

MATH2270 Assignment 3

Interactive Storytelling

Student Details

- Zihao Liu (s3633311)

Story URL

- Dan, K. (2018, February 22). Stop obsessing about GDP growth—GDP per capita is far more important. *QUARTZ*. Retrieved from <https://qz.com/1194634/the-world-bank-wont-stop-reporting-gdp-instead-of-gdp-per-capita-and-it-is-driving-me-crazy/> (<https://qz.com/1194634/the-world-bank-wont-stop-reporting-gdp-instead-of-gdp-per-capita-and-it-is-driving-me-crazy/>)

Data Source

- World Development Indicators (2018). *The World Bank* . Available online at: <http://cait2.wri.org> (<http://cait2.wri.org>).

Visualisation URL

- https://hipposon.shinyapps.io/GDPPC_VS_INCOME/ (https://hipposon.shinyapps.io/GDPPC_VS_INCOME/)

RPub Link

- <http://rpubs.com/hipposon/434914> (<http://rpubs.com/hipposon/434914>)

Code

```
library(shiny)
library(plotly)
library(rgeos)
library(maptools)
library(ggmap)
library(broom)
library(dplyr)
library(ggplot2)
library(maps)
library(mapdata)
library(readr)
library(gdata)
library(rgdal)          # for readOGR(...)
library(ggthemes)
```

```

library(scales)
library(ggrepel) # new labels ggplot
library(tidyr)
library(readr)
library(shinydashboard)
#### READ THE DATA

#Read the data
world_development <- read_csv("/Users/hipposon/GDPPC_VS_INCOME/0e893ae8-c7a8-41ae-8871-7edd63821f8d_Data.csv")

#Rename the colname
colnames(world_development) <- c("Country_Name", "Country_Code", "Series_Name", "Series_Code", 1960,
                                1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971,
                                1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983,
                                1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994,
                                1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
                                2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016,
                                2017)

#Change the type of lists
world_development[,5:62] <- lapply(world_development[,5:62], as.numeric)
world_development <- world_development[-(5:34)]

#Create a dataframe for GDP per capital
GDPPC <- world_development %>% filter(`Series_Code` == "NY.GDP.PCAP.PP.KD")
GDPPC <- GDPPC %>% gather('1990', '1991', '1992', '1993', '1994',
                          '1995', '1996', '1997', '1998', '1999', '2000', '2001',
                          '2002', '2003', '2004', '2005',
                          '2006', '2007', '2008', '2009', '2010', '2011', '2012',
                          '2013', '2014', '2015', '2016',
                          '2017', key = "Year", value = "GDP_Per_Capita")

#Create a dataframe for income
INCOME <- world_development %>% filter(`Series_Code` == "NY.ADJ.NNTY.PC.KD")
INCOME <- INCOME %>% gather('1990', '1991', '1992', '1993', '1994',
                          '1995', '1996', '1997', '1998', '1999', '2000', '2001',
                          '2002', '2003', '2004', '2005',
                          '2006', '2007', '2008', '2009', '2010', '2011', '2012',
                          '2013', '2014', '2015', '2016',
                          '2017', key = "Year", value = "Net_Income_Per_Capita")

#Merge two dataset
Final <- inner_join(GDPPC, INCOME, by = c("Country_Name", "Year", "Country_Code"))
Final <- Final[-c(3, 4, 7, 8)]

#Create a dataframe for School Enrolment
# ENROL <- world_development %>% filter(`Series Code` == "School enrollment, seco

```

```

ndary (% net)")
# ENROL <- ENROL %>% gather('1990', '1991', '1992', '1993', '1994',
#                             '1995', '1996', '1997', '1998', '1999', '2000',
#                             '2001', '2002', '2003', '2004', '2005',
#                             '2006', '2007', '2008', '2009', '2010', '2011',
#                             '2012', '2013', '2014', '2015', '2016',
#                             '2017', key = "Year", value = "School Enrollmen
t")

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

  # Application title
  dashboardHeader(title = "GDPPC vs INCOME"),
  dashboardSidebar(disable = TRUE),

  dashboardBody(
    #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; }")
    # ),
    fluidRow(
      box( h3("Select Year"),
          #### enter the year for the map
          sliderInput("Year",
                      "Year",
                      min = 1990,
                      max = 2017,
                      value = 2000, sep = "", animate = animationOptions(in
terval = 1300, loop = FALSE)),

          helpText("Select Year to see GDP per capital 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(Final$'Country_Name'), multiple =
T,

                        options = list(maxItems = 4, placeholder = 'Select
at least one Country'),

                        selected = "Australia"),

          helpText("Choose Maximum 4 countries to compare for time comparis
on"),

