

TB Burden Data Analysis and Visualization with Rshiny Application

Christina Lu Jin | 03.30.2021

RShiny App 1 – Exploration of Cases and Rates by Region and Country

1.1 Application Overview

The following image showcases the main interface of the application, which includes two parts – a side bar and a main panel, as well as a title on top. The side bar on the left is a user defined section, and the main panel on the right is an interactive geo-map, which highlights the aggregated number of TB cases and rates at each country by bubbles with different colors and sizes. Each color represents a different region, users could easily figure out which WHO region does each dot (country) belong to by reading the legend at the bottom right. Size of the bubble is associated with the number of cases / rates, where the higher the cases are at a certain country, the bigger this bubble would be. The following image is the initial set up, so whenever you run the App, this is always the page that you will perceive first.

TB Burden Exploratory App - by Region and Country



This second image shows , with the parameter changed to prevalence rate per 100k population, and the bubble size change to 100. Comparing both images, India and China (big green bubble in image 1) didn't ranked high in prevalence rate (per 100k population) as some other countries in Africa, this might be due to the prominent population of both countries.



1.2 Application Function Introduction

The function of this application is to help users explore both quantitative and qualitative statistics from the dataset by visually represent the chosen metric on a geo-map. The drop-down menu, at the user defined input section, contains 9 metrics of measurement, which allow users to choose the particular statistic they would want to explore.

All data metrics are aggregated by country (categorized by year). Prevalence indicates existing number of cases, incident indicates newly added cases, and mortality means number of death cases. Retrieved data includes TB records from 1990 to 2014

Select a measure metric:

Incidence

Total_Population
Prevalence
Incidence
Death_no_HIV
Death_HIV
Prevalence_Rate_per_100k
Incidence_Rate_per_100k
Mortality_no_HIV_Rate_per_100k

All data metrics are aggregated by country (categorized by year). Prevalence indicates existing number of cases, incident indicates newly added cases, and mortality means number of death cases. Retrieved data includes TB records from 1990 to 2014

Select a measure metric:

Mortality_no_HIV_Rate_per_100k

Prevalence
Incidence
Death_no_HIV
Death_HIV
Prevalence_Rate_per_100k
Incidence_Rate_per_100k
Mortality_no_HIV_Rate_per_100k
Mortality_HIV_Rate_per_100k

The slide bar below the drop-down menu gives user ability to universally adjust the size of the bubbles which will show up as on the map. Since the number of cases/rates are sometimes at huge differences among all the countries, this will result in some extremely large and tiny sizes of bubbles. The default value was pre-set to a fair size of 5, but being able to adjust the visual size of the bubbles at a scale from 1 to 10, gives user full control of the visibility. In the following cases, the slider was dragged to show bubbles in that particular view at a slightly smaller size of 3, and a slightly larger size of 8.

All data metrics are aggregated by country (categorized by year). Prevalence indicates existing number of cases, incident indicates newly added cases, and mortality means number of death cases. Retrieved data includes TB records from 1990 to 2014

Select a measure metric:

Total_Population

Adjust Bubble Size

1 3 10

1 2 3 4 5 6 7 8 9 10

Author: Christina Lu Jin

All data metrics are aggregated by country (categorized by year). Prevalence indicates existing number of cases, incident indicates newly added cases, and mortality means number of death cases. Retrieved data includes TB records from 1990 to 2014

Select a measure metric:

Total_Population

Adjust Bubble Size

1 8 10

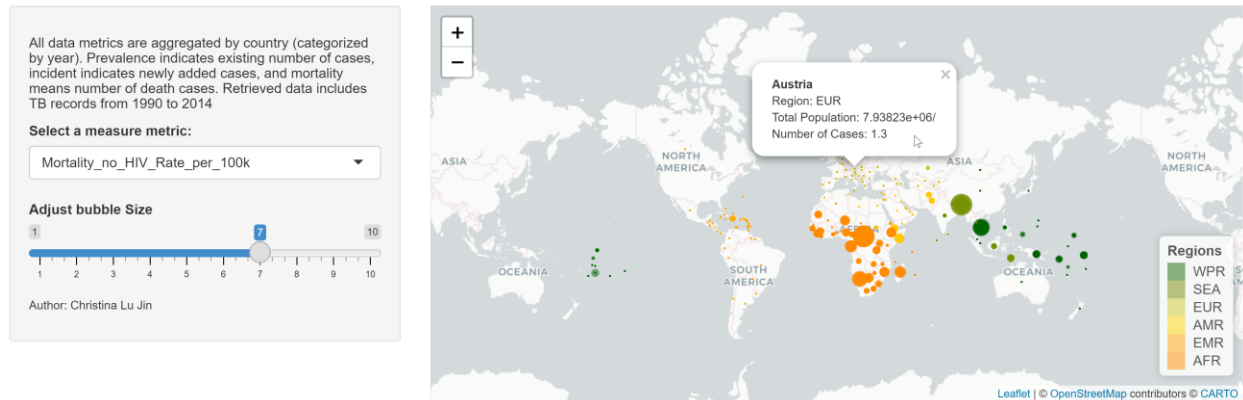
1 2 3 4 5 6 7 8 9 10

Author: Christina Lu Jin

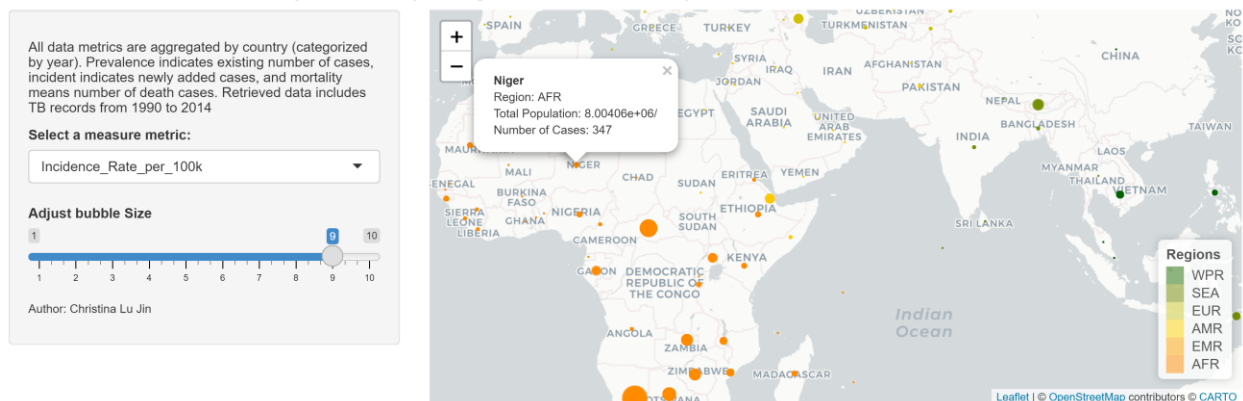
The geo-map is an interactive one, what's more powerful about it than a static map is that users could zoom in and out on the map to navigate to specific regions or countries, as well as gaining a big picture of the entire world. In addition, each bubble is a data point that includes the aggregated information of all data entries at that country, and users could retrieve an overview of these information by clicking on the desire bubble. A popup window will then appear near the bubble, which includes the name of the

