Data Analysis Case Study: Water Table Depth in Jakarta, in 2017

Dharma Kurnia Septialoka - 13514028[1]
Informatics Engineering Study Program
School of Electrical Engineering and Informatics
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13514028@std.stei.itb.ac.id

***Abstract*—As stated in Undang-Undang Nomor 14 Tahun 2008 tentang Keterbukaan Informasi Publik, then each of the government institutes should provide a public data freely to ensure the right of citizens to know the policy, program, and condition in this country by which the data does not contain sensitive or secret information of the country, government, individual, or other parties regulated in UU. Jakarta has been the pioneer of having open-data program by providing "Jakarta Open Data" in 2014. Open data means whatever data is released in a format that is free of royalties and IP restrictions in a specific way that could be used, re-used, and re-distributed by public freely. Open data has shown us the era of government which are transparent, accountable, and reliable which increases public trust. Jakarta has started to be an open-data government by proving data of ten sectors in Jakarta freely for the sake of transparency, increasing public participation, and bringing awareness to create new innovation for better Jakarta based on data. As Jakarta Open Data has more than 1047 dataset on its website, and potentially will have more and grow bigger, many of Jakarta's citizen still do not take the advantage of the data properly as seen by the small number of visitors. From various sectors and various dataset available on the website, inspired by experiencing flood in Jakarta occasionally, the dataset chosen for this paper is the data of water table depth in Jakarta in the year of 2017. This paper will talk about theory that the author used and focus on processing the dataset. We will see what we could get and analyse from the data.**

***Keywords*—data analysis, water table, water level, flood, jakarta**

I. Introduction

As government works for citizens, it has a responsibility to provide a public data freely to ensure the right of citizens to know the policy, program, and condition in the country by which the data does not contain sensitive or secret information of the country, government, individual, or other parties regulated in UU (Undang-Undang Nomor 14 Tahun 2008 tentang Keterbukaan Informasi Publik).

Starting the era of Mr. Basuki as a governor in 2014, Jakarta has been the first province in Indonesia that has open-data program. Open data means whatever data is released in a format that is free of royalties and IP restrictions in a specific way that could be used, re-used, and re-distributed by public freely. Open data has shown us the era of government which are transparent, accountable, and reliable which increases public trust. Jakarta has started to be an open-data government by proving data of ten sectors in Jakarta freely for the sake of transparency, increasing public participation, and bringing awareness to create new innovation for better Jakarta based on data. Right now, there are more than 1047 dataset available on its website and will have more and grow bigger as the time passes by.

Data is useful. There are lots of values from knowing the collection of specific data. However, even the data is opened nowadays, taking advantage of those data in a conventional way is practically hard for some reasons. Firstly, the data is so big that the computer takes a long time to open it. It is also just still a "data" so there is still no information and meaning. In addition, even if the government already opens the data, if we still do not know how to process the data properly, it will just be a waste for both parties. A good analytics enables us to make decisions using fact. We have to learn data analytics so that we could understand what the data is telling us and be a data-driven person.

The basic of data analysis is to get to know your data. If it is a simple data, we simply look at it. But what about if there is a thousand, million, or even billion of data? Big data?

II. Fundamental Theories

2.1 Data Analysis vs Data Science

To avoid misinterpret in this paper as it is restricted to data analysis, we would see the definition of both data analysis and data science.

Data analysis, also known as analysis of data or data analytics, is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, suggesting conclusions, and supporting decision-making.[1]

Data science, also known as data-driven science, is an interdisciplinary field about scientific methods, processes and systems to extract knowledge or insights from data in various forms, either structured or unstructured,[2][3] similar to Knowledge Discovery in Databases (KDD).

2.2 Basic of Python

Python is a general-purpose programming language that is becoming more popular for doing data science. People nowadays use Python to get insights from data as it provides powerful ways to store, manipulate, data cleaning, data munging, and other data-science tools to start the analyses.

