Yayın Geliş Tarihi (Submitted): 25/03/2019
Yayın Kabul Tarihi (Accepted): 20/06/2019
***Makele Türü (Paper Type):* Araştırma Makalesi – Research Paper**

HADOOP ÜZERİNDE ÖLÇEKLENEBİLİR BETİMLEYİCİ İSTATİSTİK UYGULAMALARI

Özgür YILMAZEL[1]

ÖZET

Büyük Veri, İngilizce dilindeki karşılığı ile Big Data, çağımızın en güncel teknolojilerinden biri olarak karşımıza çıkmaktadır. Sosyal medya, sensör verileri, Nesnelerin İnternet'i gibi seri halde veri üreten teknolojilerin sayesinde veri hacmi gün geçtikçe artmaktadır. Dünyada veri miktarındaki büyük artış, büyük verinin depolanması, işlenmesi ve analiz edilmesi için farklı yaklaşımlar gerektirmektedir. Bir nicel veriseti birçok özelliğe sahiptir ve betimleyici istatistikler veri setindeki bu özellikleri her bir değeri listelemek zorunda kalmadan anlamlı ve yönetilebilir bir biçimde tanımlayabilir. Bununla birlikte, standart istatistiksel teknikler, verinin büyüklüğü, karmaşıklığı ve hızı nedeniyle büyük verilere uygun olmayabilir. Nicel verileri analiz etmek için kullanıma hazır çok sayıda istatistiksel araç olmasına rağmen, her zaman büyük veri dosya sistemleri ile çalışmak için uyumlu değildir. Bu yazıda, betimleyici istatistik algoritmalarının büyük veri setleri üzerindeki uygulamaları sergilenmektedir ve deneylerin 196 yivli küçük bir Hadoop kümesinde ölçeklenebilirliğini gösterilmektedir. Bu çalışma, büyük veri kümeleri için tanımlayıcı istatistiklerin bir Hadoop kümesinin dağıtılmış hesaplama özelliklerinden yararlanabileceğini göstermektedir. Çalışma TÜBİTAK TEYDEB desteği ile tamamlanmıştır.

Anahtar Kelimeler: Büyük Veri, Betimleyici İstatistik, Hadoop, MapReduce

[1] Sorumlu Yazar, Doç. Dr., Eskişehir Meslek Yüksekokulu, Anadolu Üniversitesi, Eskişehir, Türkiye, ORCID:https://orcid.org/0000-0002-8932-9587

SCALABLE IMPLEMENTATIONS OF DESCRIPTIVE STATISTICS ON HADOOP

ASBTRACT

Big Data is one of the most trendy technologies of our time. The volume of data is increasing day by day, thanks to serial data generation technologies such as social media, sensor data, Internet of Things. The massive increase in the amount of data accumulated around the world requires different approaches to store, process and analyze the big data. A set of quantitative data has many features and the descriptive statistics can describe these features in a meaningful and manageable form without having to list every value in the dataset. However, the standard statistical techniques cannot suit big data due to the size, complexity and velocity of the data. Though there are many off-the-shelf statistical tools available to analyze quantitative data they are not always compatible with the big data file systems. In this paper, we describe our implementations of the descriptive statistics algorithms over big data and show the scalability of our experiments on a small Hadoop cluster with 196 threads. This study presents that descriptive statistics for large datasets can benefit from distributed computation features of a Hadoop cluster.

Keywords: Big Data, Descriptive Statistics, Hadoop, MapReduce

1. INTRODUCTION

Big data is a collection of datasets that are large and complex enough that traditional database management or processing tools cannot process. National Science Foundation (NSF, 2012) solicitation refers to Big data as “large, diverse and complex, longitudinal, and/or distributed data sets generated from instruments, sensors, Internet transactions, emails, videos, click streams, and/or all other digital sources available today and in the future”. Bigness is not just about the size as Gobble (2013) mentions that data is considered ‘big’ because the volume is big, because the data is moving fast, or because it is not structured in a usable way. Gartner (Douglas, 2001) defined big data as ‘Big data is high-volume, high-velocity, and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight, decision making and process automation” and introduced the 3V's for big data which are volume, velocity and variety as follows:

- Volume: This feature refers to large amounts of data that is being collected or processed. The size of data has been growing at an increasing rate and this applies to both companies and individuals. More and more data is being generated not only by employees of organizations but also by partners, customers, sensors, networks, machines and so on. As the source of data increases the volume of data increases. Peta byte is now a commonly known measure for data size.

