FIT5192 – Data Wrangling – S2 2016
Assignment 2

Data Cleansing

Submitted
By
Ramprasath Karunakaran
(26994437)
on
02-09-2016

1. INTRODUCTION

The following report discusses the data cleansing task performed using python language for the given dataset in data.csv file. The python program discussed in this report will imports the dataset given, explores the given dataset using graphical and statistical methods, perform cleaning task using different techniques and finally load the cleaned data to a CSV file. It also discusses various problems encountered while performing data cleansing.

Programming Environment: Python 2.7.12 and Jupyter Notebook Anaconda version 2
Python libraries used: pandas, re, datetime, seaborn, matplotlib
Input Data file: data.csv
Output Data file: 26994437_data_cleaned.csv

2. Task 1 – Data Auditing

Before loading the dataset in Jupyter Notebook, the CSV file was opened in Microsoft Excel to have a general idea. The dataset was about disasters and their happening details. Things noted in the dataset at that point were as follows.

- Columns Start and End are the dates of the disasters but they are poorly formatted
- Some dates have only month and year and some have year alone.
- Location column does not have proper format and is not consistent. (Example: 'au large de' is French for 'Off the coast of')
- There are many missing values in the numeric part of the dataset

After the dataset was loaded into a dataframe using pandas, basic operations were done over the dataframe to bring it to a format suitable for cleaning.

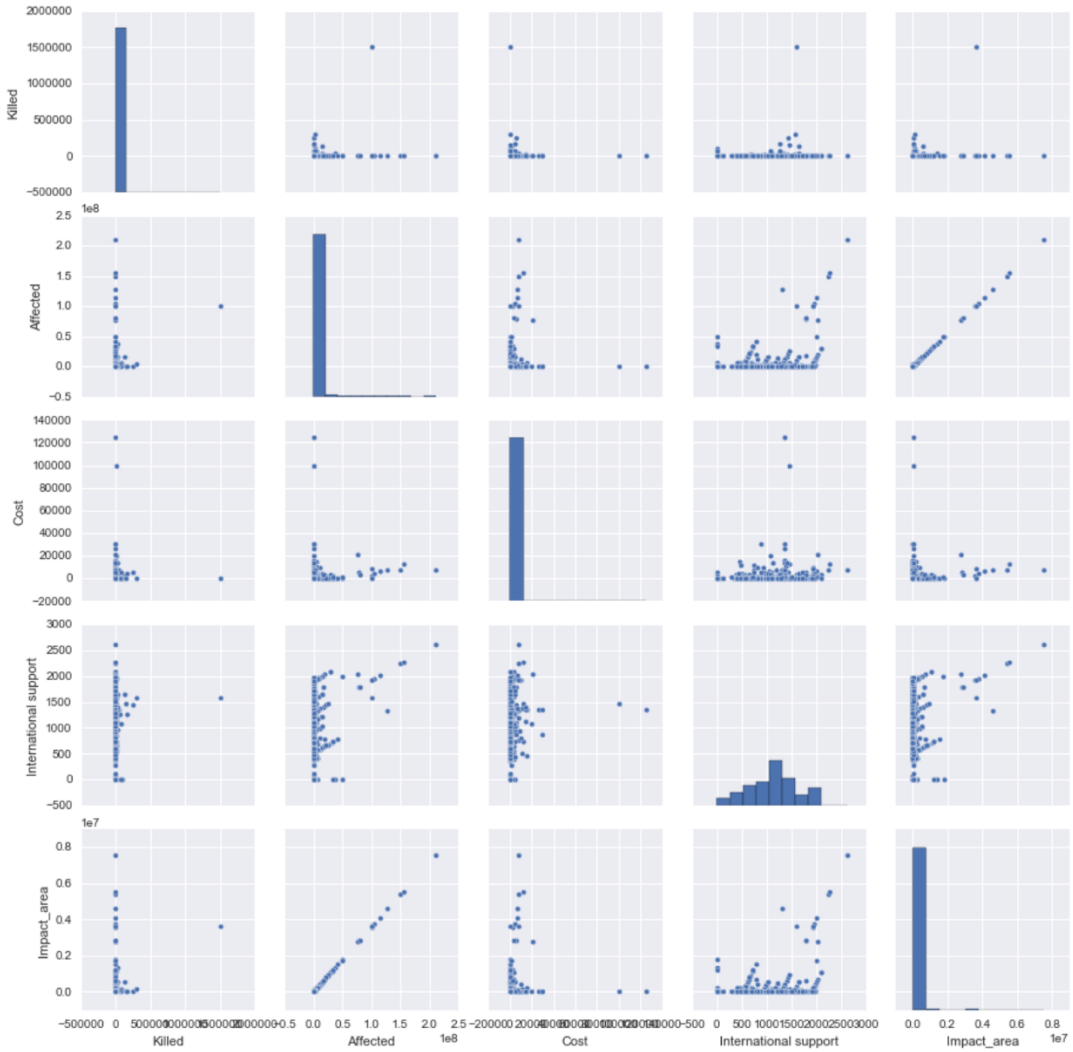
- Additional Index column was dropped
- First row was made headers for the dataframe and the entire dataframe was re-indexed.
- Numeric columns (Killed, Affected, International Support and Affected area) were converted to numeric data types.

