Building a Shiny web application to visualize spatial and spatio-temporal data

Paula Moraga

RaukR Summer School

Visby, 19 June 2018

Outline

Shiny

SpatialEpiApp

Tutorial: Shiny

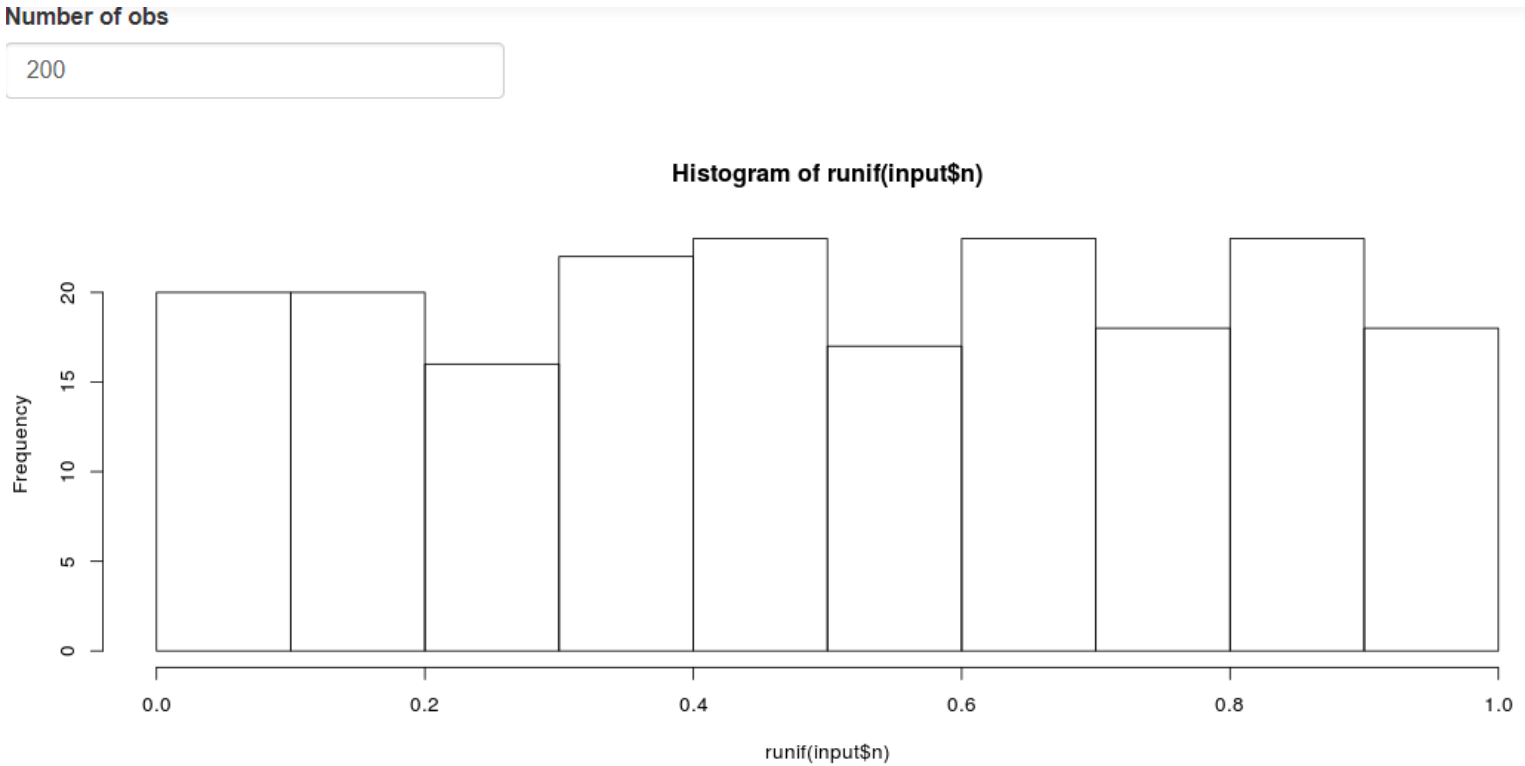
Shiny

Shiny

- Shiny is a web application framework for R that enables to build interactive web applications
- https://shiny.rstudio.com/

