ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

Visual Analysis of National Epidemic Data Based on PyEcharts

Lei Zhang[1], Hao Wu[1,*]

[1]School of Management Science and engineering, Anhui University of Finance & Economics, Bengbu, 233030, Anhui, China
**Corresponding author*

Abstract: *In order to help the public fully understand the epidemic situation and epidemic development trend throughout the country. In this paper, Pycharm software is used to form a web page visual interactive image through PyEcharts to display and analyze the existing epidemic data from multiple angles and aspects.*

Keywords: *Data visualization; PyEcharts; Epidemic situation; Data analysis*

1. Introduction

The outbreak of novel coronavirus in early 2020 is characterized by high transmissibility, high infection rate and regional outbreak. Novel coronavirus can cause pulmonary dysfunction, abnormal body temperature, cough and other symptoms, and may lead to death in severe cases. Up to now, the international epidemic situation is steep, and the epidemic problem is still one of the most concerned topics of the Chinese people.

The case data about confirmed epidemic, death and cure are updated in real time all over the country every day. How to intuitively and concisely express the distribution characteristics of these spatial data is an important issue in the analysis and display of epidemic data. Among them, GIS shows powerful functions in this multi-level, such as the processing, analysis and visualization of spatial information, as well as the prediction of the development trend of events on this basis. Information visualization plays an important role in the macro-control, analysis and decision-making of epidemic situation, and will have an important impact on human future social life, production mode and work. At the same time, it will also change the methods of human storage, expression and transmission of geospatial information. Therefore, the real-time monitoring, visual display and analysis of epidemic data is of great significance.

According to the actual needs of current epidemic prevention and control, this paper uses Python (PyEcharts Library) to visually process the existing epidemic data, realize the multi angle display of epidemic situation, and help people fully understand the latest epidemic data and recent epidemic development trend.

Visualization technology is to display a large amount of abstract and difficult to compare data intuitively and clearly by using relevant visualization tools and processing technologies, so as to facilitate the analysis, comparison and processing of data, get the information and conclusions people want[1], help people find the laws of unknown things[2], and strengthen the role of data mining[3].

The data visualization technology platform has been applied earlier abroad, such as IMany Eyes visualization platform of IBM, data visualization Laboratory of New York Times, ICharts and Rawdesign. The corresponding analysis tools, such as OpenGL and Gephi, are mature, widely used and have good results.

Data visualization technology was introduced and developed late in China, but it was also quickly applied. For example, the "360 Star Map" and "cheater map" developed by Qihoo company allow users to identify Internet scams by using the massive data obtained from the Internet. Alibaba uses the massive data of Taobao to analyze and visualize the Taobao index, which not only provides a purchase reference for a large number of Taobao sellers, but also provides a shopping guide for Taobao buyers, and provides new opportunities for third-party users. The B /S-type data visualization platform built by MATLAB and Java Web can analyze, process and visualize a large amount of data in the marine field[4]. In terms of visualization tools, Echarts mainly focuses on the visualization framework of the front-end interface, and

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

pays less attention to the structure that users pay more attention to.

This paper is based on B/S model. The front end adopts HTML page display, with intuitive data display. Users only need to use mainstream browsers to access. The back end uses Python's third-party tools, such as numpy, pandas, Matplotlib, wordcloud, JSON, request, and then uses the visualization tool PyEcharts to draw visual charts.

2. Methods

The data visualization platform is mainly divided into three modules: receiving module, processing module and visualization module. The specific structure is shown in Figure 1:

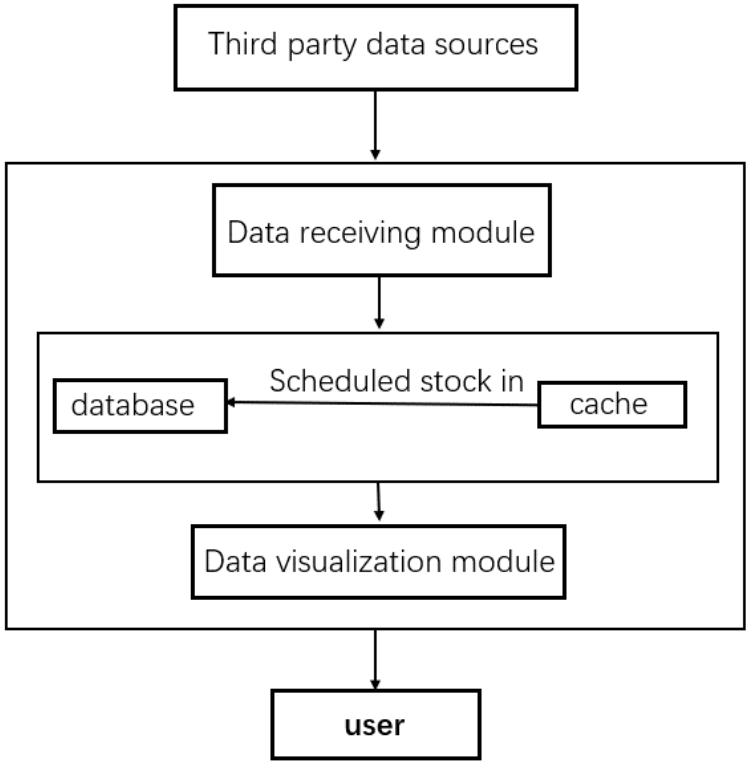

Figure 1: Module architecture of data visualization platform.

