ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO
TRỰC QUAN HOÁ DỮ LIỆU
LAB 02: TRỰC QUAN HOÁ DỮ LIỆU VỚI TABLEAU

Nhóm 5:
1. 1712117 – Nguyễn Huỳnh Thảo Nhi
2. 1712710 – Lê Quang Quí
3. 1712713 – Lê Bá Quyền
4. 1712775 – Nguyễn Lê Trường Thành

GVHD:..... Th.S Lê Ngọc Thành

Lớp: ..... Trực quan hoá dữ liệu CQ2017/1

TP.HCM, 2019-2020

Mục Lục

I. Giới thiệu đồ án:

- Đây là bài tiếp nối bài lab 01 mà phần trước chúng ta đã thực hiện việc lấy dữ liệu, tìm hiểu một số quan hệ trên đó. Trong lab này, chúng ta vận dụng Tableau để việc trực quan hóa dữ liệu trở nên tốt hơn.
- Cụ thể trong lab này, nhóm được yêu cầu thực hiện các nhiệm vụ sau:
    - Lý thuyết: Tìm hiểu công cụ Tableau:
        - Giới thiệu về Tableau.
        - Các tính năng hỗ trợ của Tableau kèm các ví dụ minh họa. Các ví dụ được thực hiện trên các mẫu dataset có sẵn, không liên quan đến dataset chính của bài.
    - Thực hành: Vận dụng Tableau để trực quan hóa dữ liệu Woldometer.
        - NSV dựa trên bài đã thực hiện ở phần trước để thực hiện tiếp.
        - NSV cũng quan sát dữ liệu từ đơn giản đến phức tạp, từ thuộc tính đơn đến kết hợp thuộc tính, từ quan hệ độc lập đến quan hệ phụ thuộc, ...
        - Chọn lựa nhiều dạng biểu đồ khác nhau, đánh giá sự phù hợp, có thể dùng lại các lý luận ở phần trước. Thể hiện các biểu đồ này trong Tableau.
        - Sử dụng màu sắc để thể hiện dữ liệu, giải thích ý nghĩa các màu và tại sao mình sử dụng màu như vậy. Sau khi bổ sung màu, NSV có rút ra thêm ý nghĩa gì không?
        - Thể hiện trực quan một số dữ liệu biến đối qua từng ngày. Rút ra ý nghĩa.
        - Sử dụng các kỹ thuật được giới thiệu trong bài Manipulate View, Facet, Reduce, Embed để trình diễn trên Tableau với dữ liệu Woldometer. Giải thích việc chọn lựa và ý nghĩa mang lại.
        - Chạy một số thuật toán học máy đơn giản để hiểu thêm về dữ liệu, thuật toán học máy được quyền sử dụng code có sẵn, ghi rõ nguồn gốc.

II. Phân công công việc:

STT	Công việc	Thành viên thực hiện	Ghi chú
1	- Tìm hiểu về công cụ Tableau. - Trực quan biểu đồ Global heatmap.	Nguyễn Huỳnh Thảo Nhi	
2	Trực quan dữ liệu một ngày và nhiều ngày bằng công cụ Tableau.	Lê Quang Quí	
3	Sử dụng các kỹ thuật được giới thiệu trong bài Manipulate View, Facet, Reduce, Embed để trình diễn trên Tableau với dữ liệu Woldometer.	Lê Bá Quyền	
4	- Chạy một số thuật toán Machine Learning đơn giản.	Nguyễn Lê Trường Thành	

III. Môi trường cài đặt:

- Table Desktop 2020.2
- Ngôn ngữ lập trình: Python

IV. Tìm hiểu công cụ Tableau:

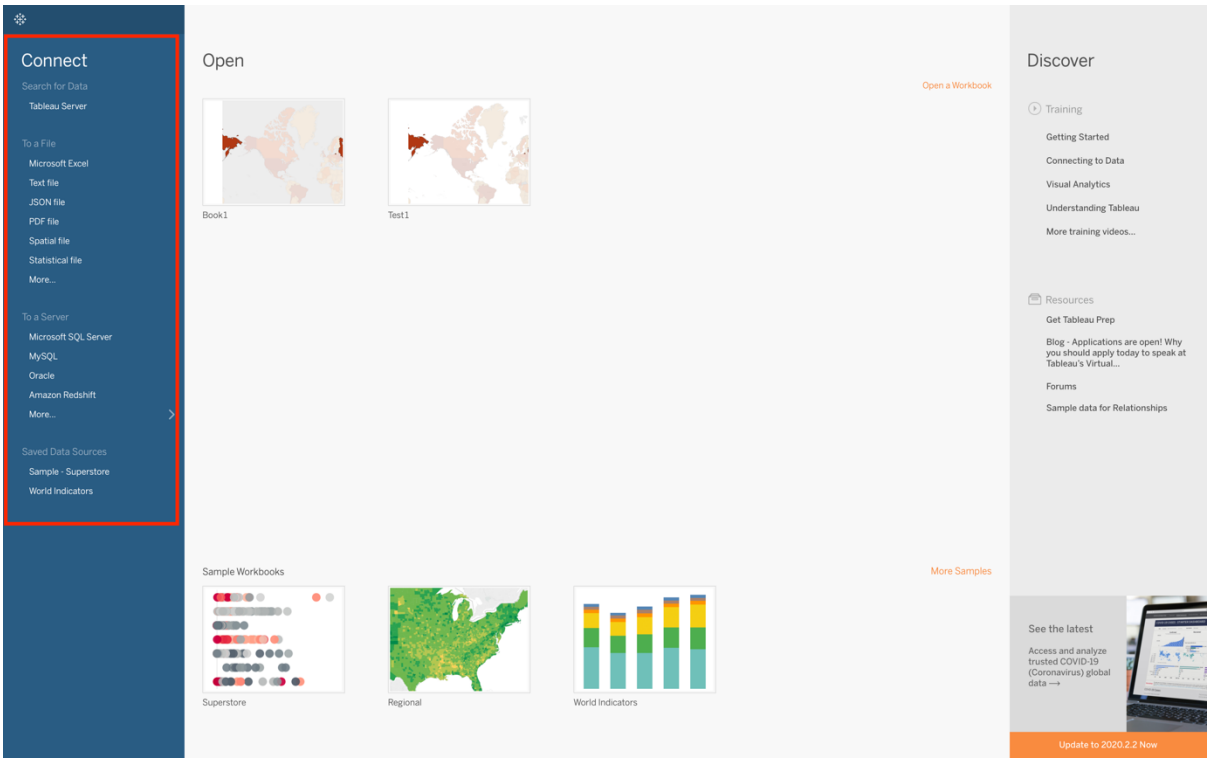
1. Giới thiệu về Tableau:

- **Tableau** là phần mềm hỗ trợ phân tích và trực quan hóa dữ liệu (***Data Visualization),*** phát triển mạnh mẽ và được dùng nhiều trong ngành BI (***Business Intelligence).*** Nó giúp tổng hợp các dữ liệu dạng thô và chuyển đổi những liệu này từ các dãy số thành những hình ảnh, biểu đồ trực quan.
- Phân tích dữ liệu với **Tableau** rất nhanh chóng và hiệu quả, các hình ảnh được tạo ra ở dạng ***Dashboard*** (Bảng điều khiển) và ***WorkSheet*** (Bảng tính). Dữ liệu được tạo bằng **Tableau** có thể được hiểu bởi chuyên gia ở mọi cấp độ trong một tổ chức. Nó thậm chí còn cho phép người dùng không có kỹ thuật tạo bảng điều khiển (***Dashboard***) tùy chỉnh.
- Mục tiêu của **Tableau** rất đơn giản: "Help people see and understand their data".

2. Các tính năng hỗ trợ của Tableau:

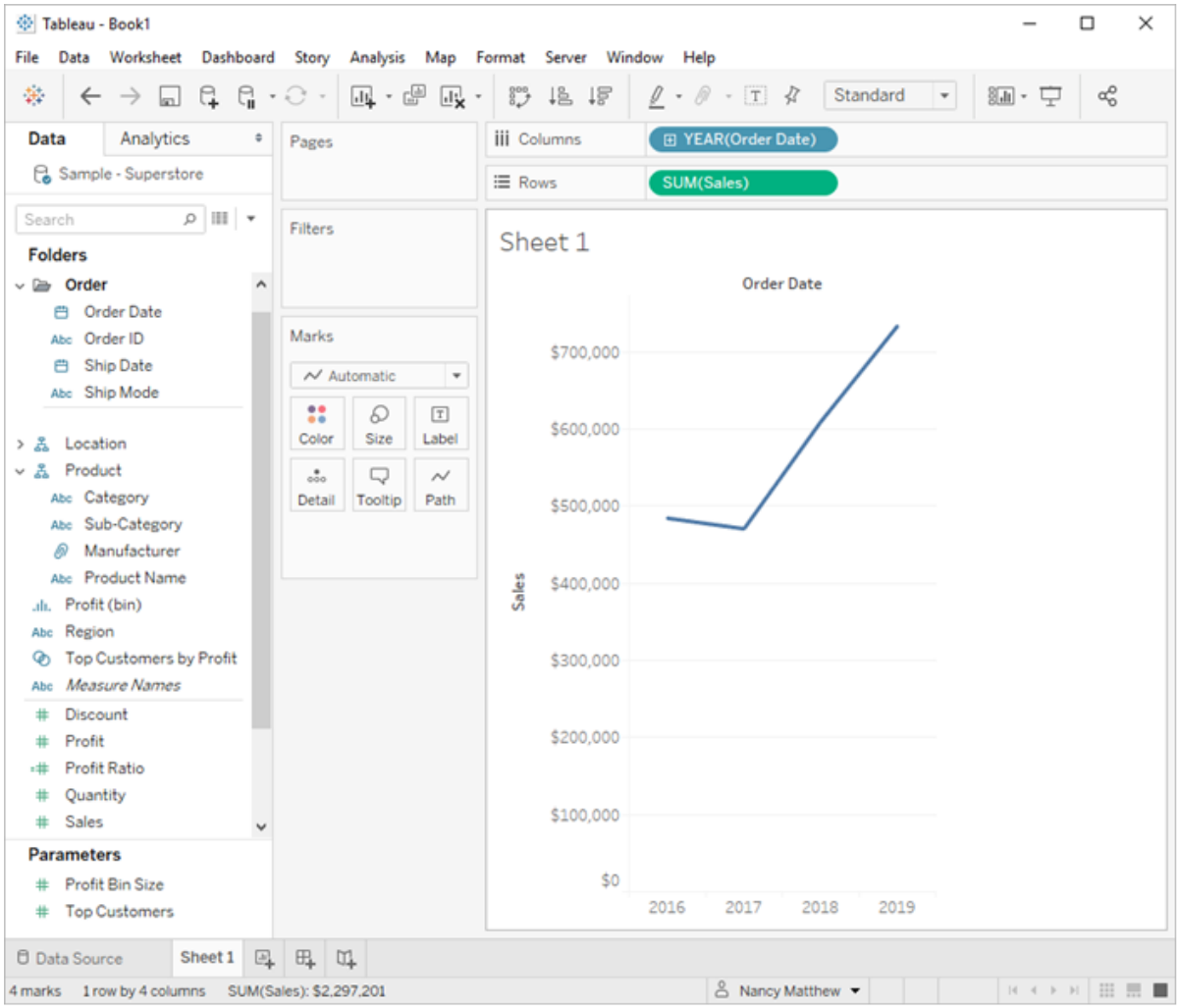
a. Kết nối với dữ liệu:

- Tableau có thể kết nối với các loại dữ liệu khác nhau:
  - Dữ liệu được lưu trữ trong các tệp tin: Microsoft Excel, PDF, Spatial files, v.v….
  - Dữ liệu được lưu trữ trên server: Tableau Server, Microsoft SQL Server, Google Analytics, v.v…
  - Dữ liệu đã được kết nối trước đó.

