VISUALIZATION OF OPEN DATA: A CASE STUDY OF CLIMATE DATA

Wan-Hsin Mao[1] and Jihn-Fa Jan[2]
[1]National Chengchi University, NO.64, Sec.2, ZhiNan Rd., Wenshan District, Taipei City 11605, Taiwan
Email: 103257005@nccu.edu.tw
[2]National Chengchi University, NO.64, Sec.2, ZhiNan Rd., Wenshan District, Taipei City 11605, Taiwan
Email: jfjan@nccu.edu.tw

KEY WORDS: data visualization, open source software, CKAN, Python

ABSTRACT: The development of technology has promoted large amount of data to transmit through the Internet rapidly. As a result, data has accumulated and become big data. Data visualization plays an important role in revealing the hidden information in such a tremendous amount of data. On the other hand, governments around the world are making efforts in "open data". Although a lot of data generated by government become accessible, the majority are still text files and tables such as txt, xls, csv, xml files. If the citizens want to have comprehensive understanding of those data, data visualization will be necessary. Presenting all the data in graphs and figures can maximize the value of open data and are favorable for further use. In recent years, governments around the world trend to use the Comprehensive Knowledge Archive Network (CKAN), which was developed by Open Knowledge Foundation (OKF), as the tool to build their open data platforms. Hence, this research attains the open data released by governments by using the API provided by CKAN, takes the rainfall data of the United Kingdom area, which was on the open data platform (DATA.GOV.UK) released by the Met Office of the United Kingdom government, as an example, and visualizes these data by collocating with programming in Python. Through the experiment, the feasibility of this open data visualization process has been proved. If different themes of research and depths of the data are adopted this process, the value of open data might be amplified.

1. INTRODUCTION

In the era of information explosion, to the users, no matter they are scientist themselves, funders, or related public, the most important thing is to discover the relations among the results of data analyses and queries. In face of the need of dealing with the complicated data and observing things happening around us, data visualization is one of the pivotal methods. However, with the rapid development of the technology, the cost that takes on attaining data has become much lower than before. It leads to the dramatical growth of the amount of data and makes the data visualization and analysis much more difficult. The appearance of new database techniques and web-based visualization methods may reduce the cost on data visualization (Fox and Hendler, 2011).

Open data has become the new trend in recent years. This approach has been pioneered in 2010 by governments in the United States and the United Kingdom (with the launch of two web portals – www.data.gov and www.data.gov.uk respectively) inspired in part by applications developed by grassroots civil society organizations (CSOs) on, for example, mapping the bicycle accidents. A variety of topics, such as geographic, economics, education, environment, and climate, of data are included on the portals. There are substantial social and economic gains to be made from opening government data to the public in an open format on the web, and can even improve services as well as create future economic growth (Hogge, 2010). As a result, in order to promote the transparency of government data, participation of the public and the efficiency of government, more and more governments around the world are defining and implementing strategies about open data(Huijboom and Van den Broek, 2011). However, open data and the concept of a data management system (DMS) are relatively new. Hence, in view of this situation, the OKF (Open Knowledge Fundation) developed its flagship project, CKAN, and hope that it can both help the growth of these respective initiatives as well as respond to them (Winn, 2013).

CKAN, which can record and manage open data, is a project developed by the OKF. It is an open source software and aims to be a lightweight open source piece of software for helping people to find and reuse open data (Hogge, 2010). As a result, there are more and more governments around the world using CKAN as the tool to develop their open data portals. For this reason, this research attains the open data released by governments by using the API provided by CKAN, takes the rainfall data of the United Kingdom area, which was on the open data platform (DATA.GOV.UK) released by the Met Office of the United Kingdom government, as an example, and visualizes these data by collocating with programming in Python. Finally, test the feasibility of this open data visualization process through these experiments.

2. RESEARCH TOOLS AND DATA

2.1 Research Tools

Under the considerations of cost, the difficulty level of programming and attaining data, this research first attains the rainfall data released by the Met Office of the United Kingdom government on the open data platform (DATA.GOV.UK) by using the API provided by CKAN. After having the data, we use Python and some open source packages to process the data and visualize them.

2.1.1 CKAN

CKAN is the acronym of Comprehensive Knowledge Archive Network. It is an open-source data management platform maintained by the Open Knowledge Foundation. Currently, it is used by around 50 out of 330 data catalogues worldwide, including the European Open Data portal, developed by the Belgian company Tenforce. CKAN furthermore allows catalogue federation through its APIs (Reiche et al., 2014). An introduction about the functions of CKAN is shown in table 1.

Table 1. Introduction about the functions of CKAN (Taiwan National Development Council, 2013)

Categories of the functions	Illustrations
Data management	• It is allowed to upload the complete data and edit the interface. • It is allowed to customize the forms and support multiple permission modes.
Back-end data management	• Each dataset can add title, URLs, license, sources and contact information etc. • Each dataset can include multiple files and sources and can give different categories and tags.
Search	• Full text search, tag filter and multi-criteria lookup are available. • It is able to filter the search results by geographical location, specifying a bounding box to limit the area where the user is interested in.
Community	• It allows site visitors to share pages on social networking sites such as Google+, Facebook, Twitter etc. • RSS/Atom feeds can be inserted into pages in order to learn the update of certain data.
Visualization	• One resource can have multiple views of the same data, for example, a grid and some graphs for tabular data. • For the data that contains spatial data, it can be viewed on a map.
Flexibility	• More than 140 of different extensions are available. • Since CKAN is an open source software, the developers can write their own extensions and the online tutorials and documents are sufficient.
Collecting data	• It can fetch and import records from many different repository sources including the geospatial Catalog Service for the Web (CSW) Servers, the existing web catalogues, simple HTML index pages or web accessible folders, other CKAN instances etc. • It is allowed to expose and consume metadata from other catalogs using RDF documents serialized using DCAT.
API	• The JSON API is provided and enables the reading, writing, and query functions for datasets and back-end data.
Management and analysis	• It has thorough authorization and access control of data and the editing privileges for each dataset can be set. • There are built-in statistics and analysis functions and can be integrated with Google Analytics.
Internationalization	• Its interface supports more than 40 languages. • Data can be present in multi-languages and can set certain language for each dataset.

