

BINF 5003: Data Mining, Modeling, and Biostatistics

Week 13

Module 7 – Interactive Data Visualization

Overview

- Rather than outputting all views of a complex dataset, interactive presentations can allow viewers or collaborators to access and explore the aspects of the data that interest them
- Anatomy of a Shiny Web App

Ontario COVID reporting – Aug 8, 2023

Table 2. Number and percentage of cases by Pango lineage and week, representative surveillance, Ontario, June 25 to July 22, 2023

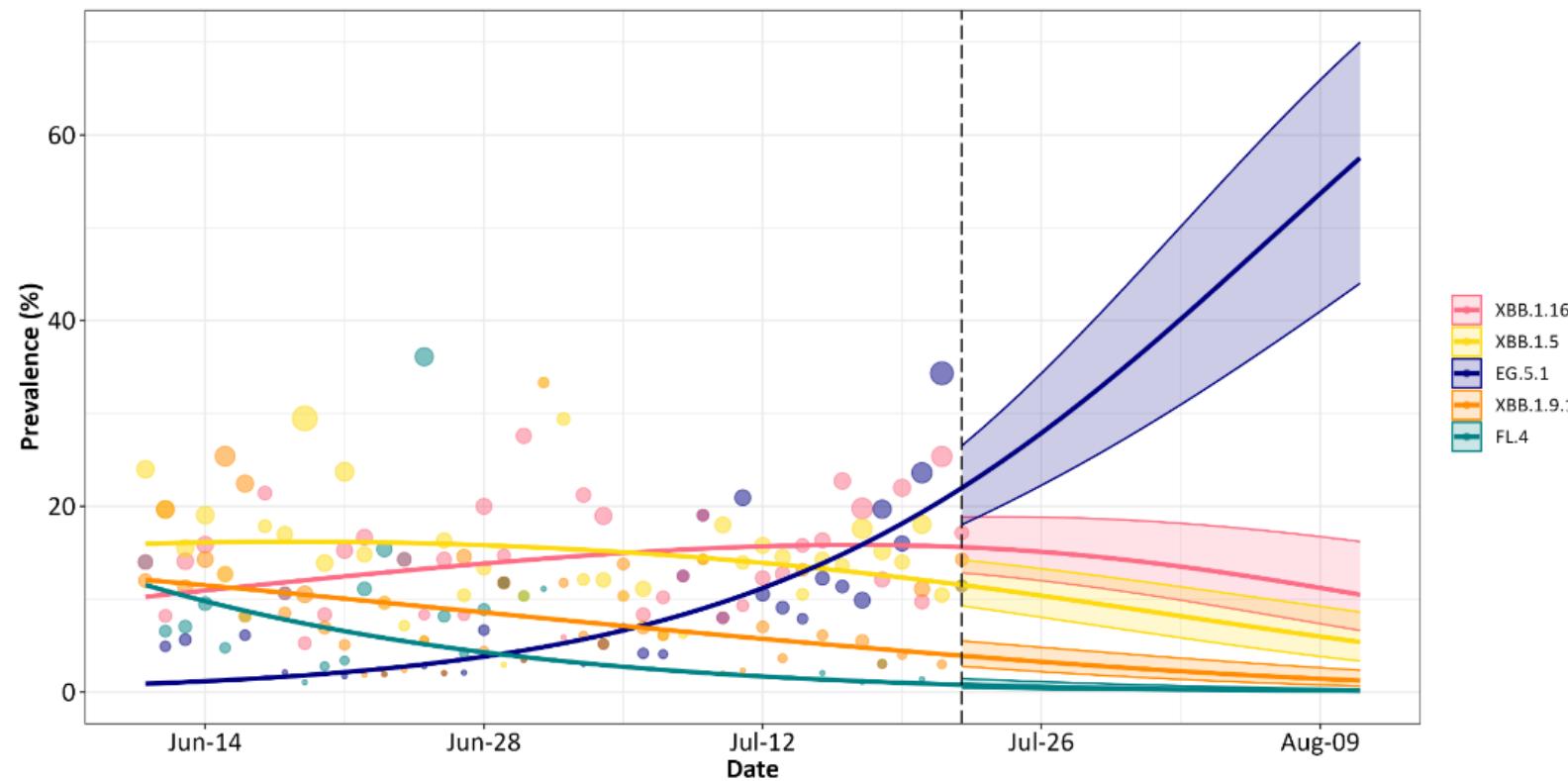
Pango lineage	Week 26 (June 25 - July 1)	Week 27 (July 2 - July 8)	Week 28 (July 9 - July 15)	Week 29 (July 16 - July 22)	Total (June 25 - July 22)
XBB.1.16	39 (15.6%)	37 (12.8%)	40 (12.8%)	77 (18.1%)	193 (15.1%)
XBB.1.5	28 (11.2%)	32 (11.0%)	46 (14.7%)	63 (14.8%)	169 (13.2%)
EG.5.1	11 (4.4%)	15 (5.2%)	37 (11.9%)	79 (18.6%)	142 (11.1%)