For starting, we should know some Python basics, like variable, type, and operation. Then, because we will work with huge amount of data, we have to know the basic of Python list, which is to store, access, and manipulate data in list. In addition, we should also know about function, method, package, and how to work with it in Python. Last but not least, we have to be familiarized with some packages available for any scientific computations and data analysis, such as:

1. Numpy – stands for Numerical Python, a faster and more powerful alternative to the list for scientific computing and data exploration. The feature is n-dimensional array, sophisticated functions, tools for integrating C/C++ and Fortran code, and contain linear algebra function, Fourier transform, and random number capabilities.
2. Matplotlib – 2D plotting library which produces vast variety of graphs, starting from histograms to line plots to heat plots.
3. Pandas – for structured data operations and manipulations. It is extensively used for data munging and preparation.
4. Scikit Learn – simple and efficient tools for machine learning and statistical modelling, including classification, regression, clustering, dimensionality reduction, model selection, and preprocessing for the purpose of data analysis and data mining

2.3 Water Table

The water table is the upper surface of the zone of saturation. The zone of saturation is where the pores and fractures of the ground are saturated with water.[4] People like to call *water table* as *water level.* Sometimes, it has a little different meaning. The upper of the saturated zone is the water table, while the level of water seen in a well is commonly referred to as water level. However, for this paper, it could mean the same.

There are 4 alert status in water level:

1. Siaga IV: Safe: There has not been a noticeable increase in the water flow
2. Siaga III: Alert: Public should start to be careful and prepare everything from various possibilities of flood
3. Siaga II: Critical: Prolonged rainfall causes the flow of water in a stream becomes so high. Person in Charge is Head of Regional Disaster Management Jakarta Province
4. Siaga I: Disaster: High risk and threat. Ready for evacuate. Person in Charge is the governor of Jakarta

III. Analysis of Water Table in Jakarta

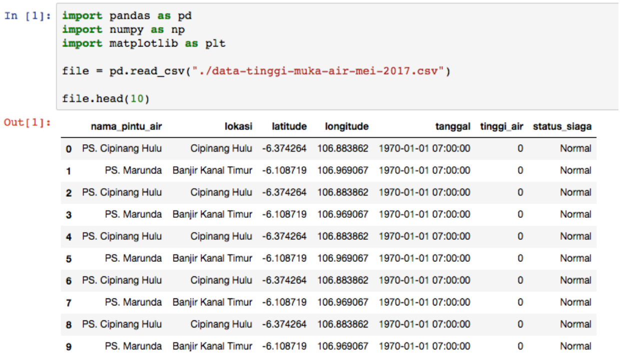
Firstly, we set up the python environment of our computer. If you are a mac user, install python3 and pip using brew. For the convenience of interactive computing that could contain live code, equation, visualization, and many more, it is suggested we use jupyter notebook. Jupyter could also be installed using pip.

This dataset shows us the data of water level of river/ sea in Jakarta by real time starting from 1 January 2017.

Variable used in this dataset:

- nama pintu air = location of the observation
- lokasi = observed object (name of the sea, river, etc)
- latitude
- longitude
- tanggal = date and time when the data is taken
- tinggi air = water level when the data is taken
- status siaga = alert status relative to water level in the area

We will load the dataset under the name *file.* We then see the first ten rows of the data.

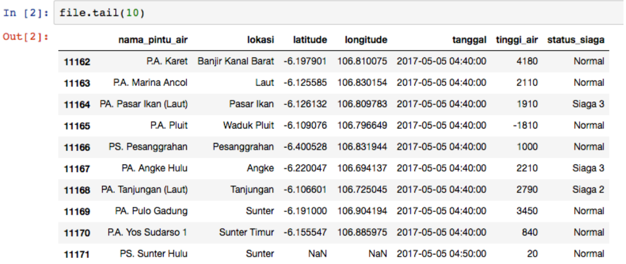


In [1]:
```
import pandas as pd
import numpy as np
import matplotlib as plt

file = pd.read_csv("./data-tinggi-muka-air-mei-2017.csv")

file.head(10)
```

Out[1]:

	nama_pintu_air	lokasi	latitude	longitude	tanggal	tinggi_air	status_siaga
0	PS. Cipinang Hulu	Cipinang Hulu	-6.374264	106.883862	1970-01-01 07:00:00	0	Normal
1	PS. Marunda	Banjir Kanal Timur	-6.108719	106.969067	1970-01-01 07:00:00	0	Normal
2	PS. Cipinang Hulu	Cipinang Hulu	-6.374264	106.883862	1970-01-01 07:00:00	0	Normal
3	PS. Marunda	Banjir Kanal Timur	-6.108719	106.969067	1970-01-01 07:00:00	0	Normal
4	PS. Cipinang Hulu	Cipinang Hulu	-6.374264	106.883862	1970-01-01 07:00:00	0	Normal
5	PS. Marunda	Banjir Kanal Timur	-6.108719	106.969067	1970-01-01 07:00:00	0	Normal
6	PS. Cipinang Hulu	Cipinang Hulu	-6.374264	106.883862	1970-01-01 07:00:00	0	Normal
7	PS. Marunda	Banjir Kanal Timur	-6.108719	106.969067	1970-01-01 07:00:00	0	Normal
8	PS. Cipinang Hulu	Cipinang Hulu	-6.374264	106.883862	1970-01-01 07:00:00	0	Normal
9	PS. Marunda	Banjir Kanal Timur	-6.108719	106.969067	1970-01-01 07:00:00	0	Normal

It is still in 1970 and water level is 0, so we try to see the last ten rows of the data.

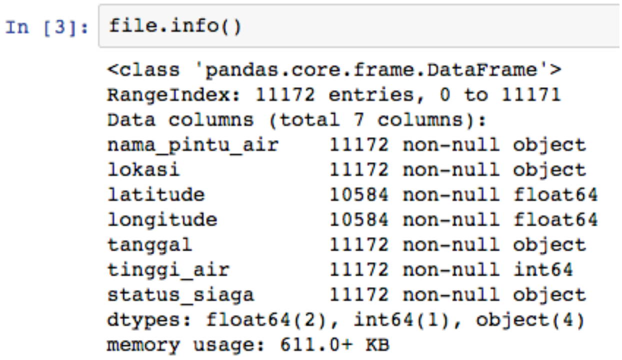


In [2]:
```
file.tail(10)
```

Out[2]:

	nama_pintu_air	lokasi	latitude	longitude	tanggal	tinggi_air	status_siaga
11162	P.A. Karet	Banjir Kanal Barat	-6.197901	106.810075	2017-05-05 04:40:00	4180	Normal
11163	P.A. Marina Ancol	Laut	-6.125585	106.830154	2017-05-05 04:40:00	2110	Normal
11164	PA. Pasar Ikan (Laut)	Pasar Ikan	-6.126132	106.809783	2017-05-05 04:40:00	1910	Siaga 3
11165	P.A. Pluit	Waduk Pluit	-6.109076	106.796649	2017-05-05 04:40:00	-1810	Normal
11166	PS. Pesanggrahan	Pesanggrahan	-6.400528	106.831944	2017-05-05 04:40:00	1000	Normal
11167	PA. Angke Hulu	Angke	-6.220047	106.694137	2017-05-05 04:40:00	2210	Siaga 3
11168	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-05 04:40:00	2790	Siaga 2
11169	PA. Pulo Gadung	Sunter	-6.191000	106.904194	2017-05-05 04:40:00	3450	Normal
11170	P.A. Yos Sudarso 1	Sunter Timur	-6.155547	106.885975	2017-05-05 04:40:00	840	Normal
11171	PS. Sunter Hulu	Sunter	NaN	NaN	2017-05-05 04:50:00	20	Normal

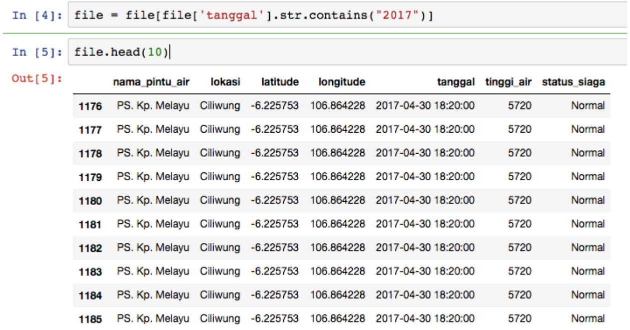
We see that those are the updated data by which the date is on 5 May 2017. Beforehand, we will see the information of the dataset.