2.1.2 Python

Python is an open source, interpreted, and high-level programming language. Because of the simple syntax and code readability, Python reduces the cost of program maintenance. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, allowing Python code execution on a wide variety of systems and making debugging programs easier. Python supports modules and packages, which encourages program modularity and code reuse (https://www.python.org/).

Hence, in the data visualization section of this research, Python is adopted as the programming language and its packages play important roles. The packages used in the study include, Requests, BeautifulSoup4, Numpy, Matplotlib and Basemap. Each package has different characteristics. A brief introduction about these packages is shown in table 2.

Table 2. The python packages used in this research

Package name	Characteristics of the package
Requests	Requests is an Apache2 Licensed HTTP library. It allows you to send HTTP/1.1 requests and access the response as well. (http://docs.python-requests.org/en/latest/)
BeautifulSoup4	BeautifulSoup is a Python library for pulling data out of HTML and XML files. BeautifulSoup3 is no longer being developed, so that BeautifulSoup4 is recommended. (http://www.crummy.com/software/BeautifulSoup/bs4/doc/)
NumPy	NumPy is for scientific computing with Python, mainly for dealing with N-dimensional array objects in this research. (http://www.numpy.org/)
Matplotlib	Matplotlib is a python plotting library which produces publication quality 2D graphics in a variety of hardcopy formats and interactive environments across platforms. There is a 'pylab' module which emulates matlab graphics. (http://matplotlib.org/)
Basemap (Matplotlib_toolkits)	The matplotlib basemap toolkit is a library for plotting 2D data on maps in Python. Basemap provides the facilities to transform coordinates to one of 25 different map projections and reading shapefiles. (http://matplotlib.org/basemap/index.html)
MPLD3	MPLD3 brings together Matplotlib, the popular Python-based graphing library, and D3js, the popular Javascript library for creating interactive data visualizations for the web. (http://mpld3.github.io/)

2.2 Experiment Data

2.2.1 The Rainfall Data of the United Kingdom Area

The rainfall data is attained from the open data portal of the United Kingdom (DATA.GOV.UK) and is released by the Met Office. The regions of the data are defined by the Met Office in two ways:

- The general regions (figure 1): including Scotland, England N, England S, Wales and Northern Ireland, 5 regions in total.
- The district regions (figure 1): including Scotland N, Scotland E, Scotland W, England E & NE, England NW/Wales N, Midlands, East Anglia, England SE/Central S, England SW/Wales S and Northern Ireland, 10 regions in total.

The content and format of the rainfall data used in this research is shown as figure 2 (take Scotland as an example).

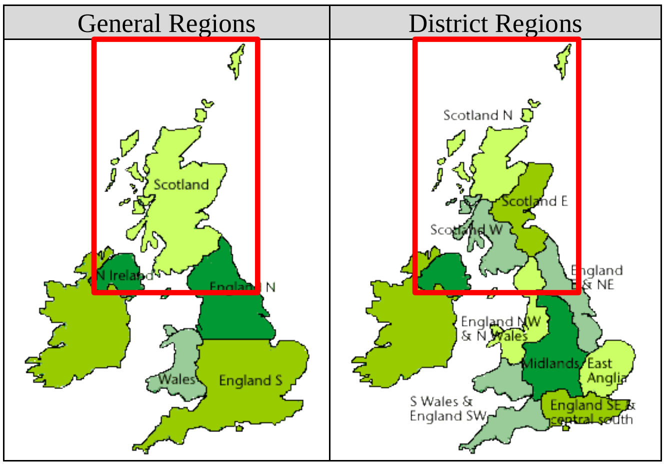
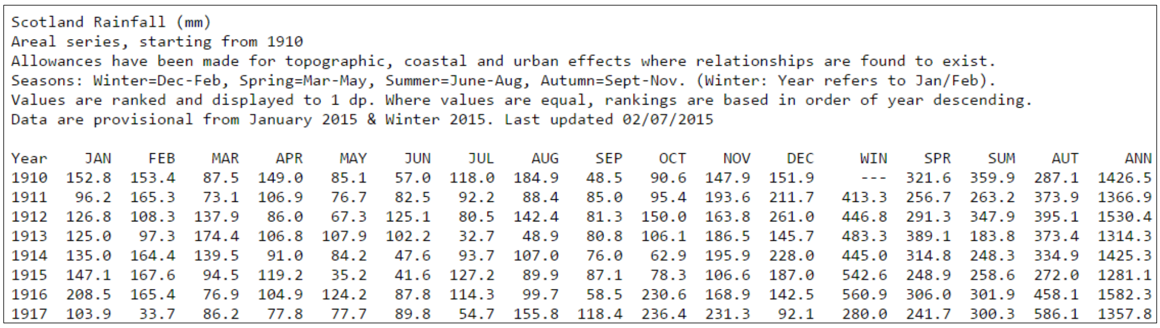


