

BÀI THỰC HÀNH 3: THU THẬP DỮ LIỆU TỪ CÁC NGUỒN KHÁC NHAU

1. Các nguồn thu thập dữ liệu.

Một số nguồn thu thập dữ liệu phổ biến:

- Dữ liệu trong các hệ quản trị CSDL: MySQL (dữ liệu dạng quan hệ), MongoDB (dữ liệu phi quan hệ - dạng tài liệu),
 - Dữ liệu từ các trang web: Sử dụng chức năng Web scraping hoặc web crawler để lấy dữ liệu về từ các trang web.
 - Dữ liệu từ các API: Một số hệ thống cho phép lấy dữ liệu về thông qua các API cung cấp sẵn, ví dụ như: Facebook, Twitter hoặc Zalo có cung cấp API để có thể truy xuất dữ liệu.
 - Dữ liệu từ các trang web chia sẻ dữ liệu: Các trang web này cung cấp sẵn các dataset, điển hình là: Kaggle và UCI.
 - Các nguồn dữ liệu khác: Google sheet, Microsoft Excel, XML files, H5 files.
- Tuỳ thuộc vào từng loại nguồn dữ liệu mà ta có cách thu thập dữ liệu khác nhau.

2. Thu thập dữ liệu từ Web

Dữ liệu từ trang web đa phần được biểu diễn ở định dạng HTML - là một dạng ngôn ngữ đánh dấu (Markup Language). Để thu thập được dữ liệu, chúng ta cần "bắt" được cấu trúc của HTML trong trang web, từ đó mới trích xuất ra được dữ liệu cần dùng.

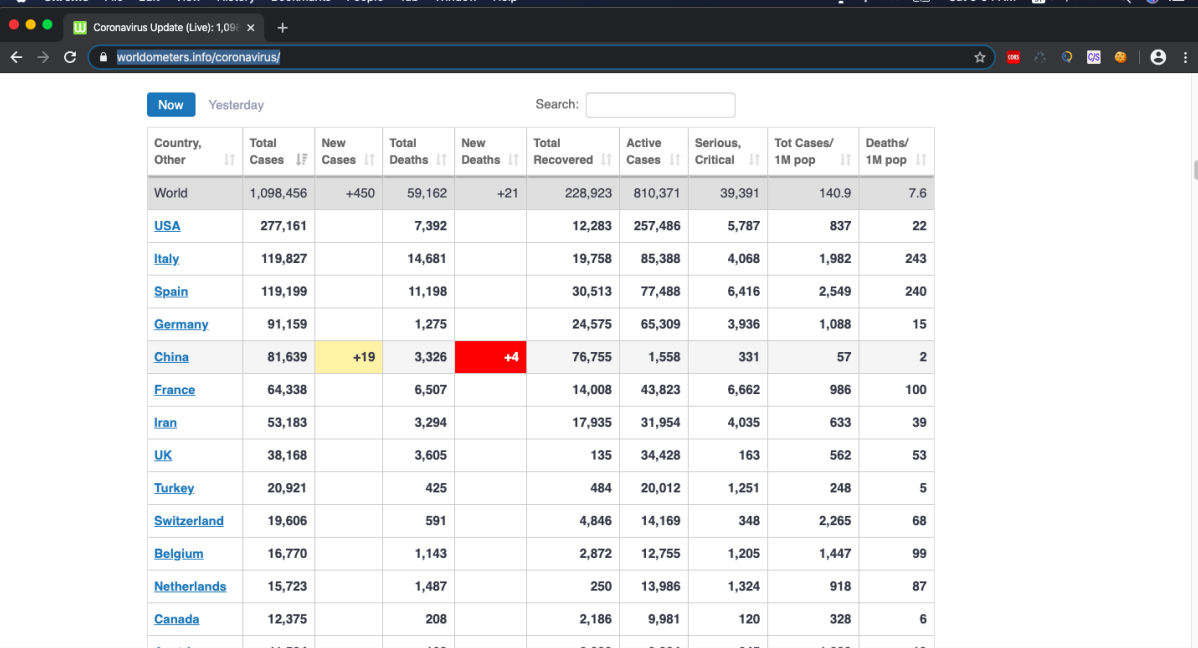
R cung cấp thư viện xml2 dùng để phân tích và bắt được cấu trúc của HTML và XML.