- Velocity: This feature is the measure of how fast the data is coming in. The flow of data is massive and continuous as a result of interconnection and advances in network technology. As a result, the data is flowing faster than we can make sense out of it.
- Variety: This feature refers to the different kinds of sources for data. Data variety is a measure of the richness of the data representation, including text, images, video, audio, data bases, SMS, sensor data and so on. Both structured and unstructured data is the subject of big data.

When the above characteristics of big data are considered, the traditional computational and analyzing methods cannot keep up with big data. In order to capture the value from big data we need new sets of techniques and technologies. Big data techniques involve a number of disciplines, including statistics, data mining, machine learning, neural networks, social network analysis, signal processing, pattern recognition, optimization methods and visualization approaches (Chen & Zhang, 2014). Among these techniques, statistics is the science to collect, organize and interpret data. In this study, we describe our implementations of the descriptive statistics algorithms over big data and show the scalability of our experiments on a small Hadoop cluster with 196 threads. This study presents that descriptive statistics for large datasets can benefit from distributed computation features of a Hadoop cluster.

2. LITERATURE REVIEW

Statistical techniques are usually preferred to explore correlations between objects or present numerical descriptions. A set of quantitative dataset has many features. One of the goals of descriptive statistics is to describe these features within the dataset in a meaningful and manageable form without listing every value of the dataset. This goal is also valid for big data where we would need statistics to describe the big dataset in a meaningful form. However, the standard statistical techniques cannot suit big data due to the size, complexity and velocity of the data. Therefore, there has been research done to propose extensions for the traditional statistical techniques or for exploring new statistical computational methods for that purpose (Ciaccio et al., 2012). It is essential for the researchers to consider and deal with the scalable and distributed versions of statistical methods when working with big data (Cheung, 2012; Daas et al.,2012; Glasson et al., 2013). The big data concept requires parallel processing and processing in real time. Processing a regular dataset versus processing big data to calculate descriptive statistics require different approaches. For instance, the problem with quantile computation, while well-solved in the classical model of computation, assumes a new and challenging character within the constraints of single-pass computation (Buragohain & Suri, 2009). Munro and Paterson (1980) suggest that it is not possible to compute the quantile precisely when the algorithm's memory is limited and therefore the best solution is an approximation. Munro and Paterson (1980) also proved that computing the true median would require memory which is linear in the size of the set. As the size of the dataset gets bigger the computation time will also get bigger linearly. Therefore, calculating descriptive statistics may require approximation approach where the computation time increases linearly with the data size.

Sysoev (Sysoev et al., 2011) proposed that efficient approximate algorithm for large-scale multivariate monotonic regression which is an approach for estimating functions that are monotonic with respect to input variables. Philippe (Philippe et al. 2011) discusses several parallel statistics algorithms. Klemens (2008) and Wilkinson (2008) discuss statistical computing and Hastie (Hastie et al., 2002) focuses on statistical learning which is one of the leading research fields within statistics domain.

3. STUDY

In this study, we implement various descriptive statistics algorithms to work on a big dataset for the following most commonly used descriptive statistics features: frequency, mean, median, mode, variance, standard deviation, maximum, minimum, range, skewness, kurtosis, quantiles, histogram diagram, sequence graph and scatter plot. Our goal in this study is to provide scalable implementations for the above descriptive statistics algorithms and show that these algorithms are scalable over a Hadoop cluster.

Hadoop is a software framework for the distributed computing. It is designed to scale up from single servers to thousands of machines to offer local computation and storage. Horizontal scalability is the ability to connect multiple hardware or software entities so that they work as a single logical unit in order to increase capacity. Hadoop has horizontal scalability feature as more nodes can do more work within the same time and as a result the data size and the computing resources are linear.

We implement the descriptive statistics algorithms by using the MapReduce framework of the Hadoop environment. MapReduce is a programming model where each calculation is executed over a small part of data in a local node and then results are collected and represented at a reduce step. Therefore, MapReduce breaks up computation tasks into units which is then distributed around a cluster, as a result it provides cost-effective and horizontal scalability ("Hadoop Releases", 2011). There are two major phases in MapReduce which are Map Phase and Reduce Phase. The raw data is the input to the map phase where the key-value pairs are sorted and grouped by the map function. The output of the map phase is the input to the reduce phase. The output of the map phase is in the form of an iterable list of values with matching keys. The reduce function iterates through this list and performs the operations on the data and then calculates the final result (White, 2012).

