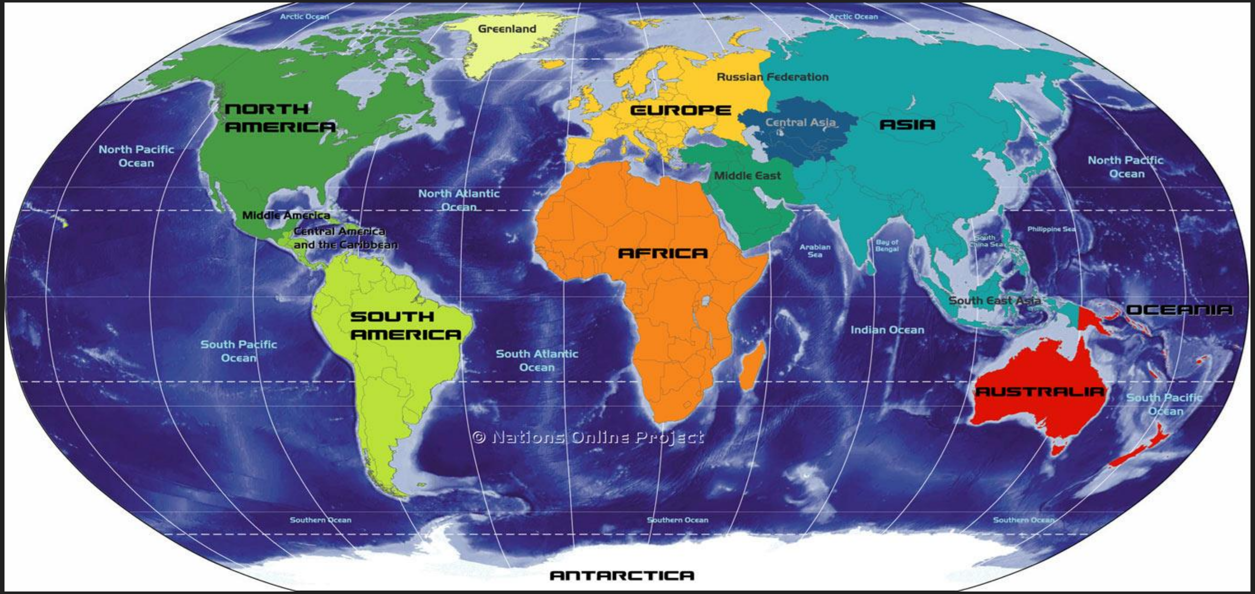
General Wealth Analysis by Country Across the World



Seattle OG

Shucheng Zhang

Weihua Pan

Yufei Sun

Rishabh Chawla

Introduction

The analysis of general wealth of countries around the world is a complex and multifaceted problem, requiring the integration of economic, social, and political factors. Despite the availability of various indicators and metrics to measure wealth, such as human capital and natural capital, there is no consensus on a single comprehensive measure of general wealth.

Based on the rapid development of technology, it's been an amazing start to 2023 with the growth of artificial intelligence, neural science, medical and pharmaceutical engineering, nuclear and sustainable clean energy. However, there has always been a significant problem for the human beings during the recent decades because of the continued fossil energy exploitation, the problem about our future, our destiny should be considered priorly.

Here we have designed a application which can analyze the annual economic situation of almost all countries in the world over the past 23 years. We hope that this can serve as a good tool and reference for analyzing future economic trends. Beyond the economic, this application also includes the data from the nature such as agriculture and fishery. Using this data and analysis to adjust a country's policies on environmental governance is also a practical and effective approach. The combination of three types of analysis method can fulfill most of situations.

For example, we can generate a time series reference by using the trend analysis for the nature capital: Timber over Asia continent for recent 10 years. It will give us a clear look that which country developed.

