Data Modelling and Visualisation of UK Government Open Data Based on the State of Business in the Post-COVID-19 Era

Yue Li(✉)

School of Electronics and Computer Science, University of Southampton, University Road, Southampton, UK
yl5a21@soton.ac.uk

Abstract. The emergence of the big data age and the movement toward open data have been driven by the Internet's quick development. UK is the country with the highest degree of government data openness in the world. With the development of open data, the UK government's open data has been in a leading position in terms of execution, influence and perfection. The sheer volume of data, both national and corporate, is undoubtedly a challenge. Therefore, in order to improve the value of data, it is necessary to ensure the authenticity, accuracy and good management of data, which has become a problem to be solved. The original data used in this paper is about the impact of COVID-19 on businesses, obtained from the UK government website. This paper deals with and analyses data from three aspects: data cleaning, data modelling and data visualisation.

Keywords: The UK Government's Open Data · Data Cleaning · Data Modelling · Data Visualisation

1 Introduction

With the advent of the Big Data era, the sheer volume of data makes people need to strictly consider the authenticity and accuracy of data when using data. Most developed countries are highly open to government data and have very open data management policies that enable ordinary people to access real government data. When these data are obtained, they need to be processed. The first step is data cleaning. Data cleaning is the process of repairing or deleting incorrect, corrupt, incorrectly formatted, duplicate or incomplete data from a data set [5]. After cleaning up the data, the next step that must be taken is data modelling, which is the process of analysing data objects and their relationships with other objects. This step is used to analyse the data requirements needed for business processes and create a data model for the data stored in the database. Finally, the data is visualised and presented in different charts to make the information contained in the data clearer and easier to understand.

The indicators and analysis in the public data sheet come from the survey response to the new voluntary fortnightly business survey during the period of COVID-19, which collected responses about the impact on their turnover, labour prices, trade and business

Z. Zhan et al. (Eds.): EIMSS 2022, AHCS 7, pp. 998–1016, 2023.
https://doi.org/10.2991/978-94-6463-024-4_104

resilience from businesses during the two-week reference period in the United Kingdom [4]. By processing the above data, we can get a very clear picture of the impact of COVID-19 on many British enterprises.

2 Methods

The data processing part will be divided into the following three steps.

Step 1: Data cleaning.
Step 2: Data modelling.
Step 3: Data Visualisation.

2.1 Data Cleaning

The tool used to clean the data in this article is Google Open Refine. After downloading the dataset from the UK government website, we scanned and checked the dataset and found that there were a lot of dirty data. Because of the huge amount of data, making changes manually will be a difficult task, and it is not always possible to ensure that the data is processed cleanly. Open Refine can help process many datasets quickly and is compatible with formats such as CSV, Excel, and XML. In addition, it can easily correct the data format, standardise the data representation, that is, unify the data format, delete duplicates, edit numbers to ensure accuracy, and so on. Therefore, in this article, open refine is selected to clean up the dataset.

The following are the errors in the dataset and data cleaning process.

Error 1: As shown in Fig. 1, there is an extra single quote at the beginning of the sentence. This error occurs in the #government schemes form. The error was reported when executing a query using SPARQL.

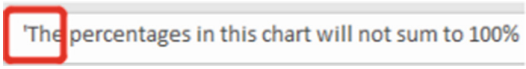


Fig. 1. A redundant single quote. (Photo credit: Original)

Error 2_1: As shown in Fig. 2, one number is floating-point, while the other numbers are integer. The steps of operation in Google Open Refine were to split the value into different columns with the separator ".", and then delete the new column composed of decimal parts.

Fig. 2. The floating-point number. (Photo credit: Original)

Figure 3 indicates the cleaned data.

145	149	294
415	303	718
108	29	137
10442	7181	17623

Fig. 3. The cleaned data. (Photo credit: Original)

Next, the table above is exported in CSV format.

CSV Lint is a tool for testing the validity of CSV files. As shown in Fig. 4, CSV Lint shows that the CSV file is valid.

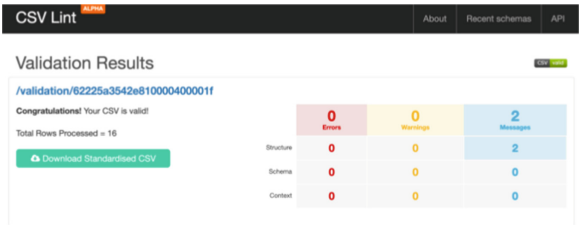


Fig. 4. CSV valid result. (Photo credit: Original)

Error 2_2: Similar to error 2_1, the same method is used to clean up data. After CSV Lint validation, the validation is passed, and the CSV file is valid (Figs. 5 and 6).

