**Paid Leave Microsimulation Model:**

**Python Version User Guide**

March 5, 2020



**Paid Leave Microsimulation Model:**

**Python Version Model User Guide**

March 5, 2020

**PAID Leave microsimulation model**

**Python version Model User Guide**

This document provides a step-by-step guide for users of the IMPAQ-IWPR paid leave microsimulation model. 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

1.1 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 16GB RAM or higher for better runtime performance. To store ACS data for all states, a disk space of 25GB is required.[[1]](#footnote-1) 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 (5.5MB and 6MB, respectively) compared to ACS thus have limited impact on the hardware requirements.

1.2 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 local directory of the model, where the file *Microsimulator.py* is located. For example, if model folder is *C:\Microsimulation*, then user should execute *cd C:\Microsimulation* and hit enter.
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 either in Anaconda using the command *conda install –c mord*, or in Windows Command Prompt or PowerShell using the command  *pip install mord*.

1.3 Dataset requirements

* ACS – Current model has been tested on 5-year ACS PUMS for periods 2012-2016, 2013-2017, and 2014-2018. 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 of ACS 2012-2016 should be placed in *./data/acs/2016/person\_files*, while household files of ACS 2012-2016 should be placed in *.data/acs/2016/household\_files*. ACS data of other years should be placed in the corresponding year subfolders. For place-of-work based ACS state data, we have generated the state person and household files which are placed in *./data/acs/[year]/pow\_person\_files*, and *./data/acs/[year]/household\_files* respectively. The model also mandates the following specific file name formats for ACS data files:

|  |  |
| --- | --- |
| Directory | Format of state ACS file name |
| ./…/household\_files | *ss[yy]h[st].csv* where yy=last 2 digits of ending year of 5-year ACS, and st=lower case 2-letter state postal code. For example, *ss16hri.csv* should be the file name for original 2012-2016 ACS PUMS household file for Rhode Island.[[2]](#footnote-2) |
| ./…/person\_files | *ss[yy]p[st].csv* where yy=last 2 digits of ending year of 5-year ACS, and st=lower case 2-letter state postal code. For example, *ss16pri.csv* should be the file name for original 2012-2016 ACS PUMS household file for Rhode Island. |
| ./…/pow\_household\_files | *h[sc]\_[st]\_pow.csv* where sc=2-digit FIPS state code, and st= lower case 2-letter state postal code. |
| ./…/pow\_person\_files | *p[sc]\_[st]\_pow.csv* where sc=2-digit FIPS state code, and st= lower case 2-letter state postal code. |

* FMLA – Current model uses public FMLA 2012 data downloaded from [DOL website](https://www.dol.gov/agencies/oasp/evaluation/fmla/fmla2012), and should be placed in *./data/fmla\_2012*.
* FMLA aggregate level leave length distribution data – These are population level data on empirical leave length distribution estimated from restricted FMLA data employee survey. The restricted FMLA dataset contains individual level leave lengths in exact number of days, which are restricted variables unavailable in public FMLA data. For the simulation model, only population level distribution of these leave lengths is needed. Therefore, an aggregate level distribution data file is generated as JSON format and placed in *./data/fmla\_2012*. File name is *length\_distributions\_exact\_days.json.* The restricted FMLA dataset(s) will not be included in any public dissemination of the model.
* CPS – the CPS microdata are used for auxiliary simulation of a few program eligibility variables for the FMLA population. Current model uses CPS March Annual Social and Economic (ASEC) Supplement in 2014, 2015, or 2016, the middle year corresponding to the three available 5-year ACS PUMS respectively. Data files containing the needed CPS data columns should be placed in *./data/cps*. In current model folder, the CPS data files have been reduced to needed columns, and are named as *CPS2014extract.csv*, *CPS2015extract.csv*, etc.

## Running the model

2.1 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*.[[3]](#footnote-3) The GUI will then be launched, as shown in **Exhibit 1** below.

**Exhibit 1: Launching the Model**

Exhibit 1

Exhibit 1 shows the how to launch the microsimulation model from Windows PowerShell, and the graphic user interface (GUI) after model is launched.