```

```

        br(),
        h3("Select Measure"),
        selectInput("measure", "Enter unique Measure to see trend", c("GDP Per Capita" = "GDP_Per_Capita", "Net Income Per Capita" = "Net_Income_Per_Capita"),
                    selected = "GDP Per Capita"),
        helpText("Choose a metric to plot against years in the timeline")
    ,

    br(),
    h3("About this App"),
    helpText("Emphasizing GDP per capita rather than GDP growth is just a start. An even better step would be for the World Bank to put more focus on median household income rather than GDP per capita. Knowing that a country's GDP per capita is growing does not necessarily tell you that the typical person is doing better—all of that growth might be going to a small group of already wealthy people. Median incomes tell you more about how most people are getting along."),
    helpText(a("See the full article", href="https://qz.com/1194634/the-world-bank-wont-stop-reporting-gdp-instead-of-gdp-per-capita-and-it-is-driving-me-crazy/")),
    ), width = 4),
    box( mainPanel( #width = 8,

        h3(textOutput("selected_year")),

        plotlyOutput("mapPlot", height = 400, width = 1080),

        tabsetPanel(type = "tabs",
                    tabPanel("GDP Per Capita Top 10", plotlyOutput("barPlot", height = 400, width = 1080)),
                    tabPanel("Timeline Comparison", plotlyOutput("trendPlot", height = 400, width = 1080)))

        ), width = 8)
    )

) #width=4,

# Show a plot of the generated distribution
)

#####
# 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 <- Final[Final$Year == input$Year, ]

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

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

plot_geo(FinalByYear) %>%
  add_trace(
    z = ~FinalByYear$'GDP_Per_Capita', color = ~FinalByYear$'GDP_Per_Capita', colors = 'Reds',
    text = ~FinalByYear$`Country_Name`, locations = ~FinalByYear$`Country_Code`, marker = list(line = l)
  ) %>%
  colorbar(title = 'GDP per capita US$', tickprefix = '$', limits = c(0, 105000)) %>%
  layout(
    title = 'Global GDP per capita<br>Source:<a href="http://www.worldbank.org/">World Development Indicator</a>',
    geo = g
  )
})

###Line graph
output$trendPlot <- renderPlotly({
  if (length(input$name) < 1) {
    print("Please select at least one country")
  } else {
    finalbyCountry <- reactive({
      finalbyCountry <- Final[Final$Country_Name %in% input$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 ')
    }

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

    p <- ggplot(data = finalbyCountry(), mapping = aes_string(x = "Year",

```

```

y = input$measure, group = "Country_Name", color = "Country_Name"))
  p <- p + geom_line() + geom_point() + labs(x = "Year", y = TitleMeasure, title = graph_title) +
    scale_colour_hue("Country", l = 70, c = 150) +
    ggthemes::theme_few() +
    theme(legend.direction = "horizontal", legend.position = "bottom",
text = element_text(size=7), axis.text.x = element_text(angle = 90,), plot.title =
element_text(size=16)) + scale_y_continuous(labels=comma) +
    geom_vline(xintercept = input$Year, linetype="dotted", color = "black", size=0.5)

  pp <- plotly_build(p)
  pp$layout$annotations <- NULL # Remove the existing annotations (the legend label)
  pp$layout$annotations <- list()

  pp$layout$showlegend <- FALSE # remove the legend
  pp$layout$margin$r <- 170 # increase the size of the right margin to accommodate more room for the annotation labels
  pp
}

})

###bar chart
output$barPlot <- renderPlotly({

  Top10 <- reactive({
    topset <- Final %>% filter(Year == input$Year) %>% arrange(desc(GDP_Per_Capita))
    topset <- topset[1:10,]
    topset
  })

  Title2 <- paste("The Top 10 GDP per capita (US$) vs their Net income per capita (US$) in ", input$Year, sep = " ")

  b <- plot_ly() %>%
    add_bars(
      x = Top10()$GDP_Per_Capita,
      y = Top10()$Country_Name,
      width = 0.5,
      marker = list(
        color = 'rgba(222,45,38,0.8)'
      ),
      name = 'GDP Per Capita (US$)'
    ) %>% add_bars(
      x = Top10()$Net_Income_Per_Capita,
      y = Top10()$Country_Name,
      width = 0.5,
      marker = list(
        color = 'Purple'
      ),

```

```

      name = 'Net Income Per Capita (US$)'
    ) %>% layout(title = Title2, titlefont = list(size=16), legend=list(font
ont =list(size=8)),
               margin = list(l = 130), xaxis = list(title = "US$", range
= c(0, 140000)), yaxis = list(title = "Country",
categoryorder = "array",categoryarray = ~rev(Top10())$Country_Name))) #order
by GDP per capita
      b
    })
  }
# Run the application
shinyApp(ui = ui, server = server, options=list(
  width="100%",
  height="100%") #options = list(height=1080)
)

```