
FRED Local

['Epistemix Inc.']

Aug 30, 2021

CONTENTS

1	Chapter 1: FRED Local Installation	3
1.1	Preparing to Install	3
1.2	Installing FRED Local	5
1.3	Verifying the Installation	8
1.4	Next Steps	11
2	Chapter 2: Using FRED Local	13
2.1	Direct Shell Access	13
2.2	Epistemix Tutorial Models	14
2.3	Learning FRED programming	15
2.4	Create a New Model	15
2.5	Getting Help	15

[Return to docs home](#)

This document provides a guide to the installation and use of the FRED Local solution from [Epistemix](#), part of the FRED Modeling Platform™.

With FRED installed locally via the Docker container:

- You may run simulation jobs in a detached fashion or you may access the Docker container interactively via the shell.
- You may use whatever text editors and tools you prefer on your host operating system to create and modify FRED program files.
- You may navigate and process simulation results files via tools on your local operating system (for example, you may open and parse .CSV files via your own Python scripts or in a spreadsheet like Excel).
- If you are familiar with managing and configuring Docker images and containers, you may alter this image to add additional tools (like your preferred text editors or python modules) to enhance your “interactive” sessions.

The FRED Docker image Epistemix provides and supports is based on the Ubuntu Linux LTS (Long Term Support) distribution. This image is known to run properly on **macOS**, **Ubuntu Linux**, and **Microsoft Windows 10** with [WSL 2 \(The Windows Subsystem for Linux\)](#). Other host operating systems may be viable, though they have not been tested and are not recommended or supported by Epistemix at this time.

NOTE: This documentation assumes use of the bash shell for Linux / macOS environments; use of another shell is expected to be compatible but may require slight modifications to the commands shown in this document.

For general background on the FRED platform, please see the [FRED Modeling Platform](#) documentation. If you have questions about the company or are interested in speaking with our sales, recruiting, or other teams, please visit the [Epistemix](#) website.

This document provides installation and usage instructions for FRED Local product.

CHAPTER 1: FRED LOCAL INSTALLATION

This chapter describes how to install the FRED Local product, which is provided as a Docker image, and running an initial FRED model to configure and verify the installation.

Note: This documentation assumes use of the bash shell for Linux / macOS environments; use of another shell is expected to be compatible but may require slight modifications to the commands shown in this document.

1.1 Preparing to Install

Before you install the FRED Local solution, this section provides some background and configuration steps that are important to install successfully.

1.1.1 FRED Local Docker Hardware Requirements

- Adequate physical memory (RAM): FRED requires at least 8 gigabytes of RAM for basic models and small population sizes. For most models 32 gigabytes is recommended, and complex simulations may require much larger memory sizes. At Epistemix, we regularly use containers with 256 GB or more available memory.
- Adequate disk space: FRED's engine is quite reasonable in size, but the synthetic population files can be quite large as can some of the results produced by FRED simulations. At least 50 GB of disk space is required, and we recommend at least 256 GB to accommodate population files, model data, results files, visualizations, and other files.

1.1.2 Installation Prerequisites

These instructions are for installing and running FRED version 7. Before getting started, please ensure you have the following software installed on the computer where you plan to run FRED.

- **Git.** Instructions for installing and setting up `git` can be found here: <https://github.com/git-guides/>.
- **Docker.** Instructions for installing `docker` can be found here: <https://docs.docker.com/get-docker/>.
- **AWS CLI.** Instructions for installing the Amazon Web Services (AWS) Command Line Interface (CLI) can be found here: <https://aws.amazon.com/cli/>

You should also have been granted a username and password (known in AWS parlance as a client ID and client secret) to access the Epistemix Container Registry.

Please contact Epistemix customer service if you encounter any problems.

1.1.3 Create FRED Directories

It is useful to have a local fred directory to serve as the root of your working area for FRED. Under this directory we recommend creating the following directories. It is not technically required to organize your directories in this manner, though this document will assume you have done so in the examples and discussion.