Thông tin về thư viện xml2: <https://cran.r-project.org/web/packages/xml2/xml2.pdf>

Trong ví dụ này, chúng ta sẽ thu thập dữ liệu về bảng tổng kết các quốc gia có số ca nhiễm COVID-19 trên toàn thế giới từ trang web Wordometer.

Link trang web chứa dữ liệu: <https://www.worldometers.info/coronavirus/#countries>

Thực hành Thu thập và tiền xử lý dữ liệu

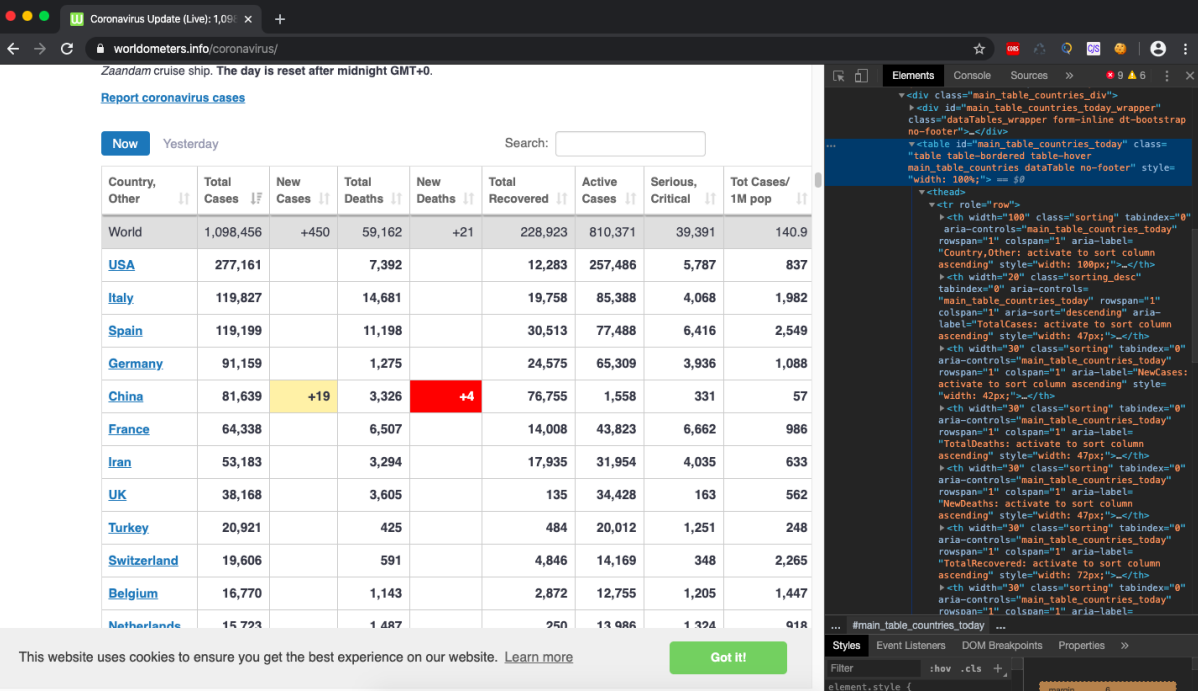


| Country, Other | Total Cases | New Cases | Total Deaths | New Deaths | Total Recovered | Active Cases | Serious, Critical | Tot Cases/1M pop | Deaths/1M pop |
|-----------------------------|-------------|-----------|--------------|------------|-----------------|--------------|-------------------|------------------|---------------|
| World | 1,098,456 | +450 | 59,162 | +21 | 228,923 | 810,371 | 39,391 | 140.9 | 7.6 |
| USA | 277,161 | | 7,392 | | 12,283 | 257,486 | 5,787 | 837 | 22 |
| Italy | 119,827 | | 14,681 | | 19,758 | 85,388 | 4,068 | 1,982 | 243 |
| Spain | 119,199 | | 11,198 | | 30,513 | 77,488 | 6,416 | 2,549 | 240 |
| Germany | 91,159 | | 1,275 | | 24,575 | 65,309 | 3,936 | 1,088 | 15 |
| China | 81,639 | +19 | 3,326 | +4 | 76,755 | 1,558 | 331 | 57 | 2 |
| France | 64,338 | | 6,507 | | 14,008 | 43,823 | 6,662 | 986 | 100 |
| Iran | 53,183 | | 3,294 | | 17,935 | 31,954 | 4,035 | 633 | 39 |
| UK | 38,168 | | 3,605 | | 135 | 34,428 | 163 | 562 | 53 |
