**Paid Leave Microsimulation Model:**

**R Version Setup and Tutorial**

Draft

October 31, 2019



**Paid Leave Microsimulation Model:**

**R Version Setup and Tutorial**

Draft

October 31, 2019

**Paid Leave Microsimulation Model – R Version**

The purpose of this document is to provide users a guide to setup and basic use of the R version of the IMPAQ Paid Leave Microsimulation Model. For full details on the use, capabilities, and methodology of the model, see the accompanying Paid Leave Microsimulation Model Documentation document.

Currently, the Graphical User Interface (GUI) is functional only with the Python model. For users unfamiliar with programming or technical aspects of simulation modeling, we suggest using the Python GUI as a more accessible way of using the draft version of the model. This tutorial is currently targeted toward those experienced with R programing. In the final draft of the microsimulation model, the GUI will be functional with both Python and R model versions, and the final version of this tutorial will be revised accordingly.

## 1. Setup

The R model was built with R version 3.6.1 and RStudio 1.2.1335. Both are required to run this model.

**To Install R**

1. Open an internet browser and go to www.r-project.org.
2. Click the "download R" link in the middle of the page under "Getting Started."
3. Select a CRAN location (a mirror site) and click the corresponding link.
4. Click on the "Download R for [your operating system]" link at the top of the page.
5. Click on the file containing the latest version of R under "Files."
6. Save the .pkg file, double-click it to open, and follow the installation instructions.
7. Now that R is installed, you need to download and install RStudio.

**To Install RStudio**

1. Go to www.rstudio.com and click on the "Download RStudio" button.
2. Click on "Download RStudio Desktop."
3. Click on the version recommended for your system, save the .dmg file on your computer, double-click it to open, and then drag and drop it to your applications folder.

All model files and folders can be found in the zip file 10292019\_R\_Microsim\_Model.zip. First, unzip this file to a folder in your local drive. Then, open RStudio, and open the project file: “10292019\_microsim\_R.Rproj”. Open Example\_Execution.R, and run it to test that the model successfully executes without errors. The model uses a number of external libraries, listed below in Exhibit 1. These libraries should be automatically installed when the main function, policy\_simulation(), is called. Please note that when first running this, installation may take several minutes.

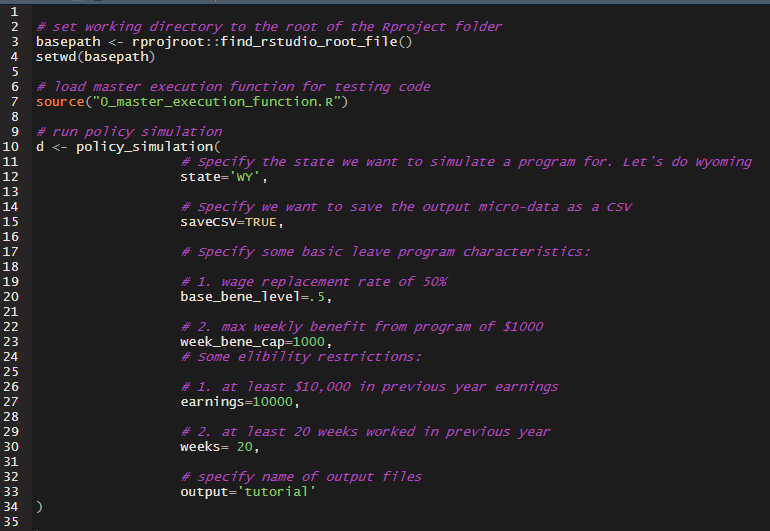
**Exhibit 1. Library Dependencies**

| **Library** | **Version** | **Model’s Use of Library** |
| --- | --- | --- |
| stats | 3.6.1 | General tools for data analysis; used for logit regression |
| rlist | 0.4.6.1 | General tools for list manipulation |
| MASS | 7.3-51.4 | Ordinal regression function (polr) |
| plyr | 1.8.4 | General tools for data manipulation |
| dplyr | 0.8.3 | General tools for data manipulation |
| survey | 3.36 | Tools for using survey weights in logit regression |
| class | 7.3-15 | Alternate K Nearest Neighbor function |
| dummies | 1.5.6 | Function for creating dummy variables |
| varhandle | 2.0.3 | Reformatting factor variables |
| oglmx | 3.0.0.0 | Alternate ordinal regression function |
| foreign | 0.8-71 | Writing data files in additional formats foreign to R |
| ggplot2 | 3.2.1 | Graphics generator |
| reshape2 | 1.4.3 | Flexibly restructure and aggregate data |
| e1071 | 1.7-2 | Naïve Bayes classifier |
| pander | 0.6.3 | Formatting plain text tables |
| ridge | 2.4 | Ridge regression for classifier |
| DMwR | 0.4.1 | SMOTE technique |
| xgboost | 0.90.0.2 | xGboost classifier function |
| bnclassify | 0.4.2 | Naïve Bayes classifier function |
| randomForest | 4.6-14 | Random Forest classifier function |
| magick | 2.1 | Graphics and image processing |
| reader | 1.3.1 | Reading rectangular text data |

## 2. Tutorial

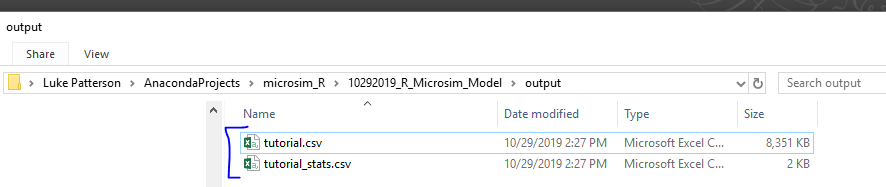
After verifying Example\_Execution.R executes correctly, we are ready to run our own custom simulations. Let’s walk through how to create a custom simulation. policy\_simulation() is the master function that governs the entire simulation. There are 67 different parameters that can be passed as named arguments to policy\_simulation() to customize a simulation run. Most of these parameters have default values if not passed explicitly. The keyword name and specifics of each parameter are described in more detail in “R model – Parameter Dictionary.xlsx”, found in the “Documentation” subfolder. Open up Tutorial\_Execution.R for a simple example and explanation of each of the key parameters to pass.

**Exhibit 2. Tutorial\_Execution.R Screenshot**



Try running this file. If successful, you should see two files appear in the “output” subfolder.

**Exhibit 3. Output Folder Screenshot**



“tutorial.csv” is the output micro-level data set from the simulation. The simulation takes the specified state’s American Community Survey (ACS) file, and adds several columns to represent simulated leave taking, program participation, and other relevant behaviors. You can also analyze the output micro-level data set just like you would an unaltered ACS data set or any other survey data set. For more details on the variables created by the simulation, see the “R Model - Derived Data Dictionary.xlsx” excel file in the “Documentation” subfolder. The current version of the R model can perform some basic summary statistics of key leave taking and program participation automatically. These are saved as a CSV; in this case “tutorial\_stats.csv”.

Now, you should feel more comfortable with how to run the simulation model. Feel free to try running the model with different program characteristics, eligibility restrictions, and other parameters to simulate different scenarios. For more details about all aspects of the model, full documentation to the entire R model is found in the “Documentation” subfolder in the “Microsim R Model Documentation.docx” word document.