2.2 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 subdirectories representing years, e.g. *./data/acs/2016*, and within each year folder, 4 subdirectories: *household\_files, person\_files, pow\_household\_files,* and *pow\_person\_files*. Within the 4 subdirectories, the former 2 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. There is also an option “*All*” which represents all 50 states plus DC. If “*All*” is selected, the model will run the simulation for each state, and aggregate all results across states.
* *Program* tab – This tab contains the following parameters (a full list of *Program* parameters is shown in **Exhibit 2**):
  + 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.
  + Eligible Employee Types – These are checkboxes for types of employees that are eligible for the program, including private employees, self-employed, and government employees. Under the *Government Employees* checkbox, user can further check or uncheck subtypes including *Federal Employees*, *State Employees*, and *Local Employees*.
  + Benefit Financing – These are parameters of the payroll tax on eligible workers. *Payroll Tax* is an integer value representing percentage points of tax rate. *Maximum Taxable Earnings Per Person* is an integer value that places a cap on annual taxable earnings that are subject to this payroll tax. *State Average Tax Rate* is the estimated average state income tax, which would be applicable if the *Apply Benefit Tax* checkbox above 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.
  + 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.

**Exhibit 2: Model Parameters under Program Tab**

Exhibit 2

Exhibit 2 shows the graphic user interface (GUI) with model parameters displayed under the Program Tab, including the following parameters:
Eligibility Rules - Earnings,
Eligibility Rules - Weeks
Eligibility Rules - Hours
Eligibility Rules - Employer Size
Max Weeks - Own Health
Max Weeks - Maternity
Max Weeks - New Child
Max Weeks - Ill Child
Max Weeks - Ill Spouse
Max Weeks - Ill Parent
Eligible Employee Types - Private Employees
Eligible Employee Types - Self Employed
Eligible Employee Types - Government Employees (including separate checkboxes for subtypes: Federal Employees, State Employees, and Local Employees).
Benefit Financing - Payroll Tax Rate
Benefit Financing - Maximum Taxable Earnings
Benefit Financing - Apply Benefits Tax
Benefit Financing - State Income Tax Rate
Replacement Ratio
Weekly Benefit Cap

* *Population* tab – This tab contains the following parameters (a full list of *Population* parameters is shown in **Exhibit 3**):
  + 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 actual leave-taking workers under program (thus equivalent to all cases approved and paid in model) 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, out of all eligible workers who receive any leave pay benefit from employer. This share should be a value between 0 to 1.

**Exhibit 3: Model Parameters under Population Tab**

Exhibit 3

Exhibit 3 shows the graphic user interface (GUI) with model parameters displayed under the Population Tab, including the following parameters:
Take Up Rates - Own Health
Take Up Rates - Own Health
Take Up Rates - Maternity
Take Up Rates - New Child
Take Up Rates - Ill Child
Take Up Rates - Ill Spouse
Take Up Rates - Ill Parent
Share of Dual Receivers

* *Simulation* tab– This tab contains the following parameters*:*
  + 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. Current options in this dropdown menu include CA (California), NJ (New Jersey), and RI (Rhode Island), the three states that have published sufficient program data for estimating the needed model parameters.

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

Exhibit 4

Exhibit 4 shows the graphic user interface (GUI) with model parameters displayed under the Simulation Tab, including the following parameters:
Existing State Program


* + Compare Button – When clicked, this button will show the full list of parallel simulation to run for comparison, next to the *Compare* button. Each parallel simulation will be performed using an alternative set of program parameters from a selected state, based on user choice of state in a separate *Existing State Program* dropdown list for that parallel simulation. For example, in **Exhibit 5**, after clicking *Compare*, a new button ‘*Main Simulation*’ appears, representing the current simulation. Next the ‘*Main Simulation*’ button is a “+” button, which can be clicked to add additional parallel simulations for comparison. For example, after clicking the “+” button once, a *“Comparison 1*” button will appear. Once user clicks “*Comparison 1*”, the model parameter template that consists of *Program*, *Population*, and *Simulation* tabs will be refreshed as default values, awaiting for user inputting desired values for the simulation *Comparison 1*. Likewise, user can further add more parallel simulations by clicking the *plus* button next to *Compare*, resulting in *Comparison 2*, *Comparison 3*, etc. The comparison results will be eventually displayed in GUI result window, as well as saved in the output folder. To remove existing parallel simulation such as *Comparison 1*, use can click the *Comparison 1* button, and a cross mark will appear, allowing user to click and remove *Comparison 1*.

