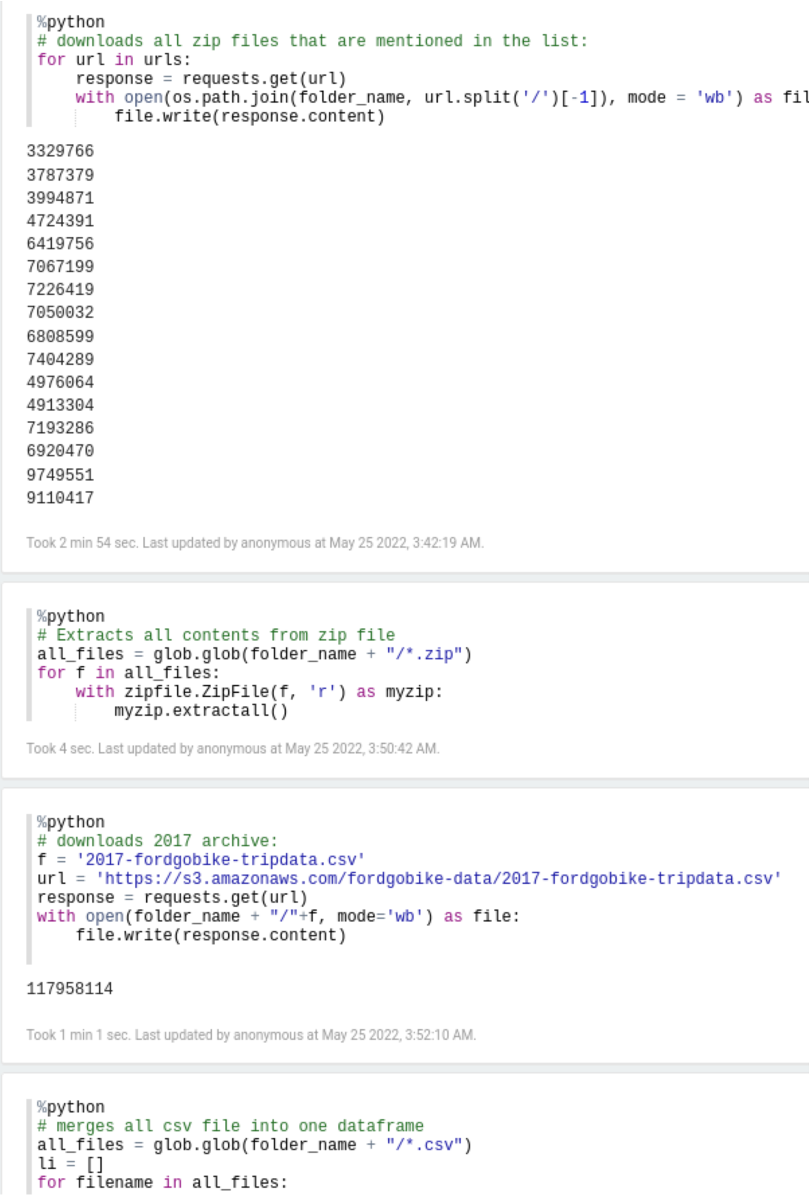
Data Analysis

a) Questions

The first step involves asking questions which are the basis for the following data analysis. The data research can be made by including your questions, too.

b) Download

It's better, to handle the download of all the related files, e.g. csv, programmatically. As soon as new files exist, e.g. the latest files containing all bike rides for one month, you can run the download script again so that the data analysis will be updated.



```
%python
# downloads all zip files that are mentioned in the list:
for url in urls:
    response = requests.get(url)
    with open(os.path.join(folder_name, url.split('/')[-1]), mode = 'wb') as fil
        file.write(response.content)
```

```
3329766
3787379
3994871
4724391
6419756
7067199
7226419
7050032
6808599
7404289
4976064
4913304
7193286
6920470
9749551
9110417
```

