SynthPops

Institute for Disease Modeling

Sep 11, 2020

Contents

SynthPops is used construct synthetic networks of people that satisfy statistical properties of real-world populations (such as the age distribution, household size, etc.). SynthPops can create generic populations with different network characteristics, as well as synthetic populations that interact in different layers of a multilayer contact network. These synthetic populations can then be used with agent-based models like COVID-19 Agent-based Simulator (Covasim) to simulate epidemics. SynthPops is available on GitHub. For more information on Covasim see Covasim on GitHub.

CHAPTER 1

Installation

Follow the instructions below to install SynthPops.

1.1 Requirements

Python 3.6 64-bit. (Note: Python 2 is not supported.)

We also recommend, but do not require, using Python virtual environments. For more information, see documentation for venv or Anaconda.

1.2 Installation

Complete the following steps to install SynthPops:

1. Fork and clone the SynthPops GitHub repository.
2. Open a command prompt and navigate to the SynthPops directory.
3. Run the following script:

```
python setup.py develop
```

Note: while *synthpops* can also be installed via pypi, this method does not currently include the data files which are required to function, and thus is not recommended.

1.3 Quick start guide

The following code creates a synthetic population for Seattle, Washington:

```
import synthpops as sp

sp.validate()

datadir = sp.datadir # this should be where your demographics data folder resides

location = 'seattle_metro'
state_location = 'Washington'
country_location = 'usa'
sheet_name = 'United States of America'
level = 'county'

npop = 10000 # how many people in your population
sp.generate_synthetic_population(npop,datadir,location=location,
                                 state_location=state_location,country_
↪location=country_location,
                                 sheet_name=sheet_name,level=level)
```

CHAPTER 2

SynthPops overview

Fundamentally, the population network can be considered a multilayer network with the following qualities:

- Nodes are people, with attributes like age.
- Edges represent interactions between people, with attributes like the setting in which the interactions take place (for example, household, school, or work). The relationship between the interaction setting and properties governing disease transmission, such as frequency of contact and risk associated with each contact, is mapped separately by Covasim or other *agent-based model*. SynthPops reports whether the edge exists or not.

If you are using SynthPops with Covasim, note that the relevant value in Covasim is the parameter **beta**, which captures the probability of transmission via a given edge per *time step*. The value of this parameter captures both number of effective contacts for disease transmission and transmission probability per contact.

The generated network is a multilayer network in the sense that it is possible for people to be connected by multiple edges each in different layers of the network. The layers are referred to as *contact layers*. For example, the *workplace contact layer* is a representation of all of the pairwise connections between people at work, and the *household contact layer* represents the pairwise connections between household members. Typically these networks are clustered; in other words, everyone within a household interacts with each other, but not with other households. However, they may interact with members of other households via their school or workplace. Some level of community contacts outside of these networks can be configured using Covasim or other model being used with SynthPops.

SynthPops functions in two stages:

1. Generate people living in households, and then assign individuals to workplaces and schools. Save the output to a cache file on disk. Implemented in `generate_synthetic_population()`.
2. Load the cached file and produce a dictionary that can be used by Covasim. Implemented in `make_population()`. Covasim assigns community contacts at random on a daily basis to reflect the random and stochastic aspect of contacts in many public spaces, such as shopping centers, parks, and community centers.

CHAPTER 3

SynthPops algorithm

This topic describes the algorithm used by SynthPops to generate the connections between people in each of the *contact layers* for a given location in the real world. The fundamental algorithm is the same for homes, schools, and workplaces, but with some variations for each.

The method draws upon the following previously published models to infer high-resolution age-specific contact patterns in different physical settings and locations:

- Mossong et al. 2008
- Fumanelli et al. 2012
- Prem et al. 2017
- Mistry et al. 2020

The general idea is to use age-specific contact matrices that describe age mixing patterns for a specific population. By default, SynthPops uses Prem et al.'s (2017) matrices, which project inferred age mixing patterns from the POLYMOD study (Mossong et al. 2008) in Europe to other countries. However, user-specified contact matrices can also be implemented for customizing age mixing patterns for the household, school, and workplace settings (see the social contact data on Zenodo for other empirical contact matrices from survey studies).

The matrices represent the average number of contacts between people for different age bins (the default matrices use 5-year age bins). For example, a household of two individuals is relatively unlikely to consist of a 25-year-old and a 15-year-old, so for the 25-29 year age bin in the household layer, there are a low number of expected contacts with the 15-19 year age bin (c.f., Fig. 2c in Prem et al.).