In the next few sections we present implementation approaches for different descriptive statistics features:

- Mean is the average of all numbers and is sometimes called the arithmetic mean. Mean is very sensitive to outliers so that it needs to be calculated precisely. Calculation of mean is horizontally scalable when programmed with MapReduce. MapReduce partitions the data into chunks and distributes the job to the chunks so that each piece sends the count and the sum values to the reduce process within MapReduce. All the sums and counts are added and divided by so that the mean is calculated with MapReduce.
- Median is the "middle" number in a sequence of numbers. Although the median does not represent a true average it is not greatly affected by the presence of outliers as the mean. Therefore, the approximate calculation is possible with mean (Manku et al., 1998; Battiato et al., 2000; Kelley & Blumenstock, 2014) in large datasets. The algorithm in (Battiato et al., 2000) is programmed for the Hadoop environment to calculate the approximate median in a big dataset.
- Mode is the number that occurs most often within a set of numbers. Since mode calculation is horizontally scalable where a map stage generates a key and the reduce stage places all the elements into a data array for each key within the MapReduce algorithm it can use the exact calculation for the algorithm.
- Variance is the measure of how the dataset is spread out. It is calculated as the average of the squared differences from the mean. Variance is horizontally scalable for MapReduce programming.

- Standard deviation is the square root of the variance. While the variance gives a rough idea for the spread of the dataset the standard deviation is more concrete as it gives the exact distance from the mean value. Standard deviation formula is also horizontally scalable and as a result it can be programmed for exact calculation with MapReduce.
- Maximum is the value that is greater than or equal to all other values in the dataset. The minimum is the value that is less than or equal to all other values in the dataset. The difference between the maximum and the minimum gives us the range of the dataset. Both values are very sensitive to outliers and as a result both should be calculated precisely. The maximum, minimum and the range values can be calculated with MapReduce functions.
- Skewness is the measure of symmetry. A distribution is symmetric if it looks the same to the left and right of the center point. Kurtosis is a measure of whether the data is heavy-tailed or light-tailed relative to a normal distribution. That is, data sets with high kurtosis tend to have heavy tails, or outliers. Data sets with low kurtosis tend to have light tails, or lack of outliers. The histogram is an effective technique to see the skewness and the kurtosis of the dataset (NIST/SEMATECH, n.d.). For the skewness Fisher-Pearson coefficient (NIST/SEMATECH, n.d.) and Galton skewness (NIST/SEMATECH, n.d.) algorithms are conducted with MapReduce functions. For the kurtosis calculation, the algorithm that Mardia and Zemroch (1975) declared are executed with MapReduce functions.
- In statistics quartiles are values that divide your data into quareters provided data is sorted in an ascending order. Quartiles are divided by the 25th, 50th and 75th percentile, also called first, second and third quartile. The P2 algorithm (Jain & Chlamtac, 1985) is coded in MapReduce to calculate the quartiles.

3.1. System Description

First of all a virtual Hadoop cluster is implemented. The MapReduce algorithms that implement the descriptive statistics algorithms are run on this cluster via multiple processors. CentOS is installed over VirtualBox and Hadoop services are run in pseudo-distributed mode.

The Hadoop cluster is consisted of different services such as storing data and handling calculations. In order to keep the distributed, scalable and reliable nature of Hadoop cluster at all times all these services have to run on different servers at the same time. In pseudo distributed mode all services required by the Hadoop cluster is run on a single node. Pseudo distributed mode does not support security, availability and scalability capabilities of a Hadoop cluster, it provides a basic development environment for software developers. Development of the algorithms are done using virtual Hadoop cluster in pseudo distributed mode. We also test the scalability of our algorithms on a 7 node Hadoop cluster with 196 Threads. Figure 1 shows the pseudo distributed Hadoop configuration on a single machine.

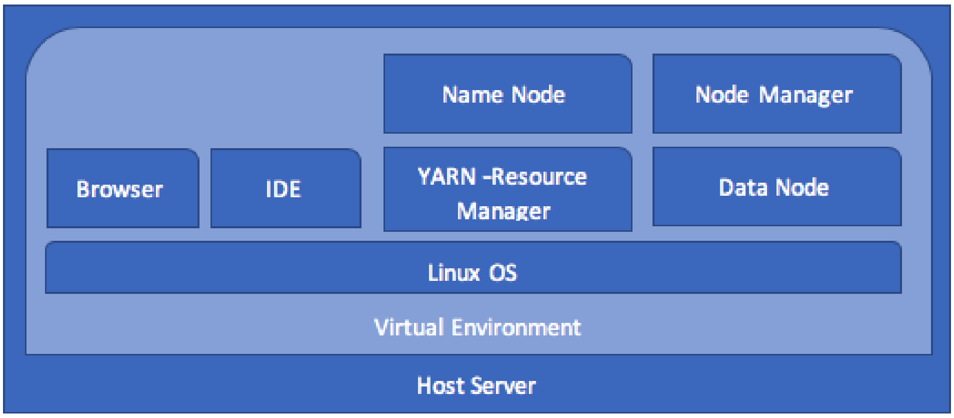


Figure 1. Pseudo distributed Hadoop configuration on a single machine.