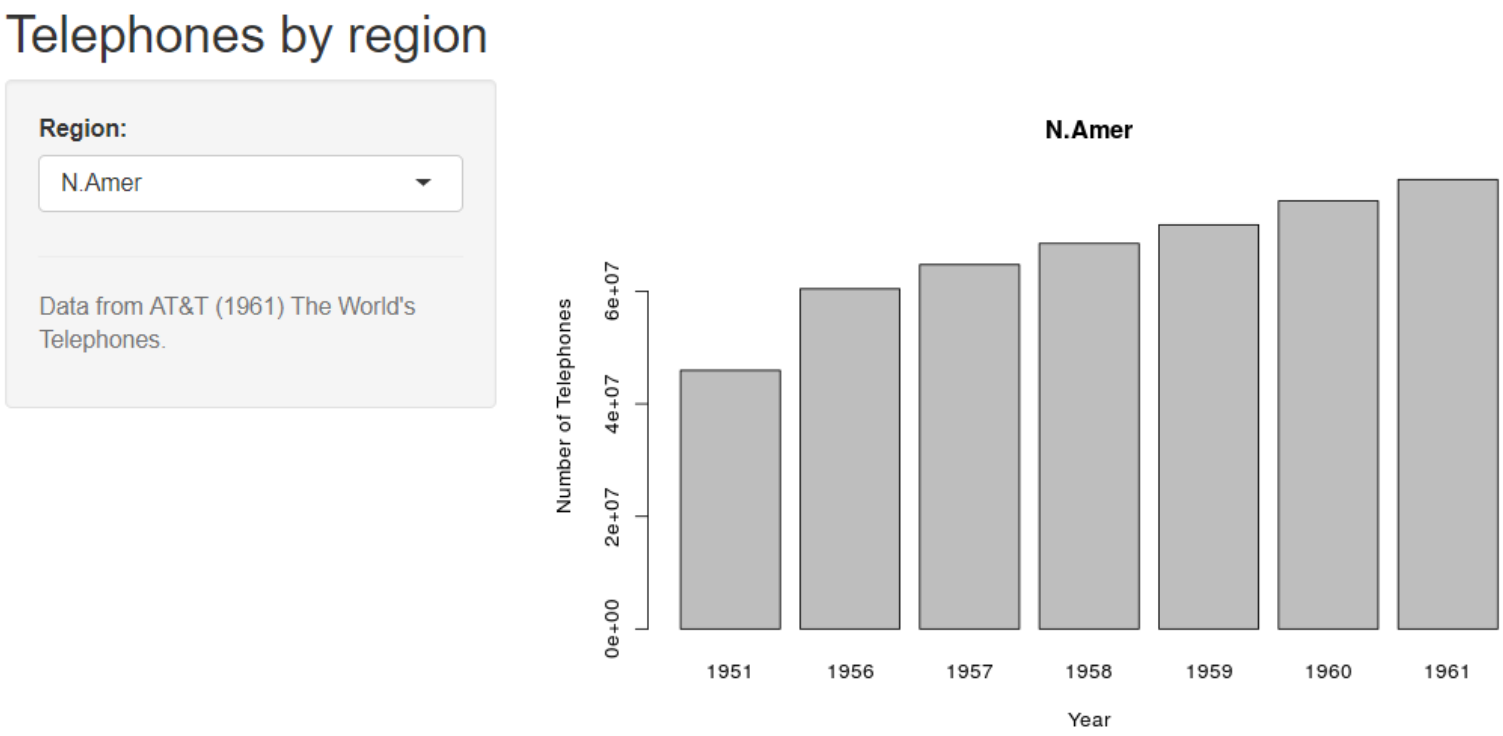
Examples

https://shiny.rstudio.com/gallery/single-file-shiny-app.html



Examples

https://shiny.rstudio.com/gallery/telephones-by-region.html



Structure of a Shiny App

A Shiny app is a directory that contains an R file called `app.R`. `app.R` has three components:

- user interface object (`ui`) which controls the layout and appearance of the app
- `server()` function with the instructions to build the objects displayed in the `ui`
- call to `shinyApp()` that creates the Shiny app from the `ui`/`server` pair

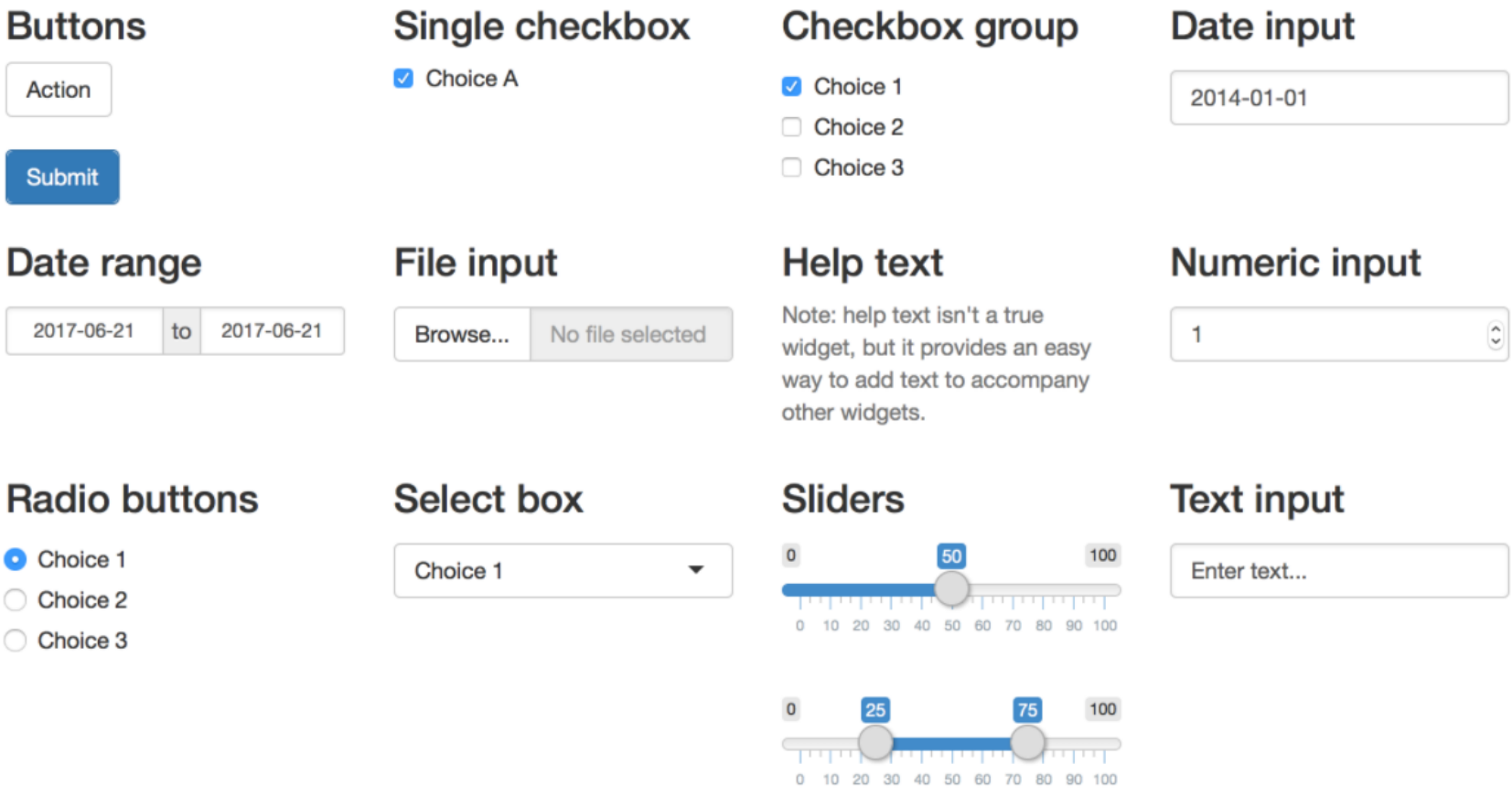
Content app.R

```r
# define user interface object
ui <- fluidPage( )

# define server() function
server <- function(input, output){ }

# call to shinyApp() which returns the Shiny app
shinyApp(ui = ui, server = server)
```

