

FRED Exercises

12 June 2012

The following exercises assume that FRED has been installed and compiled properly. See the installation guide for help on installing FRED.

When running FRED from the command line, it is recommended that you use the scripts provided in the directory \$FRED_HOME/bin. These scripts will store all metadata associated with each job in the FRED RESULTS database. This database is currently a directory-tree stored in the user's directory under \$FRED_HOME/RESULTS.

It is recommended that you run FRED jobs in a working directory, for example \$FRED_HOME/work.

Exercise 1. Running a baseline epidemic for Allegheny County.

Here is a sample parameter file:

```
% cat params.baseline
days = 100
```

This will run FRED for 100 simulated days using the default influenza parameters for Allegheny County. Any run-time parameter not given in the parameter file will be given default values specified in file \$FRED_HOME/input_files/**params.default**.

To execute fred, use the command

```
% fred_job -p params.baseline -k baseline &
```

The -p argument is the parameter file and the -k argument gives a user-defined name for the job. The job will run in the background.

You can check the status of the job using the fred_status command:

```
% fred_status -k baseline
RUNNING-00 Tue Nov 29 21:50:15 2011
```

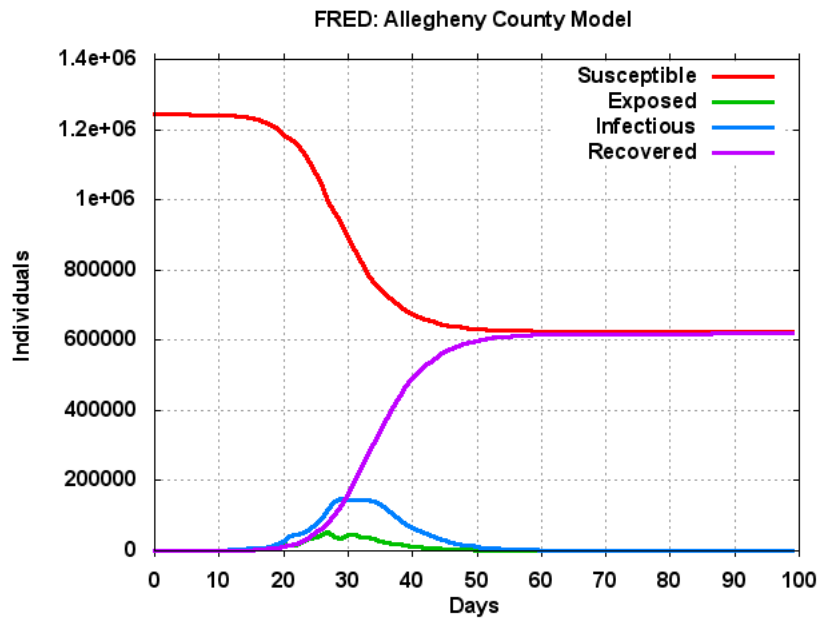
```
% fred_status -k baseline
FINISHED Tue Nov 29 21:52:10 2011
```

The command fred_jobs shows the status of all jobs:

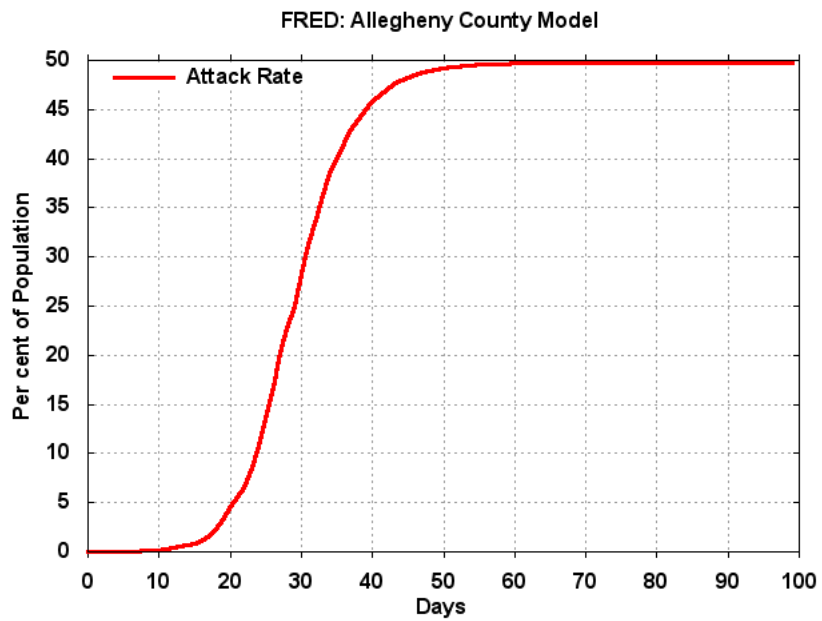
```
% fred_jobs
KEY = baseline      JOB =    1 STATUS = FINISHED Thu Aug  4 14:28:23 2011
KEY = stay-0.2      JOB =    5 STATUS = FINISHED Wed Sep 21 14:52:57 2011
KEY = stay-0.25     JOB =    8 STATUS = FINISHED Wed Sep 21 15:22:14 2011
KEY = stay-0.3      JOB =    9 STATUS = FINISHED Wed Sep 21 15:46:36 2011
KEY = stay-0.35     JOB =   10 STATUS = FINISHED Wed Sep 21 16:10:50 2011
```

When the job is finished, you can display results. The variable(s) to be display are listed after the -v flag.

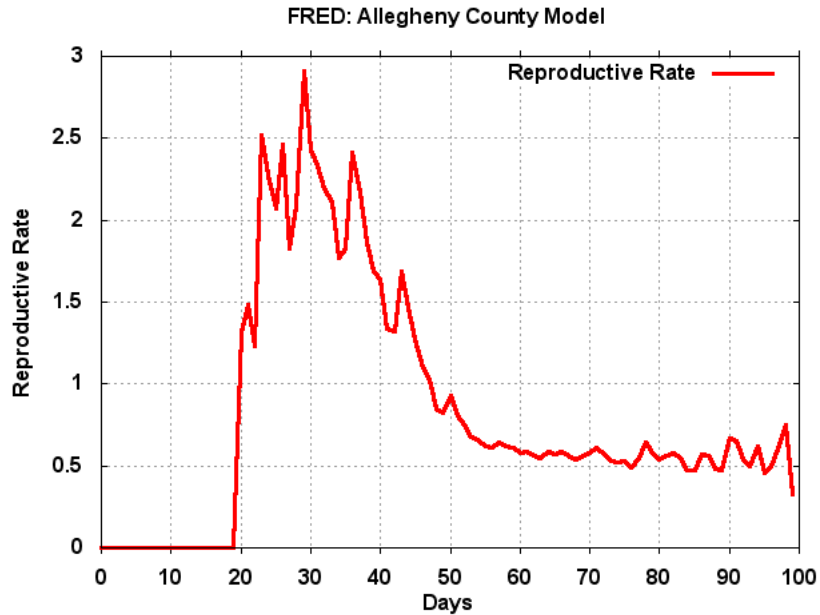
```
% fred_display_plot -k baseline -v S -v E -v I -v R
```



```
% fred_display_plot -k baseline -v AR
```



```
% fred_display_plot -k baseline -v RR
```



You can also get the data associated with a plot:

```
% fred_plot_data -k baseline -v AR
```

```
0 0.01 0
1 0.01 0
2 0.01 0
3 0.02 0
4 0.03 0
5 0.03 0
.
.
.
95 49.75 0
96 49.75 0
97 49.75 0
98 49.75 0
99 49.75 0
```

Exercise 2: Varying the transmission parameter

The transmissibility of disease 0 (the default disease) is indicated by the parameter `trans[0]`, and has a default value of 1 for influenza (this is calibrated to a serological attack rate of 50%). We will define two new parameter files:

```
% cat params.weak
days = 100
trans[0] = 0.8
```

```
% cat params.strong
days = 100
trans[0] = 1.2
```

Now run two FRED jobs:

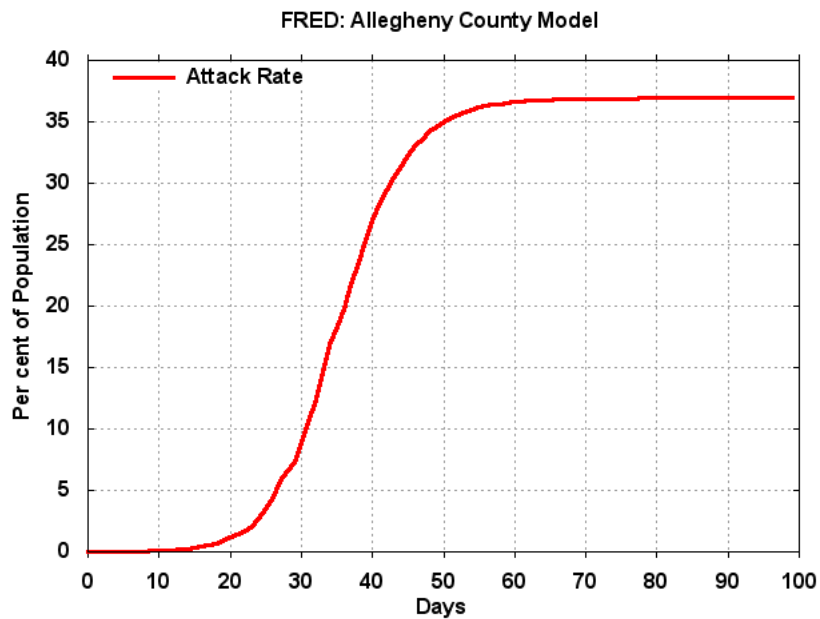
```
% fred_job -p params.weak -k weak
```

```
% fred_job -p params.strong -k strong
```

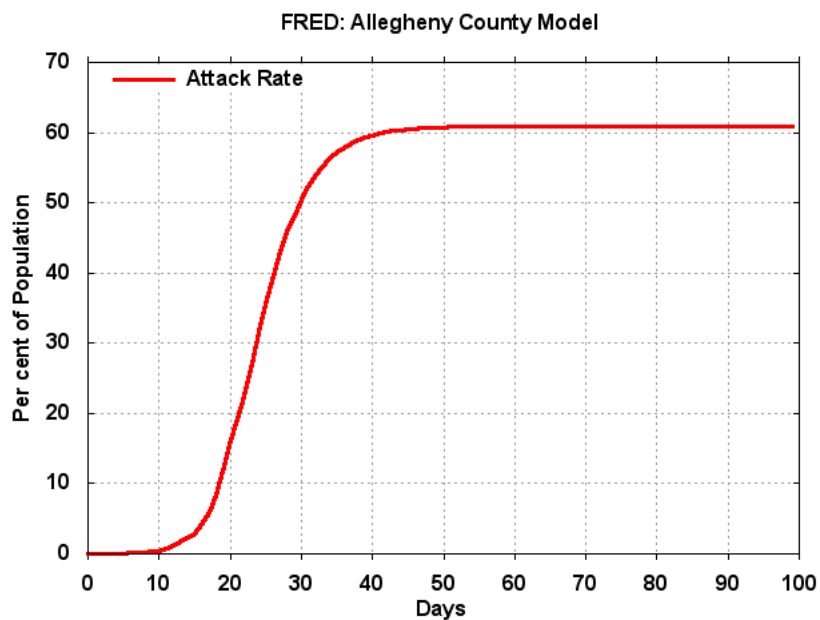
The jobs can be run in parallel if you have sufficient memory (approximately 2GB per job for Allegheny County).

