**<** **Visualization of global COVID-19 data on maps>**

**Testing Plan & Report**

**<Version 2.0>**

**<Date 6.1>**

<Project team members>

陈可轩1930026012

戴靓1930026020

李涵1930026064

齐子瞻1930026109

吴越1930001085

**Computer Science and Technology Program**

**United International College**

1. Document Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| ***Change Date*** | ***Changed By*** | ***Version*** | ***Change Description*** |
| *01/07/2012* | *Lisa Lee* | *1.0* | *Prepared Document* |
| 06/01/2021 | All group | 2.0 | *Put test information* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Table of Contents

[1. Document Change Log 2](#_Toc394839007)

[2. Table of Contents 3](#_Toc394839008)

[3. Design Overview 4](#_Toc394839009)

[4. Tools and Standards 5](#_Toc394839010)

[4.1. Tools 5](#_Toc394839011)

[4.2. Standards 5](#_Toc394839012)

[5. User Interface Design 6](#_Toc394839013)

[5.1. Usage Scenario 1 6](#_Toc394839014)

[5.2. Usage Scenario 2 6](#_Toc394839015)

[6. Database Design 7](#_Toc394839016)

[7. Diagrams 8](#_Toc394839017)

1. Testing Plan

*<Please provide your test plan here. You can refer to the following sections.>*

## Overview of the Testing Plan

*<Please give an overview of your test plan, including the purpose, brief description of your testing, name of the persons responsible for the test, etc.>*

Purpose: To make sure our date is correct for us to use. And make sure the type and the data are correct.

We divide our project into several parts, and each team member is responsible for his own part. Finally, because the final result we do is a complete map and data graph, we combine these functions in a GUI interface to facilitate the use of users. There are corresponding buttons in GUI. You can jump to different interfaces by clicking different buttons. Each button corresponds to the running of a method, and everyone debugs and finally integrates it into GUI.

The basic test method of each method is to read and display the data. Some images need animation and time line, so we should pay special attention to the accuracy of dynamic data graph.

## Testing Methods

*<Please give the methods are used in the test. You can refer to professional testing methods or give the steps and scenarios of each method you design.>*

### Cumulative

First, the data we used have something wrong. For example, they treat Taiwan as a country. So, we use replace method to put all Taiwan’s information to China. After we put the Taiwan to China, we merge the china data and other world data together. Second, for the csv files it has Province/State for some country. So, the map will show some countries by States and some countries by country itself. For this, we use ‘groupby()’ method to let them group by country and use ‘sum()’ method to put their information together as the country.

### World

First, because of colonization some European countries was locate in the ocean. To deal with this problem, we first group by country first and use ‘mean()’ to put them to the correct position. And through this we put use folium to create a static confirmed map about the world. Second, we use melt method to uniform data format about the information. Our data format is not uniform, so we use ‘strftime()’ to uniform the data format. For some country they don’t have confirmed data we want also see their information as 0. We use fillna(0) to put the empty to the 0.

### China

I used pandas to read the data and Pyechart module to draw the map in my part. In order to read the data successfully, the excel file must in specific form. After the tolist() function change the data form, the pyechart can draw the map. In the timeline function, I used loop to draw different days’ map. It can play the maps automatically.

### US

At the beginning, I used readcsv to directly read the data from GitHub. However, I found that I needed to turn on the VPN when I was running. Otherwise, I would report an error and it took a long time to run. So I changed to read the data from the local CSV file instead. Here, in order to use the scattergeo function, we need to convert the U.S. state name to two letter abbreviation, so we read another CSV containing the full name and abbreviation of the U.S. state to convert. Successfully read the data in the CSV and draw pictures with scattergeo, including the production of animation and time bar.

### ChinaAll & ChinaToday

Method used in ChinaAll plot&ChinaToday

<1> Imports: json, requests, matplotlib.pyplot, numpy

<2> Steps:

1. Crawl the data: Since we need to get real-time data, I refer to a blog about crawling the weather forecast, so we request the address to get the data. And since the data in the web page is of json type, json.loads () method is used to convert the json type to the basic Python type dictionary.
2. Deal with data: Use print() method to check which elements the webpage have, the plot is about China, so we need extract data from province. After successful, use a loop to initialize each extracted data to 0, and then read today's data. If don’t initialize each data of province, it will get KeyError:台湾. After updating the data to 0, there is no problem.
3. Draw the figures: Because the data is in Chinese, two sentences are added to prevent garbled statements (plt.rcParams['font.sans-serif'] = ['SimHei'], plt.rcParams['axes.unicode\_minus'] = False)

Then, starting just normal drawing, the zip () method is also used in the drawing to convert the extracted provinces and data into lists, respectively.

### Comparing

<1>Method used in the bar chart race plot :

1.Imports: pandas, matplotlib.pyplot, matplotlib.ticker, matplotlib.animation, numpy, IPython.display.

