ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

A Study on Big Data Analytics through R

Prity Vijay[1], Bright Keshwani[2]

Research Scholar, Computer Science, Suresh Gyan Vihar University, Jaipur, India[1]

Professor & HOD of Computer Dept, Suresh Gyan Vihar University, Jaipur India[2]

ABSTRACT: Data Explosion had been marked to its extent from preceding few years. Social networking sites like Twitter, Facebook, LinkedIn had given lots of scope for data analysis. But extremely large datasets are actual headache for scientist, as traditional statistical software's are not well suited for analyzing massive data sets. Working with conventional statistical software programs can cause many performance issues when applied to Big Data sets, resulting time-consuming computational time or frequent crashing of the program. As the volume of obtainable data is mounting much rapidly than that of its computational ability, scientist were bound to choose smarter technologies for Big Data analytics. Researchers puts tons of efforts and time in cleaning the data in spite of that, there had been very few research on an effective way of data cleaning. Therefore, with the medium of this paper we tried to put some small but vital part of data cleaning, with the help of R. Although R package contain lots of readymade functions which can help scientist in many ways during the process of analysis but in this paper, we tried to explore some very basic and important function which can help out analyst in many ways while cleaning of Big Data.

KEYWORDS: R, Big Data, Hadoop, Map Reduce, Cloud Computing, Data Cleaning, Machine Learning

I. INTRODUCTION

We are living in an era of Big Data. An era where data is growing day-by-day terrifically in a tremendous speed[1] [2]. According to IDC, from 2005 data growth is happening by a factors of 300 all over the world and will become 4000 Exabyte till 2020, which mean that data will become double in every two years. Due to emergence in Big Data, Big Data Science is also becoming broader thus changing our world[4]. The flood of Big Data Science is not going to stop anywhere.

Huge amount of diverse data is coming from variety of places in an incredible speed thus forming Big Data. These data's are precious but only if we can manage it properly. Lots of challenges arise when we want to store, integrate and analyze Big Data from scattered place. Big Data analytics tools should be powerful enough to deal with these kind of data[1]. It should support visualization[2], prediction and optimization in order to uncover the hidden facts to improve decision making which is helpful in almost all kinds of business. To address these challenges, various solutions are given by the industry experts. Today, Cloud computing is one of the reliable and cheapest solution[2]. No SQL databases and Distributed file systems are appropriate for storing and managing huge datasets[1]. Hadoop framework provide solution for huge amount of data storage, data management as well as data processing. R is most famous programming framework for Big Data analytics and statistics tasks[3].

II. R: POWERFUL TOOL FOR BIG DATA ANALYTICS

R is the open-source programming language as well as data analysis environment to perform, a number of tasks essential for the effective processing and analysis of big data[5]. The reason for R popularity is augmentation of Big Data[6]. At present R is one of the 20 most popular general purpose programming languages because of its paranormal statistical capabilities. R contains number of ready-to-use machine learning and statistical algorithms which allow users to create reproducible research and develop data products. Researchers and data science experts are R lover as it is very simple yet influential environment providing lots of feature to its user. Today, it is possible to manage, analyze and visualize terabytes or even petabytes of data in an efficient manner just because supremacy of R[6].

Even though, Python and other analytical tools are available, R leads because it is the only open source programming language specially for statistics. it's framework contain enormous in built machine learning and statistical algorithm along with mathematical models . R is compatible with NoSql as well as Sql databases, providing users with the option

to import data in a variety of formats. R is an object oriented language giving facility to its user in creating their own grammar. Furthermore, R is very much flexible and can runs on all common operating systems. Distributions of R include the core R console for Windows, Mac or Linux and a number of graphical interfaces such as RStudio[7] .
R allows the user to generate data outputs in an immeasurable number of graphical formats, including plots, graphs, diagrams and dashboards. Recent R developments have enabled the building of interactive data dashboards using javascript libraries such as D3.js, rCharts, Google Charts, leaflet.js, high charts or html widgets. As there is a lot of activity around R at the moment, we can expect many other innovative ideas to develop from the R community in the coming years. R facilitates complete Data Management processes such as: (a) Data transformations (b) Data cleaning (c) Data analysis (d) Data visualization (d) Preparing the data for statistical (e) testing, etc. But this paper is concerned about the significance of data cleaning while analysis of Big Data set and some fair example of it[6].