2.1. Data receiving module

The work of the data receiving module is to obtain the daily epidemic diagnosis, cure and death toll of each city in the country through the web crawler in the third-party data source.

2.2. Data processing module

Integrate the crawled epidemic data to extract useful information, format the obtained useful data, and store the data in a database that is not easy to lose and easy to call.

2.3. Data visualization module

The development platform of this paper is Pycharm. The development language is Python 3.

2.4. Key technologies

PyEcharts is a third-party information visualization framework in Python. It can directly run in the Python environment to generate corresponding HTML files. Open the corresponding HTML files to display the generated Echarts charts.

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

3. System Framework

The back-end main structure of the platform includes data layer, code running layer, front-end display layer, etc. the system framework is shown in Figure 2:

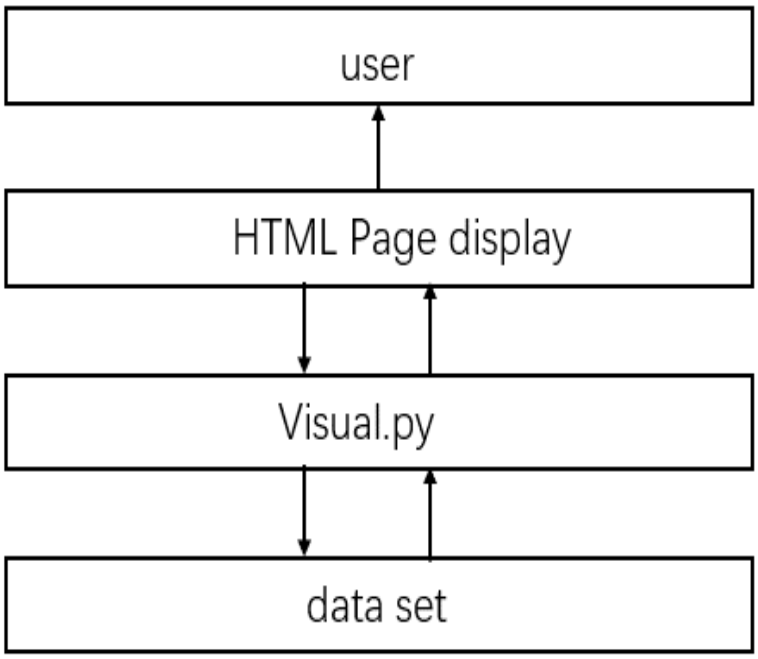

Figure 2: System framework.

The visual analysis process includes data collection and processing, data visual display and other steps. The purpose of data visualization analysis is to display a large amount of abstract and difficult to compare data intuitively and clearly by using relevant visualization tools and processing technology, so as to facilitate the analysis, comparison and processing of data and get the information and conclusions people want.

(1) Data acquisition and processing.

Collect data by looking for a third-party database and preprocess the data to make it meet the needs of users; Load the data into the database, etc., so that the data can be saved for a long time.

(2) Visual display of data.

Third party libraries such as visual tools PyEcharts, Matplotlib and wordcloud are used to process and extract data with the help of corresponding Python algorithms. At the end of the algorithm, corresponding HTML files will be generated, and the images generated by PyEcharts can be displayed on the appropriate browser.

4. Display and Analysis of Visualization Results of Epidemic data

4.1. Distribution map of the number of confirmed epidemic cases nationwide

The rendering effect of the distribution map of the number of confirmed epidemic cases nationwide is shown in Figure 3. Different colors reflect the current number of confirmed cases in different provinces. When the mouse hovers over different provinces, the current number of confirmed cases in that province will be displayed. The final data visualization display file is in the "visualization of the number of confirmed epidemics nationwide. Html". By clicking the corresponding browser, users can visually see the distribution of the number of confirmed epidemics nationwide and in each province. At present, most of the cities with serious confirmed cases are coastal cities, and the main confirmed cases come from overseas imported cases. It is necessary to strengthen the epidemic control of the entry of overseas personnel.

