Modeling Data and Parameters

Dawn Nekorchuk, Michael Wimberly, and EPIDEMIA Team Members
Department of Geography and Environmental Sustainability, University of Oklahoma
dawn.nekorchuk@ou.edu; mcwimberly@ou.edu

Updated May 09, 2019

Contents

1 Input Data Formats, Model Specifications, and Event Detection Parameters

1.1 Data & Data Formats

The epidemiar modeling and code requires 3 main sets of data:

- epidemiological data,
- daily environmental data, and
- historical environmental reference data,

plus several information/reference/specification inputs.

1.1.1 Epidemiology Data, `epi_data`

For the epidemiology data, you will need weekly case counts of the disease/illness per the geographic unit (group) with population values (to calculate incidence).

When calling the epidemiar function:

- `epi_data`: Data table/tibble of epidemiological data with case numbers per week, with date field labeled as `obs_date`. The date should be the *last* day of the epidemiological week. Must contain columns for `{casefield}`, `{populationfield}`, and `{groupfield}`. It may contain other variables/columns, but these are ignored.
- `casefield`: Give the field name for the case counts.
- `populationfield`: Give the population field to give population numbers over time. It is used to calculated incidence, and also optionally used in Farrington method for populationOffset.
- `groupfield`: Give the field name for districts or area divisions of epidemiological AND environmental data. If there are no groupings (all one area), user should give a field with the same value throughout the entire datasets.
- `inc_per`: At what rate should incidence be calculated for? Default is "1000", meaning x cases per 1000 population.
- `week_type`: For the `obs_date` in `epi_data`, you need to specify if you are using "CDC" epiweeks, or ISO-8601 ("ISO") standard weeks of the year (what WHO uses), the default assumption is ISO. The date should be the *last* day of the epidemiological week.

1.1.1.1 Missing Data

There should be a line for each week and geographic grouping, even for missing data (i.e. explicit missing data). Any missing data will be filled in by linear interpolation inside of the epidemiar modeling functions.

1.1.2 Environmental Data, `env_data`

For the environmental data, daily data is expected for each environmental variable for each geographic unit. Based on the lag length chosen, you must have at least that number of days *before* the first epidemiology data date.

When calling the epidemiar function:

- `env_data`: Data table/tibble of environmental data values for each geographic grouping, with date field labeled as "obs_date".
- `groupfield`: Give the field name for districts or area divisions of epidemiological AND environmental data. If there are no groupings (all one area), user should provide a field with the same value throughout the entire datasets.
- `obsfield`: Give the field name of the environmental data observation types.
- `valuefield`: Give the field name of the value of the environmental data observations.

1.1.2.1 Non-daily or Missing Data

If you do not have daily data (e.g. weekly, or irregular data), or have implicit missing data, you can use the `data_to_daily()` function to add any missing rows. This function will also use linear interpolation to fill in values if possible (default 'interpolate = TRUE'). It is not recommended if you have a lot of missing/non-daily data. It will group on every field in the dataset that is not `obs_date`, or the user-given `{valuefield}`. Note: this will not fill out ragged data (different end dates of environmental variable data), but that will be handled inside of epidemiar.

1.1.3 Environmental Reference / Weekly Climate Data, `env_ref_data`

The environmental reference / climate data should contain a reference value (column "ref_value") of the environmental variable per geographic group for each week of the year. For example, this could be the historical mean for that week of the year.

- `{groupfield}`: Geographic grouping field, and must match the field names in the environmental & epidemiological datasets.
- `{obsfield}`: Environmental variable field, and must match the field names in the environmental dataset.
- `week_epidemiar`: Week of the year (1 to 52 for CDC, or 1 to 52 or 53 for WHO/ISO).
- `ref_value`: Historical mean, or other reference value, for that week of the year for that `groupfield` for that `obsfield`.
- `ref_*`: You can have other field(s) in here that begin with `ref_`. These fields will propogate through to the `environ_timeseries` dataset in the ouput, which you can then use for plotting or other uses.

If you have `env_data`, but do not yet have a reference/climatology built from it, you can use the `env_daily_to_ref()` function to create one in the format accepted by `run_epidemiar()` for `env_ref_data`. Because of processing time (especially for long histories), it is recommended that you run this infrequently to generate a reference dataset that is then saved to be read in later, rather than regenerated each time. The `week_type` defaults to "ISO" for ISO8601/WHO standard week of year. This function also requires the `env_info` data, see below.