Then the dataset was looked upon for different errors.

Exploratory Analysis:

The data set was drawn in a pair plot. Some observations from the pair plot are:

- Since Impact_area is a response variable, Affected attribute has a strong positive correlation
- International support attribute is also having a association with Impact_area but surrounded by lot of noise



Applying describe function for the non numerical values, we derive at the following points.

- China has high frequency of disaster registered in the dataset
- Transport Accident type has more frequency in the dataset

Lexical Errors

When the country column was looked upon closely, there were no missing values but it had many typos and many countries were not in their proper naming convention. Hence, to clean this column, following steps were done.

- A proper list which had all countries was adapted from an online source.
- Using the list, a function where each country in the dataframe was compared with the list was used.
- All the invalid countries returned by the function were properly replaced with their respective country names. The list of invalid countries and their respective replacement are given below.

Country Names	Names With Issues
Bosnia And Herzegovina	Bosnia-Hercegovenia
Spain	Canary Is
Cape Verde	Cape Verde Is
Cook Islands	Cook Is
China	Chine,China P Rep,Hong Kong (China)
Dominican Republic	Dominican Rep
Germany	Germany Fed Rep,Germany Dem Rep
Gambia	Gambia The
Iran	Iran Islam Rep
Korea	Korea Dem P Rep, Korea Rep
Laos	Lao P Dem Rep
Libya	Libyan Arab Jamah
Macedonia	Macedonia FRY
Marshall Islands	Marshall Is
Micronesia	Micronesia Fed States
Moldova	Moldova Rep
Northern Mariana Islands	Northern Mariana Is
Palestine	Palestine (West Bank)
Sao Tome	Sao Tome et Principe
Solomon Islands	Solomon Is
Syrian Arab Republic	Syrian Arab Rep
Taiwan (China)	Taiwan
Tanzania	Tanzania Uni Rep
Turks and Caicos Islands	Turks and Caicos Is
United States	USA,United Stats
Virgin Islands U.S.	Virgin Is (US)
Wallis and Futuna	Wallis,Wallis and Futuna Is
Yemen	Yemen Arab Rep,Yemen P Dem Rep
Zaire	Zaire/Congo Dem Rep

Irregularities:

The start and end date columns did not have a proper format. They looked like numbers. So following steps were done to parse it to the right format.

- A parsing function which parses each of the date and converts them into *dd-mm-yyyy* format.
- Since the date had lot of missing dates and month, removing improper dates was not possible as it would result in loss of data.
- So, the parsing function also converted the dates to the following formats: *mm-yyyy* and *yyyy*

Integrity constraint violations:

After parsing the dates to the above format, they were still in object datatype. A function was used to check the following Integrity constraint violations.

- Some start dates for a disaster was after the end dates which is not valid
- Some dates had invalid days like 31st of September which is a constraint violation.

Since there were only 6 instances, which violated the constraints. Manual imputation of correct dates of the respective disasters were found online and were replaced.
Still, the columns had many missing dates and month and they have to be in datetime format. So, the function was used for the imputation of the missing values which had the following rules.
For Start Dates:

- If the month was "00", month will be set to "01"
- If the day was "00", day will be set to "01"
- If the day was greater than the days in the month, the last day of the month will be set.