Save app.R inside the appdir directory. Launch the app:

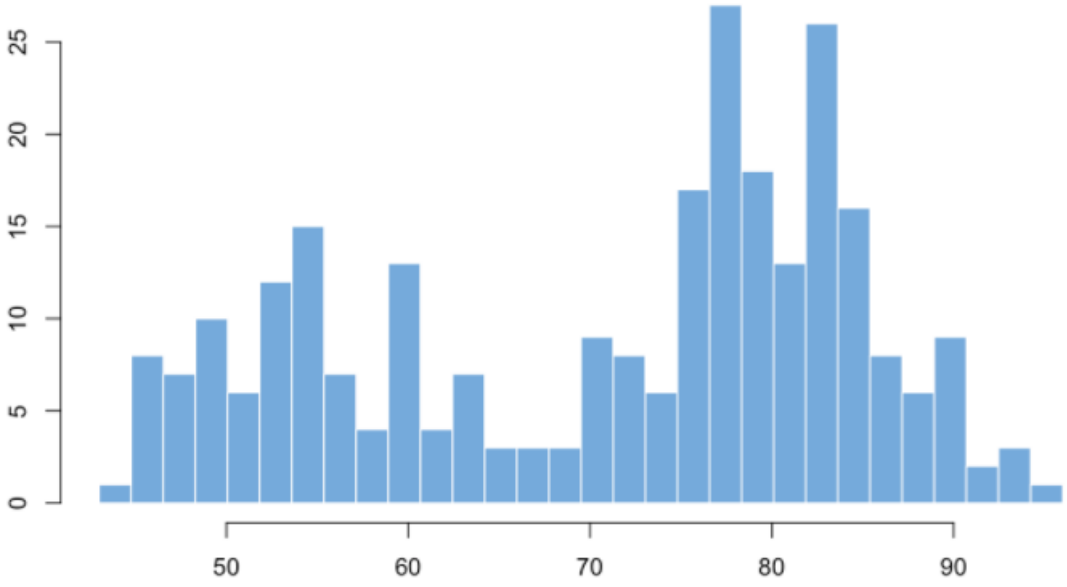
```r
library(shiny)
runApp("appdir_path")
```

Inputs


Buttons
Action
Submit
Single checkbox
Choice A
Checkbox group
Choice 1
Choice 2
Choice 3
Date input
2014-01-01
Date range
2017-06-21
to
2017-06-21
File input
Browse...
No file selected
Help text
Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.
Numeric input
1
Radio buttons
Choice 1
Choice 2
Choice 3
Select box
Choice 1
Sliders
0
50
100
0 10 20 30 40 50 60 70 80 90 100
0
25
75
100
0 10 20 30 40 50 60 70 80 90 100
Text input
Enter text...

Outputs

- Plots, tables, texts, images



area	peri	shape	perm
4990	2791.90	0.09	6.30
7002	3892.60	0.15	6.30
7558	3930.66	0.18	6.30
7352	3869.32	0.12	6.30
7943	3948.54	0.12	17.10
7979	4010.15	0.17	17.10
9333	4345.75	0.19	17.10
8209	4344.75	0.16	17.10
8393	3682.04	0.20	119.00
6425	3098.65	0.16	119.00

First level title

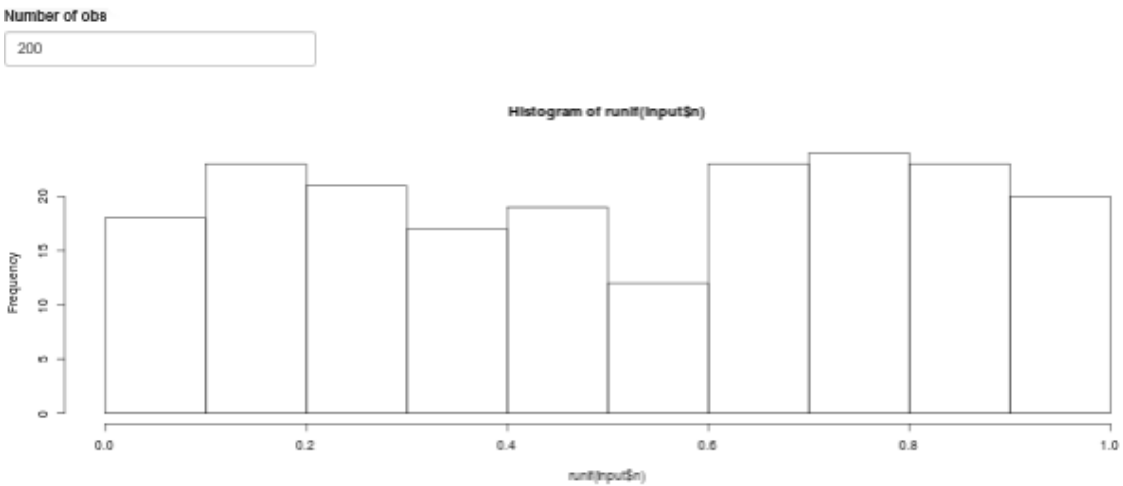
Second level title

Inputs, outputs and reactivity

- Inputs: we can interact with the app by modifying their values
- Outputs: objects we want to show in the app

```
ui <- fluidPage(
  *Input(inputId = myinput, label = mylabel, ...)
  *Output(outputId = myoutput, ...)
)

server <- function(input, output){
  output$myoutput <- render*({
    # code to build the output.
    # If it uses an input value (input$myinput),
    # the output will be rebuilt whenever
    # the input value changes
  })}
```

Inputs, outputs and reactivity



app.R

```r
# Global variables can go here
n <- 200


# Define the UI
ui <- bootstrapPage(
  numericInput('n', 'Number of obs', n),
  plotOutput('plot')
)


# Define the server code
server <- function(input, output) {
  output$plot <- renderPlot({
    hist(runif(input$n))
  })
}

# Return a Shiny app object
shinyApp(ui = ui, server = server)
```