Dataset & EDA

Continent	Country.Name
Africa	Benin, Botswana, Burkina Faso, Burundi, Cameroon
Asia	Bahrain, Bangladesh, Cambodia, China, India
Europe	Albania, Armenia, Austria, Azerbaijan, Belarus
North America	Belize, Canada, Costa Rica, Dominican Republic, El Salvador
Oceania	Australia, Papua New Guinea, Solomon Islands
South America	Argentina, Bolivia, Brazil, Chile, Colombia

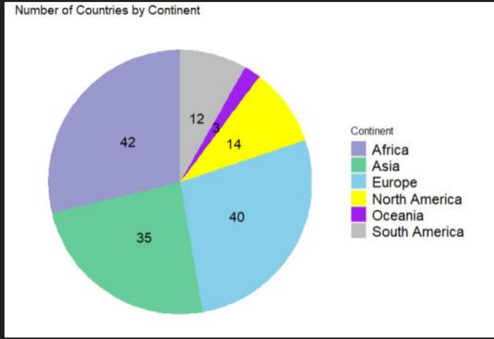
This a public dataset named wealth account from the World Bank.

https://databank.worldbank.org/source/wealth-accounts#

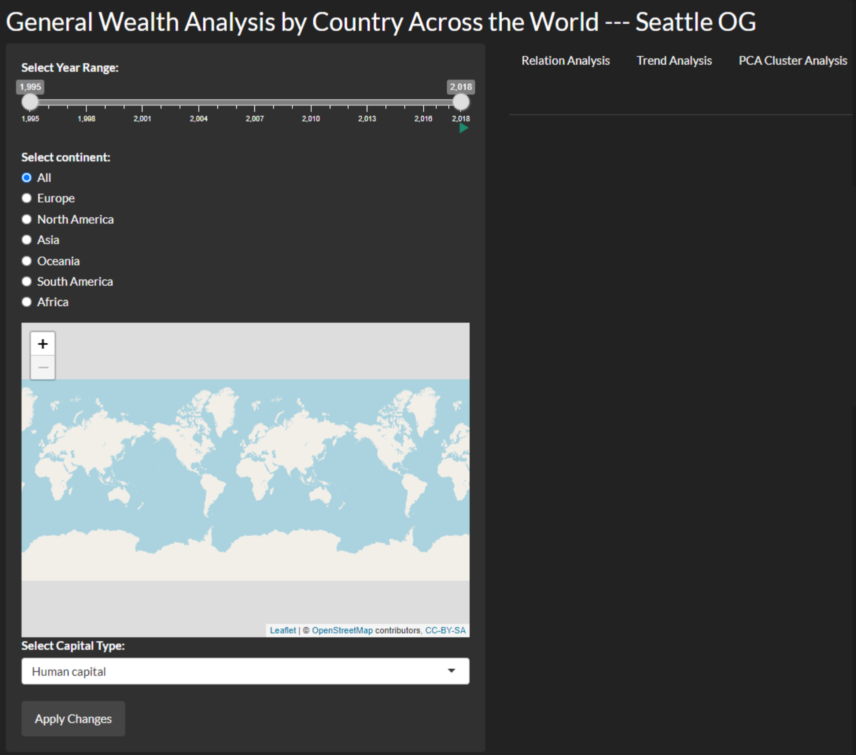
The dataset consists 182277 rows and 5 columns after cleaning the missing value, unrecognized characters, and useless columns. There are 146 countries group by 6 continents, 17 capital types through year from 1995 to 2018.

capitalType
Natural capital per capita, fisheries (constant 2018 US$)
Natural capital per capita, forests: ecosystem services (const...
Natural capital per capita, forests: timber (constant 2018 US$)
Natural capital per capita, fossil fuels (constant 2018 US$)
Natural capital per capita, mangroves (constant 2018 US$)
Natural capital per capita, nonrenewable assets (constant 20...
Natural capital per capita, nonrenewable assets: coal (consta...

```
Country.Name        capitalType              Year
Length:182277       Length:182277      Min.   :1995
Class :character    Class :character   1st Qu.:2000
Mode  :character    Mode  :character   Median :2006
                                       Mean   :2006
                                       3rd Qu.:2012
                                       Max.   :2018
  Total_USD          Continent
Length:182277       Length:182277
Class :character    Class :character
Mode  :character    Mode  :character
```