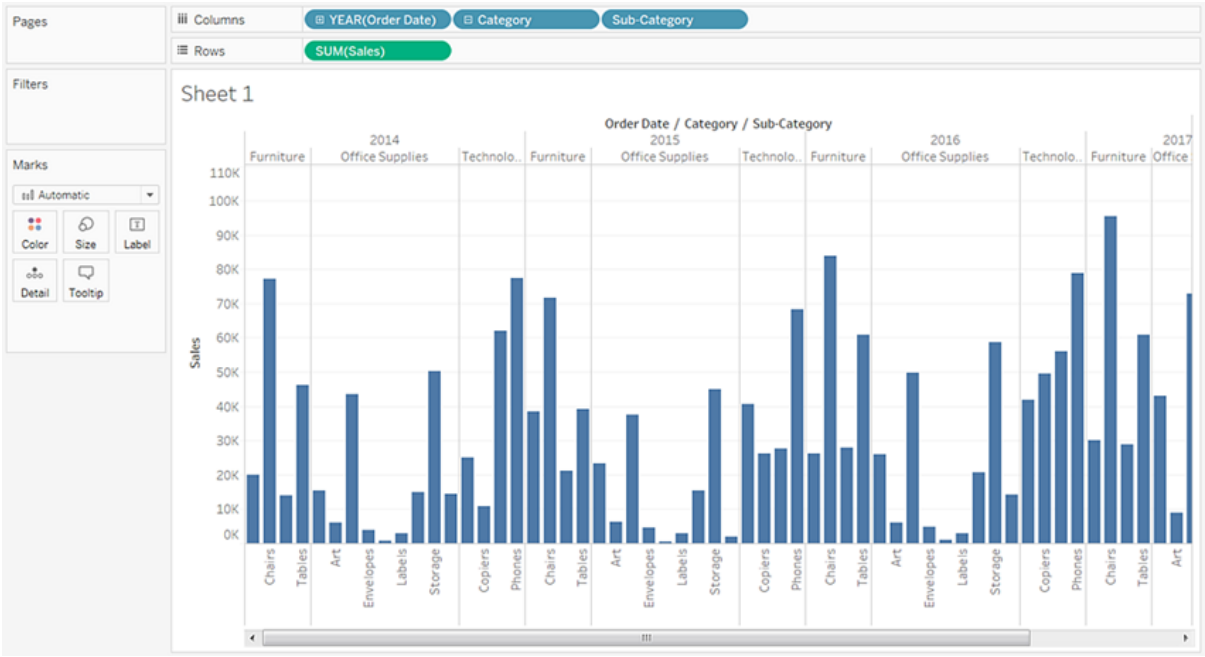


b. Điều khiển kéo thả:

- Tableau hỗ trợ giao diện điều khiển kéo thả giúp người sử dụng dễ dàng thao tác với dữ liệu, kể cả những người dùng không chuyên về kỹ thuật.

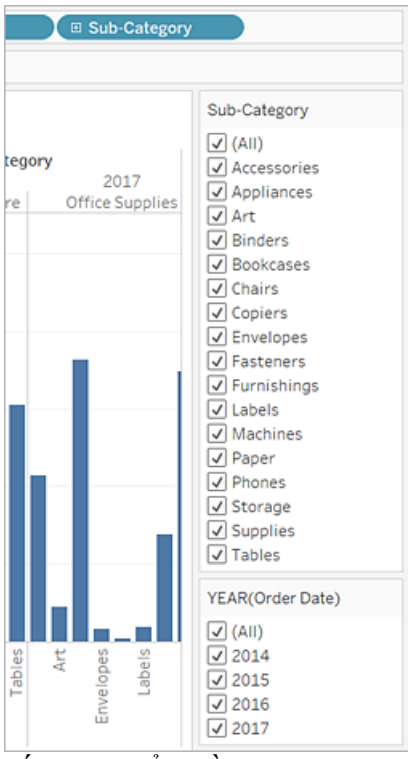


- Như trong hình minh hoạ, các trường thuộc tính của dữ liệu được trình bày ở góc trái màn hình chương trình. Người dùng chỉ cần kéo các thuộc tính vào thanh Columns hoặc Rows để lựa chọn trường dữ liệu thích hợp đối với trục X hoặc Y của biểu đồ.

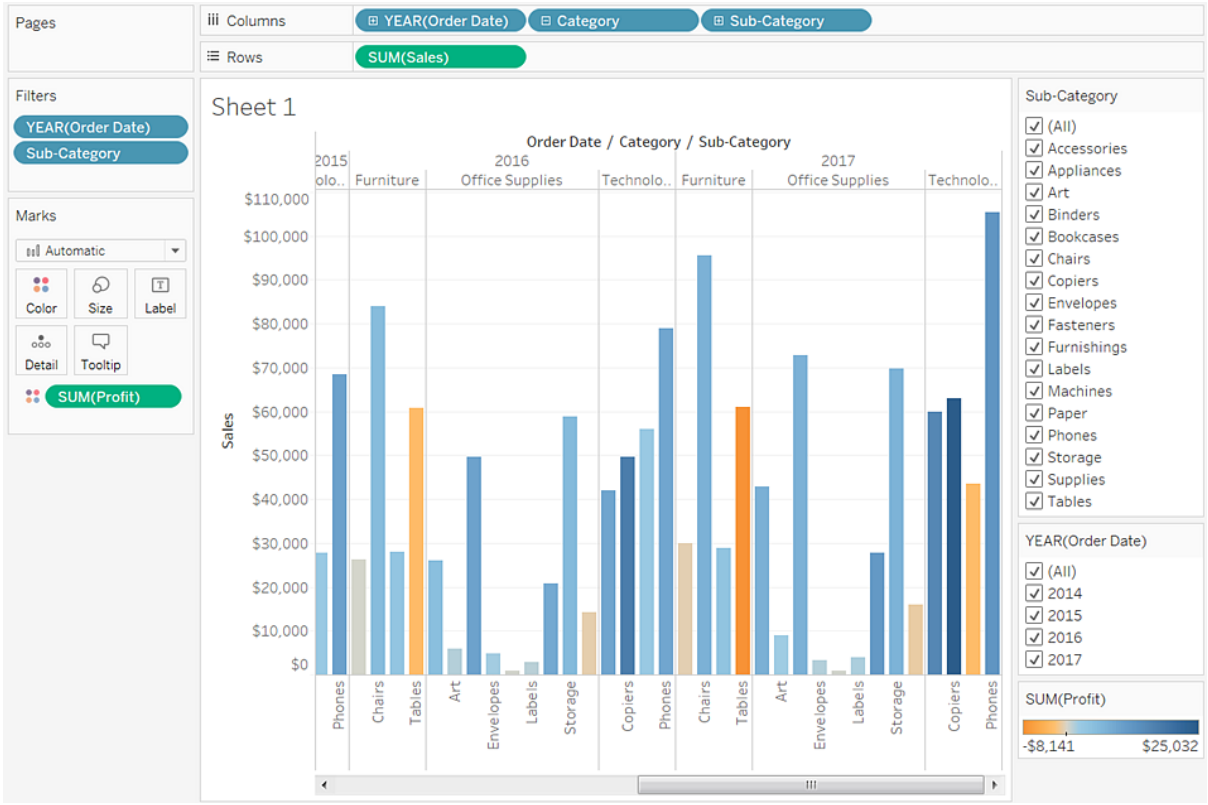


c. Tập trung vào kết quả:

- Áp dụng các bộ lọc (filters) vào dữ liệu của bạn để chọn ra các trường dữ liệu cần thiết để biểu diễn theo mục đích tuỳ chỉnh.