Took 2 min 54 sec. Last updated by anonymous at May 25 2022, 3:42:19 AM.

```
%python
# Extracts all contents from zip file
all_files = glob.glob(folder_name + "/*.zip")
for f in all_files:
    with zipfile.ZipFile(f, 'r') as myzip:
        myzip.extractall()
```

Took 4 sec. Last updated by anonymous at May 25 2022, 3:50:42 AM.

```
%python
# downloads 2017 archive:
f = '2017-fordgobike-tripdata.csv'
url = 'https://s3.amazonaws.com/fordgobike-data/2017-fordgobike-tripdata.csv'
response = requests.get(url)
with open(folder_name + "/"+f, mode='wb') as file:
    file.write(response.content)
```

```
117958114
```

Took 1 min 1 sec. Last updated by anonymous at May 25 2022, 3:52:10 AM.

```
%python
# merges all csv file into one dataframe
all_files = glob.glob(folder_name + "/*.csv")
li = []
for filename in all_files:
```

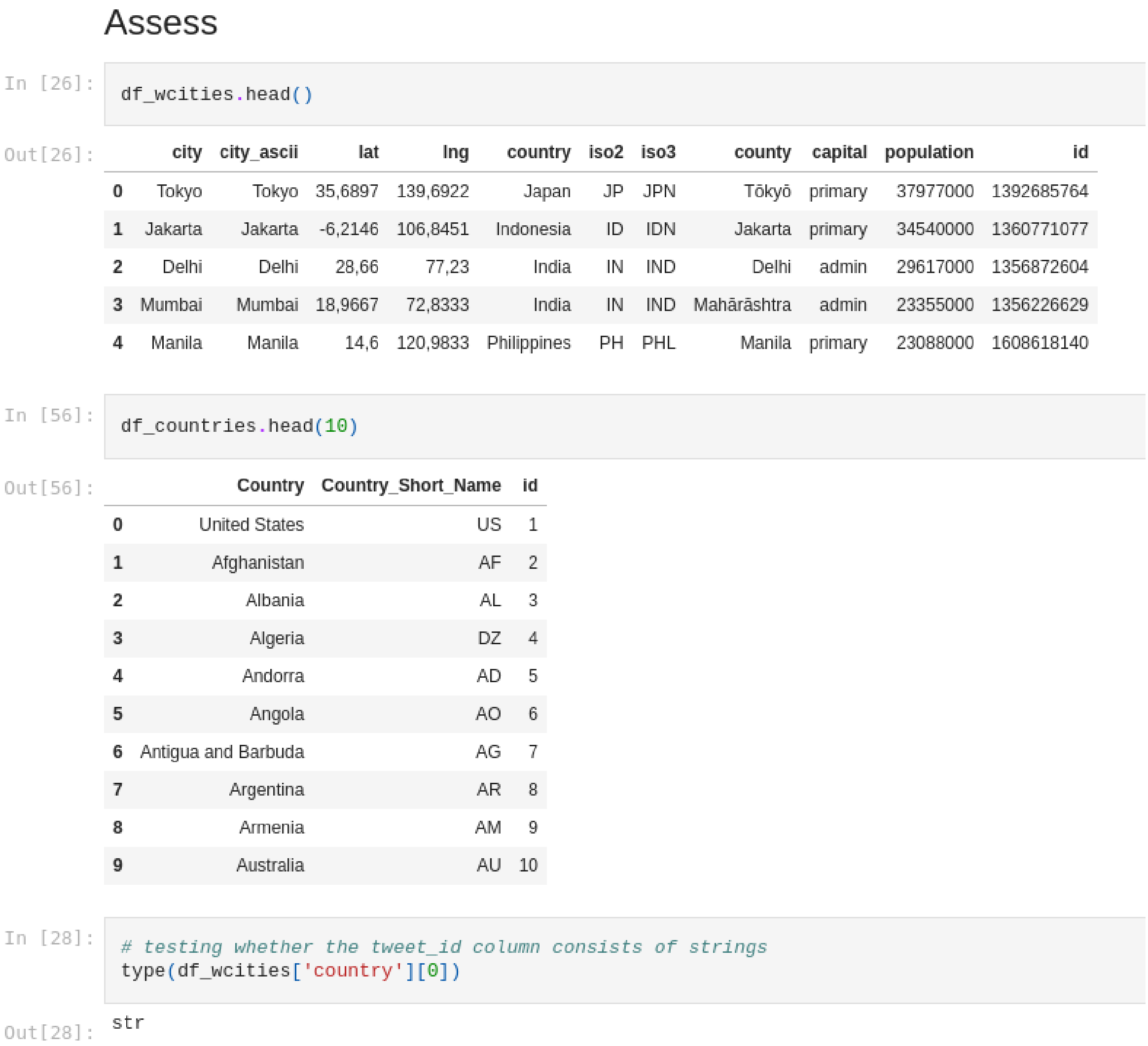
Figure I: Downloading, unzipping and merging the files into one dataframe.