HTML widgets

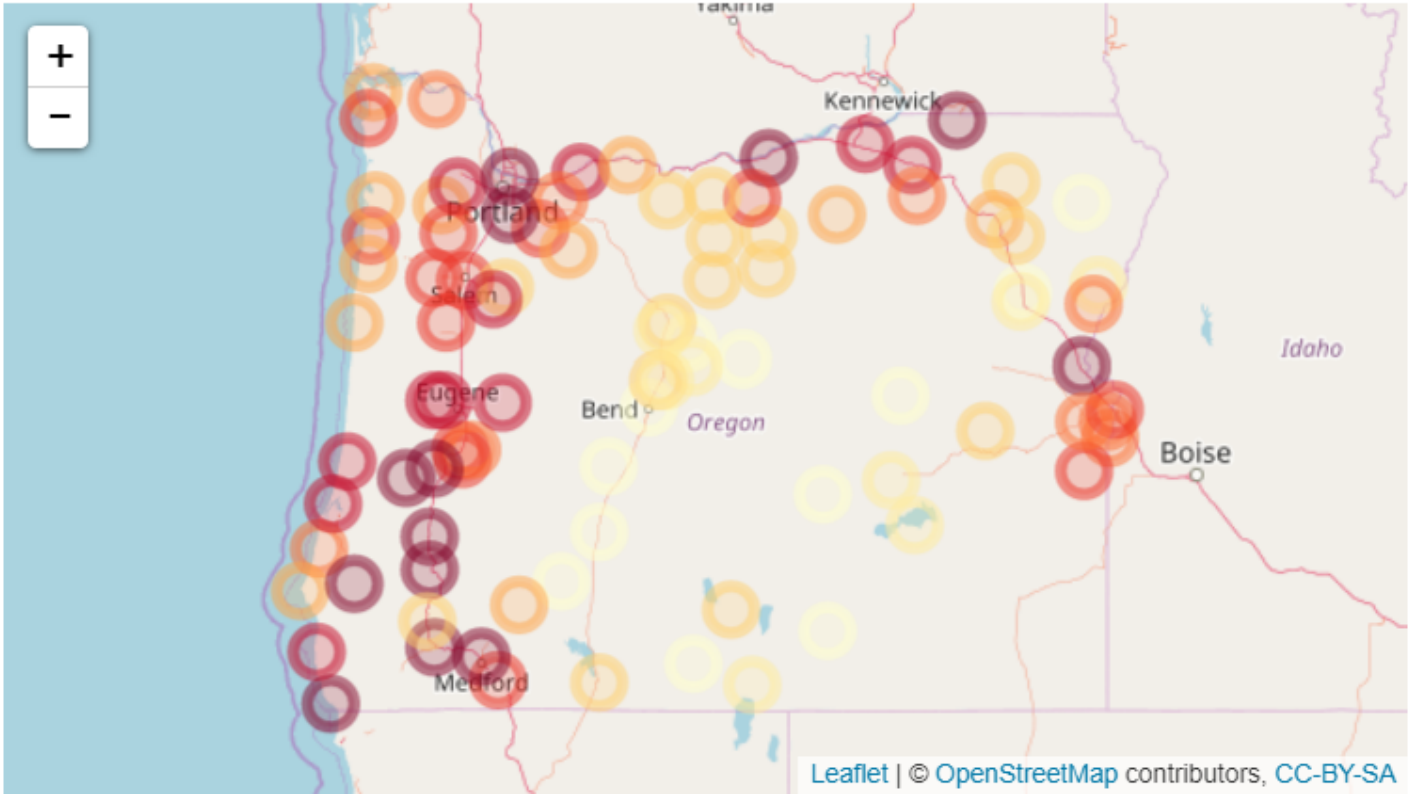
Interactive web visualizations using JavaScript

http://www.htmlwidgets.org/

Leaflet

http://rstudio.github.io/leaflet/

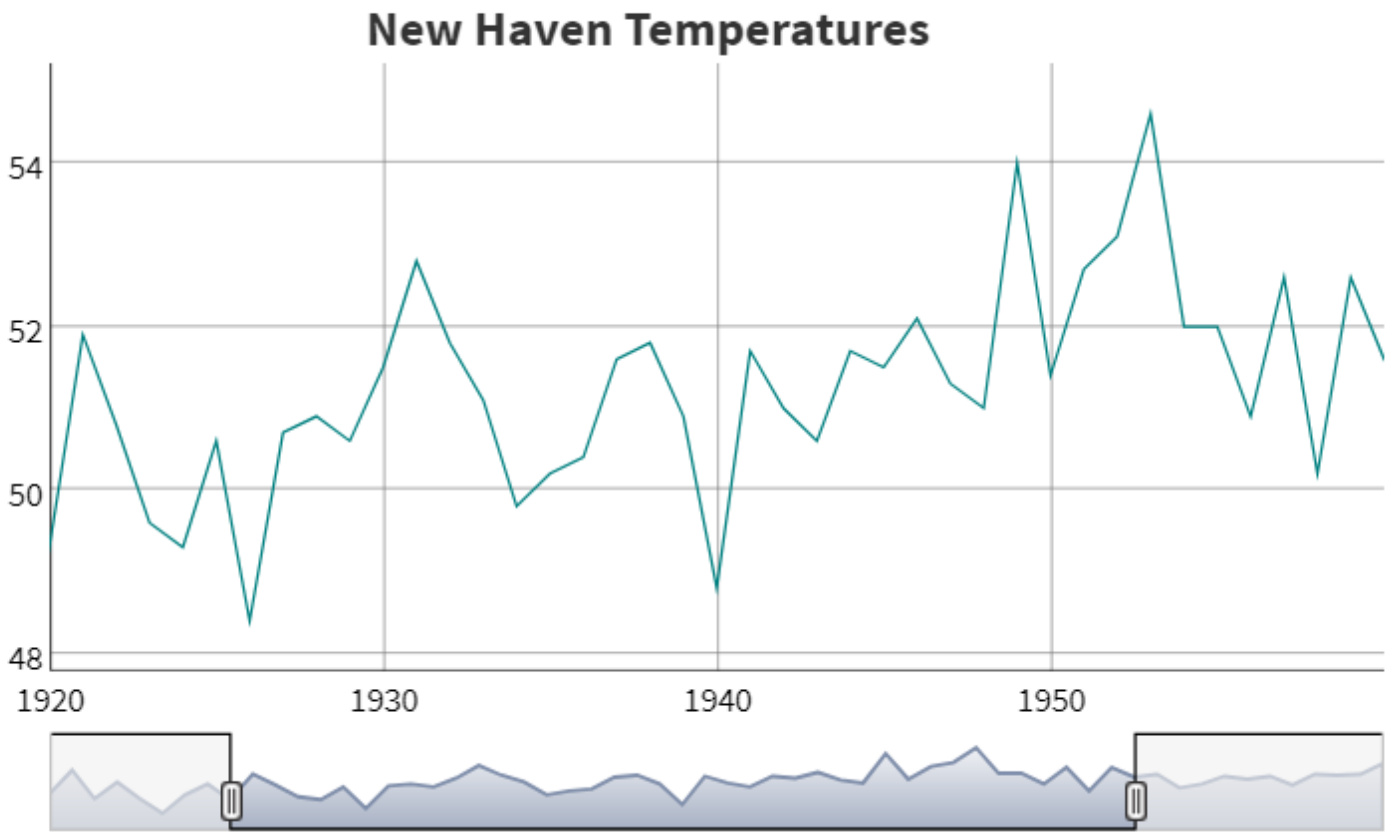
```
library(leaflet)
pal <- colorQuantile("YlOrRd", NULL, n = 8)
leaflet(orstationc) %>%
  addTiles() %>%
  addCircleMarkers(color = ~pal(tann))
```