Specific code display:

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

```
import pandas as pd
import numpy as np
from pyecharts.charts import Map
import pyecharts.options as opts
from pyecharts.globals import ChartType
from pyecharts.render import make_snapshot

# Read the file data and display it in the form of DataFrame
df = pd.read_csv(r'D:\pythonProject\ visualization \china-covid19(2020-12-06).csv')
# The number of confirmed cases was extracted from the total data, grouped by provinces
area_data = df.groupby("prov")["confirmed"].sum().reset_index()
#    Extract columns in area_data
area_data.columns = ["prov", "confirmed"]
print(area_data)
# Using PyEcharts to generate HTML file of data display page
(
# Mapping of epidemic situation in China and configuring corresponding parameters
    Map()
        .add("", [list(z) for z in zip(list(area_data["prov"]), list(area_data["confirmed"]))], "china",
             is_map_symbol_show=False)
        .set_global_opts(title_opts=opts.TitleOpts(title="2020_nCoVMap of the total number of confirmed cases in various regions of China"),
                         visualmap_opts=opts.VisualMapOpts(is_piecewise=True,
                                                           pieces=[
                                                               {"min": 100000, "label": '>100000', "color": "#893448"},# maxNo upper limit
                                                               {"min": 50000, "max": 99999, "label": '50000-99999',"color": "#ff585e"},
                                                               {"min": 20000, "max": 49999, "label": '20000-49999',"color": "#fb8146"},
                                                               {"min": 10000, "max": 19999, "label": '10000-19999',"color": "#ffA500"},
                                                               {"min": 1000, "max": 9999, "label": '1000-9999',"color": "#ffb248"},
                                                               {"min": 0, "max": 999, "label": '0-999',"color": "#fff2d1"}]))
    ).render("Visualization of the number of confirmed epidemics nationwide.html")
```

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

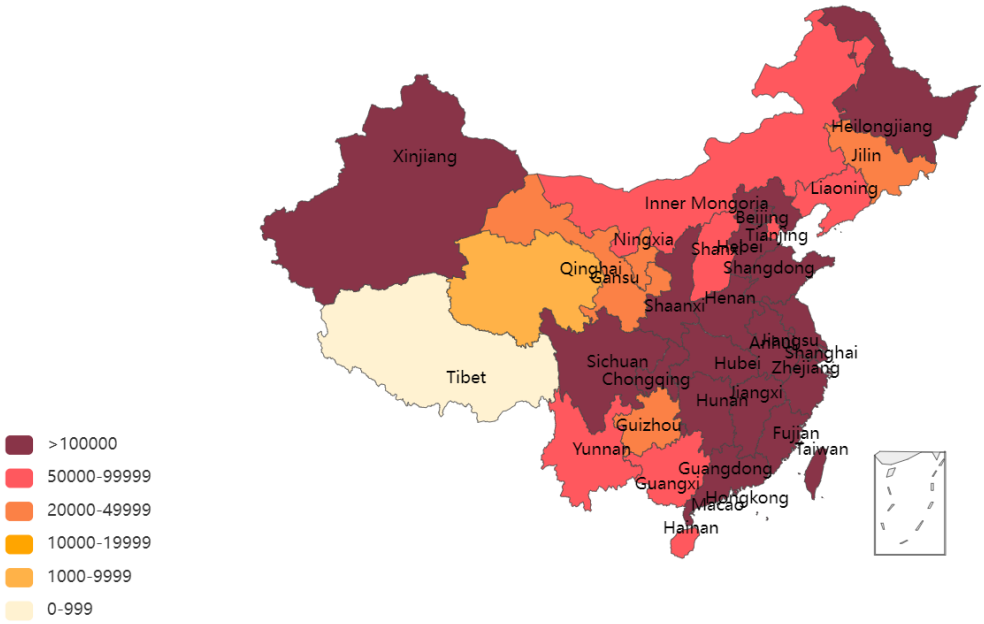

Figure 3: Distribution of confirmed cases in China.

4.2. Distribution map of the number of cured people in the national epidemic situation

The rendering effect of the national epidemic cured population distribution map is shown in Figure 4. Different colors reflect the current cured population in different provinces. When the mouse hovers over different provinces, the current cured population in that province will be displayed. The final data visualization display file is in the "visualization of the number of cured epidemic cases nationwide. Html". By clicking the corresponding browser, users can visually see the distribution of the number of cured epidemic cases nationwide and in each province. At present, most of the cured people are in coastal cities. Because there are more people diagnosed in coastal cities, the number of cured people is more than that in central cities.