| Turkey | 20,921 | | 425 | | 484 | 20,012 | 1,251 | 248 | 5 |
| Switzerland | 19,606 | | 591 | | 4,846 | 14,169 | 348 | 2,265 | 68 |
| Belgium | 16,770 | | 1,143 | | 2,872 | 12,755 | 1,205 | 1,447 | 99 |
| Netherlands | 15,723 | | 1,487 | | 250 | 13,986 | 1,324 | 918 | 87 |
| Canada | 12,375 | | 208 | | 2,186 | 9,981 | 120 | 328 | 6 |

Hình 1: Trang web Worldometer về Corona virus.

Mục tiêu của chúng ta là thu thập được bảng số liệu về COVID-19 ở từng quốc gia từ trang web Worldometer như hình trên.

Đối với các trình duyệt hiện tại như Firefox hay Chrome đều cung cấp sẵn công cụ Inspect - cho phép xem nội dung của trang web dưới dạng mã HTML. Công cụ này cực kỳ hữu ích khi chúng ta cần tìm hiểu trước nội dung trang web.



The screenshot shows the Worldometer website with the browser's developer tools open. The 'Elements' panel on the right shows the HTML structure of the table, highlighting the 'tbody' element. The 'Console' panel shows the JavaScript code for the table's functionality. The table data is the same as in the previous image.

Thực hành Thu thập và tiền xử lý dữ liệu

Hình 2: Nội dung trang web bằng HTML khi sử dụng công cụ Inspect trên Chrome

Sau khi tìm hiểu sơ qua code HTML của trang web, chúng ta "bắt" được nội dung cần thu thập của chúng ta, là bảng chứa dữ liệu về số ca nhiễm.



Hình 3: Vị trí trong code HTML của trang web chứa dữ liệu cần thu thập.

