

# Assignment 3:

Storytelling with Open Data

**Name:** Faith Zhong-Xian Ha

**Student Number:** s3890479

**URL:** <https://9fvv6i-faith-ha.shinyapps.io/Emissions/>

## Code:

```
library(shiny)
```

```
library(shinydashboard)
```

```
library(ggplot2)
```

```
library(plotly)
```

```
library(readxl)
```

```
library(dplyr)
```

```
library(tidyverse)
```

```
library(viridis)
```

```
library(RColorBrewer)
```

```
# Define your data frame df1 here
```

```
df <- read_excel("emissions.xlsx", sheet = "Emissions")
```

```
df1 <- df %>%
```

```
  pivot_longer(cols = -c(Company, Status, Rank),
```

```
    names_to = "Year",
```

```
values_to = "Emissions")
```

```
df_total <- df1[df1$Company %in% c("Top 100 Companies", "Global"), ]
```

```
df_total1 <- df1[df1$Company %in% c("State", "Investor"), ]
```

```
df_total1$Company <- as.factor(df_total1$Company)
```

```
df_total1$Year <- as.numeric(df_total1$Year)
```

```
df2 <- df1 %>%
```

```
  filter(!Company %in% c("Global", "Remaining", "Top 100 Companies", "State", "Investor", "T", "S",  
"I"))
```

```
df3 <- df2 %>%
```

```
  group_by(Company) %>%
```

```
  mutate(value_change = ifelse(Emissions[Year == 2015] > Emissions[Year == 1998], "Increased",  
"Decreased"))
```

```
filtered_df <- df2 %>% filter(Year == 2015)
```

```
filtered_df$Rank_Group <- cut(filtered_df$Rank, breaks = c(0, 25, 50, 75, 100), labels = c("1-25", "26-  
50", "51-75", "76-100"), include.lowest = TRUE)
```

```
grouped_df <- filtered_df %>%
```

```
  group_by(Rank_Group, Status) %>%
```

```
  summarise(Total_Emissions = sum(Emissions))
```

```
company_counts <- filtered_df %>%
```

```
  filter(Year == 2015) %>%
```

```
  group_by(Rank_Group, Status) %>%
```

```
  summarise(Company_Count = n())
```

```
grouped_df <- left_join(grouped_df, company_counts, by = c("Rank_Group", "Status"))
```

```
df2$Year <- as.numeric(df2$Year)
```

```
count <- merge(filtered_df, df3, by = c("Company", "Status", "Rank", "Year", "Emissions"))
```

```
count1 <- count %>%
```

```
  filter(Status %in% c("Investor", "State") & value_change %in% c("Increased", "Decreased")) %>%
```

```
  group_by(Rank_Group, Status, value_change) %>%
```

```
  summarise(Count = n())
```

```
count2 <- count1 %>%
```

```
  filter(Status == "Investor")
```

```
count3 <- count1 %>%
```

```
  filter(Status == "State")
```

```
# UI
```

```
ui <- dashboardPage(
```

```
  dashboardHeader(title = HTML("Breakdown of GtCO<sub>2</sub>e")),
```

```
  dashboardSidebar(
```

```
    sidebarMenu(
```

```
      menuItem("Overview", tabName = "global", icon = icon("area-chart")),
```

```
      menuItem("Top 100 Comparison", tabName = "investorstate", icon = icon("area-chart")),
```

```
      menuItem("Individual Top 100", tabName = "linechart", icon = icon("line-chart"))
```

```
    )
```

```
  ),
```

```
  dashboardBody(
```

```
    tabItems(
```

```

# Global tab

tabItem(tabName = "global",

  fluidRow(

    infoBox(

      "Top 100 Companies Represent", "72%", HTML("of the world's GtCO<sub>2</sub>e"), icon
= icon("exclamation-circle"),

      fill = TRUE

    ),

    infoBox(

      "Global Emissions Reached", HTML("38 GtCO<sub>2</sub>e"), "in 2015", icon =
icon("tree"), color = "blue",

      fill = TRUE

    ),

    infoBox(

      HTML("GtCO<sub>2</sub>e refers to"), "Gigatonnes", "of Carbon Dioxide", icon =
icon("info"), color = "teal",

      fill = TRUE

    )

  ),

  fluidRow(

    column(width = 6,

      plotlyOutput("areaChart")

    ),

    column(width = 6,

      plotlyOutput("areaChart1")

    )

  ),

),

# Investor vs State Chart tab

tabItem(tabName = "investorstate",

  fluidRow(

    infoBox(