Linux/macOS

Windows

The following illustrates how to create these directories from a shell / command-line:

```
$ cd $HOME
$ mkdir fred
$ mkdir fred/results
$ mkdir fred/models
$ mkdir fred/data
$
```

The following illustrates how to create these directories from a Windows shell / command-line:

```
C:\>md fred
C:\>md fred\results
C:\>md fred\models
C:\>md fred\data
```

1.1.4 Configuring the FRED_HOME environment variable

It is also useful to create a local environment variable FRED_HOME that points to the location of your fred directory. We will assume this environment variable exists in our examples and discussions. It is suggested but not required that you make this environment variable “permanent” via mechanisms appropriate to your operating system (such as modifying your .bashrc file).

Linux/macOS

Windows

```
$ FRED_HOME=$HOME/fred
$ cd $FRED_HOME
$
```

```
C:\>set FRED_HOME=C:\fred
C:\>cd %FRED_HOME%
C:\fred>
```

1.2 Installing FRED Local

Installing the FRED Local product requires that the FRED synthetic population be available as well. We also download some FRED models that we will use to verify the installation. The solution itself is a Docker image.

1.2.1 Install the FRED Population Files

A **Synthetic Population** is a data set that represents each person and household in a given location with geospatial accuracy and contains no personally identifiable information.

Epistemix provides a synthetic population of the United States, though this does not come pre-installed on the Docker image. Because population files are large and you may only need a subset of them for your modeling purposes, these are not included “inside” the FRED Docker image. The full set of population files is distributed separately from the simulation platform and is downloaded as a zip file.

The steps below describe how to download and unzip the entire USA synthetic population. The end result is that the contents of the zip reside within the *data* directory of your *FRED_HOME* directory. The example code uses the command line tool *cURL* though any tool capable of downloading a zip file will work. You download these population files “outside” the Docker image and make them accessible within the container.

Linux/macOS

Windows

```
$ curl -o fred-data.zip https://synthetic-population-epistemix-usa-2010v4-all.s3.
↪amazonaws.com/FRED-data.zip
$ tar xvf fred-data.zip --directory $FRED_HOME/data
$ rm fred-data.zip
```

Warning: Many of FRED’s files are “text” files which in their native “unix” environment leverage a single “line feed” character to denote a line ending. FRED requires the unix line endings. Windows does not come with an unzip CLI utility installed, an open source one known to work is available here: <https://www.7-zip.org/download.html>. Likewise, you may download windows binaries of curl here: <https://curl.se/windows/>. The commands below assume both 7z and curl are in your path. Alternatively you may use the Windows GUI interface to extract the contents of the fred-data.zip file into %FRED_HOME%data.

```
C:\>cd %FRED_HOME%\data
C:\fred\data>curl -o fred-data.zip https://synthetic-population-epistemix-usa-2010v4-all.
↪s3.amazonaws.com/FRED-data.zip
C:\fred\data>7z x fred-data.zip

7-Zip 19.00 (x64) : Copyright (c) 1999-2018 Igor Pavlov : 2019-02-21

Scanning the drive for archives:
1 file, 5900347920 bytes (5628 MiB)

Extracting archive: fred-data.zip
--
Path = fred-data.zip
Type = zip
Physical Size = 5900347920
64-bit = +
```

(continues on next page)

(continued from previous page)

```
Characteristics = Zip64
```

```
<progress info...>
```

```
Everything is Ok
```

```
Folders: 3260
```

```
Files: 10413
```

```
Size:      827431530
```

```
Compressed: 5900347920
```

```
C:\fred\data>del fred-data.zip
```

1.2.2 Download a Sample Model

Epistemix provides a public Tutorials repository with a collection of models that demonstrate essential features of the FRED Modeling Language. The documentation for these models is part of the [Epistemix Documentation](#) and you can download this repository from GitHub.