Figure 1. Regions defined by Met Office (Met Office, 2015)

Although there are two different regions defined by the Met Office, there are still some regions that are corresponded. Hence, the experiment data of this research are:

- The annual and monthly average rainfall data from 1910 to 2014 of the corresponding areas shown in the red box in figure 1: (1) Scotland (in the general regions) and (2) Scotland N、Scotland E、Scotland W (in the district regions)
- The annual rainfall data of 1910 and 2014 (since the rainfall data of 2015 is not completed yet) of the district regions

Using these data, we then visualize them through Python and compare and analyze the results.



```
Scotland Rainfall (mm)
Areal series, starting from 1910
Allowances have been made for topographic, coastal and urban effects where relationships are found to exist.
Seasons: Winter=Dec-Feb, Spring=Mar-May, Summer=June-Aug, Autumn=Sept-Nov. (Winter: Year refers to Jan/Feb).
Values are ranked and displayed to 1 dp. Where values are equal, rankings are based in order of year descending.
Data are provisional from January 2015 & Winter 2015. Last updated 02/07/2015

Year    JAN    FEB    MAR    APR    MAY    JUN    JUL    AUG    SEP    OCT    NOV    DEC     WIN    SPR    SUM    AUT    ANN
1910  152.8  153.4   87.5  149.0   85.1   57.0  118.0  184.9   48.5   90.6  147.9  151.9     ---  321.6  359.9  287.1 1426.5
1911   96.2  165.3   73.1  106.9   76.7   82.5   92.2   88.4   85.0   95.4  193.6  211.7   413.3  256.7  263.2  373.9 1366.9
1912  126.8  108.3  137.9   86.0   67.3  125.1   80.5  142.4   81.3  150.0  163.8  261.0   446.8  291.3  347.9  395.1 1530.4
1913  125.0   97.3  174.4  106.8  107.9  102.2   32.7   48.9   80.8  106.1  186.5  145.7   483.3  389.1  183.8  373.4 1314.3
1914  135.0  164.4  139.5   91.0   84.2   47.6   93.7  107.0   76.0   62.9  195.9  228.0   445.0  314.8  248.3  334.9 1425.3
1915  147.1  167.6   94.5  119.2   35.2   41.6  127.2   89.9   87.1   78.3  106.6  187.0   542.6  248.9  258.6  272.0 1281.1
1916  208.5  165.4   76.9  104.9  124.2   87.8  114.3   99.7   58.5  230.6  168.9  142.5   560.9  306.0  301.9  458.1 1582.3
1917  103.9   33.7   86.2   77.8   77.7   89.8   54.7  155.8  118.4  236.4  231.3   92.1   280.0  241.7  300.3  586.1 1357.8
```

Figure 2. The content and format of the rainfall data (take Scotland as an example)

2.2.2 The Shapefile of the United Kingdom Area

Because the regions defined by the Met Office are different from the administrative divisions, we cannot directly use Python Basemap to draw these areas. As a result, it is necessary to use the shapefile of the United Kingdom as an auxiliary. The shapefile has to be first processed by the GIS software, ArcGIS or QGIS for example, to ensure that regions are corresponded to which defined by the Met Office as figure 1.

Shapefile of the United Kingdom can be attained from Global Administrative Areas (GADM) as shown in figure 3. GADM has been developed by Robert Hijmans, in collaboration with colleagues at the University of California, Berkeley Museum of Vertebrate Zoology, the International Rice Research Instituteand the University of California, Davis, and with contributions of many others. Their goal is to map the administrative areas of all countries, at all levels. The downloading is free and there are three levels in the downloading of the United Kingdom. Level 2 is the most detailed one and level 0 is the outline of the United Kingdom. Not only the shapefile format, but also the geopackage (SpatialLite), R (SpatialPolygonDataFrame), ESRI file database, Google Earth .kmz and ESRI personal database are provided by GADM and the coordinate reference system is longitude/latitude and theWGS84 datum. The illustrations about the file format provided by GADM are shown in table 3.

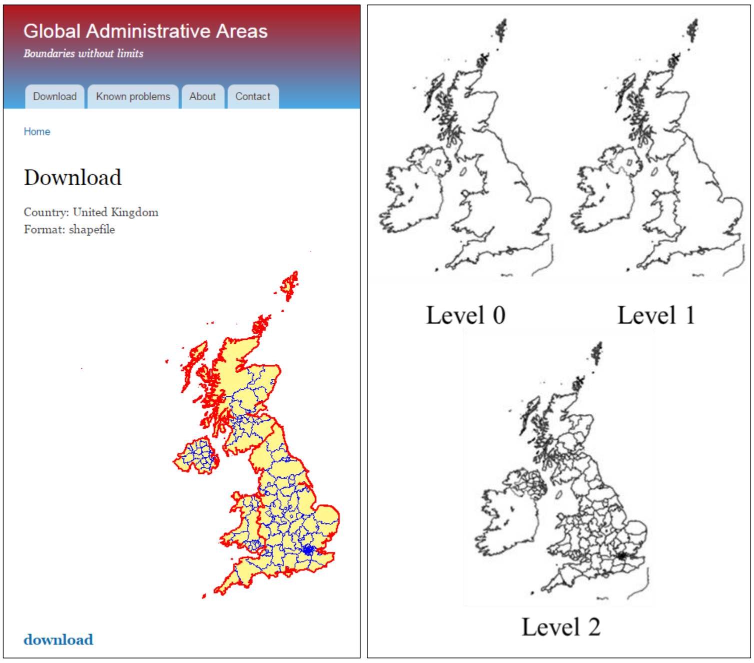


Figure 3. Downloaded shapefile of the United Kingdom for GADM

Table 3. File format provided by GADM (Hijmans, 2009)

