



Interactive Graphs

陳曾基

國立陽明交通大學醫學院醫務管理研究所
國立陽明交通大學醫學院急重症醫學研究所
國立陽明交通大學醫學院醫學系家庭醫學科
臺北榮民總醫院家庭醫學部
臺北榮民總醫院醫學研究部大數據中心

課前作業

```
# 上課前請先安裝套件 : shiny DT plotly leaflet  
dygraphs zoo tesseract opencv
```

```
install.packages(c("shiny", "DT", "plotly", "leaflet",  
"dygraphs", "zoo", "tesseract", "opencv"))
```

```
# 需另外安裝
```

```
library(tesseract)
```

```
tesseract_download("chi_tra")
```

```
# 先載入tesseract套件 再利用其內的函數安裝資料
```

Topics

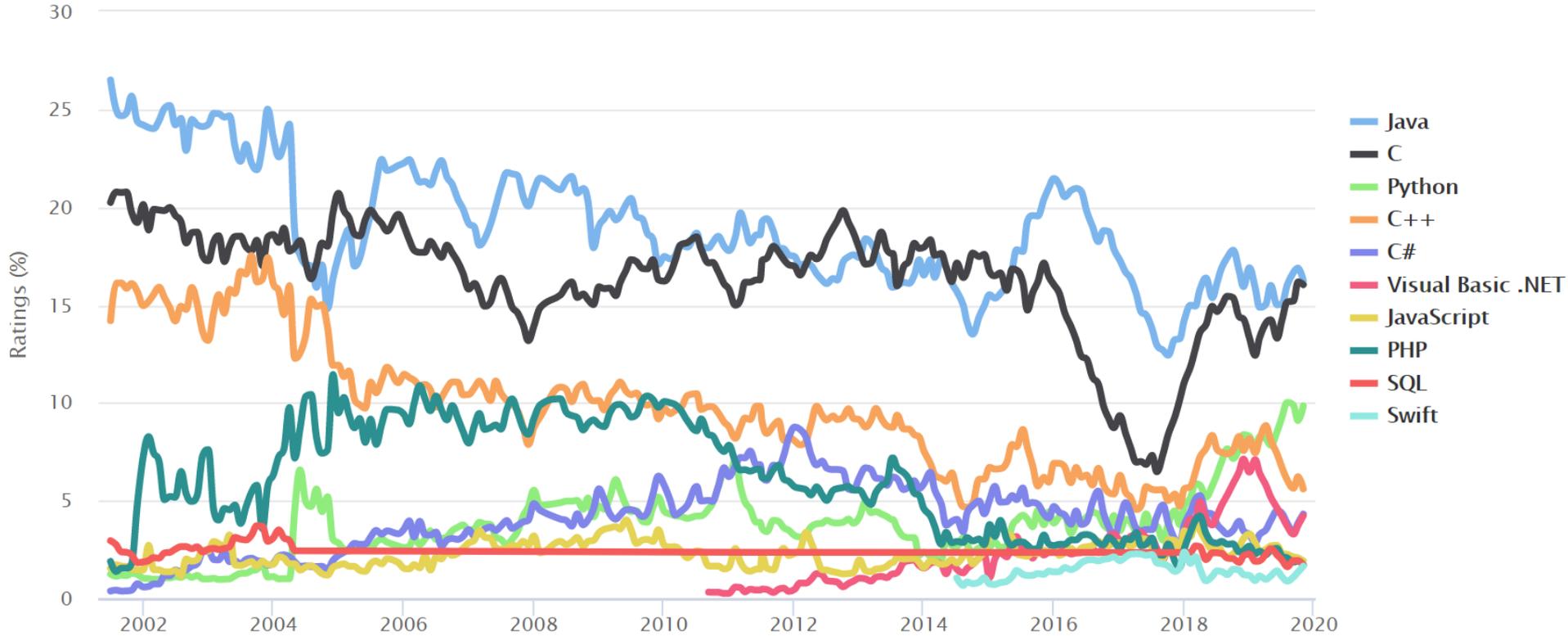
- Shiny
- Plotly for ggplot2
- htmlwidgets
- R for AI
- Resources

Interactive Data Visualization

- Users can manipulate and explore graphical representations of data directly.
- Common techniques:
 - Brushing
 - Painting
 - Identifying
 - Scaling
 - Linking

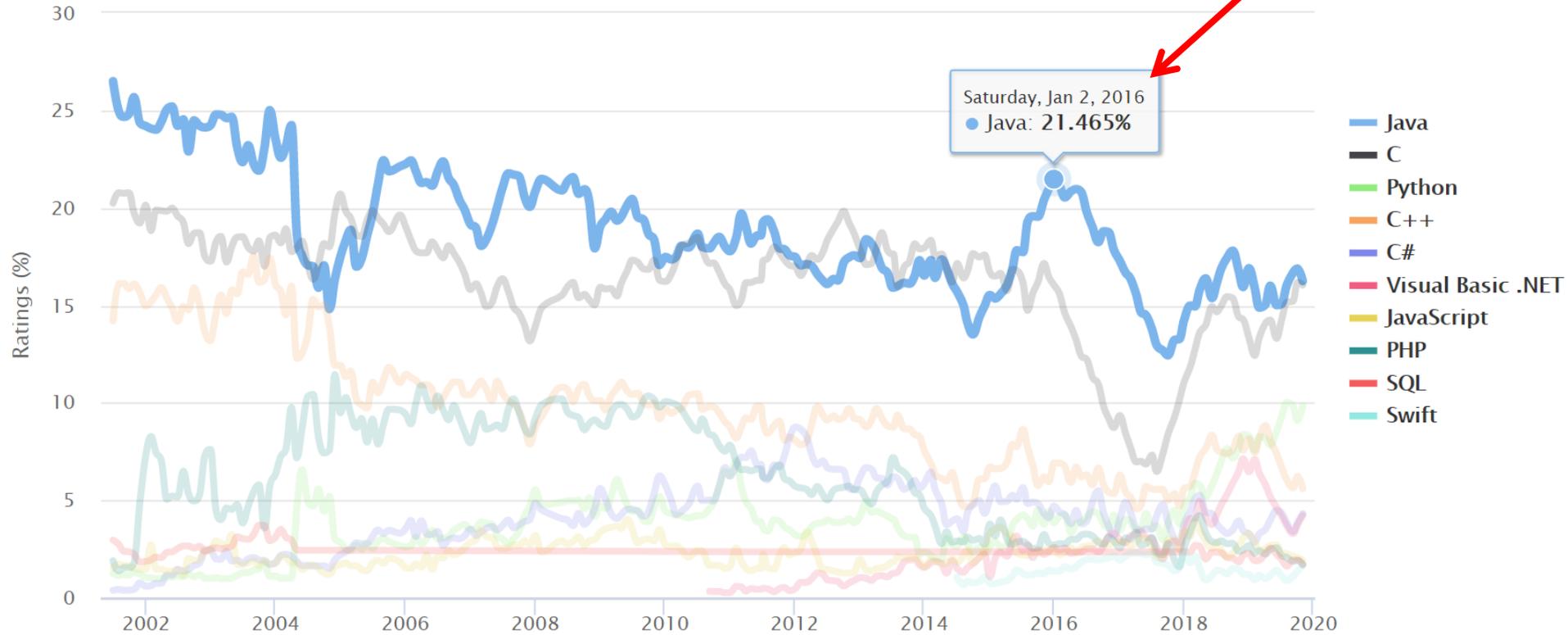
TIOBE Programming Community Index

Source: www.tiobe.com



TIOBE Programming Community Index

Source: www.tiobe.com



首頁 » 台股 » 財務分析 » 損益表

- 個股概覽
- 技術分析
- 基本分析
- 個股新聞
- 基本資料
- 股利政策
- 重大行事曆
- 營收盈餘
- 轉投資
- 本益比
- 淨值比
- 籌碼分析
- 籌碼K線

台泥 (1101) 1101 立即追蹤 模擬下單 資料日期：2019-04-03

股價	漲跌	漲幅	成交量	股本(百萬)	產業	發行權證檔數
41.7	↑0.3	0.72%	19,031	53,081	傳產 - 水泥 (上市)	52

[季合併損益表\(單季\)](#)
[年合併損益表](#)
[季合併損益表\(累計\)](#)

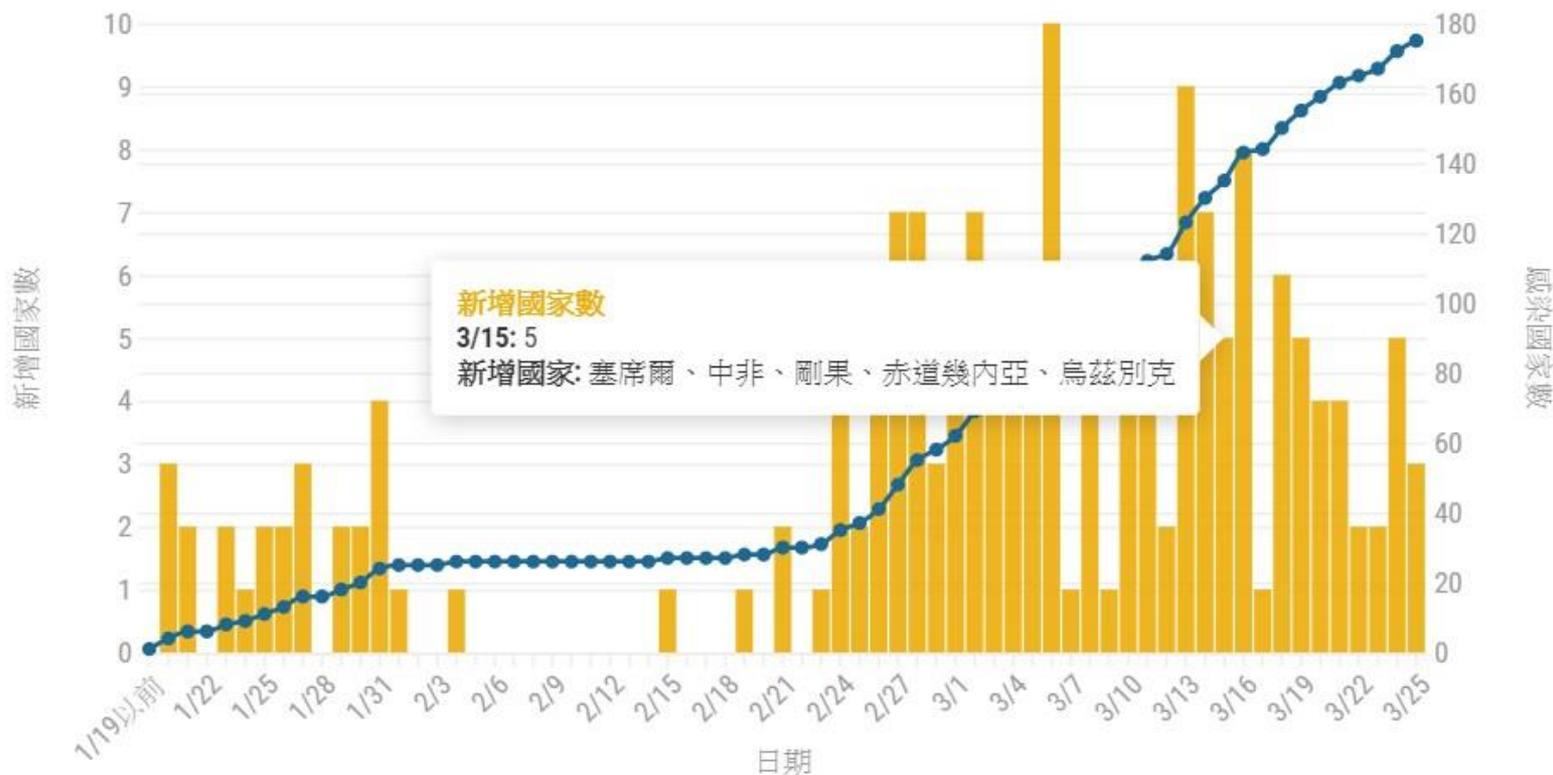




全球武漢肺炎感染國家數

截至3/25 全球共175國淪陷

● 感染國家數 ● 新增國家數



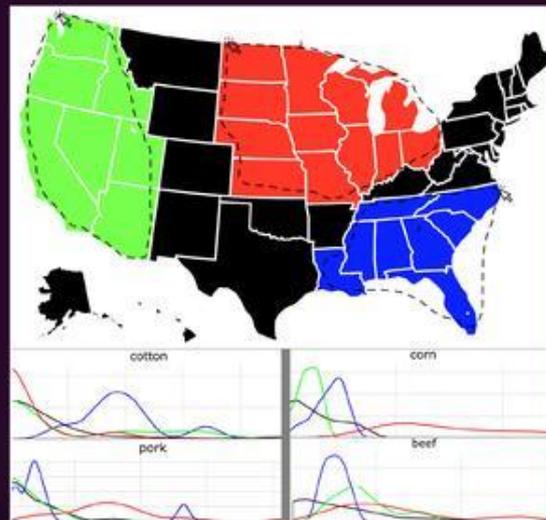
Source: 蘋果新聞網 · FB: 壹顆蘋果



<https://www.crcpress.com/Interactive-Web-Based-Data-Visualization-with-R-plotly-and-shiny/Sievert/p/book/9781138331457>

