COMP6214: Open Data Innovation

Coursework#1 Documentation

Shanchuan Wu

sw9n14@soton.ac.uk

Project Overview

This project is a web application for COMP6214: Open Data Innovation coursework#1. The website is developed using Bootstrap and Font Awesome. Two data visualisations on this webpage are designed using DC.js and DataTables. The website is hosted in my ECS personal homepage:

http://users.ecs.soton.ac.uk/sw9n14/ODI_CW1/.

Also, I upload all files to my Github homepage:

https://github.com/358203708/Open_Data_Innovation_CW1

Data Cleaning

1 Mixed Date Formats

Mixed date formats are one of the most common errors in datasets. In this case, date formats of most cells are *dd/mm/yyyy (i.e. 7/12/2012)*, while those of other cells are *yyyy-dd-mm (i.e. 2012-12-01)*. In order to fix this kind of errors, find the columns and select *Edit cells -> Common transforms -> To date* to transform all dates into the same format: *yyyy-mm-ddThh:mm:ssZ*".

2 Blank Records

Sometimes there are some blank records in datasets. In order to remove these blank records, firstly create a text facet for *Unique Investment Identifier* column, then find the blank rows and select *All -> Edit Rows -> Remove all matching rows* to remove them.

3 Duplicate Records

To eliminate duplicate records, it is necessary to bring them together at first. This is most easily done by sorting them on unique value, such as *project ID*. First of all, select the triangle of *project ID* column and sort these values as numbers. Then, click *Sort -> Reorder Row Permanently* to ensure that identical rows now are adjacent to each other. Next, click the triangle of *project ID* and select *Edit Cells ->Blank Down*. Then create a facet on blank cells and select duplicate records by clicking on true. Finally, use *Remove all matching rows* to remove these duplicate records. Apart from this method, there exists another one. Select *Facet -> Customized facets -> Duplicates facet* to find duplicate records and remove them.

4 Multiple Representations

Many abbreviations existing in datasets can result in ambiguity. In this dataset, for instance, *DOA* stands for *Department of Agriculture*. This problem can be solved manually using *Edit*. Besides, some whitespaces where there should not be can be a problem. But two methods are provided by OpenRefine. The first method is to trim whitespaces separately using *Edit*. Another one is to select *Edit cells ->Common transforms -> Trim leading and trailing whitespace*. Moreover, cluster function is a good choice to clean the dataset, which can help to correct typos and minor spelling differences.

5 Redundant Commas

Putting commas in numbers helps a human, but to a computer these commas are redundant. To get rid of the commas, create a facet by number and select *Non-numeric* values at first. Then select *Edit cells -> Transforms* and delete commas using Expression:*value.replace(",", "")*. Finally, transform these new cell into numbers.

6 Mixed Use of Numerical Scales

Using numeric facet to *Lifecycle Cost* column is a good method to illustrate the distribution of these values, but the distribution is so large and it may lead to audit failure. In this case, expression: *value.log()* can be used to make the distribution more even. First of all, create a facet by number. Then, use expression: *value.replace("($m)", "")* to delete *($m)*. At last, normalize the values or distribute the values evenly using expression:*value.log()*.

7 Spelling Errors

Spelling errors are one of the most common errors in datasets. While this kind of error is not critical in all cases, it can lead to awkwardness when querying and visualising data. This kind of error can be fixed manually by using *Edit* function.

8 Redundant Data

Sometimes, not all data will be used for data manipulation and visualisation. So, datasets will contain some redundant data within them. In this dataset, for instance, *Project Description* can be redundant data, thus it is common that errors are made when entering it. In order to get rid of this column, click the triangle of *Project Description* and select *Edit column*. Finally, use *Remove this column* to delete them.

9 Summation Records

When datasets are extracted from a spreadsheet application, it is common that datasets contain total records, which make it inconvenient to process the data. In this case, they should be removed. In order to remove them, apply a text facet to the *Unique Investment Identifier* column and select all *Total* records and star them. Next, select *Facet by star* and use *Edit Rows -> Removing all matching rows* to remove them.

Data Manipulation

1 Add DurationYear Column

Start Date and *Completion Date* are given in this dataset, so duration year can be obtained according to the differences between these two columns. So firstly, click the triangle of *Completion Date*, then select *Edit column -> Add column based on this column* to add new column. The final step is to use expression: *diff(value,cells["Start Date"].value, "years")* to get values of *Duration Year* for each project.

2 Add Project Status Column

Likewise, the status of each project can be obtained by using expresion: *if (cells["Scedule Variance (in days)"].value < 0, "delay", if cells["Schedule Variance (in days)"].value > 0, "advance", "onTime"))*. So, if the value of *Schedule Variance* is less than 0, equals 0 or more than 0, the status of project is *delay*, *onTime* or *advance* respectively.

3 Re-order Columns

In order to make the dataset more readable and organized, columns can be re-ordered by selecting *Edit columns -> Re-order/remove columns*.

Data Visualisation

Visualisation#1

Visualisation#1 consists of three bar charts and one donut chart. These charts can provide a multi-faceted filter to the list of projects, shown in the table below. Three bar charts illustrate the number of projects in different *years*, *months* and *duration years* (ps: *0* means less than 1 year) respectively. By contrast, the donut chart shows the status *(advance, onTime and delay)* and the number of projects.

Dragging the brushes on bar charts or selecting the status on the donut chart will filter the projects displayed in the table, as well as the display of the other graphs. The table displays a list of all projects that match the conditions selected in the graphs. Besides, the records in the table can be searched by entering keywords and sorted in ascending or descending order.

Visualisation#1 is designed for readers who want to know the number of projects in different *years*, *months*, *duration years* and types of *status*.

Visualisation#2

In visualisation#2, there is one row chart, three bar charts, and one composite (grouped) chart. The row chart lists all agencies/departments according to the numbers of their projects in

descending order. Three bar charts show the sum of *Lifecycle Cost*, *Planned Cost* and *Projected/Actual Cost* of selected agencies/departments in different years respectively. Likewise, the composite (grouped) chart illustrates the sums of three kinds of costs in different years, but this chart can help to compare them at ease.

Similarly, the row chart and three bar charts can be used to filter the projects displayed in the table. Moreover, the row chart can show the number of different agencies/departments' projects. When the mouse hovers over one of three legends in the composite (grouped) chart, the bars of another two kinds of costs will become transparent, which helps to compare the costs in different years. Apart from detailed information about filtered projects, the table can provide readers with searching and sorting function.

Visualisation#2 can help audiences to know the total number of selected agencies/departments' projects and compare the costs of them.