**Paid Leave Microsimulation Model:**

**Python Version User Manual**

Draft

October 31, 2019



**Paid Leave Microsimulation Model:**

**Python Version User Manual**

Draft

December 10, 2019

**PAID Leave microsimulation model**

**Python version USER MANUAL**

This version of the model builds upon the previous version IMPAQ and IWPR used for model testing, and model demonstration at Department of Labor (DOL) in June 2019. Since then, we have received useful feedback from DOL and state stakeholders, which has greatly facilitated our model development. The current model has a fully functional graphic user interface (GUI) that is ported to the Python simulation engine at the backend, and all model parameters made available to users in the GUI have been implemented (i.e. there is no placeholder parameters). This version should be considered as one that has the entire core set of functionalities. Any future updates and additions of model features should be based upon this version.

## 1. Summary of Model Updates

We have made the following key updates to the Python model:

Runtime improvement – Runtime has been greatly reduced with the power of vectorization in Python. For example, during the simulation of counterfactual leave lengths under a proposed state leave program for tens of thousands of rows in ACS dataset, the computation is implemented via Python’s numpy package that has C-runtime, which is by magnitudes faster than implementation via looping through all rows.

State data based on place of work – Based on feedback collected from interviewing states, state leave programs commonly determine eligibility of workers based on state of workplace, but not state of residence of workers. We therefore constructed state ACS datasets for all 50 states and DC based on state of work (variable POW in ACS), using the national ACS microdata. In the GUI, when the option State of Work is checked, a ‘POW’ version of state ACS data files will be used as the underlying population for simulation. Alternatively, if the option is unchecked, original state ACS data files based on state of residence will be used.

Dual benefit receivers – Based on feedback collected from interviewing states, it is common that workers can choose to receive paid leave benefits from employer and the state simultaneously. To reflect this feature in the model, we implemented a parameter that represents the share of dual receivers in eligible worker population, which user can specify under the Population tab in the GUI. Please note that, in current model implementation, this parameter will be capped by the share of workers who receive positive leave pay from employer, which is simulated based on the FMLA data.

Take up rates – We have fixed the issue in applying program take up rates to the population under simulation. In implementation, the application of take up rates occurs during the computation of program cost, where the population weight represented by each ACS person is multiplied by a fraction factor to reflect partial take up. This fraction factor would be strictly larger than take up rates which is defined as total number of workers with approved cases divided by total eligible worker population, as long as there are eligible ACS persons who do not take any leave. This is illustrated in **Exhibit 1** below. Following this logic, we updated the code to restore the fraction factor to be applied to leave taking ACS persons based upon user-supplied take up rates for each leave type, which would lead to proper estimates of program outlays, and administrative budget financing (ABF) estimates.

**Exhibit 1: Example of Applying Take Up Rates**

*If take up rate = 0.5, then there should be (100+100+100)\*0.5 = 150 workers who participate the program. This means each of ACS person 2 and person 3 should represent 75 program participants, thus we need to apply a fraction factor 0.75 to persons 2 and 3. Namely, we multiply each row of population weight for leave-taking persons by 0.75.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Eligibility | Population Weight | Leave Taking | Population Weight –  Program Participants |
| 1 | 1 | 100 | 0 | 0 |
| 2 | 1 | 100 | 1 | 75 |
| 3 | 1 | 100 | 1 | 75 |

Full participation of leave needers – Under *Population* tab in the GUI, if Needers Fully Participate is checked, then all leave needers (ACS persons who are simulated to have unmet leave needs under status-quo) are assumed to take up the state benefit under consideration. This option will override the take up rates for all leave types for leave needers, but will not override the take up rates for non-needers.

Weight factor – Under *Simulation* tab in the GUI, user can specify a positive value to scale up or down the population represented by ACS persons. This feature can be useful if user would like to simulate scenarios where there is overall population growth.

Random seed – Under *Simulation* tab in the GUI, if Random Seed is checked, then each run of the model will corresponds to a machine-generated pseudo-random state, such that difference between runs should be attributed to randomness if all parameters and input data are held constant.