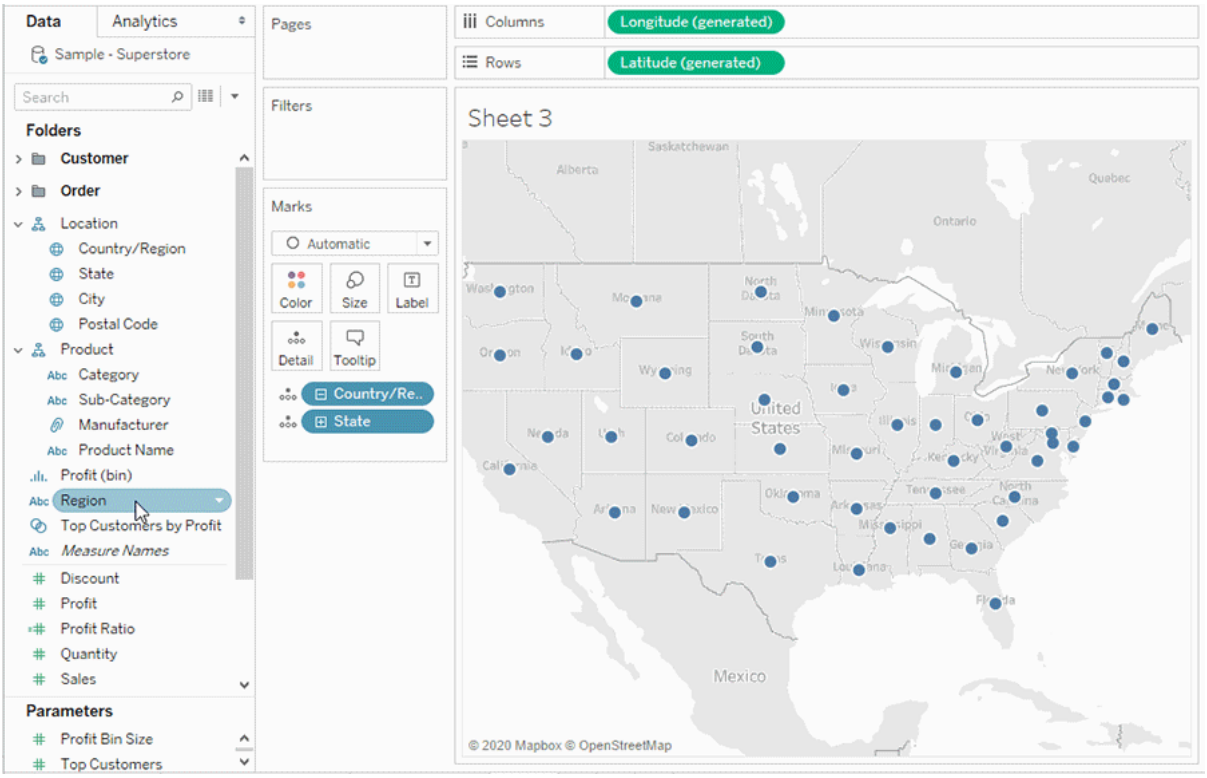
- Tuỳ chỉnh màu sắc cho biểu đồ của bạn. Tableau hỗ trợ chức năng giúp bạn có thể dễ dàng điều chỉnh màu sắc cho biểu đồ trực quan của bạn.



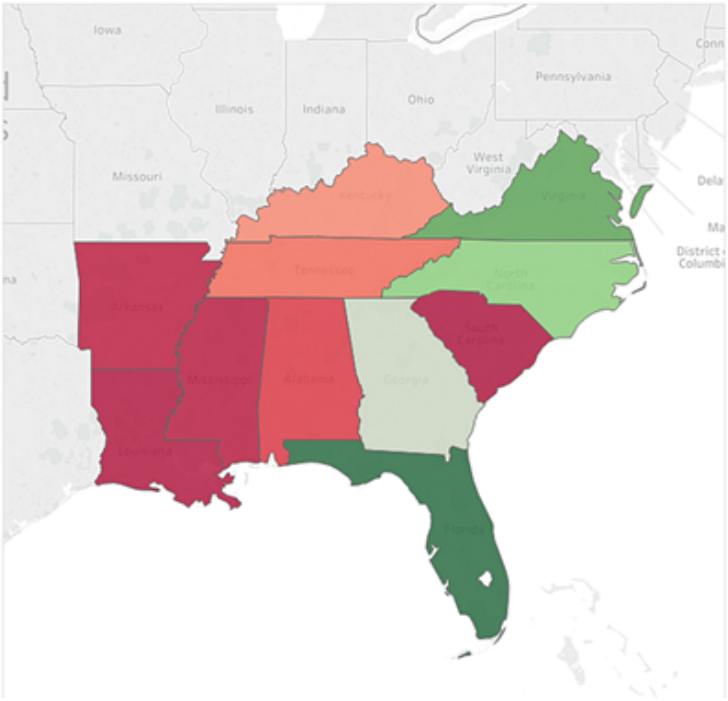
- Tương tác với biểu đồ của bạn bằng các công cụ mà Tableau cung cấp.
- Sao chép worksheets và lưu các thay đổi để tiếp tục khai thác và khám phá dữ liệu của bạn theo các cách khác nhau mà không làm mất các tác vụ trước đó của bạn.

d. Khám phá dữ liệu kiểu địa lý:

- Chúng ta cũng có thể xây dựng biểu đồ dạng bản đồ với Tableau.



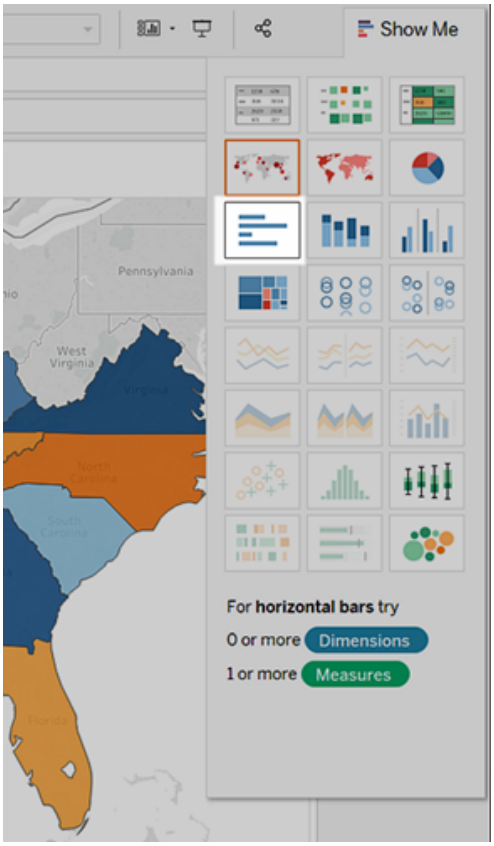
- Khi Tableau xác định được trường dữ liệu thược kiểu geographic, nó sẽ tự động tạo ra được một biểu đồ dạng bản đồ.



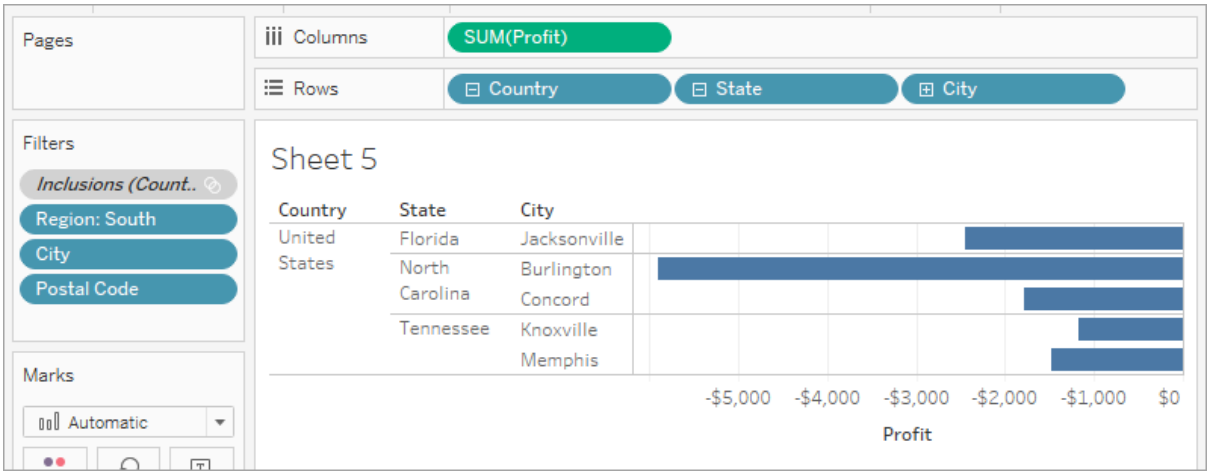
- Chúng ta cũng có thể tuỳ chỉnh màu sắc bản đồ với chức năng tuỳ chỉnh màu sắc bản đồ trên thẻ Marks.

e. Đi sâu vào chi tiết:

- Tableau hỗ trợ đa dạng và hầu như bao quát các loại biểu đồ trực quan phù hợp với nhu cầu trực quan thông dụng.



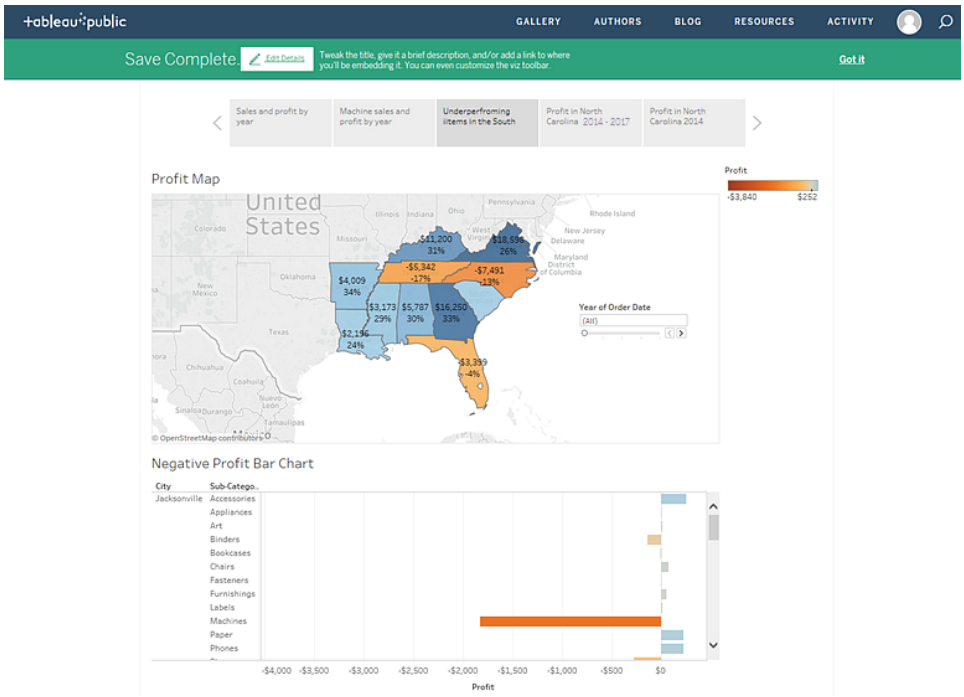
- Áp dụng đa dạng nhiều lớp filters dữ liệu khác nhau để lựa chọn được chính xác dữ liệu để trực quan.



f. Chia sẻ những kết quả của bạn:

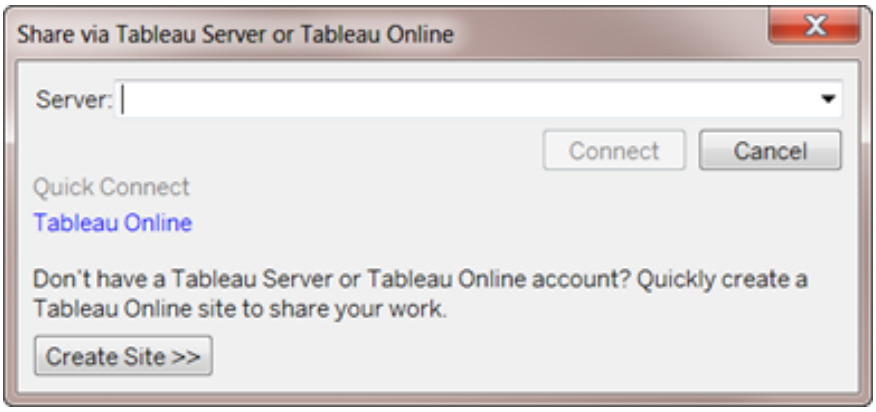
- Nếu bạn hoặc tổ chức của bạn không sử dụng Tableau Server, hoặc bạn muốn học hỏi, chia sẻ đơn thuần, hãy lựa chọn Tableau Public. Ngược lại, nếu muốn điều chỉnh quyền riêng tư, đảm bảo bảo mật dữ liệu hơn, bạn có thể sử dụng Tableau Server.