When the jobs are finished you can display the results:

```
% fred_display_plot -k weak -v AR
```

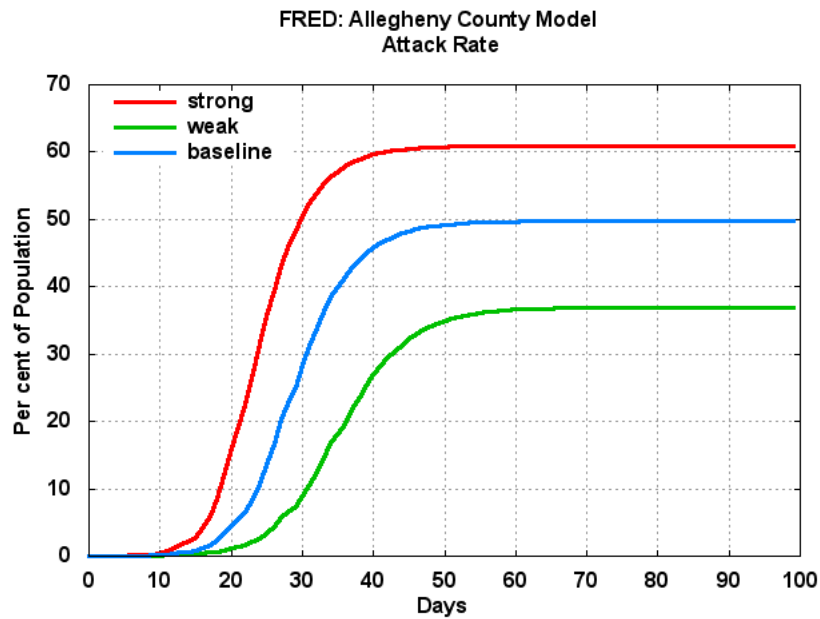


```
% fred_display_plot -k strong -v AR
```

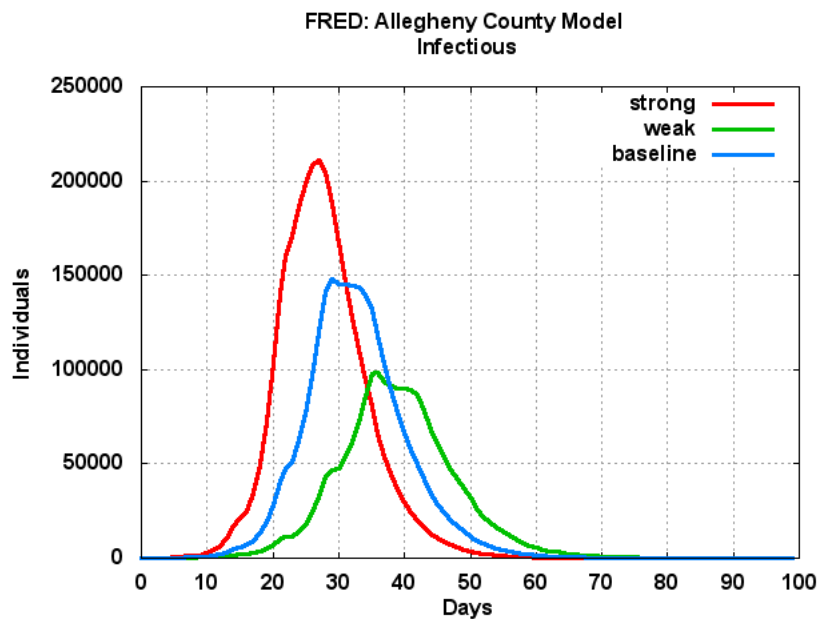


You can display multiple plots on the same graph:

```
% fred_display_plots -v AR -k strong -k weak -k baseline
```



```
% fred_display_plots -v I -k strong -k weak -k baseline
```



Exercise 3: Varying a behavioral parameter

As an example of varying the behavior of FRED agent, we will change the probability that agents stay home when sick. The default is that 50% of the agents stay home when sick. We will define two new parameter files:

```
% cat params.stay-40
days = 100
stay_home_when_sick_enabled = 1
stay_home_when_sick_strategy_distribution = 7 60 40 0 0 0 0 0

% cat params.stay-60
days = 100
stay_home_when_sick_enabled = 1
stay_home_when_sick_strategy_distribution = 7 40 60 0 0 0 0 0
```