Workforce Size 0 - 99	Workforce Size 100 - 249	Workforce Size 250 +	Total
1326	9116	7181	17623.23

Fig. 5. Another floating-point number. (Photo credit: Original)

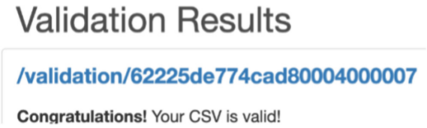


Fig. 6. CSV valid result. (Photo credit: Original)

Error 3–7: Duplicate columns in the dataset (error 3), negative numbers that should not appear (error 4), inconsistent precision (error 5), numbers not displayed as percentages (error 6), percentages with numbers greater than 1 (error 7).

As shown in Figs. 7 and 8, a # response rate table item is created.

This form has the same errors as the previous form, for example, the accuracy of numbers is different.

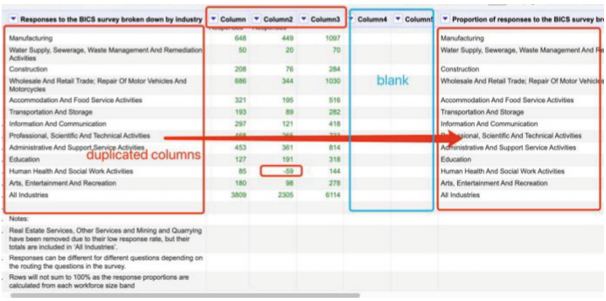


Fig. 7. Sheet errors_1. (Photo credit: Original)

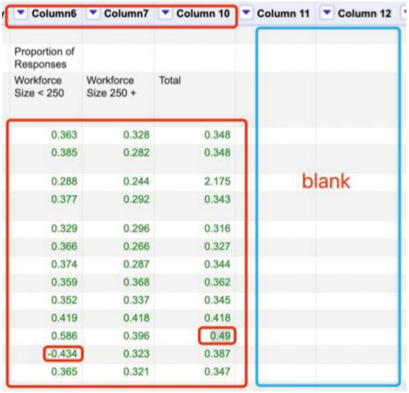


Fig. 8. Sheet errors_2. (Photo credit: Original)

"Numeric facet" is used to filter negative numbers and correct them (Figs. 9 and 10).

Fig. 9. Negative numbers_1. (Photo credit: Original)

Fig. 10. Negative numbers_2.

The number displayed as a percentage is required to be no more than 1. Therefore, the calculated value of the total response percentage is 27.5%. In order to maintain consistency in later formats, it is written as 0.275 (Fig. 11).

Fig. 11. Uniform format. (Photo credit: Original)

This step requires changing the decimal point format to the percentage format.

A function called round in Python can be used to specify the precision. After using it, the numeric facet function is simply needed to filter for numbers greater than 100. Finally, the value + "%" statement is used to append the percent sign to the number (Figs. 12, 13, 14 and 15).

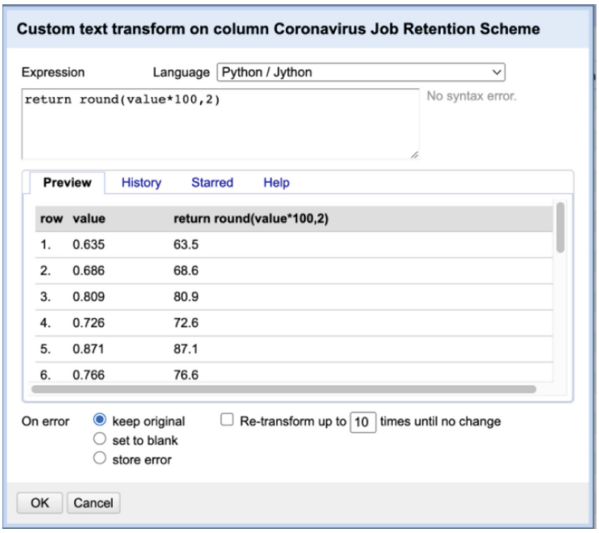


Fig. 12. The round function. (Photo credit: Original)

