**Microsimulation**

**Model Documentation **

First Draft

November 2, 2018



**Microsimulation**

**Model Documentation for R Implementation**

November 2, 2018

CONTENTS

[Chapter 1. Introduction 6](#_Toc529203697)

[1.1. Purpose of Report 6](#_Toc529203698)

[1.2 Model Overview 6](#_Toc529203699)

[1.3 Structure of Report 6](#_Toc529203700)

[Chapter 2. Model Overview 7](#_Toc529203701)

[2.1 Model Purpose 7](#_Toc529203702)

[2.1.1 Microsimulation Model 7](#_Toc529203703)

[2.1.2 Benefit Financing Module 7](#_Toc529203704)

[2.2 Model Inputs 7](#_Toc529203705)

[2.2.1 FMLA Survey 7](#_Toc529203706)

[2.2.2 American Community Survey 7](#_Toc529203707)

[2.2.3 Current Population Survey 7](#_Toc529203708)

[2.2.4 User-Defined Inputs 7](#_Toc529203709)

[2.3 Model Outputs 7](#_Toc529203710)

[2.4 User Interface 7](#_Toc529203711)

[Chapter 3. R Implementation 8](#_Toc529203712)

[3.1 Overview of Code Structure 8](#_Toc529203713)

[3.2 Main Simulation Function 8](#_Toc529203714)

[3.2.1 Preprocessing 9](#_Toc529203715)

[3.2.2 CPS Imputation 9](#_Toc529203716)

[3.2.3 FMLA Imputation 9](#_Toc529203717)

[3.2.4 ACS Simulation 10](#_Toc529203718)

[3.3 Pre-processing Functions 10](#_Toc529203719)

[3.4 Imputation Functions 10](#_Toc529203720)

[3.4.1 CPS Imputation 11](#_Toc529203721)

[3.4.2 FMLA Imputation 12](#_Toc529203722)

[3.4.2 ACS Imputation 12](#_Toc529203723)

[3.5 Simulation Functions 12](#_Toc529203724)

[3.5.1 Logistic Regression 12](#_Toc529203725)

[3.5.2 K-Nearest Neighbors 12](#_Toc529203726)

[3.5.3 Random Forest 12](#_Toc529203727)

[3.6 GUI functionality 12](#_Toc529203728)

[3.7 File Summary 12](#_Toc529203729)

[Chapter 4. Python Implementation 14](#_Toc529203730)

[3.1 Overview of Code Structure 14](#_Toc529203731)

[3.2 Main Simulation Function 14](#_Toc529203732)

[3.3 Pre-processing Functions 14](#_Toc529203733)

[3.4 Imputation Functions 14](#_Toc529203734)

[3.4.1 CPS Imputation 14](#_Toc529203735)

[3.4.2 FMLA Imputation 14](#_Toc529203736)

[3.4.2 ACS Imputation 14](#_Toc529203737)

[3.5 Simulation Functions 14](#_Toc529203738)

[3.5.1 Logistic Regression 14](#_Toc529203739)

[3.5.2 K-Nearest Neighbors 14](#_Toc529203740)

[3.5.3 Random Forest 14](#_Toc529203741)

[3.6 GUI functionality 14](#_Toc529203742)

[Chapter 5. Algorithms 15](#_Toc529203743)

[5.1 Logistic Regression 15](#_Toc529203744)

[5.1.1 R Implementation 15](#_Toc529203745)

[5.1.2 Python Implementation 15](#_Toc529203746)

[5.2 K-Nearest Neighbor 15](#_Toc529203747)

[Chapter 6. Validation 16](#_Toc529203748)

[6.1 FMLA Internal Validation 16](#_Toc529203749)

[6.2 Robustness Checking 16](#_Toc529203750)

[6.3 Sensitivity Checks 16](#_Toc529203751)

[6.4 Standard Error Estimation 16](#_Toc529203752)

[Chapter 7. Conclusion 17](#_Toc529203753)

[Bibliography 18](#_Toc529203754)

**Table of Exhibits**

[Exhibit 1: Overview of Model 6](#_Toc528747820)

[Exhibit 2: Overview of R Code Structure 8](#_Toc528747821)

[Exhibit 3: Summary of inputs to policy\_simulation() function 8](#_Toc528747822)

[Exhibit 4: Pseudocode for K-Nearest Neighbor 11](#_Toc528747823)

# Chapter 1. Introduction

## 1.1. Purpose of Report

This document describes in detail the implementation of the IMPAQ-DOL worker leave microsimulation model. It describes the purpose and functionality of the model from a high level, as well as detailing the technical methodology. Further included are instructions on the set up and use of the microsimulation model for typical users, as well as finer configuration options for advanced users seeking additional customization.

