micro-simulation model. It models a workforce by simulating the pathway of individual virtual workers through the food and fibre sector. The simulation progresses in monthly time steps, updating the workforce to reflect the impact from a range of workforce processes (e.g., movement into/out of the workforce, changes in skills, progression to higher role levels). These simulations add layers of insight to basic workforce forecasts, like how many people work in a certain industry with a specific combination of employment (e.g., tenure, job level) and individual (e.g., age, ethnicity) attributes. They enable 'what-if' scenarios to be explored, which simulate how different factors might change under various future conditions.

Every virtual worker in the simulation is defined by various employment and personal attributes. Central to these attributes is the worker's ID—a unique integer assigned to each virtual worker within a simulation. This ID is crucial for tracking the progress and changes of each worker over time and across different simulation outcomes. It is important to note, however, that this ID is specific to each simulation run; an ID used in one simulation does not refer to the same virtual worker in another. This ensures each simulation is independent and its data distinct. Each virtual worker is only tracked while part of the simulation. That is, once they leave the simulation (e.g., retire, go overseas, transition out of the food and fibre sector), they are no longer tracked by the model.

Many processes within the simulation, including workforce entries and exits, are driven by stochastic models. This approach ensures that the outcomes of each simulation will vary, even when the initial parameters remain unchanged. Such variability reflects the inherent uncertainty in the scenarios generated by the model.

The simulation can operate on different scales. In a smaller-scale model, each virtual worker might represent several people, whereas in a full-scale model, each virtual worker corresponds to an individual person. The smaller the scale of the simulation, the greater the variation in outcomes across different simulation runs.

The simulation model generates outputs that detail the number of virtual workers, categorised by a variety of employment and individual attributes (such as industry group, job level, employer size, age, gender, and ethnicity)[1] at each time step. These outputs can be used to examine – and compare – different 'what-if' scenarios, simulating how different factors might evolve under various future conditions.

[1] See Appendix A: Data dictionary for a full breakdown of attributes and their definitions.

## Overview of the simulation sequence

The sequence of the simulation steps is presented in Figure 1. The steps are colour coded according to their function, which are:

Orange indicates the **start** and **end** of the simulation.

Grey indicates **setup** and **closing** procedures.

Red indicates the **main cycle** of the programme, which are where workforce processes take place. After setup procedures are completed, the simulation repeats the main cycle (coloured red) until the final time step is reached.

Figure 1: Process flow of the simulation sequence

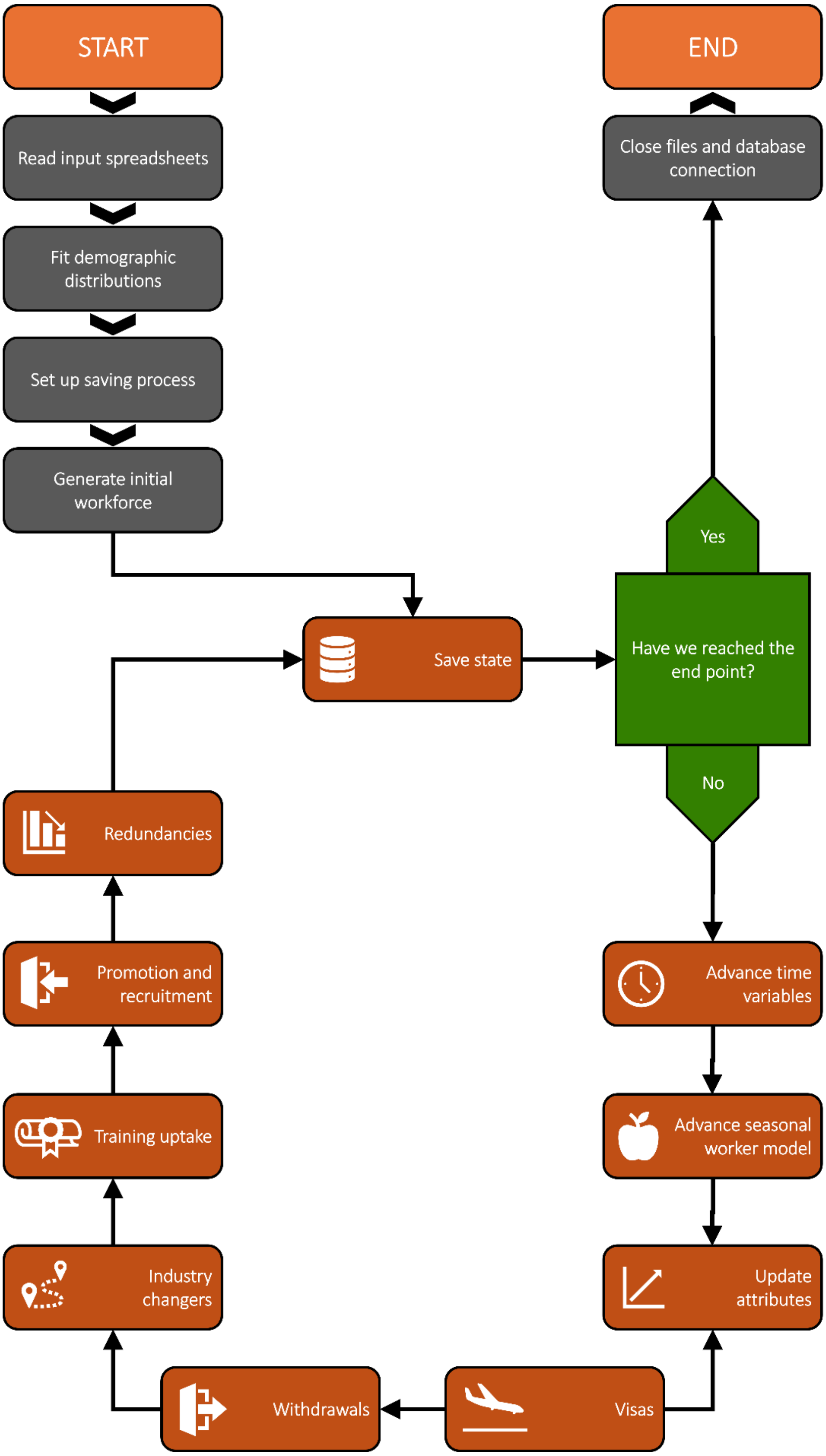


Table 8: Description of steps in the simulation sequence

| Step in simulation sequence | Python file | Description |
|---|---|---|
| Read input spreadsheets | data.py | • Read inputs from overall parameters (model_input_overall_parameters) and people needed (model_input_people_needed) input spreadsheets<br>• Perform data validation, checking that data is complete and within expected ranges<br>• Perform pre-processing, arranging the data into forms that are easy and quick to use in the main simulation loop<br>• Read inputs from new entrants (model_input_new_entrants) and initial population (model_input_population) input spreadsheets, to be used in the next step. |
| Fit demographic distributions | ipfp_fitting.py | • Initialise an 'initial guess' matrix for the IPFP algorithm to iterate on<br>• Apply IPFP to fit a many dimensional probability distribution matrix to account for complex demographic correlations<br>• Arrange the resulting matrix into a long two-dimensional table for easier use when generating workers. |
| Set up saving process | saver.py | • Connect to the database, and upload parameter values when in database output mode<br>• Create output files when in file output mode<br>• Set up annual summary variables when in annual summary mode. |
| Generate initial workforce | recruiter.py | • Generate enough workers, and assign roles, to satisfy the year 2000 demand, using the year 2000 IPFP results. See the section on **Fit demographics distribution and generate initial workforce** for more details. |
| Save state | saver.py | • Update annual summary variables if in annual summary mode<br>• Output the state of each worker to an output file or the database, depending on the output mode. |
| Advance time variables | time_class.py | • Increment the time, which is stored as a pair of integers representing the year and the month:<br>– The month is stored with a 'month index', 0 represents the start of January, and 11 represents the start of December |

| Step in simulation sequence | Python file | Description |
|---|---|---|
| | | – If the month index exceeds 11, reset it to 0 and increment the year. |
| Advance seasonal worker model | seasonal_worker_model.py | • Advance seasonal workers and manage their contracts, using parameters from the overall parameters (model_input_overall_parameters) input spreadsheet. See the section on **Advance seasonal worker model** for more details. |
| Update attributes | workers.py | • Increment age and tenure by the time step for each virtual worker<br>• Reset the worker's availability for promotion. |
| Visas | visas.py | • Assign visas to new entrants, particularly from the migrants talent pool<br>• Transition the worker to a new visa or remove them from the model when a visa expires. |
| Withdrawals | simulation_engine.py | • Remove workers from the food and fibre sector entirely, using parameters from the overall parameters (model_input_overall_parameters) input spreadsheet. See the section on **Withdrawals** for more details. |
| Industry changers | simulation_engine.py | • Move workers between the industry groups, using parameters from the overall parameters (model_input_overall_parameters) input spreadsheet. See the section on **Industry changers** for more details. |
| Training uptake | training_uptake_model.py | • Run on-the-job training processes. |
| Promotion and recruitment | recruiter.py<br>talent_pools.py | • Fill vacancies through internal promotion and external recruitment. See the section on **Training uptake** for more details.<br>• Generate new workers where external recruitment is required. See the section on **Fit demographics distribution and generate initial workforce** for more details. |
| Redundancies | redundancies.py | • Simulate redundancies and employers going out of businesses when demand suffers a shock. |
| Close files and database connection | saver.py | • Close all files and database connections after completing the final time step and saving the state. |

# Set up

## Set up logging file

The first action the simulation takes, irrespective of the contents of the input files, is to establish a log file. This immediate creation of the log file ensures that any initial errors are promptly documented. For more details on how to read these outputs, see the section on **Log files**.