Workforce Size < 250 Proportion of Responses	Workforce Size 250 + Proportion of Responses	Total Proportion of Responses
36.3%	32.8%	34.8%
38.5%	28.2%	34.8%
28.8%	24.4%	27.5%
37.7%	29.2%	34.3%
32.9%	29.6%	31.6%
36.6%	26.6%	32.7%
37.4%	28.7%	34.4%
35.9%	36.8%	36.2%
35.2%	33.7%	34.5%
41.9%	41.8%	41.8%
58.6%	39.6%	4.9%
43.4%	32.3%	38.7%
36.5%	32.1%	34.7%

Fig. 13. The data has been cleaned. [1] (Photo credit: Original)

Workforce Size < 250 Number of Responses	Workforce Size 250 + Number of Responses	Total Number of Responses
648	449	1097
50	20	70
208	76	284
686	344	1030
321	195	516
193	89	282
297	121	418
468	265	733
453	361	814
127	191	318
85	59	144
180	98	278
3809	2305	6114

Fig. 14. The data has been cleaned. [2] (Photo credit: Original)

Fig. 15. The valid result. (Photo credit: Original)

Error 8: As shown in Figs. 16 and 17 below, less than 46.1-% and more than 42.2+% of the numbers are incorrectly formatted. Numeric facet is used to filter the non-numeric data, and then fix them. Then the two statements mentioned above are used to clean the data (Fig. 18).

Coronavirus Job Retention Scheme	Business rates holiday	Deferring VAT payments	HMRC Time To Pay scheme	Government-funded small business grant or loan schemes	Accredited finance agreements	We haven't received any that we applied for
0.221	0.186	0.407	0.168	0.085	0.023	0.205
0.224	0.2	46.1-%	0.168	0.046	0.011	0.163

Fig. 16. Wrong format_1. (Photo credit: Original)

HMRC Time To Pay scheme	Government-funded small business grant or loan schemes	Accredited finance agreements	None of the above
0.092	0.028	0.066	0.481
0.143	0.029	0.071	0.457
0.096	0.05	0.11	42.2+%

Fig. 17. Wrong format_2. (Photo credit: Original)

Fig. 18. The valid result. (Photo credit: Original)

Error 9: As shown in Fig. 19, there are two different symbols (a comma and a semicolon) that split the statement. Therefore, the formatting needs to be normalised. When querying with SPARQL, the error was found in the same way as error 1 by reporting the error here.

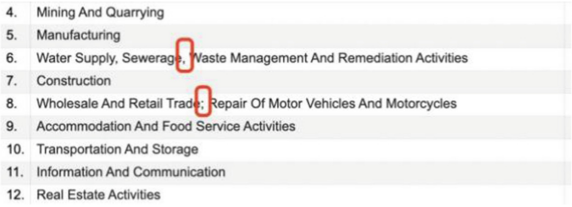


Fig. 19. The different formats of the split. (Photo credit: Original)

Error 10: As shown in Fig. 20, there is no space between the words and the sample. The process of finding errors was to put the data into Microsoft Word, and the automatic spell check function displays this error.

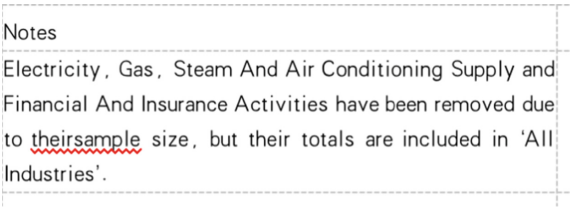


Fig. 20. Words split error. (Photo credit: Original)

2.2 Data Modelling

2.2.1 Knowledge Representation of the RDF Model

The basic data model consists of resources, properties and statements. As shown in Fig. 21, they are referred toas subjects, predicates, and objects [1].

RDF defines a simple model that describes the relationships between resources through specified attributes and corresponding values. The underlying structure of the Resource Description Framework (RDF) is a triple, also known as an utterance, that is, subject-predicate-object. It provides a unified standard for describing entities/resources.

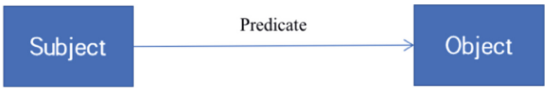


Fig. 21. RDF model. (Photo credit: Original)

The significance of RDF is that it can describe resources according to resources, which can easily capture non-XML document information. RDF itself does not prescribe semantics, but provides a framework for the semantic structure of each meta-data system [3].

Due to its simplicity and ease of writing [2], turtle (Terse RDF Triple Language) has become a genera standard for serialised RDF data models. It is more readable than N-Triples and more compact than RDF/XML.