Trong hình 3, bảng số liệu của chúng ta cần thu thập, là một thẻ HTML `<table>`, với `id = main_table_countries_today`. Nếu như bắt được nội dung của thẻ này, chúng ta sẽ thu được dữ liệu của toàn bảng.

```
1245 <table id="main_table_countries_today" class="table table-bordered table-hover main_table_countries" style="width:100%">
1246 <thead>
1247 <tr>
1248 <th width="100">Country,<br />Other</th>
1249 <th width="20">Total<br />Cases</th>
1250 <th width="30">New<br />Cases</th>
1251 <th width="30">Total<br />Deaths</th>
1252 <th width="30">New<br />Deaths</th>
1253 <th width="30">Total<br />Recovered</th>
1254 <th width="30">Active<br />Cases</th>
1255 <th width="30">Serious,<br />Critical</th>
1256 <th width="30">Totenbsp;Cases<br />1M pop</th>
1257 <th width="30">Deaths<br />1M pop</th>
1258 </tr>
1259 </thead>
1260 <tbody>
1261 <tr style="">
1262 <td style="font-weight: bold; font-size:15px; text-align:left;"><a class="mt_a" href="country/us/">USA</a></td>
1263 <td style="font-weight: bold; text-align:right;">277,161</td>
1264 <td style="font-weight: bold; text-align:right;"></td>
1265 <td style="font-weight: bold; text-align:right;">7,392 </td>
1266 <td style="font-weight: bold; text-align:right;"></td>
1267 <td style="font-weight: bold; text-align:right;">12,283</td>
1268 <td style="text-align:right;font-weight:bold;">257,486</td>
1269 <td style="font-weight: bold; text-align:right;">5,787</td>
1270 <td style="font-weight: bold; text-align:right;">837</td>
1271 <td style="font-weight: bold; text-align:right;">22</td>
1272 </tr>
1273 <tr style="">
1274 <td style="font-weight: bold; font-size:15px; text-align:left;"><a class="mt_a" href="country/italy/">Italy</a></td>
1275 <td style="font-weight: bold; text-align:right;">119,827</td>
1276 <td style="font-weight: bold; text-align:right;"></td>
1277 <td style="font-weight: bold; text-align:right;">14,481 </td>
1278 <td style="font-weight: bold; text-align:right;"></td>
1279 <td style="font-weight: bold; text-align:right;">19,758</td>
1280 <td style="text-align:right;font-weight:bold;">85,388</td>
1281 <td style="font-weight: bold; text-align:right;">4,068</td>
1282 <td style="font-weight: bold; text-align:right;">1,982</td>
1283 <td style="font-weight: bold; text-align:right;">243</td>
1284 </tr>
1285 <tr style="">
1286 <td style="font-weight: bold; font-size:15px; text-align:left;"><a class="mt_a" href="country/spain/">Spain</a></td>
1287 <td style="font-weight: bold; text-align:right;">119,199</td>
1288 <td style="font-weight: bold; text-align:right;"></td>
1289 <td style="font-weight: bold; text-align:right;">11,198 </td>
1290 <td style="font-weight: bold; text-align:right;"></td>
1291 <td style="font-weight: bold; text-align:right;">30,513</td>
1292 <td style="text-align:right;font-weight:bold;">77,488</td>
1293 <td style="font-weight: bold; text-align:right;">6,416</td>
1294 <td style="font-weight: bold; text-align:right;">2,549</td>
```

Hình 4: Cấu trúc HTML của bảng chứa dữ liệu.

Ứng với thẻ `<table>`, chúng ta thấy rằng:

- **<thead>** là thẻ đánh dấu được dòng tiêu đề của bảng dữ liệu. Thẻ `<tr>` trong `<thead>` định nghĩa 1 dòng - ứng với dòng tiêu đề. Trong thẻ `<tr>` có các thẻ con `<th>`, mỗi thẻ con định nghĩa 1 ô tiêu đề - tương ứng với các thuộc tính trong dataset. VD: Country, Total Cases, New Cases,

Thực hành Thu thập và tiền xử lý dữ liệu

- **<tbody>** là thẻ đánh dấu phần nội dung của bảng. Trong **<tbody>** có rất nhiều thẻ con, mỗi thẻ con **<tr>** trong **<tbody>** ứng với 1 dòng dữ liệu về một quốc gia có số ca nhiễm - tương đương với 1 record hay 1 điểm dữ liệu cần thu thập (datapoint). Trong mỗi **<tr>** lại có các thẻ **<td>** con chứa dữ liệu theo từng thuộc tính của 1 điểm dữ liệu.

Bằng cách "bắt" ra các thẻ chứa dữ liệu ở trên, chúng ta có thể lấy được dữ liệu của bảng số liệu trên.

Thực hiện bằng R:

Đầu tiên, cần khai báo thư viện xml2 để có thể đọc được định dạng HTML của trang web trong R:

```
library(xml2)
```

Tiếp theo, chúng ta cần khai báo thư viện httr để có thể thao tác được với các phương thức GET hoặc POST từ WWW.

```
library(httr)
```

Sử dụng phương thức GET để lấy dữ liệu về:

```
data <- GET("https://www.worldometers.info/coronavirus/#countries")
```

Sau khi lấy được dữ liệu về, chúng ta sử dụng thư viện xml2 để đọc và chuyển cấu trúc HTML về dạng cây (tree).

```
content <- content(data, as="text")
```

```
html_data <- read_xml(content, as_html = TRUE)
```

Lưu ý: as_html = TRUE tức là dữ liệu đang đọc ở dạng HTML. Nếu không set bằng TRUE thì mặc định hàm read_xml() hiểu là đang đọc dữ liệu dạng XML.