Dygraphs

http://rstudio.github.io/dygraphs

```
library(dygraphs)
dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyRangeSelector(dateWindow = c("1920-01-01", "1960-01-01"))
```



DataTables

http://rstudio.github.io/DT/

```
library(DT)
datatable(iris, options = list(pageLength = 5))
```

Show 5 entries Search:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa

Showing 1 to 5 of 150 entries

Previous 1 2 3 4 5 ... 30 Next

Options to share a Shiny app

1. Share R scripts with other users

   - need R

```
library(shiny)
runApp("appdir_path")
```

2. Host app as a web page at its own URL

   - do not need R
   - app can be navigated through the internet with a web browser
   - host apps on own servers or using one of the ways RStudio offers such as shinyapps.io and Shiny Server

https://paulamoraga.shinyapps.io/spatialepiapp/

SpatialEpiApp

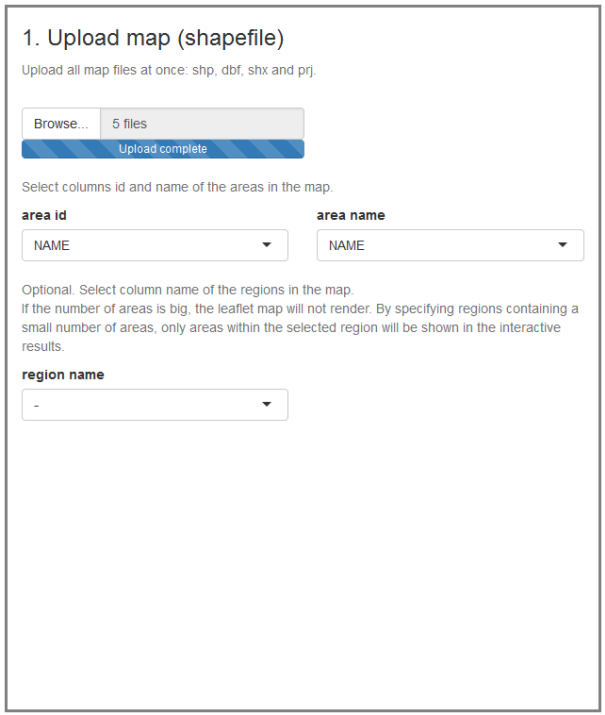
R package SpatialEpiApp

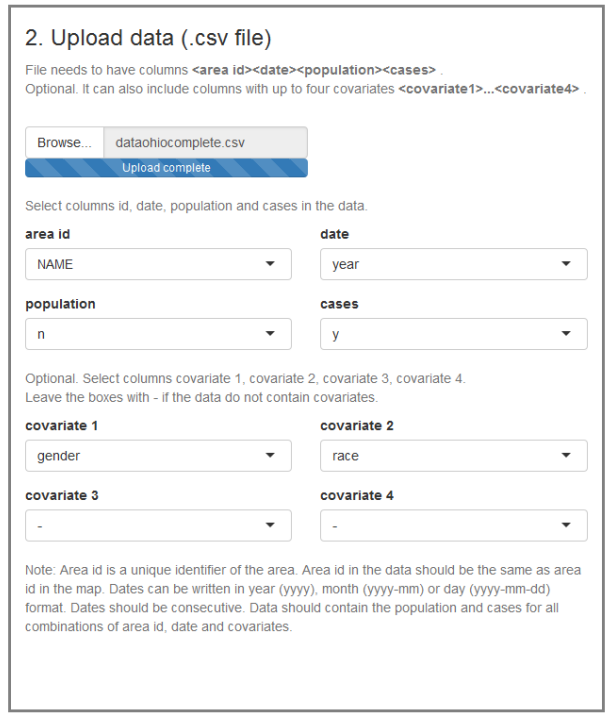
- Shiny web application that allows to visualize spatial and spatio-temporal disease data, estimate disease risk and detect clusters
- Risk estimates by fitting Bayesian models with INLA
- Detection of clusters by using the scan statistics in SaTScan