2.2.2 Method of Data Ingestion

A project based on Spring Boot and Apache POI was created in IntelliJ IDEA using Java. Figure 22 shows the file structure of the project. The main code is in the files PoiUtil.java and PoiServiceImpl.java.

In PoiUtil.java, readExcel (para1, para2,…) function is used to read the specified excel file. In PoiServiceImpl.java, the foreach() function is used to print the information read in the excel file (Figs. 23, 24 and 25).

After running the project, as shown in Fig. 26, a browser was opened and the URL was entered to go to the user interface. Then, the specified file was selected and the "Execute" button was clicked. The .ttl file is then ready to be generated.

Figure 27 shows how the output format of the prefix was printed out.

As shown in Figs. 28 and 29, during the operation, it was found that the RDF word list called scovo was no longer available, so a new word list called qb was found for this data modelling. And yueli is a word list created by itself, as explained in the .tll file that generates the schema.

Fig. 22. The file structure of the project. (Photo credit: Original)

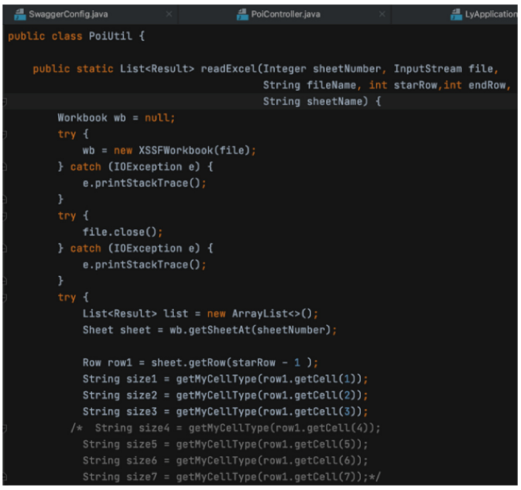


Fig. 23. Read excel_1. (Photo credit: Original)

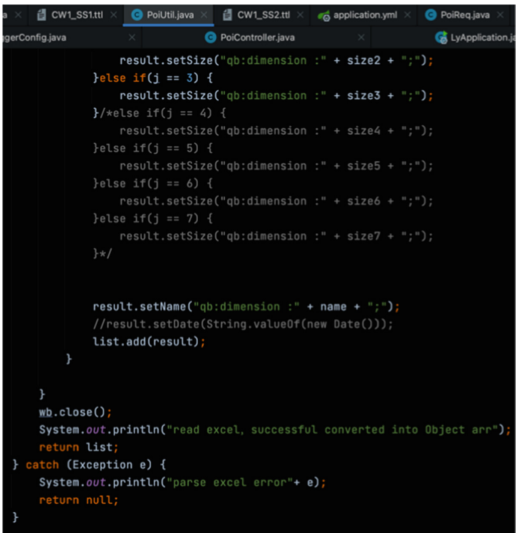


Fig. 24. Read excel_2. (Photo credit: Original)

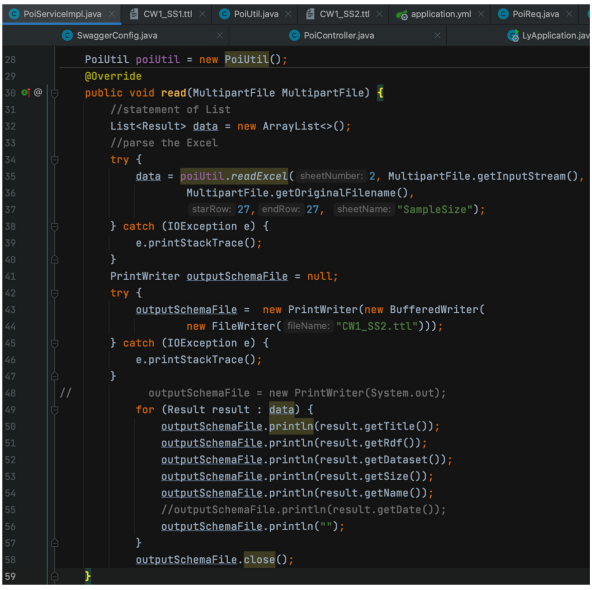


Fig. 25. Print information. (Photo credit: Original)