The following Hadoop ecosystem components are used for the development of this study:

- HDFS (HDFS Users Guide, n.d.): The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems though the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.
- MapReduce (“MapReduce, n.d.): Hadoop MapReduce is a software framework which processes vast amounts of data in-parallel on large clusters in a reliable, fault-tolerant manner. A MapReduce job splits the input data-set into independent chunks which are processed by the map tasks in a parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically, both the input and the output of the job are stored in a file-system. The framework is responsible to schedule tasks, monitor them and re-execute the failed tasks.
- YARN (“Yarn”, n.d.): YARN splits up the functionalities of resource management and job scheduling/monitoring into separate daemons within Hadoop. It basically manages the distributed resource management level in Hadoop.

- Spark (“Spark”, n.d.): The Apache Spark is a powerful open source processing engine built around speed, ease of use, and sophisticated analytics. Spark handles most of its operations in-memory. As a result, this reduces the amount of time consuming writing and reading to and from slow mechanical hard drives.
- Hive (“Hive”, n.d.): The Apache Hive data warehouse software facilitates reading, writing and managing large datasets on distributed storage using SQL.

The following services are used for the development of the tool (“HDFS Architecture”, n.d.):

- NameNode: NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files in the file system. NameNode does not store the data files though it keeps all the metadata (filename, access rights, nodes etc) of the files in HDFS.
- DataNode: DataNode is responsible for storing the actual data in HDFS. DataNode is usually configured with a lot of hard disk space as it stores the actual data.
- ResourceManager: ResourceManager knows where the slaves are located and how many resources they have.
- NodeManager: When the NodeManager starts it announces himself to the ResourceManager. Each node manager offers some resources to the cluster.
- SparkHistoryServer: Spark History Server is the web UI for completed and running Spark applications.
- MapReduceHistoryServer: MapReduce History Server allows the user to get status on finished MapReduce applications.
- SparkJobServer: Spark Job Serves provides an interface for submitting and managing Apache Spark jobs, jars and job contexts. This repo contains the complete Spark job server project including unit tests and deploy scripts.
- WebServer: Webserver is a service that enables the end user to run statistical analysis.

The system architecture for the services that are used within tool is shown in Figure 2.

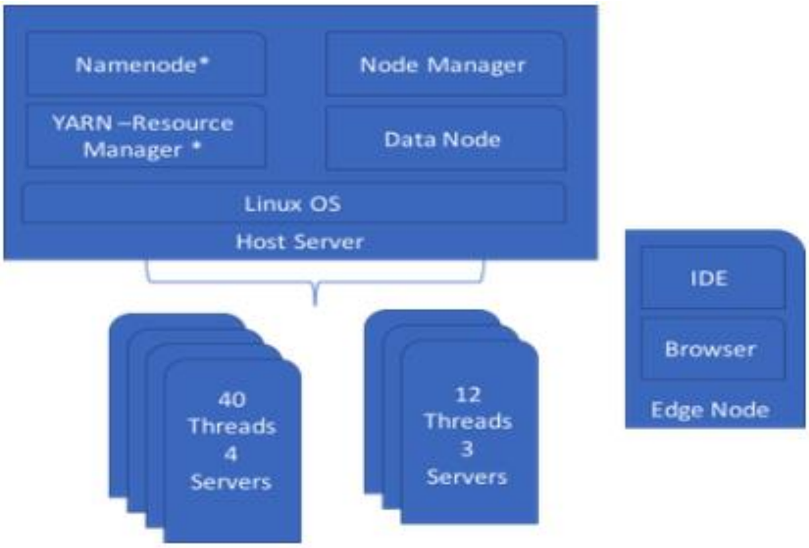
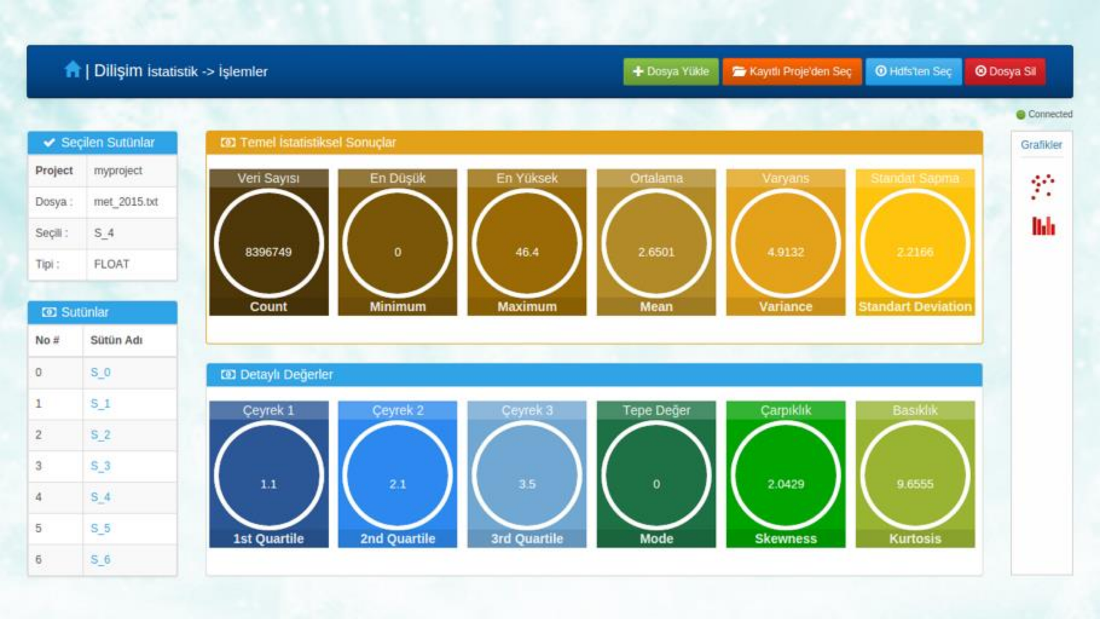