country, the WHO region it belongs to, total population of this country, and the specific number for the selected statistic. The following are two examples both with the popup window, one looking at Austria with a zoomed-out view of the entire world map, and the other looking at Niger with a view zoomed in at African region.

TB Burden Exploratory App - by Region and Country



TB Burden Exploratory App - by Region and Country

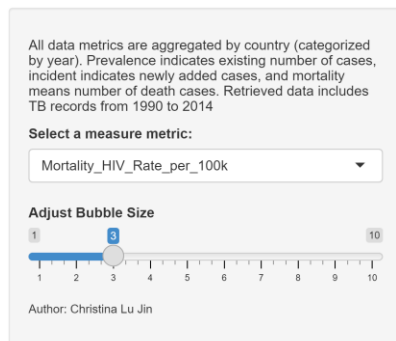
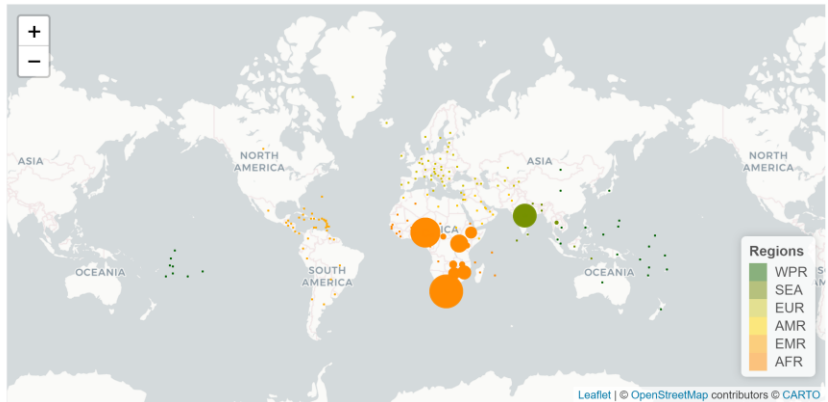
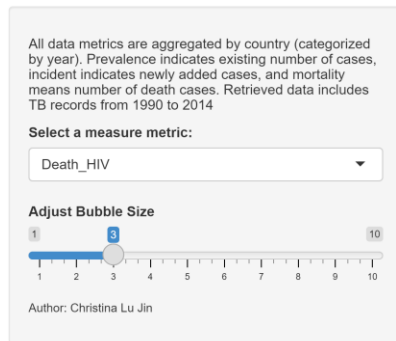


1.3 Application Interaction and Storytelling

This application can help users easily navigate through the dataset, dig for the valuable information, as well as propose and answer interesting business questions. Some of my findings through using this app are as follows:

Looking at the first image, death TB cases with positive HIV are particularly high at multiple African countries and India. But when moving on to look at the TB mortality rate with positive HIV, rate for India had dropped down by a lot, while the rate for African region are still remaining at a quite high level. This is probably due to the prominent population of India, thus, although there are tremendous amount of death cases with HIV, the base population is large enough to dilute the mortality rate of TBHIV. And the significantly high TBHIV mortality rate at many African countries indicates that HIV was a major issue back then at African region. And the infection of HIV had increased the chance of getting TB in some degree. If African region were to find ways to

control/prevent the infection of HIV, it could possibly reduce the number of cases / rates of TB in those areas. But further investigations are needed with other applications.



(end of the first application)

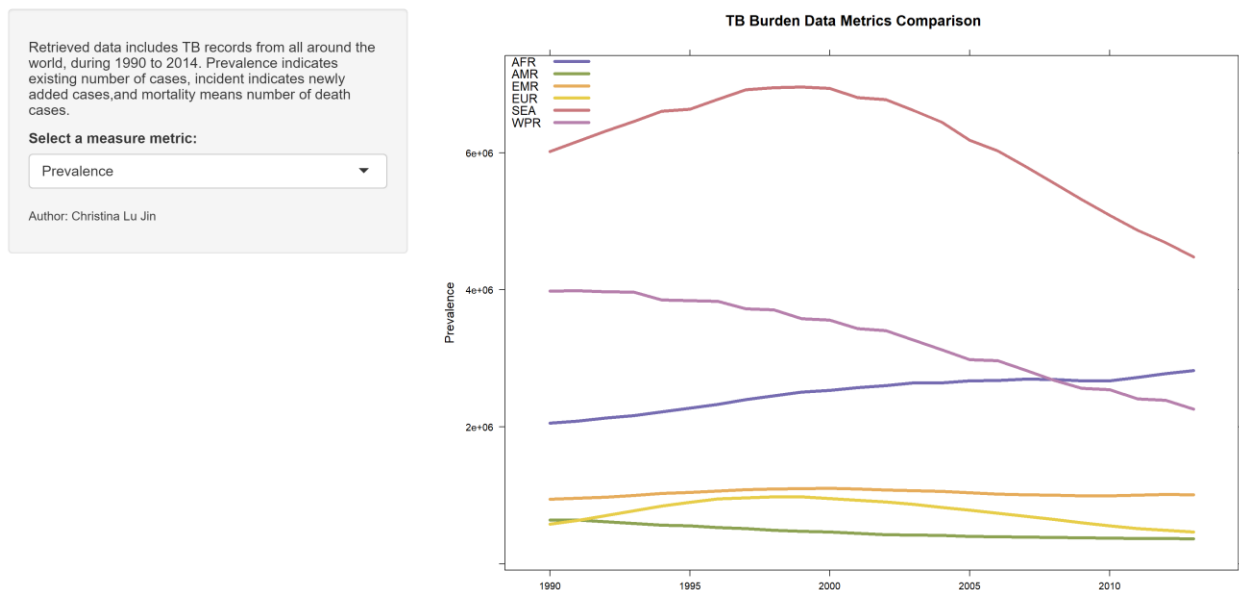
RShiny App 2 – Exploration of Cases and Rates Trend by Year and Region

2.1 Application Overview

The following image showcases the main interface of the second application, similar to the first app, it also includes a user input side bar and a main panel. The main panel, in this case, is a line chart which tracks the changes and trend of each measurement throughout years from 1990 to 2014.