File Format	Illustrations
Geopackage (SpatialLite)	It is a very good general spatial data file format (for vector data) and is based on the SpatiaLite format. It can be read by software using GDAL/OGR, including QGIS and ArcMap.
R (SpatialPolygonsDataFrame)	A .rds file can be used in R. Load the sp package using library(sp) first and then use readRDS("filename.rds").
ESRI file geodatabase	The standard format used by ArcGIS.
Google Earth .kmz	It is a format that can be opened in Google Earth.
ESRI personal geodatabase	It is a MS Access file that can be opened in ArcGIS. One of its advantages, compared to a shapefile, is that it can store non-latin characters. You can also query the (attribute) data in Access or via ODBC.
Shapefile	A shapefile consist of at least four actual files (.shp, .shx, .dbf, .prj) and is a commonly used format that can be directly used in Arc-anything, DIVA-GIS, and many other programs. However, many of the non-standard latin (Roman / English) characters are lost in the shapefile, so user should be careful while using it.

3. RESEARCH METHOD

The research method is based on the open data portal that is developed by CKAN. Shown in figure 4, there are two parts of the method in this research which are data fetching and data visualization:

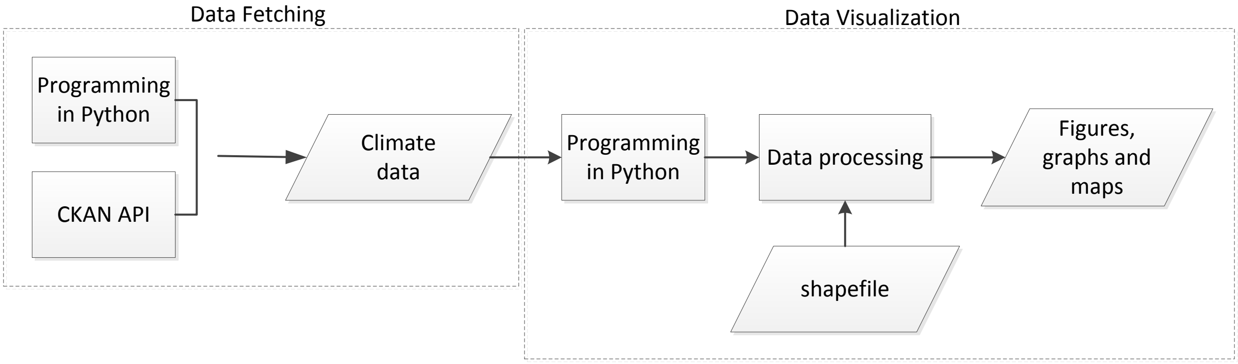


Figure 4. The flowchart of this research

The process will be explained in detail in the following two parts.

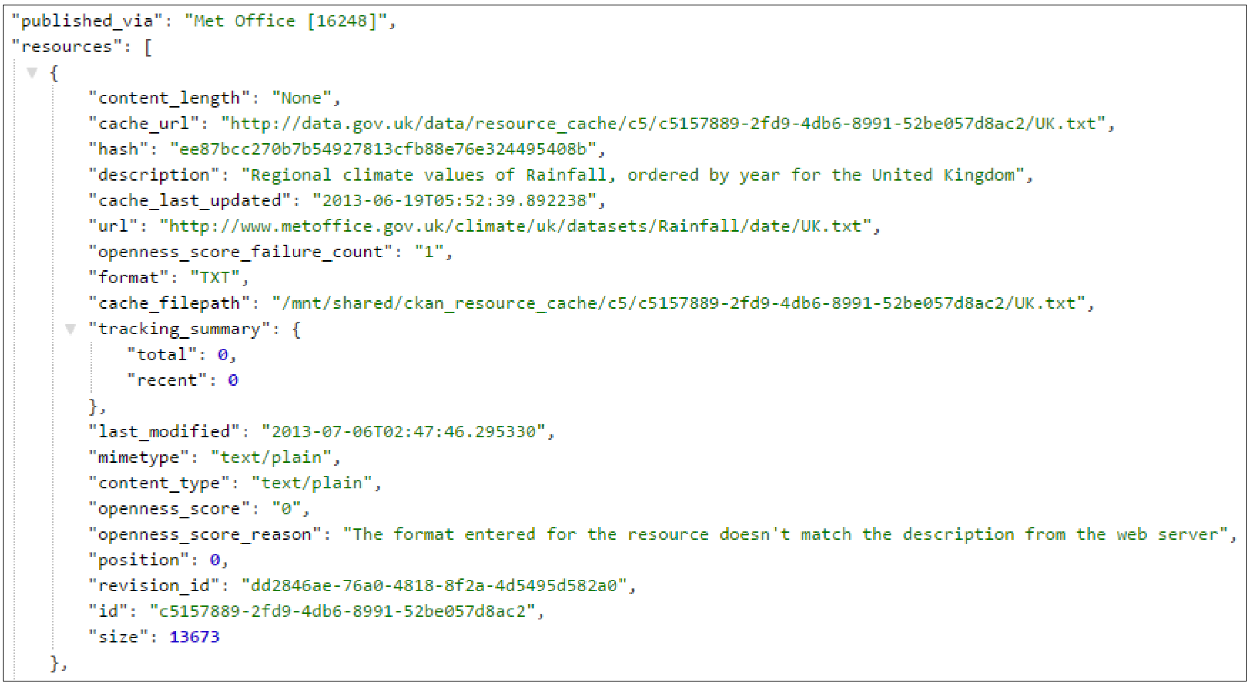
3.1 Data Fetching

By using Python and its Requests package collocated with CKAN API, we can send requests (figure 5) to the open data portal of the United Kingdom (DATA.GOV.UK):

```
import requests
r = requests.get('http://data.gov.uk/api/3/action/package_search?q=rainfall')
```