## Read input spreadsheets

This step involves reading input spreadsheets to set up the simulation. Initially, the program reads the 'running arguments' tab from the overall parameters spreadsheet to determine whether to run the full simulation or to conduct a test on a single file.

Subsequently, it processes the spreadsheets for overall parameters, new entrants, and the initial population, storing this information in an *Inputs* object for easy access during the simulation. Lastly, the initial population spreadsheet is read, and its data is used in the subsequent step to generate the initial workforce.

## Fit demographics distribution and generate initial workforce

This step is handled in recruiter.py, with the initial generation of individual workers being controlled in talent_pools.py. The initial workforce is generated in the following steps:

1. **Fit demographics distribution**. Use IDI data to fit a complex demographics distribution to sample initial workers from
2. **Initial calculation.** Determine the number of non-seasonal workers required for each combination of industry group, employment status, and full/part time
3. **Population generation.** For each combination identified above, generate the necessary number of workers from the initial population distribution to fulfil the people needed
4. **Role assignment.** Use the capability model to hierarchically assign specific job levels, starting from strategic managers (filled by those with the most capability) down to entry-level positions, ensuring the most capable individuals are matched to appropriate roles.

### Iterative Proportional Fitting Procedure

IPFP is performed during the model setup and fits the high dimensional demographic distribution to the data provided. This method finds the distribution that matches the marginal distributions provided which is 'closest' to some initial guess. Note, due to a lack of data on role levels as defined within this model, role levels are excluded from IPFP and managed using the capability model.[2]

As was done for other similar projects conducted by Scarlatti, the initial matrix used to seed IPFP is a random matrix. All elements are between 0.01 and 1.00, avoiding zeros which would compromise IPFP.

[2] IPFP was implemented using the NumPy methods provided by the Python package ipfn. See https://pypi.org/project/ipfn/

### Set up saving process

This step sets up the method for how the results of the simulation are to be saved. There are 3 methods of saving results that can be selected in the overall parameters (model_input_overall_parameters) input spreadsheet.

#### CSV output

The output files are created, and their header rows are written – see the section on **Simulation results** for their full specification. During this setup, the *output_config.csv* file is created in full, and is not modified for the rest of the simulation.

#### SQL output

This output mode is not currently supported. In previous versions, the model would connect to a database, and obtain all metadata required to upload the results.

#### No output

If the 'no output' mode is selected, the set up saving process skipped.

## Advance time variables

Time is managed in this simulation with a custom class written in the file time_class.py. It is stored as a pair of two integers specifying the year and the month index. With each time step, the month is incremented by the time step size as specified in the overall parameters (model_input_overall_parameters) input spreadsheet.

In this simulation, time is represented using a *month index* instead of calendar months. This index ranges from 0 to 11, where 0 corresponds to the beginning of January (or the end of December in the preceding year), and 11 represents the beginning of December (or the end of November). This method of storing time is employed because each time step denotes a specific point within the month, rather than spanning the entire month.

## Advance seasonal worker model

Seasonal workers are employed *on a temporary basis during specific times of the year* to support tasks that recur annually. In the fruit industry group, for example, seasonal workers are brought in during the summer to assist fruit picking and packing activities. Similarly, in dairy farming, seasonal workers are hired during spring months to help with calving.

### Modelling seasonal workers

Seasonal workers are modelled using *contracts*. When a seasonal worker is recruited, they are given a start date and duration. They officially become part of the workforce on their start date and exit once their contract expires. Seasonal roles are strictly temporary – they cannot transition into permanent positions nor are they eligible for rehire in future years.

### Contract start dates and durations

Contract start dates are modelled using a normal distribution., while the length of the contracts is modelled using a gamma distribution. The parameters of these distributions can vary by industry group and be adjusted in the overall parameters (model_input_overall_parameters) input spreadsheet.

The contract parameters are aligned with workforce size data that has undergone seasonal decomposition. Adjustments to the decomposition parameters in the people needed (model_input_people_needed) input spreadsheet would require a recalibration of the contract parameters using the Python notebook ‘fitting_seasonal_parameters.ipynb’.

### Size of the seasonal workforce

The recruitment of seasonal workers is determined as a fraction of the permanent workforce at the time of generation. This approach guarantees that the size of the seasonal workforce is both proportional to and reflective of the current staffing needs at the time, and is consistent with the method of seasonal decomposition used in the people needed (model_input_people_needed) input spreadsheet.

## Update attributes

This step updates a worker’s time-dependent attributes, which include:

- Age
- Food and fibre tenure
- Remaining visa duration
- Availability for promotion
- Availability for regional migration
- Availability for industry group movement
- Availability for training.

Age, tenure, and remaining visa duration are all increased by a single time step. The remaining time-dependent attributes reset to make them available for these processes during the subsequent time step.

## Visas

Visas are modelled by tracking the visa category duration of stay. When a worker’s visa expires, they have a chance to transition between visas or leave New Zealand.

### Visa categories

Table 9 summarises the range of visa categories included in the simulation model, along with their conditions in relation to employment and study restrictions and durations.

**Table 9: Visa categories included in simulation model**

| Category | Type | Work limits | Study limits | Role limits |
|---|---|---|---|---|
| Student visa | Temporary | Part-time | None | None |
| NZ citizen or resident | Permanent | None | None | None |

| | | | | |
|---|---|---|---|---|
| Accredited employer work visa | Temporary | Full-time only | Any study for employment, otherwise 3 months per year | None |
| Working holiday visa | Temporary | None | 6 months | None |
| Recognised seasonal employer limited visa | Temporary | Seasonal only | No study | Managed only, certain industries only |
| Skilled migrant category resident visa | Permanent | None | None | None |
| Australian | Permanent | None | None | None |
| Other working visa | Temporary | None | 3 months per year | None |

### Stay and visa durations

Visa durations are drawn from a distribution specified in the overall parameters (model_input_overall_parameters) input spreadsheet. These distributions only depend on the visa category, can be finite or indefinite, and are based on the stay duration limitations of visas in the categories.

### Visa transitions

When a visa expires, a transition matrix is used to determine whether the worker leaves New Zealand (and exits the model), or transitions to a different type of visa. Table 10 shows an example transition matrix where each row represents the expired visa category, and each column indicates the subsequent visa category / state. The student visa row shows that after a student visa expires, the worker has a 20% chance to transition to an 'other working visa' (e.g., a post study work visa) and an 80% chance to leave New Zealand.

Table 10: Visa transition matrix used to determine a worker's subsequent state

| Category | Student visa | Working holiday visa | Other working visa | ... | Leaves New Zealand |
|---|---|---|---|---|---|
| Student visa | 0% | 0% | 20% | ... | 80% |
| Working holiday visa | 0% | 0% | 0% | ... | 100% |
| ... | ... | ... | ... | ... | ... |

## Withdrawals

At each time step, each permanent worker has a probability of leaving the workforce. This probability of withdrawal is estimated using a logistic function and informed by the following predictors:

- Age

- Employment status
- Industry group
- Tenure.

The probability of leaving is expressed explicitly as:

$$P(\text{worker leaves over 1 year}) = \frac{1}{1 + e^{-(\beta_0 + \beta_{\text{age}} + \beta_{\text{tenure}} + \beta_{\text{employment status}} + \beta_{\text{industry group}})}}.$$

The parameters $\beta_i$ were fitted using data from Stats NZ IDI. The fitted values are shown in Table 11.