## 1.2 Motivation and Background

Nearly every developed country in the world has a public paid maternity leave available for its workers. However, the United States remains an outlier; there is no federal requirement for employers to offer paid leave or sick days to employees. As a result, access to leave is scant among the American workforce. In 2016, only 14 percent of all US workers have access to paid family leave through their employers, and 68 percent have paid sick leave.[[1]](#footnote-2) Absent federal policy, Some states and municipalities have moved forward on paid family leave. California enacted paid family leave legislation in 2002, New Jersey in 2008, Rhode Island in 2013, New York in 2016 (effective January 2018), the District of Columbia in 2017 (effective July 2020), and Washington in 2017 (effective January 2020). Other states and municipalities have shown interest in the feasibility of adopting their own paid-leave policy.[[2]](#footnote-3) However, one often-cited obstacle to providing paid family and medical leave in the US is the anticipated cost. Policymakers need to ensure the program will be financially sustainable.

Estimating cost of state and local paid leave programs are not straightforward due to limitations of current data sources. The lion’s share of program costs are wage replacement benefits paid out to eligible leave takers. To properly estimate this, policymakers must estimate the number of eligible workers within their constituency is and their leave taking behavior. The best data source available for estimating leave taking behavior of the US population is USDOL’s Family and Medical Leave survey (referred to as the FMLA survey).[[3]](#footnote-4) The survey captures in great detail the leave taking behavior and demographic characteristics from a nationally representative sample. However, with only 2852 respondents, insufficient sample size and respondent privacy become barriers to employing traditional data analysis techniques at the state and local level to obtain these estimates.

Microsimulation methods overcome these barriers through imputing leave taking behavior observed in the FMLA data to the larger, more robust American Community Survey (ACS) from the Census Bureau. The modeling approach relies on the significant overlap of demographic characteristics (such as age, sex, and race) collected in both the FMLA and ACS surveys that are related to leave taking behavior. The associations between these characteristics and behavior in FMLA data have been fit to ACS data via logistic regressions in previous microsimulation models. [Research on ACM and other previous models]. With a larger and more robust sample, ACS is then able to more precisely estimate leave taking behavior at a national or state level.

## 1.3 Model Overview

The IMPAQ-DOL model is a robust, accessible tool to assist in the design of paid leave programs and research of leave taking behavior at the national and state level. The simulation model proceeds in six broad steps as indicated in Exhibit 1. First, the input data sets are individually cleaned and prepared for use in the model. Second, FMLA data is used to calibrate the leave taking estimation model for application in ACS data. Third, leave taking behavior is imputed on an ACS data set using the estimation model. The ACS data is selected based on the user-defined geography of interest; be it national-level leave taking, or leave taking for a specific state. Fourth, participation and benefits received are calculated in the ACS using user-specified leave program characteristics and behavioral assumptions. Fifth, if the user has elected to calculate what tax structure will be required to pay for the program, the benefit financing module calculations are run. Finally, the model displays simulation results and financing estimates in tabular and graphical form.

Exhibit 1: Overview of Model

6. Output simulation results and estimates

3. Impute leave taking for ACS data

4. Adjust ACS data based on the characteristics of the simulated leave program

1. Clean Census & FMLA input data sets

2. Calibrate leave taking estimation model from FMLA data

5. Apply benefit financing module calculations

## 

## 1.4 Structure of Report

This report proceeds as follows. Chapter 2 presents a high level overview of the microsimulation model by describing its purpose, its main components and its logical flow. This chapter is aimed at guiding less technical users who may not be familiar with the intricacies of microsimulation modeling and R programming. Chapter 3 describes the R implementation, detailing the primary functions and interconnectedness of the various files and functionalities. This chapter is aimed at the technical programmer familiar with modelling in R.

# Chapter 2. Model Overview

This chapter is intended to provide enough detail for the average user to understand the broad picture of what the model intends to accomplish, what the model’s inputs and outputs are, and how the average user can specify and run a leave-taking simulation. This chapter focuses mostly on the big picture of what the model intends to accomplish, and how the user can utilize the Graphical User Interface (GUI) to customize this. Detailed technical discussions of internal, back-end components are reserved for Chapters X-X.

## 2.1 Model Purpose

The primary purpose of this model is to provide a robust, accessible tool to assist in the design of paid leave programs and research of leave taking behavior at the national and state level. There are two distinct components of this model; the microsimulation module and the benefit financing module.

### 2.1.1 Microsimulation Module

This microsimulation module’s primary purpose is to provide accurate estimates of leave taking behavior and leave program participation for both the US and individual states. To facilitate broader use of this model compared to predecessors, we designed the model and its results to be accessible, flexible, and transparent to a non-technical audience. Our focus in development was also to on the technical performance of the model. We have built and tested a number of behavioral estimation methods to establish the best way(s) to accurately perform leave taking estimation.