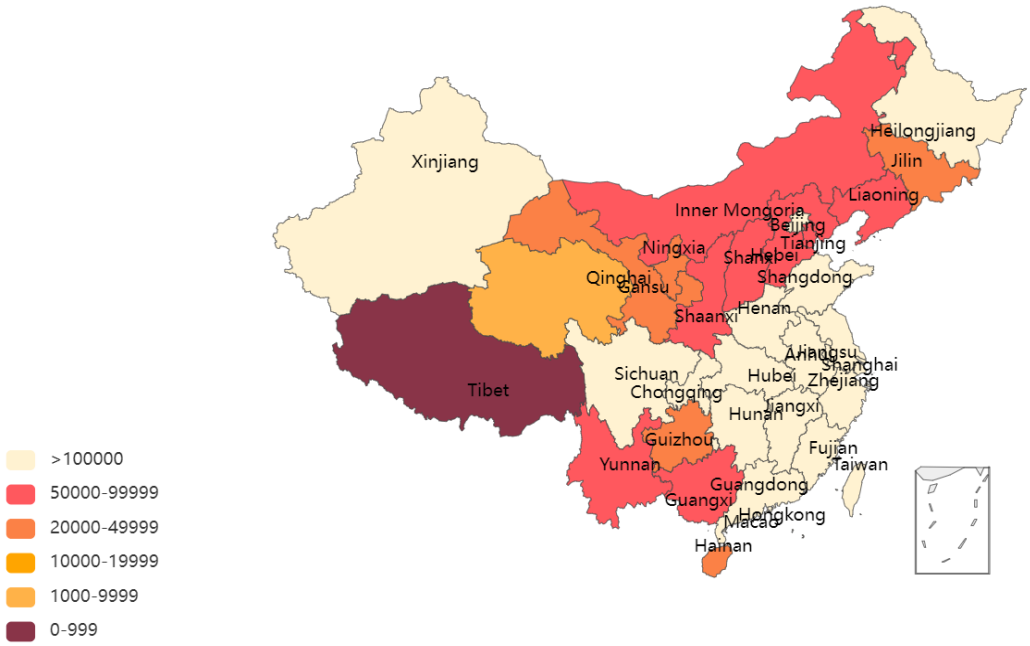

Figure 4: Distribution of cured persons of epidemic situation in China.

Specific code display:

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

```
import pandas as pd
import numpy as np
from pyecharts.charts import Map
import pyecharts.options as opts
from pyecharts.globals import ChartType
from pyecharts.render import make_snapshot

# Read the file data and display it in the form of DataFrame
df = pd.read_csv(r'D:\pythonProject\ visualization \china-covid19(2020-12-06).csv')
# The number of confirmed cases was extracted from the total data, grouped by provinces
area_data = df.groupby("prov")["confirmed"].sum().reset_index()
#    Extract columns in area_data
area_data.columns = ["prov", "confirmed"]
print(area_data)
# Using PyEcharts to generate HTML file of data display page
(
# Mapping of epidemic situation in China and configuring corresponding parameters
    Map()
        .add("", [list(z) for z in zip(list(area_data["prov"]), list(area_data["confirmed"]))], "china",
             is_map_symbol_show=False)
        .set_global_opts(title_opts=opts.TitleOpts(title="2020_nCoVMap of the total number of confirmed cases in various regions of China"),
                         visualmap_opts=opts.VisualMapOpts(is_piecewise=True,
                                                           pieces=[
                                                               {"min": 100000, "label": '>100000', "color": "#893448"},# maxNo upper limit
                                                               {"min": 50000, "max": 99999, "label": '50000-99999',"color": "#ff585e"},
                                                               {"min": 20000, "max": 49999, "label": '20000-49999',"color": "#fb8146"},
                                                               {"min": 10000, "max": 19999, "label": '10000-19999',"color": "#ffA500"},
                                                               {"min": 1000, "max": 9999, "label": '1000-9999',"color": "#ffb248"},
                                                               {"min": 0, "max": 999, "label": '0-999',"color": "#fff2d1"}]))
    ).render("Visualization of the number of confirmed epidemics nationwide.html")
```

4.3. Distribution map of national epidemic deaths

The rendering effect of the national epidemic death distribution map is shown in Figure 5. Different colors are used to reflect the current death toll values in different provinces. When the mouse hovers over different provinces, the current death toll of the province will be displayed. The final data visualization display file is in "national epidemic mortality visualization. Html". By clicking the corresponding browser, users can visually see the national epidemic situation and the distribution of epidemic mortality

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

in each province. According to the visual display, it can be seen that Hubei Province and neighboring Henan Province, where the outbreak initially occurred, have a large number of deaths, while other provinces have a small number of deaths. It can be seen that China's epidemic prevention and control is very effective.