Dataset was treated for this application by aggregating all entries by their region and year. So each plot point on the chart indicates the number for cases / rates of the y-axis metrics for that specific year, and color coded by different regions. By connecting these plot points, we get the trend lines, as displayed below, showing the changes over time at each region. The x-axis, in this case, remains same to be from year 1990 to 2014.

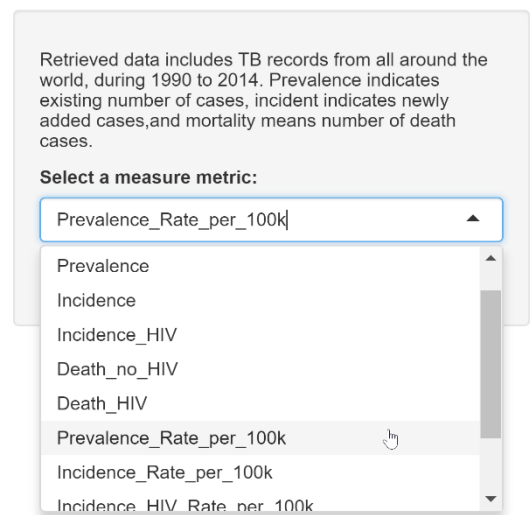
TB Burden Trend App - by year



2.2 Application Function Introduction

The function of this application was made easy to assist users to explore how the quantitative statistics change over time, by plotting out trending lines. The user-defined input section of this app is pretty straight forward, with only one drop-down menu which contains 9 metrics of measurement, allowing users to choose particular statistics they would want to explore.

The selected metric defines the y-axis, thus the plot will be updated with the change of selections. X-axis will stay the same 1990 - 2014 as mentioned before. Each line represents a single region, which are color coded as indicated in the legend.

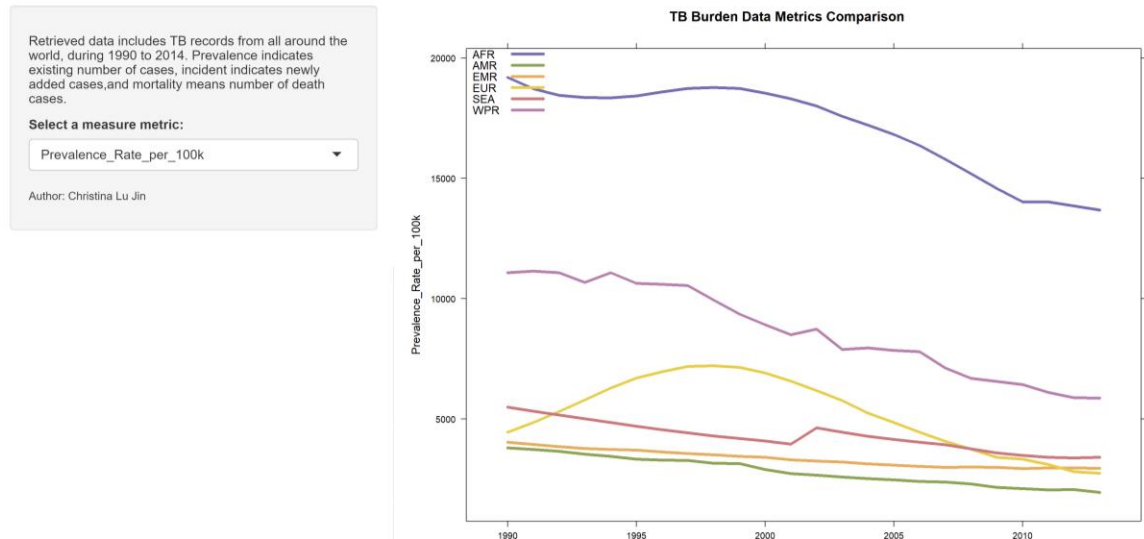


2.3 Application Interaction and Storytelling

This application can help users easily navigate through the dataset, explore patterns and relationships, digest valuable information, as well as propose and answer interesting business questions. Some of my findings through using this app are as follows:

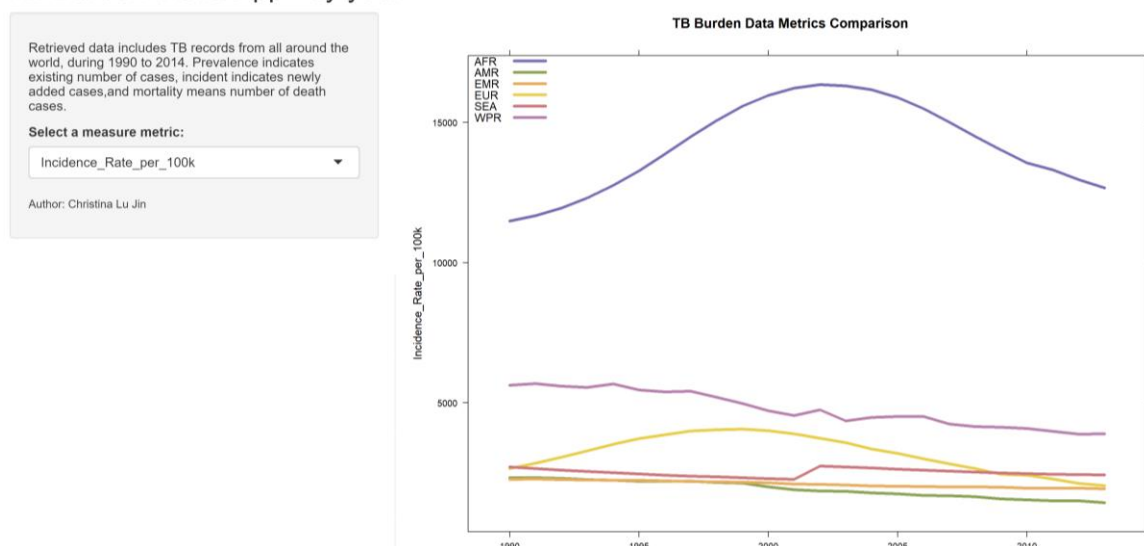
- 1) The first image looks at the prevalence rate per 100k population for different WHO regions. The rate flatly decreases while having a bit of fluctuation for most regions. European is the only region was increasing at first and started to decrease at around year 1997.