c) Assessing and Cleaning

However, the datasets are not always tidy. Consequently, I may need to tidy up your dataset first before the actual data analysis.

When I start looking at a dataset, I view it by scrolling down using either **Jupyter Notebook** or **Apache Zeppelin**. This is somehow similar to looking through the dataset in your spreadsheet program. However, when it comes to data analysis I'd rather use more suitable software such as Jupyter Notebook or Apache Zeppelin. During the data analysis the file will be saved in the corresponding notebook file which I can export later into Html, PDF or simply save the final version into a proper notebook file.

After the visual assessment, the programmatic assessment follows. For this step, I stick to built-in functions provided by the **pandas** dataframe. With these tools, I can among others find out the number of rows in a dataset, the names of the columns together with their types, etc.



Assess

```
In [26]: df_wcities.head()
```

Out[26]:

	city	city_ascii	lat	lng	country	iso2	iso3	county	capital	population	id
0	Tokyo	Tokyo	35,6897	139,6922	Japan	JP	JPN	Tōkyō	primary	37977000	1392685764
1	Jakarta	Jakarta	-6,2146	106,8451	Indonesia	ID	IDN	Jakarta	primary	34540000	1360771077
2	Delhi	Delhi	28,66	77,23	India	IN	IND	Delhi	admin	29617000	1356872604
3	Mumbai	Mumbai	18,9667	72,8333	India	IN	IND	Mahārāshtra	admin	23355000	1356226629
4	Manila	Manila	14,6	120,9833	Philippines	PH	PHL	Manila	primary	23088000	1608618140

```
In [56]: df_countries.head(10)
```

Out[56]:

	Country	Country_Short_Name	id
0	United States	US	1
1	Afghanistan	AF	2
2	Albania	AL	3
3	Algeria	DZ	4
4	Andorra	AD	5
5	Angola	AO	6
6	Antigua and Barbuda	AG	7
7	Argentina	AR	8
8	Armenia	AM	9
9	Australia	AU	10

```
In [28]: # testing whether the tweet_id column consists of strings
         type(df_wcities['country'][0])

Out[28]: str
```