Linux/macOS

Windows

```
$ cd $FRED_HOME/models
$ git clone https://github.com/Epistemix-com/FRED-tutorials.git
Cloning into 'FRED-tutorials'...
remote: Enumerating objects: 300, done.
remote: Counting objects: 100% (300/300), done.
remote: Compressing objects: 100% (175/175), done.
remote: Total 300 (delta 176), reused 231 (delta 120), pack-reused 0
Receiving objects: 100% (300/300), 2.14 MiB | 6.32 MiB/s, done.
Resolving deltas: 100% (176/176), done.
```

Warning: Many of FRED’s files are “text” files which in their native “unix” environment leverage a single “line feed” character to denote a line ending. Windows (based on it’s DOS legacy) text files end lines with a carriage return as well as a line feed. FRED requires the unix line endings. Git, by default, will convert text files into the format typically desired by the OS it is cloning a repository to. You must override this behavior on your Windows system’s Git installation. To prevent Git from converting text files “globally”, you may run `git config --global core.autocrlf false`. This documentation will repeat this command in various sections purposefully, though it is only required that you do this once. There are other ways to accomplish this that may better suit your needs, though they are beyond the scope of this document.

```
C:\>git config --global core.autocrlf false
C:\>cd %FRED_HOME%\models
C:\fred\models>git clone https://github.com/Epistemix-com/FRED-tutorials.git
Cloning into 'FRED-tutorials'...
remote: Enumerating objects: 300, done.
remote: Counting objects: 100% (300/300), done.
remote: Compressing objects: 100% (175/175), done.
remote: Total 300 (delta 176), reused 231 (delta 120), pack-reused 0
```

(continues on next page)

(continued from previous page)

```
Receiving objects: 100% (300/300), 2.14 MiB | 6.32 MiB/s, done.
Resolving deltas: 100% (176/176), done.
```

1.2.3 Retrieve the FRED Docker Image

Assuming Docker and the AWS CLI are installed, start a terminal or command window and log in to the Epistemix container repository `335566905560.dkr.ecr.us-east-1.amazonaws.com` using the **aws configure** and **aws ecr get-login-password** commands and piping results to **docker login**. You will be prompted for your AWS Access Key ID and AWS Secret Access Key. Replace the placeholders below with the values provided to you by Epistemix.

Linux/macOS

Windows

```
$ aws configure
AWS Access Key ID [None]: YOUR_KEY_ID_PROVIDED_BY_EPISTEMIX
AWS Secret Access Key [None]: YOUR_SECRET_ACCESS_KEY_PROVIDED_BY_EPISTEMIX
Default region name [None]: us-east-1
Default output format [None]:

$ aws ecr get-login-password --region us-east-1 | docker login --username AWS \
--password-stdin 335566905560.dkr.ecr.us-east-1.amazonaws.com
Login Succeeded
```

```
C:\FRED>aws configure
AWS Access Key ID [None]: YOUR_KEY_ID_PROVIDED_BY_EPISTEMIX
AWS Secret Access Key [None]: YOUR_SECRET_ACCESS_KEY_PROVIDED_BY_EPISTEMIX
Default region name [None]: us-east-1
Default output format [None]:

C:\FRED>aws ecr get-login-password --region us-east-1 | docker login ^
More? --username AWS --password-stdin 335566905560.dkr.ecr.us-east-1.amazonaws.com
Login Succeeded
```

Once you are logged in, retrieve the latest FRED image from the repository using **docker pull**. This will make the image available locally. The example output below will likely differ from what you see as it depends on the specific version of the Docker image.

Linux/macOS

Windows

```
$ docker pull 335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-local
Using default tag: latest
latest: Pulling from fred-local
f08d8e2a3ba1: Pull complete
3baa9cb2483b: Pull complete
94e5ff4c0b15: Pull complete
1860925334f9: Pull complete
123eba5a6d56: Pull complete
81d2ef2d3de4: Pull complete
f838d164515f: Pull complete
51261d5cb74a: Pull complete
```

(continues on next page)