TB Burden Trend App - by year



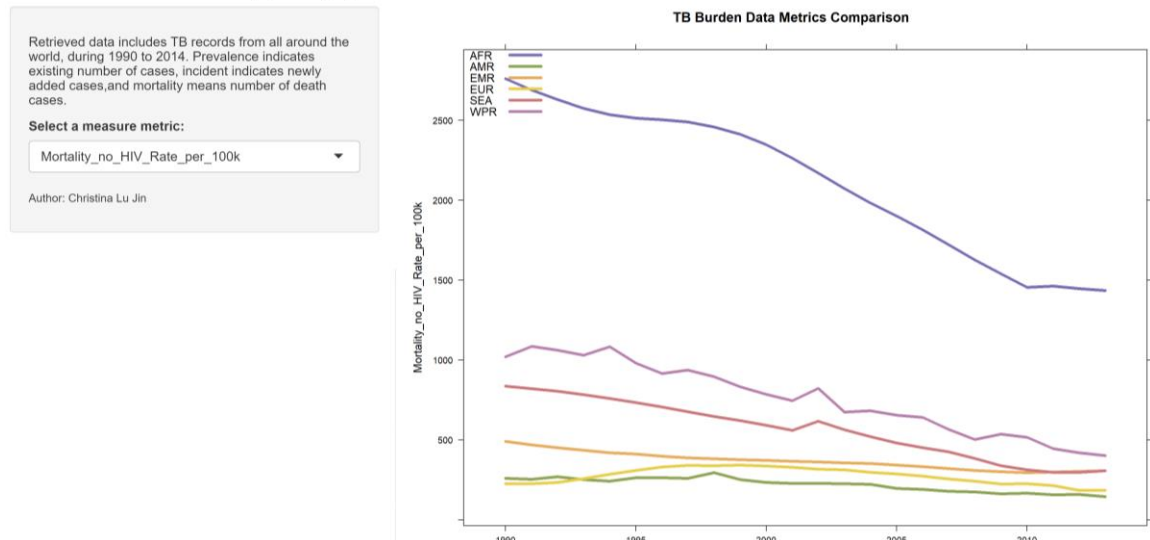
- 2) The second image looks at the incidence rate per 100k population for different WHO regions. The pattern for almost all regions are quite similar to the prevalence rate from last image, except African. African had a steep upward trend from year 1990 to around 2002, and started to fall back to a downward trend after that. This indicates the newly added case was increasing at a high rate at African until 2002, and got under control and started decreasing after that.

TB Burden Trend App - by year



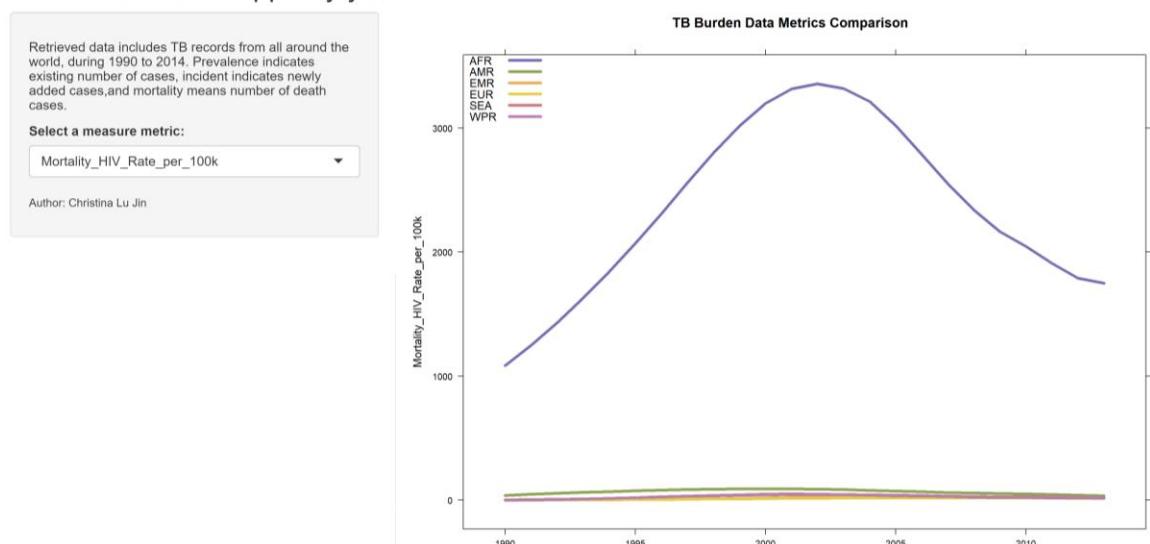
- 3) The third image looks at the TB mortality rate of those who doesn't have HIV per 100k population for different WHO regions. All downward trends indicate that there were less HIV-negative people dying from TB each year back then. African region, particularly, had decreases in negative-HIV death cases at a very high rate comparing to other regions. But since it started with a more significant mortality rate, it still ended up with a much higher rate than others at the end of 2014.

TB Burden Trend App - by year



- 4) This last image looks at the TB mortality rate of those who were also HIV infected per 100k population for different WHO regions. All 5 regions are cluster at the very bottom of the graph, this mean TB-HIV was not a huge concern at those regions. Whilst, African had a tremendous TB-HIV mortality rate, and it was surging from 1990 to around 2002, we could possibly imagine how severe the HIV issue was at African back then. Good thing is the line started to curve down to a downward trend since 2002 and kept decreasing. But again, due to the significant base rate and the accumulation of the 12-year period increase, TB-HIV mortality rate was still super high by the end of 2014 and even higher than where it started off from, at 1990.