The R Series

Interactive Web-Based Data Visualization with R, plotly, and shiny



Carson Sievert

 **CRC Press**
Taylor & Francis Group
A CHAPMAN & HALL BOOK

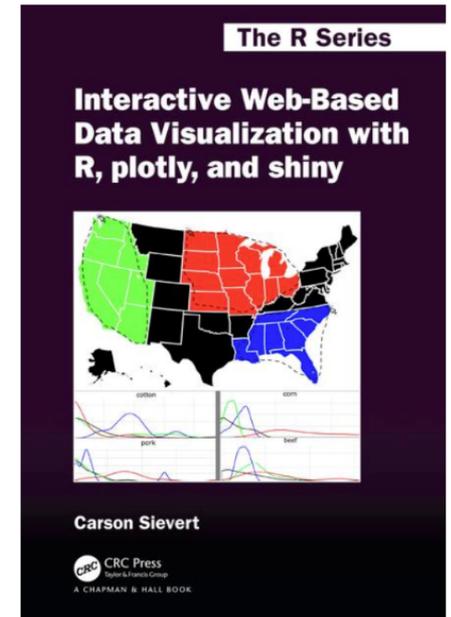
- Welcome
- 1 Preface
- I Creating views
- 2 Overview
- 3 Scattered foundations
- 4 Maps
- 5 Bars & histograms
- 6 Boxplots
- 7 2D frequencies
- 8 3D charts
- II Publishing views
- 9 Introduction
- 10 Saving and embedding HTML
- 11 Exporting static images
- 12 Editing views for publishing

Welcome



This is the website for “**Interactive web-based data visualization with R, plotly, and shiny**”. In this book, you’ll gain insight and practical skills for creating interactive and dynamic web graphics for data analysis from `R`. It makes heavy use of **plotly** for rendering graphics, but you’ll also learn about other `R` packages that augment a data science workflow, such as the **tidyverse** and **shiny**. Along the way, you’ll gain insight into best practices for visualization of high-dimensional data, statistical graphics, and graphical perception. By mastering these concepts and tools, you’ll impress your colleagues with your ability to *generate more informative, engaging, and repeatable interactive graphics* using free software that you can share over email, export to pdf/png, and more.

An online version of this book, available at <https://plotly-r.com/>, is free to use and is licensed under the [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 United States License](#). If you’d like a **physical copy** of the book, you can order it from [CRC Press](#) and [Amazon](#).

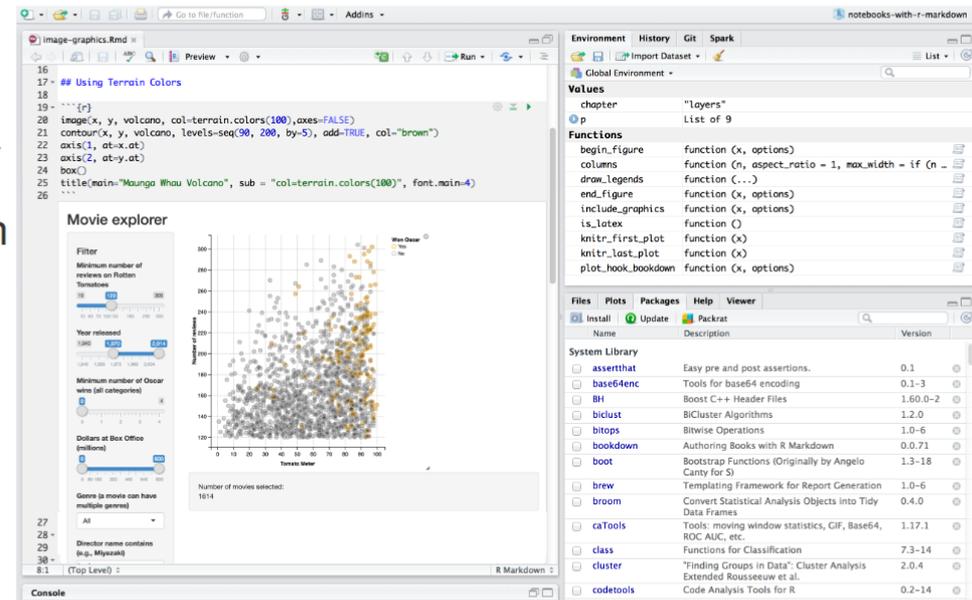


SECTION I

SHINY



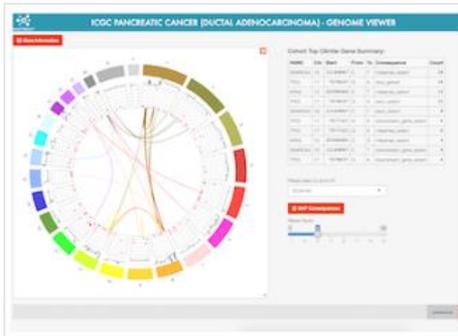
Shiny is an R package that makes it easy to build interactive web apps straight from R. You can host standalone apps on a webpage or embed them in [R Markdown](#) documents or build [dashboards](#). You can also extend your Shiny apps with [CSS themes](#), [htmlwidgets](#), and JavaScript actions.



Gallery

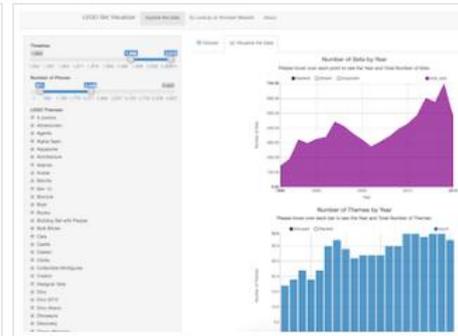
Shiny User Showcase

The Shiny User Showcase contains an inspiring set of sophisticated apps developed and contributed by Shiny users.



Genome browser

Paprr



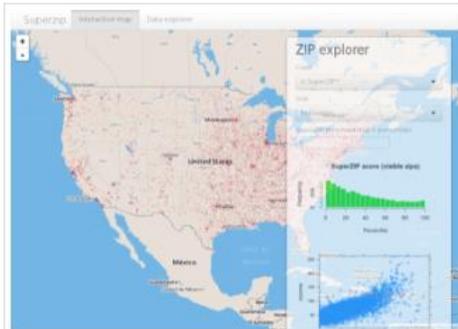
Lego Set Database Explorer



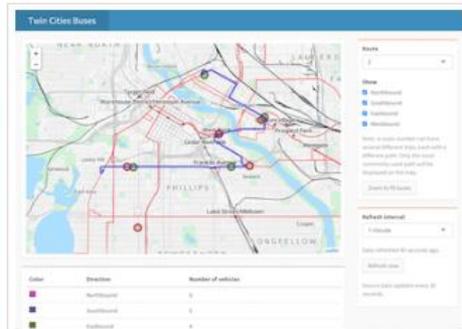
See more

Interactive visualizations

Shiny is designed for fully interactive visualization, using JavaScript libraries like d3, Leaflet, and Google Charts.



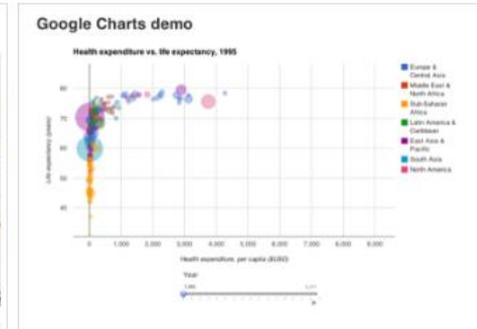
SuperZip example



Bus dashboard



Movie explorer



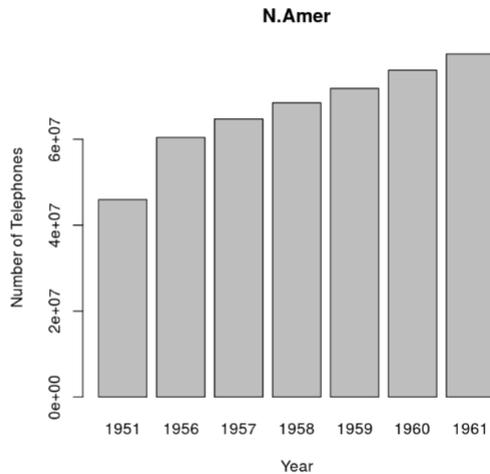
Google Charts

程式碼

Telephones by region

Region:
N.Amer

Data from AT&T (1961)
The World's
Telephones.



server.R

ui.R

show below

```
# Rely on the 'WorldPhones' dataset in the datasets
# package (which generally comes preloaded).
library(datasets)

# Define a server for the Shiny app
function(input, output) {

  # Fill in the spot we created for a plot
  output$phonePlot <- renderPlot({

    # Render a barplot
    barplot(WorldPhones[,input$region]*1000,
            main=input$region,
            ylab="Number of Telephones",
            xlab="Year")

  })
}
```

Part 1 - How to build a Shiny app

1. [Introduction](#)
2. [R](#)
3. [App architecture](#)
4. [App template](#)
5. [Inputs and outputs](#)
6. [The server function](#)
7. [Sharing apps](#)
8. [Shinyapps.io](#)
9. [Shiny servers](#)
10. [Recap - Part 1](#)

Part 2 - How to customize reactions

11. [Introduction](#)
12. [Review of Part 1](#)
13. [Reactivity](#)
14. [Reactive values](#)
15. [Reactive functions](#)
16. [render*\(\)](#)
17. [reactive\(\)](#)
18. [isolate\(\)](#)
19. [observeEvent\(\)](#)
20. [eventReactive\(\)](#)
21. [reactiveValues\(\)](#)
22. [Recap - Part 2](#)
23. [Parting tips](#)

Part 3 - How to customize appearance

24. [Introduction](#)
25. [Review of Parts 1 and 2](#)
26. [HTML UI](#)
27. [Adding static content](#)
28. [Building layouts](#)
29. [Panels and tabsets](#)
30. [Prepackaged layouts](#)
31. [CSS](#)
32. [Recap - Part 3](#)

No description, website, or topics provided.

474 commits

17 branches

0 packages

1 release

19 contributors

View license

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

 cpsievert Merge pull request #175 from rstudio/joe/178-delayed-widget

Latest commit ebd356e on 8 Oct

 001-hello	Convert to single-file app.	2 years ago
 002-text	Convert to single-file app.	2 years ago
 003-reactivity	Convert to single-file app.	2 years ago
 004-mpg	Convert to single-file app.	2 years ago
 005-sliders	Convert to single-file app.	2 years ago
 006-tabsets	Convert to single-file app.	2 years ago
 007-widgets	Convert to single-file app.	2 years ago
 008-html	Convert to single-file app.	2 years ago

顯示內建示範圖例

- 安裝與載入 shiny 套件後，一一執行範例

```
if (!require(shiny)) install.packages("shiny")
```

```
library(shiny)
```

```
runExample("01_hello")
```

```
runExample("02_text") # 資料表格
```

```
runExample("03_reactivity") # 回應設計 + 資料表格
```

```
runExample("04_mpg") # 繪圖
```

```
runExample("05_sliders") # 側欄工具 + 表格
```

```
runExample("06_tabsets") # tab 頁面
```

```
runExample("07_widgets") # 行為按鈕
```

```
runExample("08_html") # 結合 html
```

```
runExample("09_upload") # 上傳檔案工具
```

```
runExample("10_download") # 下載檔案工具
```

```
runExample("11_timer") # 計時器
```

A **Shiny** app is a web page (**UI**) connected to a computer running a live R session (**Server**)



Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

在 RStudio 會跳出視窗顯示

按此可另開
瀏覽器顯示

app.R

```
library(shiny)

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

  # App title ----
  titlePanel("Hello Shiny!"),

  # Sidebar layout with input and output definitions ----
  sidebarLayout(

    # Sidebar panel for inputs ----
    sidebarPanel(

      # Input: Slider for the number of bins ----
      sliderInput(inputId = "bins",
                  label = "Number of bins:",
                  min = 1,
                  max = 50,
```

程式碼

另開瀏覽器顯示

Browser address: 127.0.0.1:6570

Number of bins: 1 30 50

Histogram of waiting times

Frequency

Waiting time to next eruption (in mins)

程式碼

```
app.R
library(shiny)
# Define UI for app that draws a histogram ----
ui <- fluidPage(
  # ...