XBB.1.9.1	17 (6.8%)	19 (6.6%)	19 (6.1%)	24 (5.6%)	79 (6.2%)
XBB.1.22.1	0 (0.0%)	11 (3.8%)	27 (8.7%)	15 (3.5%)	53 (4.2%)
FL.4	31 (12.4%)	0 (0.0%)	1 (0.3%)	4 (0.9%)	36 (2.8%)
FL.10	1 (0.4%)	1 (0.3%)	5 (1.6%)	24 (5.6%)	31 (2.4%)
XBB.1.19.1	2 (0.8%)	19 (6.6%)	8 (2.6%)	2 (0.5%)	31 (2.4%)
XBB.1.5.10	3 (1.2%)	14 (4.8%)	7 (2.2%)	7 (1.6%)	31 (2.4%)
FD.1.1	8 (3.2%)	8 (2.8%)	3 (1.0%)	11 (2.6%)	30 (2.3%)
XBB.1.5.55	0 (0.0%)	14 (4.8%)	5 (1.6%)	8 (1.9%)	27 (2.1%)
XBB.1.16.1	6 (2.4%)	5 (1.7%)	7 (2.2%)	8 (1.9%)	26 (2.0%)
FL.1	1 (0.4%)	12 (4.1%)	11 (3.5%)	1 (0.2%)	25 (2.0%)
FL.2	10 (4.0%)	4 (1.4%)	2 (0.6%)	6 (1.4%)	22 (1.7%)
XBB.2.3.3	5 (2.0%)	7 (2.4%)	4 (1.3%)	5 (1.2%)	21 (1.6%)
XBB.1.5.59	4 (1.6%)	5 (1.7%)	5 (1.6%)	6 (1.4%)	20 (1.6%)
Other recombinant	77 (30.8%)	74 (25.5%)	77 (24.7%)	83 (19.5%)	311 (24.4%)
Other Omicron	7 (2.8%)	13 (4.5%)	8 (2.6%)	2 (0.5%)	30 (2.3%)
Total sequenced	250 (100%)	290 (100%)	312 (100%)	425 (100%)	1,277 (100%)

