

CDCPlot: an Application for Viewing Weekly CDC MMWR Disease Count Data

Nicholas L. Michaud*¹, Aaron Kite-Powell² and Jarad Niemi¹

¹Iowa State University Statistics Department, Ames, IA, USA; ²Armed Forces Health Surveillance Center, Silver Spring, MD, USA

Objective

To demonstrate the current features and functionality of the CDCPlot application, and to introduce potential new features of the application.

Introduction

The CDC provides data on incidences of diseases on its website (<https://data.cdc.gov/>). Data is available at national, regional, and state levels, and is uploaded to the CDC's website on a weekly basis. The CDCPlot web application (available at <https://michaud.shinyapps.io/CDCPlot/>), built using the Shiny package in R, provides a quick and user-friendly method of visualizing this data. Users are able to select timeframes, locations, and diseases which they wish to view, and plots are produced. There is an optional alert threshold, which will alert users when a disease increases significantly from one week to the next. In addition, CDCPlot provides visualizations of CDC data on Pneumonia and Influenza mortality.

Methods

An integral feature of CDCPlot is automated weekly updating. Every Thursday, a script is run which scrapes data for individual diseases from the <https://data.cdc.gov/> website. This individual disease data is reformatted and combined into a single data file which contains counts for all diseases, across all locations. For each disease and location, we compute alert thresholds, and set alerts when the weekly disease counts surpass these thresholds. Alert thresholds are calculated using a moving average technique.

The Shiny R package lets us create menus which allow users to specify diseases, locations, and time ranges to plot. Data can be plotted as weekly counts or cumulative counts, with the option to overlay multiple year's worth of data for comparison. Plots are made using the ggplot2 R package. Source code for the app is available at <https://github.com/NLMichaud/WeeklyCDCPlot>.

The app can also be run locally from within R by a call to: `runGitHub('NLMichaud/WeeklyCDCPlot')`

Results

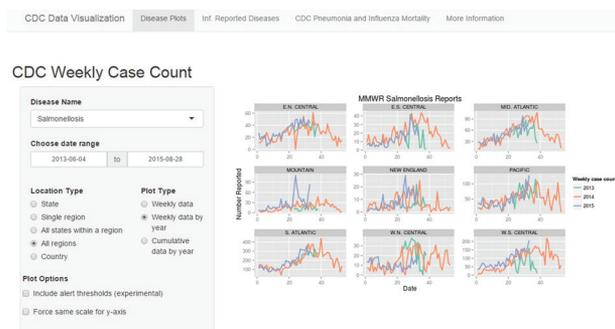
CDCPlot can be used to quickly compare disease counts across locations and time periods, simplifying the process of obtaining and viewing data. The different geographical scales that the app provides can also be used to easily identify specific locations that are giving rise to high disease counts. For example, looking at Legionellosis data for the whole country shows a spike in cases in mid-August of 2015. Refining our plots to look at individual regions shows that most of these cases came from the Mid-Atlantic region. Refining further still to look at all states and locations within the Mid-Atlantic region reveals a large spike in cases in New York City, which was experiencing an outbreak of Legionellosis at that time.

Conclusions

Potential future additions to the app include:

- Tables displaying data counts for user-selected diseases and time ranges.
- Additional methods for calculating alert thresholds.
- Maps displaying geographical distributions of disease counts.

- An automated news filtering option, which would search through recent news stories to see if disease incidence spikes for user selected diseases can be tied to news reports.



Keywords

CDC; Disease; Visualization

*Nicholas L. Michaud

E-mail: michaud@iastate.edu