```

Shiny 程式製作流程

- 挑選範例，複製程式碼供參考，修改程式碼
- 程式分為
 - 載入 shiny 套件
 - 匯入資料
 - 資料彙整計算，用以輸出圖形

- 前端介面控制 ui (ui.R 部分)
- 後端資料運算 server (server.R 部分)
- 產製 Shiny app : shinyApp(ui, server)

也可產製App供網頁與手機用



```
library(shiny)
ui <- fluidPage()
server <- function(input, output){}
shinyApp(ui = ui, server = server)
```

- **ui** - nested R functions that assemble an HTML user interface for your app
- **server** - a function with instructions on how to build and rebuild the R objects displayed in the UI
- **shinyApp** - combines ui and server into an app. Wrap with **runApp()** if calling from a sourced script or inside a function.

多條件篩選資料表

- 利用 `Shiny_BasicDatatable.r`
- 假設已先利用 `Import_tidyverse.r` 匯入資料，如果資料檔放置於其他位置，請更改程式碼內的檔案路徑
- 請在 RStudio 裡執行程式，也可在 RGui 裡執行
- 參考 <https://shiny.rstudio.com/gallery/basic-datatable.html>

程式碼置於 Shiny 子目錄

Basic DataTable

Manufacturer: Transmission: Cylinders:

Show entries Search:

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2	2008	4	manual(m6)	4	20	28	p	compact

```

server.R ui.R

# Load the ggplot2 package which provides
# the 'mpg' dataset.
library(ggplot2)

function(input, output) {

  # Filter data based on selections
  output$table <- DT::renderDataTable(DT::datatable({
    data <- mpg
    if (input$man != "All") {
      data <- data[data$manufacturer == input$man,]
    }
    if (input$cyl != "All") {
      data <- data[data$cyl == input$cyl,]
    }
    if (input$trans != "All") {
      data <- data[data$trans == input$trans,]
    }
    data
  })))
}

```

server.R

ui.R

```
# Load the ggplot2 package which provides
# the 'mpg' dataset.
library(ggplot2)

function(input, output) {

  # Filter data based on selections
  output$table <- DT::renderDataTable(DT::datatable({
    data <- mpg
    if (input$man != "All") {
      data <- data[data$manufacturer == input$man,]
    }
    if (input$cyl != "All") {
      data <- data[data$cyl == input$cyl,]
    }
    if (input$trans != "All") {
      data <- data[data$trans == input$trans,]
    }
    data
  }))
}
```

server.R

ui.R

```
# Load the ggplot2 package which provides
# the 'mpg' dataset.
library(ggplot2)

fluidPage(
  titlePanel("Basic DataTable"),

  # Create a new Row in the UI for selectInputs
  fluidRow(
    column(4,
      selectInput("man",
        "Manufacturer:",
        c("All",
          unique(as.character(mpg$manufacturer))))
    ),
    column(4,
      selectInput("trans",
        "Transmission:",
        c("All",
          unique(as.character(mpg$trans))))
    ),
    column(4,
      selectInput("cyl",
        "Cylinders:",
        c("All",
          unique(as.character(mpg$cyl))))
    )
  ),
  # Create a new row for the table.
  DT::dataTableOutput("table")
)
```

```
# 更改網站上的程式碼  
# 紅字為更改處  
# Shiny_BasicDatatable.r
```

```
if (! require(tidyverse)) { install.packages("tidyverse"); require(tidyverse) }  
if (! require(shiny)) { install.packages("shiny"); require(shiny) }
```

```
# 匯入資料集cd 可視CD2009.DAT實際擺放位置 更動路徑  
# .....  
# .....  
# 假設已先利用Import_tidyverse.r匯入資料 => 此匯入步驟可省略  
# 否則依Shiny_BasicDatatable.r程式碼，匯入CD2009.DAT
```

```

# Define UI for random distribution app ----
ui <- fluidPage(
  titlePanel("Basic DataTable"),

  # Create a new Row in the UI for selectInputs
  fluidRow(
    column(4,
      selectInput("feeym",
        "Year-Month:",
        c("All",
          sort(unique(as.character(cd$FEE_YM)))))),
    ),
    column(4,
      selectInput("specialty",
        "Specialty:",
        c("All",
          sort(unique(as.character(cd$FUNC_TYPE)))))),
    ),
    column(4,
      selectInput("idsex",
        "Gender:",
        c("All",
          sort(unique(as.character(cd$ID_SEX)))))),
    )
  ),

  # Create a new row for the table.
  DT::dataTableOutput("table")
)

```

```

# Define server logic for random distribution app ----
server <- function(input, output) {
  # Filter data based on selections
  output$table <- DT::renderDataTable(DT::datatable({
    data <- cd
    if (input$feeym != "All") {
      data <- data[data$FEE_YM == input$feeym,]
    }
    if (input$specialty != "All") {
      data <- data[data$FUNC_TYPE == input$specialty,]
    }
    if (input$idsex != "All") {
      data <- data[data$ID_SEX == input$idsex,]
    }
    data
  )))
}

```

```

# Create Shiny app ----
shinyApp(ui, server)

```

在 RStudio 會跳出視窗顯示

R -- Shiny
http://127.0.0.1:15551 | Open in Browser | Publish

Basic DataTable

Year-Month: Specialty: Gender:

Show entries Search:

	FEE_YM	APPL_TYPE	HOSP_ID	APPL_DATE	CASE_TYPE	SEQ_NO	CURE_ITEM_NO1	CURE_ITEM_NO2	CURE_ITEM_NO3	CURE_ITEM_NO4	FUNC_TYPE	FUNC_DATE	TREAT_END_DATE	
1	200901	1	0000000000000000000000000000000017430	20090204	01	3866					04	20090131		1
2	200901	1	0000000000000000000000000000000018622	20090212	01	247					11	20090105		1
3	200901	1	0000000000000000000000000000000018622	20090212	01	919					11	20090109		1
4	200901	1	0000000000000000000000000000000018622	20090212	01	752					11	20090114		1
5	200901	1	0000000000000000000000000000000018622	20090212	01	844					11	20090121		1
6	200901	1	0000000000000000000000000000000018622	20090212	01	852					11	20090126		1
7	200901	1	000000000000000000000000000000004734	20090209	01	6542					04	20090126		2
8	200901	1	0000000000000000000000000000000017745	20090207	09	389	D0				06	20090108	20090113	1
9	200901	1	0000000000000000000000000000000017745	20090207	09	369					06	20090111		1
10	200901	1	0000000000000000000000000000000017089	20090211	08	3196	33	92			10	20081221	20090119	1

Showing 1 to 10 of 598,574 entries

Previous 2 3 4 5 ... 59858 Next

另開瀏覽器顯示

The screenshot shows a web browser window titled "Basic DataTable" with the URL "127.0.0.1:5551". The browser's address bar contains navigation icons and a search bar. Below the browser window, there is a web application interface with three filter dropdowns: "Year-Month:" (set to "All"), "Specialty:" (set to "All"), and "Gender:" (set to "All"). To the left of the table is a "Show 10 entries" selector, and to the right is a "Search:" input field. The table itself has columns for "FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE", "SEQ_NO", "CURE_ITEM_NO1", and "CURE_ITEM_NO2". It contains 10 rows of data. At the bottom of the table, there is a pagination bar showing "Showing 1 to 10 of 598,574 entries" and a set of navigation buttons: "Previous", "1", "2", "3", "4", "5", "...", "59858", and "Next".

	FEE_YM	APPL_TYPE	HOSP_ID	APPL_DATE	CASE_TYPE	SEQ_NO	CURE_ITEM_NO1	CURE_ITEM_NO2
1	200901	1	0017430	20090204	01	3866		
2	200901	1	0018622	20090212	01	247		
3	200901	1	0018622	20090212	01	919		
4	200901	1	0018622	20090212	01	752		
5	200901	1	0018622	20090212	01	844		
6	200901	1	0018622	20090212	01	852		
7	200901	1	004734	20090209	01	6542		
8	200901	1	0017745	20090207	09	389	D0	
9	200901	1	0017745	20090207	09	369		
10	200901	1	0017089	20090211	08	3196	33	92

Showing 1 to 10 of 598,574 entries

Previous 1 2 3 4 5 ... 59858 Next

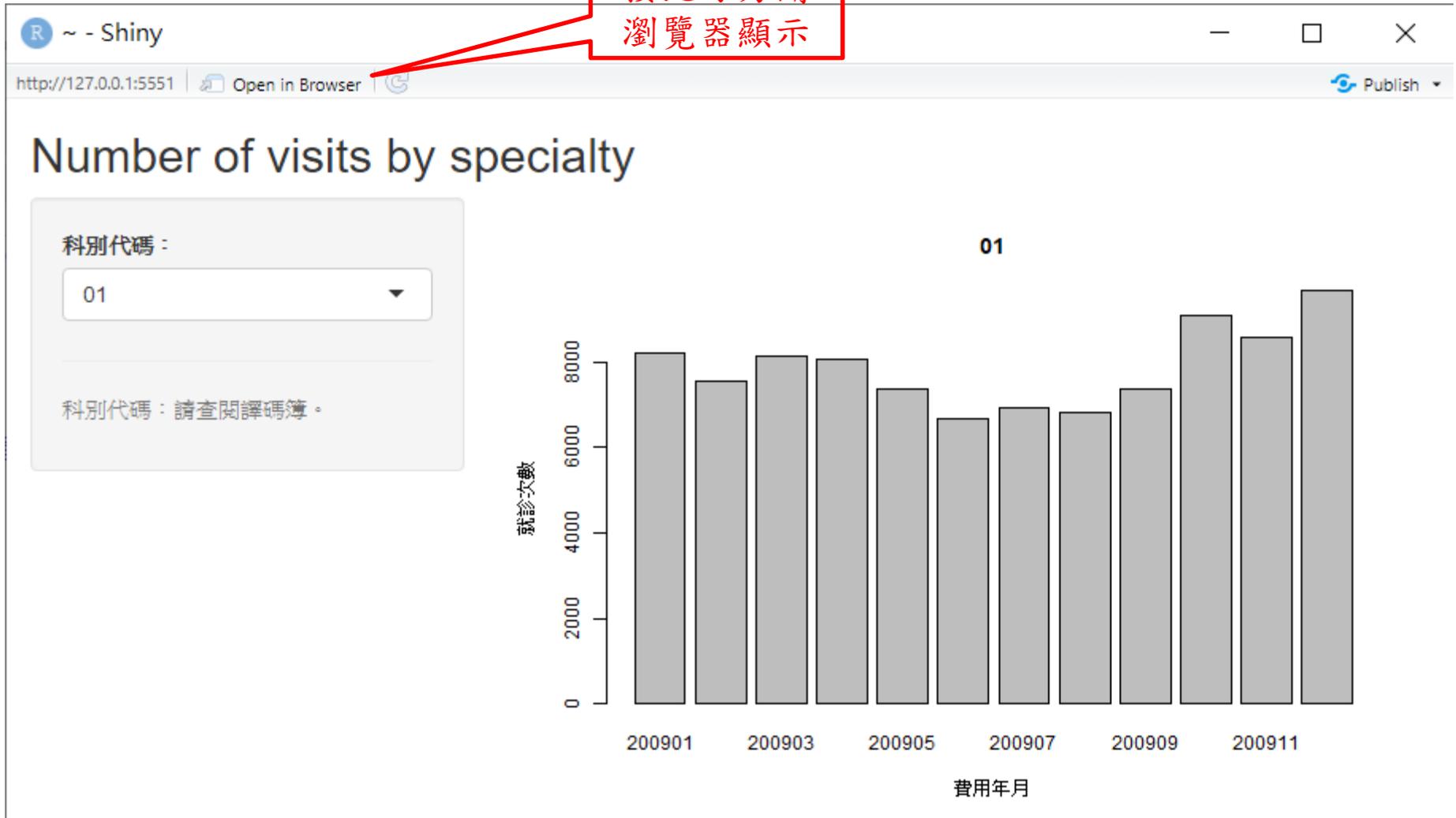
Bar Chart 長條圖

- 利用 `Shiny_BarChart.r`
- 假設已先利用 `Import_tidyverse.r` 匯入資料，如果資料檔放置於其他位置，請更改程式碼內的檔案路徑
- 參考 <https://shiny.rstudio.com/gallery/telephones-by-region.html>
- `Shiny_BarChart_WithChineseNames.r` 裡，將科別代碼改成中文