Note: Includes the most prevalent lineages detected in the past month. Details on the proportion of eligible samples sequenced by the OCGN can be found in the technical notes. Week was assigned based on the earliest date available for the sample. Not all sequencing and bioinformatics analyses for recent weeks were complete at the time of data extraction. Case counts for these weeks may increase in subsequent reports.

Data sources: Ontario Health Data Platform - Public Health Analytic Environment (OHDP-PHAE)

Ontario COVID reporting – Aug 8, 2023

Figure 2. Estimated daily prevalence (%) by Pango lineage, using Nowcast model, Ontario, June 11, 2023 to August 12, 2023



What do we like about this data?

Are there different views that you would like to see?

Note: Each curve represents the estimated prevalence of a given lineage from Nowcast modelling, which uses six weeks of daily representative surveillance data in a multinomial logistic regression. Each set of dots represents the observed daily prevalence of a given lineage, while their size represents the relative number of samples. The vertical dashed line indicates the most recent day of data, after which projected Nowcast prevalence estimates are presented with their 95% confidence intervals. The vertical grey lines indicate the mid-point of the week. Lineages with at least 21 days of non-zero case counts were included in the model and lineages that did not have at least 21 days of non-zero case counts were included but not shown. Figure includes all lineages with at least one day of an estimated prevalence of 5% or greater during the 12 week period (six observed and six projected). Only three weeks of projected data are shown. Prevalence projections may be overestimated for emerging lineages.

Data sources: Ontario Health Data Platform - Public Health Analytic Environment (OHDP-PHAE)

<https://www.publichealthontario.ca/-/media/documents/ncov/epi/covid-19-sars-cov2-whole-genome-sequencing-epi-summary.pdf>

Ontario COVID reporting – Aug 8, 2023

Strengths

- Well organized data table
 - Some basic stats/percentages calculated
- Clearly formatted graph