Figure 5. Sending a request to DATA.GOV.UK

After sending a request, what will be responded is a JSON-format list of all the data in the portal that contain the keyword “rainfall” in its metadata (figure 6). The content of the data can next be obtained by using the URLs in the metadata in Python. In this research, because the URL we got actually leads to a HTML download page of the Met Office, it is necessary to use the BeautifulSoup4 Python package to pull the actual URL of the data out of the page so that we can get the content of the data. In general, this step can be skipped if we can already get the content of the data through the URLs in the JSON-format list of metadata. After getting the URLs of the data, the content of the data can be obtained by programming in Python.



```
"published_via": "Met Office [16248]",
"resources": [
  {
      "content_length": "None",
      "cache_url": "http://data.gov.uk/data/resource_cache/c5/c5157889-2fd9-4db6-8991-52be057d8ac2/UK.txt",
      "hash": "ee87bcc270b7b54927813cfb88e76e324495408b",
      "description": "Regional climate values of Rainfall, ordered by year for the United Kingdom",
      "cache_last_updated": "2013-06-19T05:52:39.892238",
      "url": "http://www.metoffice.gov.uk/climate/uk/datasets/Rainfall/date/UK.txt",
      "openness_score_failure_count": "1",
      "format": "TXT",
      "cache_filepath": "/mnt/shared/ckan_resource_cache/c5/c5157889-2fd9-4db6-8991-52be057d8ac2/UK.txt",
      "tracking_summary": {
          "total": 0,
          "recent": 0
      },
      "last_modified": "2013-07-06T02:47:46.295330",
      "mimetype": "text/plain",
      "content_type": "text/plain",
      "openness_score": "0",
      "openness_score_reason": "The format entered for the resource doesn't match the description from the web server",
      "position": 0,
      "revision_id": "dd2846ae-76a0-4818-8f2a-4d5495d582a0",
      "id": "c5157889-2fd9-4db6-8991-52be057d8ac2",
      "size": 13673
  },
```

Figure 6. The JSON-format list of metadata

3.2 Data Visualization

There are two parts in the data visualization section. First, the content of the data need to be processed before using it to draw line chart. Second, the shapefile also need to be processed into the corresponding regions as defined by the Met Office.

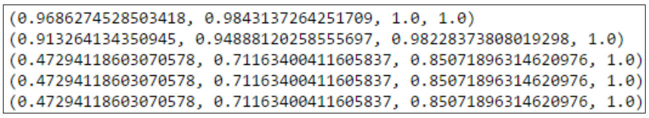
3.2.1 Data Processing and Charting

After getting the data by Python, what we are dealing with is the data content in a string type. Hence, it needs to be processed by programming in Python in order to split the string and turn it into float type so that it can be mathematically operated. After getting the monthly, seasonal average and annual rainfall data in float type, it is the Python NumPy package that we use to further process it into, for example, the average of the seasonal average rainfall from 1910 to 2014. All the values will be transferred into the format of array so that the Python Matplotlib package can draw as line charts.

3.2.2 Shapefile Processing and Mapping

Since the regions defined by the Met Office are not based on the administrative divisions and are different from the boundaries that drawn by Python Basemap, we cannot directly use Python Basemap to draw these areas. Hence, after getting the shapefile of the United Kingdom of level 2 from GADM, we use the ArcGIS software to do georeferencing. The shapefile needs to be aligned to the regional division figure defined by the Met Office (figure 1) so that we can know where the corresponding areas are and aggregate them by using the dissolve tool. After dissolving and adding the place name as attribute, the shapefile can be read by Python Basemap and overlap the map drawn in Basemap (figure 8) for the purpose of drawing the same regions as defined by the Met Office.

Based on the name of each region, we pick the maximum and minimum rainfall values within the selected data as the basis for normalizing the color into the interval [0, 1] (figure 7). For example, if comparing the change of annual rainfall in 1910 and 2014, the maximum and minimum rainfall within these two years will be selected out as the basis for calculating the corresponding color in the interval [0, 1]. After having the RGBA value of each region, we can draw all the polygons in the normalized color by Python Basemap and use MPLD3 to export it to HTML code.



```
(0.9686274528503418, 0.9843137264251709, 1.0, 1.0)
(0.913264134350945, 0.9488812025855697, 0.9822837380801929, 1.0)
(0.47294118603070578, 0.71163400411605837, 0.85071896314620976, 1.0)
(0.47294118603070578, 0.71163400411605837, 0.85071896314620976, 1.0)
(0.47294118603070578, 0.71163400411605837, 0.85071896314620976, 1.0)
```