III. DATA CLEANING AND R

Dirty data are everywhere. In fact, most real world dataset start-off with grubbiness in one way or another. But after the completion of data cleaning process , they make their way to be clean and prepare for analysis. With the emergence of Big Data, data cleaning become more important than ever before[9]. Every industries, health care, finance, hospitality and even education discussions are equivalent in a large sea of data. Data cleaning is essential as well as critical part of Data Science process[10]. Whenever ,we talk about analysis of large dataset ,we broke down this process into certain steps which starts from collecting of data ,to cleaning then analyzing the data and lastly to make a report on it.

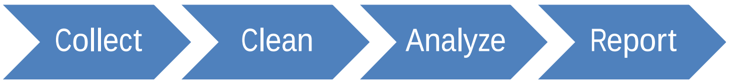


Figure 1: Big Data Analysis

Data cleaning is a first step and repeats many times over the course of analysis until new problems come to light or new data is collected. If you try to omit this step, then you pick-up a problem of getting raw data to work with R or Python. And this can be tricky for many reason. It is also time consuming process. Approximately,50 - 80% of time is needed for data scientist for collecting and preparing raw digital data before it can be explore for analysis[8]. Therefore, it should be done very carefully .Cleaning can be classified into couple of steps:

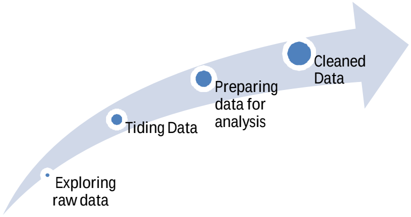


Figure 2:Data cleaning process starts from (a)Exploring raw data to (b) Tiding data and then (c) Preparing data for analysis ,finally to (d) get clean data for analysis:

IV. CLEANING DATA WITH R FUNCTION'S

Raw data is the data as it comes in. The very first job in the process of data cleaning is to understand the structure of the raw data which literally mean to have in-depth knowledge about the data which you want to clean. Raw data files may lack headers, contain wrong data types (e.g. numbers stored as strings), wrong category labels, unknown or unexpected character encoding and so on. In short, reading such files into an R data.frame directly is either difficult or impossible without some sort of preprocessing. The first thing to do when you get your hands on a new dataset is to understand its structure. There are several ways to go about this in R, each of which may reveal different issues with your data that require attention. Throughout this paper we have used a dataset called bmi. The data, give the mean body mass index (BMI) among males in each country for the years 1980-2008, come from the School of Public Health, Imperial College London.

The very first function which we can use in the process of data cleaning is class().One can use this function to determine the type of an object. This function is very simple yet powerful.

```
> class(bmi)
[1] "data.frame"
```

dim() function in R gives the whole dimension of the dataset. On applying this function to any dataset we can very easily find out the total number of rows and column. dim(bmi) will give altogether 199 rows and 30 column.

```
> dim(bmi)
[1] 199  30
```

Well, dim() will give only the numbers of rows and column but If we want to know the names of all the column of a large dataset ,we can take help of names() function in R. In the below example we can be see the name of all 30 column.

```
> names(bmi)
[1] "Country" "Y1980"   "Y1981"   "Y1982"   "Y1983"   "Y1984"   "Y1985"
[8] "Y1986"   "Y1987"   "Y1988"   "Y1989"   "Y1990"   "Y1991"   "Y1992"
[15] "Y1993"   "Y1994"   "Y1995"   "Y1996"   "Y1997"   "Y1998"   "Y1999"
[22] "Y2000"   "Y2001"   "Y2002"   "Y2003"   "Y2004"   "Y2005"   "Y2006"
[29] "Y2007"   "Y2008"
```