Figure II: Assessing two dataframes

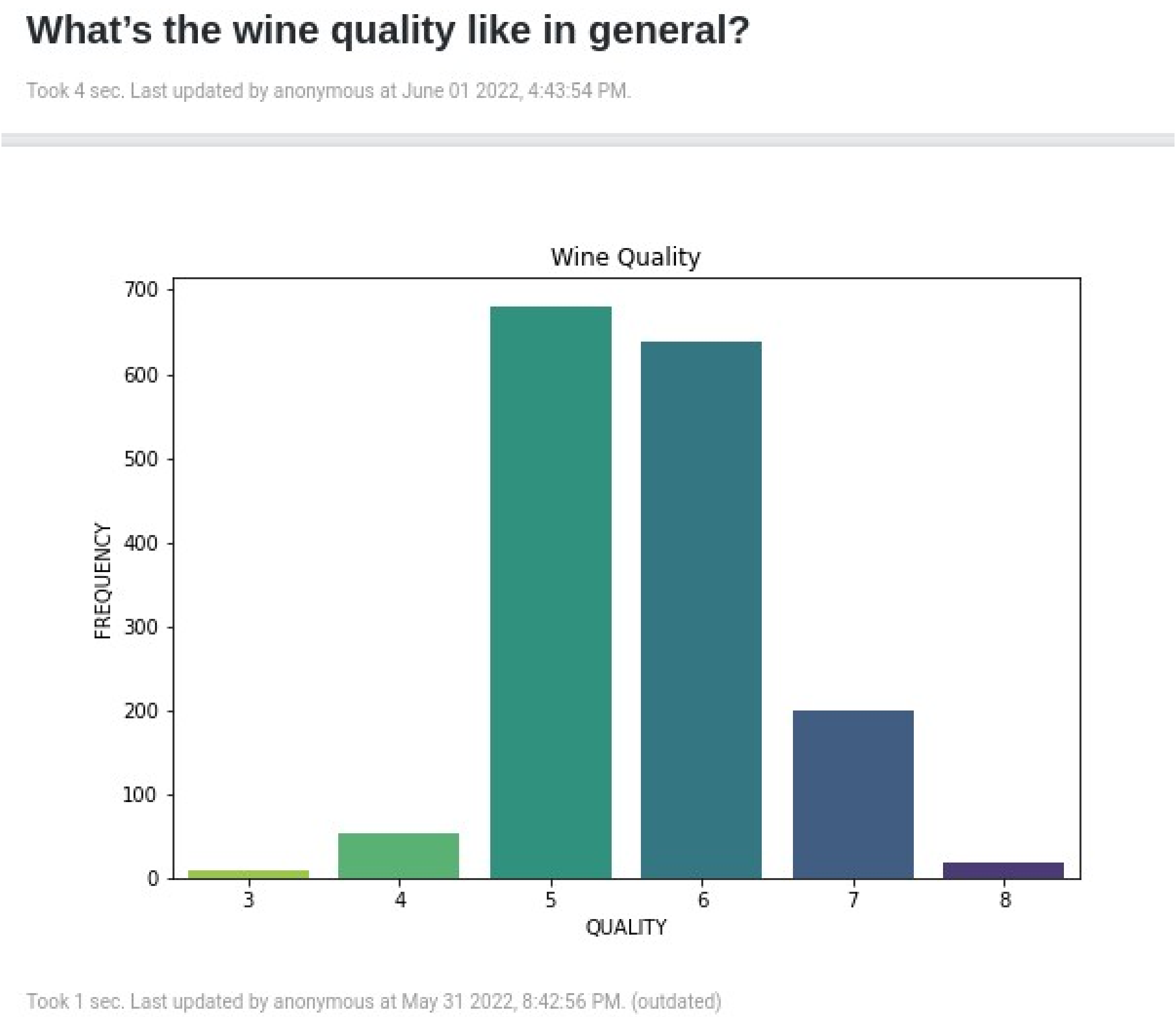
Having gathered enough information about one or several datasets, I start cleaning the dataframes and store the new versions into an extra csv file. Let's take the datasets in figure II as an example. The cities are stored in one dataframe whereas the countries are stored in another dataframe. Both datasets come from different sources that's why the country ID in the cities dataframe doesn't match with the id column of the countries dataframe. In order to import these datasets into database such as Mysql successfully it's important that the id column in the cities dataframe obtains the ids from the countries dataframe. For such a case, I usually write a function which loops over the id column in the cities dataframe and obtains the corresponding country id from the countries dataframe.