- Sử dụng Tableau Public để chia sẻ công khai những kết quả trực quan của bạn để nhóm của bạn hoặc cộng đồng có thể theo dõi và tham khảo trực tuyến.
- Kết quả được thể hiện trên trình duyệt web sau khi công khai lên Tableau Public.

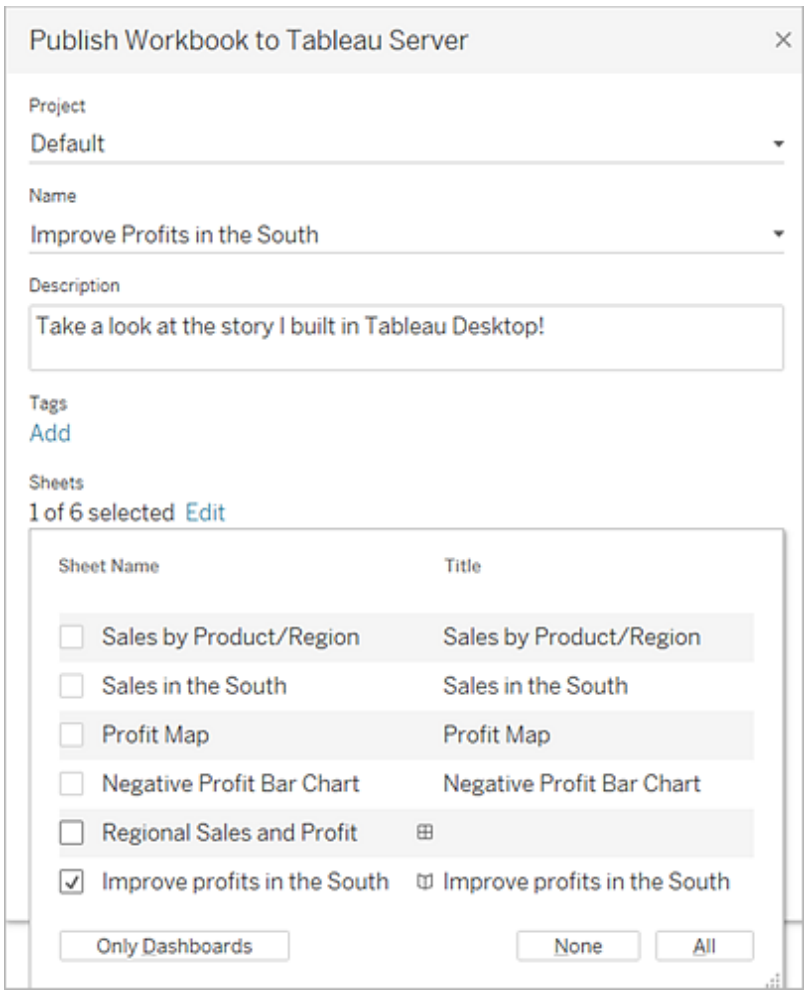


- Ngoài ra, chúng ta cũng có thể lấy link nhúng để chèn vào trang web, ứng dụng hoặc đơn giản là chia sẻ link qua các mạng xã hội phổ biến hiện nay như Facebook, Twitter.

- Cũng với những chức năng tương tự vậy, nhưng Tableau Server cho phép chúng ta lựa chọn server (địa chỉ IP) tuỳ ý để kết nối và chia sẻ dữ liệu.



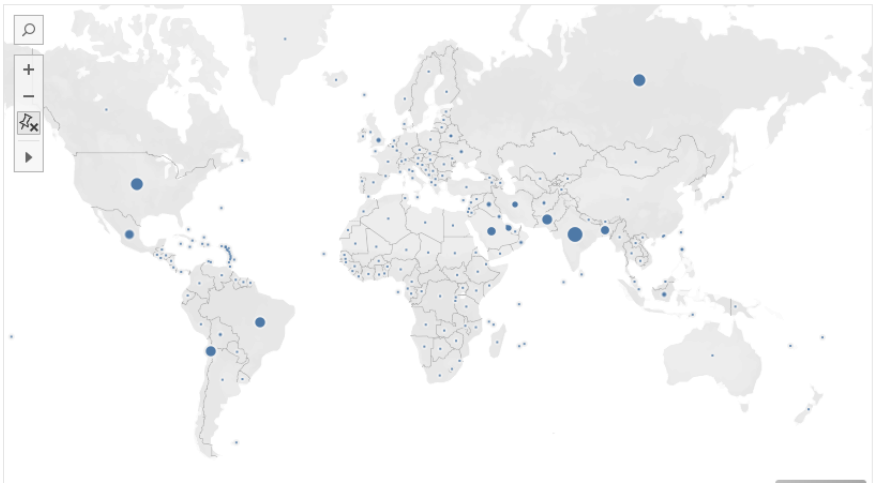
- Tuỳ chỉnh các tham số để chia sẻ kết quả trực quan dữ liệu lên server.



V. Trực quan dữ liệu với Tableau:

1. Trực quan một số loại biểu đồ với Tableau:

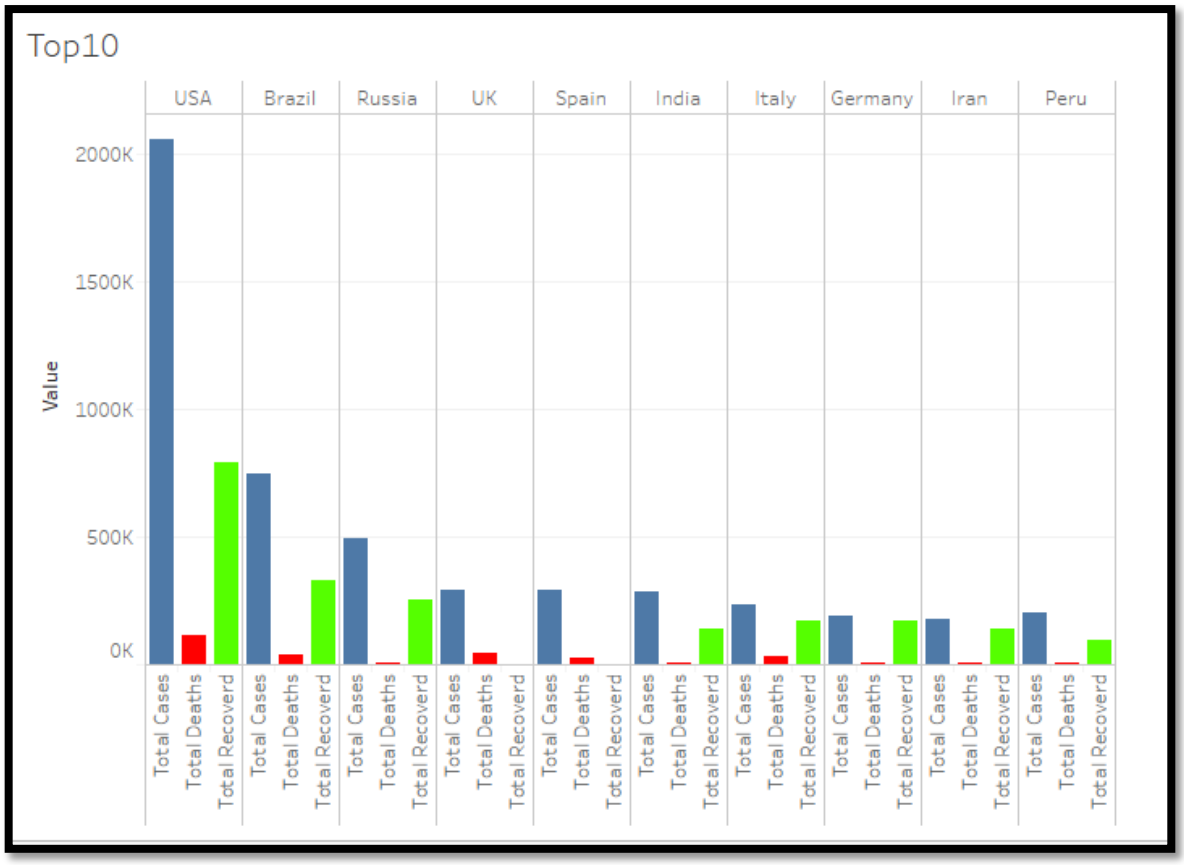
a. Symbol Maps:



- Để trực quan biểu đồ trên, nhóm lựa chọn hai trường dữ liệu là Country và New Cases để thể hiện thông tin số lượng ca nhiễm bệnh mới của tất cả các quốc gia trên thế giới.
- Nhìn qua biểu đồ ta có thể dễ dàng nhận ra quốc gia nào đang có số lượng ca nhiễm mới tăng nhanh và những khu vực nào đang có tình hình diễn biến phức tạp qua kích cỡ của các chấm tròn và mật độ của các chấm tròn.
- Chẳng hạn như qua biểu đồ này ta nhận thấy rằng Mỹ, Ấn Độ và Nga đang là 3 quốc gia có số lượng ca nhiễm tăng mạnh nhất. Khu vực Nam Á và Tây Nam á đang có tình hình bệnh khá nghiêm trọng.
- Ngoài ra biểu đồ này còn có tính năng tìm kiếm tên quốc gia bằng các nhập tên quốc gia đó vào mục tìm kiếm với icon là cái kính lúp.