In [3]:
```
file.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11172 entries, 0 to 11171
Data columns (total 7 columns):
nama_pintu_air      11172 non-null object
lokasi              11172 non-null object
latitude            10584 non-null float64
longitude           10584 non-null float64
tanggal             11172 non-null object
tinggi_air          11172 non-null int64
status_siaga        11172 non-null object
dtypes: float64(2), int64(1), object(4)
memory usage: 611.0+ KB
```

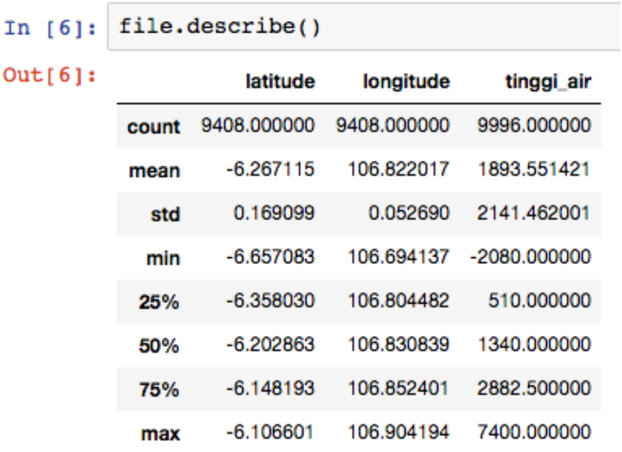
Basically, it contains about 11000+ rows, and the type of data are object, float, and integer. We then try to see the updated data which are in 2017.



```
In [4]: file = file[file['tanggal'].str.contains("2017")]

In [5]: file.head(10)
Out[5]:
```

	nama_pintu_air	lokasi	latitude	longitude	tanggal	tinggi_air	status_siaga
1176	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1177	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1178	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1179	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1180	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1181	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1182	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1183	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1184	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal
1185	PS. Kp. Melayu	Ciliwung	-6.225753	106.864228	2017-04-30 18:20:00	5720	Normal

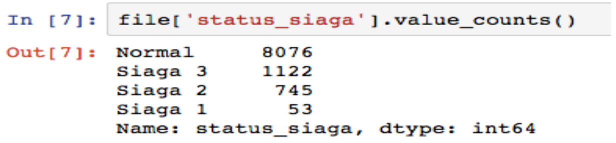
It removes about 1000 rows of data whose year is not in 2017. Then we would like to see the summary of numerical variables, which is the population distribution of the data.



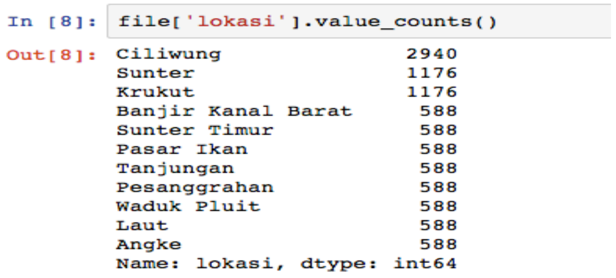
```
In [6]: file.describe()
Out[6]:
```

	latitude	longitude	tinggi_air
count	9408.000000	9408.000000	9996.000000
mean	-6.267115	106.822017	1893.551421
std	0.169099	0.052690	2141.462001
min	-6.657083	106.694137	-2080.000000
25%	-6.358030	106.804482	510.000000
50%	-6.202863	106.830839	1340.000000
75%	-6.148193	106.852401	2882.500000
max	-6.106601	106.904194	7400.000000

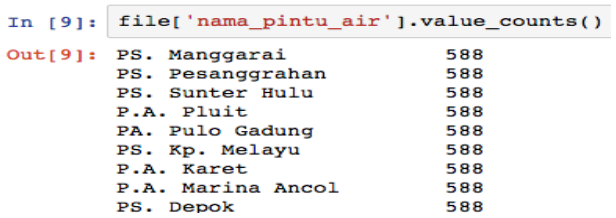
Recall the basic statistic knowledge, we could see the average, quartiles, min value, max value, variance, and standard deviation. We could get some inferences for the water level. It has a high standard deviation, means the inequality of water table in Jakarta. The min and max is also too far, so we could know there will be some different level status even though water level in particular area is surely disparate. Then, we would like to see the summary for non-numerical values.



```
In [7]: file['status_siaga'].value_counts()
Out[7]: Normal      8076
        Siaga 3     1122
        Siaga 2      745
        Siaga 1       53
        Name: status_siaga, dtype: int64
```

