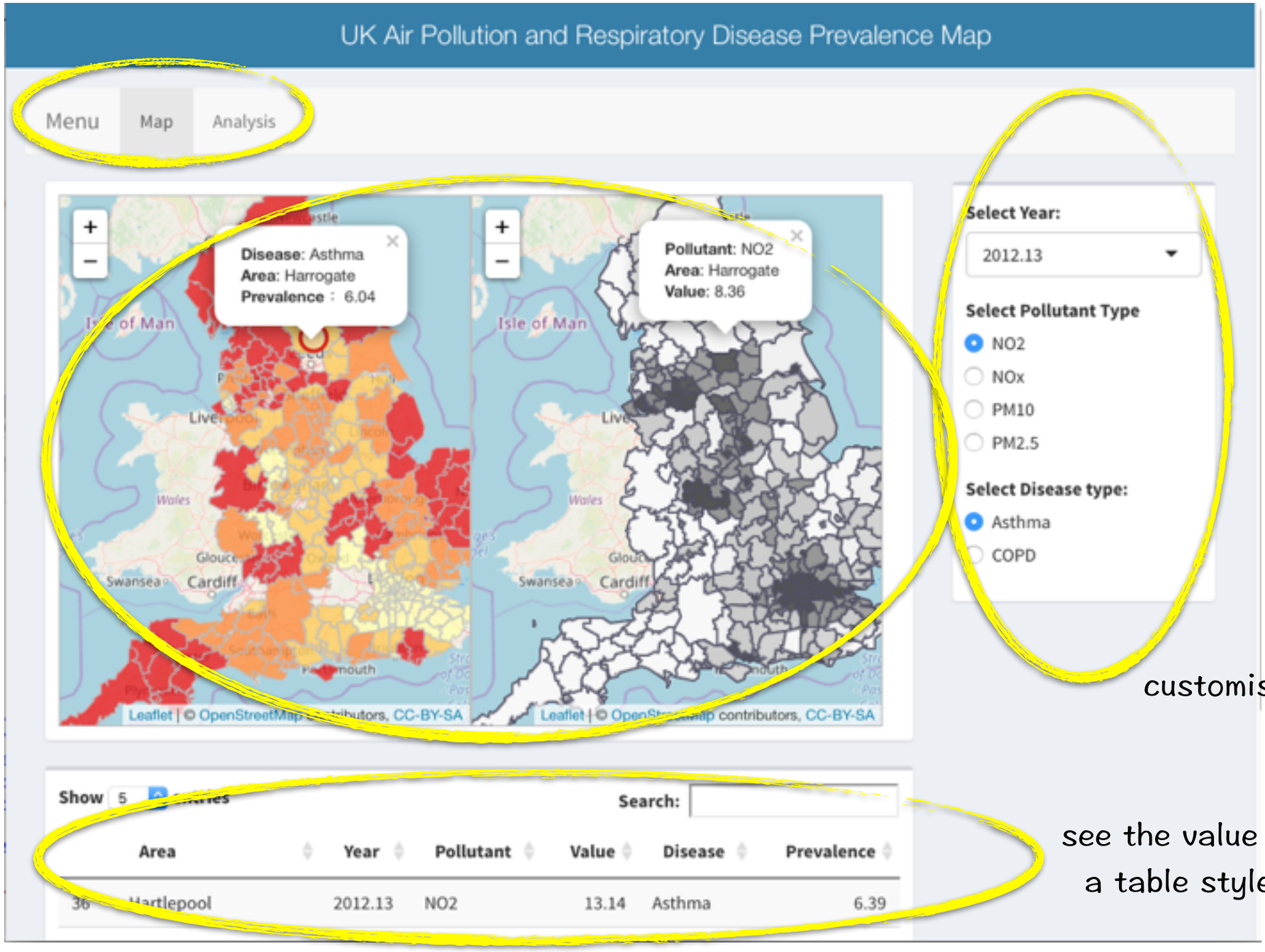
Shiny UK Map

Air Pollution vs Respiratory Disease

Tzu-hsien, Yang
August, 2019

different pages for visual impact and statistical analysis

maps



customised selection box

see the value in a table style

My Shiny app shows annual average value of air pollution and respiratory disease prevalence by UK area in an interactive map