Figure 2. The system architecture for the scalability tests (*These services are run only on the 1st node).

3.2. Functionality of the System

The system is web-based and provides a functionality to upload a file from a local disk or select data from the existing HDFS. Some of the screenshots to present the functionality of the system are shown in Figure 3 through 6. Figure 3 is a descriptive statistics results interface for a given dataset. This interface displays the min, max, average, median, standard deviation, quartiles and skewness values for a given dataset in the system. Figure 4 shows the scatter plot diagram, figure 5 is the histogram plot and figure 6 is the sequence graph for the dataset within the system.



Seçilen Sutünlar

Project	myproject
Dosya :	met_2015.txt
Seçili :	S_4
Tipi :	FLOAT

Sutünlar

No #	Sütün Adı
0	S_0
1	S_1
2	S_2
3	S_3
4	S_4
5	S_5
6	S_6

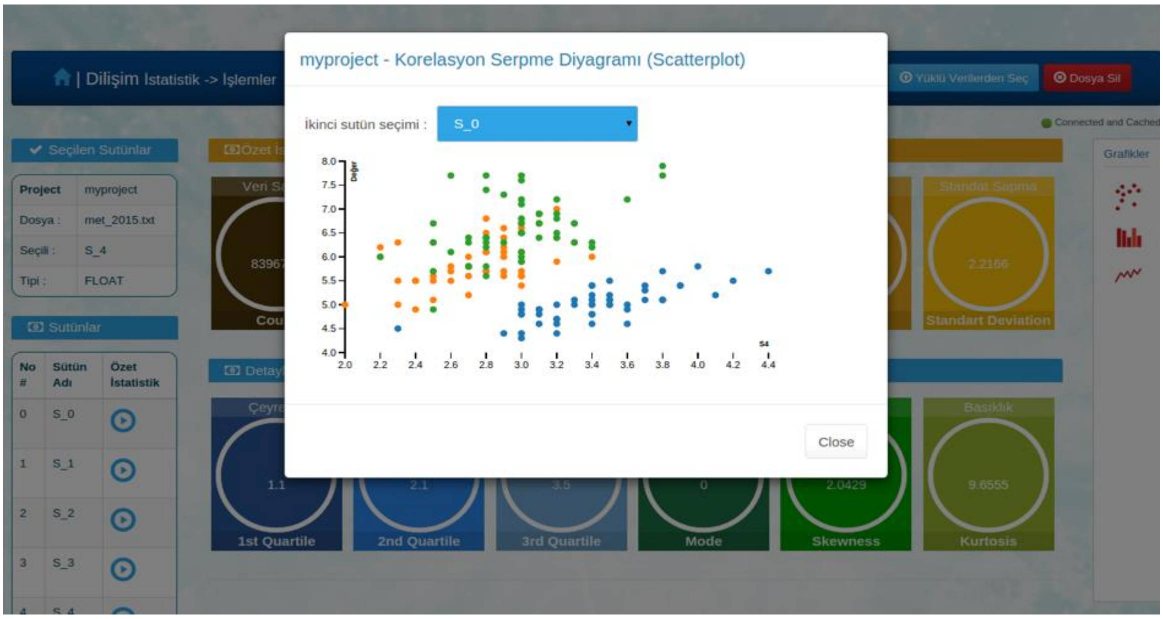
Temel İstatistiksel Sonuçlar

Veri Sayısı	En Düşük	En Yüksek	Ortalama	Varyans	Standart Sapma
8396749	0	46.4	2.6501	4.9132	2.2166
Count	Minimum	Maximum	Mean	Variance	Standart Deviation

Detaylı Değerler

Çeyrek 1	Çeyrek 2	Çeyrek 3	Tepe Değer	Çarpıklık	Basıklık
1.1	2.1	3.5	0	2.0429	9.6555
1st Quartile	2nd Quartile	3rd Quartile	Mode	Skewness	Kurtosis

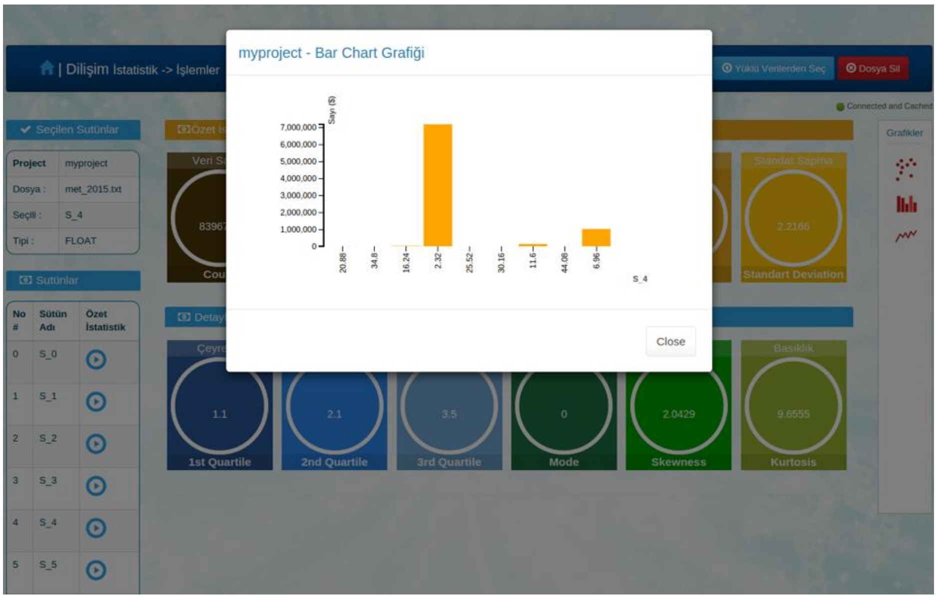
Figure 3. Descriptive statistics results interface



myproject - Korelasyon Serpme Diyagramı (Scatterplot)

İkinci sutün seçimi : S_0

Figure 4. Scatter plot diagram for the dataset.