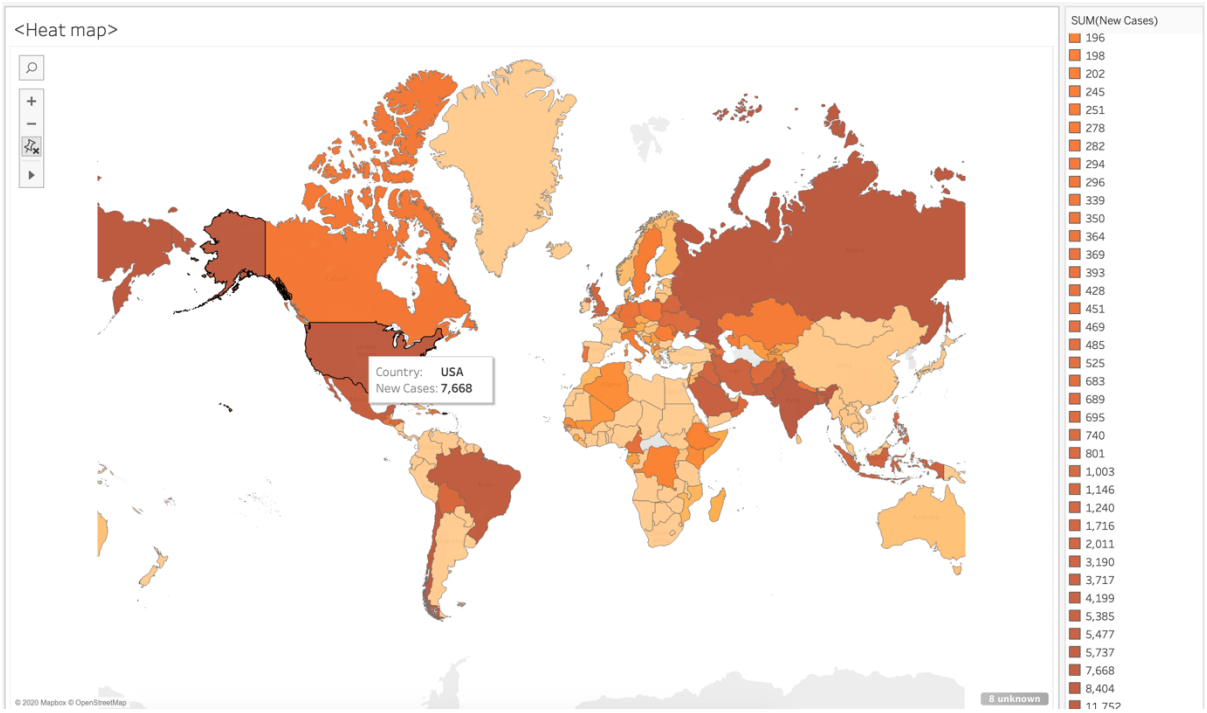
b. Side - By - Side Bars:

- Để trực quan biểu đồ trên, nhóm lựa chọn hai trường dữ liệu là Total Case, Total Deaths và Total Recovereds của top 10 quốc gia có tình hình dịch nghiêm trọng nhất để thể hiện thông tin tình hình dịch bệnh ở các quốc gia này.
- Ta thấy ở top 3 tỉ lệ phục hồi bé hơn 50% so với tổng ca mất, tuy vậy số ca tử vong ở top khá thấp so với tổng ca mất. Ở 2 quốc gia kế tiếp là UK và Spain thì dường như tỉ lệ phục hồi là không có hoặc quá thấp so với tổng ca mất, khả năng chữa dịch ở 2 quốc gia này còn khá kém. Các quốc gia còn lại trong top 10 cho ta thấy sự khả quan hơn khi tỉ lệ phục hồi lớn hơn 50% so với tổng ca mất và tỷ lệ tử vong khá thấp.
- Ở đây ta sử dụng màu đỏ để hiển thị cho Total Deaths vì ta cần sự nổi bật để báo động về cái chết trong cơn dịch. Màu xanh neon để hiện thị cho Total Recoverds nó mang lại cảm giác tích cực khi biểu thị chỉ số dữ liệu này.



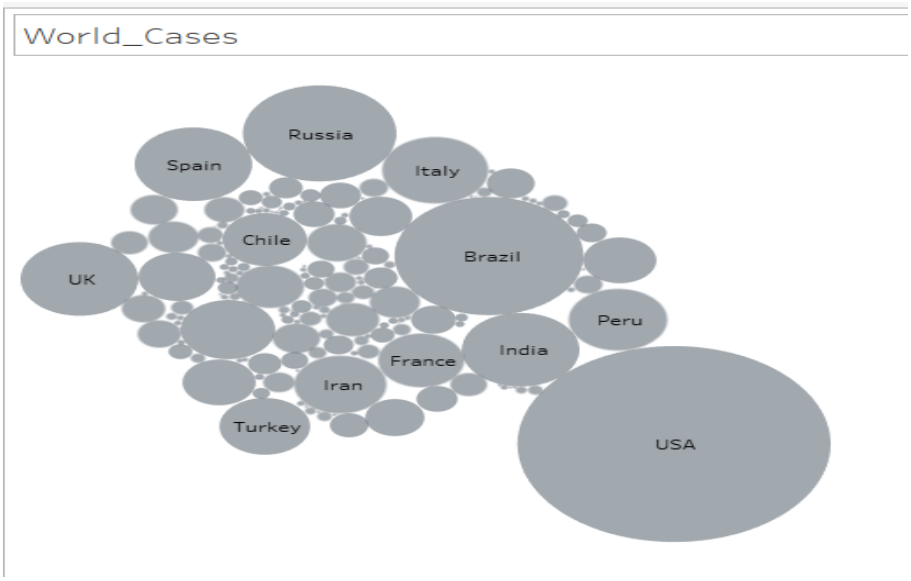
c. Global Heatmap:

- Để trực quan biểu đồ trên, nhóm lựa chọn hai trường dữ liệu là Country và New Cases để thể hiện thông tin số lượng ca nhiễm bệnh mới của tất cả các quốc gia trên thế giới.
- Ngoài ra, với lựa chọn màu sắc tông cam từ nhạt đến đậm thể hiện cho mức độ số lượng ca nhiễm. Mức độ màu cam càng đậm chứng tỏ cho quốc gia có số ca nhiễm mới càng lớn, ngược lại, khi sắc cam càng nhạt, chứng tỏ quốc gia có số ca nhiễm càng thấp.



- Với biểu đồ trên, ta có thể quan sát được số lượng ca nhiễm ứng với từng quốc gia bằng cách di chuyển chuột vào vùng quốc gia tương ứng.
- Như hình trên, ta có thể thấy Mỹ đang có 7668 ca nhiễm bệnh mới.
- Qua biểu đồ trên, ta có thể nhìn được khái quát toàn cảnh tình trạng nhiễm bệnh mới của các quốc gia trên toàn thế giới. Biết được quốc gia nào vẫn đang trong tầm nguy hiểm, quốc gia nào không còn ca nhiễm mới. Từ đó, có thể tổng kết được tình trạng dịch bệnh đang diễn tiến thế nào.

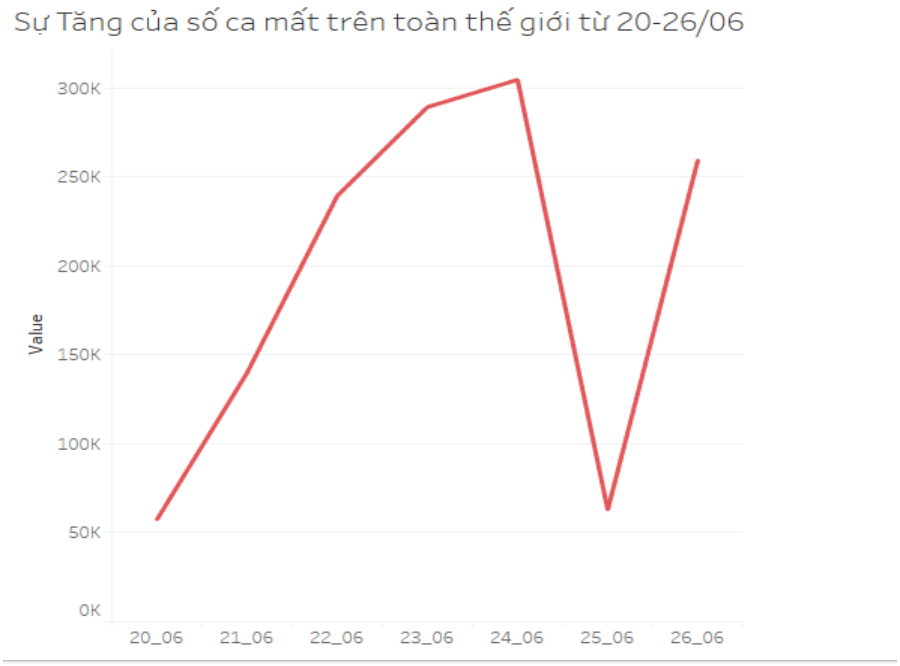
d. Packed Bubbles:



- Để trực quan biểu đồ trên, nhóm lựa chọn hai trường dữ liệu là Total Cases, Total Deaths và Country để thể mức độ biểu hiện tiêu cực dịch bệnh trên toàn cầu .
- Qua đây, ta dễ dàng thấy các quốc gia mà đang chịu ảnh hưởng tiêu cực nặng nề nhất của dịch bên thông qua kích cỡ của các hình tròn, khi lê chuột vào các hình tròn thì nó sẽ hiển thị lên chỉ số của tổng ca mất bệnh và tổng ca chết kèm theo tên quốc gia mà hình tròn đó hiển thị.
- Việc lựa chọn mày xám cho biểu đồ này để thể hiện sự tiêu cực, không sáng sủa của tình hình dịch bệnh đang diễn ra trên toàn cầu.
- Qua biểu đồ ta thấy Mỹ, Brazil, Nga đang là các quốc gia có tình hình dịch bệnh nghiêm trọng nhất trên thế giới.