程式碼置於 Shiny 子目錄

在 RStudio 會跳出視窗顯示

按此可另開
瀏覽器顯示



另開瀏覽器顯示

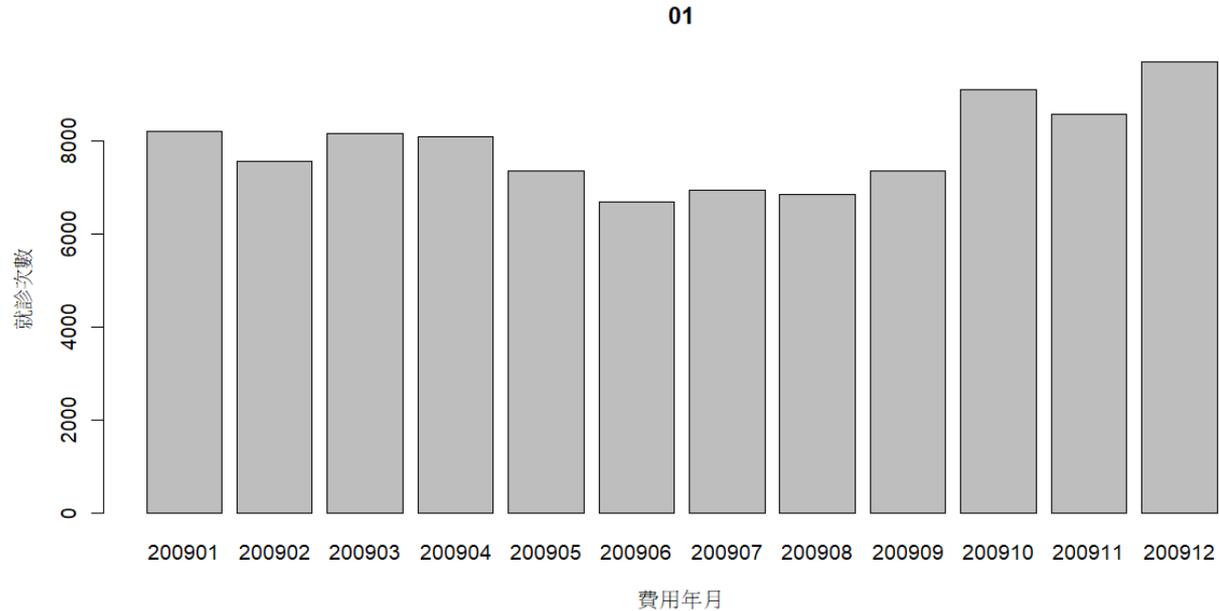


Number of visits by specialty

科別代碼：

01

科別代碼：請查閱譯碼簿。



科別代碼以中文顯示

R ~ - Shiny

http://127.0.0.1:5551

Open in Browser



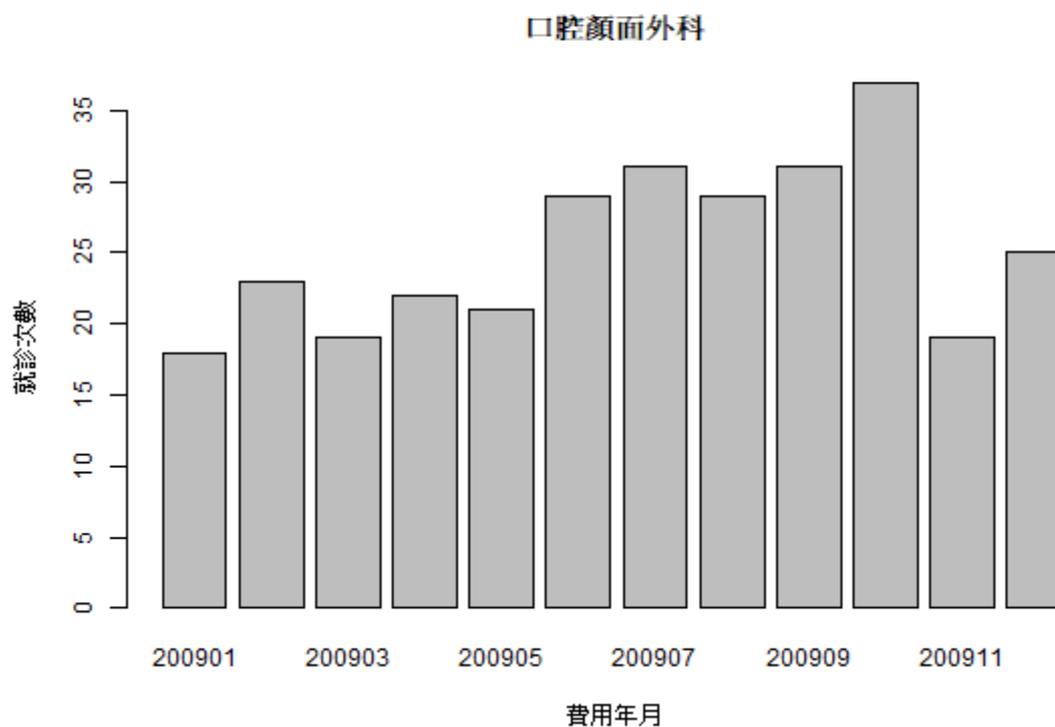
Publish

Number of visits by specialty

科別代碼：

口腔顏面外科

請由上方挑選科別。



Shiny Widgets Gallery

For each widget below, the Current Value(s) window displays the value that the widget provides to shinyServer. Notice that the values change as you interact with the widgets.

Action button

Action

Current Value:

```
[1]  $\emptyset$   
attr(,"class")  
[1] "integer" "shinyActionButtonValue"
```

See Code

Single checkbox

Choice A

Current Value:

```
[1] TRUE
```

See Code

Checkbox group

Choice 1
 Choice 2
 Choice 3

Current Values:

```
[1] "1"
```

See Code

Date input

2014-01-01

Current Value:

```
[1] "2014-01-01"
```

See Code

Date range

2019-11-21 to 2019-11-21

Current Values:

```
[1] "2019-11-21" "2019-11-21"
```

See Code

File input

Browse... No file selected

Current Value:

```
NULL
```

See Code

* 有12種元件及其程式碼可供參考

Mastering Shiny

Table of contents

Welcome

Preface

Getting started

Introduction

1 Your first Shiny app

2 Basic UI

3 Basic reactivity

4 Case study: ER injuries

Shiny in action

Introduction

5 Workflow

6 Layout, themes, HTML

7 Graphics

8 User feedback

Welcome

This is the online version of *Mastering Shiny*, a book **currently under early development** and intended for a late 2020 release by O'Reilly Media.

Shiny is a framework for creating web applications using R code. It is designed primarily with data scientists in mind, and to that end, you can create pretty complicated Shiny apps with no knowledge of HTML, CSS, or JavaScript. On the other hand, Shiny doesn't limit you to creating trivial or prefabricated apps: its user interface components can be easily customized or extended, and its server uses reactive programming to let you create any type of back end logic you want. Shiny is designed to feel almost magically easy when you're getting started, and yet the deeper you get into how it works, the more you realize it's built out of general building blocks that have strong software engineering principles behind them.



SECTION II

PLOTLY FOR GGPLOT2

Search...

Plotly ggplot2 Library

With `ggplotly()` by Plotly, you can convert your `ggplot2` figures into interactive ones powered by `plotly.js`, ready for embedding into Dash applications.

Building AI apps or dashboards in R? Deploy them to [Dash Enterprise](#) for hyper-scalability and pixel-perfect aesthetic.

10% of the Fortune 500 uses Dash Enterprise to productionize AI & data science apps.

[Find out if your company is using Dash Enterprise](#)

[Install Dash Enterprise on Azure](#) | [Install Dash Enterprise on AWS](#)

Quick Start

Getting Started

GitHub

community.plotly.com

Examples

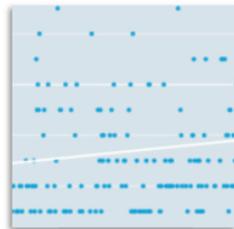
Basic Charts

Statistical Charts

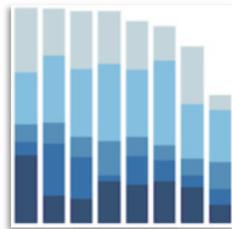
Animations

Layout Options

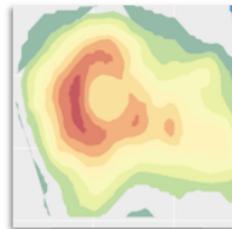
Basic Charts



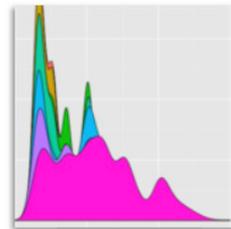
geom_abline



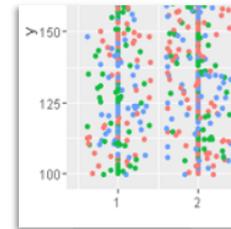
geom_bar



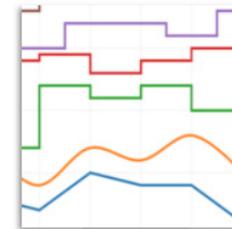
geom_contour



geom_density



geom_jitter



geom_line

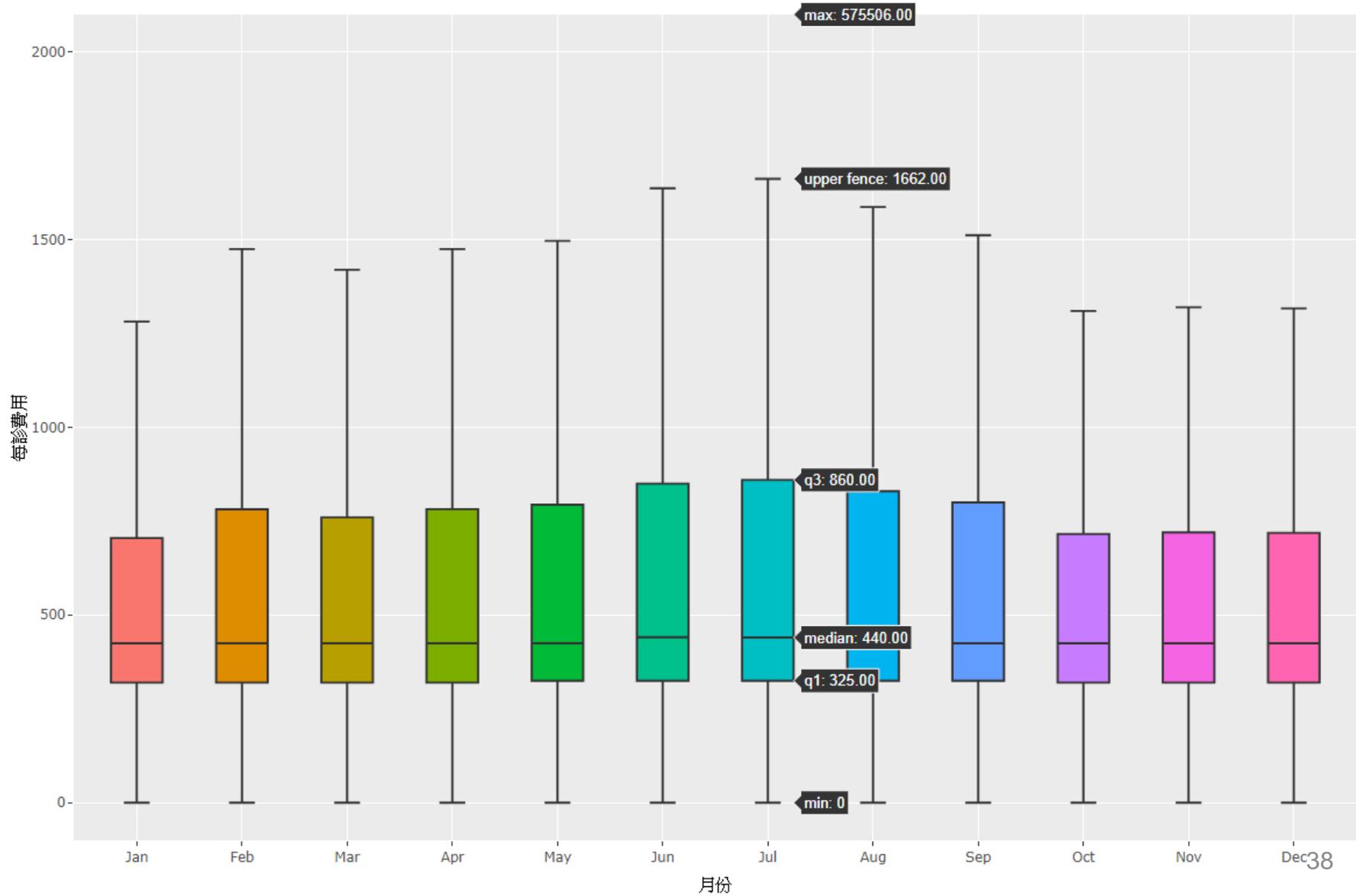
Box Plot 箱形圖(盒鬚圖)