TB Burden Trend App - by year



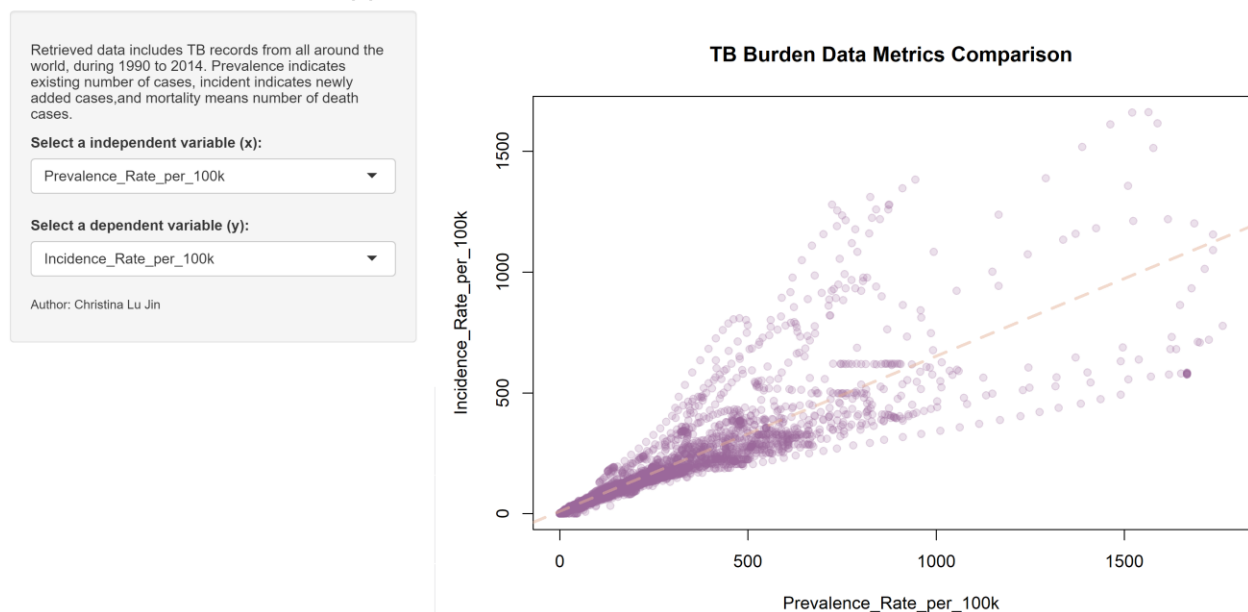
RShiny App 3 – Exploration Correlations Between Different Metrics

3.1 Application Overview

The following image showcases the main interface of the third application. The consistency of the interface layout was kept for all three apps, thus it also has a user input side bar and a main panel. The main panel in this app is the combination of a scatter plot and a regression line.

No special treatment was done towards the dataset for this app, therefore, all the data point in this entire date frame was captured in this scatter plot, each dot links to a single data point. All scatters had a 50% transparency, so if there is a cluster of scatters in the plot, app users would be able to tell by the darker color of that area.

TB Burden Correlation App - between different metrics



3.2 Application Function Introduction

The function of this application was made easy to assist users to explore the possible correlation between each quantitative statistic. The user-defined input section of this app is also quite straight forward, with two identical drop-down menus, one for the x-axis input as independent variable, and the other for the y-axis as dependent variable. Each drop-down contains 9 metrics of measurement, allowing users to choose any two statistics they would want to compare and explore.

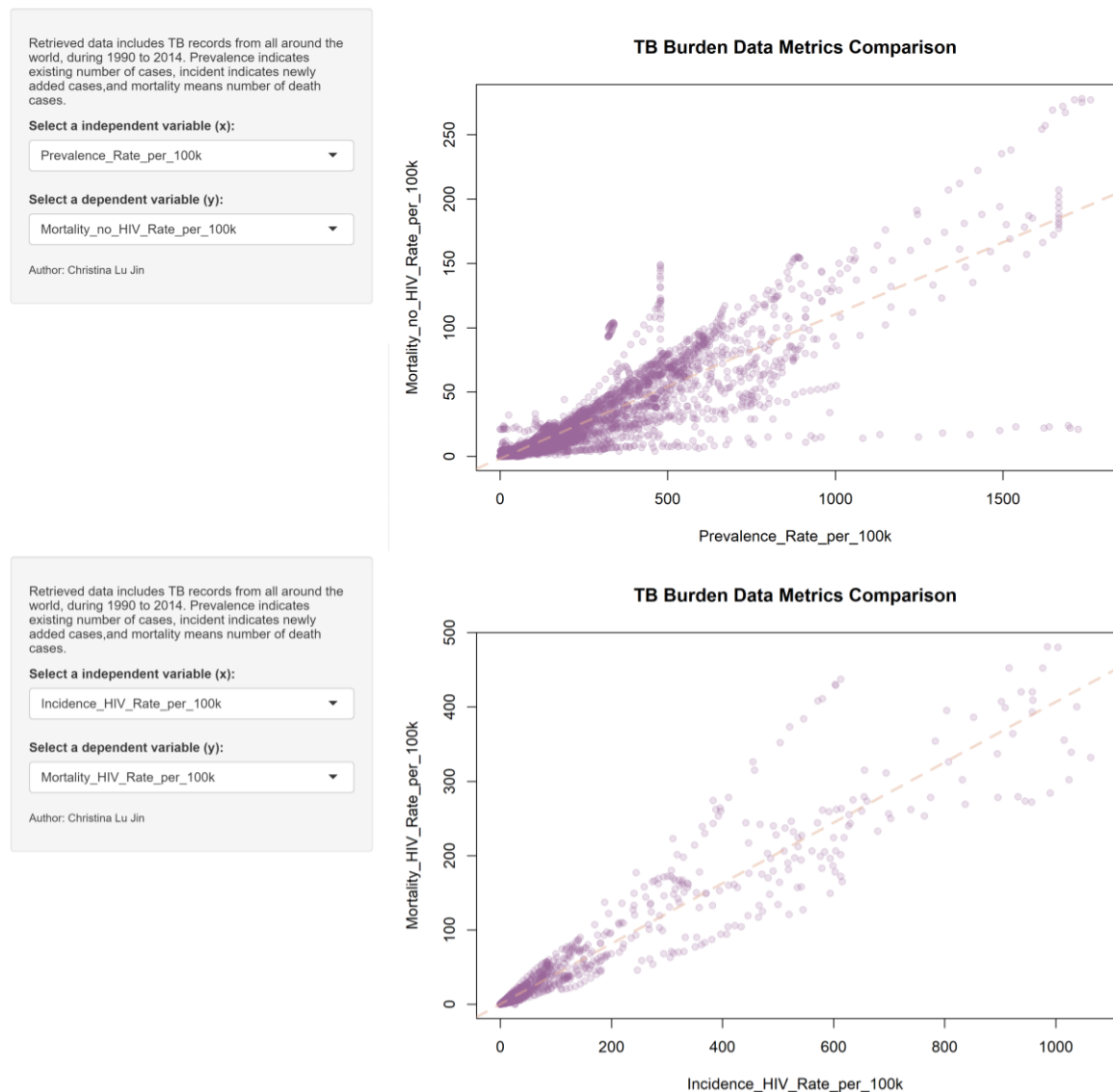
Two screenshots of the app's input interface are shown. The left screenshot shows the "Select a independent variable (x):" dropdown menu with "Incidence_HIV" selected. The right screenshot shows the "Select a dependent variable (y):" dropdown menu with "Prevalence" selected. Both screenshots include the same explanatory text and author information as the main interface.