Advanced parameter button – Above the Run button in GUI, we added an Advanced button to show/hide advanced model parameters, which are listed in **Exhibit 2**:

**Exhibit 2: List of Advanced Parameters and Location in GUI**

|  |  |
| --- | --- |
| Parameter Location in GUI | Parameter |
| Main panel | Simulation Method |
| Population tab | Needers Fully Participate |
| Simulation tab | Weight Factor |
| Simulation tab | Random Seed |

Existing state program parameters – In the main panel, we added an Existing State Program dropdown menu so that users can quickly specify parameters for programs that have been implemented in different states.

Population analysis histogram selection – In the Population Analysis tab in GUI output, we added a button for user to specify which histogram to show based on choice of population or subpopulation.

## 2. Model Use Guide

This section provides a step-by-step guide for users. With this guide, users should be able to set up the needed computing environment, properly specify model parameters, launch the microsimulation program, interpret the model output, and understand the potential of model extension.

Setting up the computing environment

Hardware requirements – Current version of the model has been tested on mainstream workplace and home computers with Intel i5 and i7 multicore processors, resulting in manageable runtime (within an hour) even for large ACS states such as California. Runtime is less than 5 minutes for small states such as Rhode Island. Minimum RAM tested is 8GB which is sufficient to handle large ACS states (California data is less than 2GB), although we recommend 16GM RAM or higher for better runtime performance. To store ACS data for all states, a disk space of 25GB is required, as the original ACS state household files and state person files (based on state of residence) have a total file size of about 12GB, while user would also need another 12GB to store the place-of-work based ACS state files, which are essentially an alternative partition of all ACS persons in the country. The file sizes of FMLA and CPS data sets are minimal compared to ACS thus have limited impact on the hardware requirements.

Software requirements – Current Python model is coded in Python 3, and the GUI can be launched from terminal by running Microsimulator.py. Windows users would need to have Python 3 as well as the necessary packages installed before running the model. Users can quickly install the needed packages by following these steps:

1. Open the Windows Command Prompt or PowerShell.
2. Navigate to the *Microsimulator/* directory.
3. Type the command *pip install –r requirements.txt* and then hit enter.

Users who already have any of the required packages installed but are unable to update them to the latest versions can use the command *pip install <package>*. Replace *<package>* with one of the packages below.

|  |  |
| --- | --- |
| * + cycler==0.10.0   + kiwisolver==1.1.0   + matplotlib==2.2.3   + mord==0.5   + numpy==1.17.4   + pandas==0.23.0 | * + pyparsing==2.4.5   + python-dateutil==2.8.1   + pytz==2019.3   + scikit-learn==0.20.1   + scipy==1.3.3   + six==1.13.0 |

Advanced Python users who might not be able to make changes to their environment can install the packages to a virtual environment. Follow the steps below if you have these restrictions:

1. Open the Windows Command Prompt or PowerShell.
2. Navigate to the *Microsimulator/* directory.
3. Type the following commands, hitting enter after each command.
   * *python -m venv microsim-env*
   * *microsim-env\Scripts\activate.bat* (from Command Prompt) or *microsim-env\Scripts\activate.ps1* (from PowerShell)
   * *pip install –r requirements.txt*