CHAPTER 4

Using SynthPops

The overall SynthPops workflow is contained in `generate_synthetic_population()` and is described below. The population is generated through households, not a pool of people.

You can provide required data to SynthPops in a variety of formats including .csv, .txt, or Microsoft Excel (.xlsx.

1. Instantiate a collection of households with sizes drawn from census data. Populations cannot be created outside of the *household contact layer*.
2. For each household, sample the age of a "reference" person from data that maps household size to a reference person in those households. The reference person may be referred to as the head of the household, a parent in the household, or some other definition specific to the data being used. If no data mapping household size to ages of reference individuals are available, then the age of the reference person is sampled from the age distribution of adults for the location.
3. The age bin of the reference people identifies the row of the contact matrix for that location. The remaining household members are then selected by sampling an age for the distribution of contacts for the reference person's age (in other words, normalizing the values of the row and sampling for a column) and assigning someone with that age to the household.
4. As households are generated, individuals are given IDs.
5. After households are constructed, students are chosen according to enrollment data by age to generate the *school contact layer*.
6. Students are assigned to schools using a similar method as above, where we select the age of a reference person and then select their contacts in school from an age-specific contact matrix for the school setting and data on school sizes.
7. With all students assigned to schools, teachers are selected from the labor force according to employment data.
8. The rest of the labor force are assigned to workplaces in the *workplace contact layer* by selecting a reference person and their contacts using an age-specific contact matrix and data on workplace sizes.

4.1 Examples

Examples live in the *examples* folder. These can be run as follows:

- `python examples/make_generic_contacts.py`

  Creates a dictionary of individuals, each of whom are represented by another dictionary with their contacts contained in the `contacts` key. Contacts are selected at random with degree distribution following the Erdos-Renyi graph model.

- `python examples/generate_contact_network_with_microstructure.py`

  Creates and saves to file households, schools, and workplaces of individuals with unique IDs, and a table mapping IDs to ages. Two versions of each contact layer (households, schools, or workplaces) are saved; one with the unique IDs of each individual in each group (a single household, school or workplace), and one with their ages (for easy viewing of the age mixing patterns created).

- `python examples/load_contacts_and_show_some_layers.py`

  Loads a multilayer contact network made of three layers and shows the age and ages of contacts for the first 20 people.

In the *tests* folder, you can view the following to see examples of additional functionality.

- `test_synthpop.py`

  Reads in demographic data and generates populations matching those demographics.

- `test_contacts.py`

  Generates random contact networks with individuals matching demographic data or reads in synthetic contact networks with three layers (households, schools, and workplaces).

- `test_contact_network_generation.py`

  Generates synthetic contact networks in households, schools, and workplaces with Seattle Metro data (and writes to file).

The other topics in this section walk through the specific data sources and details about the settings for each of the *contact layers*.

4.1.1 Household contact layer

The *household contact layer* represents the pairwise connections between household members. The population is generated within this contact layer, not as a separate pool of people.

As locations, households are special in the following ways:

- Unlike schools and workplaces, everyone must be assigned to a household.
- The size of the household is important (for example, a 2-person household looks very different in comparison to a 5- or 6-person household) and some households only have 1 person.
- The reference person/head of the household can be well-defined by data.

Data needed

The following data sets are required for households:

1. **Age bracket distribution** specifying the distribution of people in age bins for the location. For example:

```
age_bracket , percent
0_4         , 0.0594714358950416
5_9         , 0.06031137308234759
10_14       , 0.05338015778985113
15_19       , 0.054500690394160285
20_24       , 0.06161403846144956
25_29       , 0.08899312471888453
30_34       , 0.0883533486774803
35_39       , 0.07780767611060545
40_44       , 0.07099017823587304
45_49       , 0.06996903280562596
50_54       , 0.06655242534751997
55_59       , 0.06350008343899961
60_64       , 0.05761405140489549
65_69       , 0.04487122889235999
70_74       , 0.030964420778483555
75_100       , 0.05110673396642193
```

2. **Age distribution of the reference person for each household size**

   The distribution is what matters, so it doesn't matter if absolute counts are available or not, each *row* is normalized. If this is not available, default to sampling the age of the reference individual from the age distribution for adults:

```
family_size , 18-20 , 20-24 , 25-29 , 30-34 , 35-39 , 40-44 , 45-49 , 50-54 , 55-
↪64 , 65-74 , 75-99
2           , 163   , 999   , 2316  , 2230  , 1880  , 1856  , 2390  , 3118  ,␣
↪9528  , 9345  , 5584
3           , 115   , 757   , 1545  , 1907  , 2066  , 1811  , 2028  , 2175  ,␣
↪3311  , 1587  , 588
4           , 135   , 442   , 1029  , 1951  , 2670  , 2547  , 2368  , 1695  ,␣
↪1763  , 520   , 221
5           , 61    , 172   , 394   , 905   , 1429  , 1232  , 969   , 683   , 623␣
↪  , 235   , 94
6           , 25    , 81    , 153   , 352   , 511   , 459   , 372   , 280   , 280␣
↪  , 113   , 49
7           , 24    , 33    , 63    , 144   , 279   , 242   , 219   , 115   , 157␣
↪  , 80    , 16
```