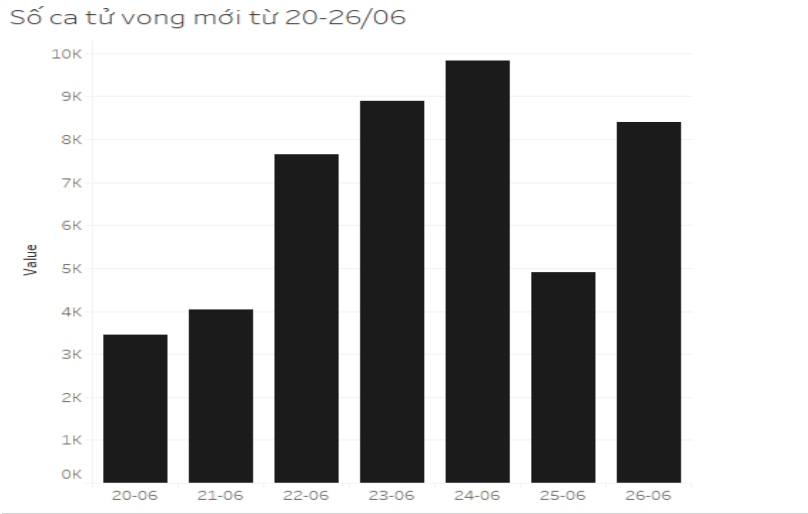
2. Thể hiện trực quan một số dữ liệu biến đối qua từng ngày. Rút ra ý nghĩa:

a. Lines:



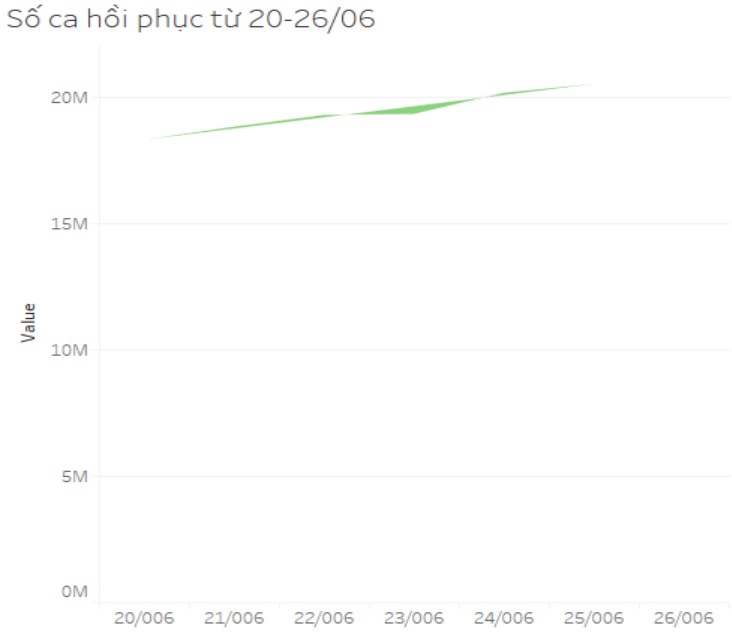
- Để trực quan biểu đồ trên, nhóm lựa chọn trường dữ liệu là New Cases từ ngày 20/06 - 26/06 để thể hiện mức độ tăng trưởng của dịch bệnh trên toàn cầu .
- Qua biểu đồ này ta có thể dễ dàng quan sát được sự tăng hay giảm của số lượng ca mất mới qua từng ngày trong giai đoạn từ 20/06 - 26/06. Ta có thể rê chuột vào từng đoạn của lines để biết rõ hơn về số lượng ca mất mới cụ thể của từng ngày.
- Như ta thấy số lương ca mất mới tăng nhanh và liên tục từ ngày 20 đến ngày 24/06 nhưng lại đột nhiên giảm mạnh vào ngày 25 và tăng manh trở lại ở ngày tiếp theo 26/06. Nó cho ta thấy sự phức tạp khó lường trước được của tình hình dịch bệnh.
- Ở đây ta chọn màu đỏ cho màu sắc của line để hiển thị sự cảnh báo đáng quan tâm và lưu ý nhất của dịch bệnh.

b. Horizontal bars:



- Để trực quan biểu đồ trên, nhóm lựa chọn trường dữ liệu là New Deaths từ ngày 20/06 - 26/06 để thể tình hình của các ca tử vong qua từng ngày của dịch bệnh trên toàn cầu.
- Qua biểu đồ này ta có thể nhìn thấy một cách rõ ràng nhất về sự biến động số lượng các ca tử vong. Ta có thể rê chuột vào vị trí các cột để có thể xem thông số rõ ràng hơn về số lượng ca chết mới của ngày.
- Như ta thấy ở biểu đồ trên lượng ca tử vong mới tăng nhanh và liên tục từ ngày 20 đến ngày 24/06 nhưng sau đó giảm mạnh trong ngày 25 rồi lại tăng mạnh vào ngày 26/06. Điều này cho ta thấy sự bất thường trong số ca tử vong, dịch đang diễn biến phức tạp và không nên chủ quan.
- Đáng chú ý khi so sánh biểu đồ này với biểu đồ line thể hiện cho diễn biến ca mất mới thì xu hướng của 2 biểu đồ này khá giống nhau, điều tăng nhanh và liên tục từ ngày 20/06 đến ngày 24/06 rồi lại giảm đột ngột vào ngày 25/06 và tăng mạnh vào ngày kế tiếp, qua đây nó cũng cho ta thấy rằng mối liên hệ cặp đôi của kiểu trường dữ liệu này.
- Việc lựa chọn màu đen để thể hiện cho số ca chết mới mà thay vì thể hiện màu đỏ. Vì ở đây chúng ta chỉ biểu diễn cho một trường dữ liệu nên không có sự tương phản trong dữ liệu ở đây, nó còn mang lại cảm giác tiêu cực cho chúng ta khi nhiều vào biểu đồ này và nó thể hiện đúng tính chất của dữ liệu.

c. Polygon:

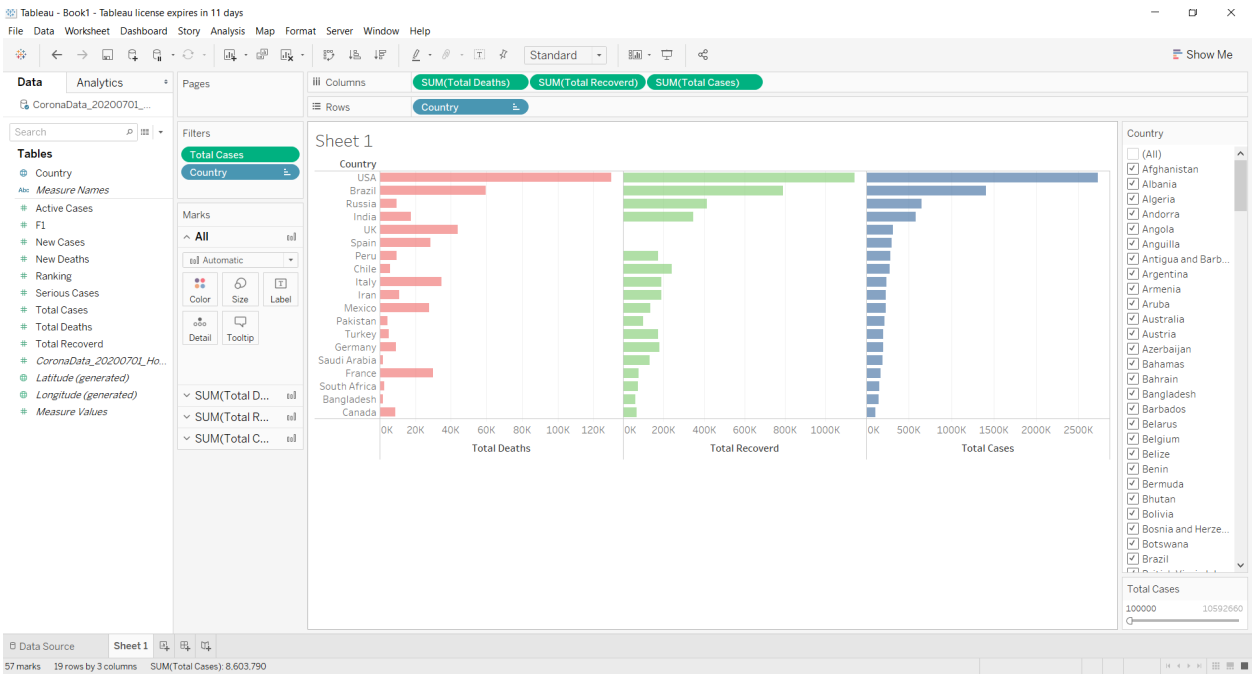


- Để trực quan biểu đồ trên, nhóm lựa chọn trường dữ liệu là Total Recoverds từ ngày 20/06 đến ngày 26/06 để thể tình hình phục hồi của thế giới trong dịch bệnh trên toàn cầu.
- Ở đây ta chỉ xét đến giá trị của sự phục hồi qua từng ngày nên việc sử dụng biểu đồ này cũng sẽ đáp ứng được nhu cầu của chúng ta thông qua độ lớn của vùng hiển thị. Ta cũng có thể rê chuột vào vùng hiển thị để biết chính xác hơn về con số của sự phục hồi.
- Như ta thấy trên biểu đồ thì tình hình phục hồi trong giai đoạn từ ngày 20 đến ngày 22 và giai đoạn từ ngày 24 đến ngày 26 khá tương đồng với nhau. Chỉ riêng từ 23 đến ngày 24 thì chỉ số phục tăng vượt trội so với các ngày còn lại trong giai đoạn thông việc độ to vùng hiển thị phình ra.
- Việc lựa chọn màu xanh neon để hiển thị cho tình hình phục hồi trong cơn dịch nhằm mang lại cảm giác lạc quan và tích cực cho người xem biểu đồ. Và nó cũng là tông màu mát nên cũng đem lại cảm giác dễ chịu hơn cho người xem.