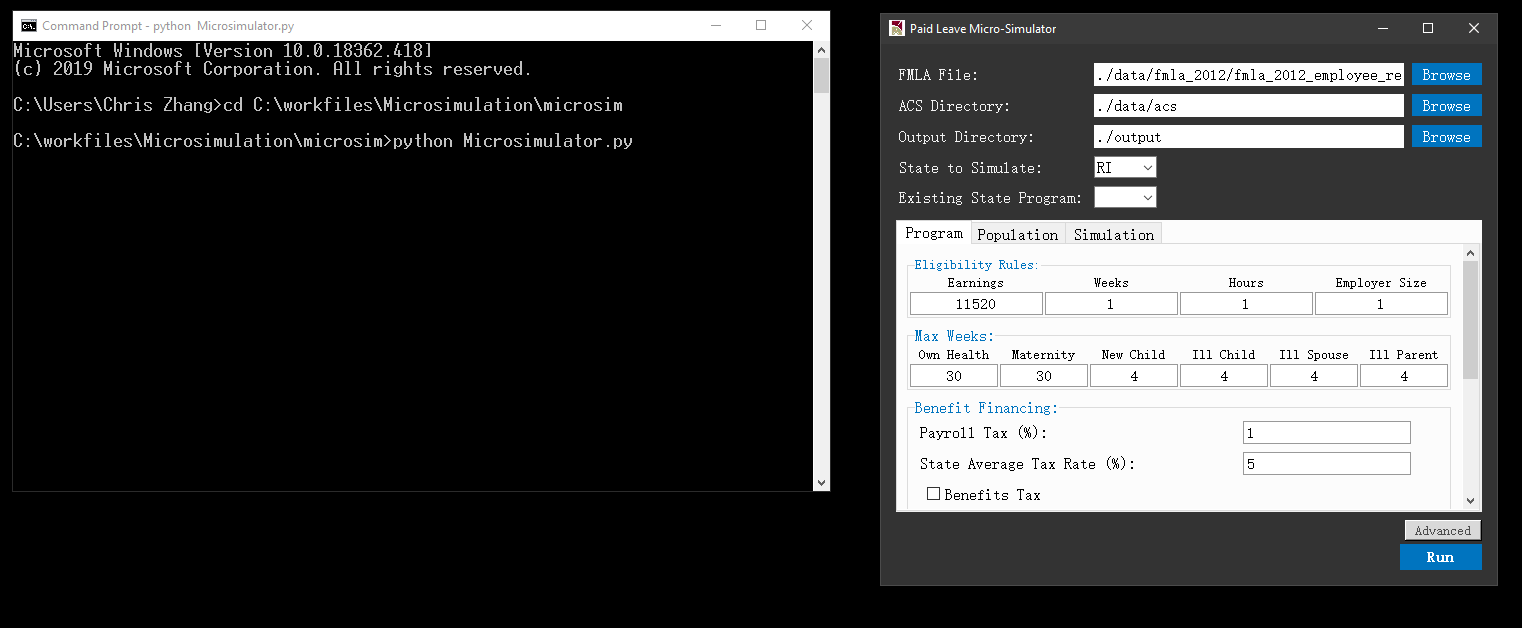
For the greatest simplicity, we recommend installing [Anaconda](https://www.anaconda.com/), which can greatly facilitate Python package installation for Windows users. The tool will work with the Anaconda 5.3.0 distribution. Users with older releases can update their packages by running the command *conda update --all*. The only package not included in Anaconda distributions is *mord*, which can be installed using the command *conda install –c mord*.

* Dataset requirements
  + ACS – Current model has been tested on ACS 2012-2016 5-year PUMS. For original ACS data files, user should download the desired state files from [Census](https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_pums_csv_2012_2016&prodType=document). Person files should be placed in *./data/acs/person\_files*, while household files should be placed in *.data/acs/household\_files*. User should keep the original file name of these Census data files for proper data reading by the program. For place-of-work based ACS state data, we have generated the state person and household files which are placed in *./data/acs/pow\_person\_files*, and *./data/acs/household\_files* respectively.
  + FMLA – Current model uses FMLA 2012 data, and should be placed in *./data/fmla\_2012*
  + CPS – the CPS microdata are used for auxiliary simulation of a few program eligibility variables for the FMLA population. Current model uses CPS March 2014 Annual Social and Economic (ASEC) Supplement. A data file containing the needed CPS data columns should be placed in *./data/cps*.

Running the model

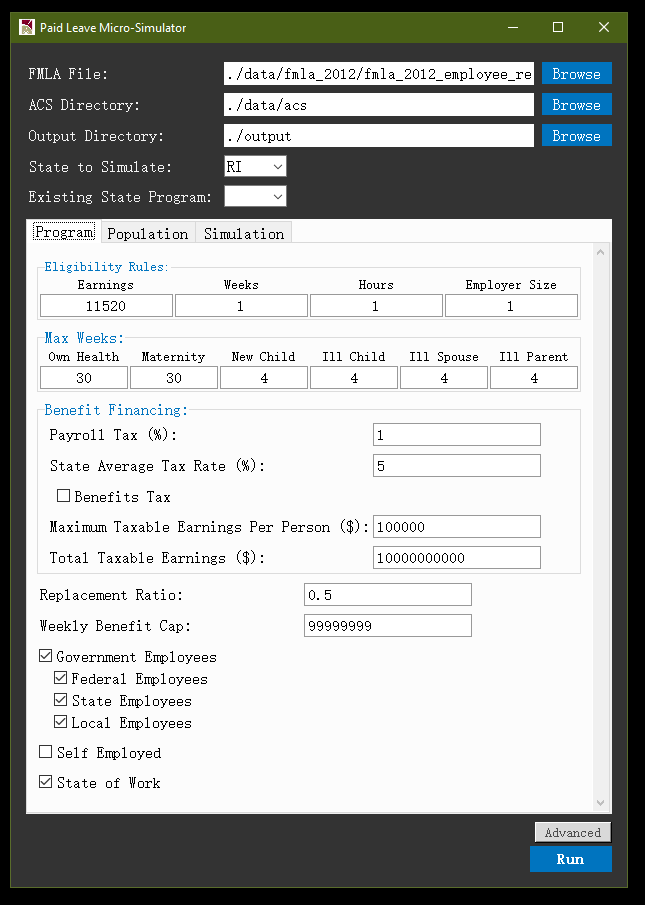
* Launching the model – After Python 3 and above-listed Python packages are installed, the model can be launched in a terminal (e.g. a Windows command line window). User should first change directory to the *microsim* folder, and run the command python *Microsimulator*.*py*. The GUI will then be launched, as shown in **Exhibit 3** below.