- 利用 `Plotly_ggplot2_Boxplot.r`
- 假設已先利用 `Import_tidyverse.r` 匯入資料，如果資料檔放置於其他位置，請更改程式碼內的檔案路徑
- 參考 <https://plotly.com/ggplot2/box-plots/>

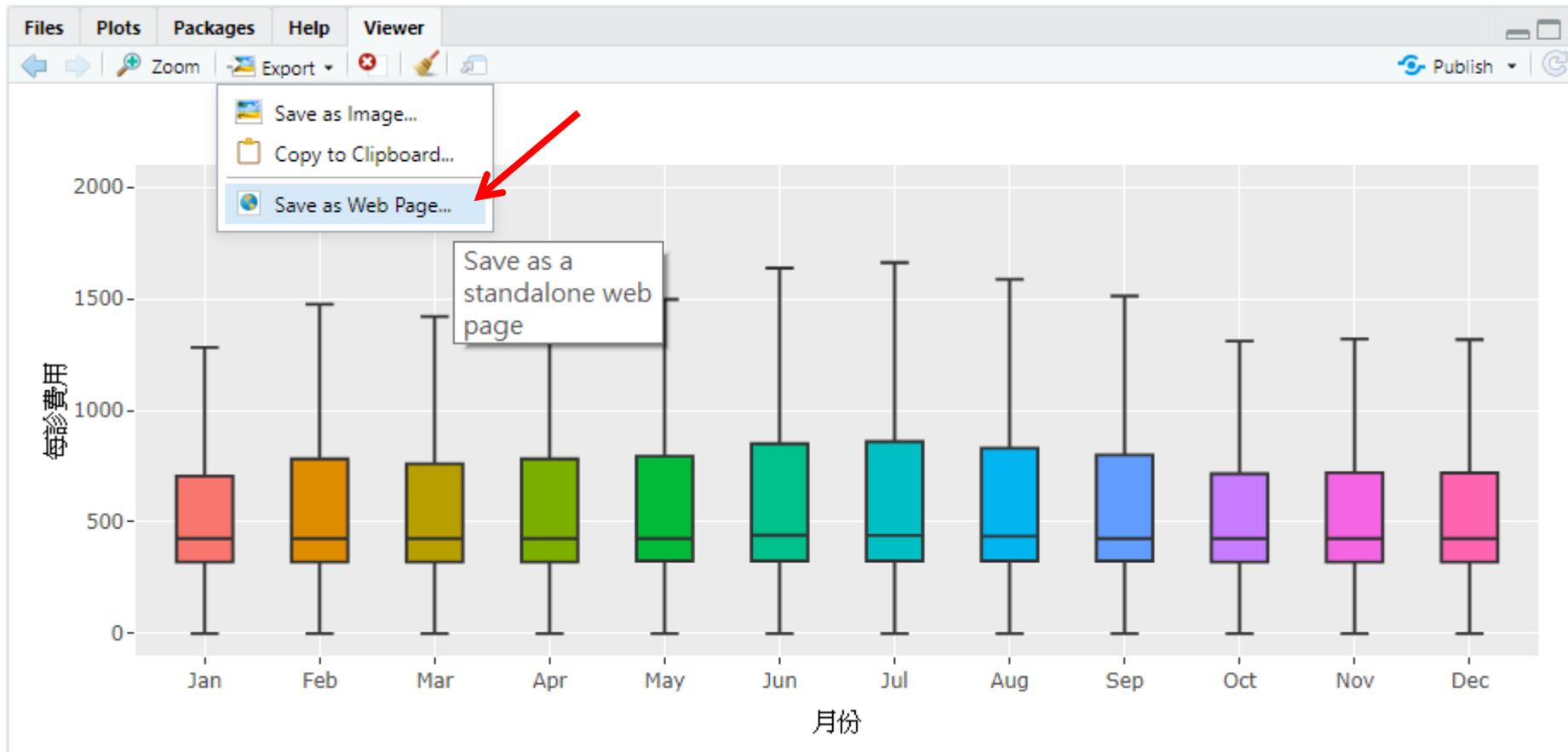
程式碼置於 `Plotly_ggplot2` 子目錄

在 RStudio 顯示 (Viewer)

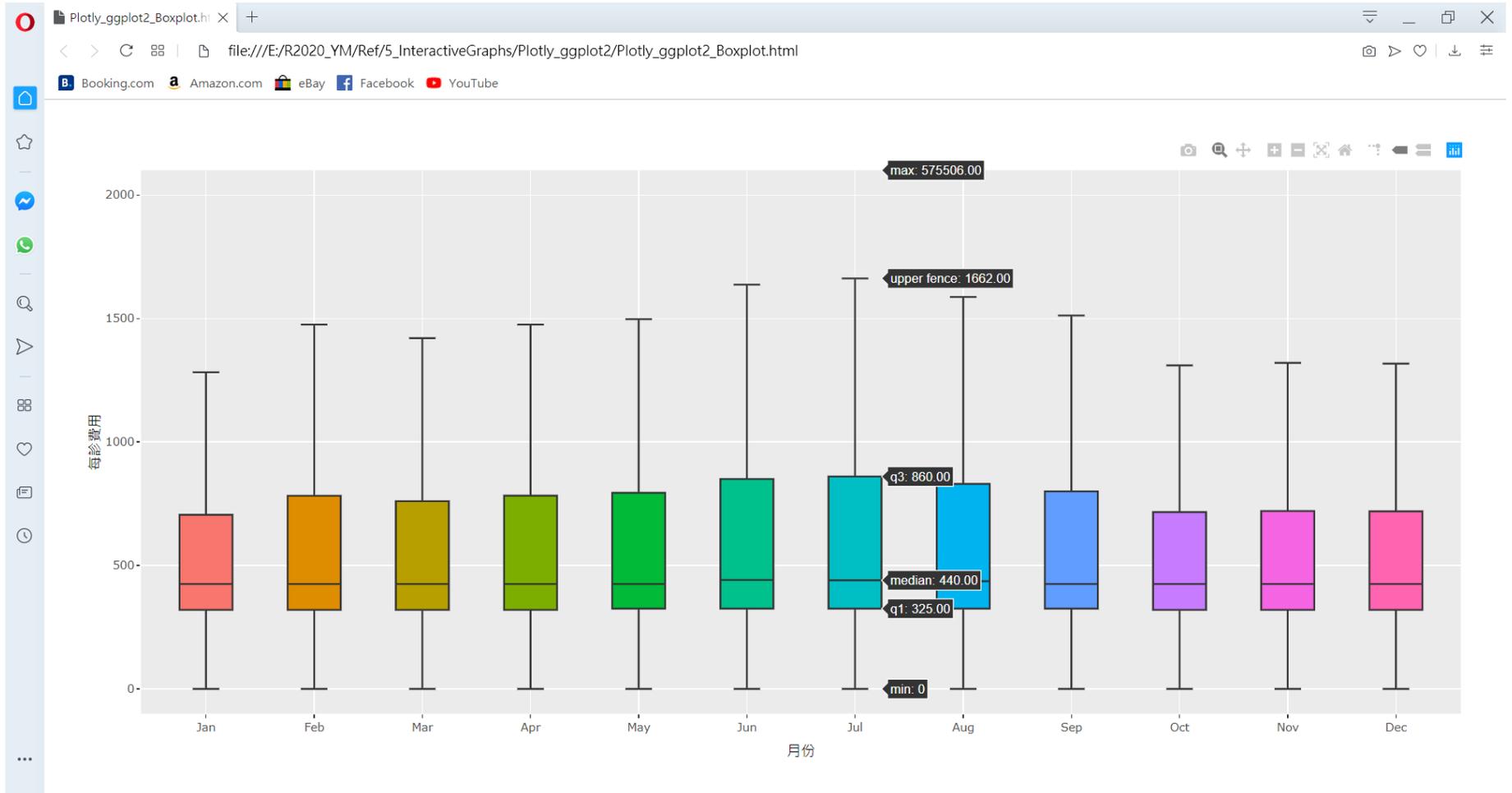
Viewer Zoom



可存成網頁(內含互動式圖形)



在瀏覽器裡顯示網頁



SECTION III

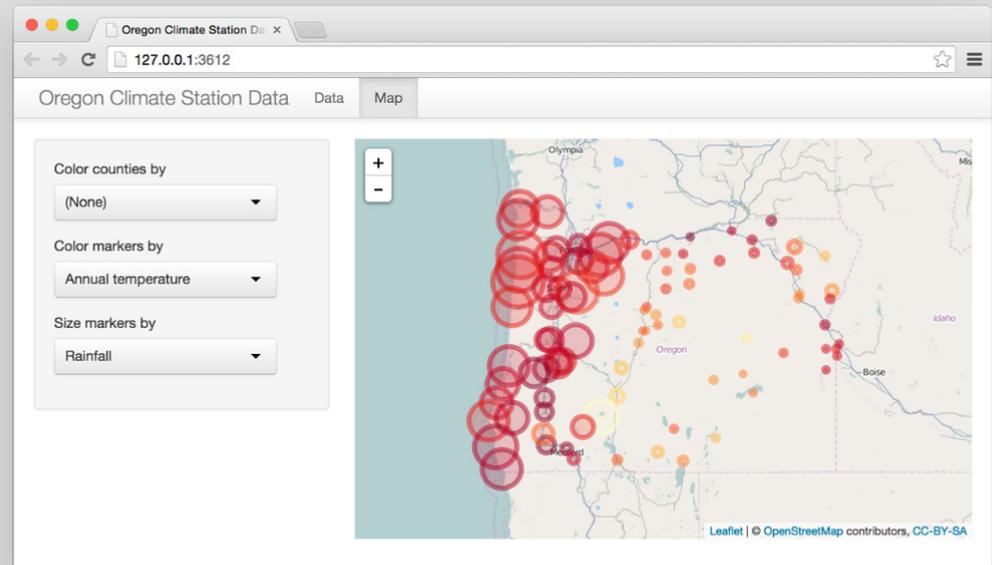
HTMLWIDGETS

Bring the best of JavaScript data visualization to R

Use JavaScript visualization libraries at the R console, just like plots

Embed widgets in R Markdown documents and Shiny web applications

Develop new widgets using a framework that seamlessly bridges R and JavaScript



At the R console

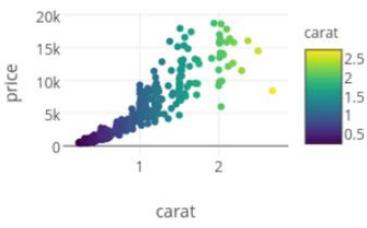
In R Markdown docs

In Shiny apps

126 registered widgets available to explore

Sort: Github stars
Text Filter: search name, author, desc
Author Filter
Tag Filter
CRAN Only:

Showing 66 of 126



price

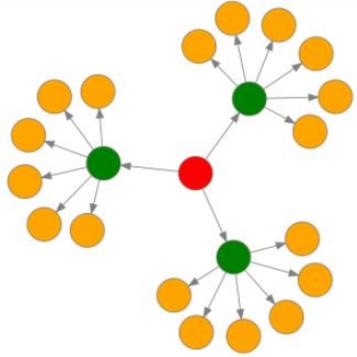
carat

carat

plotly  1609

Create interactive web graphics via Plotly's JavaScript graphing library.

