CO2 World Application

I.Navarro s3473295

17/10/2017

CO2 World Application

This app has been designed to support the article from The Conversation: "Developing countries can prosper without increasing emissions September 22, 2017 5.40am AEST". Developing countries account for a big Part of CO2 Emmissions accross the world. But how do other countries with their population, GDP perform in different years? Find answers to the questions with the data in the app.

This is a Shiny web application. You can run the application by accessing the link provided below the code.

```
library(shiny)
library(plotly)
library(rgeos)
library(maptools)
library(ggmap)
library(broom)
library(dplyr)
library(ggplot2)
library(maps)
library(mapdata)
library(gdata)
library(rgdal)        # for readOGR(...)
library(ggthemes)
library(scales)
library(ggrepel) # new labels ggplot

#### READ THE DATA

#Read the data
co2 = read.xls ("CAIT_Country_GHG_Emissions.xlsx", sheet = 3, header = TRUE)


#Change the name of country column to COUNTRY
co2$COUNTRY<-co2$Country

#Change the name to co2
co2$Total.CO2.Emissions.Excluding.Land.Use.Change.and.Forestry..MtCO2. <- as.numeric(co2$Total.CO2.Emiss

co2$co2 <- co2$Total.CO2.Emissions.Excluding.Land.Use.Change.and.Forestry..MtCO2.

names(co2)

co2 <- co2[,-1]
co2 <- co2[,-2]
```

```r
# Read the map data
map <- read.csv('https://raw.githubusercontent.com/plotly/datasets/master/2014_world_gdp_with_codes.csv

# Drop GDP
names(map)
map <- map[,-2]

final <- left_join(map, co2)

# Same Data
wdata <- read.csv("CAIT_Country_GHG_Emissions.csv")
wdata$COUNTRY<-wdata$Country
wdata$CO2 <- as.numeric(wdata$CO2)

final2 <- left_join(map, wdata)

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

  # Application title
  titlePanel("Find out who Polutes the most, when and possible reasons"),
  # Sidebar with a slider input for number of bins

  sidebarPanel(
    #tags$head(
     # tags$style(type="text/css", "select { max-width: 140px; }"),
     # tags$style(type="text/css", ".span4 { max-width: 190px; }"),
     # tags$style(type="text/css", ".well { max-width: 180px; }")
   # ),

    h3("Select Year"),
    #### enter the year for the map
    sliderInput("Year",
                "Year",
                min = 1960,
                max = 2013,
                value = 2000, sep = "", animate = animationOptions(interval = 1300, loop = FALSE)),

    helpText("Select Year to see CO2 Emissions across the world"),
    br(),
    br(),
    br(),
    br(),
    br(),
    br(),
    br(),
    br(),
    br(),

    h3("Select Country"),
    # Select Country name here
    selectizeInput("name", label = "Country Name(s) of Interest",
```

```
                    choices = unique(wdata$Country), multiple = T,
                    options = list(maxItems = 4, placeholder = 'Select at least one Country'),
                    selected = "Australia"),

    helpText("Choose Maximum 4 countries to compare"),

    br(),
    h3("Select Measure"),
    selectInput("measure", "Enter unique Measure to see trend", c("Population","CO2", "GDP_PPP","GDP_USI
                selected = "Population"),
    helpText("Choose a metric to plot against years in the timeline"),

    br(),
    h3("About this App"),
    helpText("If developing countries are the higher poluters, let's find out how do they compare to Dev
    helpText(   a("See the full article",     href="https://theconversation.com/developing-countries-car
    )
  ),  #width=4,

  # Show a plot of the generated distribution
  mainPanel( #width = 8,

    h3(textOutput("selected_year")),

    plotlyOutput("mapPlot", height = 500),

    tabsetPanel(type = "tabs",
                tabPanel("See Timeline Comparision", plotlyOutput("trendPlot", height = 500)),
                #tabPanel("Relative", verbatimTextOutput("summary")),
                tabPanel("CO2 vs GDP", plotlyOutput("scatterPlot", height = 500))


    #plotlyOutput("trendPlot")
   # fluidRow(
   #   splitLayout(cellWidths = c("50%", "50%"), plotlyOutput("trendPlot"), plotlyOutput("scatterPlot"))

  )

))

#################################################################
# Define server logic
server <- function(input, output, session) {

  output$selected_year <- renderText({
    paste("Year", input$Year)
  })

  ## First get the Map
  output$mapPlot <- renderPlotly({

    # generate dataframe based on input$Year from ui.R
    FinalByYear <- final2 %>% filter(Year==input$Year)
```

```
  # light grey boundaries
  l <- list(color = toRGB("grey"), width = 0.5)

  # specify map projection/options
  g <- list(
    showframe = FALSE,
    showcoastlines = FALSE,
    projection = list(type = 'Mercator')