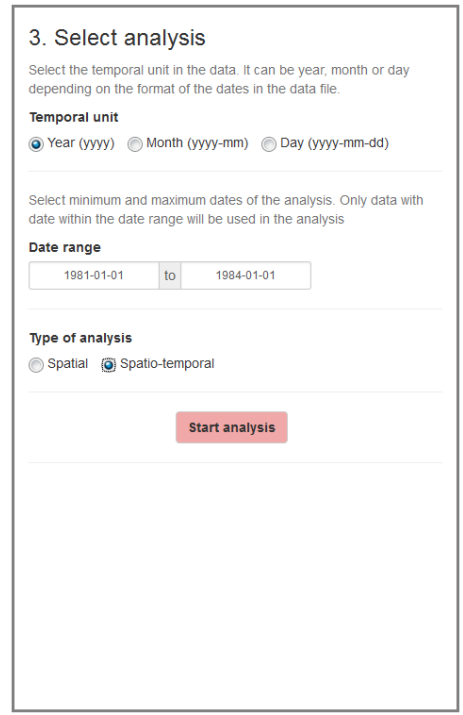
Launch SpatialEpiApp:

```
install.packages("SpatialEpiApp")
library(SpatialEpiApp)
run_app()
```

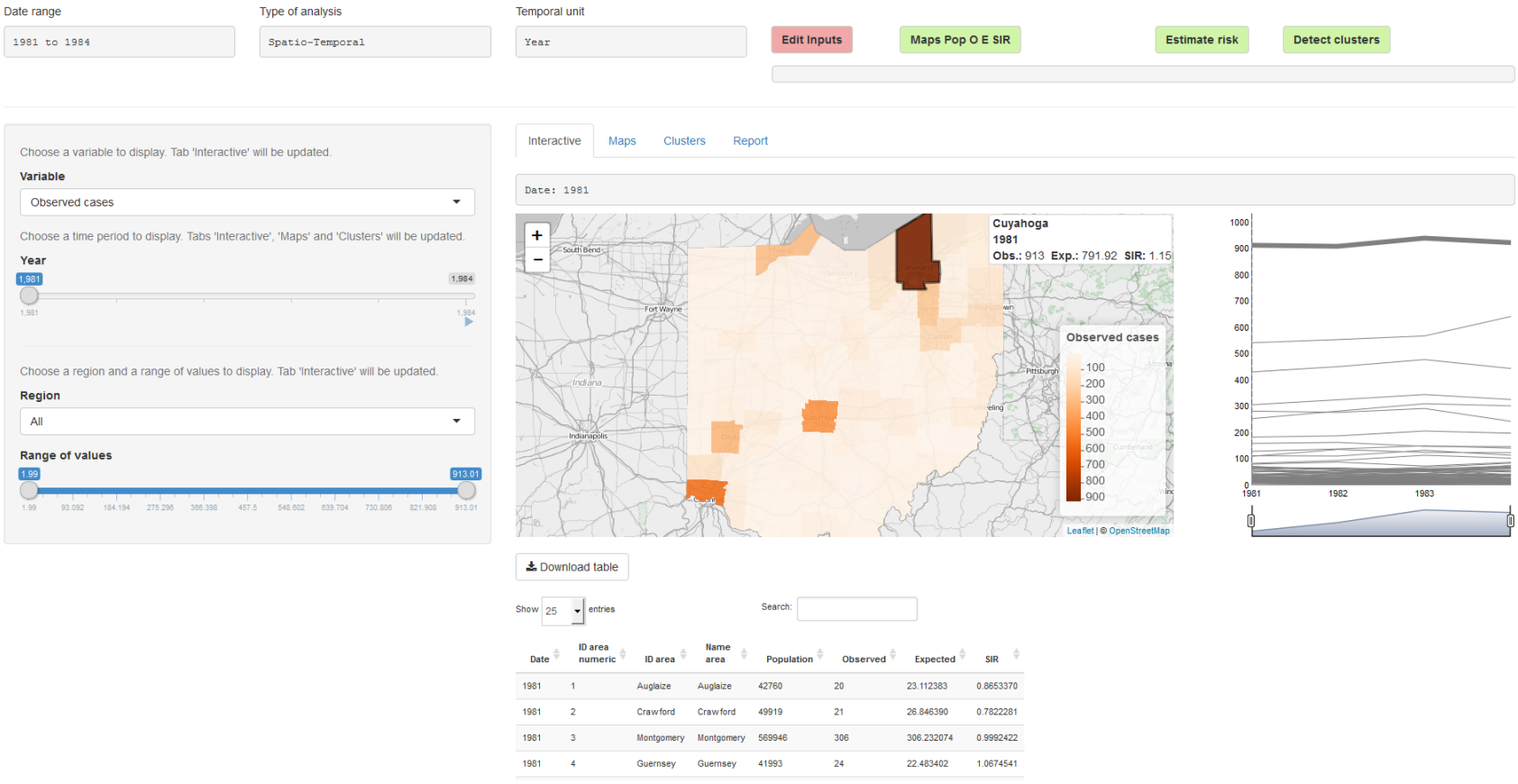
Data entry


1. Upload map (shapefile)
Upload all map files at once: shp, dbf, shx and prj.
Browse...
5 files
Upload complete
Select columns id and name of the areas in the map.
area id
NAME
area name
NAME
Optional. Select column name of the regions in the map.
If the number of areas is big, the leaflet map will not render. By specifying regions containing a small number of areas, only areas within the selected region will be shown in the interactive results.
region name
-


2. Upload data (.csv file)
File needs to have columns <area id><date><population><cases> .
Optional. It can also include columns with up to four covariates <covariate1>...<covariate4> .
Browse...
dataohiocomplete.csv
Upload complete
Select columns id, date, population and cases in the data.
area id
NAME
date
year
population
n
cases
y
Optional. Select columns covariate 1, covariate 2, covariate 3, covariate 4.
Leave the boxes with - if the data do not contain covariates.
covariate 1
gender
covariate 2
race
covariate 3
-
covariate 4
-
Note: Area id is a unique identifier of the area. Area id in the data should be the same as area id in the map. Dates can be written in year (yyyy), month (yyyy-mm) or day (yyyy-mm-dd) format. Dates should be consecutive. Data should contain the population and cases for all combinations of area id, date and covariates.