For End Dates:

- If the month is "00" set the month to "12"
- If the day is "00" set to the last day of the month
- If the day is greater than the days in the month, set to the last day of the month

Duplicates:

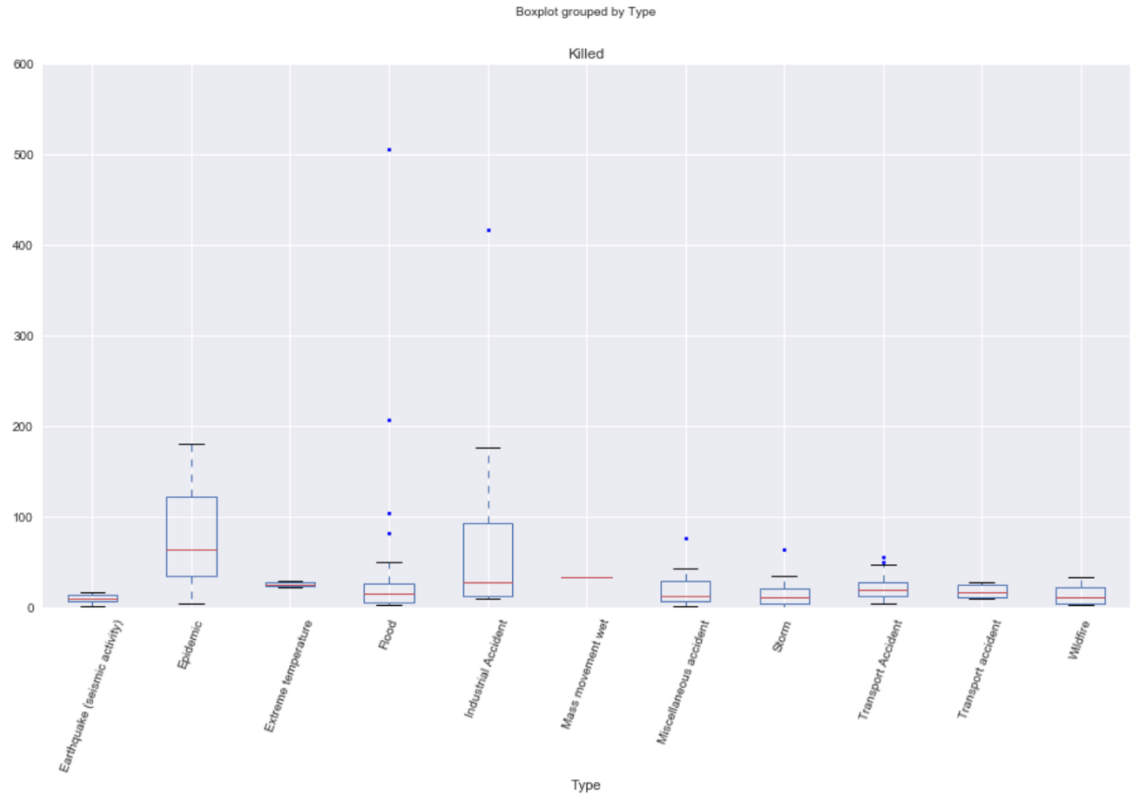
The entire dataset had no proper duplicates like where two rows were entirely identical. But when subset of attributes were selected, there were many duplicates in which one numerical value alone was different. For example, the Sao Paulo storm in Brazil had the 'Killed' value alone different. This might be due to updates in the dataset. Removing these duplicates will affect the originality of the dataset because we do not whether the value was increased or decreased. Hence, removing duplicates is difficult with the dataset.

Inconsistencies:

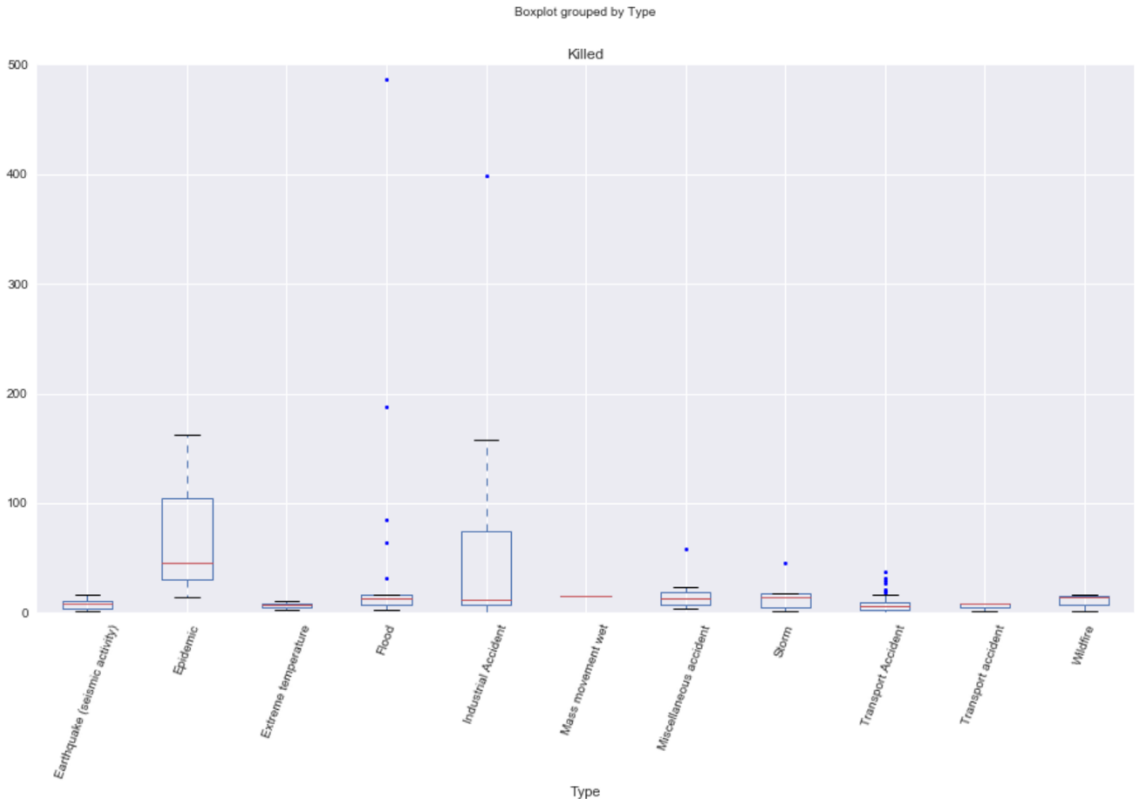
The 'Type' and 'Sub-type' had lot of inconsistencies. Since they are categorical attributes, converting them to categories and listing them showed lot of inconsistencies in naming the category. For example, *Transport Accident and Transport accident* were two different categories. So, a lambda function to replace all the initials to Uppercase letter was done to rectify the inconsistency.

3. Task 2 – Boxplot and Hampel Identifier

The given task was to draw a boxplot for number of people killed in South Africa for each type of disaster. The boxplot obtained was as follows.



There are some very wide outliers for categories like Flood and Industrial accident. To test the genuineness of the outliers, Hampel Identifier(Median Absolute Deviation) was used for the Killed column and boxplot was drawn again.



The entire range of the dataset has reduced but the outliers remain the same. The reason why MAD was chosen over Standard Deviation was median was more resistant to outliers than mean [1]. Hence, the outliers are genuine and was not imputed. To double check, the outliers for flood was checked online and results were the same.

4. Task 3 – Dealing with missing data

3.1 Deletion

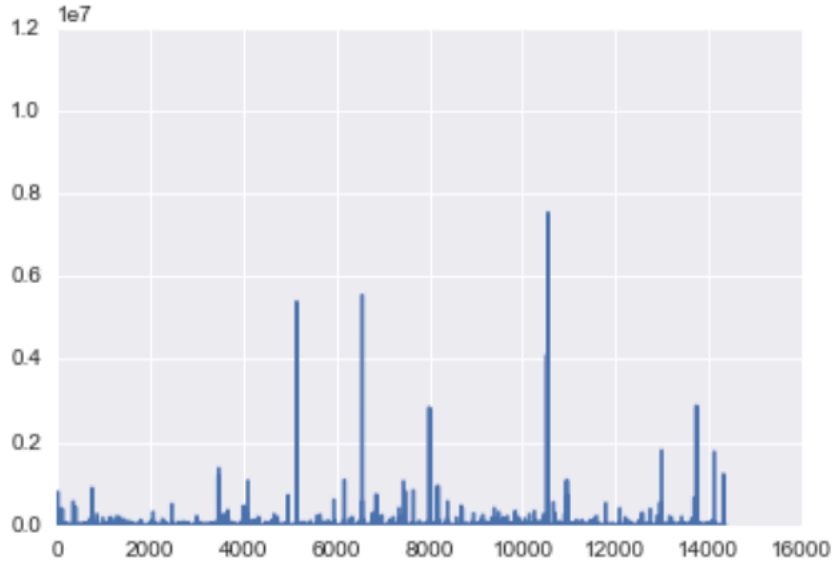
When the missing values of all the columns were evaluated, 'Cost' column had 11323 out of 14350(79%) rows missing. There was no proper collection of data to impute the missing values. Hence, the column will be dropped. Remaining attributes had enough data to be imputed.