**Table 11: Withdrawal sub-model parameter values**

| Family | Grouping | Value $\beta_i$ |
|---|---|---:|
| Constant | - | -3.21 |
| Age | 15 to 24 | -0.50 |
| | 25 to 34 | -0.40 |
| | 35 to 44 | -0.30 |
| | 45 to 54 | 0.00 |
| | 55 to 64 | 0.30 |
| | 65 and over | 0.20 |
| Tenure (months) | 0 to 11 | 2.65 |
| | 12 to 23 | 2.25 |
| | 24 to 35 | 1.70 |
| | 36 to 47 | -18.09 |
| | 48 to 59 | 3.81 |
| | 60 and over | 0.00 |
| Employment status | Employee | 0.00 |
| | Employer | 0.00 |
| | Self-employed | 0.00 |
| Industry group | Apiculture | -0.63 |
| | Arable | -0.52 |
| | Dairy farming | -0.26 |
| | Equine, dogs and racing | 0.22 |
| | Forestry | 0.01 |
| | Fruit | 0.13 |
| | Grapes and wine | 0.05 |
| | Nursery, turf and gardening | 0.30 |

| Family | Grouping | Value $\beta_i$ |
| --- | --- | --- |
| | Poultry, pigs and other livestock farming | 0.12 |
| | Seafood - production | 0.24 |
| | Seafood - processing | 0.05 |
| | Sheep, beef and deer farming | -0.05 |
| | Support services | 0.00 |
| | Vegetables | 0.28 |
| | Veterinary | -0.55 |
| Visa category | Student visa | 0.00 |
| | NZ citizen / resident | 0.00 |
| | Accredited employer work visa | 0.00 |
| | Working holiday visa | 0.00 |
| | Recognised seasonal employer limited visa | 0.00 |
| | Skilled migrant category resident visa | 0.00 |
| | Australian | 0.00 |
| | Other working visa | 0.00 |

As values were fitted to withdrawals in a one-year period, the probability for a single time step is given by:

$$P(\text{worker leaves over one time step}) = 1 - \big(1 - P(\text{worker leaves over 1 year})\big)^{\Delta t},$$

where $\Delta t$ is the length of the time step expressed as a fraction of a year.

## Industry changers

Industry group changers are implemented using a similar model to workforce withdrawals. Each time step, a worker has a certain probability of moving to another industry group based on the following attributes:

- Age
- Employment status
- Industry group
- Tenure.

The fitted parameter values are shown in Table 10. If the random process results in the worker changing industry groups, the destination industry group is chosen from a distribution contained in the overall parameters (model_input_overall_parameters) input spreadsheet. This distribution only depends on the worker's origin industry group.

Table 12: Industry group changer sub-model parameter values

| Family | Grouping | Value $\beta_i$ | p-value |
|---|---|---|---|
| Constant | - | -4.43 | 4.65E-15 |
| Age | 15 to 24 | 1.3050 | 0.0102 |
| | 25 to 34 | 1.2336 | 0.0155 |
| | 35 to 44 | 0.8297 | 0.1041 |
| | 45 to 54 | 0.7794 | 0.1271 |
| | 55 to 64 | 0.6823 | 0.1829 |
| | 65 and over | 0.5561 | 0.2850 |
| Tenure (months) | 0 to 11 | 0.0 | - |
| | 12 to 23 | -0.7706 | < 2e-16 |
| | 24 to 35 | -1.3174 | < 2e-16 |
| | 36 to 47 | -1.8027 | < 2e-16 |
| | 48 to 59 | -2.1318 | < 2e-16 |
| | 60 and over | -2.6480 | < 2e-16 |
| Employment status | Employee | 0 | - |
| | Employer | 0 | - |
| | Self-employed | 0 | - |
| Industry group | Apiculture | 0.0 | - |
| | Arable | 0.2610 | 3.21E-11 |
| | Dairy farming | 0.2499 | 2.24E-05 |
| | Equine, dogs and racing | 0.2751 | 3.66E-05 |
| | Forestry | 0.2603 | 2.29E-05 |
| | Fruit | 0.2518 | 8.13E-08 |
| | Grapes and wine | 0.2608 | 0.0016 |
| | Nursery, turf and gardening | 0.2619 | 0.0005 |
| | Poultry, pigs and other livestock farming | 0.2673 | 1.41E-07 |
| | Seafood - production | 0.2606 | 0.0005 |
| | Seafood - processing | 0.2914 | 0.6807 |
| | Sheep, beef and deer farming | 0.2499 | 8.47E-09 |
| | Support services | 0.2495 | 1.13E-05 |

| Family | Grouping | Value $\beta_i$ | p-value |
|---|---|---|---|
| | Vegetables | 0.2568 | 0.0002 |
| | Veterinary | 0.3071 | 0.1078 |

## Training uptake

On-the-job training of employees in the model is handled through the Training Uptake Model. The function of this model consists of three main parts:

1. **Determining the base probabilities** that each qualification is studied, with modifiers to this probability based on certain attributes. This information is contained in the overall parameters (model_input_overall_parameters) input spreadsheet and was extracted from the IDI.
2. **Setting scenario-specific modifiers** to the base probabilities. These are manual inputs in the overall parameters input spreadsheet.
3. **The Python training uptake model** which, for each person, determines the resulting final probability of studying each qualification and then chooses one qualification from this list to add to that person.

### Determining the base probabilities

This is the core data that the training uptake model is based on; it can be found in the "quals" tab in the input spreadsheet and is based on an IDI query. The key steps in creating this file were to:

1. For the past nine years identify all qualification studied by Muka Tangata staff while being employed.
2. Group these into logical groups based on their NZSCED code, NZQF level, and duration/type.
3. Calculate the *base probability* for each qualification group as proportion of the people completing each qualification in a year relative to the total number of people employed in MT industries in that year.
4. Calculate the conditional modifiers to these base probabilities based on multiple factors under the assumption that the correlations between the factors are negligible. The factors considered are age_groups, roles (which were approximated through the income percentile of the people in their industry), tenure_groups, industry_groups, and visa_groups. The groupings used for the training uptake model are displayed in Table 13. The values for these modifiers are based on the proportion of people that completed each qualification among the number of people in each group and then expressed as factor for the base probabilities. Be aware that these factors had to be suppressed if the contributing count of people, number of education providers offering the qualifications, or number of entities these people were associated with were lower than the respective values for suppression.

Table 13: Groupings for the training uptake model

| Age groups | Roles | Tenue groups | Industry groups | Visa groups | |
|---|---|---|---|---|---|
| | | | | Model | Original |
| 15-24 | Managed | 0-11 | Same as above | Resident | Citizen/resident |
| 25-34 | Semi-autonomous | 12-35 | | | Skilled migrant resident |
| 35-44 | Manager | 36+ | | | Australian |
| 45-54 | Strategic Manager | | | | Accredited employer work |
| 55-64 | | | | Short | Student |
| 65+ | | | | | Working holiday |
| | | | | | Seasonal employer |
| | | | | | Other working |

## Setting scenario-specific modifiers

The next part in the model, after the base-probabilities and modifiers for the groups have been determined, is to adjust the variables used to investigate the influence of what-if scenarios. The variables considered for this by the model are the cost of the qualification, its duration, and whether the qualification is otherwise promoted. The variables and the associated levels of these are displayed in Table 14. The resulting modifiers to the base probabilities are calculated automatically in the spreadsheet when a modifier is manually set for a qualification. The modifiers were determined based on comparisons of average enrolment numbers of qualifications across specific years – to estimate the effect of cost- or policy-changes – and between similar qualifications with different durations.

Table 14: Variables that adjust the probability of training uptake in the model and their levels

| Cost | Duration | Promotion/attractiveness |
|---|---|---|
| Free | Days | Normal |
| 50% off | 1 month | Encouraged |
| Normal | 3 months | Required/regulated by policies |
| 50% more | 6 months | |
| 100% more | 1 year | |
| | 2 years + | |

## The Python training uptake model

The Python model works by loading the data from the previous two steps and then using these, together with the attributes for the simulated workers in the model, to determine the final probability for a worker to enrol in each qualification as well as to not study and then randomly choose one of these qualifications – including not studying – based on the respective probability of each. Whether a qualification is completed or failed is determined after the number of months associated with each qualification. The chance of the completion can be modified in the model, but is set to 70% by default, which has been the average completion rate over the past several years as reported by multiple sources. On a more technical level, the model has the following steps in order:

1. At each timestep of the model, reduce the remaining months of study by one for each person currently studying. If the remaining duration goes down to 0, it is checked whether the person is awarded the qualification or if they fail. In both cases they are removed from the pool of currently studying people and therefore are available to study again, but only in the former case is the qualification added to their list of obtained qualifications (using the saver function).
2. Determine which groups each person belongs to among the groups shown in Table 13. Their probabilities to study each qualification are then calculated by multiplying all of the person's group modifiers with the base probability of each qualification, and their chance to *not* study is determined by subtracting the sum of all probabilities from 100%.
3. Choose a qualification – including not studying – from the list created in the previous step randomly using the calculated probabilities of each.
4. Add the people for whom a qualification other than "not studying" has been selected to the list of currently studying workers, with the qualification that was chosen and its associated duration.
5. For the workers where the remaining duration in months is 0 after the previous step, which is only the case when the duration of the qualifications is "days", the check of whether the qualifications is completed or failed happens right after the qualifications are chosen for all workers.