The material data needed for this map

+

Local_Auth_Code	x	y	Total_PM2.5_12
104	544500	317500	10.947582
104	545500	317500	11.107196
104	546500	317500	10.944833
104	542500	316500	10.879427
104	543500	316500	10.887577
104	544500	316500	10.987135
104	545500	316500	11.194945
104	546500	316500	10.90475
104	539500	315500	10.989733
104	540500	315500	11.029616
104	541500	315500	11.058794
104	542500	315500	11.029599
104	543500	315500	11.020825

CCG.code	CCG.geograp	CCG.name.x	Year	Disease	Prevalence
02N	E38000001	NHS AIRED	2017.18	Asthma	7.21
02P	E38000006	NHS BARNS	2017.18	Asthma	5.71
02Q	E38000008	NHS BASSE	2017.18	Asthma	5.7
02R	E38000019	NHS BRADF	2017.18	Asthma	6.61
02T	E38000025	NHS CALDE	2017.18	Asthma	6.61
02W	E38000018	NHS BRADF	2017.18	Asthma	6.36
02X	E38000044	NHS DONCA	2017.18	Asthma	6.59
02Y	E38000052	NHS EAST R	2017.18	Asthma	6.71
03A	E38000064	NHS GREAT	2017.18	Asthma	6.54
03D	E38000069	NHS HAMBI	2017.18	Asthma	6.25
03E	E38000073	NHS HARRO	2017.18	Asthma	6.22

GIS Boundary

shapefile of local authority boundaries

Statistics

air pollution value and disease prevalence value

Key Packages

Package	Description	Author
shiny	For building interactive web applications in an easy way	Winston et al, 2019
dplyr	For easier data manipulation	Hadley et al, 2019
leaflet	For creating a map widget	Joe et al, 2018
rgdal	For providing bindings to the Geospatial Data Abstraction Library and access to projection/ transformation operations from the 'PROJ.4' library	Roger et al, 2019

Key Packages

Package	Description	Author
ggplot2	For graphically visualising data	H. Wickham, 2016
DT	For displaying data as tables on HTML pages, with functions	Yihui Xie et al, 2019
lme4	For fitting and analysing mixed models	Douglas et al, 2015
leafsync	a plugin for leaflet to produce synchronised leaflet web maps	Tim & Kenton, 2019

Key Packages

Package	Description	Author
shinyWidgets	For extending functions as custom widgets in package shiny	Victor et al, 2019
shinydashboard	For creating dashboards with Shiny in a creative way	Winston & Barbara, 2018

CODE !