3.3 Application Interaction and Storytelling

This application can help users easily explore patterns and possible correlations, digest valuable information, as well as propose and answer interesting business questions. Some of my findings through using this app are as follows:

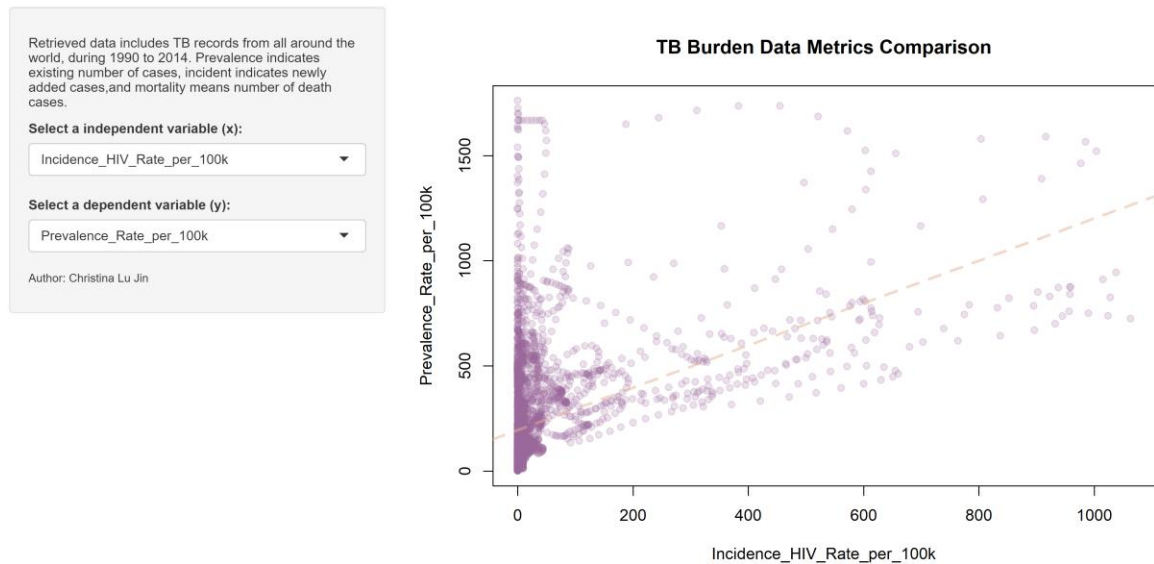
- 1) The following two images were set to discover relationships between prevalence rate vs TB-no-HIV mortality rate, and TB-HIV incidence rate vs TB-HIV mortality rate. We could see from the graph that both plots show some sort of linear correlation, the first graph has a bit of points that doesn't locate near the linear regression line, but the linear regression line for the second graph is almost perfectly fitted in the plot. These explain that there are relative strong positive correlations between both pair of metrics. Therefore, we could infer that the higher the prevalence rate is, the higher the mortality rate would be; the higher TB-HIV incidence rate is, the higher the TB-HIV mortality rate would be. Both correlations make perfect sense.

TB Burden Correlation App - between different metrics

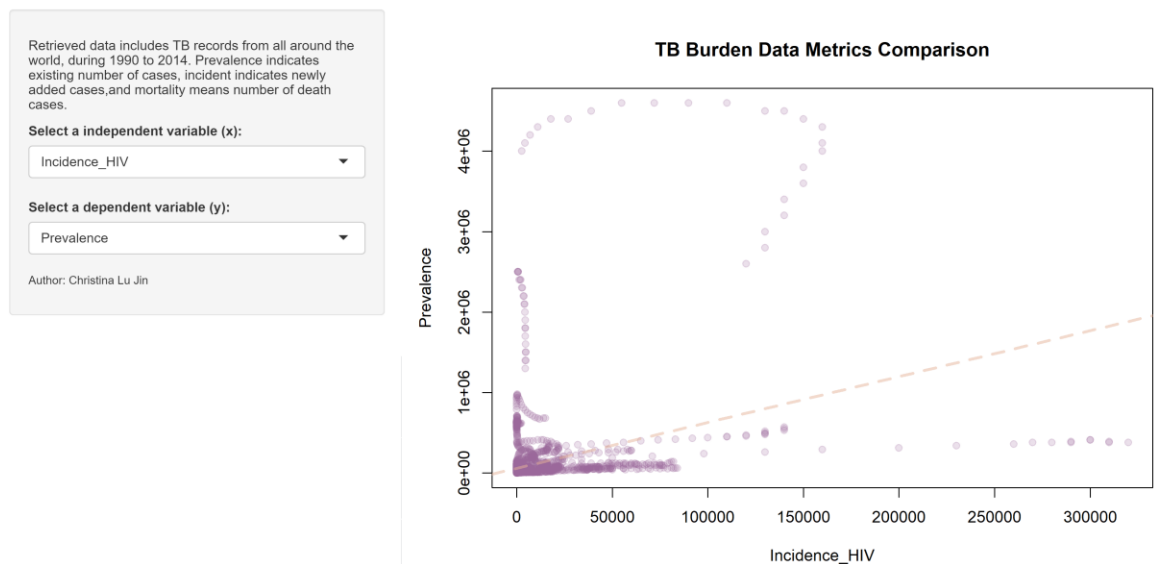


- 2) The next two images were set to discover relationships between TB-HIV incidence rate vs prevalence rate, and TB-HIV incidence cases vs prevalence cases. Surprisingly, there are not any strong correlation had been detected from the graph, points are all over the place and with no special pattern. Before plotting out, I would imagine that TB-HIV incidence would somehow affect the prevalence case, and I would assume the two graphs to be pretty similar. But in fact, changes in TB-HIV incidence rate doesn't impact prevalence rate much, and although both graphs share some similarities, yet they are still quite different. When this a second thought, I think possibilities could be although the new cases of TB-HIV increase, due to the increase in recovery or mortality rate, the number of aggregated patients (prevalence cases) stays the same.