Specific code display:

```
import pandas as pd
import numpy as np
from pyecharts.charts import Map
import pyecharts.options as opts
from pyecharts.globals import ChartType
from pyecharts.render import make_snapshot
# Read the file data and display it in the form of DataFrame
df = pd.read_csv(r'D:\pythonProject\ visualization \china-covid19(2020-12-06).csv')
# The number of confirmed cases was extracted from the total data, grouped by provinces
area_data = df.groupby("prov")["confirmed"].sum().reset_index()
#    Extract columns in area_data
area_data.columns = ["prov", "confirmed"]
print(area_data)
# Using PyEcharts to generate HTML file of data display page
(
# Mapping of epidemic situation in China and configuring corresponding parameters
    Map()
        .add("", [list(z) for z in zip(list(area_data["prov"]), list(area_data["confirmed"]))], "china",
             is_map_symbol_show=False)
        .set_global_opts(title_opts=opts.TitleOpts(title="2020_nCoVMap of the total number of confirmed cases in various regions of China"),
                         visualmap_opts=opts.VisualMapOpts(is_piecewise=True,
                                                           pieces=[
                                                               {"min": 100000, "label": '>100000', "color": "#893448"},# maxNo upper limit
                                                               {"min": 50000, "max": 99999, "label": '50000-99999',"color": "#ff585e"},
                                                               {"min": 20000, "max": 49999, "label": '20000-49999',"color": "#fb8146"},
                                                               {"min": 10000, "max": 19999, "label": '10000-19999',"color": "#ffA500"},
                                                               {"min": 1000, "max": 9999, "label": '1000-9999',"color": "#ffb248"},
                                                               {"min": 0, "max": 999, "label": '0-999',"color": "#fff2d1"}]))
    ).render("Visualization of the number of confirmed epidemics nationwide.html")
```

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

Figure 5: Distribution of deaths caused by epidemic in China.

4.4. City word cloud display of the number of confirmed epidemic cases nationwide

Through the statistics of the number of confirmed cases in each province, a word cloud statistics of the number of confirmed cases in each province is formed as Figure 6. From the word cloud statistics, it can be concluded that there are more confirmed cases in Hubei, Hong Kong and other provinces and cities across the country, and users need to consider the epidemic situation when traveling in provinces and cities.

Specific codes are as follows:

```
import pandas as pd
import numpy as np
from wordcloud import WordCloud, ImageColorGenerator# WordCloud
import matplotlib.pyplot as plt

# Read the file data and display it in the form of DataFrame
df = pd.read_csv(r'D:\pythonProject\ visualization\china-covid19(2020-12-06).csv')
# print(df)
# area_data = df.groupby('prov')['confirmed', 'recovered', 'deaths'].sum()
# Group according to provinces and extract the number of confirmed patients in the total data
area_data = df.groupby("prov")["confirmed"].sum().reset_index()
area_data.columns = ["prov", "confirmed"]
print(area_data)
area_data.to_csv('prov_confirmed.csv', index=False)

# Word cloud display
def draw_cloud(read_name):
# Font, background color is black
    wc = WordCloud(font_path="C://Windows//Fonts//STXINGKA.TTF",
                   background_color="black",
                   )
    kt= pd.read_csv(read_name, encoding='utf-8')
    name = list(fp.prov)
    value = kt.confirmed
    for i in range(len(name)):
        name[i] = str(name[i])
    dic = dict(zip(name, value))
    wc.generate_from_frequencies(dic)    # Generate word cloud based on word frequency
    plt.imshow(wc)
plt.axis("off")    # Axis off
plt.show()
    wc.to_file('prov_confirmed.png')    # Picture naming

if __name__ == '__main__':
    draw_cloud(r"D:\pythonProject\ visualization \prov_confirmed.csv")
```

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

Figure 6: Word cloud display of the number of confirmed cases in epidemic cities nationwide.

4.5. City word cloud display of national epidemic deaths

Through the statistics of the number of deaths in each province of the country, the word cloud statistics of the number of deaths in each province is formed as Figure 7. From the word cloud statistics, it can be concluded that Hubei, Hong Kong, Henan and other provinces in the country have more deaths.

Specific codes are as follows:

```
import pandas as pd
import numpy as np
from wordcloud import WordCloud, ImageColorGenerator# WordCloud
import matplotlib.pyplot as plt

# Read the file data and display it in the form of DataFrame
df = pd.read_csv(r'D:\pythonProject\ visualization \china-covid19(2020-12-06).csv')
# print(df)
# area_data = df.groupby('prov')['confirmed', 'recovered', 'deaths'].sum()
# Grouping by province, extract the number of deaths in the total data
area_data = df.groupby("prov")["deaths"].sum().reset_index()
area_data.columns = ["prov", "deaths"]
print(area_data)
area_data.to_csv('prov_deaths.csv', index=False)


def draw_cloud(read_name):
    # The font and background color are black
    wc = WordCloud(font_path="C://Windows//Fonts//STXINGKA.TTF",
                   background_color="black",
                   )
    kt= pd.read_csv(read_name, encoding='utf-8')
    name = list(fp.prov)
    value = kt.deaths
    for i in range(len(name)):
        name[i] = str(name[i])
    dic = dict(zip(name, value))
    wc.generate_from_frequencies(dic)    # Generate word cloud based on word frequency
    plt.imshow(wc)
plt.axis("off")    # Axis off
    plt.show()
    wc.to_file('prov_deaths.png')    # Picture naming


if __name__ == '__main__':
    draw_cloud(r"D:\pythonProject\ visualization \prov_deaths.csv")
```