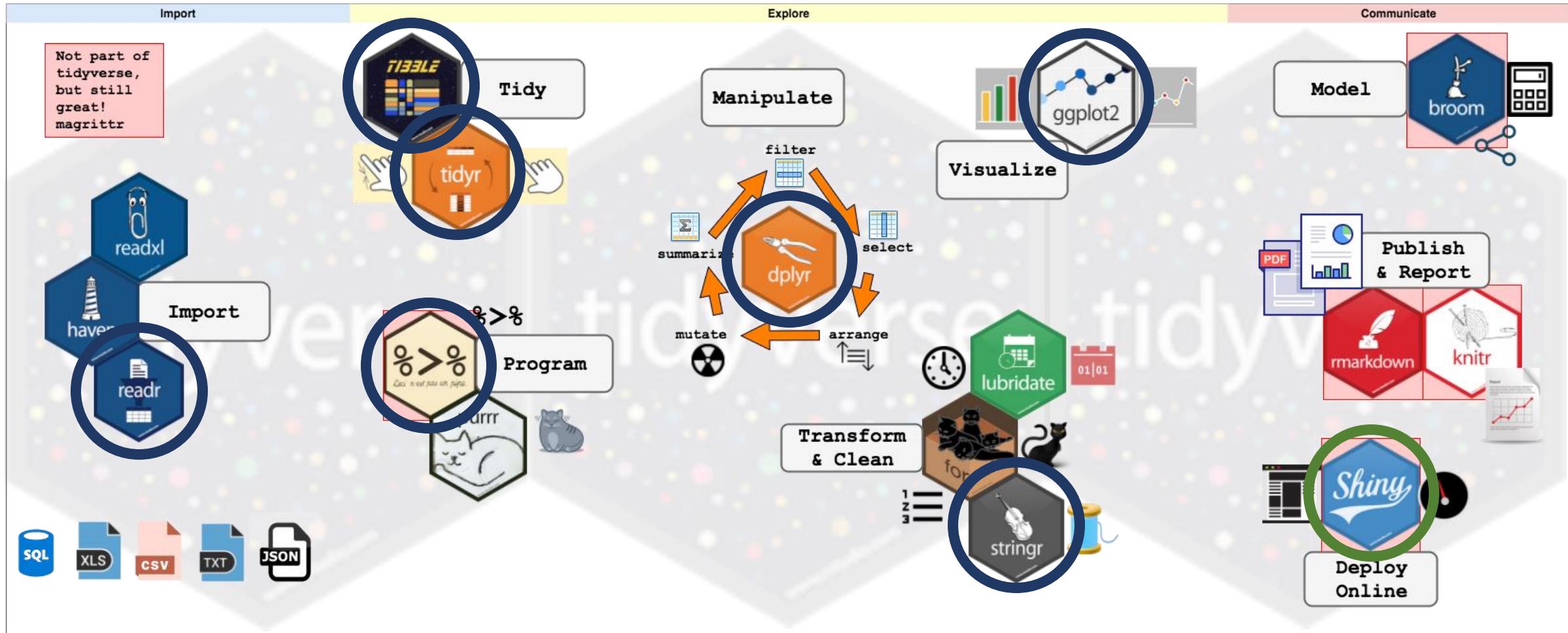
Other views

- Changing which strains are being
- Filter by increasing or decreasing trends

Interactive app for linear regression

- <https://shiny.posit.co/r/gallery/education/didacting-modeling/>

Last piece of the puzzle



Increasing accessibility to your data

- Effective communication of your results for publications and presentations
- Promote and facilitate collaborations with team members who do not code

Shiny apps

- Define elements that allow the user to interact with
- To share:
 - Share code and dataset for others to launch
 - Would likely require some familiarity with R
 - Deploy the app and host it online for others
 - They would not have access to the raw data or underlying code

Structural of app.R

- Static elements that are run only once
- UI to define elements that the user sees, how they interact with the code, and how the results are returned
- Server to execute the actual computation using the user input

```
# Load libraries
library(shiny)
library(DT)
library(tidyverse)
library(ggplot2)
library(gridExtra)
library(cowplot)

# Move the dataset into the same folder as
load("breastCancerData.csv")
```

```
# Define UI for application that draws a histogram
ui <- fluidPage(
  input
```

```
)
```

```
# Define server logic required to draw a histogram
server <- function(input, output) {
  output
}
```

```
# Run the application
shinyApp(ui = ui, server = server)
```

Structural of app.R

- Take a custom input from the UI to be used in the computation at the server section
 - Just like writing generic code!

```
# Load libraries
library(shiny)
library(DT)
library(tidyverse)
library(ggplot2)
library(gridExtra)
library(cowplot)

# Move the dataset into the same folder as
load("breastCancerData.csv")
```

```
# Define UI for application that draws a histogram
ui <- fluidPage(
  input
```

```
)
```

```
# Define server logic required to draw a histogram
server <- function(input, output) {
  output
}
```

```
# Run the application
shinyApp(ui = ui, server = server)
```

Refresher: Generic Code

```
# Subset out data for one cut, display the cut that is currently being
# analyzed

selected_cut <- "Ideal"

message("Fitting linear model for diamonds with a(n) ", selected_cut, "
cut -----")

# Subset out data for one cut

cut_subset <- filter(diamonds, cut == selected_cut) # Edit the cut
dim(cut_subset)

# Fit the model

cut_subset_fit <- lm(price_log2 ~ carat_log2, data = cut_subset)

# Display the results

summary(cut_subset_fit)
```