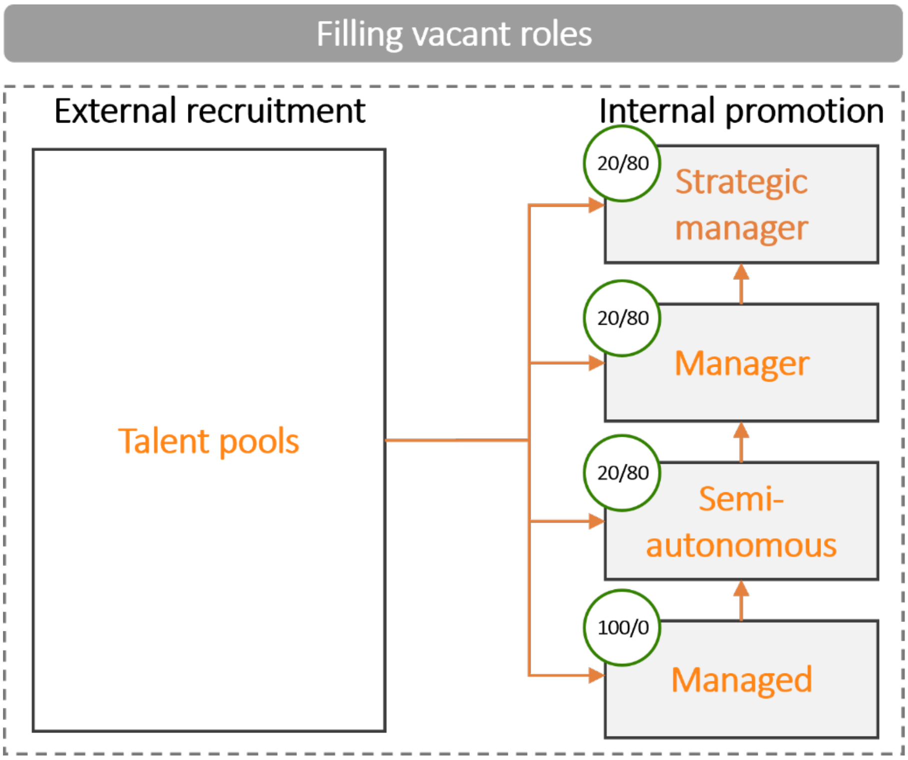
## Promotion and recruitment

The simulation model fills vacancies in the following sequence:

1. **Internal promotion** which advances workers already employed to the next job level
2. **External recruitment** which brings in new entrants into the workforce.

The proportion of job level vacancies filled by internal promotion and external recruitment varies by job level but is consistent across industry groups – see Figure 2 below. The simulation model attempts to fill a maximum of 80 percent of all job level vacancies through internal promotion, then recruits the remaining 20 percent from external talent pools. These percentages are adjustable within the overall parameters (model_input_overall_parameters) input spreadsheet.

Figure 2: Split between internal promotions and external recruitment to fill in vacant roles




## Capability model

When promoting a worker into a vacancy or hiring external recruits, the *worker's capability* in relation to the role must be evaluated. To facilitate this, a capability model has been developed to quantify a worker's capability through their *charge out rate,* denoted in dollars per hour. This capability is a function of:

- Years of industry relevant experience
- Years of experience in any food and fibre industry group
- Years of external work experience
- Qualifications
- Intrinsic ability.

The capability is expressed as the sum of an experience term, an intrinsic ability term, and a constant term.

$$C = E + \beta_I I + C_0$$

- **Experience contribution $(\boldsymbol{E})$.** This part considers the worker's qualifications and their experience in both the food and fibre sector and other industries. It acknowledges that while experience is valuable, its impact doesn't increase indefinitely. Beyond a certain point, additional experience contributes less to capability. The experience contribution is specified as:

$$E(q, e_{ig}, e_{FF}, e_{ext}) = f_{T_{ig}}\left(Q(q) + \beta_{ig} e_{ig} + f_{T_{FF}}\left(\beta_{FF} e_{FF} + f_{T_{ext}}(\beta_{ext} e_{ext})\right)\right),$$

where,

$$f_T(x) = \begin{cases} \beta x, & \beta x < T \\ \beta T, & \beta x \geq T \end{cases}.$$

Terms with $e$ are the experience (measured in months), terms with $\beta$ are the value of a year of experience (measured in \$/hr per year), and terms with $T$ (measured in \$/hr) are the maximum value that each type of experience can bring. The subscripts $ig$, $FF$, and $ext$ represent the industry group doing the hiring, other food and fibre industry groups, and industries external to food and fibre respectively. For example, $e_{ig}$ is the workers experience in the industry group in question, and $T_{ext}$ is the maximum capability in \$/hr that can be gained by experience external to food and fibre (e.g., retail experience). The $Q$ term represents the value of a qualification.

- Intrinsic ability contribution ($\boldsymbol{\beta_I I}$). This reflects how much a worker's innate qualities add to their capability. It is a way to quantify the *extra* that someone brings to the table (e.g., motivation and work ethic), beyond what can be measured by experience and qualifications alone. Here $\beta_I$ is how much extra capability is added at one standard deviation of intrinsic ability, and $I$ represents the intrinsic ability of the worker.

  Each worker is assigned an intrinsic ability upon being generated. It is not changed at any other point. We assume that intrinsic ability is independent of all other attributes and is generated with a standard normal distribution:

  $$I \sim N(0,1)$$

  This distribution is truncated to between -3 and 3.

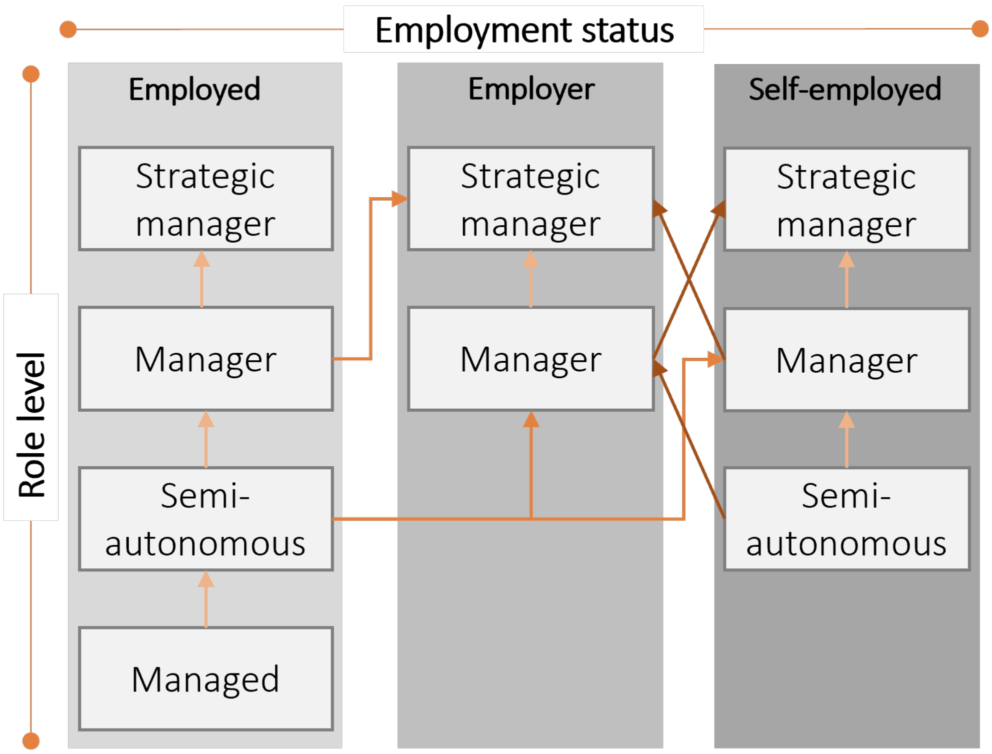
- Base capability ($\boldsymbol{C_0}$). This can be interpreted as the starting point for capability, representing what a person with average intrinsic ability but no experience would bring to a role.

## Internal promotion criteria and pathway

Promotions are modelled by first identifying vacancies to be filled and then promoting workers into the role based on the following criteria:

- They are currently working in the industry group where the vacancy exists
- If region is enabled, the worker is employed within the region of the vacancy
- Their current job level aligns with the vacancy's level according to the predefined promotion pathways (see Figure 3), which dictates that workers may move:
  - Vertically within the same employment status
  - Horizontally or vertically across different statuses, including transitions from:
    - Employed to employer
    - Employed to self-employed
    - Between employer and self-employed positions.
- The individual selected for promotion possesses the highest capability score relevant to the job requirements.

Figure 3: Promotion pathways by job level




## External recruitment

During each time step, there are new workers who enter the workforce. The number of new entrants for each time step is equal to the number needed to satisfy demand (as specified by the *'Export table'* tab in the people needed (model_input_people_needed) input spreadsheet, multiplied by an 'under-recruitment adjustment' that can be set in the overall parameters (model_input_overall_parameters) input spreadsheet. External recruitment is executed in the following steps:

### Step one – identifying talent pools

Workers entering the workforce are drawn from the following talent pools:

- Beneficiaries
- Industry changers
- Migrants
- Other
- Returning kiwis
- School leavers
- Tertiary graduates.

Each talent pool contributes a specific percentage of candidates to fill vacant roles. This percentage is allowed to vary by industry group and can be adjusted in the overall parameters (model_input_overall_parameters) input spreadsheet.