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

Figure 7: City word cloud display of national epidemic deaths.

4.6. City word cloud display of the number of confirmed cases in Anhui Province

Through the statistics of the number of confirmed cases in each province of Anhui Province, a word cloud statistics of the number of confirmed cases in cities of Anhui Province is formed as Figure 8. From the word cloud statistics, it is concluded that there are more confirmed cases in Hefei and Bengbu in Anhui Province.

Specific codes are as follows:

```
import pandas as pd
import numpy as np
from wordcloud import WordCloud, ImageColorGenerator# WordCloud
import matplotlib.pyplot as plt

# Read the file data and display it in the form of DataFrame

df = pd.read_csv(r'D:\pythonProject\ visualization \china-covid19(2020-12-06).csv')
# Extract the number of confirmed cases in the corresponding provinces and cities
anhui_data = df.loc[df.prov == 'Anhui', ['city', 'confirmed']]
anhui_data.to_csv('anhui.csv', index=False)


def draw_cloud(read_name):
    # The font and background color are black
    wc = WordCloud(font_path="C://Windows//Fonts//STXINGKA.TTF",
                   background_color="black",
                   )
    ktt= pd.read_csv(read_name, encoding='utf-8')
    name = list(fp.city)    # Specify phrase City
    value = ktt.confirmed
    for i in range(len(name)):
        name[i] = str(name[i])
    dic = dict(zip(name, value))    # Dictionary storage City confirmed number word frequency
wc.generate_from_frequencies(dic)    # Generate word cloud based on word frequency
plt.imshow(wc)
plt.axis("off")    # Axis off
plt.show()
    wc.to_file('anhui_confirmed.png')    # Picture naming


if __name__ == '__main__':
    draw_cloud(r"D:\pythonProject\ visualization \anhui.csv")
```

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

Figure 8: City word cloud display of the number of confirmed cases in Anhui Province.

4.7. Pie chart and histogram display of the number of confirmed epidemic cases nationwide

Through the pie chart display (Figure 9) and histogram display of the national epidemic data (Figure 10), the comparison of the number of confirmed cases among provinces and cities is more intuitive. Among them, the number of confirmed cases in Hubei, Taiwan and Hong Kong accounts for a large proportion, which is significantly different from that in other cities.

Specific codes are as follows:

```
# Import library
import requests, json
from pyecharts.charts import Map, Page, Pie, Bar
from pyecharts import options as opts
from pyecharts.globals import ThemeType

# Crawling Chinese provincial and municipal data
def chinaTotal():
    re = requests.get(
"https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5&callback=jQuery341045890055561903065_1592206473904&_=1592206473905")
    data = str(re.text)[42:-1]
    data = json.loads(data)
    data = json.loads(data["data"])
    print(data["chinaTotal"])
    data = data["chinaTotal"]
    confirm = data["confirm"]
    heal = data["heal"]
    dead = data["dead"]
    nowConfirm = data["nowConfirm"]
    suspect = data["suspect"]
    nowSevere = data["nowSevere"]
    importedCase = data["importedCase"]
    noInfect = data["noInfect"]
    print(
        "confirm:" + str(confirm) + "\n"
                                    "heal:" + str(heal) + "\n"
                                                          "dead:" + str(dead) + "\n"
"nowConfirm:" + str(nowConfirm) + "\n"
                  suspect) + "\n"
                             "nowSevere:" + str(nowSevere) + "\n"
"importedCase:" + str(importedCase) + "\n"
```