Circular format

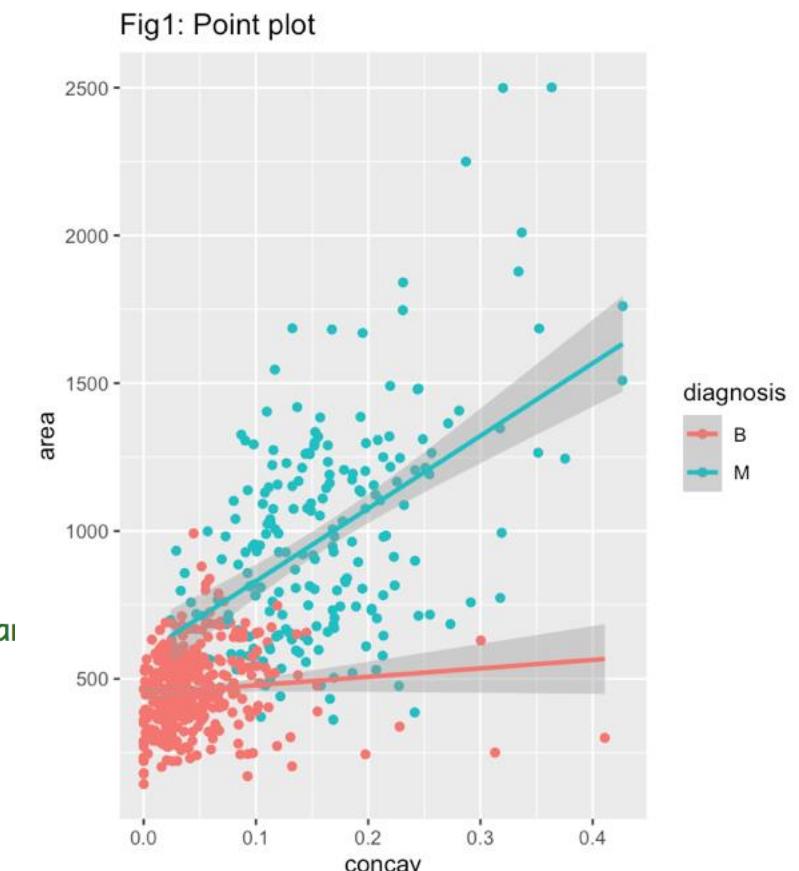
UI

```
selectInput(  
  inputId = "colorVariable",  
  label = "Please select variable to color point plot Fig1",  
  choices = colnames(bCancer),  
  selected = colnames(bCancer)[3],  
  multiple = F  
  
  plotOutput("pointPlot"))  
  
output$pointPlot <- renderPlot({  
  xVariable = "concav" # Create these as constants right now  
  yVariable = "area"  
  
  bCancer %>%  
    ggplot(aes_string(x = xVariable, y = yVariable, color = input$colorVariable)) + # Change  
    the input value  
    geom_point() +  
    geom_smooth(method = "lm") +  
    ggtitle("Fig1: Point plot") # Add title to identify which input it matches to  
})
```

Server

Please select variable to color point plot
Fig1

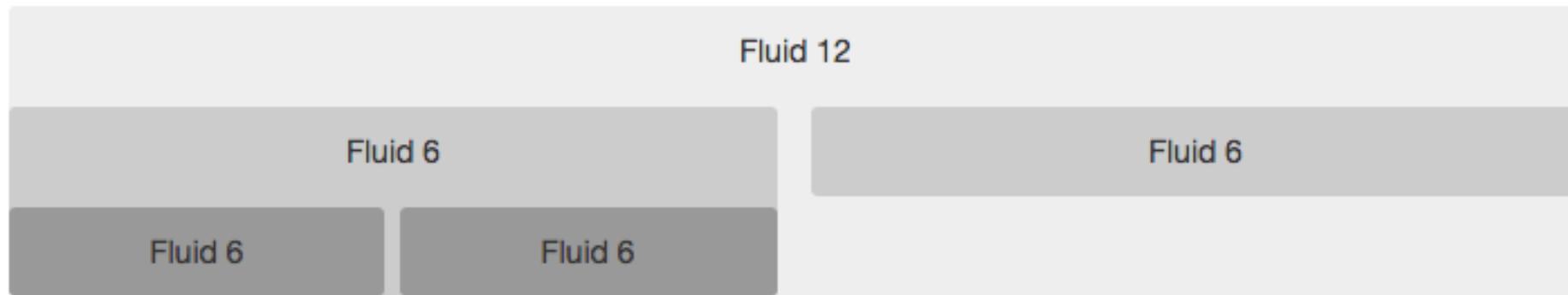
diagnosis



Keep the file name app.R

- Not a markdown document, no code chunks
 - Recommend troubleshooting your generic code in a markdown document first
- Any notes will need to be made with hashtag comments
 - Very important to keep the code organized!

Layout is defined by rows



```
ui <- fluidPage(  
  fluidRow(  
    column(6,  
          ),  
    column(6,  
          ))  
  )
```

Wrap-up

- Revisiting scientific communication and visualization
- Shiny Web App have a UI and a Server portion
 - Always start with a foundation of working code for one instance
 - Convert it to be generalized – this will help A LOT when reshaping it into the format required for Shiny Web Apps