UI & Server




```r
# Sidebar layout
sidebarLayout(
  # Sidebar panel
  sidebarPanel(
    # Slider input for selecting year range
    sliderInput("yearRange", "Select Year Range:",
                min = 1995, max = 2018, value = c(1995, 2018),
                step = 1, animate = TRUE),

    # Checkboxes for selecting regions
    #Revise1 add South America
    #Revise2 change to the radio button
     radioButtons("continent", "Select continent:",
             choices = c("All", "Europe", "North America", "Asia", "Oceania", "South America", "Africa")),
     leafletOutput("map"),

    # Select input for selecting capital type
    selectInput("capitalType", "Select Capital Type:",
                choices = c("Human capital", "Human capital per capita", "Human
                            "Human capital per capita, male", "Human capital, f
                            "Natural capital, agricultural land", "Natural capi
                            "Natural capital, forests: timber", "Natural capita
                            "Natural capital, nonrenewable assets: gas", "Natur
                            "Natural capital, renewable")),

    # Action button to apply changes
    actionButton("applyChanges", "Apply Changes")
  ),
```

```r
# Define the Shiny app UI
ui1 <- fluidPage(
  # add a theme
  theme = shinytheme("darkly"),
  # App title
  titlePanel("General Wealth Analysis by Country Across the World --- Seattle OG"),

  # add a theme
  tags$head(
    tags$style(HTML("
    .irs-bar {
      ...
    }
    /* Add your custom CSS styles here */
    .sidebar {
      background-color: #f8f9fa;
      border-right: 1px solid #dee2e6;
      padding: 15px;
    }
    .sidebar h3 {
      color: #0c5460;
    }
    .sidebar .form-group {
      margin-bottom: 15px;
    }
    .sidebar .btn {
      background-color: #0c5460;
      color: #ffffff;
      font-weight: bold;
    }
    .sidebar .btn:hover {
      background-color: #0c5460;
      color: #ffffff;
      opacity: 0.9;
    }
  "))
  ),
```

```r
# Main panel
mainPanel(
  img(src="https://raw.githubusercontent.com/mark1472834185/6600-final-project/main/shinyApp/www/figures/earth.png", align = "right", width

  tabsetPanel(
    tabPanel("Relation Analysis",
             conditionalPanel(
               condition = "input.applyChanges > 0",
               p("The generated barchart and piechart regarding the information on the side panel. Showing detailed relationship between
                 countries in the capital type and year range selected"),
               div(style = "width: 100%; display: block;",
                   plotOutput("histogram")),
               div(style = "width: 100%; height: 2px; background-color: gray; display: block; margin-top: 20px; margin-bottom: 20px;"),
               div(style = "width: 100%; display: block;",
             ),
                   plotOutput("piechart"))
    ),

    tabPanel("Trend Analysis",
             conditionalPanel(
               condition = "input.applyChanges > 0",
               p("The generated Time-Series plot regarding the information on the side panel. Showing in detailed about the trend of each
                 country in the capital type selected and year range selected"),
               plotOutput("trendPlot")
             ),

             uiOutput("planetImage_trend")

    ),
```