**Exhibit 5: Comparison Button**

Exhibit 5

Exhibit 5 shows the graphic user interface (GUI), with Comparison button activated, and an example parallel simulation name Comparison 1 added.

* *Advanced* *Parameters* button – When clicked, this button will display advanced parameters in GUI as shown in **Exhibit 6**. Current model has following advanced parameters:

Advanced Parameters under Main Panel

* + 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. This parameter is checked as default, following common practice of currently implemented state programs.
  + Year – This represents the ending year of the 5-year ACS period. For example, if 2016 is chosen, then ACS PUMS 2012-2016 will be used.
  + 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 under General Linear Model (GLM), Logistic Regression, Ridge Classifier, K Nearest Neighbor, Naïve Bayes, Support Vector Machine, and Random Forest.[[4]](#footnote-4)
  + Random seed – The seed can be set to any integer between 0 and 232. Given a seed value, the model running will correspond to a machine-generated pseudo-random state. Therefore, under the same seed value, the simulation results will remain identical. Specifying random seed can help model users control the effect of randomness across different simulation runs.
  + Engine Type – User can specify whether model should be run using Python or R.

**Exhibit 6: Advanced Parameters**

Exhibit 6

Exhibit 6 shows the graphic user interface (GUI), with Advanced Parameters switch turned on, showing following advanced parameters under Program tab:
Leave Types Allowed - Own Health, Maternity, New Child, Ill Child, Ill Spouse, and Ill Parent
Waiting Period
Dependency Allowance
Recollect

Advanced Parameters under Program Tab

* + Leave Types Allowed – User can use these checkboxes to specify leave types allowed under the program. When a type is unchecked, the corresponding value in Max Weeks will be overridden. For example, if Ill Parent is unchecked, then the value of Ill Parent in Max Weeks would become irrelevant, as the model will not simulate leave taking behavior for this leave type.
  + Waiting Period – This is the number of days that applicants need to wait, starting from submission of program application to receive benefit payments. Current programs have a typical length of waiting period of 5 days. By default, the waiting period is not covered by the program.
  + Dependency Allowance (**Exhibit 7**) – This checkbox indicates whether wage replacement ratio should be adjusted based on number of eligible dependents of applicant. Eligible dependents include non-working spouse and children under 18. When checked, user can use the “+” and “–” buttons to add or remove dependents, and enter the corresponding *increment* in replacement ratio of each additional dependent added. Note that the rightmost value corresponds to number of dependents equal to *or more than* the maximum number specified, and if for any workers with replacement ratio over 1 after accounting for dependency allowance, the model will cap the total replacement ratio at 1.
  + Recollect (**Exhibit 7**) – This checkbox indicates whether workers can recollect the benefits that would occur during the waiting period, subject to the length of waiting period specified above. When checked, an additional parameter *Minimum Leave Length* will be required. This is the minimum leave length threshold below which recollection is not allowed. For example, if *Minimum Leave Length* = 5, then leave length applied under the program must be at least 5 days so that the worker can recollect benefits of the waiting period.

**Exhibit 7: Advanced Parameters: Dependency Allowance and Recollect**

Exhibit 7

Exhibit 7 shows the graphic user interface (GUI), with Advanced Parameters switch turned on, showing following advanced parameters under Program tab:
Dependency Allowance (checked)
Numbers of dependents
Increment of Replacement Ratio associated with each additional dependent
Recollect (checked)
Minimum Leave Length

Advanced Parameters under Population Tab

* + Minimum Leave Length Applied (**Exhibit 8**) – This is the minimum leave length assumed by the model that workers will apply for. Default value is set to 5 work days, which is inferred from current state program’s reporting of benefit distribution and caseload using weekly statistics. For leave lengths that are allocated to the program by simulation model, if they are below this threshold value, the model assumes that workers would not use the program to cover the leave. This also reflects the learning and time costs of workers associated with the application process, so that the leave length needs to be sufficiently long to motivate program use.
  + Alpha (**Exhibit 8**) – This is a parameter that controls the weighted random draw of program takers from leave takers simulated within the ACS sample. Alpha can take any numeric value. For each leave type, if Alpha = 0, workers are randomly drawn with equal weights from simulated ACS leave takers, until reaching the target program taker population, which is the product of the take up rates specified for the leave type, and the estimated eligible population for the program within the ACS sample. For Alpha > 0, the random draw would bias towards workers with longer leave lengths, i.e. workers with more leave needs are assumed more likely to use the program. For Alpha < 0, the random draw would bias towards workers with shorter leave lengths.