**Exhibit 3: Launching the Model**



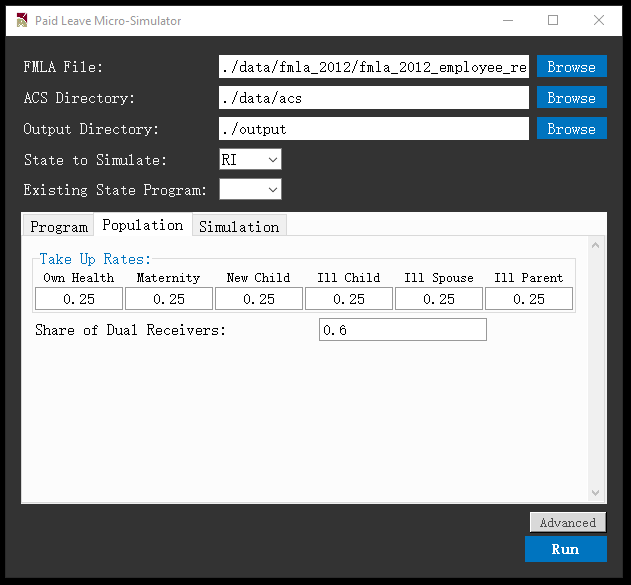
* Specifying input data and parameters – Once the above GUI is displayed. User can specify data sources and model parameters.
  + FMLA File – File path to the FMLA dataset. Current default is to use the FMLA 2012 data located at *./data/fmla\_2012/*
  + ACS Directory – Directory of ACS datasets. Current default is *./data/acs* which contains 4 subdirectories: *household\_files, person\_files, pow\_household\_files,* and *pow\_person\_files*. The former 2 subdirectories contain ACS state PUMS datasets that will be used if user choose to uncheck *State of Work* under the *Program* tab. The latter 2 subdirectories contain ACS state PUMS datasets that will be used if *State of Work* is checked.
  + Output Directory – Directory where output files will be stored upon completion of simulation.
  + State to Simulate – ACS state PUMS dataset to use as underlying worker population. The dropdown menu contains 50 states plus DC.
  + Existing State Program – Existing state leave program parameters to use. If a state is selected, parameters under *Program* tab will be overridden by a set of pre-determined parameters that best represent the leave program of the selected state.
  + *Program* tab – This tab contains the following parameters (a full list of *Program* parameters is shown in **Exhibit 4**):
    - Eligibility Rules – These are minimum requirements on annual earnings (in dollars), number of weeks worked over a year, number of hours worked over a year, and number of employees at workplace for a worker to be eligible to receive leave benefits from the program.
    - Max Weeks – These are maximum number of weeks for which an eligible worker can receive leave benefits from the program. User can set different value of maximum number of weeks for each of the 6 leave types.
    - Benefit Financing – These are parameters of the payroll tax on eligible workers. *Payroll Tax* is an integer value representing percentage points of tax rate. *State Average Tax Rate* is the estimated average state income tax, which would be applicable if the *Benefit Tax* checkbox below is checked, in which case the state leave program benefits would also be subject to payroll taxation, allowing state to recoup tax revenue from paid program benefits. *Maximum Taxable Earnings Per Person* is an integer value that places a cap on annual taxable earnings that are subject to this payroll tax.
    - Replacement Ratio – Share of wage that would be replaced by program benefits during leave. The ratio should be a positive value between 0 and 1.
    - Weekly Benefit Cap – Maximum weekly benefits in dollars for each leave type. Current model assumes a uniform cap for all 6 leave types.
    - Government Employees – Type of government employees to be included as eligible workers. If Government Employee is checked, all 3 child checkboxes will be automatically checked. Otherwise, user can selectively check 0, 1, 2, or all of the child checkboxes, including Federal Employees, State Employees, and Local Employees.
    - Self Employed – If checked, workers under self-employment will be included as eligible workers.
    - State of Work – If checked, state of workers will be determined by state of workplace. If unchecked, state of workers will be determined by state of residence.