- author: cpsievert
- tags: d3, webgl
- js libraries: plotly.js



DiagrammeR  1281

Easily create graph diagrams using R.

- author: rich-iannone
- tags: visualization, diagram, networks
- js libraries: d3.viz, mermaid



leaflet  578

Leaflet is an open-source JavaScript library for interactive maps. This package makes it easy to create Leaflet maps from R.

- author: rstudio
- tags: visualization, maps
- js libraries: leaflet

id	name	age	grade	test1_score	test2_score
1	Bob	28	C	8.9	9.1
2	Ashley	27	A	9.5	9.1
3	James	20	A	9.6	9.2
4	David	28	C	8.9	9.1
5	Jenny	29	B	9.1	8.9
6	Hans	29	B	9.3	8.5
7	Leo	27	B	9.3	9.2
8	John	27	A	9.9	9.3
9	Emily	31	C	8.5	9.1
10	Lee	30	C	8.6	8.8

formattable  517

Formattable data structures.

- author: renkun-ken
- tags: table
- js libraries:

htmlwidgets

- 一種簡便的框架，讓 R 可以使用 JavaScript library
- 讓 JavaScript 資料視覺化的套件能在 R 中使用
- 可以輸出為單獨的網頁

- widget : 控制項 / 控制元件
([https://en.wikipedia.org/wiki/Widget_\(GUI\)](https://en.wikipedia.org/wiki/Widget_(GUI)))

htmlwidgets

- **Leaflet** : geo-spatial mapping
- **dygraphs** : time series charting
- **Plotly** : interactive graphics with **D3**
- **rbokeh** : R interface to Bokeh
- **Highcharter** : R interface to Highcharts
- **visNetwork** : graph data visualization with vis.js
- **networkD3** : graph data visualization with **D3**
- **DataTables** : tabular data display
- **threejs** : 3D scatterplots and globes
- **rglwidget** : render scenes created with rgl
- **DiagrammeR** : diagrams and flowcharts
- **MetricsGraphics** : scatterplots and line charts with **D3**

Data-Driven Documents



Like visualization and creative coding? Try interactive JavaScript notebooks in **Observable!**

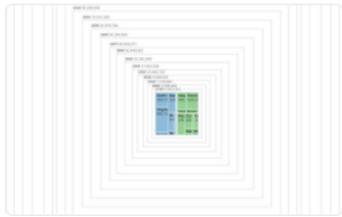
D3.js is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

Gallery

Looking for a good D3 example? Here's a few (okay, 168...) to peruse.

Animation

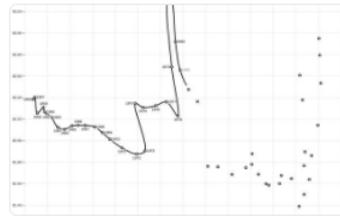
D3's [data join](#), [interpolators](#), and [easings](#) enable flexible [animated transitions](#) between views while preserving [object constancy](#).



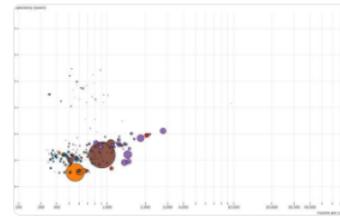
Animated treemap



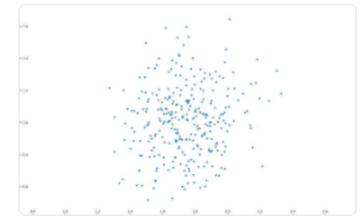
Temporal force-directed graph



Connected scatterplot



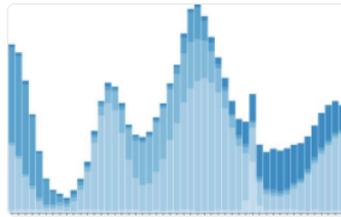
The wealth & health of nations



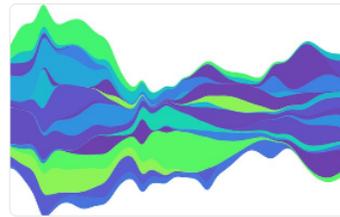
Scatterplot tour



Bar chart race



Stacked-to-grouped bars



Streamgraph transitions



Smooth zooming



Zoom to bounding box

SECTION III-A

LEAFLET FOR R

Leaflet for R

Fork me on GitHub

Introduction

The Map Widget

Basemaps

Markers

Popups and Labels

Lines and Shapes

GeoJSON and TopoJSON

Raster Images

Shiny Integration

Introduction

[Leaflet](#) is one of the most popular open-source JavaScript libraries for interactive maps. It's used by websites ranging from [The New York Times](#) and [The Washington Post](#) to [GitHub](#) and [Flickr](#), as well as GIS specialists like [OpenStreetMap](#), [Mapbox](#), and [CartoDB](#).

This R package makes it easy to integrate and control Leaflet maps in R.

Features

- Interactive panning/zooming
- Compose maps using arbitrary combinations of:
 - Map tiles
 - Markers
 - Polygons
 - Lines
 - Popups
 - GeoJSON

Interactive Map 互動式地圖

```
if (!require(leaflet)) install.packages("leaflet")
library(leaflet)
leaflet() %>%
  addTiles() %>%
  addMarkers(lng = 121.515504, lat = 25.122318, popup
= "上課地點")
# 可先用 Google 地圖找出欲顯示地點的經緯度
# 預設地圖為 OpenStreetMap
# 請在 RStudio 裡執行程式，如果以 RGui 執行，會開啟瀏覽器
器來顯示地圖
# 程式碼置於 Map 子目錄
```



SECTION III-B

PLOTLY FOR R

- Quick Start
 - Getting Started
 - Cheat Sheet
 - Full Reference
 - User Guide
 - Use Offline
 - ggplot2
 - Shiny Gallery
 - Shiny for Python
- Examples
 - Plotly Fundamentals
 - Basic
 - Statistical
 - Scientific
 - Financial
 - Maps
 - 3D



Plotly R Open Source Graphing Library

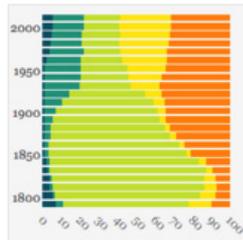
Plotly's R graphing library makes interactive, publication-quality graphs online. Examples of how to make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, and 3D (WebGL based) charts.

Search

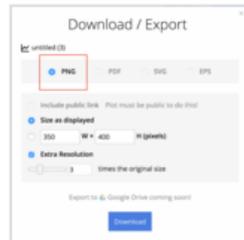
Plotly Fundamentals [↗](#)



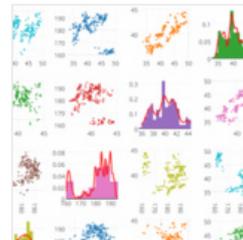
Dashboards



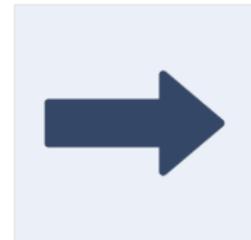
Updating Plotly Graphs



Static Image Export



Embedding Graphs with Knitr



More Plotly Fundamentals



GETTING STARTED

1. Install

In the console:
`install.packages('plotly')`

2. Sign Up & Configure

`plot.ly/r/getting-started`

3. A Hello World Figure

```
library(plotly)
p <- plot_ly(
  x = rnorm(1000),
  y = rnorm(1000),
  mode = 'markers')
```

4. Plot the Figure!

In the console, either:

Plot Offline by printing the figure:
`p` OR `print(p)`

Plot and Save in Cloud:
`plotly_POST(p)`

BASIC CHARTS

Line Plots

```
plot_ly(
  x = c(1, 2, 3),
  y = c(5, 6, 7),
  type = 'scatter',
  mode = 'lines')
```

Bubble Charts

```
plot_ly(
  x = c(1, 2, 3),
  y = c(5, 6, 7),
  type = 'scatter',
  mode = 'markers',
  size = c(1, 5, 10),
  marker = list(
    color = c('red', 'blue',
              'green')))
```

Scatter Plots

```
plot_ly(
  x = c(1, 2, 3),
  y = c(5, 6, 7),
  type = 'scatter',
  mode = 'markers')
```

Heatmaps

```
plot_ly(
  z = volcano,
  type = 'heatmap')
```

Bar Charts

```
plot_ly(
  x = c(1, 2, 3),
  y = c(5, 6, 7),
  type = 'bar',
  mode = 'markers')
```

Area Plots

```
plot_ly(
  x = c(1, 2, 3),
  y = c(5, 6, 7),
  type = 'scatter',
  mode = 'lines',
  fill = 'tozeroy')
```

LAYOUT

Legends

```
set.seed(123)
x = 1:100
y1 = 2*x + rnorm(100)
y2 = -2*x + rnorm(100)
```

```
plot_ly(
  x = x,
  y = y1,
  type = 'scatter') %>%
```

```
add_trace(
  x = x,
  y = y2) %>%
```

```
layout(
  legend =
    list(x = 0.5,
         y = 1,
         bgcolor = '#F3F3F3'))
```

Axes

```
set.seed(123)
x = 1:100
y1 = 2*x + rnorm(100)
y2 = -2*x + rnorm(100)
```

```
axis_template <- list(
  showgrid = F,
  zeroline = F,
  nticks = 20,
  showline = T,
  title = 'AXIS',
  mirror = 'all')
```

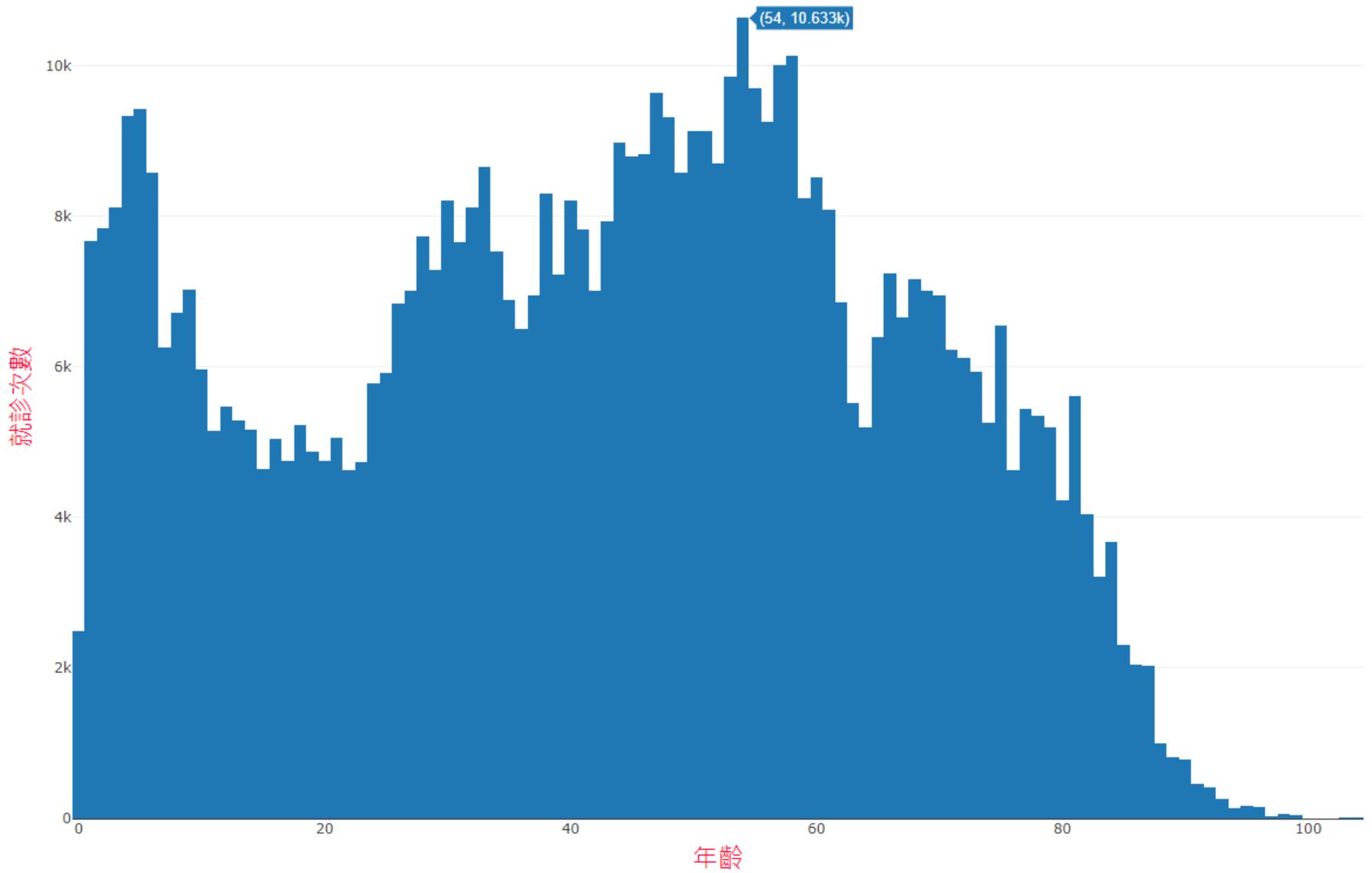
```
plot_ly(
  x = x,
  y = y1,
  type = 'scatter') %>%
```

```
layout(
  xaxis = axis_template,
  yaxis = axis_template)
```