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

```
                "noInfect:" + str(
                    noInfect) + "\n\n"
            )

    # Crawl diagnostic data
    def areatotal():
        global province_distribution
        re = requests.get(
    "https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5&callback=jQuery3410458900555619
03065_1592206473904&_=1592206473905")
        data = str(re.text)[42:-1]
        data = json.loads(data)
        data = data["data"]
        data = json.loads(data)
        data = data["areaTree"]
        data = data[0]
        data = data["children"]
        print(data)
        for i in data:
            temp = []
            areaname = str(i["name"])
            nowConfirm = str(i["total"]["nowConfirm"])
            confirm = str(i["total"]["confirm"])
            suspect = str(i["total"]["suspect"])
            dead = str(i["total"]["dead"])
            deadRate = str(i["total"]["deadRate"])
            heal = str(i["total"]["heal"])
            healRate = str(i["total"]["healRate"])
            temp.append(areaname)
            temp.append(confirm)
            kv.append(temp)
            province_distribution[areaname] = province_distribution.get(areaname, confirm)
            print(
                "areaname:" + str(areaname) + "\n"
                                                  "nowConfirm:" + str(nowConfirm) + "\n"

"confirm:" + str(confirm) + "\n"

"suspect:" + str(
                                suspect) + "\n"
                                    "dead:" + str(dead) + "\n"
                                                          "deadRate:" + str(deadRate) + "\n"

"heal:" + str(heal) + "\n"

"healRate:" + str(
                                healRate) + "\n\n"

            )


    def initMap():
    # Draw map image
        map = Map()
        map.set_global_opts(
           title_opts=opts.TitleOpts(title=" Epidemic distribution map of cities nationwide "),
            visualmap_opts=opts.VisualMapOpts(max_=3600, is_piecewise=True,
                                              pieces=[
                                                  {"max": 100000, "min": 10001, "label":
```

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

```
">10000", "color": "#680606"},
                                                {"max": 10000, "min": 5001, "label":
"5001-10000", "color": "#8A0808"},
                                                {"max": 5000, "min": 1001, "label":
"1001-5000", "color": "#B40404"},
                                                {"max": 1000, "min": 600, "label": "600-
1000", "color": "#DF0101"},
                                                {"max": 599, "min": 100, "label": "100-
599", "color": "#F78181"},
                                                {"max": 99, "min": 1, "label": "1-99",
"color": "#F5A9A9"},
                                                {"max": 0, "min": 0, "label": "0", "color":
"#FFFFFF"},
                                            ], )    # Define maximum and minimum ranges
    )
    # Draw pie chart
     pie = (
            Pie()

                .add("", kv, center=["50%", "80%"], radius=[30, 100])    # Add data
                .set_global_opts(title_opts=opts.TitleOpts(title=" National urban epidemic statistics
pie chart "),
                                 legend_opts=opts.LegendOpts(pos_left=160)) # Global settings
item
                .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {c}")))
    # Draw histogram
        bar = Bar(init_opts=opts.InitOpts(bg_color='rgba(255,250,205,0.2)',
                                          width='2000px',
                                          height='600px',
                                          page_title='page',
                                          theme=ThemeType.ESSOS
                                          ))
        bar.add_xaxis(xaxis_data=list(province_distribution.keys()))
        bar.add_yaxis("Total number of infections ", list(province_distribution.values()))
        bar.set_global_opts(title_opts=opts.TitleOpts(title=" main ", subtitle=" Vice- "))
        bar.set_series_opts(markpoint_opts=opts.MarkPointOpts(
            data=[opts.MarkPointItem(type_='max', name=' Maximum '),
opts.MarkPointItem(type_='min', name=' minimum value ')]))
        bar.render(r"testBar.html")
        map.add("Epidemic distribution map of cities nationwide ",
data_pair=province_distribution.items(), maptype="china", is_roam=True)
        page.add(map)
        page.add(pie)
        page.add(bar)
    if __name__ == '__main__':
        province_distribution = {}
        kv = []
        chinaTotal()
        areatotal()
        page = Page()
        initMap()
        print(province_distribution)
        page.render(' Comprehensive image display of epidemic situation in China.html')
```

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

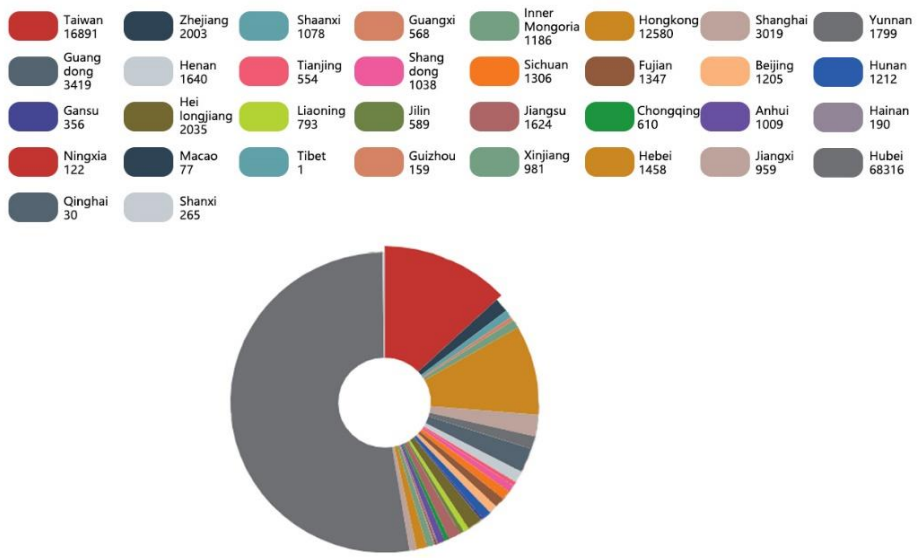

Figure 9: National epidemic data pie chart display.