**Exhibit 4: Model Parameters under Program Tab**



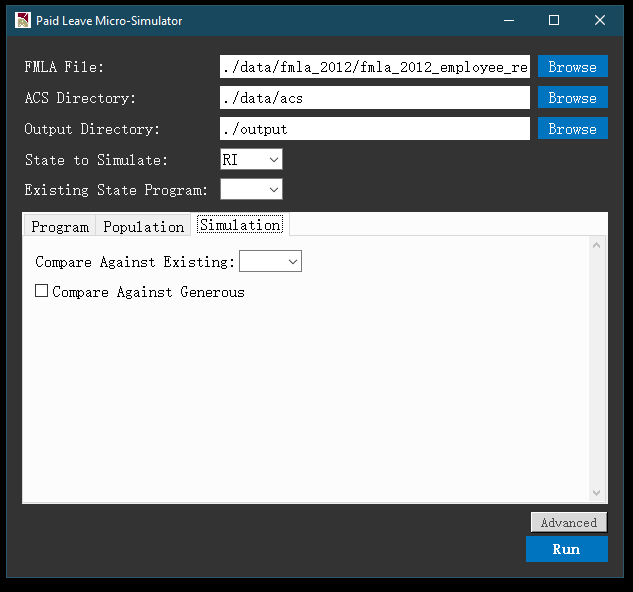
* + *Population* tab – This tab contains the following parameters (a full list of *Population* parameters is shown in **Exhibit 5**):
    - Take Up Rates – These are take up rates of the benefit for each leave type among all eligible workers in the state. Namely, take up rate is total number of leave-taking (thus equivalent to all cases approved in model) workers divided by total number of eligible workers in the state.
    - Share of Dual Receivers – Share of eligible workers who can receive leave benefits simultaneously from both employer and state program. In current model, this parameter will be capped in computation by the share of workers who receive positive leave pay from employer, which is simulated based on the FMLA data.

**Exhibit 5: Model Parameters under Population Tab**



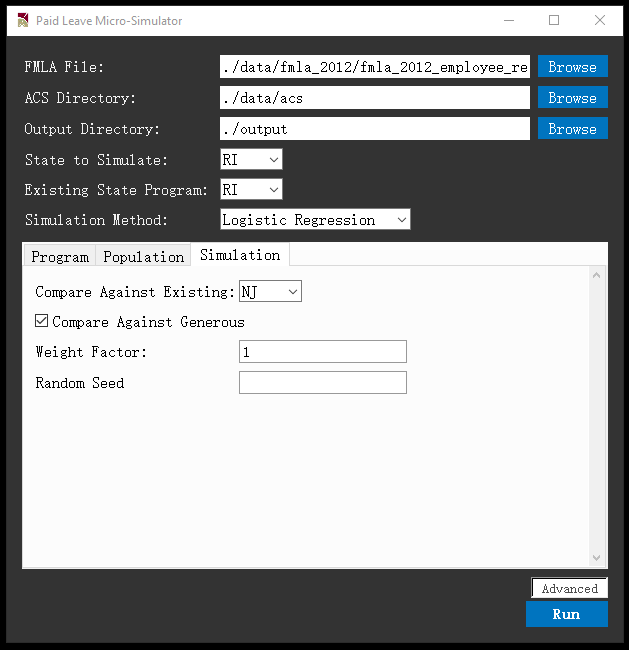
* + *Simulation* tab – This tab contains the following parameters (a full list of *Simulation* parameters is shown in **Exhibit 6**):
    - Compare Against Existing – This dropdown menu contains states for which user can select to re-run the simulation with the state’s program parameters. If a state is selected, upon completion of simulation, the result window will display results for a counterfactual program based on the worker population in the state chosen under *State to Simulate*, while using program parameters of the program for the state chosen under *Compare Against Existing.*
    - Compare Against Generous – If checked, upon completion of simulation, the result window will display results for a counterfactual program based on the worker population in the state chosen under *State to Simulate*, while using program parameters of a most generous program, in which under the *Program* tab all values are set to 0 in *Eligibility Rules*, *Replacement Ratio* is set to 1, and all type of government employees and self-employed workers are checked as eligible.