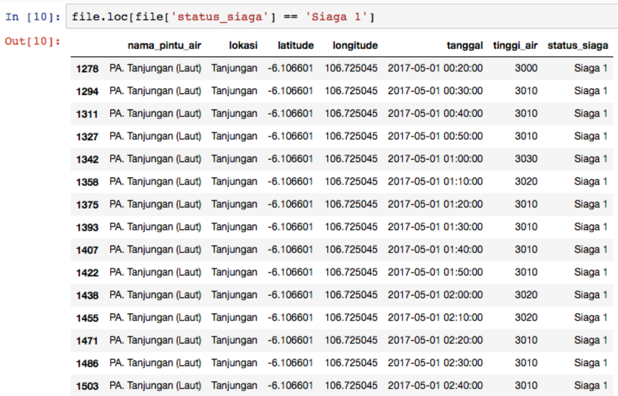


```
In [8]: file['lokasi'].value_counts()
Out[8]: Ciliwung             2940
        Sunter               1176
        Krukut               1176
        Banjir Kanal Barat    588
        Sunter Timur          588
        Pasar Ikan            588
        Tanjungan             588
        Pesanggrahan          588
        Waduk Pluit           588
        Laut                  588
        Angke                 588
        Name: lokasi, dtype: int64
```



```
In [9]: file['nama_pintu_air'].value_counts()
Out[9]: PS. Manggarai              588
        PS. Pesanggrahan           588
        PS. Sunter Hulu            588
        P.A. Pluit                 588
        PA. Pulo Gadung            588
        PS. Kp. Melayu             588
        P.A. Karet                 588
        P.A. Marina Ancol          588
        PS. Depok                  588
```

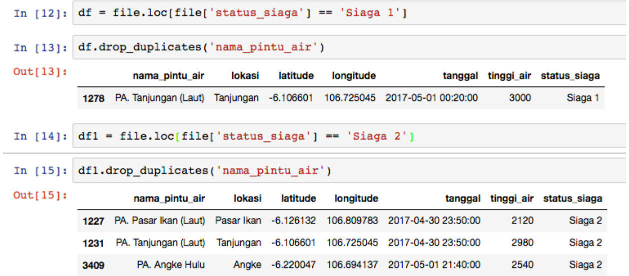
We then would like to see which floodgate that is in *siaga 1* status.



```
In [10]: file.loc[file['status_siaga'] == 'Siaga 1']
Out[10]:
```

	nama_pintu_air	lokasi	latitude	longitude	tanggal	tinggi_air	status_siaga
1278	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 00:20:00	3000	Siaga 1
1294	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 00:30:00	3010	Siaga 1
1311	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 00:40:00	3010	Siaga 1
1327	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 00:50:00	3010	Siaga 1
1342	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 01:00:00	3030	Siaga 1
1358	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 01:10:00	3020	Siaga 1
1375	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 01:20:00	3010	Siaga 1
1393	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 01:30:00	3010	Siaga 1
1407	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 01:40:00	3010	Siaga 1
1422	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 01:50:00	3010	Siaga 1
1438	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 02:00:00	3020	Siaga 1
1455	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 02:10:00	3020	Siaga 1
1471	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 02:20:00	3010	Siaga 1
1486	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 02:30:00	3010	Siaga 1
1503	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 02:40:00	3010	Siaga 1

Apparently there are a lot of redundant names for floodgate. So we use drop duplicate function.



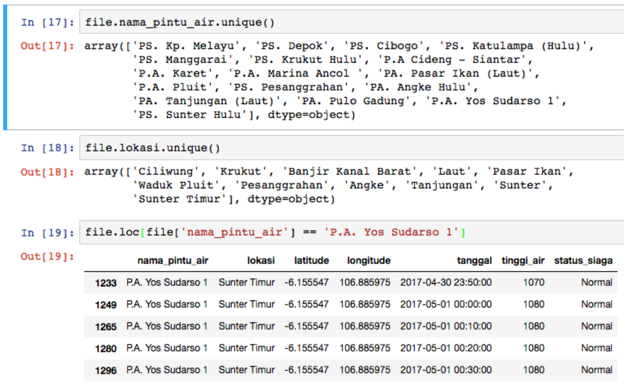
```
In [12]: df = file.loc[file['status_siaga'] == 'Siaga 1']

In [13]: df.drop_duplicates('nama_pintu_air')
Out[13]:
```

	nama_pintu_air	lokasi	latitude	longitude	tanggal	tinggi_air	status_siaga
1278	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-05-01 00:20:00	3000	Siaga 1

```
In [14]: df1 = file.loc[file['status_siaga'] == 'Siaga 2']

In [15]: df1.drop_duplicates('nama_pintu_air')
Out[15]:
```