### Step two – allocating job levels across talent pool

An integer matrix was populated to allocate job levels to new entrants from each talent pool. In this matrix, each cell represents the count of individuals from a particular talent pool assigned to a certain job level. The general allocation process is below:

1. Job levels were first assigned to tertiary graduates with relevant qualifications. The allocation for each job level is determined through a random process, based on the number of job openings and specific parameters found in the overall parameters (model_input_overall_parameters) input spreadsheet
2. This method is then applied to the rest of the tertiary graduates and other talent pools.

The allocation of the talent pool does not take into account the job levels of the vacancies, occasionally leading to situations where some vacancies cannot be filled. In such cases, industry changers step in to fill the remaining vacancies. This happens rarely and thus does not substantially affect the initial talent pool distribution.

### Step three – generating new entrants to match vacancy requirements

The types of workers that are needed vary by industry group and role level. To ensure that a worker is a suitable candidate for the vacant role, we generate a worker whose *capability function* matches those who have been promoted into the role. This process is guided by the **Capability model**. For details on how these external workers are generated, see the section on **Fit demographics distribution and generate initial workforce**.

Occasionally, the simulation engine may not be able to generate an externally recruited worker with a satisfactory capability function. In such instances, the vacancy remains open until the next time step.

## Generating new workers

When performing recruitment, any new entrants that have not previously worked in the food and fibre sector need to be generated. These newly generated workers need to have attributes that match the job level they are being generated for. For example, a strategic manager will have more experience and higher-level qualifications than a managed employee. This is done by repeatedly generating workers until we get one whose capability function for that vacancy is close to that of recent promotions to this job level.

### Assigning categorical attributes

Each worker comes with a set of categorical attributes that needs to be determined, including:

- Employer size
- Ethnicity
- Gender
- Previous qualifications
- Visa categories.

These are sampled from distributions defined in the new entrants (model_input_new_entrants) input spreadsheet. These distributions can be correlated, and a select few may vary over time.

For attributes like employment status and industry group, these are predefined based on the role being filled. The distribution is then tailored to only produce candidates matching the specific employment status and industry required.

### Assigning continuous attributes

Each worker also has the following continuous attributes that need to be assigned:

- Age
- Tenure within food and fibre
- Tenure within their industry group
- Tenure within their current role
- Previous tenure other sectors.

These attributes are grouped into bands and treated as categorical variables for the assignment process. Once a worker has been assigned a band, the continuous attribute is chosen uniformly from within this band. For example, a worker is initially assigned an age band of '45 to 54', then their age is chosen uniformly from the range $45 \leq \text{age} < 55$. To avoid unrealistic ages and tenure mismatches, we apply the following caps:

- Age is capped at 75
- Tenure within food and fibre is capped at the worker's age minus 15 years
- Tenure within their industry group and role is capped by their food and fibre tenure.

### Tenures of new entrants

New entrants are considered to have no tenure in the food and fibre sector upon their generation. Instead of generating using the above process, their tenure is set between 0 and the length of one time-step across all relevant categories.

### Tenure within other sectors

New entrants are assigned a tenure within other sectors, also called 'external tenure'. This experience contributes to the capability model when deciding who is selected to fill a given vacancy. A new entrant's tenure within other sectors, $T$ (measured in months) is determined using the following stochastic formula:

$$T = 12(\text{age} - 20)(1 - \epsilon),$$

where $\epsilon$ is a sampled from a uniform distribution ranging between 0 and 0.01 (independently and identically distributed for every worker), and age is measured in years.

### Employer size

Employer size is not measured by directly tracking enterprises and businesses. Instead, it is estimated through statistical methods that consider the attributes of the workers. This estimation is currently based on two main factors: employment status (i.e., employed, employer, and self-employed) and the industry group in which they are employed. The method uses a distribution that varies based on these two factors to estimate employer size, and it can be adjusted in the overall parameters (model_input_overall_parameters) input spreadsheet.

### Visas

Each new entrant is assigned a visa upon their generation. Each visa type is chosen identically to other categorical attributes, however the visas held by each worker are unique in their other attributes (e.g., date assigned, time remaining), so a unique visa attribute is generated for each worker. For more on how the visa details are generated, see the section of **Visas**.

### Partial time step

As a worker may be generated in the middle of a time step, rather than neatly at the start or end of a month, we perform a partial time step on these workers. These partial times steps consist of just the **Update attributes** step, in which the attributes are updated by a random fraction of a time step rather than a whole time step.

## Redundancies

Sometimes, the model predicts that fewer workers are needed than are currently employed, leading to potential job cuts. The redundancy process runs during each time step for industry group and job level:

1. Compare the workforce size to the number of people needed. If the workforce size is larger than the people needed, exit the process. Otherwise, continue to step two.
2. Subtract the people needed from the workforce size to determine the number of redundancies.
3. Select workers to be made redundant. This is done randomly, with each worker having an equal chance of being chosen.
4. Remove the workers from the workforce who have been made redundant.
5. Record the redundancies in the output file output_redundancies.csv.

It should be noted that:

- This job-cutting process happens only *after* exploring other ways that might naturally reduce the number of workers, like people leaving their jobs for personal reasons or changing their visa status.
- This process doesn't apply to seasonal workers, who are expected to have temporary jobs.

## Save state

At the end of each time step, the current state of the simulation is saved. This step updates the output file *output.csv*, appending the state of each worker. For more information on the output files, see the section on **Simulation results**.

Note that only *output.csv* is updated in this step. The other output files are updated at other points of the main cycle.

## Close simulation

This is the final shut down step of the model. Any open files or SQL database connections are finalised and closed.

# Section four: Using the model

## Overview

This section details:

- **The steps for setting up and running the simulation model**, including the initial preparation, performing pre-calculation to optimise performance, and customisation through command line arguments
- **Outputs generated from the model**, including files on worker states, seasonal contracts, simulation options, qualifications, and redundancies, alongside a log file for debugging and event tracking
- **The process for developing custom scenarios**
- **How data and simulation outputs are managed.**

## How to set up and run the model

### Initial setup

To run the model, one first needs to set up a Python environment and install the necessary prerequisites. This setup only needs to be performed once.

1. **Create a Python environment.** To run the model, you will need to create a Python environment with the required packages. We recommend that you use Anaconda[3] or Microconda[4] to set up a dedicated Python 3.12 environment, as these were used during development, but other solutions are available.
2. **Install required packages.** Once you have a Python environment, install the required packages using *pip*, running the command *pip install -r requirements.txt*.
3. **Configure database connection (optional)**. Edit the *secret.py* file to specify the connection parameters if you wish to output to a SQL database. Note that this file will initially contain placeholder values only, and therefore outputting to a SQL database is non-functional at first.

### Running the model

1. **Perform (and update, when necessary) pre-calculations.** To improve performance of the model, the model performs several pre-calculations (described in Table 13) and saves them to local storage. Each pre-calculation has various parameter dependencies, and will need to be **recalculated** if these parameters are altered. Failing to update pre-calculations after changing parameters may result in slower performance and inaccurate modelling. Instructions for performing the recalculations can also be found in Table 13.

---

[3] https://www.anaconda.com/
[4] https://docs.anaconda.com/free/miniconda/index.html

Note, if dependencies are not updated, recalculations are not required. For example, if you do not modify the *worker capability model* tab of overall parameters (model_input_overall_parameters) input spreadsheet, you do not need to perform the recalculation for the capability sampling. The simulation can instead used the saved results.

Each WDC has its unique set of input spreadsheets and therefore its own pre-calculations folder. By default, Muka Tangata's pre-calculations are saved in data/Muka Tangata input spreadsheets/precalculations/. However, filepaths can be adjusted when running the model.

Table 15: Pre-calculations required, and their dependencies on various parameters

| Pre-calculation | Purpose | Parameter dependencies | Recalculation instructions |
|---|---|---|---|
| Capability sampling | Generates data used to help generate a worker whose capability function falls within a certain range. | Capability model parameters contained in the overall parameters spreadsheet. | Run the model with the -*capability_sampling* argument. For example: *python main.py -capability_sampling* |

2. **Launch the simulation.** Once the setup is complete, you can start the model by running the file ***main.py***. By default, this will read inputs from the input spreadsheets contained in the folder *data/Muka Tangata input spreadsheets*, and save the results in the *data* folder.
3. **Customise your model using command line arguments.** The model supports various command line arguments for customisation, detailed in Table 14. Many of these options are also available in the overall parameters (model_input_overall_parameters) input spreadsheet under the '*running options'* tab. An important exception is the location of the input spreadsheets, set with *-input*, *-mukatangata* and *-use_flat_directory*. This can only be set through a command line argument. Note that using a command line argument will override the spreadsheet settings.
    a. **Example of customisation – select a WDC**. To select a WDC, specify the corresponding input spreadsheet through using command line arguments -*input*, -*mukatangata*.