Fig. 26. The user interface. (Photo credit: Original)

```
outputSchemaFile.println("@prefix : <http://yueli.psi.entaking.org/data/> .");
outputSchemaFile.println("@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .");
outputSchemaFile.println("@prefix dc: <http://purl.org/dc/elements/1.1/> .");
outputSchemaFile.println("@prefix owl: <http://www.w3.org/2002/07/owl#> .");
outputSchemaFile.println("@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .");
outputSchemaFile.println("@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .");
outputSchemaFile.println("@prefix qb: <http://purl.org/linked-data/cube#> .");
outputSchemaFile.println("@prefix yueli: <http://entaking.org/schema/yueli/> .");
```

Fig. 27. How the prefix is printed out. (Photo credit: Original)

```
sources root, ~/Downloads/comp6214/testfile01/java-rdf-example-code
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix qb: <http://purl.org/linked-data/cube#> .



:InduType rdf:type owl:Class ;
      rdfs:subClassOf qb:Dimension .

:WorkforceSize rdf:type owl:Class ;
     rdfs:subClassOf qb:Dimension .
```

Fig. 28. The prefix_1. (Photo credit: Original)

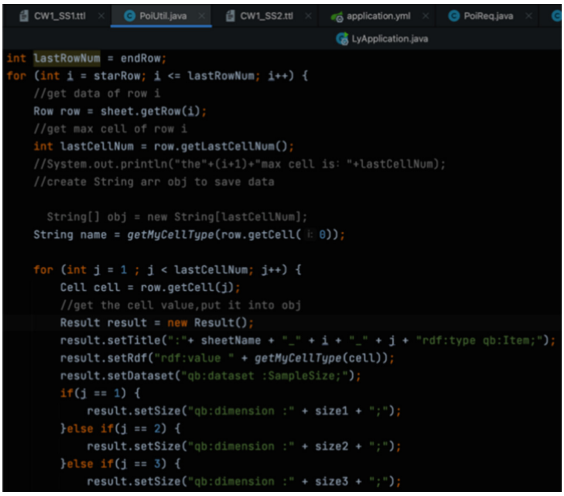


Fig. 29. PoiUtil.java. (Photo credit: Original)

2.2.3 Ontology with Instance Frequency Statistics and TTL Snippet

The ontology was drawn as shown in Fig. 30.

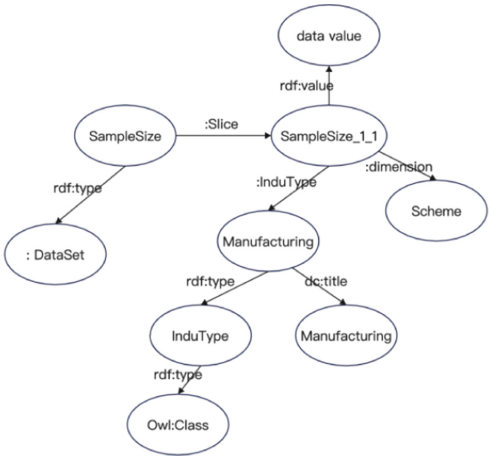
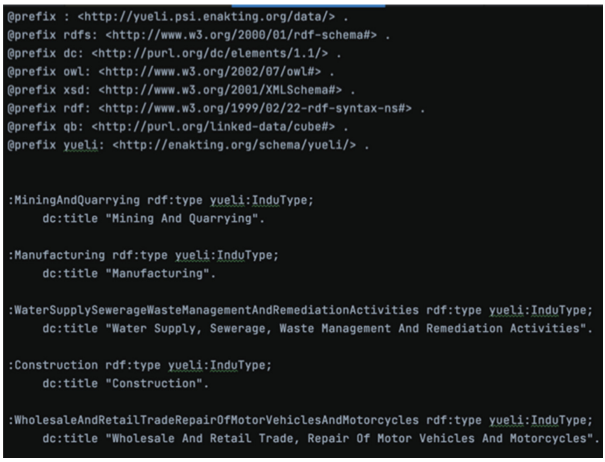


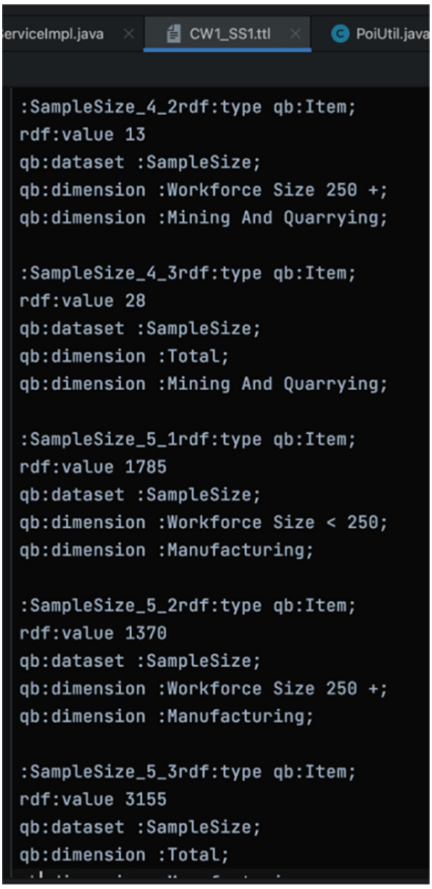
Fig. 30. The ontology. (Photo credit: Original)