3. Sử dụng các kỹ thuật được giới thiệu trong bài Manipulate View, Facet, Reduce, Embed để trình diễn trên Tableau với dữ liệu Woldometer:

a. Manipulate View:

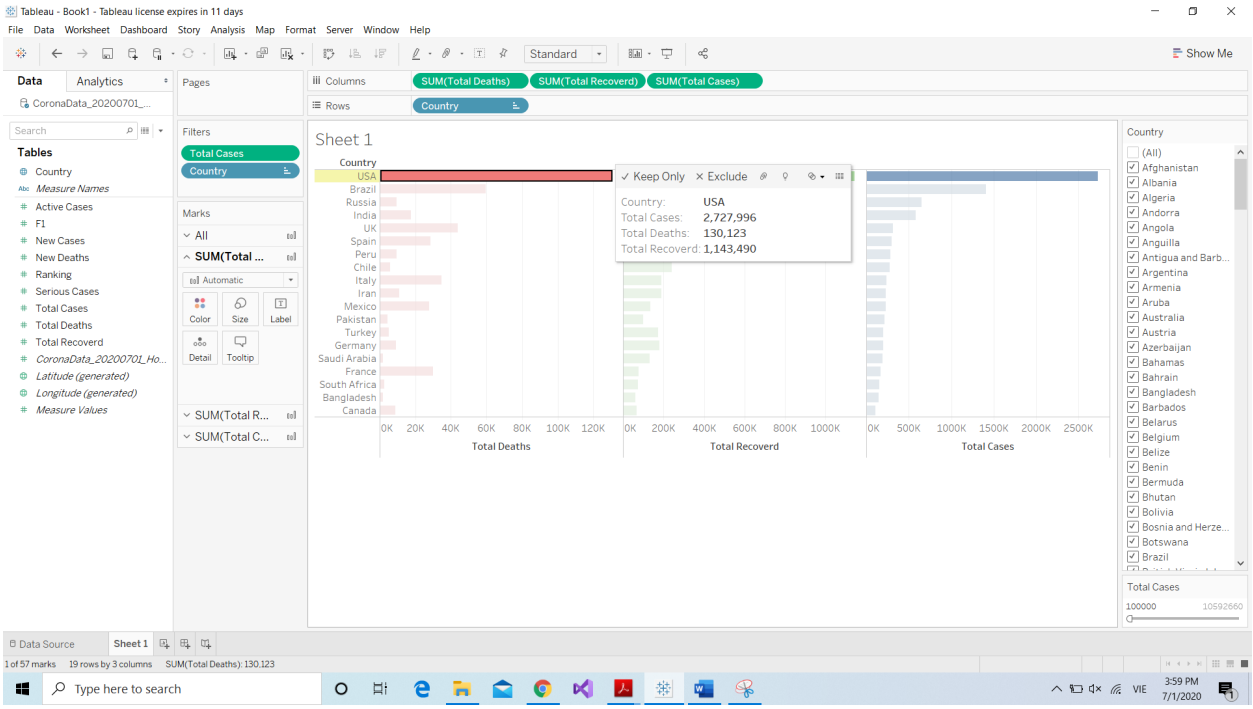
- Với Tableau, người dùng có thể **Change View over Time** tùy theo nhu cầu của mình.
- **Lý do lựa chọn**: Do dữ liệu quá nhiều để có thể visualization. Do đó đây là một khả năng hữu ích để có thể chọn ra những gì thích hợp nhất để hiển thị tùy thuộc vào nhu cầu của người sử dụng.
- Ví dụ:



Hình 1

- Hình 1: Hiện thị biểu đồ horizontal bars cho Total Cases, Total Deaths, Total Recoverd của các nước có ít nhất 100000 ca nhiễm và được sắp xếp theo tổng số ca nhiễm.
- **Ý nghĩa mang lại**: Có được cái nhìn tổng quan về tình hình hiện tại của một số nước đang có số ca nhiễm covid lớn trên thế giới. Ví dụ như USA vẫn đang là nước có số ca nhiễm và số ca tử vong lớn nhất hiện tại.

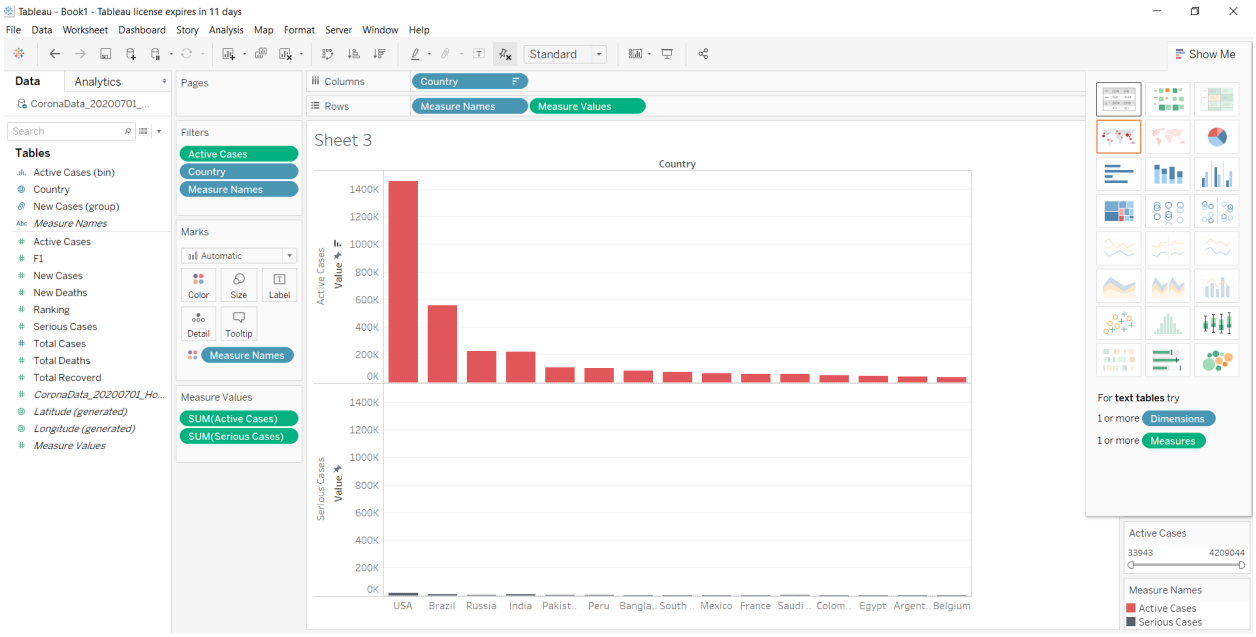
- Ngoài ra để xem được chi tiết dữ liệu của một đối tượng trên biểu đồ thì chúng ta chỉ cần **Select** vào đối tượng đó trên biểu đồ thì tableau sẽ hiển thị chi tiết thông tin về cột đó (Hình 2).



Hình 2

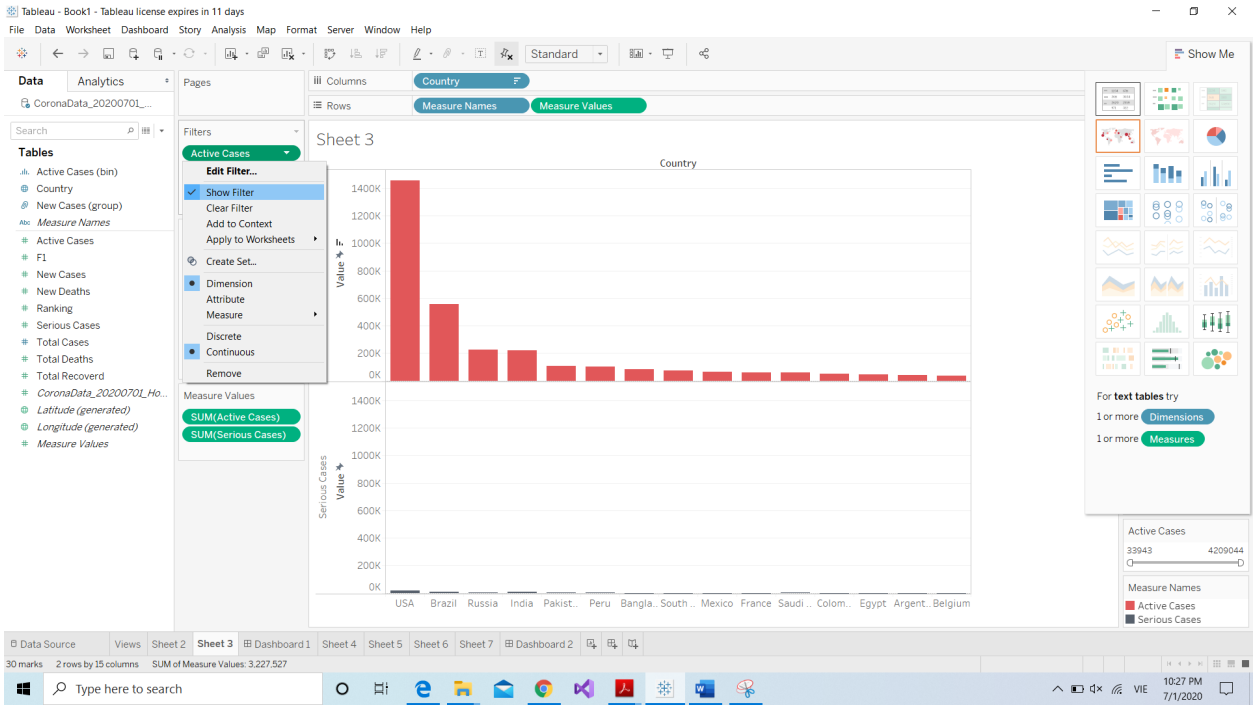
b. Reduce:

- Một trong những kỹ thuật để Reduce là Filter. Trong tableau có hỗ trợ Filters để giảm số lượng items cũng như lựa chọn các atribute mà người dùng mong muốn.
- **Lý do lựa chọn:** Những chức năng này khá dễ dàng và phù hợp với người mới bắt đầu, thao tác nhanh nhưng vẫn đem lại hiệu quả hiển thị như mong muốn.
- Ví dụ:



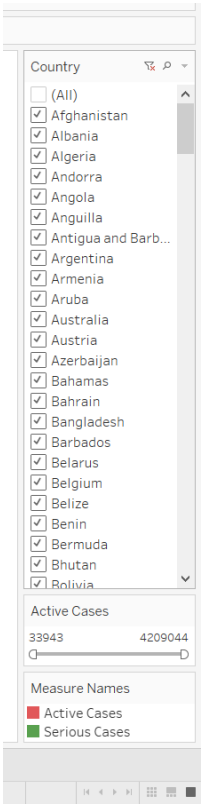
Hình 3. Biểu đồ horizontal bars hiển thị số ca nhiễm chưa khỏi bệnh và số ca nhiễm nghiêm trọng theo các nước.