UI & Server

```r
observe({

  # update barChart when reactive values changed
  output$histogram <- renderPlot({ barChart(values$df,values$capital)})

  # update pie Chart when reactive values changed
  output$piechart <- renderPlot({ pie(values$df,values$capital) })

  # update trend Plot when reactive values changed
  output$trendPlot <- renderPlot({ trendPlot(values$df, values$capital) })

  # update kmeans clustering when reactive values changed
  output$pca_cluster_plot <- renderPlotly({ km(values$df,values$k) })
})
```

```r
# Define the Shiny app server
server <- function(input, output, session) {
  source("www/functions/data_filter.R")
  source("www/functions/barChart.R")
  source("www/functions/pie.R")
  source("www/functions/trendPlot.R")
  source("www/functions/km.R")

  # imported dataset
  data_final <- read_csv("https://raw.githubusercontent.com/mark1472834185/6600-final-project/main/shinyApp/www/data/


  # create reactive variables
  values <- reactiveValues(df = NULL,
                           year = NULL,
                           continent = NULL,
                           capital = NULL,
                           k = NULL)

  # observe Apply change button
  observeEvent(input$applyChanges, {
    # reassign the values
    values$year <- input$yearRange
    values$continent <- input$continent
    values$capital <- input$capitalType
    values$k <- input$k

    # Prepare the filtered data
    values$df <- data_filter(data_final,values$year,values$continent,values$capital)
  })
```

```r
#Revise1 add interactive map into
cities <- data.frame(
  continent = c("North America", "Europe", "Asia", "Oceania", "South America", "Africa")
)

# Filter the data based on the user input
filtered_cities <- reactive({
  if (input$continent == "All") {
    cities
  } else {
    cities[cities$continent == input$continent, ]
  }
})

output$map <- renderLeaflet({
  if (input$continent == "Europe") {
    leaflet(data = filtered_cities()) %>%
      addTiles() %>%
      fitBounds(lng1 = -10, lat1 = 34, lng2 = 30, lat2 = 60)
  } else if (input$continent == "North America") {
    leaflet(data = filtered_cities()) %>%
      addTiles() %>%
      fitBounds(lng1 = -128, lat1 = 24, lng2 = -56, lat2 = 50)
  } else if (input$continent == "Asia") {
    leaflet(data = filtered_cities()) %>%
      addTiles() %>%
      fitBounds(lng1 = 80, lat1 = -10, lng2 = 150, lat2 = 45)
  } else if (input$continent == "Oceania") {
    leaflet(data = filtered_cities()) %>%
      addTiles() %>%
      fitBounds(lng1 = 112, lat1 = -47, lng2 = 180, lat2 = -10)
  } else if (input$continent == "South America") {
    leaflet(data = filtered_cities()) %>%
      addTiles() %>%
      fitBounds(lng1 = -75, lat1 = -45, lng2 = -50, lat2 = 5)
  } else if (input$continent == "Africa") {
    leaflet(data = filtered_cities()) %>%
      addTiles() %>%
      fitBounds(lng1 = -20, lat1 = -35, lng2 = 45, lat2 = 30)
  } else {
    leaflet(data = filtered_cities()) %>%
      addTiles() %>%
      fitBounds(lng1 = -180, lat1 = -90, lng2 = 180, lat2 = 90)
  }
})

}
```

Algorithm & Function

```r
# this function will draw a top10 barChart
# para: df, capital
barChart <- function(df,capital){
  if (is.null(df)) {
    return(NULL)  # Do not render any plot
  }
  # Aggregate data by country and sum the USD values
  df_top10 <- df %>%
    group_by(Country.Name) %>%
    summarize_at(capital,sum,na.rm=TRUE) %>%
    arrange(desc(.data[[capital]])) %>%
    head(10)  # Get the top 10 countries

  # draw a top10 bar
  p <- ggplot(df_top10, aes(x = reorder(Country.Name, .data[[capital]]), y = .data[[capital]] ) )  +
    geom_bar(stat = "identity", fill = "darkgrey") +
    theme(plot.background = element_rect(fill = "transparent"),
          panel.background = element_rect(fill = "transparent"),
          axis.text.x = element_text(angle = 45, hjust = 1, size = 14),
          axis.text.y = element_text(size = 14),
          axis.title.x = element_text(size = 16),
          axis.title.y = element_text(size = 16),
          plot.title = element_text(size = 18)) +
    labs(x = "Country Name", y = "Total USD", title = "Top 10 Countries by Total USD value shown in Bar chart")

  return(p)
}
```