Figure 7. An example of the normalized color for the colormap

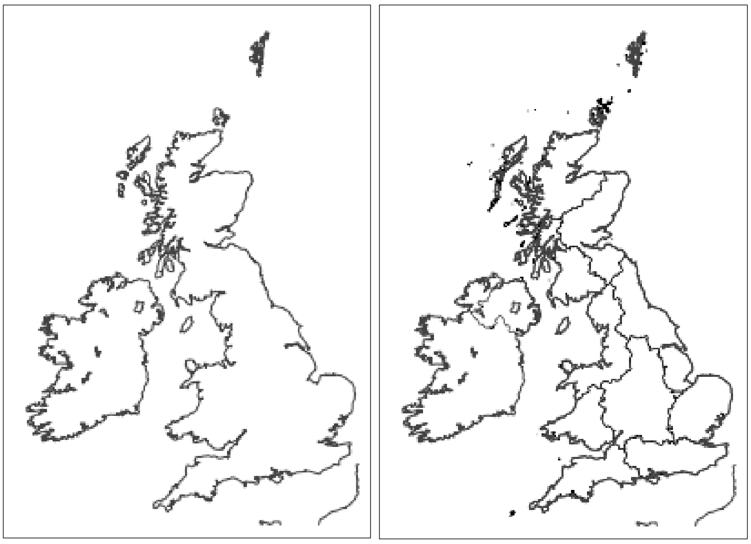


Figure 8. The map drawn by Python Basemap before (left) and after (right) the shapefile opverlapped

4. EXPERIMENT RESULTS AND DISSCUSION

The experiment can be divided into two parts as table 4: the Scotland regions and the district regions of the United Kingdom.

Table 4. Two parts of the experiment

Regions of the data	Detailed Regions	Adopted Years	Rainfall Data	Visualization Method
Scotland regions	(1) Scotland (2) Scotland N, Scotland E, Scotland W	From 1910 to 2014 (a total of 105 years)	Annual and monthly average rainfall data	Line charts
District regions of UK	10 district regions	1910 and 2014 (a total of 2 years)	Annual rainfall data	Maps , scatter plots

4.1 Scotland Regions

Figure 9 shows the change of annual rainfall in Scotland regions from 1910 to 2014. From figure 9 (left) we can see that, in Scotland, the annual rainfall value changes in a range of approximately 1100 to 1900 mm and has a trend of slightly increasing during these 105 years. If we compare the annual rainfall of 1910 with 2014, the annual rainfall in 2014 is approximately 300 mm more than that in 1910. In addition, in the aspect of the north, east and west of Scotland as shown in figure 9 (right), the rainfall mainly distributed in the north (Scotland_N) and west (Scotland_W) of Scotland. The rainfall in the east of Scotland is about 300 mm to 400 mm less than the north and west regions. The rainfall in the north and west of Scotland is similar. If the rainfall data can also be analyzed together with other data such as the terrain, wind direction or ocean current data, then a comprehensive analysis of the rainfall distribution which was mentioned previously can be done.

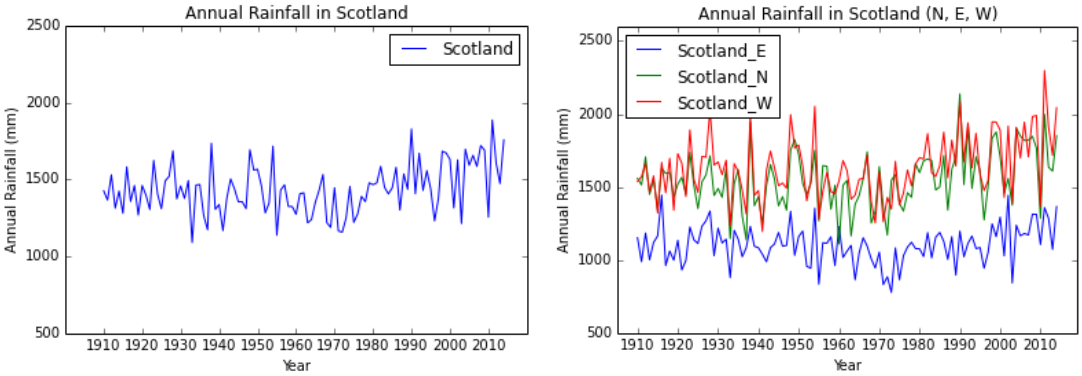


Figure 9. The change of annual rainfall in Scotland regions from 1910 to 2014

Figure 10 shows the change of the average of monthly average rainfall in Scotland regions from 1910 to 2014. From figure 10 (left) we can see that, in Scotland, the minimum of rainfall happens in July and the peak period is from October to January. Taking March to May as spring, June to August as summer, September to November as autumn and December to February as winter, the rainfall mainly distributes in winter and less in summer. In addition, in the aspect of the north, east and west of Scotland as shown in figure 10 (right), it also has the trend as figure 9 that the rainfall distributed in the east part of Scotland is the least and the rainfall in north and west part is similar. The rain mostly falls in winter as well.