```

```
    "Top 25 Companies represent", "51%", HTML("of the world's GtCO<sub>2</sub>e"), icon =  
    icon("cloud"),
```

```
        fill = TRUE
```

```
    ),
```

```
    infoBox(  
        "State Companies", "Increasing", HTML("GtCO<sub>2</sub>e in All Categories"), icon =  
        icon("angle-double-up"), color = "blue",
```

```
        fill = TRUE
```

```
    ),
```

```
    infoBox(  
        "Top Investor Companies Most", "Likely to Decrease", HTML("GtCO<sub>2</sub>e"), icon =  
        icon("angle-double-down"), color = "teal",
```

```
        fill = TRUE
```

```
    ),
```

```
    ),
```

```
    fluidRow(  
        column(6, box(width = 12, plotlyOutput("investorChart1", height = 390))),  
        column(6,
```

```
            box(width = 12, plotlyOutput("investorChart2", height = 175)),  
            box(width = 12, plotlyOutput("investorChart3", height = 175))  
        )  
    )  
),
```

```
# Line Chart tab
```

```
tabItem(tabName = "linechart",
```

```
    fluidRow(  
        infoBox(  
            "China (Coal) is the", "Largest", HTML("GtCO<sub>2</sub>e Emitter"), icon = icon("cloud"),  
            fill = TRUE
```

```
        ),
```

```
        infoBox(  
            "China (Coal) is the", "Largest", HTML("GtCO<sub>2</sub>e Emitter"), icon = icon("cloud"),  
            fill = TRUE
```

```

    "Companies Generally", "Increasing", "Emissions", icon = icon("tree"), color = "blue",
    fill = TRUE
  ),
  infoBox(
    "State Companies", "Key to Reducing", HTML("Global GtCO<sub>2</sub>e"), icon =
    icon("exclamation-circle"), color = "teal",
    fill = TRUE
  )
),
sidebarLayout(
  sidebarPanel(
    selectizeInput("company", "Select Company", choices = unique(df2$Company), multiple =
TRUE),
    selectizeInput("rank", "Select Rank", choices = 1:100, multiple = TRUE),
    helpText("Maximum 5 companies allowed")

  ),
  mainPanel(
    plotlyOutput("linechart", height = "340px", width = "100%"),
    fluidRow(
      tags$div(
        style = "margin-top: 10px; margin-left: 10px;",
        tags$sp("Data Reference:"),
        tags$ul(
          tags$li(
            "Paul, G., Heede, R., & Vlugt, I. (2017).",
            tags$em("Climate Accountability Institute"),
            ". Climate Accountability Institute .",
            tags$a(
              href = "https://climateaccountability.org/publications.html",
              "https://climateaccountability.org/publications.html"
            )
          )
        )
      )
    )
  )
)

```

$$\begin{array}{r} ) \\ ) \\ ) \\ ) \\ ) \\ ) \\ ) \\ ) \\ ) \\ ) \end{array}$$

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

```
observe({
  updateSelectizeInput(session, "company", choices = unique(df2$Company), selected =
input$company)
```

```
updateSelectizeInput(session, "rank", choices = 1:100, selected = input$rank)
```

```
selected_company <- input$company
```

```
selected_rank <- input$rank
```

```
total_count <- length(selected_company) + length(selected_rank)
```

```
excess <- max(0, total_count - 5)
```

```
if (excess > 0) {
```

```
if (length(selected_company) > excess) {
```

```
selected_company <- selected_company[1:(length(selected_company) - excess)]
```

```
} else {
```

```
selected_company <- NULL
```

```

    selected_rank <- selected_rank[1:(length(selected_rank) - (excess - length(selected_company)))]
  }
}

updateSelectizeInput(session, "company", choices = unique(df2$Company), selected =
selected_company)
updateSelectizeInput(session, "rank", choices = 1:100, selected = selected_rank)
})