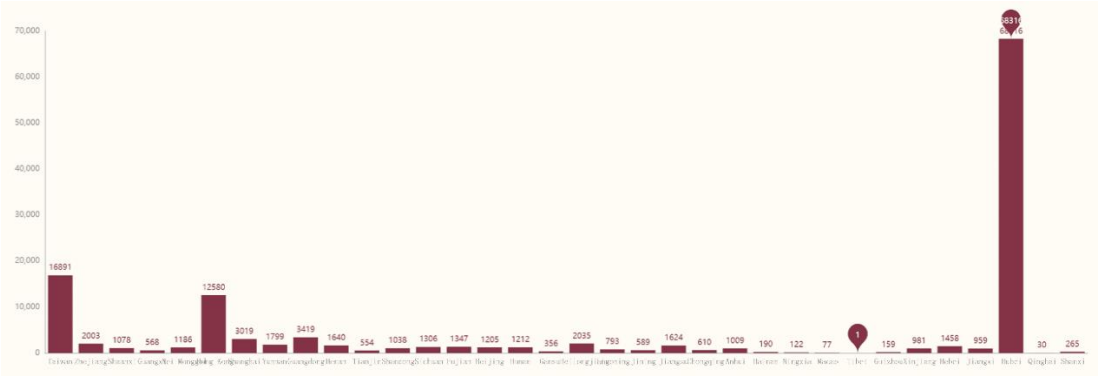

Figure 10: Histogram display of national epidemic data.

5. Conclusion and Outlook

5.1. Conclusion

In the era of rapid development of artificial intelligence and big data, the time speed of data generation is fast. In daily life, a large number of user browsing data will be generated all the time. There are many unused redundant data in different work fields. Data visualization technology can intuitively and clearly display a large amount of extracted and difficult to compare data by using relevant visualization tools and processing technologies, which is convenient for analysis Compare and process data, get the information and conclusions people want, help people find the laws of unknown things, and strengthen the role of data mining.

This paper uses Python data processing technology and PyEcharts visualization framework to crawl, analyze and visualize epidemic data. Through the multi angle and multi-faceted display of epidemic data, it is convenient for users to use data for decision-making in life or work.

5.2. Outlook

(1) This paper has not yet achieved mature system functions, and hopes to achieve an interactive design platform in the future.

(2) The analysis and visualization of single epidemic data is the first step of preliminary analysis and judgment of epidemic data. The analyzed data has not been given the value of application practice. It is hoped that the analyzed epidemic data can be applied to the field of daily life in the future, which can bring some reference for people's social life and travel or provide some data support for social policies.

ISSN 2616-5775 Vol. 5, Issue 6: 59-73, DOI: 10.25236/AJCIS.2022.050610

References

[1] Zhou, X. L. Zhang, X. L. Wang, F. Deng, H. Liang, B. Dai, W. Wei, S. L. Shi, C. M., Visualization Technology and Its Application for Massive Astronomical Data Analyses. In 2015 8TH INTERNATIONAL CONFERENCE ON INTELLIGENT NETWORKS AND INTELLIGENT SYSTEMS (ICINIS), 2015; pp 105-108.

[2] Levy, D. In Intelligent Process Management and Visualization Technologies, 12th International Conference on Business Process Management (BPM), Eindhoven, NETHERLANDS, Sep 07-11; Eindhoven, NETHERLANDS, 2014.

[3] Dong, B.; Liu, X. L.; Wang, H. R.; Ieee in RESEARCH ON THREE DIMENSION DATA MINING BASED ON VISUALIZATION TECHNOLOGY, International Conference on Machine Learning and Cybernetics, Baoding, PEOPLES R CHINA, Jul 12-15; Baoding, PEOPLES R CHINA, 2009; pp 292

[4] Ali, W. H.; Mirhi, M. H.; Gupta, A.; Kulkarni, C. S.; Foucart, C.; Doshi, M. M.; Subramani, D. N.; Mirabito, C.; Haley, P. J.; Lermusiaux, P. E. J.; Ieee in SeaVizKit: Interactive Maps for Ocean Visualization, MTS/IEEE Oceans Seattle Conference (Oceans Seattle), Seattle, WA, Oct 27-31; Seattle, WA, 2019.