Figures 31, 32 and 33 show the complete output of this rdf model (.ttl).



```
@prefix : <http://yueli.psi.enakting.org/data/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix qb: <http://purl.org/linked-data/cube#> .
@prefix yueli: <http://enakting.org/schema/yueli/> .


:MiningAndQuarrying rdf:type yueli:InduType;
     dc:title "Mining And Quarrying".

:Manufacturing rdf:type yueli:InduType;
     dc:title "Manufacturing".

:WaterSupplySewerageWasteManagementAndRemediationActivities rdf:type yueli:InduType;
     dc:title "Water Supply, Sewerage, Waste Management And Remediation Activities".

:Construction rdf:type yueli:InduType;
     dc:title "Construction".

:WholesaleAndRetailTradeRepairOfMotorVehiclesAndMotorcycles rdf:type yueli:InduType;
     dc:title "Wholesale And Retail Trade, Repair Of Motor Vehicles And Motorcycles".
```

Fig. 31. The output_1. (Photo credit: Original)



```
erviceImpl.java   CW1_SS1.ttl   PoiUtil.java

:SampleSize_4_2rdf:type qb:Item;
rdf:value 13
qb:dataset :SampleSize;
qb:dimension :Workforce Size 250 +;
qb:dimension :Mining And Quarrying;

:SampleSize_4_3rdf:type qb:Item;
rdf:value 28
qb:dataset :SampleSize;
qb:dimension :Total;
qb:dimension :Mining And Quarrying;

:SampleSize_5_1rdf:type qb:Item;
rdf:value 1785
qb:dataset :SampleSize;
qb:dimension :Workforce Size < 250;
qb:dimension :Manufacturing;

:SampleSize_5_2rdf:type qb:Item;
rdf:value 1370
qb:dataset :SampleSize;
qb:dimension :Workforce Size 250 +;
qb:dimension :Manufacturing;

:SampleSize_5_3rdf:type qb:Item;
rdf:value 3155
qb:dataset :SampleSize;
qb:dimension :Total;
```

Fig. 32. The output_2. (Photo credit: Original)

```
#Dataset
:SampleSize rdf:type qb:Dataset;
     dc:title “Survey send by gov”;
     qb:Slice :SampleSize_1_1;
     qb:Slice :SampleSize_1_2;
     qb:Slice :SampleSize_1_3;
     qb:Slice :SampleSize_2_1;
     qb:Slice :SampleSize_2_2;
     qb:Slice :SampleSize_2_3;
     qb:Slice :SampleSize_3_1;
     qb:Slice :SampleSize_3_2;
     qb:Slice :SampleSize_3_3;
     qb:Slice :SampleSize_4_1;
     qb:Slice :SampleSize_4_2;
     qb:Slice :SampleSize_4_3;
     qb:Slice :SampleSize_5_1;
     qb:Slice :SampleSize_5_2;
     qb:Slice :SampleSize_5_3;
     qb:Slice :SampleSize_6_1;
     qb:Slice :SampleSize_6_2;
     qb:Slice :SampleSize_6_3;
```

Fig. 33. The output_3. (Photo credit: Original)

2.3 Data Visualisation

2.3.1 Approach to Multi-dimensional Interactive Visualisation

ECharts is an open-source interactive graphics and visual JavaScript library for visualising the data in this article. ECharts was chosen because it runs smoothly on PCs and mobile devices, and has a variety of chart types, such as line series, bar series, tree series, pie series, relational chart series, map series, custom series and so on.

Take #samplesize as an example to visualise the data, as shown in Figs. 34 and 35. The bar chart is used to visualise this sheet. The horizontal coordinates are the name of each industry, which is replaced by the letters A, B, C etc. The vertical coordinates are the values of the different workforce sizes.

As shown in Fig. 36, because the line chart can more clearly reflect the volume change, it was selected to represent the change of the volume in all industries with labour force scales of 0–99, 100–249, 250 and above.