Lấy dữ liệu về dòng tiêu đề của bảng:

```
table_head <- xml_text(xml_find_all(html_data,  
  "//table[@id='main_table_countries_today']/thead/tr/th"))
```

Lưu ý: đoạn chuỗi sau: `"//table[@id='main_table_countries_today']/thead/tr/th"` gọi là XPath - là một dạng ngôn ngữ truy vấn dựa trên cây XML, dùng để tìm ra giá trị

Thực hành Thu thập và tiền xử lý dữ liệu

cần tìm dựa trên các nút XML. Như đoạn XPath ở trên sẽ truy vấn ra dữ liệu dạng text từ cấu trúc có dạng sau:

```
<table id='main_table_countries_today'>
  <thead>
    <tr>
      <th> Country, Other </th>
      <th> Total Cases </th>
      <th> New Cases </th>
      ....
    </tr>
  </thead>
</table>
```

Tương tự, ta lấy dữ liệu về từng quốc gia trong bảng, mỗi quốc gia ứng với một dòng.

```
table_data <- xml_text(xml_find_all(html_data,
  "//table[@id='main_table_countries_today']//tbody//tr//td"))
```

Cuối cùng, chúng ta lưu dữ liệu vào Dataframe trong R.

Code R hoàn chỉnh: file worldometer_scraping.R

```
# Lấy dữ liệu từ web - WEB SCRAPING
# =====
rm(list=ls())
library(httr)
library(xml2)
library(dplyr)

data <-
GET("https://www.worldometers.info/coronavirus/#countries"
)

content <- content(data, as="text")
```

```
html_data <- read_xml(content, as_html = TRUE)

# Lay cac thuoc tinh trong tbody - du lieu cua bang
table_data <- xml_text(xml_find_all(html_data,
"//table[@id='main_table_countries_today']//tbody//tr//td"
))

# Lay cac thuoc tinh trong thead - tieu de cua bang
table_head <- xml_text(xml_find_all(html_data,
"//table[@id='main_table_countries_today']//thead//tr//th"
))

# So cot o day tuong ung so luong table_head
# So dong o day tuong ung so luong table_data
dataset <- matrix(ncol=length(table_head), nrow =
(length(table_data)/ length(table_head) ))

data_col = length(table_head) # So luong cot trong bang

# Lan luot lay tung dong trong table_data bo vao dataset
count = 1
i = 1
while(i<=length(table_data) - data_col) {
  dataset[count, ] <- c(table_data[i:(i -1 + data_col)])
  i = i + data_col;
  count = count + 1
}

# Chuyen ma tran dataset thanh dang dataframe
dataset <- as.data.frame(dataset)
dataset <- na.omit(dataset)

# gan tieu de cho dataframe
names(dataset) <- c(table_head)
```

```
# lưu dữ liệu thành csv
write.csv(dataset, "corona_data.csv")
```

2. Thu thập dữ liệu từ API.

API - Application Programming Interface là một tập hợp các phương thức và giao thức dùng để kết nối các thư viện và ứng dụng lại với nhau. Các API tạo nên sự tiện lợi trong việc trao đổi dữ liệu giữa các hệ thống đối với nhau.

Trong lập trình web, các API thường được sử dụng để trao đổi dữ liệu giữa client và server thông qua các giao thức khác nhau: HTTP, WSS, ... Một loại giao thức API thường gặp là: RestAPI

Một API thường gồm các thành phần sau:

1. URL: chứa link truy cập tới hệ thống tài nguyên. Một URL thường sẽ có dạng:

<giao thức>://<tên hoặc địa chỉ IP máy chủ>/<tên tài nguyên 1>/<tên tài nguyên 2>/...

VD: <https://student.uit.edu.vn/tracuu/hocphi>

2. Header: chứa thông tin về các thuộc tính và yêu cầu cần gửi tới máy chủ. Các thuộc tính và yêu cầu này bao gồm:

Phương thức gửi: GET, POST, PUT, DELETE (Theo RestAPI) hoặc

Các thông tin xác thực: các chuỗi token xác thực người dùng, các đoạn mã hash xác thực phân quyền, ...