3. Select analysis
Select the temporal unit in the data. It can be year, month or day depending on the format of the dates in the data file.
Temporal unit
Year (yyyy)
Month (yyyy-mm)
Day (yyyy-mm-dd)
Select minimum and maximum dates of the analysis. Only data with date within the date range will be used in the analysis
Date range
1981-01-01
to
1984-01-01
Type of analysis
Spatial
Spatio-temporal
Start analysis

Interactive



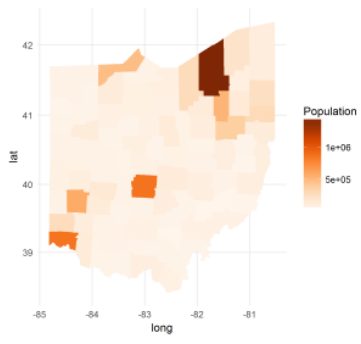
Maps

Interactive | Maps | Clusters | Report

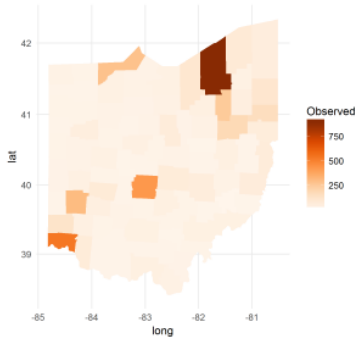
```
Date: 1981
```

```
   Population        Observed        Expected          SIR
 Min.   :  11253   Min.   :  2.00   Min.   :  6.108   Min.   :0.3274
 1st Qu.:  32643   1st Qu.: 14.00   1st Qu.: 17.723   1st Qu.:0.7179
 Median :  54588   Median : 22.50   Median : 29.381   Median :0.8641
 Mean   : 122617   Mean   : 62.58   Mean   : 66.037   Mean   :0.8709
 3rd Qu.: 104815   3rd Qu.: 50.00   3rd Qu.: 56.468   3rd Qu.:1.0115
 Max.   :1481287   Max.   :913.00   Max.   :791.923   Max.   :1.4989
```

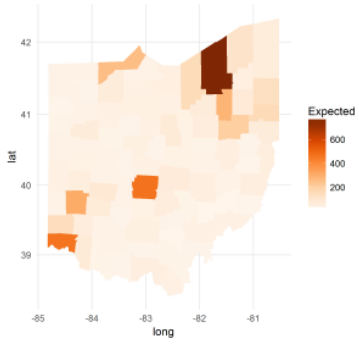
Population



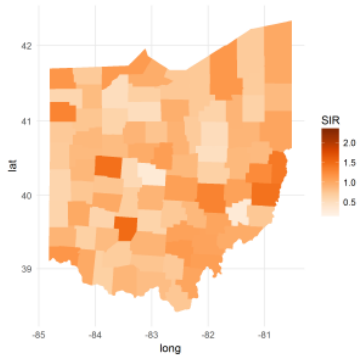
Observed

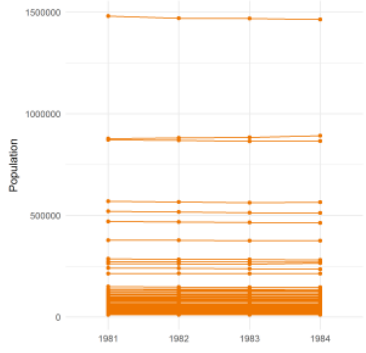


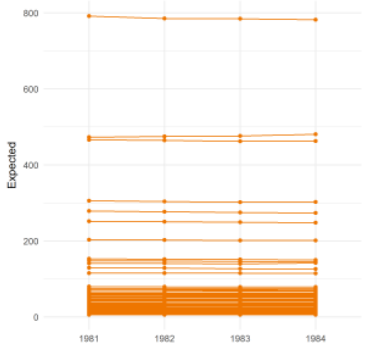
Expected

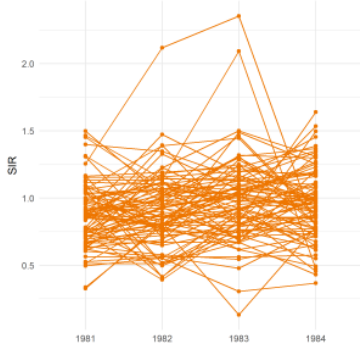


SIR





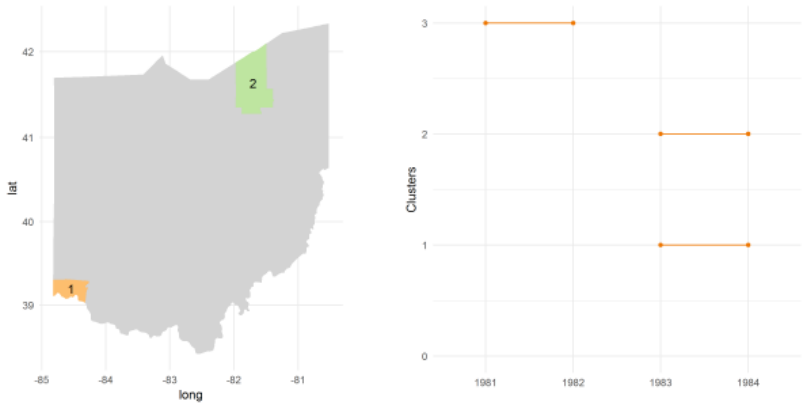




Clusters

Interactive | Maps | Clusters | Report

Date: 1984



Show 25 entries

Search:

Cluster	Central area	No. areas	Start date	End date	Risk in / Risk out	LLR	p-value	Areas
1	Hamilton	1	1983	1984	1.32	41.75818	1.23e-14	Hamilton
2	Cuyahoga	1	1983	1984	1.21	28.87297	1.04e-09	Cuyahoga
3	Belmont	5	1981	1982	1.30	10.54458	1.06e-02	Guernsey, Monroe, Harrison, Belmont, Jefferson
Cluster	Central area	No. areas	Start date	End date	Risk in / Risk ou	LLR	p-value	Areas

Showing 1 to 3 of 3 entries

Previous 1 Next

Report

Interactive | Maps | Clusters | Report

Download report

Choose the variables to include in the report. Variables that have not been calculated will not be included.