### 2.1.2 Benefit Financing Module

To start a paid leave program, a state needs to know not only what the program will cost, but what tax structure or financing plan will be adequate to cover the program’s costs. After the microsimulation model produces the amount of estimated benefit payouts, the benefit financing module helps the user obtain this. As inputs, this module uses the same ACS data and user inputs on a theoretical tax structure to simulate. The module outputs the simulated annual revenue from the tax structure, and compares it with the annual amount of benefits that will be paid out.

While bulk of program costs are leave benefit payouts, there are administrative and procedural costs to running a paid leave program. For example, staff time must be spent validating eligibility, checking for improper payments, and investigating evidence of fraud. The benefit financing module also includes a tool to calculate these kinds of costs based on user inputs.

Altogether, the benefit financing model allows users to easily take the output from the microsimulation model and come up with a tax structure or other financing plan to cover the costs of the theoretical leave program in question.

[ABF team to elaborate]

## 2.2 Model Inputs

### 2.2.1 2012 FMLA Survey

USDOL’s 2012 FMLA survey behavior is the third wave of a cross-sectional survey on paid leave.[[4]](#footnote-5) Respondents are asked about leave taking behavior in great detail, including: the number, lengths, and types of leaves taken, to what extent the employer provided pay while on leave, and whether or not some or additional pay while on leave would alter their leave-taking behavior. This survey’s data is this model’s primary data source for leave taking behavior in the US.

The survey interviewed 2852 employees, of which 1551 responded that they took or needed to take leave in the past 18 months. These 1551 respondents provided details on the leave(s) they took or needed to take. State of residence is not available in the survey due to risk of personal identification. As a result, this data alone cannot inform state-level estimates of leave taking.

### 2.2.2 American Community Survey

The American Community Survey (ACS) is a large national representative sample of individuals within the US. The ACS is conducted on a continuous basis, but public use ACS data is released on an annual basis in 1-year and 5-year data sets.

This model uses 2012-2016 ACS 5-year data (referred to as ACS data onwards) to maximize sample size and to coincide with the 2012 FMLA survey. This adds an implicit assumption to our model’s estimates: that 2012 status-quo leave taking behavior did not change significantly over the 2012-2016 time period. To simulate leave taking in a single state, the model filters the national ACS dataset to only include observations from the given state. To simulate nation-wide leave taking, no such filter is applied and the entire ACS data set is used.

In total, the entire ACS data set contains 12.9 million observations, and individual states range from approximately 20,000 (Wyoming) to 1.1 million (California) observations.

### 2.2.3 Current Population Survey

While the ACS has a very rich set of variables, there are a handful of variables the model requires that ACS does not contain or contains insufficient detail on. These variables are: whether pay is received on an hourly basis; employer size; the number of employers that the person worked for in the last 12 months; and weeks worked in the last 12 months.

To estimate these, we use the Current Population Survey (CPS). The CPS is another nationally representative Census survey that does contain the required variables in sufficient detail. We follow the ACM model’s method of imputing these values via logistic or ordinal logistic regression from the CPS to the ACS on a set of overlapping demographic variables. [may need to change this language if we make this imputation modular]

### 2.2.4 User-Defined Inputs

Prior to executing the simulation, the model allows the user many different options to modify the simulation through a graphical user interface (GUI). The GUI user inputs fall into three main categories:

* **Program inputs:** Inputs that define the characteristics, rules, and benefits of the leave program to be simulated. (e.g. weekly program benefits paid, and maximum length in weeks
* **Behavior inputs:** Inputs that define the assumptions to be used for simulating the population’s behavioral response to the presence of a leave program (if one is specified).
* **Advanced inputs:** Other **i**nputs the average user is not expected to use often, but advanced users may wish to utilize.

## 2.3 Model Outputs

The model’s output is at its core a modified version of the ACS data file. The original ACS data file is essentially a very large spreadsheet, with each row representing an individual in the ACS and each column a different characteristic variable. The simulation takes this file, and adds several columns to represent leave taking and program participation behavior. From there, the model computes the summary data necessary to create the charts and tables shown in the GUI. Exhibit 3 is a visualization of the model’s output data set and Exhibit 4 is an example graph produced by the GUI from the model output data set.