(continued from previous page)

```
cfbbb1179df2: Pull complete
588ede05e9b6: Pull complete
1bd07ab3b240: Pull complete
cd198a91dc5a: Pull complete
Digest: sha256:23c959bb1471d8ba734b3e864d5da0de7e5cdd2272c9f0e48d817d7f2a639108
Status: Downloaded image for 335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-
↪local:latest
335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-local:latest
```

```
C:\fred>docker pull 335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-local
Using default tag: latest
latest: Pulling from fred-local
f08d8e2a3ba1: Pull complete
3baa9cb2483b: Pull complete
94e5ff4c0b15: Pull complete
1860925334f9: Pull complete
123eba5a6d56: Pull complete
81d2ef2d3de4: Pull complete
f838d164515f: Pull complete
51261d5cb74a: Pull complete
cfbbb1179df2: Pull complete
588ede05e9b6: Pull complete
1bd07ab3b240: Pull complete
cd198a91dc5a: Pull complete
Digest: sha256:23c959bb1471d8ba734b3e864d5da0de7e5cdd2272c9f0e48d817d7f2a639108
Status: Downloaded image for 335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-
↪local:latest
335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-local:latest
```

1.3 Verifying the Installation

Now that the Docker image is available, we can run the sample model within this image to verify that everything is configured correctly. Running a model will access the synthetic population files.

1.3.1 Run the FRED Local Container

With the image available and directories configured, you can now run the FRED Docker image in a container. The following command establishes a local container with the data, results, and models directories accessible within the model. Note that this uses the environment variable `FRED_HOME` created in a prior section.

Linux/macOS

Windows

```
$ docker run -dit -v "$FRED_HOME"/data:/fred/data \
-v "$FRED_HOME"/results:/fred/results \
-v "$FRED_HOME"/models:/fred/models \
--name fred 335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-local
```

```
C:\fred> docker run -dit -v "%FRED_HOME%\data:/fred/data" ^
More? -v "%FRED_HOME%\results:/fred/results" ^
More? -v "%FRED_HOME%\models:/fred/models" ^
More? --name fred 335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-local
```

Note: Should you need to redo these commands for any reason, you will have to stop your running container (if it's running) and possibly remove / rebuild the docker image. This can be accomplished with **docker stop fred** and **docker rm fred** commands, respectively. Please see the Docker documentation for more details.

1.3.2 Verify the Container Configuration

Once the container has finished start-up activities, verify that the container is properly configured for FRED modeling with the following command.

Linux/macOS

Windows

```
$ docker exec -it fred fred_verify_container_configuration
No issues found! Container configuration verified!
```

```
C:\fred>docker exec -it fred fred_verify_container_configuration
No issues found! Container configuration verified!
```

If a problem is found, an error message will appear instead.

1.3.3 Run a Sample Model

Now that the container is running and verified, you can execute the sample model in the FRED-tutorials directory. We mounted this in the container as the /fred/models directory, and the container's current working directory should be this location as well.

You will see a few models in the directory. While any of these can be used, in our example we will use the model in the simpleflu directory, which simulates a simple Influenza strain in Jefferson Country, Pennsylvania.

By convention, the core FRED program file for a model is called main.fred and a sample run script file to test the model is called METHODS. The following command executes the METHODS script found in the FRED-tutorials/simpleflu directory within the container. A successful output is shown.

Linux/macOS

Windows

```
$ docker exec -it fred /bin/bash -c "cd /fred/models/FRED-tutorials/simpleflu && ./
↳METHODS"

fred_job: starting job simpleflu at Wed Mar 10 15:47:35 GMT 2021

fred_compile -p main.fred
No errors found.
No warnings.
```

(continues on next page)