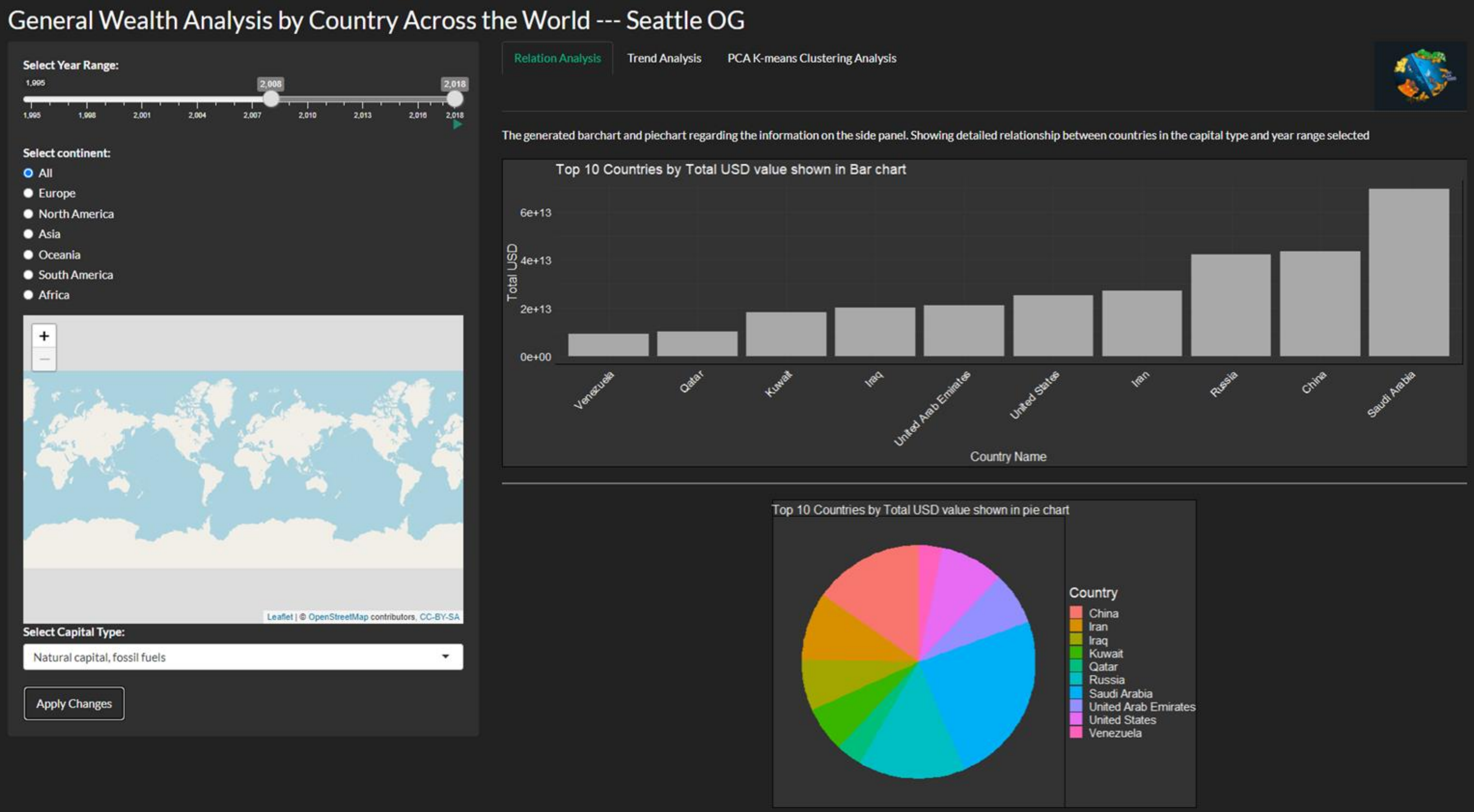
```r
library(plotly)
# this function will draw a top10 pie Chart
# para: df, capital
pie <- function(df,capital){
  if (is.null(df)) {
    return(NULL)  # Do not render any plot
  }
  # Aggregate data by country and sum the USD values
  df_top10 <- df %>%
    group_by(Country.Name) %>%
    summarize_at(capital,sum,na.rm=TRUE) %>%
    arrange(desc(.data[[capital]])) %>%
    head(10)  # Get the top 10 countries

  # draw a top10 pie chart
  # Create a pie chart using ggplot2
  p <- ggplot(df_top10, aes(x = "", y = .data[[capital]], fill = Country.Name)) +
    geom_bar(stat = "identity", width = 1) +
    coord_polar("y", start = 0) +
    theme_void() +
    theme(legend.position = "right", legend.text = element_text(size = 15),
          legend.title = element_text(size = 18),
          plot.title = element_text(size = 16, hjust = 0.5)) +
    labs(title = "Top 10 Countries by Total USD value shown in pie chart", fill = "Country")

  return(p)
}
```