Other examples may involve monthly data each stored in an extra csv file. In these cases I usually merge the dataframes into one dataframe and store the new version into a csv file.

Other forms of untidyness are inappropriate datatypes of the columns, rows with only null values, categories stored in different columns although they all belong to the same column, etc.

d) Exploratory Data Analysis

During the exploratory part the questions get answered. The results of my data research are supported by diagrams using python's **matplotlib** and **seaborn**. The univariate exploration is all about looking at one variable of your dataset and depicting it as a bar chart, for example.



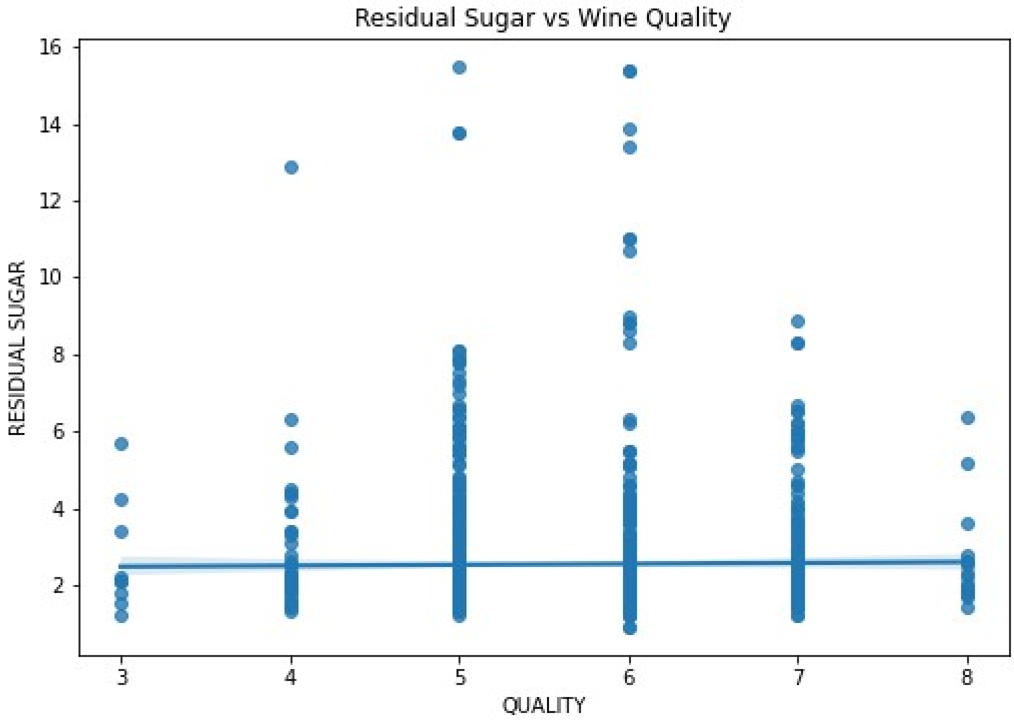
What's the wine quality like in general?

Took 4 sec. Last updated by anonymous at June 01 2022, 4:43:54 PM.

Took 1 sec. Last updated by anonymous at May 31 2022, 8:42:56 PM. (outdated)