Histogram 直方圖

- 利用 `Plotly_r_Histogram.r`
- 假設已先利用 `Import_tidyverse.r` 匯入資料，如果資料檔放置於其他位置，請更改程式碼內的檔案路徑
- 請在 RStudio 裡執行程式，也可在 RGui 裡執行
- 請參考 <https://plotly.com/r/histograms/>

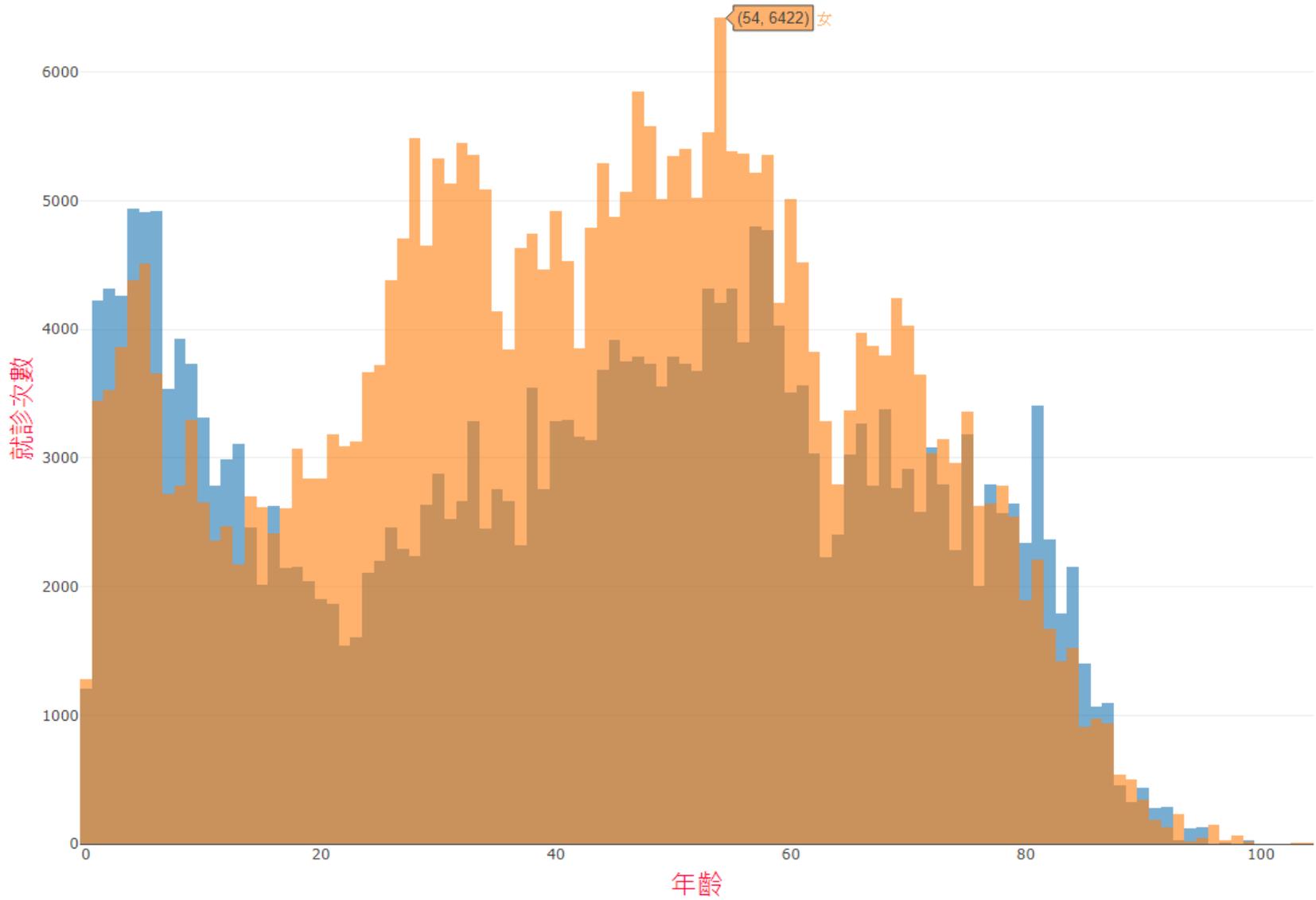
程式碼置於 `Plotly_R` 子目錄



Overlaid Histogram

- 利用 `Plotly_r_Histogram_Overlay.r`
- 假設已先利用 `Import_tidyverse.r` 匯入資料，如果資料檔放置於其他位置，請更改程式碼內的檔案路徑
- 請在 RStudio 裡執行程式，也可在 RGui 裡執行
- 請參考 <https://plotly.com/r/histograms/>

程式碼置於 `Plotly_R` 子目錄



Choropleth Map 分級著色圖

- 利用 ChoroplethMap_plotly.r
- 資料來源 Int J Health Plann Manage 2019;34:e291 (doi: 10.1002/hpm.2647) [PMID: 30204262] 的 Figure 1
- 參考 <https://plotly.com/r/choropleth-maps/> 的 World Choropleth Map
- choro : area; pleth : value

檔案置於 IMG 子目錄

<https://plotly.com/r/choropleth-maps/>

```
library(plotly)
```

```
df <- read.csv('https://raw.githubusercontent.com/plotly/datasets/master/2014_world_gdp_with_codes.csv')
```

```
# 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')  
)
```

```
fig <- plot_geo(df)
```

```
fig <- fig %>% add_trace(  
  z = ~GDP..BILLIONS., color = ~GDP..BILLIONS., colors = 'Blues',  
  text = ~COUNTRY, locations = ~CODE, marker = list(line = 1)  
)
```

```
fig <- fig %>% colorbar(title = 'GDP Billions US$', tickprefix = '$')
```

```
fig <- fig %>% layout(  
  title = '2014 Global GDP<br>Source:<a href="https://www.cia.gov/library/publications/the-world-factbook/fields/2195.htm">CIA World Factbook</a>',  
  geo = g  
)
```

```
fig
```

```
# 更改網站上的程式碼  
# 紅字為更改處  
# ChoroplethMap_plotly.r
```

```
library(plotly)
```

```
Country <-
```

```
c("Philippines", "Poland", "Myanmar", "USA", "Argentina", "Japan", "Czech", "Australia",  
Germany", "Dominican Republic", "South Africa", "Hungary", "Brazil", "UK", "Costa  
Rica", "Paraguay", "New  
Zealand", "Grenada", "Austria", "Canada", "India", "Indonesia", "Ireland", "France", "Kore  
a", "Italy", "Spain", "Belgium", "Chile", "Guatemala", "Netherlands", "Slovakia", "Sri  
Lanka", "Thailand")
```

```
CountryCode <-
```

```
c("PHL", "POL", "MMR", "USA", "ARG", "JPN", "CZE", "AUS", "DEU", "DOM", "ZAF", "HU  
N", "BRA", "GBR", "CRI", "PRY", "NZL", "GRD", "AUT", "CAN", "IND", "IDN", "IRL", "FRA",  
KOR", "ITA", "ESP", "BEL", "CHL", "GTM", "NLD", "SVK", "LKA", "THA")
```

```
IMG <-
```

```
c(550,420,364,73,56,47,30,26,23,18,16,13,12,12,8,8,7,6,5,5,5,5,5,4,4,2,2,1,1,1,1,1,  
1,1)
```

```
df <- data.frame(Country, CountryCode, IMG)
```

```
l <- list(color = toRGB("black"), width = 1.5)
```

```
# specify map projection/options
```

```
g <- list(  
  showframe = FALSE,  
  showcoastlines = TRUE,  
  projection = list(type = 'Mercator')  
)
```

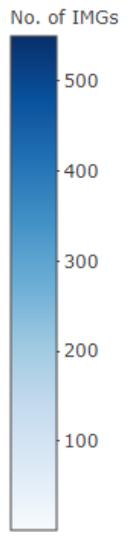
```
fig <- plot_geo(df)
```

```
fig <- fig %>% add_trace(  
  z = ~IMG, color = ~IMG, colors = 'Blues',  
  text = ~Country, locations = ~CountryCode, marker = list(line = l)  
)
```

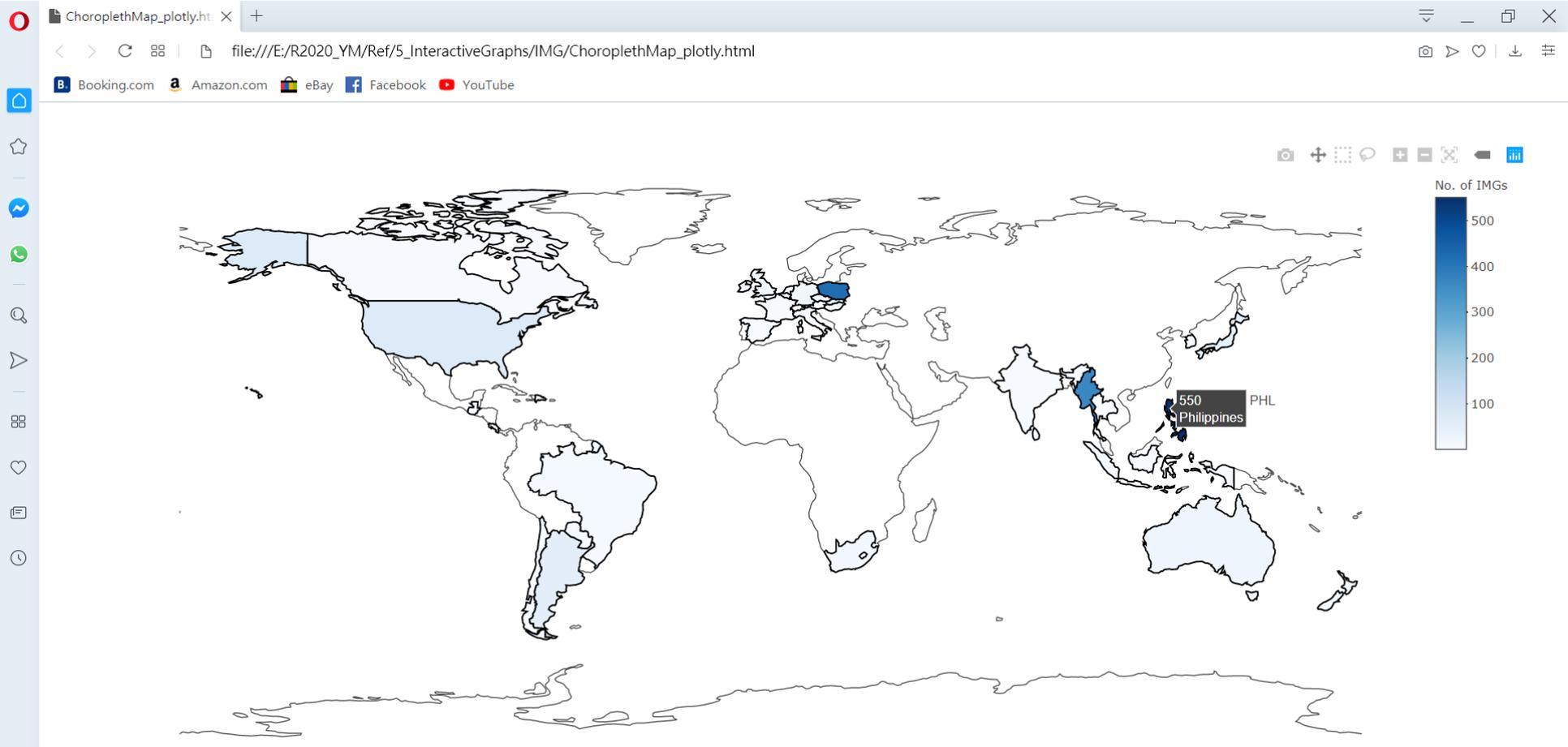
```
fig <- fig %>% colorbar(title = 'No. of IMGs')
```

```
fig <- fig %>% layout(  
  geo = g  
)
```

```
fig
```