1.1.4 Reference Data

1. Environmental variables, `env_info` This file lists the environmental variables and their aggregation method for to create weekly environmental data from daily data, e.g. rainfall could be the 'sum' of the daily values while LST would be the 'mean' value.

- `{obsfield}`: Give the field name of the environmental data variables, should match the environmental and environmental reference data.
- `reference_method`: 'sum' or 'mean', the aggregation method for to create weekly environmental data from daily data.
- `report_label`: Label to be used in creating the formatted report graphs. This column is not used until the formatting Rnw script, so depending on your setup and how you are have formatting reports after the report data is generated, you may not need this column.

2. Shapefiles In order to create summaries from Google Earth Engine, you will need to upload assets of the shapefile of your study area. If you are not using GEE and have some other way of obtaining environmental data, you may not need this.

If you are creating a formatted report later and wish to have maps of the results, you may need shapefiles for this.

1.2 Setting up the Report and Model

1.2.1 Setting up for Forecasting

- `report_period`: Total number of weeks for the report to include, including the number of future forecast weeks, `forecast_future`.
- `forecast_future`: The number of weeks to forecast into the future. As the future values of the environmental variables are being imputed based on recent and historical values, it is not recommended to extend the forecast very far into the future, probably no longer than 12 weeks.

The rest of the forecasting controls are bundled into a named list `fc_control`:

- `fc_control$env_vars`: Environmental variables. This informs the modeling system which environmental variables to actually use. (You can therefore have extra variables or data in the environmental dataset.) This is just a simple 1 column tibble with the variable names to use - `obsfield` - same field name as in the environmental data and environmental reference datasets, with entries for which variables to use in the modeling.
- `fc_control$clusters`: Geographic grouping clusters. This is a two-column list matching the geographic group to its cluster number. There must be an entry for each geographic group included in the epidemiological data. The fields are: the geographic group field, `groupfield`, and "cluster_id", the numeric ID number for each geographic group. If you only have one cluster (global model), each entry for the geographic group should contain the same "cluster_id" value. If you only have one geographic group, this should contain one row for that geographic group with a "cluster_id" (1, for example). If you want each geographic group to be in its own cluster (individual model), then each entry should contain a unique value (e.g. 1 to the number of geographic groups).
- `fc_control$lag_length`: The number of days of past environmental data to include for the lagged effects.
- `fc_control$fit_freq`: When fitting the model, either fit "once" (highly recommended) or per every "week". Per "week" will increase the processing time by the number of weeks in the model. It is recommended to only use "once" unless you are doing detailed analyses on the difference.
- `fc_control$ncores`: For the number of threads argument for model processing, the number of cores to use. If unset, it will default to the number of physical cores available minus one.
- `fc_control$anom_env`: Boolean argument indicating if the environmental variables should be replaced with their anomalies. The variables were transformed by taking the residuals from a GAM with geographic unit and cyclical cubic regression spline on day of year per geographic group. Default is true, that anomalies will be calculated and used.

1.2.2 Setting up for Event Detection

- `ed_summary_period`: The last n weeks of known epidemiological data that will be considered the early detection period for alert summaries. The algorithm will run over the entire report length for each geographic group and mark alerts for all weeks, but it will create the early detection summary alerts

only during the `ed_summary_period` weeks. The early detection summary alerts are recorded in the `summary_data` item in the output. In the demo, we have both displayed the results as a map and listed in tables.

- `ed_method`: At the moment, the only choices are "Farrington" for the Farrington improved algorithm or "None".

- `ed_control`: This is a list of parameters that are handed to the `surveillance::farringtonFlexible()` function as the `control` argument for "Farrington" option. It is unused for the "None" option. See the help for `surveillance::farringtonFlexible()` for more details. In our use of the function, the user can leave `b`, the number of past years to include in the creation of the thresholds, as NULL (not set) and epidemiar will calculate the maximum possible value to use, based on what data is available in `epi_data`.

1.3 Setting up Model Input (Optional)

- `model_run`: This is a boolean indicating if it should ONLY generate and return the regression object (`model_obj`) and metadata (`model_info`) on the model.
- `model_obj`: Once a model has been generated, it can be fed into `run_epidemiar()` using this argument. This will skip the model building portion of forecasting, and will continue start into generating predictions.

Pre-generating a model can save substantial processing time, and users can expect faster report data generation time. The trade-off of potential hits to model accuracy in the age of the model versus the time range of the requested predictions should be examined, which would vary depending on the situation/datasets.