Figure III: Univariate Exploration: average wine quality

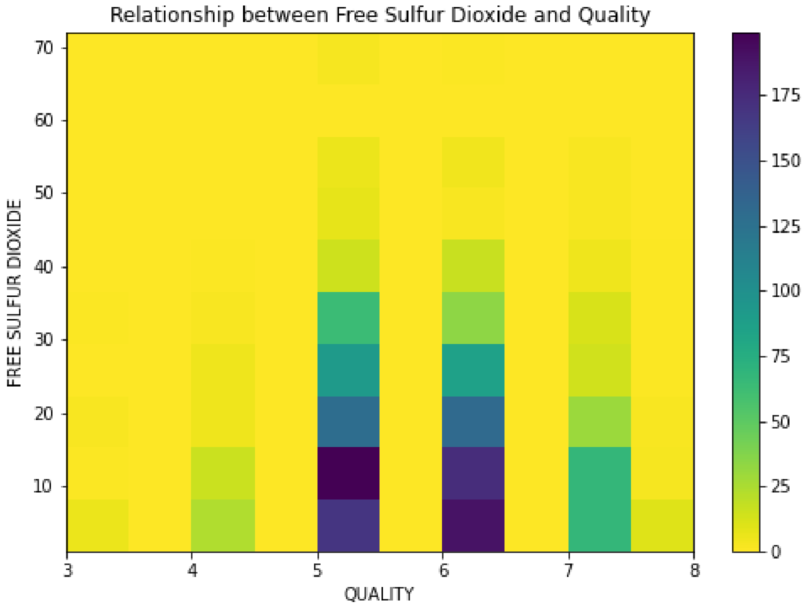
In a bivariate exploration, you'll assess how one variable influences another one. Basically you'll look at two variables and how they interact with each other. To visualize such a relationship you can make use of a regression line which shows the correlation between these two variables. A really steep regression line corresponds to a strong correlation whereas an almost horizontal line shows just a weak correlation.



Took 0 sec. Last updated by anonymous at May 26 2022, 9:27:02 PM. (outdated)

Figure IV: Regression line showing the influence of residual sugar on wine quality

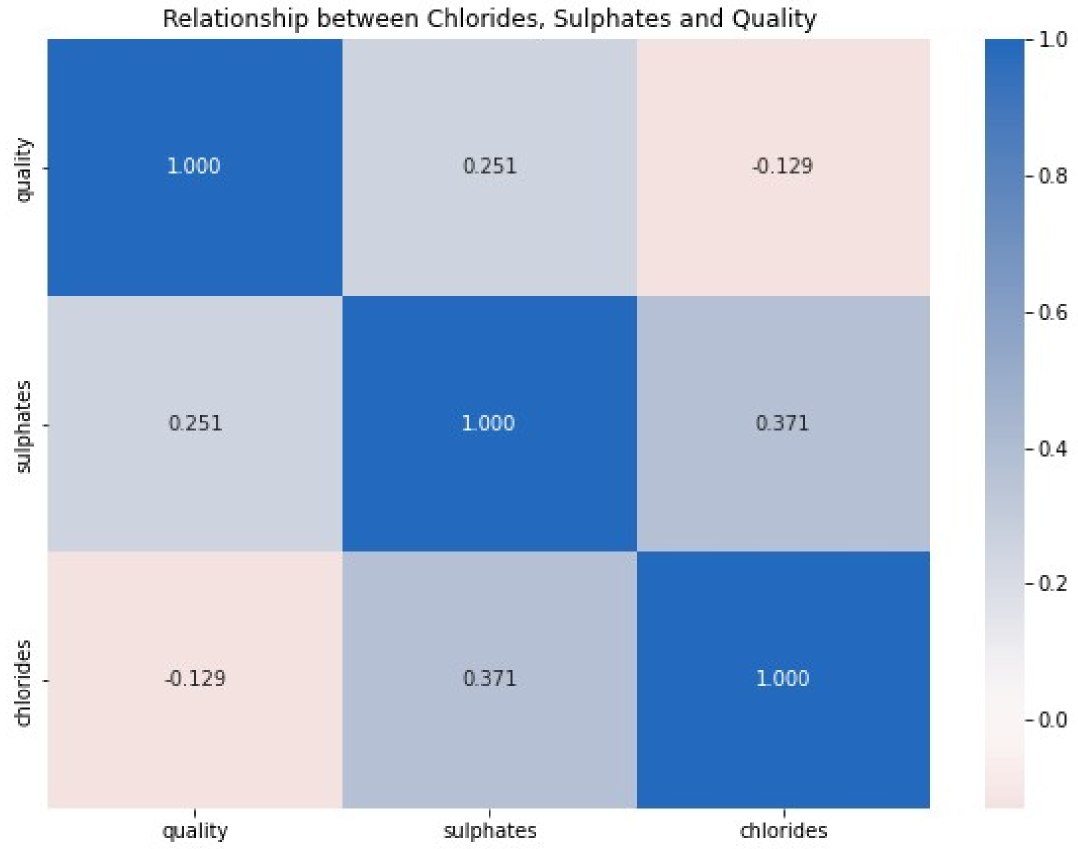
There are other forms of diagrams which are suitable for two variables such as a heatplot. A heatplot is especially useful when it comes to logarithmic scaled axis.