	nama_pintu_air	lokasi	latitude	longitude	tanggal	tinggi_air	status_siaga
1227	PA. Pasar Ikan (Laut)	Pasar Ikan	-6.126132	106.809783	2017-04-30 23:50:00	2120	Siaga 2
1231	PA. Tanjungan (Laut)	Tanjungan	-6.106601	106.725045	2017-04-30 23:50:00	2980	Siaga 2
3409	PA. Angke Hulu	Angke	-6.220047	106.694137	2017-05-01 21:40:00	2540	Siaga 2

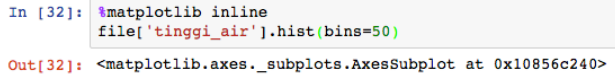
We then would like to see in which status of the nearby floodgate. Then we see the floodgate name or the location, and select it.



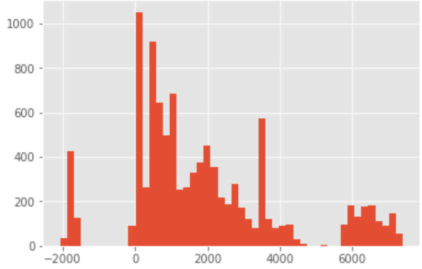
```
In [17]: file.nama_pintu_air.unique()
Out[17]: array(['PS. Kp. Melayu', 'PS. Depok', 'PS. Cibogo', 'PS. Katulampa (Hulu)',
                'PS. Manggarai', 'PS. Krukut Hulu', 'P.A Cideng - Siantar',
                'P.A. Karet', 'P.A. Marina Ancol ', 'PA. Pasar Ikan (Laut)',
                'P.A. Pluit', 'PS. Pesanggrahan', 'PA. Angke Hulu',
                'PA. Tanjungan (Laut)', 'PA. Pulo Gadung', 'P.A. Yos Sudarso 1',
                'PS. Sunter Hulu'], dtype=object)

In [18]: file.lokasi.unique()
Out[18]: array(['Ciliwung', 'Krukut', 'Banjir Kanal Barat', 'Laut', 'Pasar Ikan',
                'Waduk Pluit', 'Pesanggrahan', 'Angke', 'Tanjungan', 'Sunter',
                'Sunter Timur'], dtype=object)

In [19]: file.loc[file['nama_pintu_air'] == 'P.A. Yos Sudarso 1']
Out[19]:
```

	nama_pintu_air	lokasi	latitude	longitude	tanggal	tinggi_air	status_siaga
1233	P.A. Yos Sudarso 1	Sunter Timur	-6.155547	106.885975	2017-04-30 23:50:00	1070	Normal
1249	P.A. Yos Sudarso 1	Sunter Timur	-6.155547	106.885975	2017-05-01 00:00:00	1080	Normal
1265	P.A. Yos Sudarso 1	Sunter Timur	-6.155547	106.885975	2017-05-01 00:10:00	1080	Normal
1280	P.A. Yos Sudarso 1	Sunter Timur	-6.155547	106.885975	2017-05-01 00:20:00	1080	Normal
1296	P.A. Yos Sudarso 1	Sunter Timur	-6.155547	106.885975	2017-05-01 00:30:00	1080	Normal

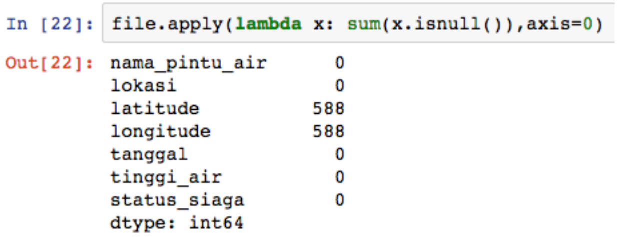
Thankfully the closest floodgate is still in normal status. Then we would like to see the histogram of the water level.



```
In [32]: %matplotlib inline
         file['tinggi_air'].hist(bins=50)

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x10856c240>
```

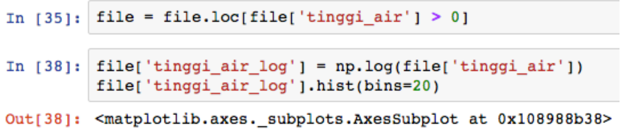


We'd like to see if there are missing values in the variables because most of models don't work with missing data.