Table 16: Command line arguments

| Argument | Values | Result |
|---|---|---|
| --help, -h | N/A | Prints a summary of the program's purpose and lists all arguments |
| --version | N/A | Prints the version of the simulation |
| -o | 0, 1, 2, 3 | Set the output mode:<br>• 0: No output<br>• 1: Upload snapshot outputs to the SQL database<br>• 2: Save snapshot outputs to the SQL database<br>• 3: Save an annual aggregate to a file, designed to be comparable to the WDC dashboard. |

| | | |
|---|---|---|
| -f | File path ending in .csv | Sets the file path for outputs |
| -s | Integer | Sets the scenario id, using a unique scenario id will avoid overwriting previous simulations already in the database |
| -seed | integer | Sets the seed used to generate random numbers. This is useful when you need to repeat a run. The seed for any simulation can be found in the log file. |
| -input | Folder path | Set the folder containing the spreadsheet to use. For example, *python main.py -input "data/Muka Tangata input spreadsheets/"* |
| -mukatangata | N/A | Use the Muka Tangata input spreadsheets. This is equivalent to -*input "data/Muka Tangata input spreadsheets/"* |
| -precalculation_folder | Folder path | Set the folder containing the precalculations. This defaults to "data/Muka Tangata input spreadsheets/precalculations/" (when -*mukatangata* is used) |
| -logging_folder | Folder path | Set the folder to save the simulation log files to. Defaults to *"logs/"* |
| -use_flat_directory | N/A | Use a flat directory for the simulation. All filepaths for input spreadsheets, pre-calculations, log files, and output files are set to the same directory as the *main.py* file. This is useful when running in a VM which does not contain subdirectories. |
| -Wall | N/A | Show all warnings |
| -Wmodel | N/A | Show modelling warnings |
| -Wother | N/A | Show warnings raised by other packages |
| -bypass_max_agents | N/A | By default the simulation has a max agent cap of 1,000,000, and will not run if the number of agents required surpasses this limit. To bypass this limit, use this command line argument. Note that the simulation may be resource intensive, and take a long time to finish. |
| -profile | N/A | Run the model with a profiler. The results of this profiler can be used to determine how long different parts of the simulation take. Result files and readable text files are store in the *profiling* folder by default. |
| -po | File path | Filepath to save the readable profiling results to |
| -capability_sampling | N/A | Run the capability sampling pre-calculation |

# Output files

## Simulation results

As the simulation runs, it outputs the results to a set of csv files. Unless otherwise specified in the overall parameters (model_input_overall_parameters) input spreadsheet or by command line argument, these will be stored in the data folder with the following filenames:

- *output.csv.* This file contains the state of the worker at each point in time in a rectangular format. One row per worker per time step they were employed. See Table 15 for a description of the fields in this output file.
- *output_seasonal.csv.* This output file contains the contract details of seasonal workers. See Table 16 for a description of the fields in this file.
- *output_config.csv.* This file contains simulation options that are useful know when interpreting the other result files. See Table 17 for a description of the simulation options recorded in this file.
- *output_qualifications.csv.* This file contains the qualification history of an individual. Each qualification a worker has is recorded here by the date on which it was achieved. See Table 18 for a description of the fields in this file.
- *output_redundancies.csv.* This file contains a record of all redundancies made by the redundancy process. See Table 19 for a description of the fields in this file.

Table 17: Field summary of output.csv

| Field | Type | Description |
|---|---|---|
| year | Float | Time corresponding to this output - e.g., 2000.0 is the start of January 2000, and 2000.5 is the start of July 2000 |
| id | Integer | Worker ID, unique within a simulation to a worker |
| Age | Float | Age of the worker |
| ff_tenure | Float | Tenure of the worker within all of food and fibre |
| industry_group_tenure | Float | Tenure of the worker within their current industry group |
| role_tenure | Float | Tenure of the worker within their current role |
| external_tenure | Float | Tenure of the worker in industries out of scope |
| employer_size | String | Number of employees associated with the worker’s employer |
| intrinsic_ability | Float | Intrinsic ability |
| ethnicity | String | Worker’s ethnicity |
| gender | String | Worker’s gender |
| role | String | Worker’s role level |

| | | |
|---|---|---|
| employment | String | Worker's employment status |
| job_hours_category | String | Is the worker full-time or part-time |
| industry_group | String | Industry group of employment |
| highest_nzqf_level | String | NZQF level associated with the worker's most recent qualification |
| origin | Origin | Talent pool that the worker was originally sourced from. When the worker is part of the initial population, then the origin is 'initial'. |
| worker_type | String | Type of worker, 'nonseasonal' or 'seasonal' |
| visa category | String | Category of visa the worker holds at this time |
| visa duration | Integer | Total duration of the visa. "inf" for indefinite visas / citizenship |
| visa months remaining | Integer | Months remaining on the worker's visa. "inf" for indefinite visas / citizenship. |
| visa started time | String | Time when the visa was started. Format: "YYYY/month index". |

Table 18: Field summary of output_seasonal.csv

| Field | Type | Description |
|---|---|---|
| id | Integer | Worker ID, unique within a simulation to a worker. A seasonal worker will have the same ID in *output_seasonal.csv* as in *output.csv*. |
| industry_group | String | Industry group of employment |
| contract_start | Float | Start time of contract. Uses the same format as year in Table 15 |
| contract_end | Float | End time of contract. Uses the same format as year in Table 15 |

Table 19: Simulation options saved to output_config.csv

| Option | Description |
|---|---|
| scale | Scale of the simulation. For example, a scale of 100% is represented as 1.0, and a scale of 10% is represented as 0.1. When counting the number of workers in other outputs, divide the counts by this scale factor to transform back to full scale. |
| time step (months) | Time step in months |
| time step (years) | Time step in years, equal to the time step in months divided by 12 |
| start year | Start year of the simulation |

| | |
|---|---|
| start month | Start month of the simulation. For example, value of 0 represents the start of January, and a value of 11 represents the start of December. |
| end year | End year of the simulation |
| end month | End month of the simulation, represented the same way as the start month |

Table 20: Field summary of output_qualifications.csv

| Field | Type | Description |
|---|---|---|
| achieved_year | Integer | Year when the qualification was achieved. Set to 1900 for initial workers. Set to year of starting employment in the food and fibre sector for pre-employment qualifications. |
| achieved_month_index | Integer | Month index (0 for start of January, 11 for start of December) when the qualification for achieved. Set to 1 for initial workers. Set to month index of starting employment in the food and fibre sector for pre-employment qualifications. |
| worker_id | Integer | ID of the worker who achieved the qualification |
| kind | String | Kind of qualification. Either<br>• Qualification of initial worker<br>• Pre-employment<br>• On-the-job. |
| qualification_index | Integer | Index of the qualification, uniquely identifying the qualification category |
| nzsced | String | NZSCED categorisation of the qualification. If the qualification is relevant to food and fibre, then an NZSCED code is specified. If the qualification is not relevant to food and fibre, then ‘other’. |
| nzqf_level | String | NZQF level group of the qualification |

Table 21: Field summary of output_redundancies.csv

| Field | Type | Description |
|---|---|---|
| year | Integer | Year the worker was made redundant |
| month | Integer | Month index (0 for start of January, 11 for start of December) corresponding to the start of the step during which the worker was made redundant |
| worker_id | Integer | ID of the worker who was made redundant |

## Log files

In addition to creating the output files, the model also creates a log file. This is stored in the *logs/* folder and is titled with the date and time that the simulation was ran. This file is useful for debugging, as it includes a detailed record of events, alongside any warnings or errors encountered

# How to develop custom scenarios

The core function of the model is to enable Muka Tangata, and other industry stakeholders, to simulate 'what-if' scenarios that explore the impacts of various supply and/or demand-side factors on the food and fibre workforce. See Table 20 for a list of scenarios that have been simulated by the model. This section provides a step-by-step guide on how to set up and run these scenarios.

**Table 22: Examples of scenarios that have been simulated by the model**

| Scenario name | Scenario definition/research question |
|---|---|
| A set target of workforce skill level | How long would it take to reach a target where X% of the food and fibre workforce held a level X qualification (or higher)? |
| Ongoing decline in training completion | If completions in relevant VET training continues to decline, what percentage of the workforce will have a formal relevant qualification by 2035? |
| Regulatory skill requirement | What would happen if the food and fibre sector had a target of X% of the workforce having a X level qualification by 2035? What enrolment numbers would be needed to achieve this target? |
| Affordable training | What would happen to the training rates and the skill profile of the workforce over the short and long term if training became more affordable? Reasons may include government funding or subsidies increasing or being introduced. |
| Unaffordable training | What happens if fewer workers or business can afford to take up or fund training? Reasons may include ongoing and depending on recession, change in personal circumstances or availability of government funding or subsidies. |
| Entry level versus technical skills | What happens to the skill profile of the food and fibre workforce if you do a big push on entry level or technical skills? |
| Domestic level fees for migrant workers | What would happen to the skill profile of the food and fibre workforce if migrant workers had access to domestic level fees? |

## Define the scenario and research question

The first step is to clarify what you want to examine and why. This clarity guides the setup of the scenario and support identifying relevant scenario assumptions.