```

```

filtered_df <- reactive({
  data <- df2

  if (!is.null(input$company) && !is.null(input$rank)) {
    selected_items <- c(input$company, input$rank)
    data <- data[data$Company %in% selected_items | data$Rank %in% selected_items, ]
  } else if (!is.null(input$company)) {
    data <- data[data$Company %in% input$company, ]
  } else if (!is.null(input$rank)) {
    data <- data[data$Rank %in% input$rank, ]
  }

  data
})

```

```

output$linechart <- renderPlotly({
  if (is.null(input$company) && is.null(input$rank)) {
    ggplot() +
      theme_minimal() +
      theme(axis.line = element_line(color = "black"),

```



```

    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks = element_blank(),
    plot.title = element_text(hjust = 0.5)) +
xlim(c(1988, 2015)) +
ylim(c(0, 100)) +
labs(x = "Year", y = HTML("GtCO<sub>2</sub>e"), title = HTML("GtCO<sub>2</sub>e of each
Company"))
} else {

    graph <- ggplot(filtered_df(), aes(x = Year, y = Emissions/1000, group = interaction(Company,
Rank), color = Company, text = paste('Name:', Company, HTML("\nGtCO<sub>2</sub>e:"),
round(Emissions/1000, 2), '\nRank: ', Rank, '\nYear:', Year))) +
    geom_line() +
    labs(x = "", y = HTML("GtCO<sub>2</sub>e"), title = HTML("GtCO<sub>2</sub>e of each
Company")) +
    theme_minimal() +
    theme(
      panel.grid.major.x = element_blank(),
      panel.grid.minor.x = element_blank(),
      panel.grid.minor.y = element_blank(),
      axis.line = element_line(color = "black"),
      axis.ticks.y = element_line(color = "black"),
      axis.text.y = element_text(hjust = .5),
      axis.text.x = element_text(hjust = .5)
    ) +
    scale_x_continuous(breaks = c(1988, 2002, 2015)) +
    scale_color_manual(values = brewer.pal(name = "Dark2"))

    plotly_graph <- plotly::ggplotly(graph, tooltip = "text")

```

```
}
})
```

```
output$areaChart <- renderPlotly({
  df_total$Year <- as.numeric(df_total$Year)
```

```
  p <- ggplot(df_total, aes(x = Year, y = Emissions, col = Company, group = Company, text =
paste(HTML("GtCO<sub>2</sub>e:"), round(Emissions, 2), '\nYear: ', Year))) +
```

```
    geom_line() +
```

```
    theme_minimal() +
```

```
    theme(
```

```
      panel.grid.major.x = element_blank(),
```

```
      panel.grid.minor.x = element_blank(),
```

```
      panel.grid.minor.y = element_blank(),
```

```
      axis.line = element_line(color = "black"),
```

```
      axis.ticks.y = element_line(color = "black"),
```

```
      axis.text.y = element_text(hjust = .5),
```

```
      axis.text.x = element_text(hjust = .5)
```

```
    ) +
```

```
    labs(x = "", y = HTML("Global GtCO<sub>2</sub>e"),
```

```
          title = HTML("Global vs. Top 100 Companies GtCO<sub>2</sub>e")) +
```

```
    scale_x_continuous(breaks = c(1988, 2002, 2015)) +
```

```
    scale_color_manual(values = c("#2E8B57", "#990033")) +
```

```
    scale_y_continuous(limits = c(0, 40), expand = c(0, 0))
```

```
  p + labs(col = "")
```

```
  p1 <- ggplotly(p, tooltip = c("text"))
```

```
  p2 <- layout(p1, showlegend = TRUE, legend = list(title = ""))
```

