EAN GIS mini module: R mapping pre-course instructions

Adapted from material by Daniel Gardiner and Amy Mikhail

8 - 10 April 2022

Table of Contents

[Preamble 1](#_Toc99643358)

[A note on package selection 1](#_Toc99643359)

[Required packages 2](#_Toc99643360)

# Preamble

This case study was originally developed by Amy Mikhail and Daniel Gardiner for a beginner’s R course for epidemiologists that was run in Public Health England from 2017 to 2019. The case study is based on an outbreak that occurred in the South West of England in 2014. The data sets used in this exercise are derived from the original outbreak, but they have been anonymised (postcodes jittered and patient identifying information removed).

The material was then adapted slightly for the UK FETP Outbreak Tools module in 2019.

Substantial changes have been made to this version to bring it up to date. In particular:

* PHE-specific packages have been replaced by publicly available packages from CRAN
* Outdated packages and soon-to-be deprecated packages have been replaced
* Where possible, R packages compatible with the tidyverse have been selected
* The example code in this R markdown guide has been substantially updated

# A note on package selection

This case study uses dplyr for manipulation of data sets (adding and updating variables) and other packages from the tidyverse for all the mapping tasks. The tidyverse suite of packages are becoming increasingly popular as they share an intuitive syntax and make code easier to write and read. If you are working with very large data sets, however, the data.table package may be more suitable as it has been designed for fast execution and to minimise demands on available memory (RAM). A relatively new package, dtplyr has been created for those who want to take advantage of the greater speed and efficacy of data.table but prefer to use dplyr syntax.

There are many different options for performing tasks such as geocoding and cartography in R. In the examples below, packages have been selected on the basis that:

* They are part of the tidyverse or compatible with it;
* They have the capacity to work with data from any country;
* The object formats are inter-operable and require little to no transformation.

Finally, note that wherever a package that is not part of base R has been used, the function is preceded by the package name, e.g.: stringr::str\_replace() is the str\_replace function from the stringr package. This should make it easier to identify which packages you will need to use when writing your own mapping code. For the same reason, wherever possible arguments within functions have been explicitly named, so that you can identify and read about the accepted values for these arguments in the package help files.

Happy mapping!

# Required packages

You can use the code overleaf to install and load the packages that you will need for this session. Some brief details on why each package was selected are provided below:

* here - setting of working directory containing materials for this practical
* tidyverse - collection of packages based on tidy data format and dplyr syntax
* tidygeocoder - geocoding (converting UK postcodes to longitude and latitude)
* sf - converts meta data and coordinates to a “simple feature” object for maps
* osmdata - free OpenStreetMap resource for creating map boundary boxes
* ggmap - “grammar of graphics” package for creating and displaying static maps
* scales - create pretty breaks for choropleth maps
* leaflet - java package for making interactive maps with more complex features
* htmlwidgets - allows saving of leaflet maps as html files

The first element in the code below compiles all the required packages into a list.

The second element will check if you already have these packages in your R library and install them if not (note - you will need an internet connection and to have selected your preferred CRAN mirror prior to running this line).

Once any missing packages have been installed, the third element will load all the required packages from your R library. If you have any problems with package installation or loading, please ask the facilitators for help before proceeding.

##################################################################  
# INSTALL PACKAGES  
  
# Create a list of CRAN packages you will need for this session:  
pkglist <- c("here",   
 "tidyverse",  
 "tidygeocoder",  
 "sf",   
 "osmdata",  
 "ggmap",  
 "scales",  
 "leaflet",  
 "htmlwidgets")  
  
# Check if packages are already installed, if not install them:  
invisible(lapply(pkglist, function(x)   
 if (!requireNamespace(x, quietly = TRUE)) install.packages(x)))  
  
  
##################################################################  
# LOAD PACKAGES:  
  
# Loop through the CRAN packages in the list with lapply and load them:  
invisible(lapply(pkglist, library, character.only = TRUE))