(continued from previous page)

```
fred_job: running job simpleflu id 55 run 1 ...
fred_job: running job simpleflu id 55 run 2 ...
run_set 0 completed at Wed Mar 10 15:47:37 GMT 2021

fred_job: running job simpleflu id 55 run 3 ...
fred_job: running job simpleflu id 55 run 4 ...
run_set 1 completed at Wed Mar 10 15:47:39 GMT 2021

fred_job: finished job simpleflu 55 at Wed Mar 10 15:47:40 GMT 2021

fred_plot: image_file = daily.pdf
fred_plot: image_file = weekly.pdf
```

```
C:\fred>docker exec -it fred /bin/bash -c "cd /fred/models/FRED-tutorials/
↳simpleflu && ./METHODS"

fred_job: starting job simpleflu at Wed Mar 10 15:47:35 GMT 2021

fred_compile -p main.fred
No errors found.
No warnings.

fred_job: running job simpleflu id 55 run 1 ...
fred_job: running job simpleflu id 55 run 2 ...
run_set 0 completed at Wed Mar 10 15:47:37 GMT 2021

fred_job: running job simpleflu id 55 run 3 ...
fred_job: running job simpleflu id 55 run 4 ...
run_set 1 completed at Wed Mar 10 15:47:39 GMT 2021

fred_job: finished job simpleflu 55 at Wed Mar 10 15:47:40 GMT 2021

fred_plot: image_file = daily.pdf
fred_plot: image_file = weekly.pdf
```

Note: If you get the following error:

```
bash: ./METHODS: /bin/bash^M: bad interpreter: No such file or directory
```

It's almost certainly the case that when you retrieved the tutorial models from GitHub, you did so in a manner that converted the line endings from Unix to DOS / Windows. Please see the previous note on how to prevent this in *Install the FRED Population Files*.

If you run this command more than once, you will see a call to **fred_delete** in the output as well, which deletes the output from the prior run. This output is not shown here.

The METHODS file calls **fred_plot**, which creates an image of the results as a PDF file. This and other tutorials are discussed in the [FRED Language Tutorials](#) available on the [Epistemix Documentation](#) site. Here, the PDF files `daily.pdf` and `weekly.pdf` are created in your model directory. You can display these by opening the PDF files directly or in the viewer of your choice to see the results of this simulation.

Linux/macOS

Windows

```
$ cd $FRED_HOME/models/FRED-tutorials/simpleflu  
$ open daily.pdf  
$ open weekly.pdf
```

```
C:\>cd $FRED_HOME/models/FRED-tutorials/simpleflu  
C:\>fred\models>AcroRd32 daily.pdf  
C:\>fred\models>AcroRd32 weekly.pdf
```

1.4 Next Steps

At this point, your FRED Local Docker installation is complete, verified, and ready! We recommend reading the rest of this guide for some ideas to get you started.

CHAPTER 2: USING FRED LOCAL

Once you have FRED Local installed and have verified the installation as described in Chapter 1, you are ready to use FRED to build and run your own simulations.

This chapter presents some suggestions for using FRED Local and provides some suggestions for updating the `simpleflu` model to become more familiar with the FRED Modeling Language™.

2.1 Direct Shell Access

Running a model as shown in Chapter 1 works well, and you can certainly use this approach to run your own models. Another approach is to run a shell directly in the container and use FRED commands more directly. This section walks through how to run the sample model using this other approach.

You should have a FRED Local container already running called `fred` as described in Chapter 1. Run a shell script with the following commands:

Linux/macOS

Windows

```
$ docker exec -it fred /bin/bash
root@a48b40b88a53:/fred/models# pwd
/fred/models
```

```
C:\fred> docker exec -it fred /bin/bash
root@a48b40b88a53:/fred/models# pwd
/fred/models
```

After this command, you are running directly in the Docker container, so the instructions are identical regardless of the underlying platform. Your actual shell prompt will vary since it includes a portion of the container identifier (the `a48b40b88a53` portion of the prompt in our example).