2.Specific methods:

(1)The designing method is : [def drawchart(Days)]

[def drawchart(Days)], the basic designing is draw a chart of the top ten country by method of drawchart using the argument(Days), and then use the method animation.FuncAnimation() to draw the chart of each day and make an animation.

(2)For the method that the imports have, there is some of it: matplotlib.rcParams['animation.embed\_limit'] : to expand animation limit

pd.read\_csv : to read csv file

animation.FuncAnimation() : to draw plot by frames

HTML(animator.to\_jshtml()) : to show the animation

And some basic method like ax.text(), plt.subplots, plt.show() and polishing method and so on.

<2>Steps of the plot designing:

1.Make the data into the form of Country, Date, Confirm, Days to meet the need of build the bar- chart-race plot in animation by frames.

2.For the code, I first use pd.read\_csv to get the data and use dataframe to get the data columns that needed.

3.Give each of the country a color in a dictionary and write the draw chart method (the draw chart method already mentioned in <1>2.(1) )

Explain draw chart method in details :

First the method of plt.xkcd() is to make it in comic style of the plot.

Then using dataframe to get the information of only the top ten country and set the color in the dict color.

Next, for each of the country parts, write the name and confirmed numbers by ax.text(), and write the date time by ax.text(), then do the polish work using ax.xaxis(), ax.margins(), ax.grid(), ax.text(), to show a nicely legend, title, and display the plot in a nice position.

4.Use the animation.FuncAnimation() to draw the chart of each day, using HTML(animator.to\_jshtml()) to show, and using animator.save() to save the animation plot into a gif file.

### GUI

It is mainly about the button layout and prompt information layout of GUI, and whether the corresponding function of each button can be realized. After a simple debugging, each function can run, click the button to jump out of the corresponding icon and data.

1. Testing Report

## Overview of the Testing

*<Please give an overview of the testing you done. You can describe the overall result, time consumed during testing, and / or other information.>*

The data we read from the web and CSV are accurate. Besides the map of China, we focus on the 2 months that COVID-19 has just produced and provide more visualization functions. The map time bar in the US and the world has been updated to May. The animation we used to compare cases in different countries was also updated to May. China's epidemic can also update the daily cases in real time, and display the data of cumulative cases in the form of histogram.

Finally, in terms of GUI, buttons can also jump to various functions. The histogram of China's epidemic situation and the map of China's epidemic situation can be run in about 1 second, and the map will be saved. The maps of the world and the United States can be run in 2 to 3 seconds. It takes about 3 to 4 minutes for the first run of the global epidemic comparison animation, but the running results will be saved in the folder.

## Error Report

*<Describe the errors that have been found during the testing.* ***Only*** *the test scenarios that produced the unexpected results should be listed>*

### US: HTTP Error 403: Forbidden

At the beginning, my map data was read from the CSV file on GitHub, and the error of rejecting the request would be reported, which can be solved after opening the VPN. In order to prevent this error, I put the CSV file in the project folder for easy reading.

### World: KeyError: ('Country/Region', '5/23/21')

We use add the [] to treat them as matrix and it will run successfully.

### ValueError: cannot set a row with mismatched columns

Since we don’t put China’s longitude and latitude to put in the data. So, we use Wuhan’s location, which is 30.9756 and 112.2707 to represent China.

### KeyError: 'Country/Region'

Because we merge countries by Country/Region, so we should reset\_index to add a columns which is that. So, we use other\_countries = other\_countries.reset\_index() to fix this problem.

### 4.2.5. KeyError: '台湾'

We get this error before you initialize it, and then we don't get it after it's initialized.

### 4.2.6. RuntimeWarning: Glyph 21488 missing from current font. font.set\_text(s, 0.0, flags=flags)

The Chinese characters on the figure will appear garbled, and neither Chinese nor negative signs will be displayed, then add this two statements plt.rcParams['font.sans-serif']=['SimHei'],

plt.rcParams['axes.unicode\_minus'] = False which could be correct.

### 4.2.7. Comparing(solved) : Animation size has reached 20989569 bytes, exceeding the limit of 20971520.0.

The error : Animation size has reached 20989569 bytes, exceeding the limit of 20971520.0. If you're sure you want a larger animation embedded, set the animation.embed\_limit rc parameter to a larger value (in MB). This and further frames will be dropped.

To avoid the limit of the animation, add the code : matplotlib.rcParams['animation.embed\_limit'] = 2\*\*256

1. `Contributions

*<Please provide the contributions of team members of your group.>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student ID | Name (in Pinyin/Chinese) | Contribution (in %) | Responsibilities/Work Done | Remarks |
| 1930026064 | Li Han | 20% | Cumulative & World |  |
| 1930001085 | Wu Yue | 20% | China |  |
| 1930026109 | Qi Zizhan | 20% | US & GUI |  |
| 1930026020 | Dai Liang | 20% | ChinaAll&ChinaToday |  |
| 1930026012 | Chen Kexuan | 20% | Comparing(bar-chart-race) |  |