- Trên thanh Columns, Rows để chứa các **Attributes** mà người dùng muốn trực quan. Cụ thể trong ví dụ này **Columns** chứa **Country**; Rows chứa **Active Cases** và **Serious Name.**
- Để có thể chọn các thuộc tính cần thiết thì có thể click chuột phải vào thuộc tính mong muốn trong cửa sổ **Filters** -> **Show Filter** (Hình 4).



Hình 3

- Sau đó một bảng bảng con sẽ xuất hiện bên phải worksheet, cho phép lọc nâng cao chi tiết từng thuộc tính (hình 5)
- Trong hình 3 sử dụng các yêu cầu lọc như sau:
  - **Country**: Chọn tất cả các nước trừ **Total:** và **World**
  - **Active Cases**: Chọn các nước có ít nhất 33943 ca nhiễm chưa được chữa khỏi.
  - **Mesure Names**: Chọn màu đỏ cho **Active Cases**, màu xanh sẫm cho **Serious cases**.
- Kết quả cho ra biểu đồ biểu thị số lượng Active Cases và Serious Cases cho 15 nước như hình 3.
- Ý nghĩa rút ra từ hình: Qua hình có thể thấy số ca nhiễm nghiêm trọng của 15 nước đừng đầu vẫn khá thấp so với số ca nhiễm chưa được chữa khỏi. Cho thấy các nước vẫn đang thực hiện khá tốt công việc chăm sóc bệnh nhân nhiễm bệnh covid-19.



Hình 4

VI. Áp dụng một số thuật toán máy học:

1. Bình phương nhỏ nhất thông thường (Ordinary Least Squares - OLS) :

- Đây là phương pháp được sử dụng rộng rãi nhất để ước lượng các tham số trong phương trình hồi quy. Để tối thiểu hoá tổng bình phương của các khoảng cách theo phương thẳng đứng giữa số liệu thu thập được và đường (hay mặt) hồi quy.

```
In [35]: from pandas import DataFrame
         import statsmodels.api as sm

         df = DataFrame(dataset,columns=['New Cases','Total Deaths','New Deaths','Total Recoverd','Active Cases','Serious Cases'])
         X = df[['New Cases','Active Cases']]
         Y = df['Serious Cases']
         X = sm.add_constant(X)
         model = sm.OLS(Y, X).fit()
         predictions = model.predict(X)

         print_model = model.summary()
         print(print_model)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          Serious Cases   R-squared:                       0.986
Model:                            OLS   Adj. R-squared:                  0.985
Method:                 Least Squares   F-statistic:                     7543.
Date:                Mon, 29 Jun 2020   Prob (F-statistic):          4.21e-204
Time:                        22:04:31   Log-Likelihood:                -1773.9
No. Observations:                 224   AIC:                             3554.
Df Residuals:                     221   BIC:                             3564.
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const          10.2060     45.440      0.225      0.822     -79.345      99.757
New Cases       0.2150      0.019     11.432      0.000       0.178       0.252
Active Cases    0.0115      0.000     26.879      0.000       0.011       0.012
==============================================================================
Omnibus:                       89.890   Durbin-Watson:                   2.267
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            11331.716
Skew:                          -0.143   Prob(JB):                         0.00
Kurtosis:                      37.843   Cond. No.                     3.50e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.5e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

- Mục tiêu ở đây là dự đoán / ước tính giá chỉ số chứng khoán dựa trên hai biến kinh tế vĩ mô: New Case và Active Case.
- Hai biến trên được dự đoán dựa vào: Serious Cases
- Một số thành phần quan trọng trong kết quả đưa ra:
  - ***Adjusted R-squared:*** Phản ánh sự phù hợp của mô hình. Các giá trị bình phương R nằm trong khoảng từ 0 đến 1, trong đó giá trị cao hơn thường biểu thị mức độ phù hợp tốt hơn, giả sử các điều kiện nhất định được đáp ứng.
  - ***Const coefficient:*** Là Y-intercept. Điều đó có nghĩa là nếu cả hai hệ số New Case và Active Case đều bằng 0, thì sản lượng dự kiến (nghĩa là, Y) sẽ bằng với hệ số const.

- ***New Case coefficient:*** biểu thị sự thay đổi của đầu ra Y do thay đổi một đơn vị New Case (mọi thứ khác được giữ cố định)
- ***Active Case coefficient:*** Đại diện cho sự thay đổi của sản lượng Y do sự thay đổi của một đơn vị trong tỷ lệ thất nghiệp (mọi thứ khác được giữ cố định).
- ***std err:*** phản ánh mức độ chính xác của các hệ số. Nó càng thấp, mức độ chính xác càng cao
- ***P >|t|:*** Là giá trị p-value. Giá trị p nhỏ hơn 0,05 được coi là có ý nghĩa thống kê
- ***Confidence Interval:*** Đại diện cho phạm vi mà hệ số của chúng tôi có khả năng giảm (với khả năng là 95%).

2. Phép phân tích thành phần chính (PCA)

- Phép phân tích thành phần chính là một thuật toán thống kê sử dụng phép biến đổi trực giao để biến đổi một tập hợp dữ liệu từ một không gian nhiều chiều sang một không gian mới ít chiều hơn nhằm tối ưu hóa việc thể hiện sự biến thiên của dữ liệu.

In [47]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df = DataFrame(dataset,columns=['New Cases','Total Deaths','New Deaths','Total Recoverd','Active Cases','Serious Cases'])
scaler.fit(df)
scaled_data = scaler.transform(df)

# we will specify number of components as 2
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(scaled_data)

# Now we can transform this data to its first 2 principal components.
x_pca = pca.transform(scaled_data)

print(scaled_data.shape)
print(x_pca.shape)

# We've reduced 30 dimensions to just 2! Let's plot these two dimensions out!
```

```
(224, 6)
(224, 2)
```

- Đầu tiên, chúng ta cần xử lý trước dữ liệu, tức là chúng ta cần chia tỷ lệ dữ liệu sao cho mỗi tính năng có phương sai đơn vị và không có tác động lớn hơn dữ liệu kia.
- Chúng ta sẽ chỉ định số lượng thành phần là 2.
- Sau đó chúng ta có thể chuyển đổi dữ liệu này thành 2 thành phần chính đầu tiên.
- Bây giờ chúng ta kiểm tra kích thước của dữ liệu trước và sau PCA
- Chúng ta đã giảm kích thước từ 6 xuống chỉ còn 2.

3. Hồi quy tuyến tính (Linear regression):

- Phân tích hồi quy tuyến tính là một phương pháp phân tích quan hệ giữa biến phụ thuộc Y với một hay nhiều biến độc lập X. Mô hình hóa sử dụng hàm tuyến tính. Các tham số của mô hình được ước lượng từ dữ liệu.

In [61]:
```
import seaborn as seabornInstance
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
%matplotlib inline
X = dataset['Active Cases'].values.reshape(-1,1)
y = dataset['Serious Cases'].values.reshape(-1,1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train, y_train) #training the algorithm

#To retrieve the intercept:
print(regressor.intercept_)
#For retrieving the slope:
print(regressor.coef_)

y_pred = regressor.predict(X_test)
df = pd.DataFrame({'Actual': y_test.flatten(), 'Predicted': y_pred.flatten()})
df
```

```
[56.04484045]
[[0.01608476]]
```

Out[61]:

	Actual	Predicted
0	0	59.245708
1	0	56.044840
2	2	58.071520
3	0	63.508169
4	0	56.044840
5	0	60.805929
6	90	156.880201
7	0	56.044840
8	5	75.443061
9	1077	1746.054493
10	0	63.379491
11	274	314.253493

- Đầu tiên chúng ta định dạng lại kiểu dữ liệu.

- Tiếp theo, chúng ta chia 80% dữ liệu cho train set trong khi 20% dữ liệu để train test sử dụng code bên dưới.
- Sau khi chia dữ liệu thành các tập train và test, cuối cùng, thời gian là để training thuật toán của chúng tôi. Để làm được điều đó, chúng ta cần nhập lớp tuyến tính, khởi tạo nó và gọi hàm fit () cùng với training data.
- Như chúng ta đã thảo luận rằng mô hình hồi quy tuyến tính về cơ bản tìm thấy giá trị tốt nhất cho phần chặn và độ dốc, dẫn đến một dòng phù hợp nhất với dữ liệu. Để xem giá trị của phần chặn và độ dốc được tính toán bằng thuật toán hồi quy tuyến tính cho tập dữ liệu, chúng ta dùng lệnh print để in ra.
- Kết quả phải là khoảng 56.0448 và 00160476 tương ứng.
  Điều này có nghĩa là cứ một đơn vị thay đổi Active Case, sự thay đổi Serious Case là khoảng 0,0016%.
- Sau đó đưa ra dự đoán về dữ liệu thử nghiệm, so sánh các giá trị đầu ra thực tế cho X_test với các giá trị dự đoán.

VII. Mức độ hoàn thành:

1. Mức độ hoàn thành đồ án:

STT	Công việc	Mức độ hoàn thành
1	- Tìm hiểu về công cụ Tableau. - Trực quan biểu đồ heat map.	100%
2	Trực quan dữ liệu một ngày và nhiều ngày bằng công cụ Tableau.	100%
3	Sử dụng các kỹ thuật được giới thiệu trong bài Manipulate View, Facet, Reduce, Embed để trình diễn trên Tableau với dữ liệu Woldometer.	100%
4	- Chạy một số thuật toán Machine Learning đơn giản.	100%

2. Mức độ hoàn thành công việc của các thành viên:

STT	Thành viên	Mức độ hoàn thành
1	Nguyễn Huỳnh Thảo Nhi	100%
2	Lê Quang Quí	100%
3	Lê Bá Quyền	100%
4	Nguyễn Lê Trường Thành	100%

VIII. Tham khảo:

[1]. https://viblo.asia/p/gioi-thieu-principal-component-analysis-07LKXpq2KV4

[2].https://towardsdatascience.com/a-beginners-guide-to-linear-regression-in-python-with-scikit-learn-83a8f7ae2b4f

[3]. https://datatofish.com/statsmodels-linear-regression/

[4]. https://datascienceplus.com/principal-component-analysis-pca-with-python/

--- Hết ---