Pie charts are a simple, clear and convincing method to explain complex statistics. Therefore, for #tradingstatus, the pie chart is used to show the proportion of businesses in different parts of the UK that have chosen to continue operations, permanently close and temporarily close. The code is in Fig. 37, and the pie chart of the results is shown in Sect. 2.3.2.

```
option = {

  title: [
  {
    text: 'Responses to the BICS survey broken down by industry'
  }
 ],


 tooltip: {
   trigger: 'axis',
   axisPointer: {
     type: 'shadow'
   }
 },
 legend: {
   data: ['Workforce Size < 250', 'Workforce Size 250+', 'Total']
 },
 toolbox: {
   show: true,
   orient: 'vertical',
   left: 'right',
   top: 'center',
   feature: {
     mark: { show: true },
     dataView: { show: true, readOnly: false },
     magicType: { show: true, type: ['line', 'bar', 'stack'] },
     restore: { show: true },
     saveAsImage: { show: true }
   }
 },
 xAxis: [
   {
     type: 'category',
     axisTick: { show: false },
```

Fig. 34. The visualisation_code. (Photo credit: Original)

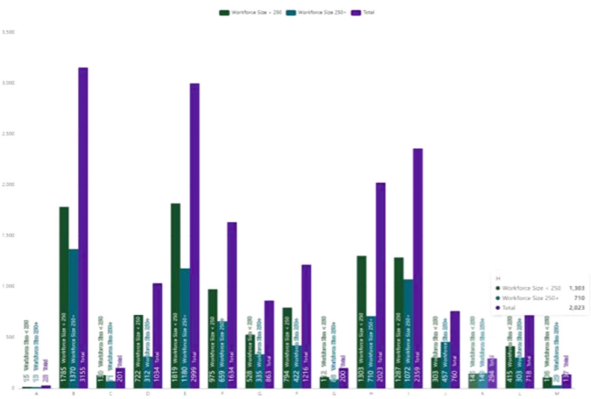


Fig. 35. The visualisation_result. (Photo credit: Original)

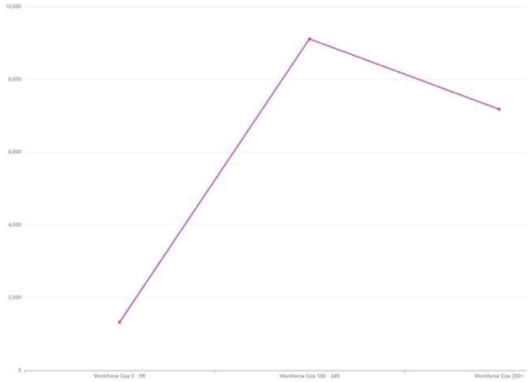


Fig. 36. The line chart. (Photo credit: Original)

```
option = {
  title: {
    text: 'the current trading status of the enterprise',
    subtext: 'Fake Data',
    left: 'center'
  },
  tooltip: {
    trigger: 'item',
    formatter: '{a} <br/>{b} : {c} ({d}%)'
  },
  legend: {
    left: 'center',
    top: 'bottom',
    data: [
      'Continuing to trade',
        'Has permanently ceased trading **',
        'Has temporarily closed or temporarily paused trading',

    ]
  },
  toolbox: {
    show: true,
    feature: {
      mark: { show: true },
      dataView: { show: true, readOnly: false },
      restore: { show: true },
      saveAsImage: { show: true }
    }
  },
  series: [
    {
      name: 'England',
      type: 'pie',
      radius: [20, 140],
      center: ['25%', '25%'],
```

Fig. 37. The pie chart codes. (Photo credit: Original)

2.3.2 Hypothetical Scenarios with Walkthrough for Multidimensional Interactive Visualisation

It is assumed that in a live demonstration, one can click on a column in a list to display the data of each list under the workforce scale. As shown in Fig. 38, this makes the presentation of data more intuitive and clearer and makes it easier and more intelligent for people to view data.