**Exhibit 8: Advanced Parameters under Population Tab**

Exhibit 8

Exhibit 8 shows the graphic user interface (GUI), with Advanced Parameters switch turned on, showing following advanced parameters under Population tab:
Minimum Leave Length Applied
Alpha

Advanced Parameters under Simulation Tab

* + Clone factor – Under the Simulation tab, user can specify an integer value to clone the ACS sample, thus be able to obtain more granular simulation results for more ACS persons, with the population weight of each person reduced proportionally to ensure proper aggregation. This feature can be useful for smaller ACS samples (such as smaller states) where simulation results may not reach a desirable level of sample variation.
* *Run* button – After configuring all parameters above, user may click the Run button to execute the simulation program

2.3 Executing the model

* Runtime display – After *Run* button being clicked, a runtime window will be displayed as in **Exhibit 9**. 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. At the beginning, warning message will be displayed if user’s Python environment has modules for which update is recommended.

**Exhibit 9: Runtime Display during Model Execution**

Exhibit 9

Exhibit 9 shows the graphic user interface (GUI) after clicking the Run button, and a separate runtime window displaying the model execution progress bar and messages indicating the steps completed during model execution, such as
File saved: clean FMLA data file before CPS imputation.
CPS cleaned and CPS variable imputation done for FMLA. Time elapsed = 9.94 seconds.
File saved: clean FMLA data file after CPS imputation.
Cleaning ACS data. State chosen = RI. Chunk size = 100000 ACS rows.

* 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 GLM*, 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.
* Users should note that if multiple comparison simulations are performed, runtime would increase as simulation steps will be performed multiple times on ACS data. However, the data loading and cleaning steps for FMLA and ACS data would be performed only once.
* Possible Errors – The runtime window will display any errors the model runs into during execution, including the following:
* FileNotFoundError: This error occurs when the needed input data files cannot be located. Typically, this is caused by either (i) any missing files in the subfolders of *./data/*, including *./data/fmla\_2012*, *./data/acs*, and *./data/cps*, or (ii) any mis-named files in *./data/acs*, and *./data/cps*.[[5]](#footnote-5)

**Exhibit 10: Example File Not Found Error**

Exhibit 10

Exhibit 10  shows runtime window displaying the model execution progress bar and messages indicating the steps completed during model execution, with an example error message indicating the following:
File Not Found Error: File does not exist


2.4 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 11**, 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 11: Summary Tab in Result Window**

Exhibit 11

Exhibit 11 shows the result window after model execution, displaying the Summary tab that contains a bar chart with following data:
State RI. Total Benefit Cost = $262.2 million, with 95% confidence interval half width of $60.8 million.
Vertical Axes is millions of dollars
Horizontal Axes is leave types.
From left to right:
Own Health: about 60 million
Maternity: about 120 million
New Child: about 40 million
Ill Child: about 20 million
Ill Spouse: about 20 million
Ill Parent: about 20 million

* + *Benefit Financing* tab – As shown in **Exhibit 12**, 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 12** shows that about $61 million would be generated by a 1% payroll tax, thus given the $262 million program outlay estimate in **Exhibit 11**, it can be estimated that the program would be funded at 61/262 = 23.3% by this payroll tax.

The *Benefit Financing* tab offers a convenient recalculation tool for updating benefit financing results based on alternative benefit financing parameters. The tool can be activated by clicking the *ABF Parameters* button at bottom right corner of the tab (shown in **Exhibit 12**(b)). After activation, user can input desired alternative parameter values for *Payroll Tax Rate*, *Minimum Taxable Earnings*, and *State Income Tax Rate* if the *Apply Benefits Tax* option is checked (shown in **Exhibit 13**).