Maps

☑ Population ☑ Observed ☑ Expected ☑ SIR ☑ Risk ☑ 2.5 percentile ☑ 97.5 percentile ☑ Clusters

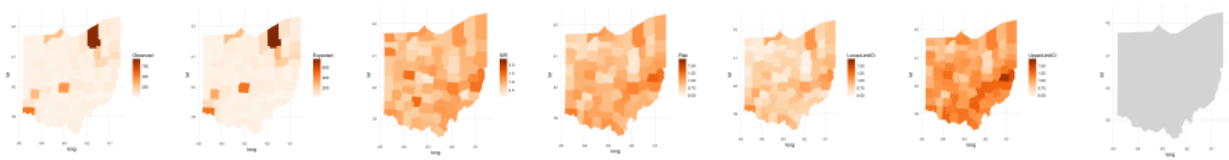
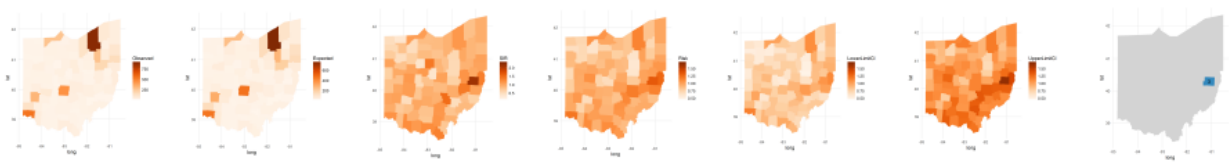
Tables summary

☑ Population ☑ Observed ☑ Expected ☑ SIR ☑ Risk ☑ 2.5 percentile ☑ 97.5 percentile

Table clusters

☑ Clusters

- Date range: 1981 to 1984
- Type of analysis: Spatio-Temporal
- Temporal unit: Year

	Observed	Expected	SIR	Risk	LowerLimitCI	UpperLimitCI	Clusters
1981							
1982							

Tutorial: Shiny

Tutorial: Shiny

https://paula-moraga.github.io/tutorial-shiny-spatial/

1 Shiny
2 Setup
3 Structure of app.R
4 Layout
5 HTML content
6 Content of app.R
7 Read data
8 Add outputs
9 Content of app.R
10 Add reactivity
11 Content of app.R
12 Upload data
13 Content of app.R
14 Handle missing inputs
15 Content of app.R
16 Conclusion
17 References

Code

Building a Shiny web application to visualize spatial and spatio-temporal data

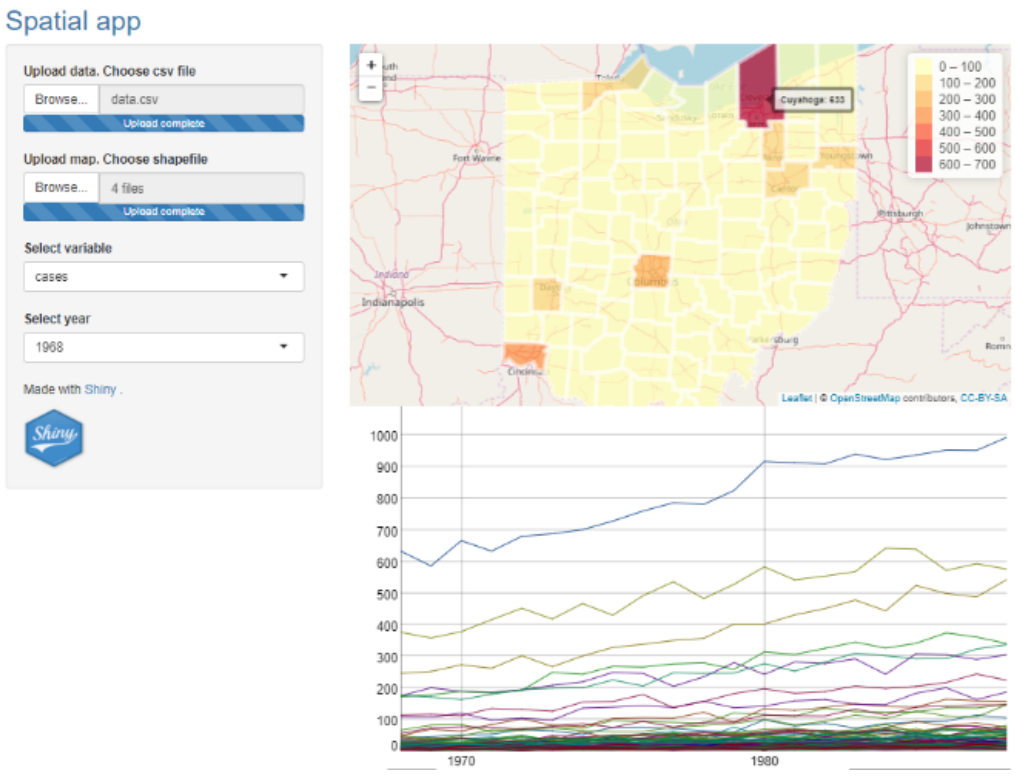
Paula Moraga

CHICAS, Lancaster University, UK

23 April 2018

In this tutorial we develop a Shiny web application to visualize spatial and spatio-temporal data. Specifically, the app shows the number of disease cases and the population in a given region using interactive maps, tables, and time series plots. The app allows the user to upload a csv file with the data and a shapefile with the map of the region. The app also permits selecting the variable and the time to be shown.

We develop the app using the R package `Shiny`. The interactive data visualizations are built using the packages `DT`, `dygraphs`, and `leaflet`. The example we use refers to data of the number of lung cancer cases and population in the 88 counties of Ohio, United States, during years 1968 to 1988. These data are in the package `SpatialEpiApp` and can also be downloaded from here.



References

- Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie and Jonathan McPherson (2017). shiny: Web Application Framework for R. https://CRAN.R-project.org/package=shiny

- Paula Moraga. SpatialEpiApp: A Shiny Web Application for the analysis of Spatial and Spatio-Temporal Disease Data, (2017), Spatial and Spatio-temporal Epidemiology, 23:47-57

Thanks!

https://Paula-Moraga.github.io

Twitter @_PaulaMoraga_