**Exhibit 6: Model Parameters under Simulation Tab**



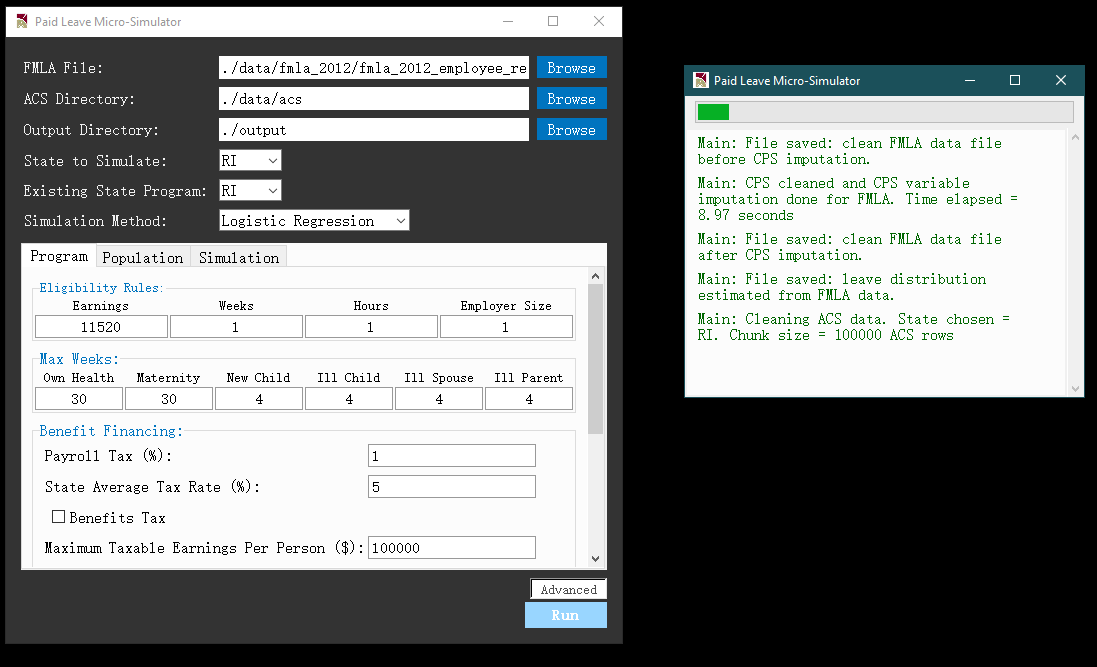
* + *Advanced* button – When clicked, this button will display advanced parameters across tabs as shown in **Exhibit 7**. Current model has following three advanced parameters:
    - Simulation Method – This dropdown menu in the GUI main panel allows user to specify the classifier to be used for simulation. Current model has implemented Logistic Regression, Ridge Classifier, K Nearest Neighbor, Naïve Bayes, Support Vector Machine, and Random Forest.
    - Weight factor – Under the Simulation tab, user can specify a positive value to scale up or down the population represented by ACS persons. This feature can be useful if user would like to simulate scenarios where there is overall population growth.
    - Random seed – Under the Simulation tab, if Random Seed is checked, then each run of the model will correspond to a machine-generated pseudo-random state, such that difference between runs should be attributed to randomness if all parameters and input data are held constant.
  + *Run* button – After configuring all parameters above, user may click the *Run* button to execute the simulation program.

**Exhibit 7: Advanced Model Parameters**



* Executing the model
  + Runtime display – After *Run* button being clicked, a runtime window will be displayed as in **Exhibit 8**. The runtime display shows a progress bar that represents estimated progress of current execution, and a series of runtime messages that indicates completion of key intermediate steps. The header of each message can be (i) *Main*, which represents the main simulation using state worker population based on *State to Simulate* and program parameters supplied under the *Program* tab, (ii) *Counterfactual*, which represents the simulation using the same state worker population but the alternative parameters based on program of the state that was chosen from the *Compare Against Existing* dropdown menu under the *Simulation* tab, if this option is turned on, or (iii) *Policy Simulation*, which represent the simulation using the same state worker population but the alternative parameters based on the most generous program, if this option is turned on.