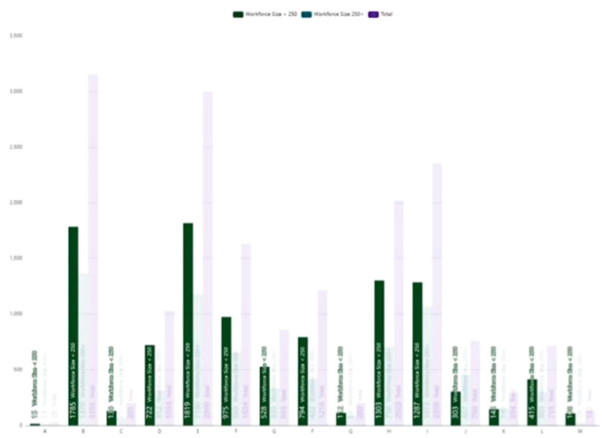


Fig. 38. The visualisation_1. (Photo credit: Original)

As shown in Fig. 39, there are three colours at the top of the table bar chart, and each colour represents a different meaning. When one of the colours is clicked on, the colour box turns grey and the values in the colour bar are not displayed.

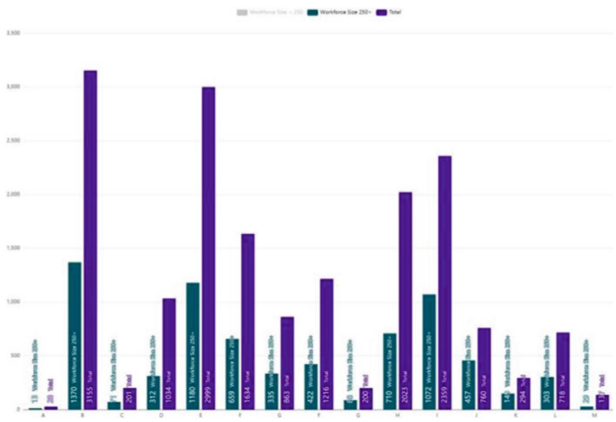


Fig. 39. The visualisation_2. (Photo credit: Original)

As shown in Fig. 40, for #tradingstatus, a pie chart was chosen to display the proportion of businesses in different regions of the UK that chose to continue, close permanently and close temporarily, because using a pie chart makes it more intuitive to see the difference in proportions.

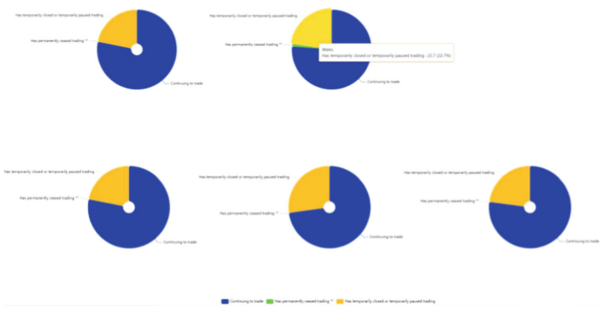


Fig. 40. The visualisation_3. (Photo credit: Original)

3 Results

By cleaning the open data, the dirty data is cleaned up, so as to achieve a more standard data format. The data was then successfully modelled using the RDF model. Finally, data visualisation was successfully achieved by using ECharts, which makes the information transmitted by the data more intuitive.

4 Conclusion and Future Considerations

There are many challenges in the data processing. Because data sets are displayed in different formats and have different openings, it is easy to confuse data that does not need to be cleaned up with data that does need to be cleaned up in the data cleaning process. This requires a read-through of the obvious dirty data before clearing the data. In addition, during data modelling, it is important to make sure that the word list of the selected RDF is available before proceeding. This problem can be solved by creating your own word list.

The work done in this paper has achieved the main research objectives and succeeds in modelling and visualising the data. In summary, using open data is the first and most important step of data research. As the cornerstone of data analysis, it is very important in the process of data analysis.

References

1. Agathangelos G, Troullinou G, Kondylakis H, Stefanidis K, Plexousakis D (2018). RDF query answering using apache spark: Review and assessment. In: 2018 IEEE 34th International Conference on Data Engineering Workshops (ICDEW). IEEE
2. Alam F, Khan MA, ur Rahman S, Khusro S, Ali S (2015) Resource description framework and topic maps: complementary or competitive? PAK ACAD SCI. 1
3. Blomqvist E (2014) The use of Semantic Web technologies for decision support–a survey. Semantic Web 5(3):177–201
4. Business impact of coronavirus, analysis over time, UK: Waves 2 to 5 panel - Office for National Statistics. Ons.gov.uk. (2022). Accessed 2 July 2022. https://www.ons.gov.uk/releases/analysisovertimebusinessimpactofcoronaviruswaves2to5uk
5. Hosseinzadeh M et al (2021) Data cleansing mechanisms and approaches for big data analytics: a systematic study. J Ambient Intell Humanized Comput. https://doi.org/10.1007/s12652-021-03590-2