p2

}}

```
output$areaChart1 <- renderPlotly({
```

```
  df_total1$Year <- as.numeric(df_total1$Year)
```

```
  df_total1$Company <- factor(df_total1$Company, levels = c("State", "Investor"))
```

```
  gp1 <- ggplot(df_total1, aes(x = Year, y = round(Emissions, 2), col = Company, group = Company,  
text = paste(HTML("GtCO<sub>2</sub>e:"), round(Emissions, 2), '\nYear: ', Year))) +
```

```
    geom_line() +
```

```
    theme_minimal() +
```

```
    theme(
```

```
      panel.grid.major.x = element_blank(),
```

```
      panel.grid.minor.x = element_blank(),
```

```
      panel.grid.minor.y = element_blank(),
```

```
      axis.line = element_line(color = "black"),
```

```
      axis.ticks.y = element_line(color = "black"),
```

```
      axis.text.y = element_text(hjust = 1),
```

```
      axis.text.x = element_text(hjust = .5),
```

```
      legend.title = element_blank()
```

```
    ) +
```

```
    labs(x = "", y = HTML("GtCO<sub>2</sub>e"),
```

```
          title = HTML("State vs. Investor GtCO<sub>2</sub>e<br><span style='font-size: 12px;'>Top 100  
Companies</span>")) +
```

```
    scale_x_continuous(breaks = c(1988, 2002, 2015)) +
```

```
    scale_color_manual(values = c("#FF6103", "#241571")) +
```

```
scale_y_continuous(limits = c(0, NA), expand = c(0, 0)) +  
labs(col = NULL)
```

```
gp1
```

```
gp <- ggplotly(gp1, tooltip = c("text"))
```

```
gp <- layout(gp, margin = list(t = 65))
```

```
gp
```

```
}}
```

```
output$investorChart1 <- renderPlotly({  
  gp2 <- ggplot(grouped_df, aes(x = Rank_Group, y = round(Total_Emissions/1000, 2), fill = Status,  
    text = paste(HTML("GtCO<sub>2</sub>e:"), round(Total_Emissions/1000, 2),  
    '\nNumber of Companies: ', Company_Count))) +  
    geom_bar(stat = "identity", position = "dodge") +  
    theme_minimal() +  
    theme(  
      panel.grid.major.x = element_blank(),  
      panel.grid.minor.x = element_blank(),  
      panel.grid.minor.y = element_blank(),  
      axis.line = element_line(colour = "black", size = 0.5)
```

```

) +
scale_y_continuous(limits = c(0, 20), expand = c(0, 0)) +
scale_fill_manual(values = c("#241571", "#FF6103")) +
labs(x = "Category", y = HTML("GtCO<sub>2</sub>e"), title = HTML("GtCO<sub>2</sub>e by
Category"), fill = NULL)

ggplotly(gp2, tooltip = c("text"))

})

output$investorChart2 <- renderPlotly({
  gp3 <- ggplot(count2, aes(x = Rank_Group, y = Count, fill = value_change, text = paste("Number: ",
Count))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Category", y = "Company Number", fill = "") +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_blank(),
    axis.title.y = element_text(size = 8),
    axis.title.x = element_text(size = 8),
    axis.line = element_line(colour = "black", size = 0.5)
  ) +
  scale_y_continuous(limits = c(0, 20), expand = c(0, 0)) +
  scale_fill_manual(values = c("#40e0d0", "#191970")) +
  labs(title = HTML("Investor GtCO<sub>2</sub>e"), subtitle = "Decrease vs. Increase")

  ggplotly(gp3, tooltip = "text")
})

output$investorChart3 <- renderPlotly({

```

```

gp4 <- ggplot(count3, aes(x = Rank_Group, y = Count, fill = value_change, text = paste("Number: ",
Count))) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_blank(),
    axis.line = element_line(colour = "black", size = 0.5),
    axis.title.y = element_text(size = 8),
    axis.title.x = element_text(size = 8)

  ) +
  scale_y_continuous(limits = c(0, 20), expand = c(0, 0)) +
  scale_fill_manual(values = c("#40e0d0", "#191970")) +
  labs(x = "Category", y = "Company Number", title = HTML("State GtCO<sub>2</sub>e"))

ggplotly(gp4, tooltip = "text") %>%
  layout(legend = list(title = list(text = "")))

})

}

shinyApp(ui = ui, server = server)

```