Exhibit 3. Simulation Output Dataset Visualization [Need a more polished version of this]

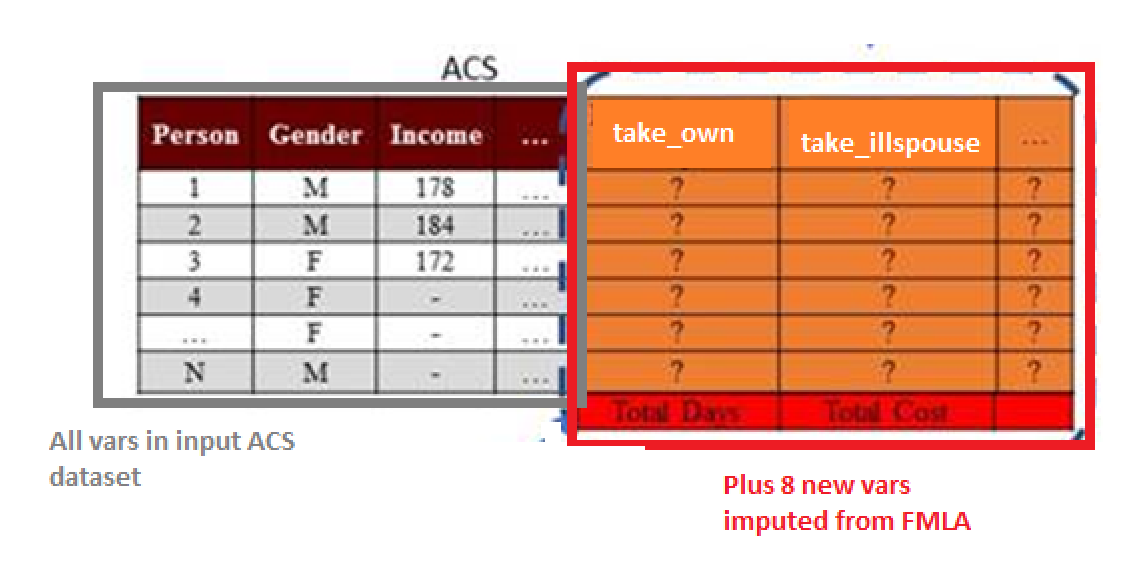
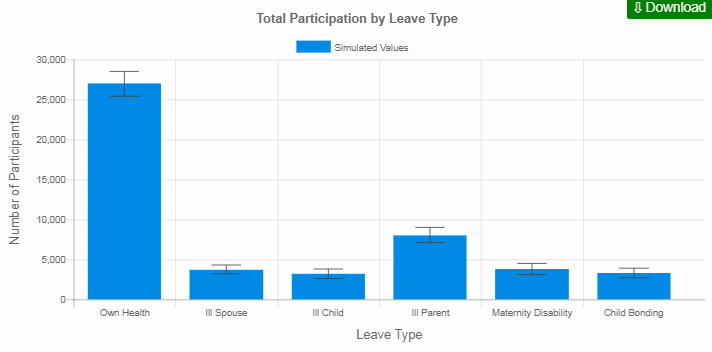


Exhibit . Simulation Output Graph Example [Need a more polished version of this]



[Description of ABF outputs]

## 2.4 User Interface

The actual implementation of the Graphical User Interface (GUI) from the perspective of source code is described in Chapter X. Once the user submits their desired inputs and the simulation completes execution, the GUI also displays the output results of the simulation. User parameters are divided among three different tabs, each corresponding to the types of parameters mentioned in the previous section: Program inputs, Behavior inputs, and Advanced inputs. Exhibit 2 is a sample screen shot of this GUI. A full description of the GUI and each of its field can be found in Appendix A.

Exhibit : Sample Screenshot of GUI [needs to be updated]



The outputs displayed by default in the GUI are [elaborate on GUI output displays once finalized].

# Chapter 3. R Implementation

This chapter describes in detail how the microsimulation model is implemented in the R programming language. It describes the purpose and interaction of the functions, variables, and files that comprise the model. This chapter is designed to provide a developer with the full information required to edit the underlying source code of each stage of the model. Section 3.1 provides an overview of

## 3.1 Overview of Code Structure

At a very high level, there are two key functions being performed by the model. The first is to infer the leave taking behavior of individuals within the ACS based on the leave taking behavior of comparable individuals within the FMLA. Because the sample size of the ACS is much larger than the FMLA, accurate inference of leave behavior within the ACS allows for broader leave across a broader demographic and geographic range within the United States. The ACS also asked a much wider range of demoquestions . This “inference component” of the microsimulation is purely statistical, in the sense that it is purely an imputation procedure.

The second key function of the model is to simulate counterfactual leave-taking behavior based on the presence of a different leave taking program. This component relies instead on assumptions regarding how individuals are likely to respond to changes in the “leave taking environment.”

The implementation in R is broken down into 5 file types, as demonstrated in Exhibit X.