**Exhibit 8: Runtime Display during Model Execution**



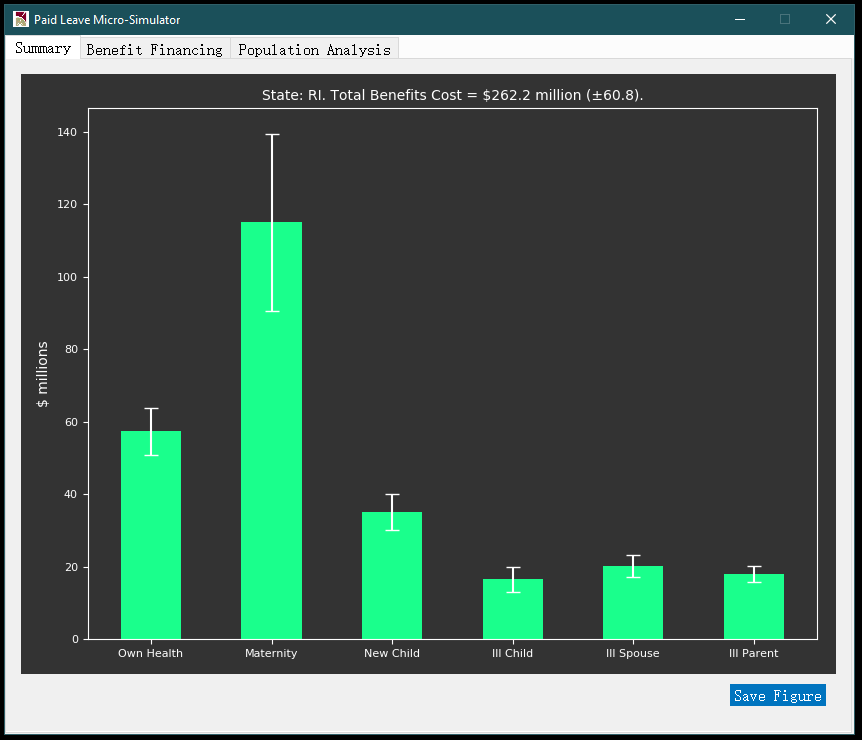
* + Runtime estimates – Time needed for completing the simulation is mainly affected by the size of the ACS state PUMS data. For smaller states with fewer rows, such as Rhode Island, executing the main simulation would take less than a minute on a Windows 10 machine with an Intel i7 processor and 16GB RAM. For the largest state California, executing the main simulation on the same machine would take about 14 minutes. These runtime estimates are all based on setting *Simulation Method* to *Logistic Regression*, which should be considered as the baseline classifier. For other classifiers, runtime may slightly differ. The exception is *Support Vector Machine*, a significantly more time-consuming method that relies on iteratively searching for a hyperplane in data features space that separates two classes (e.g. leave takers and non-takers). For example, when implemented with the Rhode Island population, runtime is 5 times slower under SVM compared to under logistic regression.

Please note that if user chooses to perform comparison simulations (i.e. comparing against existing program, or comparing against the most generous program), runtime would linearly increase in number of comparison simulations.

Post-simulation

* Simulation results in GUI – Upon completion of simulation, a result window will be displayed, with numerical and graphical results grouped in following tabs
  + *Summary* tab – As shown in **Exhibit 9**, this tab shows a graph that plots the estimates of program outlays for each of the six leave types. Total outlay estimate is displayed in the title of plot. There is a *Save Figure* button at bottom right corner for user to save the graph at a desired local directory.