Since bmi doesn't have a huge number of columns, you can view a quick snapshot of your data using the str() command. In addition to the class and dimensions of your entire dataset, str() will tell you the class of each variable and give you a preview of its contents. Since the output is too large containing the information of almost 30 variables. we are showing some portion of the whole dataset

```
> str(bmi)
'data.frame':       199 obs. of  30 variables:
$ Country: chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
$ Y1980  : num  21.5 25.2 22.3 25.7 20.9 ...
$ Y1981  : num  21.5 25.2 22.3 25.7 20.9 ...
$ Y1982  : num  21.5 25.3 22.4 25.7 20.9 ...
```

The glimpse() function from dplyr is a slightly cleaner alternative to str(). str() and glimpse()are almost same which gives a preview of your data, but glimpse() is much advanced than str() as it gives the type of each and every column which is associated with dataset.

```
> library(dplyr)
> glimpse(bmi)
Observations: 199
Variables: 30
$ Country <chr> "Afghanistan", "Albania", "Algeria", "Andorra", "Angola", "...
$ Y1980   <dbl> 21.48678, 25.22533, 22.25703, 25.66652, 20.94876, 23.31424,...
$ Y1981   <dbl> 21.46552, 25.23981, 22.34745, 25.70868, 20.94371, 23.39054,...
$ Y1982   <dbl> 21.45145, 25.25636, 22.43647, 25.74681, 20.93754, 23.45883,...
$ Y1983   <dbl> 21.43822, 25.27176, 22.52105, 25.78250, 20.93187, 23.53735,...
```

You can use the summary() command to get a better feel for how your data are distributed, which may reveal unusual or extreme values, unexpected missing data, etc. For numeric variables, this means looking at means, quartiles (including the median), and extreme values. For character or factor variables, you may be curious about the number of times each value appears in the data (i.e. counts), which summary() also reveals.

```
> summary(bmi)
Country             Y1980             Y1981             Y1982     ...
Length:199          Min.   :19.01     Min.   :19.04     Min.   :19.07 ...
Class:character     1st Qu.:21.27     1st Qu.:21.31     1st Qu.:21.36 ...
Mode:character      Median :23.31     Median :23.39     Median :23.46 ...
                    Mean   :23.15     Mean   :23.21     Mean   :23.26  ...
                    3rd Qu.:24.82     3rd Qu.:24.89     3rd Qu.:24.94  ...
                    Max.   :28.12     Max.   :28.36     Max.   :28.58 ...
```

All the functions discussed above is a very first step of data cleaning which gives us basic feel of the data. All of these function is very simple but authoritative to know the dataset and get familiars to it. Next, to it one should have to look into the data more deeply than ever before and visualizing it carefully.

Looking into intensely is very simple to hear but actually its difficult when we deal with big data. Sub setting the data can be simplest option to avoid the challenges caused by Big Data set. A. Because, data files are often much larger than we need them to be; they usually contain more variables than we need for our analysis, or we plan to run our models on subsets of the data. In these cases, loading the excess data into the R workspace only to purge it (or ignore it) with a few commands later is incredibly wasteful in terms of memory. A better approach is to remove the excess data from the data file before loading it into R. Thanks to R that we are having a handful of R command that can help to break the data

The most basic way to look at your data in R is by printing it to the console. As you may know from experience, the print() command is not even necessary; you can just type the name of the object. The downside to this option is that R will attempt to print the entire dataset, which can be a nuisance if the dataset is too large. One way around this is to use the head() and tail() commands, which only display the first and last 6 rows of data, respectively. You can view more (or fewer) rows by providing as a second argument to the function the number of rows you wish to view. These functions provide a useful method for quickly getting a sense of your data without overly cluttering the console.

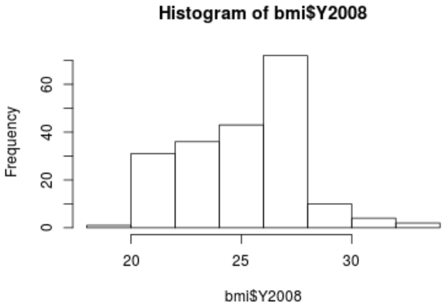
```
> head(bmi)
          Country        Y1980        Y1981        Y1982        Y1983        Y1984
1        Afghanistan     21.48678     21.46552     21.45145     21.43822     21.42734...
2        Albania          25.22533    25.23981     25.25636     25.27176     25.27901...
3        Algeria         22.25703     22.34745     22.43647     22.52105     22.60633...
4        Andorra         25.66652     25.70868      25.74681    25.78250     25.81874...
5        Angola          20.94876     20.94371     20.93754     20.93187     20.93569...
6       Antigua          23.31424     23.39054     23.45883     23.53735      23.63584...
```