Exhibit 5: Overview of R Code Structure

## 3.2 Main Simulation Function

The main function called by the GUI to run the simulations is policy\_simulation(), which is defined within the 0\_NEW\_master\_execution\_function.R file. It accepts x arguments. All arguments are optional except for the filenames of the four input datasets in csv format. Exhibit 3 presents a summary of each of the function inputs. The function returns the ACS dataset with additional columns representing the leave behavior of X.

Exhibit 6: Summary of inputs to policy\_simulation() function

|  |  |  |
| --- | --- | --- |
| Input Name | Default | Description |
| fmla\_csv | - | Cleaned FMLA data in csv format |
| acs\_house\_csv | - | Household Level ACS data in csv format |
| acs\_person\_csv | - | Individual Level ACS data in csv format |
| cps\_csv | - | CPS data in csv format |
| General | impute\_method | Boolean indicating whether model computes counterfactual leave program |
|  |  |  |
| General | leaveprogram |  |
|  |  |  |
| Program | base\_bene\_level | Desired sample proportion to be used from ACS (to speed up runtime) |
| Behavior | bene\_effect | Desired sample size to be used from ACS (to speed up runtime) |
|  |  |  |

The function proceeds in 5 broad steps – per-processing, CPS imputation, which are detailed in the following subsections.

[Maybe an Exhibit here]

## 3.3 Pre-Processing

This section describes the functions used to prepare the four raw datasets for use in the model. The function first reads in the raw datasets and cleans them using the clean\_fmla(), clean\_acs()and clean\_cps() functions which are all defined in the 1\_NEW\_cleaning\_functions.R file. These functions, which are described below, each result in cleaned datasets formatted to be used conveniently by the functions in the rest of the model.

1. The clean\_fmla()function takes …
2. The clean\_acs()function takes … To improve runtime (at the cost of higher sampling variation), the user also has the option to specify the use of a subsample of the cleaned ACS dataset using the sample\_prop and sample\_num parameters described in Exhibit 3.
3. The clean\_cps()function takes …

## 3.4 Pre-Imputation

The FMLA and the ACS are the key data sources driving the microsimulation model. However, each dataset is missing crucial information required for the model simulation. It is therefore necessary to “fill in the gaps” using imputation methods in order for the datasets to be useful. This section explains why the imputations are needed and how they are performed in each dataset.

### 3.4.1 ACS Imputation

There are four important variables within the FMLA that are not available within the ACS. These variables are key determinants/predictors of leave behavior within the FMLA and are required to infer leave-taking behavior within the ACS. These variables are the hourly worker status, the number of employers, the number of weeks worked and the size of the main employer for each individual. The CPS is used to impute these values within the ACS using the impute\_cps\_to\_acs() function contained within the 2\_NEW\_impute\_functions.R file, which results in the preprocessed ACS dataset produced by the clean\_acs()function above with four additional columns representing the four imputed variables. The imputation can be performed with any of the imputation methods and functions discussed in Section 3.X. The default imputation procedures, however, are based on the logistic regression methods used in the original ACM model. Exhibit 5 presents a summary of the functions used to implement these methods.

Exhibit 7: Summary of Logistic Regression Imputation Functions

|  |  |  |
| --- | --- | --- |
| Function Name | Inputs | Description |
| runLogitEstimate() | * Dataset * Dependent/independent variable names * Data filtering condition | Estimates a logistic regression based on the specified dependent/independent variables on the supplied dataset and returns the set of parameter estimates. |
| runLogitImpute() | * Dataset * Parameter estimates from above * Variable name to be created * Data filtering condition | Using the estimates from above, the function computes the implied probability values for each row in the dataset based on the same independent variables. It then simulates the dependent variable by comparing a random uniform draw with the computed probability and adds this result as a column to the data frame. |
| runOrdinalEstimate() | * Dataset * Dependent/independent variable names * Data filtering condition | Estimates an ordinal logistic regression based on the specified dependent/independent variables on the supplied dataset and returns the set of parameter estimates. |
| runOrdinalImpute() | * Dataset * Parameter estimates from above * Variable name to be created * Data filtering condition | Using the estimates from above, the function computes the implied “cutoff” values for each of the categories. It then simulates the dependent variable by comparing a random uniform draw with the intervals computed above and adds this result as a column to the data frame. |

The missing variables are imputed using the above functions as follows:

1. **Paid hourly status.** A logistic regression for paid hourly status is estimated by applying the runLogitEstimate()to the pre-processed CPS using dependent variables identical to those used by ACM. The resulting estimates are then used to compute the probability that each worker in the ACS is paid hourly based on the same dependent variables using the runLogitImpute() function. A new binary variable indicating whether a worker is paid hourly is created by comparing a uniform random draw with these probabilities.
2. **Number of employers.** An ordinal logistic regression for the number of employers of a given individual is estimated by applying the runOrdinalEstimate()function to the pre-processed CPS. The dependent variables are again identical to those used by ACM. The resulting estimates are then used to construct intervals on the unit line with the runOrdinalImpute()function. Each interval represents the number of employers and a new variable is created in the ACS based on where a random uniform draw falls within the intervals.
3. **Number of Weeks Worked.** The ACS indicates the number of annual weeks worked by an individual within bins of <13 weeks, 14-26 weeks, 27-39 weeks, 40-47 weeks, 48-49 weeks and 50-52 weeks. We impute the exact number of weeks within each of these bins based on an ordinal logistic regression (or a regular logistic regression in the case of the binary 48-49 weeks bin) using dependent variables and filtering conditions specific to each bin within the runOrdinalEstimate() (or runLogitEstimate()) function. Like paid hourly status and number of employers above, these estimates are used to construct an interval on the unit line which is divided by the exact number of weeks. A new variable is then constructed by comparing a uniform random draw with this interval.
4. **Employer Size.** An ordinal logistic regression for the size of the individual’s main employer measured by the number of employees at that firm is estimated with the runOrdinalEstimate()function. The dependent variables are again identical to those used by ACM. These estimates are used to construct an interval on the unit line divided by firm size. A firm size is then allocated to an individual in the ACS based on a random uniform draw within the runOrdinalImpute() function.

### 3.4.2 FMLA Imputation

A key determinant of the cost of a leave program is the type and the length of leaves. Within the FMLA, this information is only available for a maximum of two unique leaves – their last leave taken and their longest leave taken – in the previous 12 months for an individual survey respondent. Therefore, there are some circumstances under which we do not have complete leave length or type information for a respondent. To understand when this might occur, consider the examples presented in Exhibit 4. All leave information is available for individuals A and B, who took just one and two leaves, respectively, in the previous 12 months. On the other hand, although individual C also took two leaves, her latest leave taken also happened to be her longest leave over the previous 12 months. Therefore, we do not have information regarding her other (shorter) leave that she took. In the case of individual D, although we have information for two of her leaves, we do not know anything about her third leave.

Exhibit 8: Example Leave Information Responses in FMLA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Individual ID | Number of Leaves | Latest Leave Length | Latest Leave Type | Longest Leave Length | Longest Leave Type |
| A | 1 | 5 | Own Health | 5 | Own Health |
| B | 2 | 6 | Ill Spouse | 12 | Ill Child |
| C | 2 | 30 | Maternity | 30 | Maternity |
| D | 3 | 1 | Own Health | 8 | Ill Parent |

Individuals like C and D demonstrate the need to impute leave characteristics for some individuals. This is done in the impute\_intra\_fmla()function contained within the 2\_NEW\_impute\_functions.R file. Like the ACS imputation above, this imputation can be performed with any of the methods discussed in Section 3.X. Again, however, the default uses the logistic regression functions summarized in Exhibit 5.

using a simple logistic regression to estimate the probability that an individual takes a particular leave type, which is then used to simulate the leave types and lengths for an individual’s unknown leave types. This produces FMLA data with adjusted take\_leave columns to include 1s for those that would have taken leave if they could afford it. This step effectively identifies the effect of the program on the extensive margin by identifying new individuals that would take leave. The LEAVEPROGRAM()function is a simple function that

It takes the entire FMLA dataset as an input and the logic of the function proceeds as follows:

* Part A: [Need Luke to explain this part]
* Part B: For individuals that indicated taking multiple leave, we need to infer the type of leave for each of those. This is done using logistic regression with the specifications (dependent variables, sample conditionals and weights) identical to those used to perform the same task in the original ACM model. This is done using the runLogitEstimate() function, which is a straightforward application of the svyglm() function from the survey package. Using the estimates from these regressions, we apply them to each individual’s characteristics to generate a probability that they took a specific type of leave using the runLogitImpute() function. With these probabilities in hand, the model then assign a leave type to each leave that is taken but for which we do not have information on the type. This is done using the add\_leave\_types() function which proceeds as follows. First it merges all the probabilities to the FMLA dataset. [Need to finish this. Explain exactly what is produced.]

## 3.5 FMLA to ACS Inference

## 3.6 Counterfactual Simulation

A key feature of the microsimulation model is to predict changes in leave behavior resulting from changes in leave program parameters such as the wage replacement rate, the number of days and the type of eligible leave etc. This counterfactual simulation is performed by the LEAVEPROGRAM() function within the 2\_NEW\_impute\_functions.R file. The function is triggered by the user setting the leaveprogram input parameter of the main policy\_simulation() function to TRUE.

The LEAVEPROGRAM()function adjusts the “take\_x” variable for leave type x to 1 from 0 if the worker originally wanted to take leave but did not because it was unaffordable.