- **Identify the key focus of your scenario**. For example, when thinking about a training focussed scenario, does the scenario intend to introduce new training or make existing training more accessible or desirable?
- **Formulate specific research questions**. Formulating research questions also enable you to refine the focus of the scenario and provides a basis to engage stakeholders. That is, the

question we want to answer is *"if completions in relevant VET training continues to decline, what percentage of the workforce will have a formal relevant qualification by 2035?"* See Table 20 for examples of scenarios and research questions.

- **Determine the scope of your scenario**. For example, does the scenario impact the entire workforce or is it restricted to a particular industry group or population within the workforce?

## Scenario set up

To support this section, the following research question will be used as a reference point: *If completions in relevant VET training continues to decline, what percentage of the workforce will have a formal relevant qualification by 2035?*

- **Define key components** of the scenario must be clearly defined. For example, using the research question above, what constitutes "formal relevant qualifications"? In this example, 'formal' qualifications can be defined as those with an NZSCED code relevant to the food and fibre sector. 'Relevant' qualifications are determined by a relevancy score which available from the overall parameters (model_input_overall_parameters) input spreadsheet.
- **Define the underlying assumptions** of the scenario to be clearly defined. This supports the simulation and also offers transparency to the model outputs. Key consideration here may include temporal aspects (e.g., the scenario takes place over a five-year period starting from the current year or does it start in year X and continues throughout the simulation window?). In the example, assumptions may include:
  - Training only includes vocational education and training (VET)
  - From 2023, no new VET training are introduced, discontinued, or reclassified to a different NZSCED code
  - The declining trend in training volume between 2005 to 2022 continues to 2050.
- **Identify mode parameters.** This step requires the analyst to consider what parameters may need adjustment. In the example, parameters may include:
  - VET completion counts between 2023 to 2050
  - Simulation scale. In is recommended that scenarios are initially run at the 10% scale, and once reviewed and verified, can be run at full scale.
- **Determine the data needed to run the scenario.** Additional data may be required to support the scenario simulation, and could be sources from Stats NZ IDI, industry reports and stakeholders surveys. In the example, the following data may include:
  - Historical VET completion counts by NZSCED x NZQF level from 2000 to 2022 (Stats NZ IDI)
  - Qualification relevancy by industry group (high/low/none).

## Set up the scenario templates

This step involves preparing copies of the relevant input spreadsheets and updating model parameters and data as necessary. Key tasks include:

- **Make a copy of the required input spreadsheets** and save to a dedicated file location.

- **Update model parameters** as identified in the previous step, ensuring that they align with the scenario's requirements.
- **Add or update data** required for the scenario, which may involve importing new datasets or modifying existing data to reflect scenario assumptions.
- **Validate the scenario setup** by checking for consistency and completeness in the scenario parameters and data inputs.

## Run the simulation

With the scenario fully set up, the next step is to execute the simulation and generate output files. Key tasks include:

- **Performing any pre-calculations** required for the changed parameters (see page 50).
- **Select an appropriate scale and enter this into the overall parameters input spreadsheet.** We recommend simulating 1 out of 10 workers for a testing run, then only running at full scale when you are happy with your test results. Note that this testing scale may vary depending on the results you need. Any statistics involving a low count of workers (e.g., in the smaller industry groups or regions) will require a larger scale.
- **Upload the input spreadsheets**, then trigger a run.
- **Retrieve the output files** from the appropriate location and use these to construct your desired statistics and figures.

## Review simulation outputs of the scenario

After the simulation is complete, an immediate next step is to review the outputs. Some questions to consider may include:

- **Are the results expected?** If the results are unexpected, it would worthwhile to undertake a deep dive on the research question and simulation output to determine whether new insights are being provided, or whether an error may have been made in relation to definitions, assumptions or parameter changes.
- **How sensitive are the results to variations in the model parameters?** This question prompts an examination of the robustness of the results and can help identify which parameters are most influential in determining the outcomes.

# Section five: User interfaces

## Overview

Microsoft Power BI dashboards were used as both general user interface and expert user interface for the model. Both interfaces are designed to cater to different user needs:

- **The general user interface** offers an accessible view of model forecasts and scenario comparisons for the public,
- **The expert interface** provides a more detailed analysis tool with calibration data for comparison and debugging of custom scenarios

## General user interface

Designed to be public facing, the general user interface visualises model outputs, allowing users to delve into the "business as usual" workforce forecast and compare different scenarios.

### Scenarios

Currently, there are a range of supply-side and demand-side (i.e., people needed) scenarios available to explore.

#### Supply-side scenarios

Supply-side scenarios govern the rate at which new workers can be recruited into different roles from different talent pools (e.g., school leavers, immigrants, industry changers), the level of skills and/or qualifications they bring with them, the way in which the workforce responds to shocks in demand and the availability of training for new and existing workers. The supply-side scenarios included in the dashboard are presented in Table 20.

#### Demand-side (i.e., people needed) scenarios

Demand-side scenarios forecast workforce (individuals) needed at a disaggregated level in terms of occupations, roles, and levels of experience. Existing and hypothetical forecasts can be mapped against this framework to provide demand scenarios for the model. The dashboard includes three demand-side scenarios taken from the NZIER and MPI workforce forecasts for 2032.[5] These are presented below.

Table 23: Demand-side scenarios included in the dashboard

| Scenario name | Scenario definition/research question |
| --- | --- |
| Business as usual (baseline) | Reflects the past performance of the food and fibre sector to project the workforce in 2032. It is based on trends in the recent past around investment, productivity and technology.<br><br>**Note, this is the baseline scenario against which the impact of scenarios are compared.** |

[5] See the report here.



SCARLATTI

| | |
|---|---|
| Increased use of technology | Assumes the food and fibre sector takes maximum advantage of existing and emerging technologies, involving mechanisation, automation or greater use of digital and information technologies. |
| Transformed sector | Builds on the increased use of technology with and increased focus on sustainability and high-value products and markets. |

## Pages and filters

See the table below for an overview of the pages and filters incorporated into the dashboard to facilitate general users to navigate and interact with the model outputs.

**Table 24: Pages and filters in the general user interface**

| Category | Page | Filters on page |
|---|---|---|
| Overall workforce view | Workforce size | • Industry group<br>• Age group<br>• Food and fibre tenure<br>• Employment status<br>• Entry point |
| | Entry points | • Industry group<br>• Age group<br>• Employment status<br>• Visa status |
| | Industry group tenure | • Industry group<br>• Age group<br>• Employment status<br>• Entry point |
| | Age group | • Industry group<br>• Food and fibre tenure<br>• Employment status<br>• Entry point |
| Scenarios | Return on investment | • Industry group<br>• Age group<br>• Employment status<br>• Timeframe |
| | Completions | • Industry group<br>• Age group<br>• Employment status<br>• NZQF level |
| | Enrolments | • Industry group<br>• Age group |

| | | |
|---|---|---|
| | | • Employment status<br>• NZQF level |
| | Skill change | • Industry group<br>• Age group<br>• Employment status<br>• Food and fibre tenure |

## KPIs

Some pages in the general user interface include KPIs to provide a simple indication of how the observed attribute has changed over time. The KPIs are as follows:

- **Change in workforce size:** Calculates the percentage difference between the workforce sizes at the start and end of the selected timeframe.
- **Change in median tenure:** Calculates the difference (in months) between the raw (ungrouped) median tenures at the start and end of the selected timeframe.
- **Change in median age:** Calculates the difference (in years) between the raw (ungrouped) median ages at the start and end of the selected timeframe.

## Other

The dashboard also allows the user to adjust the time frame by year between 2000-2050. All visualisations and KPIs adjust dynamically.

## Adding new scenarios

A user can add new scenarios to the general user interface by following these general steps:

1. Run the model to generate the output files for your scenario.
2. In Power BI, select “Get data” from the Home ribbon, then select the “Text/CSV” option.
3. Select the output_config file, and choose “Transform Data”.
4. In the Power Query Editor window, go to any of the existing “_config” files (e.g., BAU_config), and select “Advanced Editor” in the Query section of the Home ribbon.
5. Copy all except the first two lines of the query (so excluding “let” and “Source = …”).
6. Close that pop-up window, return to the output_config for your scenario, and open the Advanced Editor again.
7. Paste the query from the other “_config” file here, then select “Done”.
8. Select “Close & Apply” at the start of the Home ribbon, then repeat steps 2 to 7 to also import the output file (copying the query from one of the “_output” files this time). Select “Close & Apply” again when done.
9. Update the Scenario definitions table with the name and definition of your scenario.
10. There are some measures and calculated columns which you may need to add. At the very least, you will need to copy the Yearly avg, scaled_simulation_measure and

scaled_simulation_output measures / calculated columns and paste them into new measures / calculated columns under the new output file, making sure to change the names and table references.