3. **Distribution of household sizes**:

```
household_size , percent
1              , 0.2781590909877753
2              , 0.3443313103056699
3              , 0.15759535523004006
4              , 0.13654311541644018
5              , 0.050887858718118274
6              , 0.019738368167953997
7              , 0.012744901174002305
```

4. **Household contact matrix** specifying the number/weight of contacts by age bin:

```
        0-10        , 10-20       , 20-30
0-10    0.659867911 , 0.503965302 , 0.214772978
10-20   0.314776879 , 0.895460015 , 0.412465791
20-30   0.132821425 , 0.405073038 , 1.433888594
```

   By default, SynthPops uses matrices from a study (Prem et al. 2017) that projected inferred age mixing patterns

from the POLYMOD study (Mossong et al. 2008) in Europe to other countries. SynthPops can take in user-specified contact matrices if other age mixing patterns are available for the household, school, and workplace settings (see the social contact data on Zenodo for other empirical contact matrices from survey studies).

In theory, the household contact matrix varies with household size, but in general data at that resolution is unavailable.

Workflow

Use these SynthPops functions to instantiate households as follows:

1. Call `generate_synthetic_population()` and provide the binned age bracket distribution data described above. This wrapper function calls the following functions:
    1. From the binned age distribution, `get_age_n()` creates samples of ages from the binned distribution, and then normalizes to create a single-year distribution. This distribution can therefore be gathered using whatever age bins are present in any given dataset.
    2. `generate_household_sizes_from_fixed_pop_size()` generates empty households with known size based on the distribution of household sizes.
    3. `generate_all_households()` contains the core implementation and constructs households with individuals of different ages living together. It takes in the remaining data sources above, and then does the following:
        - Calls `generate_living_alone()` to populate households with 1 person (either from data on those living alone or, if unavailable, from the adult age distribution).
        - Calls `generate_larger_households()` repeatedly with with different household sizes to populate those households, first sampling the age of a reference person and then their household contacts as outlined above.

4.1.2 School contact layer

The *school contact layer* represents all of the pairwise connections between people in schools, including both students and teachers. Schools are special in that:

- Enrollment rates by age determine the probability of individual being a student given their age.
- Staff members such as teachers are chosen from individuals determined to be in the adult labor force.
- The current methods in SynthPops treat student and worker status as mutually exclusive. Many young adults may be both students and workers, part time or full time in either status. The ability to select individuals to participate in both activities will be introduced in a later develop of the model.

Data needed

The following data is required for schools:

1. **School size distribution**:

```
school_size , percent
0-50        , 0.2
51-100      , 0.1
101-300     , 0.3
```

2. **Enrollment by age** specifying the percentage of people of each age attending school. See `get_school_enrollment_rates()`, but note that this mainly implements parsing a Seattle-specific data file to produce the following data structure, which could equivalently be read directly from a file:

```
age , percent
0   , 0
1   , 0
2   , 0
3   , 0.529
4   , 0.529
5   , 0.95
6   , 0.95
7   , 0.95
8   , 0.95
9   , 0.95
10  , 0.987
11  , 0.987
12  , 0.987
13  , 0.987
```

3. **School contact matrix** specifying the number/weight of contacts by age bin. This is similar to the household contact matrix. For example:

```
        0-10        , 10-20       , 20-30
0-10    0.659867911 , 0.503965302 , 0.214772978
10-20   0.314776879 , 0.895460015 , 0.412465791
20-30   0.132821425 , 0.405073038 , 1.433888594
```

4. **Employment rates by age**, which is used when determining who is in the labor force, and thus which adults are available to be chosen as teachers:

```
Age , Percent
16  , 0.496
17  , 0.496
18  , 0.496
19  , 0.496
20  , 0.838
21  , 0.838
22  , 0.838
```

5. **Student teacher ratio**, which is the average ratio for the location. Methods to use a distribution or vary the ratio for different types of schools may come in later developments of the model:

```
student_teacher_ratio=30
```

Typically, contact matrices describing age-specific mixing patterns in schools include the interactions between students and their teachers. These patterns describe multiple types of schools, from possibly preschools to universities.