TB Burden Correlation App - between different metrics




TB Burden Correlation App - between different metrics



References

- 1) Getting Started with Shiny. University of Virginia Library. Retrieved from <https://data.library.virginia.edu/getting-started-with-shiny/>
- 2) Katyal, V. Beginners guide to Bubble Map with Shiny (Jan 31, 2020). Retrieved from <https://www.r-bloggers.com/2020/01/beginners-guide-to-bubble-map-with-shiny/>
- 3) Gilbert, C. Make Your Own Interactive Map of COVID-19 Spread Using R Shiny. Presented by RockEDU Science Outreach. Retrieved from <https://www.youtube.com/watch?v=eIpIL6y1oQQ&t=3878s>
- 4) A&G Statworks. Creating interactive maps in R. Retrieve from <https://www.youtube.com/watch?v=dx3khWsUO1Y&t=48s>
- 5) Taking control of colors in lattice. Retrieved from https://www.stat.ubc.ca/~jenny/STAT545A/block16_colorsLattice.html
- 6) Grouping data points within a scatter plot. Retrieved from https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781849513067/3/ch03lvl1sec02/grouping-data-points-within-a-scatter-plot
- 7) Plotting multiple lines on a single graph using Shiny and ggplot2. <https://stackoverflow.com/questions/45286622/plotting-multiple-lines-on-a-single-graph-using-shiny-and-ggplot2>

Appendix 1 – App 1 R Code

```
#####  
##  
## TB Burden Exploratory App 1 - by Region & Country ##  
## ALY6070 - M4 Assignment | Christina Lu Jin ##  
##  
#####  
  
## install packages and import library   
#####  
  
# import data  
TB_data <- as.data.frame(read.csv("TB_Burden_Country.csv"))  
head(TB_data)  
# get country data with latitude and longitude info  
country_data <- as.data.frame(read.csv("countries.csv"))  
#View(country_data)  
  
# Data prep, reshape and clean up  
#TB_df <- TB_data[,c(1,5:7,45,11,31,41,35,18,24,8,28,38,15,21)]  
#colnames(TB_df) <- c(..,"Case_Detection_Rate",..,"Incident_HIV_Percent")  
TB_df <- TB_data[,c(1,5:7,11,31,41,18,24,8,28,38,15,21)]  
names(TB_df)  
colnames(TB_df) <- c("Country", "Region", "Year", "Total_Population", "Prevalence",  
  "Incidence", "Incidence_HIV", "Death_no_HIV", "Death_HIV",  
  "Prevalence_Rate_per_100k", "Incidence_Rate_per_100k",  
  "Incidence_HIV_Rate_per_100k",  
  "Mortality_no_HIV_Rate_per_100k",  
  "Mortality_HIV_Rate_per_100k")  
  
TB_df_agg <- aggregate(TB_df[,c(4:14)],  
  by = list(Country = TB_df$Country,  
    Region = TB_df$Region, Year = TB_df$Year),  
  FUN = sum)  
TB_df_merge <- merge(TB_df_agg, country_data[,c(2:4)],  
  by.x=c("Country"), by.y=c("name"))  
TB_df_metrics <- TB_df_merge[,c(1:6,8:11,13,14)]  
TB_df_metrics_num <- TB_df_metrics[,c(4:12)]  
  
# Define UI for application that plots a geo-map  
ui <- fluidPage(  
  titlePanel("TB Burden Exploratory App - by Region and Country"),  
  sidebarLayout(  
    sidebarPanel(  
      h5("All data metrics are aggregated by country (categorized by year)  
        Prevalence indicates existing number of cases,  
        incident indicates newly added cases,  
        and mortality means number of death cases.  
        Retrieved data includes TB records from 1990 to 2014"),  
      selectInput(inputId = "metrics",  
        label = "Adjust Bubble Size",  
        min = 1, max = 10, step = 1, value = 5),  
    )  
  )  
)
```

```

    h6("Author: Christina Lu Jin"),
  ),

  mainPanel(
    leafletOutput(outputId = "map")
  )
)

# Define server logic required to plot a geo-map
server <- function(input, output) {

  output$map <- renderLeaflet({

    # Set color palette
    colors <- c("darkorange", "gold", "darkgreen")
    pal <- colorFactor(colors, domain= TB_df_metrics_num$Region,
                       na.color="transparent")

    # Set popup label
    labels = sprintf(
      "<strong>%s</strong><br/>Region: %s<br/>
      Total Population: %g<br/>Number of Cases: %g",
      TB_df_metrics$Country, TB_df_metrics$Region,
      TB_df_metrics$Total_Population,
      TB_df_metrics[,input$metrics]) %>% lapply(htmltools::HTML)

    #initialize the leaflet object
    basemap = leaflet() %>%
      addProviderTiles(provider = "CartoDB.Positron") %>%
      addCircleMarkers(data=TB_df_merge, lat = ~TB_df_merge$latitude,
                      lng = ~TB_df_merge$longitude,
                      radius = ~(TB_df_metrics_num[,c(input$metrics)]/
                                sum(TB_df_metrics_num
                                    [,c(input$metrics)])
                                )*500*input$bubblesize,
                      weight=1, popup = labels,
                      color=FALSE, fillColor=~pal(Region),
                      fillOpacity = 0.5, stroke = FALSE) %>%
      addLegend("bottomright",
                #values = ~TB_df_metrics$Prevalence,
                colors = c("#1b6800", "#748b00", "#cec929",
                          "gold", "orange", "darkorange"),
                labels = c("WPR", "SEA", "EUR", "AMR", "EMR", "AFR"),
                title = "Regions",
                opacity = 0.5
              )
  })
}

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