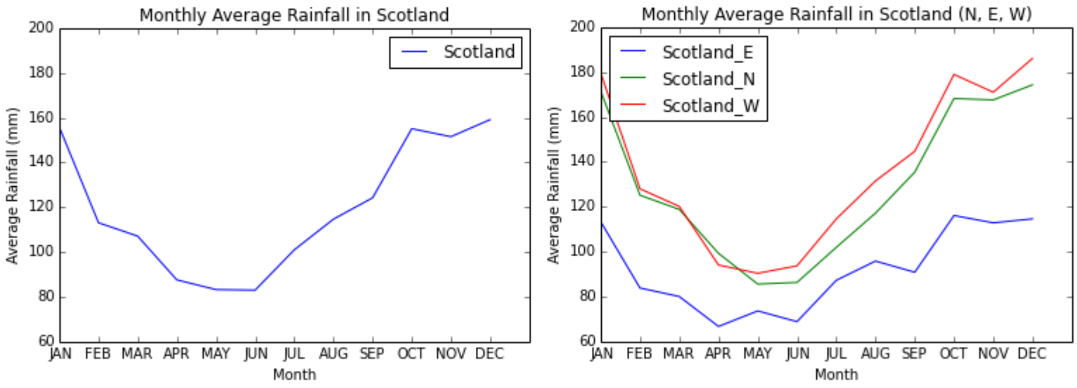


Figure 10. The change of the average of monthly average rainfall in Scotland regions from 1910 to 2014

4.2 District Regions of the United Kingdom

Figure 11 shows the annual rainfall in each district region in 1910 and 2014 and the difference between these two years (using the rainfall values of 2014 to minus the 1910 ones). In 1910, the rain falls mainly in west part of the Great Britain and the northwest part has the highest rainfall value. Also, in 2014, the rainfall distribution has a similar trend with 1910, only the rainfall that falls in the northwest part of the Great Britain (Scotland regions) is more than 1910. The differences of the rainfall of other areas between 1910 and 2014 may not be obvious if we only look at (a) and (b) of figure 11. Hence, by using the values of 2014 to minus the 1910 ones, (c) of figure 11 is made.

From (c) of figure 11, we can not only see that the greatest change is in the Scotland regions but also find out that the Northern Ireland and the southeast of England (England SE/Central S) have certain changes in the annual rainfall between 1910 and 2014. If (c) of figure 11 is not drawn, the changes in these two places might be overlooked.

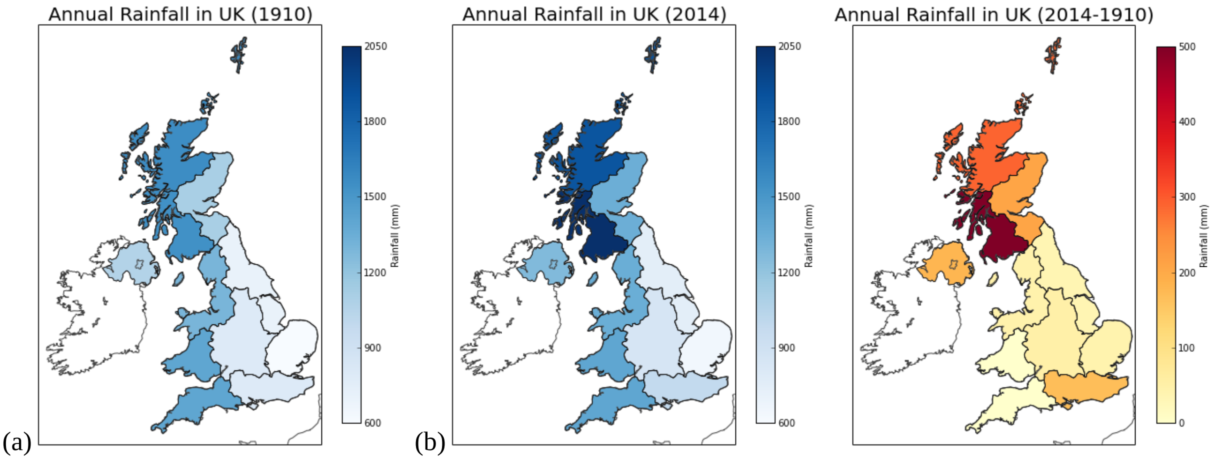


Figure 11. (a) The annual rainfall in each district region in 1910. (b) The annual rainfall in each district region in 2014. (c) The difference of the annual rainfall between 1910 and 2014.

Figure 12 also shows the same information as figure 11 does. What is different is that figure 12 is the result of exporting the map and plot drawn by Python Matplotlib to HTML code by using MPLD3. By doing this, the maps and plots are no longer static. They become interactive and enable the users to drag and zoom in or out depending on their needs. Moreover, when the cursor hovers over the plot or map, the corresponding point or polygon labels will show. This can be another way to provide more detailed information to the public.