You can now run FRED Simulation Information Management System (SIMS) commands directly, as described in the [FRED SIMS](#) documentation. For example, you can verify the container configuration as follows:

```
root@a48b40b88a53:/fred/models# fred_verify_container_configuration
Verifying FRED container configuration..
No issues found! Container configuration verified!

root@a48b40b88a53:/fred/models#
```

As another example, you could invoke the `METHODS` script in the `simpleflu` model, or execute the commands from this script directly as shown here.

```

root@a48b40b88a53:/fred/models# cd FRED-tutorials/simpleflu/

root@a48b40b88a53:/fred/models/FRED-tutorials/simpleflu# fred_delete -f -k \
↳ simpleflu
fred_delete: KEY = simpleflu JOB = 129
fred_delete: /fred/results/JOB/129 deleted

root@a48b40b88a53:/fred/models/FRED-tutorials/simpleflu# fred_job -k simpleflu \
↳ \
-p main.fred -n 4 -m 2

fred_job: starting job simpleflu at Mon May  3 14:33:06 GMT 2021

fred_compile -p main.fred
No errors found.
No warnings.

fred_job: running job simpleflu id 130 run 1 ...
fred_job: running job simpleflu id 130 run 2 ...
run_set 0 completed at Mon May  3 14:33:09 GMT 2021

fred_job: running job simpleflu id 130 run 3 ...
fred_job: running job simpleflu id 130 run 4 ...
run_set 1 completed at Mon May  3 14:33:12 GMT 2021

fred_job: finished job simpleflu 130 at Mon May  3 14:33:13 GMT 2021

root@a48b40b88a53:/fred/models/FRED-tutorials/simpleflu# fred_plot -o weekly \
-k simpleflu -v INFLUENZA.newExposed -b -t "Simple Flu Model" -l "Weekly \
↳ Incidence" \
-d 0 -w --clean
fred_plot: image_file = weekly.pdf

root@a48b40b88a53:/fred/models/FRED-tutorials/simpleflu#

```

Using the shell directly in this manner may be easier than invoking commands externally. For more information on the FRED command-line tools, see the document [FRED Simulation Information Management Guide](#) available on the [Epistemix Documentation](#) site.

2.2 Epistemix Tutorial Models

The other tutorial models downloaded in Chapter 1 as part of the FRED-tutorials repository provide a good overview of some key features of FRED. We recommend reading through the Tutorials documentation found at the [FRED Language Tutorials](#) link and running one or more simulations for each of the models.

Each model has a METHODS script, just list `simpleflu`, so you can run this script directly. A more instructive approach might be to execute the FRED SIMS commands directly as we show in the prior example. The [FRED SIMS](#) documentation includes reference pages for each command, if you wish to explore these commands further.

2.3 Learning FRED programming

It is beneficial to modify the `simpleflu` and other tutorial programs as a way to experience the keywords and syntax of FRED programs. In the `simpleflu` model directory, for example, here are some changes you could make after reading through the other tutorials. Copy the code to a new directory and run your model after each suggestion to see the impact on the results.

Suggested `simpleflu` Changes for Practice

1. Modify the `locations` setting in the `main.fred` file to use your local county. Note that if your county population is more than a few hundred thousand, you may need to increase the memory for your docker container. You can do this in the Preferences settings in Docker Desktop, under the Resources options. For the `simpleflu` model, we would recommend 8 GB of memory per 1 million residents of an area. FRED does use a lot of memory, and if the memory available is exceeded, then the program will be killed by the operating system.
2. Add a `variables.fred` file to define a `symptomatic_transmissibility` variable set to 1.0, and replace the `INFLUENZA.trans` assignments to use this variable in both the `InfectiousSymptomatic` and `InfectiousAsymptomatic` states. Don't forget to include the new `variables.fred` file in your `main.fred` file.
3. Add a `parameters.fred` file to the program that overrides the value of `symptomatic_transmissibility` to be 0.75. Generate the `weekly.pdf` plots for both transmissibility values to compare when the peak number of exposures occurs within the population.
4. Modify the `INFLUENZA` state so that an agent has a 5% chance of dying when they are symptomatic and a 1% chance of dying when they are asymptomatic. You will need to use the `die()` function for this scenario, which causes the agent to die.

2.4 Create a New Model

At some point you will want to create your own models with FRED. We recommend reading through the available documentation before attempting this, of course. You should also consider creating a state diagram of your conditions as a way to think through the behavior and transitions for the agents in your model. FRED's declarative programming style lends itself to such diagrams, and the modelers at Epistemix often use them.

