R Shiny web application for real-time seismic visualization – Jack Colpitt

**Abstract --**

Earthquakes play a pivotal role along the active tectonic boundaries of the planet. The availability of real-time data feeds is bringing a new instantaneous approach to analysis. This study suggests the creation of an R Shiny application for earthquake data analysis. The program will take GeoJSON data from the USGS (US Geological Survey) earthquake database and a series of user inputs to perform spatial clustering analysis via DBSCAN. The data has a 1-minute temporal resolution and is refreshed each time the application loads. R programming packages sf, rgdal, leaflet, leaflet extras, lubridate, tidyverse, shiny, and DT will be integrated into the application to provide representations of current spatiotemporal data. The interactive component of the Shiny application enables speedy consumption of current seismic events, as well as easy comparison of properties. More study is needed to obtain testing from end users to find effective analytical methods of spatial clustering. In short, the R Shiny application will use the R programming language's capabilities and the collaborative nature of web services to instantly assess current seismic events throughout the world.

**Introduction --**

Web applications allow for the distribution of information and analysis to end users in real time. In contrast, desktop-based analysis software can cause limitations and barriers to entry based on distribution hurdles and programming dependencies. This study's objective is to create an R Shiny application that makes analysis of real-time earthquake data from the USGS (US Geological Survey) accessible to a variety of users. Notably, users that interested in visualizing the data via the density-based clustering algorithm DBSCAN. The application will let users select a location on the map and define the search distance and minimum features per cluster in the DBSCAN tool. There are currently a variety of web GIS (Geographical Information Systems) applications available to the public that provide point and click information and filtering to understand where and when earthquakes are happening in real time. However, R Shiny applications leveraging open web mapping that are dedicated to visualizing spatial cluster analysis of real-time geospatial data still have potential for advancement. This project is open to replicate and extend since the source code is publicly available.

**Data --**

This study is incorporating the real-time feed of GeoJSON earthquake data within the last 30 days and has a temporal resolution of 1 minute (available here: <https://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php>). The web service contains the data structure visible in **Figure 1**. Magnitudes equal to and greater than 2.5 have been selected due to the significant load of features below that value. Reducing the data to a magnitude equal to and greater than 2.5 reduces the table size from ~12,000 rows to ~2000 rows. The latitude and longitude will provide location information for the clustering analysis. The class breaks renderer will be determined by scales of magnitude, and filtering based on magnitude and seismic monitoring stations are also available as user input widgets.

Text

Description automatically generated

**Figure 1.**

*Shows the data structure for the USGS GeoJSON web service.*

**Methodology --**

The Shiny application's interactive components allow for quick consumption of current seismic events, comparing attributes, and analyzing spatial density based on clustering. Once the data is extracted from the USGS web service utilizing the rgdal library, the data is processed, and ready analysis. The workflow for data processing can be seen in **Figure 2**. Libraries involved in the study include sf, rgdal, leaflet, leaflet extras, lubridate, tidyverse, shiny, and DT.

Diagram

Description automatically generated

**Figure 2**. *On the left provides an overview of the data wrangling process.*

R Shiny applications are created by supplying the server and user interface (UI) components in a single R script file. The application is deployed and web accessible at <https://pnw-geo.shinyapps.io/geoapp/> using shinyapps.io free tier cloud platform. Shiny was chosen due to its integration with the R programming language, and ease of deployment. Web frameworks can be time consuming on the developer’s end, however R Shiny provides deployment options that can be completed and web accessible in minutes. The current functionality of the app supports filtering, interactive popups, drawing tools for creating geometries, and a table view sorted by the most recent earthquakes. The leaflet.extras library provides the functionality for creating geometries to determine the clustering radius. Researching available libraries that will provide the processing necessary for the DBSCAN analysis is still in the development phase. Figure 3 provides an overview of the UI programming, Figure 4 showcases leaflet.extras, and Figure 5 previews server-side coding.

Text

Description automatically generated

Map

Description automatically generated with low confidence

**Figure 4.** *On the left showcases the leaflet.extras library for creating geometries by the end user. The radius for the DBSCAN parameter will be extracted from the rendered geometry.*

**Figure 3.** *Above shows an example of UI programming with the R Shiny library.*

Time constraints limit the availability of testing the application but will be required in the future to understand the effectiveness of the application. The shiny.telemetry library is available and provides developers with the tools to help understand how users interact with Shiny dashboards. It also answers questions such as: which tabs/pages are more often visited, which inputs users are changing, what is the average length of a session (Appsilon, 2023).

Text

Description automatically generated

**Figure 4.** *On the left shows an example workflow for coding the server side of a Shiny app.*

**Anticipated Results --**

The R shiny application will leverage the capabilities of the R programming language and the shareability of web services to rapidly analyze current seismic events around the globe. The ability to update data at a at a 1-minute interval will facilitate dynamic inputs for the analysis and visualization. Providing effective tools for analysis can vary based on the end user requirements. Testing the application with user feedback should be completed to find if the functionalities and analytical methods are accomplishing the goal of the study. With the source code publicly available at <https://github.com/tfjackc/R/blob/main/geoapp/app.R> the project is free to reproduce and extend.

**References --**

Appsilon. (2023, June 1). *GitHub - Appsilon/shiny.telemetry: Easy logging of users activity and session events of your Shiny App*. GitHub. https://github.com/Appsilon/shiny.telemetry