Algorithm & Function

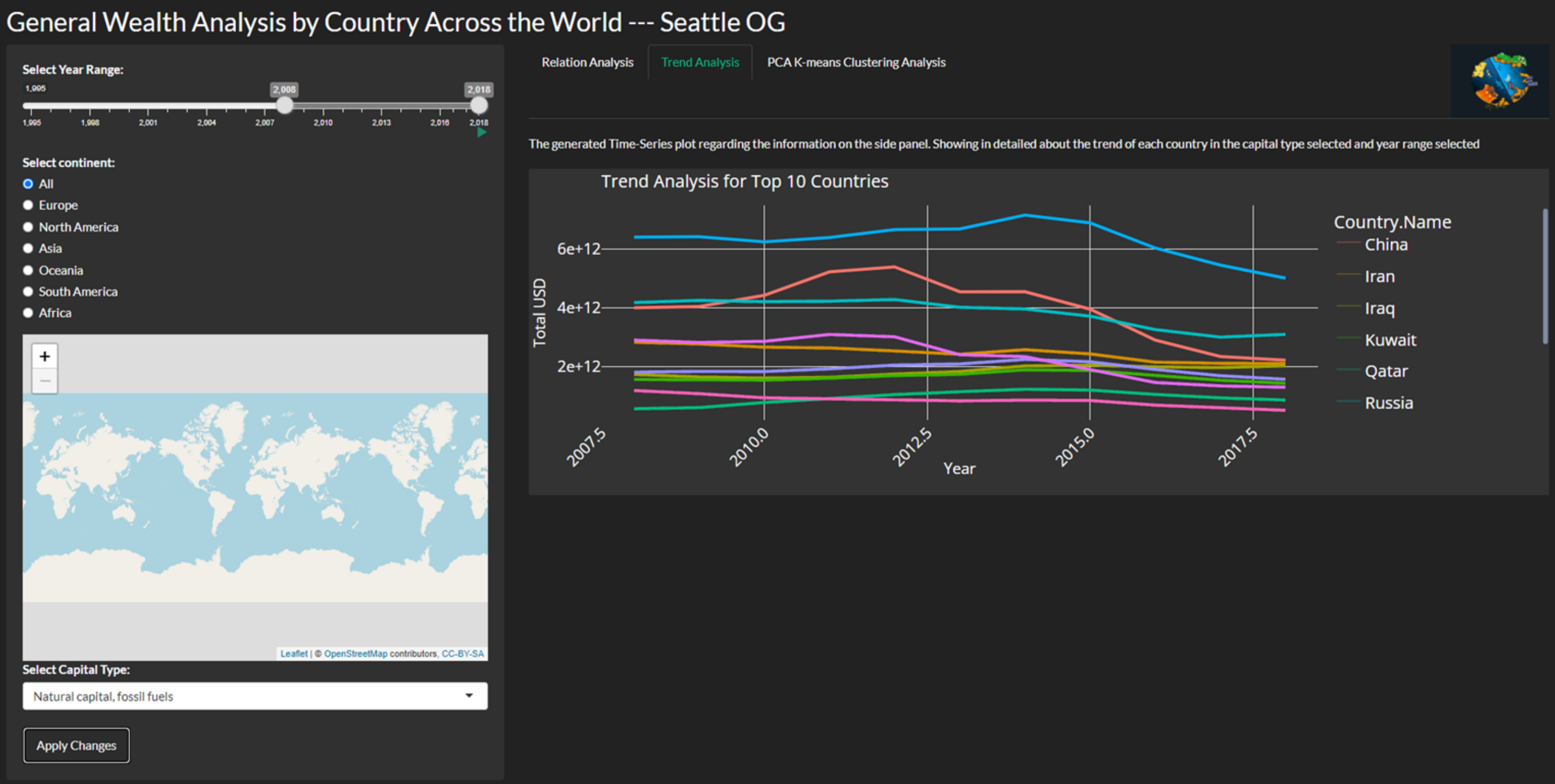
```r
# this function will draw a top10 trendPlot
# para: df, capital
trendPlot <- function(df,capital){
  if (is.null(df)) {
    return(NULL)  # Do not render any plot
  }
  # Aggregate data by country and sum the USD values
  df_top10 <- df %>%
    group_by(Country.Name) %>%
    summarize_at(capital,sum,rm.na=TRUE) %>%
    arrange(desc(.data[[capital]])) %>%
    head(10)  # Get the top 10 countries

  # Filter data to get the top 10 countries
  top_countries <- df_top10$Country.Name

  # Filter the data to include only the top 10 countries
  trend_data <- df %>%
    filter(Country.Name %in% top_countries)

  # Generate the trend line plot
  p <- ggplot(trend_data, aes(x = Year, y = .data[[capital]], color = Country.Name)) +
    geom_line(size = 1) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 14),
          axis.text.y = element_text(size = 14),
          axis.title.x = element_text(size = 14),
          axis.title.y = element_text(size = 14),
          plot.title = element_text(size = 16),
          legend.text = element_text(size = 15),
          legend.title = element_text(size = 16)) +
    labs(x = "Year", y = "Total USD", title = "Trend Analysis for Top 10 Countries")

  return(p)

}
```

```r
# this function apply kemans with df, n-clusters
# return a scatter plot with clusters label

km <- function(df,k){
  if (is.null(df)) {
    return(NULL)  # Do not render any plot
  }