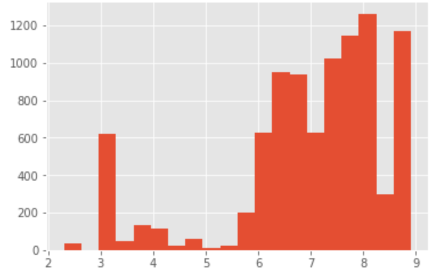


```
In [22]: file.apply(lambda x: sum(x.isnull()),axis=0)

Out[22]: nama_pintu_air      0
         lokasi              0
         latitude          588
         longitude         588
         tanggal             0
         tinggi_air          0
         status_siaga        0
         dtype: int64
```

There are only latitude and longitude whose values are still missing. It is an option if we'd like to assign the value. But because we won't use it, so it is not needed to set the value. We see the water level histogram has some extreme values, so we could treat it using logarithm.



```
In [35]: file = file.loc[file['tinggi_air'] > 0]

In [38]: file['tinggi_air_log'] = np.log(file['tinggi_air'])
         file['tinggi_air_log'].hist(bins=20)

Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x108988b38>
```



We could see even though the distribution is still not close to normal, but extreme values has been vanished. Last, building a predictive model is not needed due to dataset is still simple and doesn't contain many variables.

IV. Conclusion

Jakarta Open Data was born as the aspiration of DKI Jakarta's Government to provide an accurate, open, centralized, and integrated development database, in accordance with the mandate of the governor's regulation in Peraturan Gubernur Provinsi DKI Jakarta Nomor 181 Tahun 2014. Open Data is nothing if we could not utilize and take the advantage of those data accurately. To learn some basic data analysis is very useful as we could use the data to create new innovation and give feedback to the government for better Jakarta in the future. By always practicing to analyse data, it's expected for us to make sense of the world of data and increase each of our creativities when analysing data. Even though the author's analysis is still in fundamental level as she just started in days, the most important thing is that we could make use of those dataset provided. Based on the analysis, at least we could see the variation of water level and alert status in Jakarta. We could also make and prepare preventive actions if we see possible bad things based on data. Hopefully, dataset published in the web portal could be bigger and variative, so people will be more interested to analyse and find their own predictive model for the government so that government could make better regulation and improvement for Jakarta based on it.

V. Acknowledgement

First of all, I would like to thank to God for the abundance blessings and my parents for always being there in my life giving me the unlimited support. I would deeply express my gratitude for Mr. Rinaldi Munir, Mrs. Ayu Purwarianti, and Mrs. Dessi Puji Lestari whose knowledge, lesson, guide, and advice. I would also thank to my friends for cheering me up and taking care of me during my study here. Last, I would thank to my campus, Institut Teknologi Bandung, for providing a good place and pleasant condition for me to study and lecturers to conduct lessons. I thank you for all you have done.

References

[1] Data Analysis. Retrieved from: https://ori.hhs.gov/education/products/n_illinois_u/datamanagement/datopic.html on 5 May 2017
[2] "Data science and prediction". Communications of the ACM. 56 (12): 64. doi:10.1145/2500499.
[3] *Application of Linear Algebra. Jeff Leek (2013-12-12). "The keyword in "Data Science" is not Data, it is Science". Simply Statistics*
[4] "What is the Water Table?". imnh.isu.edu. on 5 May 2017
[5] NumPy. Retrieved from http://www.numpy.org/ on 3 May 2017
[6] Matplotlib. Retrieved from https://matplotlib.org/ on 3 May 2017
[7] Pandas. Retrieved from http://pandas.pydata.org/ on 3 May 2017
[8] Tutorial Python. Retrieved from https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/ on 2 May 2017
[9] Intro to Python. Retrieved from https://www.datacamp.com/courses/intro-to-python-for-data-science on 2 May 2017
[10] Jakarta Open Data. Retrieved from http://data.jakarta.go.id/ on 5 May 2017

Additional Notes

Implemented source code and the dataset used in this paper could be downloaded in this link: https://drive.google.com/open?id=0B_joVASCFnRxVmJlamszdFo1a28

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 5 Mei 2017

Dharma Kurnia Septialoka - 13514028