**Exhibit 9: Summary Tab in Result Window**



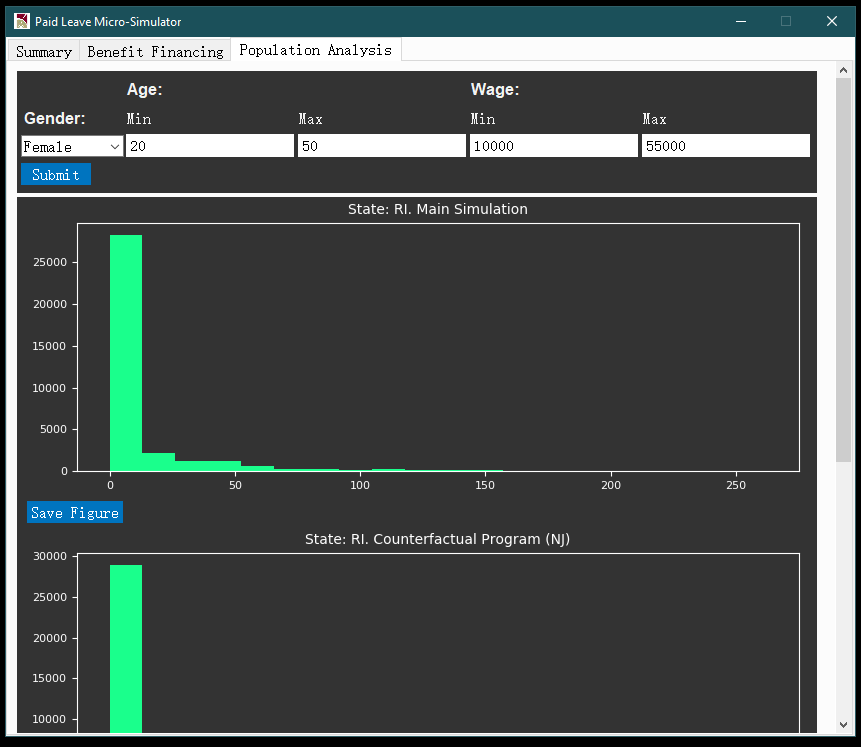
* + *Benefit Financing* tab – As shown in **Exhibit 10**, this tab displays a full set of results to illustrate how much program outlay can be financed given user-supplied Benefit Financing parameters, and how the tax revenues would be collected from different sub-populations, such as age groups, worker classes, and age groups. The top section of **Exhibit 10** shows that about $61 million would be generated by a 1% payroll tax, thus given the $262 million program outlay estimate in **Exhibit 9**, it can be estimated that the program would be funded at 61/262 = 23.3% by this payroll tax.

**Exhibit 10: Benefit Financing Tab in Result Window**

|  |  |
| --- | --- |
|  |  |

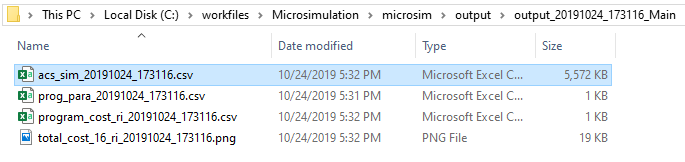
* + *Population Analysis* tab – As shown in **Exhibit 11**, this tab plots histogram of annual total leave length (in days) taken by eligible workers. Each histogram corresponds to a different simulation (Main/Counterfactual Program/Most Generous Program). On top of the panel, user can specify subpopulation of interest by selecting gender, age range, and range for annual wage income. Upon clicking the *Submit* button, the histograms below will be updated, showing the distribution of leave lengths for the specified subpopulation. Each histogram has its own *Save Figure* button should the user wish to export the graph to a local directory.

**Exhibit 11: Population Analysis Tab in Result Window**



* Simulation results in output folder – Besides results displayed in GUI, a set of analytical files will also be stored in the user-specified output folder (by default *./output*) should user have more customized analytical needs. **Exhibit 12** shows an example list of output files. Files generated from the same simulation run are stored in the same subdirectory */output\_[yyyymmdd]\_[hhmmss]\_[header]* where *yyyymmdd* and *hhmmss* respectively indicate the date and time stamp of model execution, and *header* indicate whether the simulation corresponds to *Main*, *Counterfactual*, or *Policy Simulation*.

**Exhibit 12: Files in Output Directory**



* + For each simulation header (which corresponds to a folder in the output directory), output files include the following
  + A master post-simulation ACS state PUMS data file – This is a dataset that contains all eligible ACS workers in the state chosen, with new columns generated from simulation attached.
  + A meta-data file that stores program parameters – This file allows user to keep track of the program parameters employed by the model for simulation.
  + A program cost summary file - This file summarizes program outlay across all six leave types and the total, as well as 95% confidence interval bounds for each outlay estimate.
  + A graph that plots program costs and confidence intervals – This graph plots data in the program cost summary, and is the same graph shown in the *Summary* tab in the result window.