```

Appendix 2 – App 2 R Code

```
#####  
##  
##          TB Burden Exploratory App 2 - by year          ##  
##      ALY6070 - M4 Assignment | Christina Lu Jin      ##  
##  
#####  
  
## install packages and import library   
#####  
  
# import data  
TB_data <- as.data.frame(read.csv("TB_Burden_Country.csv"))  
head(TB_data)  
  
# Data prep, reshape and clean up  
TB_df <- TB_data[,c(1,5:7,11,31,41,18,24,8,28,38,15,21)]  
colnames(TB_df) <- c("Country", "Region", "Year", "Total_Population", "Prevalence",  
                    "Incidence", "Incidence_HIV", "Death_no_HIV", "Death_HIV",  
                    "Prevalence_Rate_per_100k", "Incidence_Rate_per_100k",  
                    "Incidence_HIV_Rate_per_100k",  
                    "Mortality_no_HIV_Rate_per_100k",  
                    "Mortality_HIV_Rate_per_100k")  
TB_df_agg_rg <- aggregate(TB_df[,c(4:14)],  
                          by = list(Region = TB_df$Region, Year = TB_df$Year),  
                          FUN = sum)  
TB_df_agg_yr <- aggregate(TB_df[,c(4:14)],  
                          by = list(Year = TB_df$Year),  
                          FUN = sum)  
TB_df_metrics_rg <- TB_df_agg_rg[,c(3:13)]  
TB_df_metrics_yr <- TB_df_agg_yr[,c(2:12)]  
  
# Define UI for application that plots scatter plot  
ui <- fluidPage(  
  titlePanel("TB Burden Trend App - by year"),  
  sidebarLayout(  
    sidebarPanel(  
      h5("Retrieved data includes TB records from all around the  
         world, during 1990 to 2014. Prevalence indicates existing  
         number of cases, incident indicates newly added cases, and  
         mortality means number of death cases."),  
      selectInput(inputId = "metrics",  
                  label = "Select a measure metric:",  
                  choices = names(TB_df_metrics_rg),  
                  selected = "Prevalence"),  
      h6("Author: Christina Lu Jin")),  
    mainPanel(  
      plotOutput(outputId = 'linePlot', width = "100%",  
                 height = "600px", click = "plot_click")  
    )  
  )  
)
```

```

# Define server for application that plots scatter plot
server <- function(input, output, session) {

  output$linePlot <- renderPlot({
    title = "TB Burden Data Metrics Comparison"
    xyplot(TB_df_metrics_rg[,input$metrics] ~ TB_df_agg_rg$Year, main = title,
      xlab = "Year", ylab = input$metrics,
      groups = TB_df_agg_rg$Region, #auto.key=list(corner=c(1,1)),
      # par.settings = list(superpose.symbol = list(pch = 19, cex = 1,
      #                                              col = cividis(6))),
      type="l", lwd = 4,
      par.settings = list(superpose.line =
        list(lwd = 4, #col = cividis(6)
          col = c("#756bb1", "#8ca252", "#e6ac5b",
            "#e7cd48", "#c9777c", "#b67fac"))),
      key=list(corner=c(0,1), #col = cividis(6),
        text=list(c("AFR", "AMR", "EMR", "EUR", "SEA", "WPR")),
        lines=list(col = c("#756bb1", "#8ca252", "#e6ac5b",
          "#e7cd48", "#c9777c", "#b67fac"),
          lwd=4)
      )
    )
  })
}

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

```


Appendix 3 – App 3 R Code

```
#####  
##                                                                 ##  
##           TB Burden Correlation App 3                         ##  
##   ALY6070 - M4 Assignment   |   Christina Lu Jin             ##  
##                                                                 ##  
#####  
  
## install packages and import library   
#####  
  
# import data  
TB_data <- as.data.frame(read.csv("TB_Burden_Country.csv"))  
head(TB_data)  
  
# Data prep, reshape and clean up  
TB_df <- TB_data[,c(1,5:7,11,31,41,18,24,8,28,38,15,21)]  
colnames(TB_df) <- c("Country", "Region", "Year", "Total_Population", "Prevalence",  
                    "Incidence", "Incidence_HIV", "Death_no_HIV", "Death_HIV",  
                    "Prevalence_Rate_per_100k", "Incidence_Rate_per_100k",  
                    "Incidence_HIV_Rate_per_100k",  
                    "Mortality_no_HIV_Rate_per_100k",  
                    "Mortality_HIV_Rate_per_100k")  
TB_df_metrics <- TB_df[,c(4:14)]  
  
# Define UI for application that plots scatter plot  
ui <- fluidPage(  
  titlePanel("TB Burden Correlation App - between different metrics"),  
  sidebarLayout(  
    sidebarPanel(  
      h5("Retrieved data includes TB records from all around the  
        world, during 1990 to 2014. Prevalence indicates existing  
        number of cases, incident indicates newly added cases, and  
        mortality means number of death cases."),  
      selectInput(inputId = "x",  
                  label = "Select a independent variable (x):",  
                  choices = names(TB_df_metrics),  
                  selected = names(TB_df_metrics[[3]])),  
      selectInput(inputId = "y",  
                  label = "Select a dependent variable (y):",  
                  choices = names(TB_df_metrics[,c(2:11)]),  
                  selected = names(TB_df_metrics[[3]])),  
      h6("Author: Christina Lu Jin")),  
    mainPanel(  
      plotOutput(outputId = 'scatterPlot', width = "100%",  
                 height = "600px", click = "plot_click")  
    )  
  )  
)
```

```
# Define server for application that plots scatter plot
server <- function(input, output, session) {

  output$scatterPlot <- renderPlot({
    title = "TB Burden Data Metrics Comparison"
    plot(TB_df_metrics[,input$x], TB_df_metrics[,input$y], main = title,
         xlab = input$x, ylab = input$y, pch = 19,
         col = rgb(0.6, 0.4, 0.6, 0.2))
    abline(lm(TB_df_metrics[,input$y]~TB_df_metrics[,input$x],
              data = TB_df_metrics), col = rgb(0.9, 0.7, 0.6, 0.5),
           lty = 8, lwd = 3)
  }, res = 100)
}

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