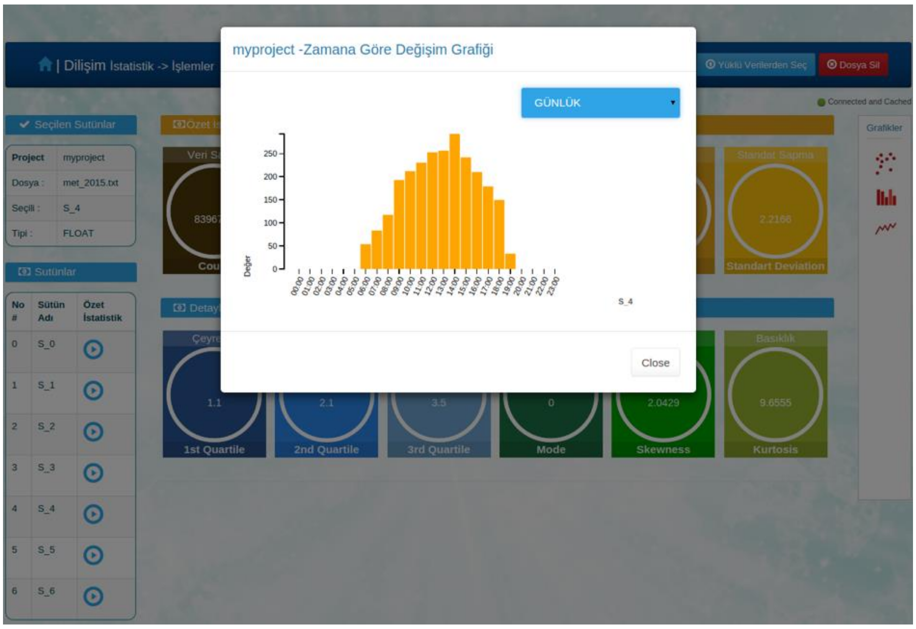


myproject - Bar Chart Grafiği

Project	myproject
Dosya :	met_2015.txt
Seçili :	S_4
Tipi :	FLOAT

No #	Sütün Adı	Özet İstatistik
0	S_0	
1	S_1	
2	S_2	
3	S_3	
4	S_4	
5	S_5	

Figure 5. Histogram plot diagram for the dataset



myproject -Zamana Göre Değişim Grafiği

GÜNLÜK

Project	myproject
Dosya :	met_2015.txt
Seçili :	S_4
Tipi :	FLOAT

No #	Sütün Adı	Özet İstatistik
0	S_0	
1	S_1	
2	S_2	
3	S_3	
4	S_4	
5	S_5	
6	S_6	

Figure 6. Sequence graph for the dataset.

4. ACCURACY AND SCALABILITY TESTS OF THE SYSTEM

In order to verify the accuracy of our descriptive statistical implementations with MapReduce, we used the meteorology dataset which includes the sensor data for wind and temperature features. In order to test the accuracy of the system all descriptive statistics are also run in R programming language for both wind and temperature datasets. The results from R and our system are reported in Table 1 and Table 2. As the results are identical, it can be concluded that the implementations are accurate.

Table 1. Accuracy test results for descriptive statistics in R and the system using wind dataset.

Descriptive Statistics	Results with R	Results with the System
Min	0.00	0.00
First Quartile	1.00	1.00
Median	1.70	1.70
Mean	2.30	2.30
Third Quartile	2.90	2.90
Max	58.20	58.20
Variance	5.06	5.06
Standard Deviation	2.25	2.25
Mode	0.00	0.00
Skewness	3.40	3.40
Kurtosis	24.98	24.98

Table 2. Accuracy test results for descriptive statistics in R and the system using temperature dataset.

Descriptive Statistics	Results with R	Results with the System
Min	-39.40	-39.40
First Quartile	12.42	12.42
Median	18.56	18.56
Mean	18.59	18.59
Third Quartile	25.50	25.50

Max	46.60	46.60
Variance	80.84	80.84
Standard Deviation	8.99	8.99
Mode	15.80	15.80
Skewness	-0.20	-0.20
Kurtosis	2.74	2.74

In order to test the scalability of our implementation, the meteorology dataset for wind feature are used. Overall 21 distributed descriptive statistics implementations are run on four different configurations to calculate the descriptive statistics on this dataset. The configurations are as follows:

i. 1 server with 1 Thread
ii. 1 server with 8 Threads
iii. 2 servers with 16 Threads
iv. 7 servers with 196 Threads

The results for each configuration are given in Table 3. The results indicate that the tool is scalable.

Table 3. Scalability test results for the descriptive statistics algorithms.

Job/Setup	**1 Server with 1 Thread (sec)**	**1 Server with 8 Threads (sec)**	**2 Servers with 16 Threads (sec)**	**2 Servers with 196 Threads (sec)**
Median	9.51	2.87	1.93	0.38
Mean	11.89	2.84	2.31	0.45
Mode	13.17	2.88	2.36	0.19
Quartiles	40.94	8.35	5.92	0.90
Min-Max-Standard Deviation- Variance	5.32	1.05	0.90	0.13
Histogram	11.51	2.80	2.54	1.34
Skewness	19.67	3.62	3.06	0.61
Kurtosis	12.39	2.79	2.30	0.85

5. CONCLUSION

The massive increase in the amount of data accumulated through sensors, mobile phones, production lines and so many other ways require us to store, process and analyze the big data. A set of quantitative data has many features and the descriptive statistics can describe these features in a meaningful and manageable form without having to list every value in the dataset. Since the standard statistical techniques cannot accommodate big data due to the size, complexity and velocity of the data we described our implementations of the descriptive statistics algorithms over big data. Many machine learning applications will greatly benefit from knowing the distribution of data and its statistical properties. Our approach allows data scientists to get insight of their data before crafting their machine learning applications and going over valuable hours of training and computational resources. We showed the scalability of our experiments on a small Hadoop cluster with 196 threads. This study shows that both approximation algorithms and exact calculations of descriptive statistics for large data can benefit from distributed computation features of a Hadoop cluster.