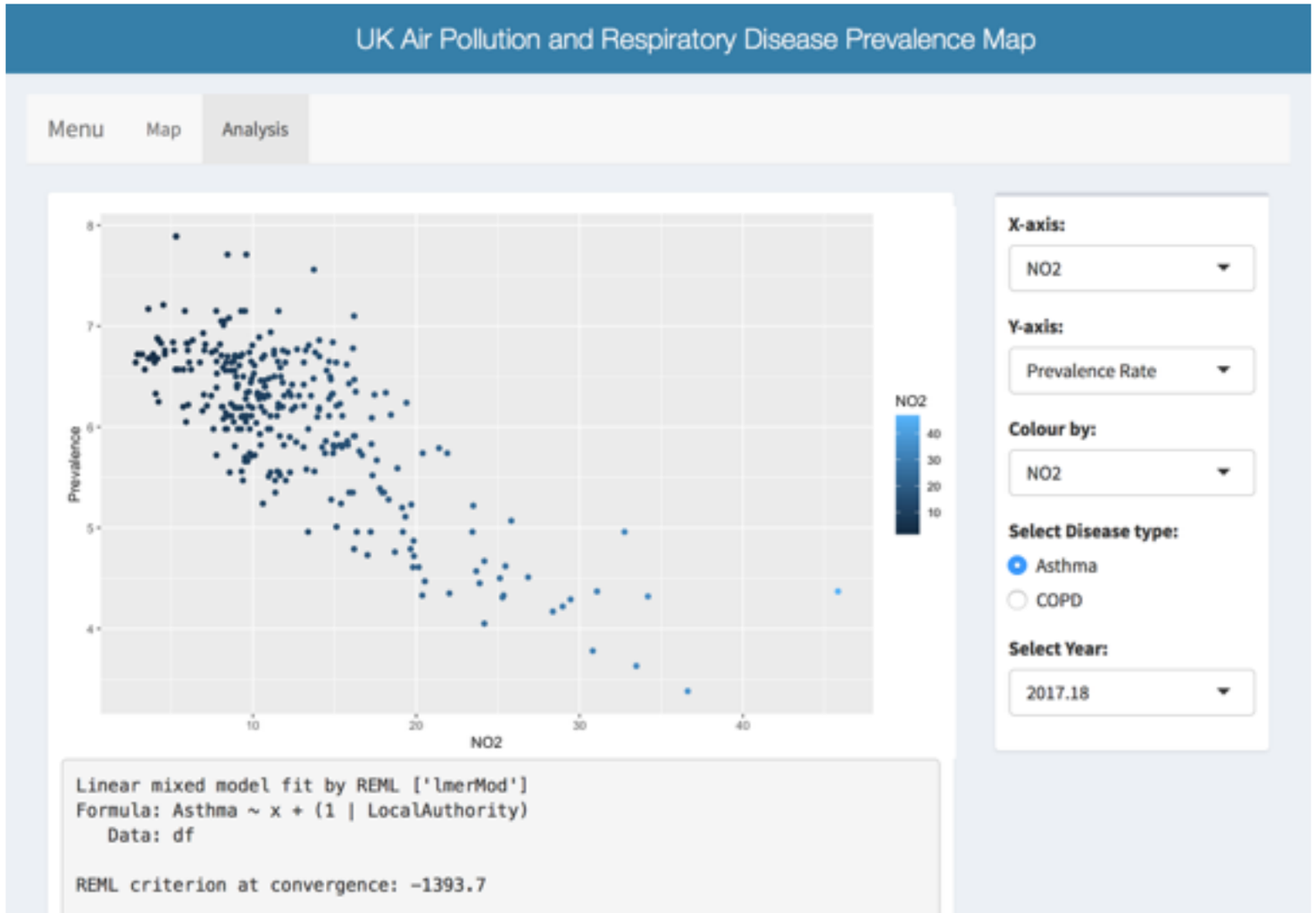
```
server<-
  function(input, output, session){


    getDataSetdisease <- reactive({

      #data_pg_df[!is.na(data_pg_df$Region) & data_pg_df$Region == input$regionInput, ]

      req(input$dataDisease)
      req(input$dataYear)
      datasubset1 <- full_shp[!is.na(full_shp$Year)&full_shp$Year==input$dataYear&
                                !is.na(full_shp$Disease)&full_shp$Disease==input$dataDisease&
                                !is.na(full_shp$PlltntT)&full_shp$PlltntT==input$dataPollu,]
      datasubset1

    })

    getDataSetair <- reactive({
      req(input$dataPollu)
      datasubset2 <- air[air$PlltntT==input$dataPollu,]
      datasubset2
                                        })
```

```
body<-
  dashboardBody(
    navbarPage(theme = "cerulean",
               "Menu",
               tabPanel("Map",
fluidRow(
  column(width = 9,
         box(width = NULL, solidHeader = TRUE,
             uiOutput("synced_maps")
             #leafletOutput("EnglandMap", height=400)
         ),
         box(width=NULL,
             dataTableOutput("results")
         )
  ),

  column(width=3,
         box(width=NULL,
             #uiOutput("yearSelect"),
             selectInput(inputId = "dataYear",
                         label = "Select Year:",
                         choices = yer,
                         selected = yer[1]),

             radioButtons("dataPollu", "Select Pollutant Type",
                          choices = c("NO2" = "NO2",
                                      "NOx" = "NOx",
                                      "PM10" = "PM10",
                                      "PM2.5" = "PM25")),

             radioButtons(inputId = "dataDisease",
                          label = "Select Disease type:",
                          choices = c("Asthma", "COPD"),
                          selected = "Asthma")
```