Các thông tin khác: Định dạng dữ liệu trả về / gửi đi.

3. Body: chứa dữ liệu trả về (đối với server) hoặc dữ liệu gửi lên (đối với client).

Dữ liệu trả về hoặc gửi lên thông qua API có thể ở nhiều định dạng khác nhau: json, xml, html, plain text, binary, ... Tuy nhiên, 2 định dạng phổ biến thường được sử dụng là: JSON (đối với các API theo thiết kế RestAPI) và XML (đối với các API theo thiết kế SOAP).

Để đọc được các dữ liệu trả về từ API, trong R cung cấp thư viện *jsonlite*.

API Open weather map: <https://openweathermap.org/>

Link lấy thông tin về thời tiết hàng giờ tại 1 thành phố theo mã ZIP CODE của thành phố đó:

```
https://samples.openweathermap.org/data/2.5/forecast/hourly?zip=94040&appid=b6907d289e10d714a6e88b30761fae22
```

Ở đây, zip và appid là 2 biến sẽ được gửi lên server từ client. zip là mã zip code của thành phố cần xem thông tin và appid là api key được cung cấp. 94040 là mã zip code của thành phố Los Angeles (Mỹ).

Note: Một số hệ thống bắt buộc người dùng phải đăng ký một tài khoản trên hệ thống, sau khi đăng ký mới có được api_key, đôi khi còn gọi là app_key. Có api_key đó đính kèm vô thì mới có thể gọi api lên server.

Để thực hiện phương thức GET, ta phải sử dụng thư viện httr.

library(httr)

Gọi thực hiện phương thức GET để lấy dữ liệu:

GET(link_api)

Để sau khi lấy được dữ liệu về, ta phải đọc dữ liệu đó ở dạng JSON. Để đọc được dữ liệu dạng json, cần thư viện jsonlite.

library(jsonlite)

Đọc dữ liệu JSON - dùng hàm: fromJSON và toJSON

fromJSON(toJSON(content), flatten = TRUE)

Code R hoàn chỉnh

```
rm(list=ls())

library(httr)
library(jsonlite)
library(anytime)
library(lubridate)

link <-
"https://samples.openweathermap.org/data/2.5/forecast/hourly?zip=70000&appid=b6907d289e10d714a6e88b30761fae22"
```



```
api <- GET(link)
content <- content(api)
json_content <- toJSON(content)

# doc du lieu dang json
data <- fromJSON(json_content, flatten = TRUE)
#data2 <- fromJSON(json_content, flatten = FALSE)
dataset <- as.data.frame(data['list'])

# chuyen nhiet do tu do F sang do C
convert_to_censius <- function(temp) {
  return(temp - 273.15)
}

# chuyen timestamp to date
convert_timestamp_to_date <- function(time) {
  return(anytime(time))
}

# chuyen nhiet do tu do K sang do C
dataset$list.main.temp = lapply(dataset$list.main.temp,
convert_to_censius)

# sap xep theo thoi gian tang dan
dataset$list.dt <- as.numeric(dataset$list.dt)

dataset <- dataset[order(dataset$list.dt), ]
dataset$list.dt <- as_datetime(anytime(dataset$list.dt))

# ve bieu do nhiet do theo tung ngay
plot(dataset$list.dt, y=dataset$list.main.temp, type="o",
main="Temperature", labels = FALSE)
```

```
axis.POSIXct(1, at=anytime(dataset$list.dt),
format="%y-%m-%d %H:%M:%S", origin="1970-01-01 00:00:00",
labels=TRUE)
axis(2, at=dataset$list.main.temp)

# vẽ biểu đồ phân bố nhiệt độ
plot(density(as.numeric(dataset$list.main.temp)))
# v là vertical line, h: horizontal line
abline(v=mean(as.numeric(dataset$list.main.temp)),
col="red")
```

3. Đọc dữ liệu từ Google Sheet (hay Google bảng tính)

Google bảng tính (Google sheet) là phần mềm bảng tính được cung cấp bởi Google và chạy trực tiếp trên nền ứng dụng Web. Bảng tính Google sheet vì vậy sẽ được cập nhật liên tục và dễ dàng bởi nhiều người. Do đó, việc đọc dữ liệu từ Google sheet sẽ tiện lợi hơn so với đọc dữ liệu từ file excel.