## 3.7 Simulation Functions

### 3.5.1 Logistic Regression

### 3.5.2 K-Nearest Neighbors

### 3.5.3 Random Forest

## 3.6 GUI functionality

## 3.7 File Summary

This section presents a brief summary of what each file contains, which can often be helpful from a developer’s perspective for ease-of-reference. Exhibit X summarizes the information already presented earlier in this chapter but categorized according to the file in which each function is contained.

Exhibit 9: Summary of Model Files

|  |  |  |
| --- | --- | --- |
| File Name | Function | Description |
| 0\_master | impute\_intra\_fmla | This function Cleaned FMLA data in csv format |
| LEAVEPROGRAM |  |
| 1\_cleaning\_functions | - | Household Level ACS data in csv format |
| 2\_impute\_functions | - | Individual Level ACS data in csv format |
| 3\_post\_impute\_functions | - | CPS data in csv format |
| leaveprogram | True | Boolean indicating whether model computes counterfactual leave program |
| GOVERNMENT |  |  |
| SELFEMP |  |  |
| impute\_method |  |  |
| sample\_prop | NULL | Desired sample proportion to be used from ACS (to speed up runtime) |
| sample\_num | NULL | Desired sample size to be used from ACS (to speed up runtime) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Chapter 4. Python Implementation

## 3.1 Overview of Code Structure

## 3.2 Main Simulation Function

## 3.3 Pre-processing Functions

## 3.4 Imputation Functions

### 3.4.1 CPS Imputation

### 3.4.2 FMLA Imputation

### 3.4.2 ACS Imputation

## 3.5 Simulation Functions

### 3.5.1 Logistic Regression

### 3.5.2 K-Nearest Neighbors

### 3.5.3 Random Forest

## 3.6 GUI functionality

# Chapter 5. Algorithms

This chapter describes the various algorithms available to the programmer at each stage of the model code.

## 5.1 Logistic Regression

**Exhibit 10: Pseudocode for K-Nearest Neighbor**

**Input:**

**Output:** Y

4: **for** ( in 1:) **do**

8: **for** ( in 1:) **do**

11:

12: **end for**

13: **end for**

### 5.1.1 R Implementation

### 5.1.2 Python Implementation

## 5.2 K-Nearest Neighbor

# Chapter 6. Validation

There are many moving parts in this model, each of which has the potential to cause significant errors in the model output. This section details the various validation procedures we performed to ensure robustness of results.