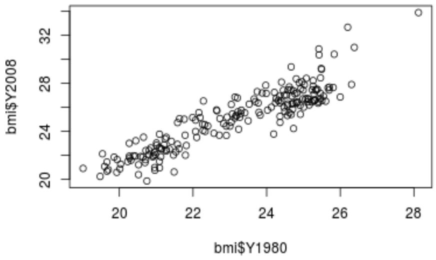
```
tail(bmi)

         Country         Y1980        Y1981        Y1982        Y1983        Y1984
194      Venezuela       24.58052     24.69666     24.80082     24.89208     24.98440
195      Vietnam         19.01394     19.03902     19.06804     19.09675      19.13046
196      West Bank       24.31624     24.40192     24.48713     24.57107     24.65582
197      Yemen           22.90384     22.96813     23.02669     23.07279     23.12566
198      Zambia          19.66295     19.69512     19.72538     19.75420      19.78070
199     Zimbabwe          21.46989    21.48867     21.50738     21.52936     21.53383
```

Some time to identify the issues of the data is to plot them. There are many ways to visualize data. There are many functions for data visualization, we will only touch on two types of plots that may be useful for quickly identifying extreme or suspicious values in your data: histograms and scatter plots.

A histogram, created with the hist() function, takes a vector (i.e. column) of data, breaks it up into intervals, then plots as a vertical bar the number of instances within each interval. A scatter plot, created with the plot()function, takes two vectors (i.e. columns) of data and plots them as a series of (x, y) coordinates on a two-dimensional plane.

- Use hist() to look at the distribution of average BMI across all countries in 2008.
- Use plot() to see how each country's average BMI in 1980 (x-axis) compared with its BMI in 2008 (y-axis).





V. CONCLUSION

An era of Big Data had began. Technologies are not growing so fast as the Data. Therefore, researchers are interested only to those approaches who can contribute more to Big Data. Although there are many other techniques and tools which suited well with big data, but we are exclusively concentrated on R. Hopefully, our contribution in explaining R as a precious piece of software for Big Data cleaning is fruitful. We are able to present only few concept but definitely, R can do much more than this. At the end we would like to add , the technology which can run fast in the race of Big Data will survive and other had to leave out.

REFERENCES

1. C.L. Philip Chen, Chun-Yang Zhang, “Data intensive applications, challenges, techniques and technologies: A survey on Big Data” Information Science 0020-0255 (2014), PP 341-347, elsevier
2. Han hui At. Al. (Fellow, IEEE),” Toward Scalable Systems for Big Data Analytics: A Technology Tutorial”, IEEE 2169-3536(2014),PP 652-687
3. Lei Wang At. Al., “BigDataBench: aBigDataBenchmarkSuitefromInternetServices”,IEEE 978-1-4799-3097- 5/14.
4. Hitesh Goyal, Surender Singh," Big Data Analysis Using R (Big Data Analysis Applications, Challenges, Techniques) ",International Journal of Advanced Research in Computer Science and Software Engineering(2015),Vol 5(9)
5. Sylvia Tippmann,"PROGRAMMING TOOLS: ADVENTURES WITH R", Macmillan Publishers Limited, Jan 2015(517)
6. Victoria Moody(2015), "The power of R: methods for processing big data", Accessed: http://blog.ukdataservice.ac.uk/the-power-of-r-methods-for-processing-big-data/
7. The Comprehensive R Archive Network Accessed: https://cran.r-project.org/
8. Dasu T, Johnson T , "Exploratory Data Mining and Data Cleaning ", Wiley-IEEE ,2003
9. R Development Core Team. 2012. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.
10. Edwin de Jonge ,Mark van der Loo(2013)," An introduction to data cleaning with R", Accessed: https://cran.r-project.org/doc/contrib/de_Jonge+van_der_Loo-Introduction_to_data_cleaning_with_R.pdf"