Để đọc được dữ liệu từ Google sheet, cần cài đặt thư viện gsheets trong R.

library(gsheet)

Để đọc dữ liệu từ Google sheet và chuyển thành dạng Dataframe trong R, ta dùng hàm gsheets2tbl(<link>), với link là liên kết tới trang google sheet.

Ví dụ: Trang google sheet sau chứa dữ liệu về COVID-19 tại Nhật Bản.

<https://docs.google.com/spreadsheets/d/1XEFg047aSbg3OsEVx9PzmgSxGbCvCidfLiHfsgRS3R0/edit?usp=sharing>

Yêu cầu:

- Đọc dữ liệu từ trang google sheet đó theo bảng Patient Data
- Cho biết số ca nhiễm theo từng ngày. Vẽ biểu đồ.

Thực hành Thu thập và tiền xử lý dữ liệu

COVID19Japan.com Data ☆

Tệp Chính sửa Xem Chèn Định dạng Dữ liệu Công cụ Tiện ích bổ sung Trợ giúp Chính sửa lần cuối được thực hiện 11 phút ...

100% Chỉ xem

| | A | B | C | D | E | F | G | H | I | J | K | L |
|-----|----------------|----------------|------------|-------------|--------|----------------------------|---------------|---------------------|--------------|----------------------------------|-----------------------------------|---|
| 1 | Patient Number | Date Announced | Date Added | Age Bracket | Gender | Residence City, Prefecture | Detected City | Detected Prefecture | Status | Click Here For Sheet Information | Notes | Source(s) |
| 129 | 128 | 2020-02-23 | 2020-02-23 | 40 | M | Matsudo | Matsudo | Chiba | Unspecified | | Works in Tokyo | https://www3.nhk.or.jp |
| 130 | 129 | 2020-02-23 | 2020-02-23 | 70 | F | Aichi | | Aichi | Hospitalized | | Contact with Aichi#4 | |
| 131 | 130 | 2020-02-23 | 2020-02-23 | 70 | M | Aichi | | Aichi | Hospitalized | | Contact with Aichi#4 | |
| 132 | 131 | 2020-02-23 | 2020-02-29 | 70 | M | Kamikawa | Kamikawa | Hokkaido | Unspecified | | | http://www.pref.hokkaido.lg.jp |
| 133 | 132 | 2020-02-24 | 2020-02-24 | 40 | M | Tokyo | | Port Quarar | Hospitalized | | Quarantine officer working in cru | https://www3.nhk.or.jp |
| 134 | 133 | 2020-02-24 | 2020-02-24 | 50 | M | Tokyo | | Port Quarar | Hospitalized | | Quarantine officer working in cru | https://www3.nhk.or.jp |
| 135 | 134 | 2020-02-24 | 2020-02-24 | 70 | F | Sapporo | Sapporo | Hokkaido | Hospitalized | | | https://www3.nhk.or.jp |
| 136 | 135 | 2020-02-24 | 2020-02-24 | 50 | M | Sapporo | Sapporo | Hokkaido | Hospitalized | | | https://www3.nhk.or.jp |
| 137 | 136 | 2020-02-24 | 2020-02-24 | 20 | F | Okhotsk | Okhotsk | Hokkaido | Unspecified | | | https://www3.nhk.or.jp |
| 138 | 137 | 2020-02-24 | 2020-02-24 | 50 | M | Ishikari | Ishikari | Hokkaido | Unspecified | | | https://www3.nhk.or.jp |
| 139 | 138 | 2020-02-24 | 2020-02-24 | 50 | M | Kanagawa | | Kanagawa | Hospitalized | | JR Sagami-hara station staff | https://www3.nhk.or.jp |
| 140 | 139 | 2020-02-26 | 2020-02-26 | 60 | M | Kanagawa | | Kanagawa | Hospitalized | | Works in JR in Tokyo. Critical co | https://www3.nhk.or.jp |
| 141 | 140 | 2020-02-24 | 2020-02-24 | 50 | M | Tokyo | | Tokyo | Hospitalized | | Works in Dentsu, Minatoku | https://www3.nhk.or.jp |
| 142 | 141 | 2020-02-24 | 2020-02-24 | 40 | M | Tokyo | | Tokyo | Hospitalized | | | https://www3.nhk.or.jp |
| 143 | 142 | 2020-02-24 | 2020-02-24 | 30 | M | Tokyo | | Tokyo | Hospitalized | | | https://www3.nhk.or.jp |
| 144 | 143 | 2020-02-25 | 2020-02-24 | 50 | F | Ishikawa | Kanazawa | Ishikawa | Hospitalized | | No symptoms, Works in hospital | https://www.fukuishi.co.jp |
| 145 | 144 | 2020-02-24 | 2020-02-24 | 60 | M | Ishikawa | Kanazawa | Ishikawa | Hospitalized | | Critical condition | https://www.fukuishi.co.jp |
| 146 | 145 | 2020-02-25 | 2020-02-25 | 50 | M | Kumamoto | | Kumamoto | | | Critical condition | https://www3.nhk.or.jp |
| 147 | 146 | 2020-02-25 | 2020-02-25 | 60 | M | Nagano | Nagano | Nagano | Discharged | | | https://headlines.yahoo.co.jp |