As an example, here is a state transition diagram of the `INFLUENZA` condition in the simple flu model, in a form you might create before writing the condition.

As an alternative, here is a diagram generated from the `simpleflu` code directly. This has the advantage of showing the states and FRED code used in the program.

2.5 Getting Help

If you need assistance or have questions using FRED Local or the FRED Modeling Language, contact our Customer Service team at support@epistemix.com.

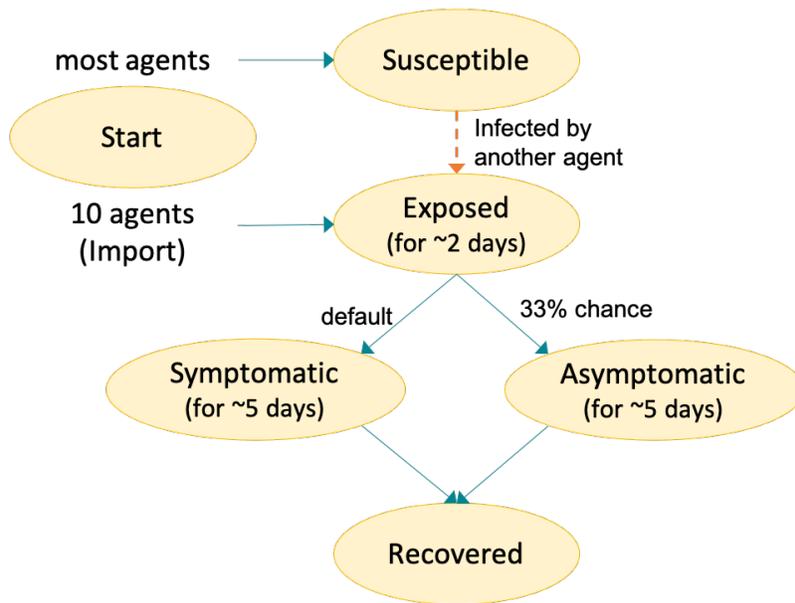


Fig. 1: A state transition diagram for the simple flu INFLUENZA condition.

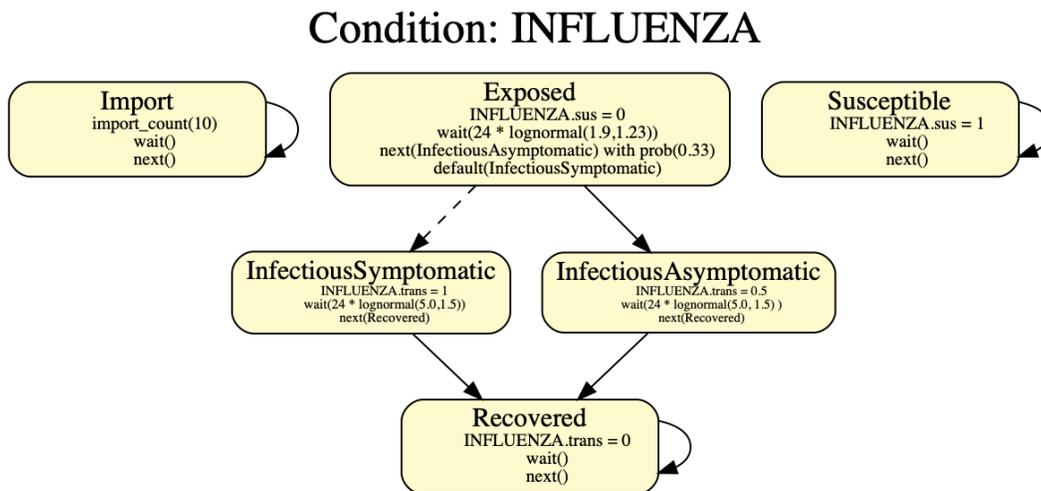


Fig. 2: A generated state transition diagram for the INFLUENZA condition. This shows the actual state names used in the FRED program, and the code for each state. Note that the implicit transition from Susceptible to Exposed is not reflected here.