  set.seed(11)

  library(dplyr)

  # Aggregate data by country and sum the USD values
  df <- df %>%
    group_by(Country.Name) %>%
    summarise(across(where(is.numeric), sum))

  # Remove columns with zero variance
  df_numeric <- df[,3:ncol(df)] %>% select(where(function(x) var(x) != 0))
  print(df_numeric)
  # dimensionality reduction (PCA)
  df_pca <- prcomp(df_numeric, scale. = TRUE)

  # Transform the data using the selected principal components
  num_pcs <- 3
  pca_transformed_data <- predict(df_pca, newdata = df_numeric)[, 1:num_pcs]

  # Perform k-means clustering on the transformed data
  kmeans_result <- kmeans(pca_transformed_data, centers = min(k, nrow(pca_transformed_data)))

  # Combine original df and cluster label
  df_clustered <- cbind(df, cluster = kmeans_result$cluster)

  # Combine the original dataset with the PCA transformed data and cluster assignments
  data_pca_clustered <- cbind(df, pca_transformed_data, cluster = kmeans_result$cluster)
  data_pca_clustered$cluster <- as.factor(data_pca_clustered$cluster)

  # Create a Plotly PCA clustering plot
  fig <- plot_ly(data_pca_clustered, type = "scatter", mode = "markers",
                 x = ~PC1, y = ~PC2,
                 text = ~Country.Name,
                 marker = list(symbol = ~cluster, color = ~cluster, size = 10, showscale = FALSE),
                 hovertemplate = "Country: %{text}<br>Cluster: %{marker.symbol}",
                 showlegend = F)

  return(fig)
}
```