☰ Patient Data ▾ Prefecture Data ▾ Sum By Day ▾ Last Updated ▾ Tokyo Patients ▾ Aggregates < > Khâm phá

Hình 5: Trang Google sheet về COVID-19 Japan

Code R đầy đủ

```
rm(list=ls())
library(gsheet)

link_patien <-
"https://docs.google.com/spreadsheets/d/1XEFg047aSbg3OseVx
9PzmgSxGbCvCidfLiHfsgRS3R0/edit?usp=sharing"

dataset <- gsheet2tbl(link_patien)

# số ca nhiễm theo ngày
data_by_date <- table(dataset$`Date Announced`)
barplot(data_by_date)
```

4. Bài tập:

Bài 1: Sử dụng R, thu thập lại dữ liệu về số ca nhiễm Corona ở từng quốc gia tại trang Worldometer: <https://www.worldometers.info/coronavirus/#countries>.

- Tìm 5 quốc gia có số ca nhiễm (Total case) nhiều nhất.
- Quốc gia nào có số ca nhiễm mới cao nhất?
- Tính tỉ lệ tổng số ca bình phục trên tổng số ca nhiễm. Xác định 3 quốc gia có tỉ lệ bình phục cao nhất.

Bài 2: Dùng lại API ở phần 2, viết chương trình bằng R thu thập dữ liệu về thời tiết của TPHCM. Mã zip của TPHCM là: **70000**.

(Gợi ý: Sửa lại biến zip trong link API).

- Vẽ biểu đồ áp suất không khí (pressure) theo từng ngày.
- Vẽ biểu đồ tốc độ gió (wind speed) theo từng ngày.

Bài 3: Sử dụng lại code thu thập dữ liệu từ Phần 3, viết chương trình gồm các chức năng sau:

- Liệt kê số ca nhiễm theo từng thành phố (Detected City).
- Liệt kê số ca nhiễm theo độ tuổi, vẽ biểu đồ (sử dụng hàm plot).
- Liệt kê số ca nhiễm tại Hokkaido theo từng ngày. Vẽ biểu đồ.

Nộp bài:

Các nội dung cần nộp: Nộp source code ứng với 3 bài tập thực hành. Đặt tên lần lượt là: baitap1.R, baitap2.R và baitap3.R.

Nén lại và đặt tên theo cú pháp <MSSV>_<Họ tên>_BT3.rar. Nộp qua course (Giảng viên sẽ tạo submission sau).

Chúc tất cả các bạn học tốt