**Exhibit 12: Benefit Financing Tab in Result Window**

|  |  |
| --- | --- |
| (a) | (b) |
| Exhibit 12a  Exhibit 12a shows the result window after model execution, displaying the Benefit Financing tab with following data and graphs: Total Income: $6126.88 million, with 95% confidence interval half width of $163.2 million Total Tax Revenue: $61.27 million, with 95% confidence interval half width of $1.6 million Tax Revenue Recouped from Benefits: $0 million A bar chart showing break down of tax revenue by gender | Exhibit 12b  Exhibit 12b continues Exhibit 12a, showing the result window after model execution, displaying the second half of the Benefit Financing tab (after scrolling down to the end) with following graphs: A bar chart showing breakdown of tax revenue by employer of workers including Federal government, state government, local government, self,, and private sector. A bar chart showing breakdown of tax revenue by age of worker. By the end of the Benefit Financing Tab, there is a ABF Parameter button. |

**Exhibit 13: Benefit Financing Recalculation Tool**

Exhibit 13

Exhibit 13 shows the result window after model execution, displaying the second half of the Benefit Financing tab (after scrolling down to the end) with the ABF Parameters button clicked, thus activating the Benefit Financing Recalculation Tool. The Tool is a box within the result window where the following parameters can be updated:
Payroll Ta Rate
Maximum Taxable Ernings
Apply Benefits Tax
State Income Tax Rate.
At the bottom of the toolbox, there is a Hide button to hide the toolbox, and a Run ABF button to use the updated parameters to re-calculate the ABF results for which the new results would be reflected in the updated numbers and graphs in the Benefit Financing tab.

* + *Population Analysis* tab – As shown in **Exhibit 14**, this tab plots histogram of annual total leave length (in days) taken by eligible workers. Each histogram corresponds to a different simulation (e.g. Main/Comparison 1). 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 14: Population Analysis Tab in Result Window**

Exhibit 14

Exhibit 14 shows the result window after model execution, displaying the second half of the Population Analysis tab, showing the following features and graphs:
There is a panel where user can specify demographic filters including gender, range of age, and range of annual wage of workers
Under the filter panel, there are histograms showing the distribution of number of workers across different leave length in number of days. Different histograms correspond to different parallel simulations specified by the user. Exhibit 14 shows a histogram for Main Simulation, and another histogram for the parallel simulation named Comparison 1.

* 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 15** 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* is a unique identifier for parallel simulation under comparison, such as *main simulation*, *comparison 1*, etc.

**Exhibit 15: Files in Output Directory** Exhibit 15

Exhibit 15 shows a screenshot of the output folder after completing the simulation.

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 master post-simulation ACS state PUMS data file containing individual level taxable income and tax revenue collected.
  + An administrative budget financing (ABF) summary containing the tax revenue statistics from the *Benefit Financing* window.
  + 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.

## References

Census (2020). American Community Survey PUMS Data. Retrieved at <https://www.census.gov/programs-surveys/acs/data/pums.html> on March 4, 2020.

Department of Labor (2020). Family and Medical Leave in 2012. Retrieved at <https://www.dol.gov/agencies/oasp/evaluation/fmla/fmla2012> on March 4, 2020.

Scikit Learn User Guide (2020). 1.1. Linear Models./1.1.11. Logistic Regression. Retrieved at <https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression> on March 4, 2020.

1. The entire Python model can be placed at any desired local directory, and the directory chosen will be recognized as “*./* ” by the model. By default, the directory that stores the ACS data set is *./data/acs/*. However, users also have the option to specify desired local directory to store the ACS datasets in the graphic user interface (GUI) of the model. More details are provided in Section 2 *Running the model*. [↑](#footnote-ref-1)
2. *The file names of original ACS household and person files follow the file names of 2012-2016 ACS PUMS as they were downloaded from the* [*Census website*](https://www.census.gov/programs-surveys/acs/data/pums.html)*.* [↑](#footnote-ref-2)
3. Please note that double quoting is needed for file path that contains spaces. For example, if current directory in PowerShell is *C:\* and user needs to navigate to *C:\microsim project*, user would need to type *cd “microsim project”* and hit enter. [↑](#footnote-ref-3)
4. The *Logistic Regression* *GLM* option represents the traditional logistic regression without penalization. It is implemented via the *statsmodel* module in Python, and is equivalent to the *glm* implementation of logistic regression in R. On the other hand, the *Logistic Regression* option represents the *sklearn* implementation of logistic regression in Python, which features L2-penalization to limit complexity of model parameters. See Scikit Learn User Guide (2020) for more details. [↑](#footnote-ref-4)
5. See 1.2 *Dataset Requirements* in Section 1 Setting up the computing environment for details on requirements on file names of ACS and CPS data files. [↑](#footnote-ref-5)