  )

  plot_geo(FinalByYear) %>%
    add_trace(
      z = ~CO2, color = ~CO2, colors = 'Blues',
      text = ~COUNTRY, locations = ~CODE, marker = list(line = l)
    ) %>%
    colorbar(title = 'CO2') %>%
    layout(
      title = 'Global CO2<br>Source:<a href="http://datasets.wri.org/dataset/cait-unfccc-annex-i-ghg-
      geo = g
    )

})

#### Line graph

wdata <- read.csv("CAIT_Country_GHG_Emissions.csv")

wdata <- wdata %>% filter(Country!="World")
wdata <- wdata %>% filter(Country!="European Union (28)")
wdata <- wdata %>% filter(Country!="European Union (15)")

wdata$CO2 <- as.numeric(wdata$CO2)

output$trendPlot <- renderPlotly({
  if (length(input$name) < 1) {
    print("Please select at least one country")
  } else {
    finalbyCountry <- wdata[wdata$Country == input$name, ]
    finalmeasure <- finalbyCountry[,input$measure]


    # Graph title
    if (length(input$name) > 2) {
      j_names_comma <- paste(input$name[-length(input$name)], collapse = ', ')
      j_names <- paste0(j_names_comma, ", and ", input$name[length(input$name)])
    } else {
      j_names <- paste(input$name, collapse = ' and ')
    }

    TitleMeasure <- paste(input$measure)
    graph_title  <- paste(TitleMeasure, " for ", j_names, sep="")

    ggideal_point <- ggplot(finalbyCountry) +
```

```
      geom_line(aes(x = Year, y = finalmeasure,  color = Country)) +
      #geom_line(aes(x=Year, y=mean(finalmeasure, color="black"))) +
      labs(x = "Year", y = TitleMeasure, title = graph_title) +
      scale_colour_hue("Country", l = 70, c = 150) +
      ggthemes::theme_few() +
      theme(legend.direction = "horizontal", legend.position = "bottom") +
      scale_y_continuous(labels=comma) +
      geom_vline(xintercept = input$Year, linetype="dotted", color = "black", size=0.5)

    # Convert ggplot object to plotly
    gg <- plotly_build(ggideal_point)

    # Use Plotly syntax to further edit the plot:
    gg$layout$annotations <- NULL # Remove the existing annotations (the legend label)
    gg$layout$annotations <- list()


    gg$layout$showlegend <- FALSE # remove the legend
    gg$layout$margin$r <- 170 # increase the size of the right margin to accommodate more room for th
    gg

  }
})

##### ScatterPlot

output$scatterPlot <- renderPlotly({


  if (length(input$name) < 1) {
    print("Please select at least one country")
  } else {

    #wdatabyCountry <- wdata[wdata$Country == input$name, ]
    #wdataByYear <- wdata %>% filter(Year==input$Year)
    wdataByYear <- wdata[wdata$Year == input$Year, ]
    ##wdataByYearbyCountry <- wdataByYear[wdataByYear$Country == input$name, ]

    wdataByYearbyCountry <- wdataByYear[wdataByYear$Country %in% input$name, ]


    # Graph title
    if (length(input$name) > 2) {
      j_names_comma <- paste(input$name[-length(input$name)], collapse = ', ')
      j_names <- paste0(j_names_comma, ", and ", input$name[length(input$name)])
    } else {
      j_names <- paste(input$name, collapse = ' and ')
    }
    inputyear <- paste(input$year)
    graph_title2  <- paste("GDP vs CO2 for ", j_names, sep="")


    t <- ggplot(wdataByYear, aes(label=Country, label2=Year)) +
```

```
        scale_size_continuous() +
        geom_point(data=wdataByYear, mapping=aes(x=GDP_USD, y=CO2, size=Population), colour="grey50") +
        geom_point(data=wdataByYearbyCountry, mapping=aes(x=GDP_USD, y=CO2, size=Population, colour=Cou
       # geom_label_repel(aes(wdataByYearbyCountry$Country), size=3) +
        labs(title = graph_title2, x = "GDP (Usd)", y = "CO2") +
        scale_x_continuous(labels=comma, limits = c(0, 12800000)) +
        scale_colour_hue("Country", l = 70, c = 150) +
        ggthemes::theme_few()

      tt <- plotly_build(t)
      tt <- ggplotly(tt,tooltip = c("Country"))
      tt

     # Use Plotly syntax to further edit the plot:
      tt$layout$annotations <- NULL # Remove the existing annotations (the legend label)
      tt$layout$annotations <- list()


      tt$layout$showlegend <- FALSE # remove the legend
      tt$layout$margin$r <- 170 # increase the size of the right margin to accommodate more room for th
      tt



    }

  })

}

# Run the application
shinyApp(ui = ui, server = server, options=list(
  #width="100%",
  height="100%") #options = list(height=1080)
)
```

Links

Access the application Here

https://iramix.shinyapps.io/CombinedCO2WorldApp/

Access the full Converation Article here

https://theconversation.com/developing-countries-can-prosper-without-increasing-emissions-84044.

Original source data (before some transformations) can be found here.

http://datasets.wri.org/dataset/cait-unfccc-annex-i-ghg-emissions-data