As a future work we would like to extend our work so that descriptive statistics can be generated for textual big data set. In this work we are planning on implementing algorithms to calculate and visualize document frequency, sparsity of a document-feature matrix, feature co-occurrence matrix on a big textual data set.

ACKNOWLEDGEMENTS

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under TEYDEB Project number 7140488.

REFERENCES

Apache Software Foundation, *Hadoop Releases*, apache.org, Dec. 10, 2011. [Online]. http://en.wikipedia.org/wiki/Apache_Hadoop. [Accessed: Oct. 06, 2018]

Battiato, S., Cantone, D., Catalano, D., Cincotti, G., and Hofri, M. (2000), An efficient algorithm for the approximate median selection problem. *Algorithms and complexity*, 226-238.

Buragohain C., and Suri S. (2009*), Encyclopedia of Database Systems*, 2235-2240, Springer US.

Chen C. P., and Zhang C.Y. (2014), Data-intensive applications, challenges, techniques and technologies: A survey on Big Data, *Information Sciences* ,275, 314-347

Cheung P. (2012), Big Data, *Official Statistics and Social Science Research: Emerging Data Challenges, Presentation at the World Bank.*

Ciaccio A. Di, Coli M., Ibanez A., and Miguel J. (2012), *Advanced Statistical Methods for the Analysis of Large Data-Sets*.

Daas P., Tennekes M., Jonge E. De, Priem A., Buelens B., Pelt M. Van, and Hurk P. Van Den (2012), *Data Science and the Future of Statistics Presentation at the first Data Science NL meetup,* http://www.slideshare.net/pietdaas/data-science-and-the-future-of-statistics.

Douglas L. (2001), *3d data management: Controlling data volume, velocity and variety*, https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf.

Glasson M., Trepanier J., Patruno V., Daas P., and Skaliotis M., Khan A. (2013), *What does "Big Data" mean for Official Statistics?* https://statswiki.unece.org/pages/viewpage.action?pageId=77170614&preview=/77170614/80805923/Big%20Data%20HLG%20Final%20Published%20Version.docx.

Gobble M. (2013), Big Data: the next big thing in innovation, *Research-Technology Management*, 56, 64–66.

Hastie T., Tibshirani R., and Friedman J. (2002), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, Stanford.

HDFS Architecture, https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html.

HDFS Users Guide, http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html.

Hive, https://hive.apache.org.

Jain, R., and Chlamtac, I. (1985). The P2 algorithm for dynamic calculation of quantiles and histograms without storing observations. *Communications of the ACM,* 28(10), 1076-1085.

Kelley, I., and Blumenstock, J. (2014). Computational challenges in the analysis of large, sparse, spatiotemporal data. *In Proceedings of the sixth international workshop on Data intensive distributed computing,* 41-46. ACM.

Klemens B. (2008), Modeling with Data: *Tools and Techniques for Statistical Computing*, Princeton University Press.

Manku, G. S., Rajagopalan, S., and Lindsay, B. G. (1998), Approximate medians and other quantiles in one pass and with limited memory. *In ACM SIGMOD Record*, 27 (2), 426-435.

MapReduce, https://wiki.apache.org/hadoop/MapReduce.

Mardia, K. V., and Zemroch, P. J. (1975), Algorithm AS 84: Measures of multivariate skewness and kurtosis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2), 262-265.

Munro J.J., and Paterson M.S. (1980), Selection and sorting with limited storage, *Theor. Comput. Sci.*, 12, 315–323.

NIST/SEMATECH, *e-Handbook of Statistical Methods*, http://www.itl.nist.gov/div898/handbook/, [Accessed: 19.01.2018].

NSF (2012), *Core techniques and technologies for advancing big data science and engineering (BIGDATA)*, https://www.nsf.gov/pubs/2012/nsf12499/nsf12499.htm.

Philippe P., Thompson D., Bennett J., and Mascarenhas A. (2011), Design and performance of a scalable, parallel statistics toolkit, *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum* (IPDPSW), 1475–1484.

Spark, https://spark.apache.org/documentation.html.

Sysoev O., Oleg B., and Grimvalla A. (2011), A segmentation-based algorithm for large-scale partially ordered monotonic regression, *Comput. Stat.Data Anal*, 55 (8), 2463–2476.

White T., (2012), *Hadoop the Definitive Guide*, 3rd Edition, O’Reilly Media.

Wilkinson L., (2008), The future of statistical computing, *Technometrics*, 50 (4), 418–435.

Yarn, https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html.