3.2 Imputation using Linear Regression

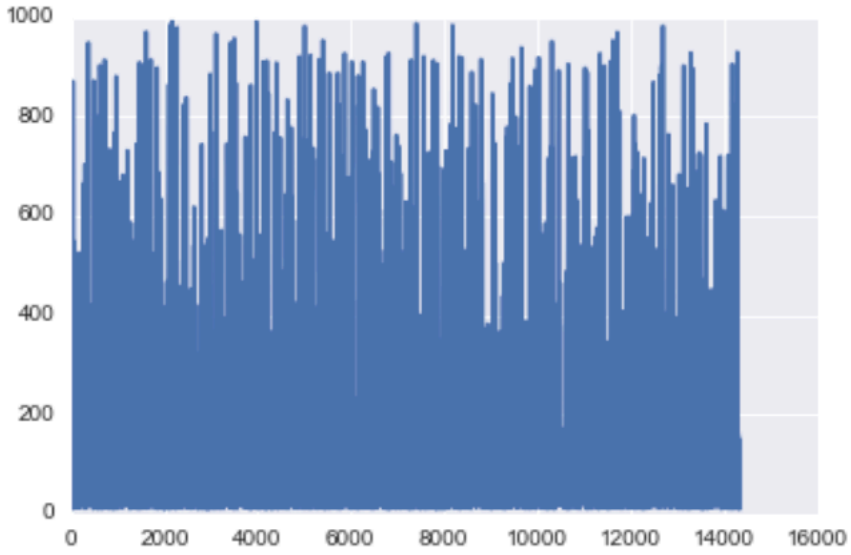
The regression model had to be trained with four attributes: "Country", "Type", "Killed" and "Affected".

Steps involved for training the data set:

- The regression was for Impact_area, so the values corresponding to the non-empty values in the Impact_area was taken.
- After training, the Standard error was 1835.22 which is very high.
- Then, we try to plot the Impact area.



- It has a very wide range
- So out of the available 7422 rows, we sample 75%(5281 rows) approximately, to reduce the range. So, the plot's range decreased and the values looked like a cluster.



- Now the range of impact area is between 10 and 1000.
- Training the model using the given regression function, Standard error was 12.61

Steps involved in predicting the values:

- To predict, the following attributes were needed: ‘Country', 'Type', 'Killed', 'Affected',' Impact_area'.
- The dataframe needed should have all missing values of impact area.
- The corresponding killed and affected columns must not be empty to predict the model. So, we are filling all the missing values with mean of their respective columns.
- Running the predict function, we get a array of predicted values which will be replaced for the missing values of original dataframe.
- The imputed Impact area column had values in scientific notation which was converted to float format.

5. Other problems:

The location column had the following problems.

- colon(:) was found in between strings and it acts as a separator between locations
- triple dots(...) was found in most of the columns
- some data manipulation process would have caused those triple dots(...). Because every instance of triple dots is at the end of the string and the string length is around 20.
- Some locations are in other languages encoded as hexadecimals.(Example : xc3\xaf\xc2\xbf\xc2\xbd)

The location column was cleaned by replacing (:) with ‘,’(commas). Triple dots were stripped from the end. And the data is incomplete because of some data extraction issues.

6. Conclusion

Hence, the cleaned dataframe is finally loaded into 26994437_cleaned_data.csv file. The dataset still has lot of problems such as handling the missing data properly in numerical attributes. Solving them would further improve the quality of the dataset.

Reference:

Pearson. R, Feb 16, 2016., Finding outliers in numerical data, Retrieved from http://exploringdatablog.blogspot.com.au/2013/02/finding-outliers-in-numerical-data.html on 01-09-2016