Workflow

Use these SynthPops functions to implement the school contact layer as follows:

1. `get_uids_in_school()` uses the enrollment rates to determine which people attend school. This then provides the number of students needing to be assigned to schools.
2. `generate_school_sizes()` generates schools according to the school size distribution until there are enough places for every student to be assigned a school.

3. `send_students_to_school()` assigns specific students to specific schools.
    - This function is similar to households in that a reference student is selected, and then the contact matrix is used to fill the remaining spots in the school.
    - Some particulars in this function deal with ensuring a teacher/adult is less likely to be selected as a reference person, and restricting the age range of sampled people relative to the reference person so that a primary school age reference person will result in the rest of the school being populated with other primary school age children
4. `get_uids_potential_workers()` selects teachers by first getting a pool of working age people that are not students.
5. `get_workers_by_age_to_assign()` further filters this population by employment rates resulting in a collection of people that need to be assigned workplaces.
6. In `assign_teachers_to_work()`, for each school, work out how many teachers are needed according to the number of students and the student-teacher ratio, and sample those teachers from the pool of adult workers. A minimum and maximum age for teachers can be provided to select teachers from a specified range of ages (this can be used to account for the additional years of education needed to become a teacher in many places).

4.1.3 Workplace contact layer

The *workplace contact layer* represents all of the pairwise connections between people in workplaces, except for teachers working in schools. After some workers are assigned to the *school contact layer* as teachers, all remaining workers are assigned to workplaces. Workplaces are special in that there is little/no age structure so workers of all ages may be present in every workplace.

Again, note that work and school are currently exclusive, because the people attending schools are removed from the list of eligible workers. This doesn't necessarily need to be the case though. In fact, we know that in any countries and cultures around the world, people take on multiple roles as both students and workers, either part-time or full-time in one or both activities.

Data required

The following data are required for generating the workplace contact layer:

1. **Workplace size distribution** - again, this gets normalized so can be specified as absolute counts or as normalized values:

```
work_size_bracket , size_count
1-4               , 2947
5-9               , 992
10-19             , 639
20-49             , 430
50-99             , 140
100-249           , 83
250-499           , 26
500-999           , 13
1000-1999         , 12
```

2. **Work contact matrix** specifying the number/weight of contacts by age bin. This is similar to the household contact matrix. For example:

```
        20-30       , 30-40       , 40-50
20-30   0.659867911 , 0.503965302 , 0.214772978
```

(continues on next page)

(continued from previous page)

```
30-40   0.314776879 , 0.895460015 , 0.412465791
40-50   0.132821425 , 0.405073038 , 1.433888594
```

Workflow

1. `generate_workplace_sizes()` generates workplace sizes according to the workplace size distribution until the number of workers is reached.
2. `assign_rest_of_workers()` populates workplaces just like for households and schools: randomly selecting the age of a reference person, and then sampling the rest of the workplace using the contact matrix.

CHAPTER 5

synthpops package

CHAPTER 6

Glossary

contact layers Each of the layers of the population network that is a representation of all of the pairwise connections between people in a given location, such as school, work, or households.

node In network theory, the discrete object being represented. In SynthPops, nodes represent people and can have attributes like age assigned.

edge In network theory, the interactions between discrete objects. In SynthPops, edges represent interactions between people, with attributes like the setting in which the interactions take place (for example, household, school, or work). The relationship between the interaction setting and properties governing disease transmission, such as frequency of contact and risk associated with each contact, is mapped separately by Covasim or other *agent-based model*. SynthPops reports whether the edge exists or not.

agent-based model A type of simulation that models the actions and interactions of autonomous agents (both individual and collective entities such as organizations or groups).

time step A discrete number of hours or days in which the "simulation states" of all "simulation objects" (interventions, infections, immune systems, or individuals) are updated in a simulation. Each time step will complete processing before launching the next one. For example, a time step would process the migration data for populations moving between nodes via rail, airline, and road. The migration of individuals between nodes is the last step of the time step after updating states.

household contact layer The layer in the population network that represents all of the pairwise connections between people in households. All people must be part of the household contact layer, though some households may consist of a single person.

school contact layer The layer in the population network that represents all of the pairwise connections between people in schools. This includes both students and teachers. The school and workplace contact layers are mutually exclusive, someone cannot be both a student and a worker.

workplace contact layer The layer in the population network that represents all of the pairwise connections between people in workplaces excluding teachers in schools. The school and workplace contact layers are mutually exclusive, someone cannot be both a student and a worker.

Index