11. In Scenario 1 table and Scenario 2 table add in the columns needed to create the charts from the new scenario output table (eg scaled_simulation_measure, XXX). These tables use field parameters so use the DAX function to add them and add the scenario name as the group (what will be shown in the drop-down selections). It is helpful to label with the name of the scenario. Group the measure by the scenario (the name that will appear in the drop down of for scenario selections).
12. Select the scenario in both cases and add the new variable to each chart in the 'chart 1' and 'chart 2' filters. You will need to do this for chart 1 in the single select pages too.
13. In the Modelling ribbon, select "Manage relationships". Check that your output file has the column industry_group in a many-to-many relationship with Muka Tangata mapping (Muka Tangata Industry), with the latter filtering the former in a single direction. Add a relationship between the new scenario's year and the sort year table.
14. Update title measures and cards to include the new scenario to ensure the title will change dynamically (It's helpful to have the new scenario selected in scenario one and two for this).

## Expert user interface

The expert user interface is intended to enable technically advanced users to directly access the model via the series of input spreadsheets and Python code to adjust model inputs and run their own simulations.

As noted in the section on **Input spreadsheets**, there are several interfaces available to the technical user can, including:

- **Initial population** (model_input_input_population)
- **New entrants** (model_input_new_entrants)
- **Overall parameters** (model_input_overall_parameters)
- **People needed** (model_input_people_needed)
- **Calibration** (model_input_input_calibration_data).

# Appendices

## Appendix A: Data dictionary

| Term | Description |
|---|---|
| **Age** | Age group of the individual worker: 15 to 24, 25 to 34, 35 to 44, 45 to 54, 55 to 65, 65 plus. |
| **Attributes** | **Employment attributes:** Employer size., external tenure, food and fibre tenure, industry group, job level<br><br>**Individual attributes:** Age, ethnicity, gender, intrinsic ability, region, entry point, visa. |
| **Employee** | An individual earning salary/wages from an enterprise. |
| **Employer size** | The number of employees in a company: 1, 2-4, 5-9, 10-49, 50-99, 100+ |
| **Employer** | An individual who earns shareholder income, company director income, sole trader income, or partner income in their tax return, AND employs staff. |
| **Entry point (talent pools)** | The origin of the individual as they enter the food and fibre sector: Beneficiaries, industry changers, migrants, other, returning kiwis, school leavers, tertiary graduates |
| **Ethnicity** | Mutually exclusive ethnic groups: Asian, European, Māori, Pacific, MELAA, Other. |
| **External tenure** | A nominal value representing the numbers of months of tenure in employment external to the food and fibre sector. |
| **Food and fibre tenure** | The number of months since an individual entered employment. Within the simulation this can be any non-negative number. Within the dashboards it is grouped into the following groups: 0-11, 12-23, 24-35, 36-47, 48-59, 60 months plus. |
| **Gender** | Gender of the individual: Male, female. |
| **Industries** | ANZSIC06 names of industries relevant to Muka Tangata. |
| **Industry group tenure** | The number of months since an individual entered employment within a specific industry group. Within the simulation this can be any non-negative number. Within the dashboards it is grouped into the following groups: 0-11, 12-23, 24-35, 36-47, 48-59, 60 months plus. |
| **Industry group** | Muka Tangata industry groups: Apiculture, Arable, Beef and sheep farming, Dairy farming, Dogs and racing, Equine, Forestry, Fruit, Grapes and wine, Nursery, turf and gardening, Poultry, pigs and other livestock farming, Seafood – processing, Seafood – production, Sheep, Support services, Veterinary, Vegetables. |
| **Intrinsic ability** | Part of the capability model, this is a random normally distributed number that is assigned to each worker acting as a proxy of motivation and ability. |
| **Job level** | Eighteen combinations of full/part-time, employment status and role level: |

| | |
|---|---|
| | • Full time employee: Managed, semi-autonomous, manager, strategic manager<br>• Full time employer: Manager, strategic manager<br>• Full time self-employed: Semi-autonomous, manager, strategic manager<br>• Part time employee: Managed, semi-autonomous, manager, strategic manager<br>• Part time employer: Manager, strategic manager<br>• Part time self-employed: Semi-autonomous, manager, strategic manager. |
| **MPI sectors** | Food and fibre industry divisions used by MPI: Arable, Bees and Other, Cross Industry group, Dairy, Forestry and Wood Processing, Horticulture, Pork, Poultry, Red Meat and Wool, Seafood. |
| **New entrant** | An individual with less than one month of tenure in the food and fibre workforce. |
| **NZQF level** | Levels of the New Zealand Qualifications Framework, where levels 1-3 = NCEA (secondary school), levels 4-6 = certificates/diplomas, and level 7+ = further tertiary education. |
| **Region** | Auckland, Bay of Plenty, Canterbury, Gisborne-Hawke's Bay, Manawatu-Wanganui, Marlborough, Nelson-Tasman-West Coast, Northland, Otago, Southland, Taranaki, Waikato, Wellington. |
| **Role level** | Workforce classifications representing role hierarchy based on individual capability and experience: Managed, semi-autonomous, manager, strategic manager. |
| **Self-employed** | An individual who earns shareholder income, company director income, sole trader income, or partner income in their tax return, but does not employ staff. |
| **Time step** | A discrete interval of time within the simulation model, specifically one month in duration, starting January 2000 to December 2050. |
| **Visa** | A category of visa assigned to each worker: Accredited employer work visa, Australian, NZ citizen / resident, other working visa, recognised seasonal employer limited visa, skilled migrant category resident visa, student visa, working holiday visa. |
| **Work status** | A classification of an individual's participation in the workforce: Full time, part time. |

# Appendix B:

## talent_pools.py code structure

The module *talent_pools.py* contains methods and classes describing talent pools. The primary purpose of this file is to generate workers, both new entrants and initial workers. This section is primarily concerned with the structure of this file. For a description of the underlying logic, see Generating new workers.

This module contains five main classes:

- *NonSkillsBasedTalentPool*: Base class, represents the industry changers, migrants, and beneficiaries talent pools.
- *OtherTertiaryGraduatesPool*: Representing the tertiary graduates talent pool for graduates in untracked NZSCED codes.
- *RelevantTertiaryGraduatesPool*: Representing the tertiary graduates talent pool for graduates who have a relevant qualification (i.e., in a tracked NZSCED code).
- *SecondaryGraduatesPool*: Representing the secondary graduates talent pool.
- *InitialWorkerTalentPool*: Used to generate workers for the initial workforce.

The *NonSkillsBasedTalentPool* class defines the default behaviour for a talent pool. Mainly, methods around sampling from demographic distributions in order to create a new entrant. The other classes in this file are *children* of this class. The inherit this default behaviour, and modify it to make it suitable for the talent pool.

The *RelevantTertiaryGraduatesPool* tracks the number of graduates from each tracked qualification category, which informs the qualification distributions. Additionally, some data management methods are modified so that this points to the same data as *OtherTertiaryGraduatesPool* when required.

The only change made by the *OtherTertiaryGraduatesPool* class are those same data management methods described above for the *RelevantTertiaryGraduatesPool* class.

The *SecondaryGraduatesPool* class modifies the age distribution to choose from a narrower range of ages than is allowed by the usual age data ranges.

The *InitialWorkerTalentPool* overrides most behaviours used in the sampling of demographics. Instead of sampling from a simpler distribution that may change over time, the model uses the IPFP results, providing a more complex but static distribution.

When generating a nonseasonal worker, the *random_worker* method is called, and when generating a seasonal worker, the *n_random_seasonal_workers* method is called. These methods are only defined in the *NonSkillsBasedTalentPool* class and are not modified by any of the children classes. Instead, the methods it calls which determine the attributes of each worker are modified by the children classes (polymorphism).

## Appendix C: Developing scenarios

| Simulation step | Adjustments available for scenario development |
|---|---|
| Advance time variables | **Adjust start and end time –** change the years in the 'timeframe' tab of the overall parameters spreadsheet.<br>**Adjust the timestep size** – change the time step size in the 'timeframe' tab of the overall parameters spreadsheet. Options are: annual time step, quarter time step, monthly time step. |
| Advance seasonal worker model | **Adjust the number of seasonal workers required** – Change the 'size relative to permanent workforce' parameters in the 'seasonal workforce' tab of the overall parameters spreadsheet.<br>**Adjust when seasonal workers are required** – Change the contract parameters in the 'seasonal workforce' tab of the overall parameters spreadsheet. Use the plot to the right of those parameters for visual feedback. |
| Update attributes | N/A |
| Visas | **Adjust the stay length of international workers** – Adjust the visa duration distribution in the 'visas' tab of the overall parameters spreadsheet.<br>**Adjust what visa transitions are possible, or alter the likelihood of certain visa pathways** – Adjust the visa transitions table in the 'visas' tab of the overall parameters spreadsheet. |
| Withdrawals | • |
| Industry changers | • |
| Training uptake | • |
| Promotion and recruitment | • |
| Redundancies | • |