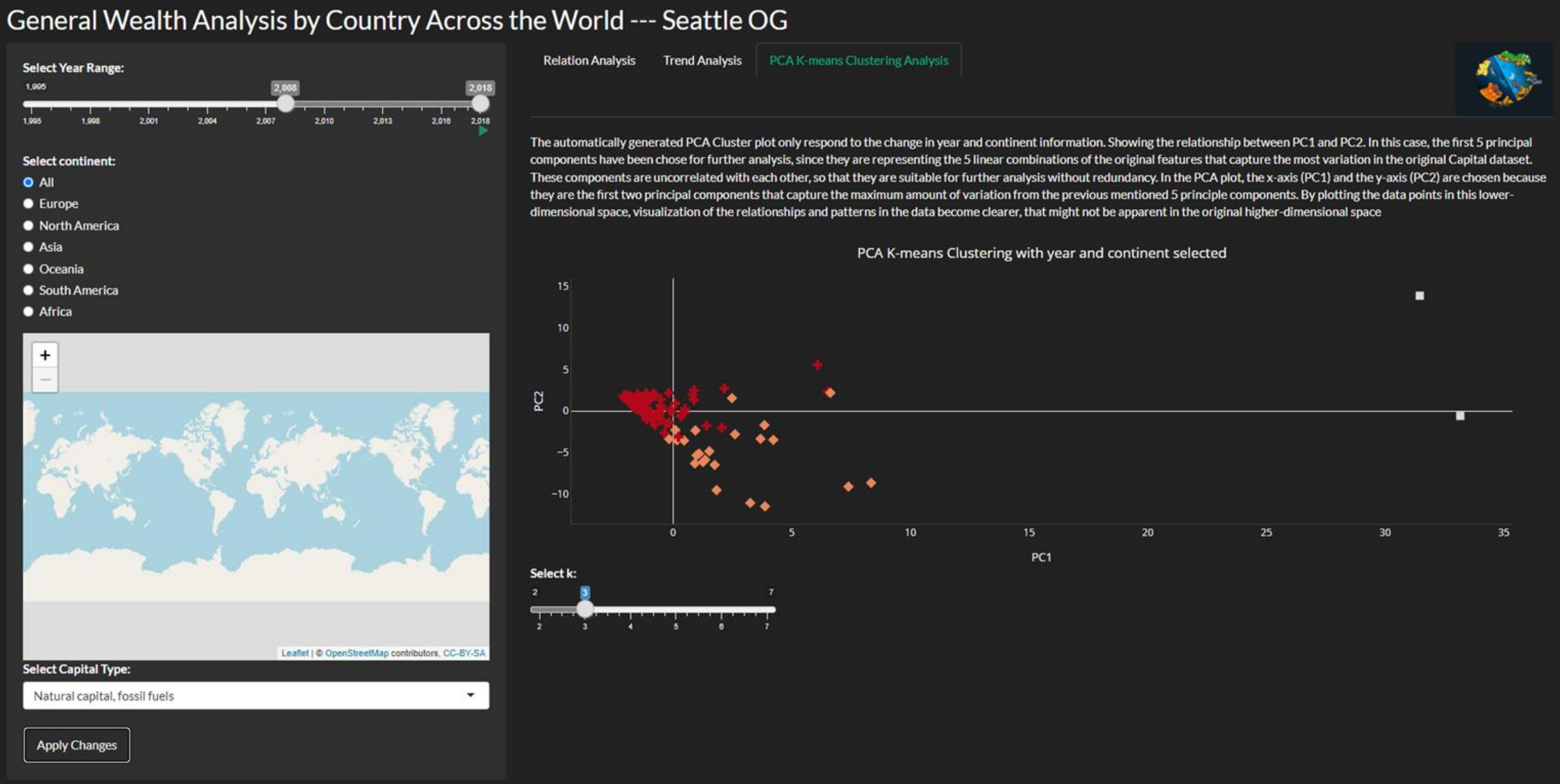
Result & Real World Examples



Result & Real World Examples



Result & Real World Examples



App

Showcase

Discussion & Conclusion

General wealth analysis is a process of examining the existing economic climate, identifying its strengths and weakness, and discovering the opportunities for improvement.A country or region that aspires to move forward is necessary to review past data and compare its performance with other countries or regions. It’s a way to summarize experiences and lessons, seek common ground while reserving differences, and take the essence while discarding the dross. It is precisely for this reason that the app we have designed has a very high degree of practicality.

The R shiny app provides a comprehensive analysis of the economic situation of almost all countries in the world over the past 23 years. The app also takes into account environmental factors like agriculture and fishery, which is an important aspect of economic analysis.

As we move towards a more digital and data-driven world, tools like our R shiny app can help us make more informed decisions and policies based on accurate and relevant information. However, it's important to keep in mind that there are **potential limitations** to the data and analysis provided by our app. For example, the dataset may not capture all aspects of a country's economic situation, and there may be factors that are not easily quantifiable or measurable. Additionally, the use of technology and data can have potential negative consequences, such as privacy concerns or biases in the data.

By using technology to analyze and visualize data, we can gain a better understanding of the economic situation of countries around the world and make more informed decisions for a sustainable future.

Contributions & Reference

https://databank.worldbank.org/home: Original data set and reports collected

https://chat.openai.com: Definitions of some terms, R functions info