## 6.1 FMLA Internal Validation

## 6.2 Robustness Checking

## 6.3 Sensitivity Checks

## 6.4 Standard Error Estimation

# Chapter 7. Conclusion

# Bibliography

XXX

# Appendix A. PArameter DictionarY – R Model

Needs to be updated

|  |  |  |  |
| --- | --- | --- | --- |
| **GUI Input Label** | **Location in GUI** | **Parameter** | **Description** |
| Imputation Method | General | impute\_method | method for imputation. |
|  |  |  |  |
| Leave Program | General | leaveprogram | Presence or absence of leave program |
|  |  |  |  |
| Wage Replacement Ratio | Program | base\_bene\_level | proportion of pay received as part of program participation |
| Benefit Effect | Behavior | bene\_effect | Whether to apply simulation of behavioral cost to applying to state program |
|  |  |  |  |
| Topoff Rate | Behavior | topoff\_rate | proportion of employers engaging in top-off substitution of paid leave with program benefits |
| Topoff Minimum Length | Behavior | topoff\_min\_length | Min length of leave required for top-off behavrior |
| Weekly Dependent Allowance | Program | dependent\_allow | weekly dependent allowance for those with children |
| Needers Fully Participate | Behavior | full\_particip\_needer | whether or not leave needers always take up benefits |
|  |  |  |  |
| Own Health | Behavior | own\_uptake | user-supplied benefit uptake rate for a given type of leave. |
| Ill Spouse | Behavior | illspouse\_uptake | user-supplied benefit uptake rate for a given type of leave. |
| Ill Child | Behavior | illchild\_uptake | user-supplied benefit uptake rate for a given type of leave. |
| Ill Parent | Behavior | illparent\_uptake | user-supplied benefit uptake rate for a given type of leave. |
| Maternity Disability | Behavior | matdis\_uptake | user-supplied benefit uptake rate for a given type of leave. |
| Bonding | Behavior | bond\_uptake | user-supplied benefit uptake rate for a given type of leave. |
| Waiting Perioid | Program | waiting\_period | how long in working days must leave takers wait to claim leave benefits |
| Clone Factor | Advanced | clone\_factor | Create clones of ACS records |
| Base Extension Effect | Behavior | ext\_base\_effect | Whether to apply base leave extension behavior in presence of program. standard leave extension effect from ACM model |
|  |  |  |  |
| Extend Probability | Advanced | extend\_prob | additional leave extension effect: probability of extension |
| Extend Days | Advanced | extend\_days | additional leave extension effect: fixed days of extension |
| Extend Proportion | Advanced | extend\_prop | additional leave extension effect: proportionate extension |
| Own Health | Program | maxlen\_own | max number of leave type days benefits can be claimed in a year |
| Ill Spouse | Program | maxlen\_illspouse | max number of leave type days benefits can be claimed in a year |
| Ill Child | Program | maxlen\_illchild | max number of leave type days benefits can be claimed in a year |
| Ill Parent | Program | maxlen\_illparent | max number of leave type days benefits can be claimed in a year |
| Maternity Disability | Program | maxlen\_matdis | max number of leave type days benefits can be claimed in a year |
| Bonding | Program | maxlen\_bond | max number of leave type days benefits can be claimed in a year |
| Bonding or Ill Relative | Advanced | maxlen\_DI | max number of bond, ill relative leave days benefits can be claimed in a year |
| Own Health or Maternity Disability | Advanced | maxlen\_PFL | max number of matdis, own leave days benefits can be claimed in a year |
| Total | Advanced | maxlen\_total | max number of total days benefits can be claimed in a year |
| Weekly Benefit Cap | Program | week\_bene\_cap | max weekly benefits that can be collected |
| Weekly Benefit Cap Proportion | Program | week\_bene\_cap\_prop | option to cap max weekly benefits that can be collected at a proportion of the mean weekly wage |
| Weekly Benefit Minimum |  | week\_bene\_min | min weekly benefits that can be collected |
| FMLA Protection | Behavior | fmla\_protect | Indicates whether or not leaves that are extended in the presence of a program that originally were less than 12 weeks in length are constrained to be no longer than 12 weeks in the presence of the program |
|  |  |  |  |
| Annual Earnings | Program | earnings | earnings in dollars in past 12 months |
| Usual Weeks Worked | Program | weeks | weeks worked in past 12 months |
| Usual Hours Worked | Program | ann\_hours | total number of hours worked in past 12 months |
| Minimum Firm Size | Program | minsize | Number of employees working at their employer |
| Weight Factor | Advanced | weightfactor | Multiply ACS weights by a certain number |
| Random Seed | Advanced | random\_seed | set random seed if user wishes analyses to be replicable |
|  |  | SELFEMP | Whether to include self employed workers in ACS data set |
|  |  |  |  |
|  |  | FEDGOV | Whether to include gov't workers in ACS data set |
|  |  |  |  |
|  |  | STATEGOV | Whether to include gov't workers in ACS data set |
|  |  |  |  |
|  |  | LOCALGOV | Whether to include gov't workers in ACS data set |
|  |  |  |  |
|  |  | formula\_prop\_cuts | Specification for formulaic benefits based on state mean wage |
|  |  | formula\_value\_cuts | Specification for formulaic benefits based on absolute wage values |
|  |  | formula\_bene\_levels | Proportion of pay those under each cut receive |
|  |  | elig\_rule\_logic | Description of the logic used when multiple eligibility criteria are specified. |
|  |  | own\_elig\_adj | user-supplied eligibility adjustment for a given type of leave. |
|  |  | illspouse\_elig\_adj | user-supplied eligibility adjustment for a given type of leave. |
|  |  | illchild\_elig\_adj | user-supplied eligibility adjustment for a given type of leave. |
|  |  | illparent\_elig\_adj | user-supplied eligibility adjustment for a given type of leave. |
|  |  | matdis\_elig\_adj | user-supplied eligibility adjustment for a given type of leave. |
|  |  | bond\_elig\_adj | user-supplied eligibility adjustment for a given type of leave. |

1. Leave benefits: Access, Civilian Workers, National Compensation Survey, March 2016. Retrieved from

   https://www.bls.gov/ncs/ebs/benefits/2016/ownership/civilian/table32a.htm [↑](#footnote-ref-2)
2. US Department of Labor, Women’s Bureau. N.d. Paid Leave Analysis Grant Program. Retrieved from

   https://www.dol.gov/wb/media/paidleavegrants.htm. From 2014 through 2016, over $3 million were awarded through this program to states and municipalities. See the above link for the grantees. [↑](#footnote-ref-3)
3. This model uses data from the latest wave from 2012. DOL conducted two previous waves in 2000 and 1995 as well. [↑](#footnote-ref-4)
4. Previous waves were conducted in 2000 and 1995. The full technical report from the 2012 wave is available here: <https://www.dol.gov/asp/evaluation/fmla/FMLA-2012-Technical-Report.pdf> [↑](#footnote-ref-5)