另存為網頁顯示



SECTION III-C

DYGRAPHS FOR R

dygraphs for R

Home

USING

[R Console](#)

[R Markdown](#)

[Shiny](#)

GALLERY

[Series Options](#)

[Series Highlighting](#)

[Axis Options](#)

[Labels & Legends](#)

[Time Zones](#)

[CSS Styling](#)

[Range Selector](#)

[Candlestick Charts](#)

dygraphs for R

The dygraphs package is an R interface to the [dygraphs](#) JavaScript charting library. It provides rich facilities for charting time-series data in R, including:

- Automatically plots [xts](#) time series objects (or any object convertible to xts).
- Highly configurable axis and series display (including optional second Y-axis).
- Rich interactive features including [zoom/pan](#) and series/point [highlighting](#).
- Display [upper/lower bars](#) (e.g. prediction intervals) around series.
- Various graph overlays including [shaded regions](#), [event lines](#), and point [annotations](#).
- Use at the R console just like conventional R plots (via RStudio Viewer).
- Seamless embedding within [R Markdown](#) documents and [Shiny](#) web applications.

Installation

You can install the **dygraphs** package from CRAN as follows:

```
install.packages("dygraphs")
```

You can use dygraphs at the R console, within R Markdown documents, and within Shiny applications. See the usage documentation linked to from the sidebar for more details. There are a

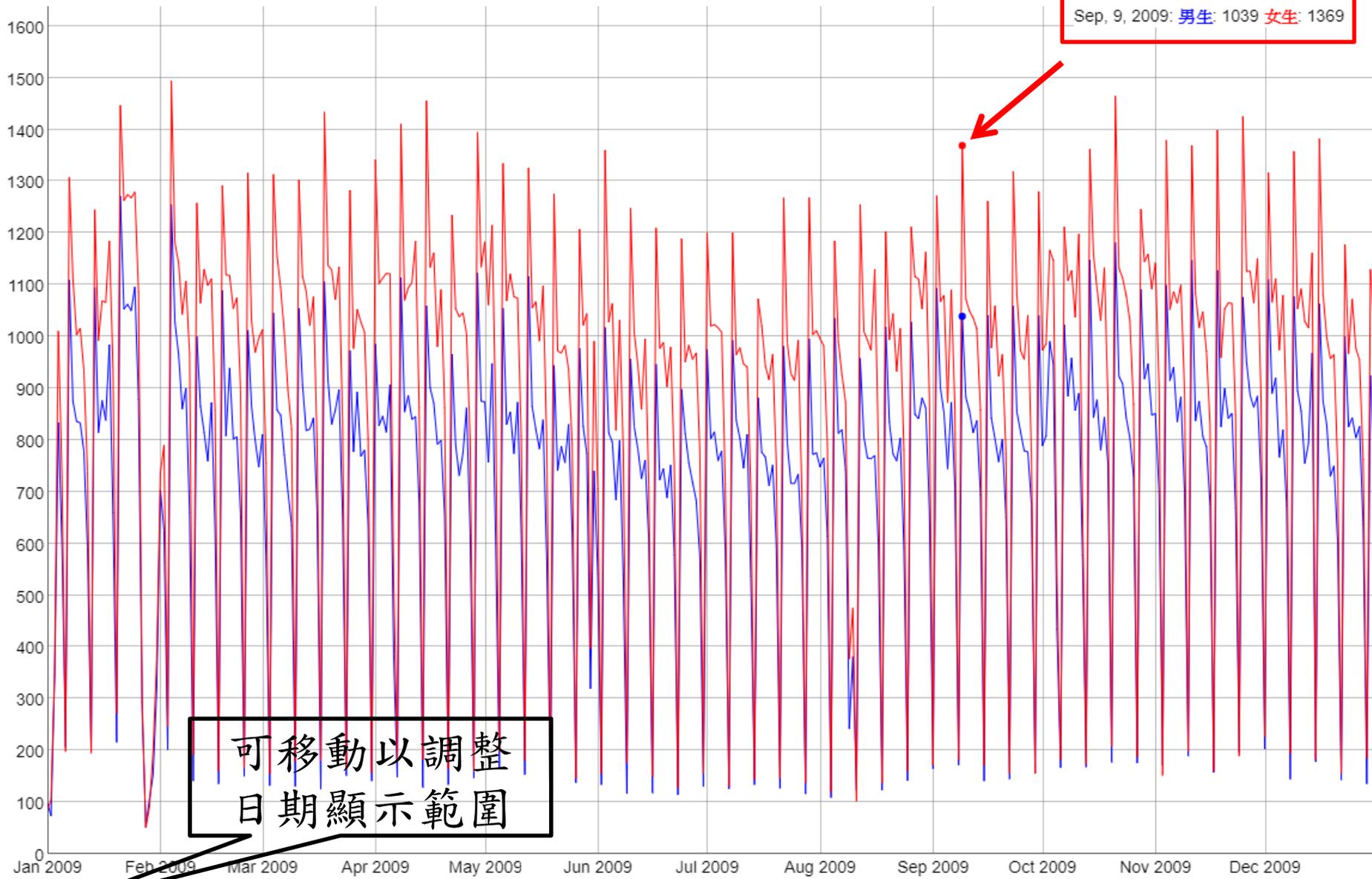
Time Series Chart 時間序列圖

- 利用 `Dygpraphs_TimeSeries.r`
- 假設已先利用 `Import_tidyverse.r` 匯入資料，如果資料檔放置於其他位置，請更改程式碼內的檔案路徑
- 請在 RStudio 裡執行程式，也可在 RGui 裡執行

程式碼置於 Dygpraphs 子目錄

每日就診次數

Sep, 9, 2009: 男生: 1039 女生: 1369



可移動以調整
日期顯示範圍

Jan 2009 Feb 2009 Mar 2009 Apr 2009 May 2009 Jun 2009 Jul 2009 Aug 2009 Sep 2009 Oct 2009 Nov 2009 Dec 2009

SECTION IV

R FOR AI

SECTION IV-A

OCR

Tesseract



- an optical character recognition (OCR) engine for various operating systems
- free software
- 1985-1994 : developed at Hewlett Packard labs
- 2005 : released as open source
- 2006 : sponsored by Google

Demo with tesseract



```
if (! require(tesseract)) install.packages("tesseract")
```

```
library(tesseract)
```

```
write( ocr("D:/1.jpg", engine = tesseract("eng")), file = "D:/1.txt" )
```

```
tesseract_download("chi_tra")
```

```
write( ocr("D:/2.png", engine = tesseract("chi_tra")), file = "D:/2.txt" )
```

請自行更改圖檔與文檔的存放位置

檔案置於 OCR 子目錄

Deep learning (also known as **deep structured learning** or **hierarchical learning**) is part of a broader family of **machine learning** methods based on **learning data representations**, as opposed to task-specific algorithms. Learning can be **supervised**, **semi-supervised** or **unsupervised**.^{[1][2][3]}



Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised [**12115**]

* 不同作業系統，不同套件版本，辨識結果略有差異

- 第一境界：（出於外而能觀）昨夜西風凋碧樹，獨上高樓，望盡天涯路。（引自晏殊的《蝶戀花》）
- 第二境界：（入於內而能寫）衣帶漸寬終不悔，為伊消得人憔悴。（引自柳永的《蝶戀花》）
- 第三境界：（內外圓融而悟）眾裡尋他千百度，驀然回首，那人卻在燈火闌珊處。（引自辛棄疾的《青玉案·元夕》）



- 第一境界：（出於外而能觀）昨夜西風凋百樹，獨上高樓，匡盡天涯路。（引自晏殊的《蝶匕花》）
- 第二境界：（入於內而熊寫）衣帶漸竟終不悔，為伊消得人憔悴。（引自柳永的《蝶匕花》）
- 第三境界：（內外圓融而悟）眾裡尋他千百度，醫然回首，那人卻在燈火閑珊處。（引自辛棄疾的《青玉案'元夕》）

* 不同作業系統，不同套件版本，辨識結果略有差異

SECTION IV-B

HUMAN FACE DETECTION

OpenCV (Open Source Computer Vision)



- a library of programming functions
- 1999年 英特爾公司啟動
 - 為推進機器視覺的研究，提供一套開源且最佳化的基礎庫
 - 提供一個共同的基礎庫，使得開發人員的代碼更容易閱讀和轉讓，促進了知識的傳播
 - 通過提供一個不需要開源或免費的軟體許可，促進商業應用軟體的開發
- 2000年 在IEEE Conference on Computer Vision and Pattern Recognition公開
- 2006年 釋出1.0版本
- 2012年 OpenCV的支援由一個非盈利性組織(OpenCV.org)來提供

<https://www.opencv.org/>

<https://cran.r-project.org/web/packages/opencv/index.html>

<https://zh.wikipedia.org/wiki/OpenCV>

Demo with opencv



```
if (!require(opencv)) install.packages("opencv")  
library(opencv)
```

```
ocv_write( ocv_face( ocv_read("D:/ym.jpg") ), "D:/ym_ocv.jpg")
```

請自行更改圖檔的存放位置

僅展示第一步驟偵測臉部位位置。若欲進一步比對臉
部，需有可供比對的人臉資料。

檔案置於 HumanFaceDetection 子目錄

```
https://github.com/ropensci/opencv
```

```
https://cran.r-project.org/web/packages/opencv/index.html
```



SECTION IV-C

DEEP LEARNING

Deep Learning

- R interface to **TensorFlow** : an open-source software library for machine intelligence
(<https://tensorflow.rstudio.com/>)
(<https://www.tensorflow.org/>)
- R interface to **Keras** : a high-level neural networks API (<https://keras.rstudio.com/>) (<https://keras.io/>)
- R interface to **H2O** : an open-source artificial intelligence platform
(<https://www.h2o.ai/>)
- R interface to Python via **reticulate**
(<https://github.com/rstudio/reticulate>)



SECTION V

VALUABLE RESOURCES

The Data Visualisation Catalogue

[About](#) • [Blog](#) • [Shop](#) • [Resources](#)

CN 中文

ES Español

RU Русский

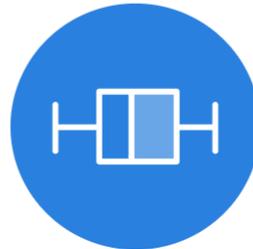
TR Türkçe

Search by Function

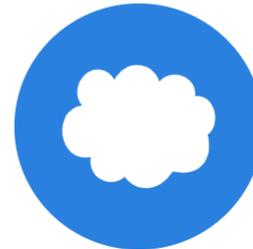
View by List



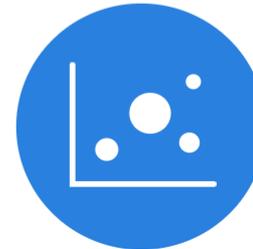
Bar Chart



Box & Whisker Plot



Brainstorm



Bubble Chart

The logo for RAWGraphs, featuring the word "RAW" in a bold, dark grey sans-serif font and "Graphs" in a teal, lowercase sans-serif font. To the left of the text is a decorative graphic of overlapping circles in teal, yellow, and red.

[About](#)

[Blog](#)

[Learning](#)

[Gallery](#)

[Documentation](#)

[User survey](#)

RAWGraphs

The missing link between spreadsheets and data visualization.

[USE IT NOW!](#)

[FORK IT ON GITHUB](#)



Interactive charts



An interactive charts allows the user to perform actions: zooming, hovering a marker to get a tooltip, choosing a variable to display and more. `R` offers a set of packages called the `htmlwidgets`: they allow to build interactive dataviz directly from `R`.

Thanks for
Your Attention
!