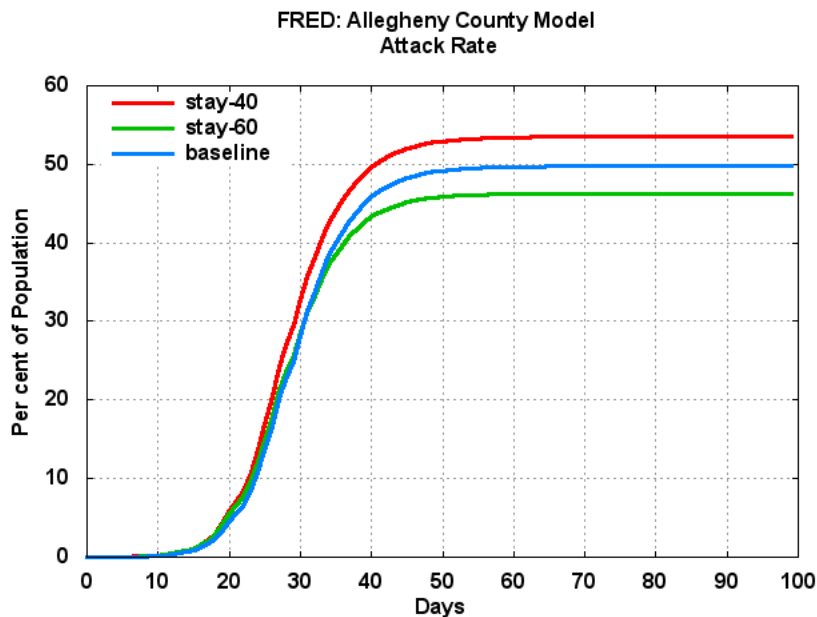
The first file says that 40% of the agents will always stay home when they have symptoms. The second file says that 60% of the agents will stay home when sick. See the FRED Users Guide for an explanation of the parameter format for the behavior strategy distributions.

Now run two more FRED jobs:

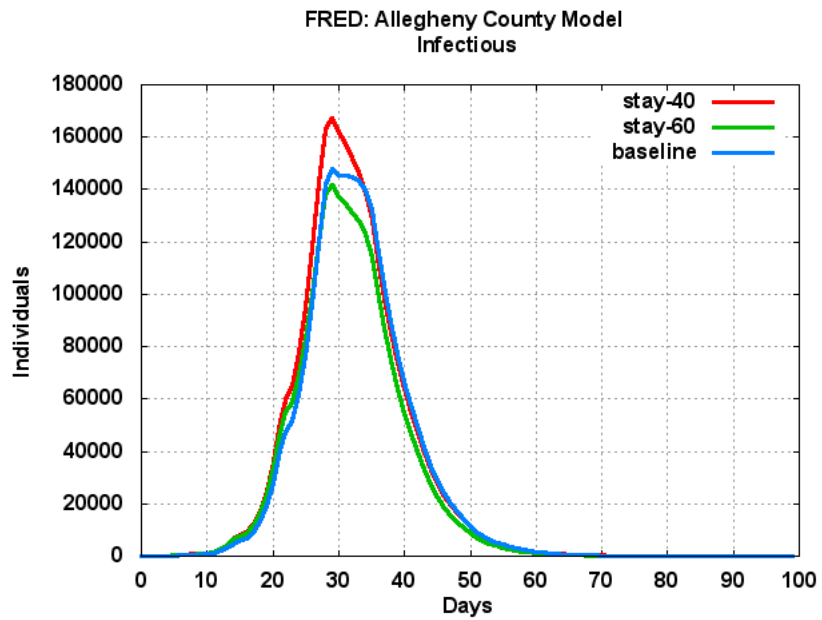
```
% fred_job -p params.stay-40 -k stay-40
% fred_job -p params.stay-60 -k stay-60
```

When the jobs are finished you can display the results

```
% fred_display_plots -v AR -k stay-40 -k stay-60 -k baseline
```



```
% fred_display_plots -v I -k stay-40 -k stay-60 -k baseline
```



Further Exercises:

Explore combinations of FRED parameters by adding additional lines to the parameter files. See the FRED Users Guide for discussions of FRED parameters.