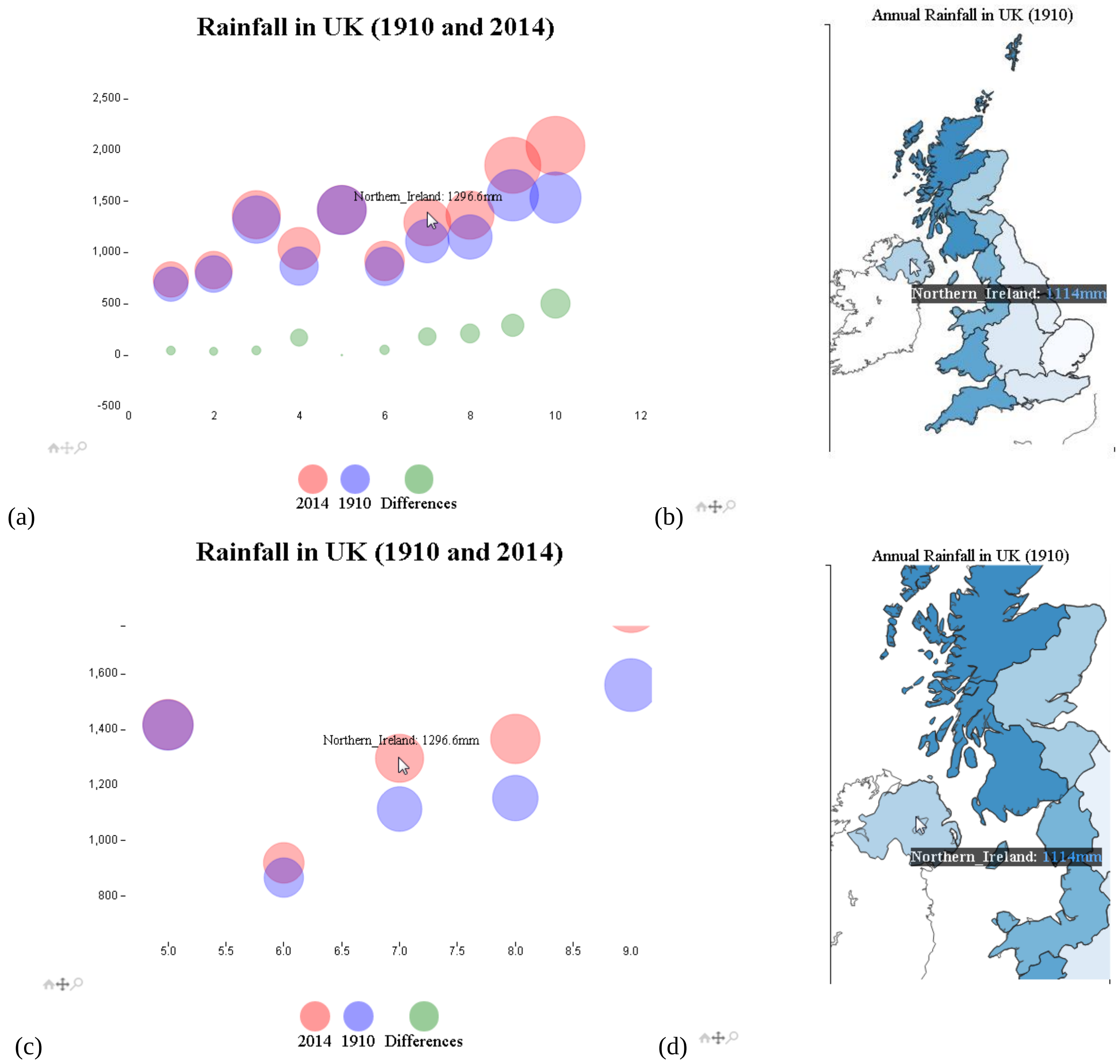


Figure 12. (a)(c) The scatter plot of the annual rainfall of 1910 and 2014 as a web-based interactive plot, using the same data as figure 11. (b)(d) The map of the annual rainfall in each district region in 1910 as a web-based interactive plot, using the same data as (a) of figure 11.

When the data is still a text file like figure 2, it is hard for users to directly understand or absorb the information hidden in the data. Through the experiment results mentioned above, it shows that after transferring the text data into line charts, scatter plots and maps, the way to learn the information becomes versatile and easier. Different information can be learned from different types of figures. For example, it will be easier to find out the change through the time if the data is turned into a line chart and the trend of the distribution if the data is turned into a map. As a result, data visualization is an important method in helping users understand the messages concealed in the text data.

5. CONCLUSION AND SUGGESTION

Through the experiment, the feasibility of this open data visualization process, which refers to the combination of CKAN API and Python, has been proved. In accordance with this process, users do not need to consider about the storage of the data. Moreover, if the hardware is powerful enough, it is possible to visualize the real time data. These text data can be directly learned and if it contains any mistake, it will be much easier to correct by visualizing it. Hence, the quality of open data will be improved.

After visualization, according to the different identities of readers, discussion of different topics and different depths of the data, a variety of aspects of research can be extended and might promote the reuse of open data and the value-added applications. One thing to mention is that if this open data visualization process can work together with tools that provided APIs like CKAN does, it will be easier in fetching data. Finally, the data used in this research is relatively simple, if it is analyzed together with other data such as the terrain, wind direction or ocean current data, then a more comprehensive analysis can be done and the value of this open data might be amplified.

REFERENCES

Council, T.N.D., 2013. Report of Open Government Data Value-added Application, available for download at http://www.ndc.gov.tw/News_Content.aspx?n=33B27A62D5D08C83&sms=90CFF1BDB7A1494C&s=CFC266A607127C45

Fox, P. and Hendler, J., 2011. Changing the equation on scientific data visualization. Science(Washington), 331(6018): 705-708.

Hijmans, R., 2009. Global Administrative Areas. Retrieved July 16, 2015, from http://www.gadm.org/.

Hogge, B., 2010. Open data study. a report commissioned by the Transparency and Accountability Initiative, available for download at: http://www. soros. org/initiatives/information/focus/communication/articles_publications/publications/open-data-study-20100519.

Huijboom, N. and Van den Broek, T., 2011. Open data: an international comparison of strategies. European journal of ePractice, 12(1): 4-16.

Reiche, K.J., Höfig, E. and Schieferdecker, I., 2014. Assessment and Visualization of Metadata Quality for Open Government Data, Conference for E-Democracy and Open Governement, pp. 335.

Winn, J., 2013. Open data and the academy: An evaluation of CKAN for research data management.