ui + server = shiny app

Feature of this app

- Interactivity
- Synchronisation
- Geography
- Statistical analysis
- User interface design



Feature of this app

- Interactivity
- Synchronisation
- Geography
- Statistical analysis
- User interface design

```
column(width=3,
                box(width=NULL,
                   selectInput(inputId = "dataYear",
                               label = "Select Year:",
                               choices = yer,
                               selected = yer[1]),

                radioButtons("dataPollu", "Select Pollutant Type",
                        choices = c("NO2" = "NO2",
                                    "NOx" = "NOx",
                                    "PM10" = "PM10",
                                    "PM2.5" = "PM25")),

                radioButtons(inputId = "dataDisease",
                             label = "Select Disease type:",
                             choices = c("Asthma", "COPD"),
                             selected = "Asthma")

            )
        )
```

Interactive select choices for desiring value

Feature of this app

- Interactivity
- Synchronisation
- Geography
- Statistical analysis
- User interface design

```
m1 <-
    leaflet(full_shp,options = leafletOptions(preferCanvas = TRUE)) %>%
    addTiles() %>%
    addPolygons(fillColor = pal(theDatadisease$Prevlnc),
                        fillOpacity = 0.8,
                        color = "#BDBDC3",
                        weight = 2,
                        popup = qof_popup,
                        data=theDatadisease)%>%

 setView(mean(bounds[1,]),
              mean(bounds[2,]),
              zoom=5.5)

m2 <-
 leaflet(air,options = leafletOptions(preferCanvas = TRUE)) %>%
 addTiles() %>%
 addPolygons(fillColor = pal_air(theDataair$Vallue),
                     fillOpacity = 0.8,
                     color = "#525266",
                     weight = 2,
                     popup = air_popup,
                    data=theDataair)%>%

 setView(mean(bounds[1,]),
              mean(bounds[2,]),
              zoom=5.5)

sync(m1, m2)
```

Create maps and synchronise two maps when click an area in a map

Feature of this app

- Interactivity
- Synchronisation
- Geography
- Statistical analysis
- User interface design

```
# Create scatterplot to see the correlation
output$scatterplot <- renderPlot({
 ggplot(data = disease_subset(),
        aes_string(x = input$x, y = input$y, color = input$z)) +
        geom_point()


})

# Create random effect model output
output$summary <- renderPrint({

 df<-
  df %>%
  spread(key = Disease, value = Prevalence)

 x <- df %>% pull(input$x)

 if (input$selected_type == "COPD") {
  print(summary(lmer(COPD ~ x + (1|LocalAuthority), data = df)))
 } else {
  print(summary(lmer(Asthma ~ x + (1|LocalAuthority), data = df)))
 }

})
```

Establish random effect model and scatter plot to see the correlation

Feature of this app

- Interactivity
- Synchronisation
- Geography
- Statistical analysis
- User interface design

```
# Select variable for colour
selectInput(inputId = "z",
            label = "Colour by:",
            choices = c("NO2"="NO2",
                        "NOx"="NOx",
                        "PM10"="PM10",
                        "PM2.5"="PM25"),
            selected = "Year"),

# Select disease type
radioButtons(inputId = "selected_type",
             label = "Select Disease type:",
             choices = c("Asthma", "COPD"),
             selected = "Asthma")
```

Create and arrange the interface !

What I’ve learned from this app

- Spatial mapping in R
- The basic knowledge of GIS
- How to visualise the statistics
- The things you need to know before app creation: the needs!
- The most important things is to keep learning