Took 0 sec. Last updated by anonymous at May 26 2022, 11:09:19 PM. (outdated)

Figure V: Heatplot: Average wines possess the highest amount of free SO2

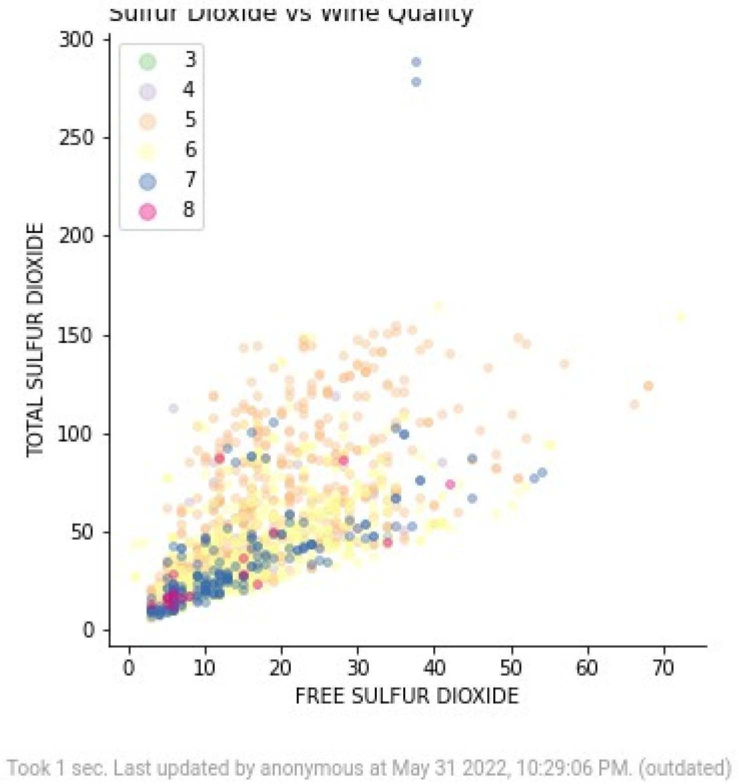
Multivariate exploration involves at least three variables. The correlation matrix is a good example which depicts the correlation between three variables. The best correlation is either 1 or -1 whereas weak correlations are below 0.5 or -0.5.



Took 1 sec. Last updated by anonymous at May 26 2022, 11:34:44 PM. (outdated)

Figure VI: there are only weak correlations between 2 different variables in this dataset.

To depict the third variable within an xy-diagram a colormap or different shapes can be used, etc. There are datasets where the values of the third variable overlap each other. That's why it's necessary to figure out a good value for opacity and the jitter value. What's more the usage of an appropriate color scheme can influence the visibility of the third variable, too.



Took 1 sec. Last updated by anonymous at May 31 2022, 10:29:06 PM. (outdated)

Figure VII: Third variable depicted with higher transparency

e) Reports

Reports can be created in both of the notebooks. Under Jupyter Notebook you can export the notebook into HTML or PDF. Additionally, you can make use of a nice feature, called a slideshow in which you present the findings to your audience. Alternatively, you can copy the best diagrams into a textprocessing software to produce PDF files.

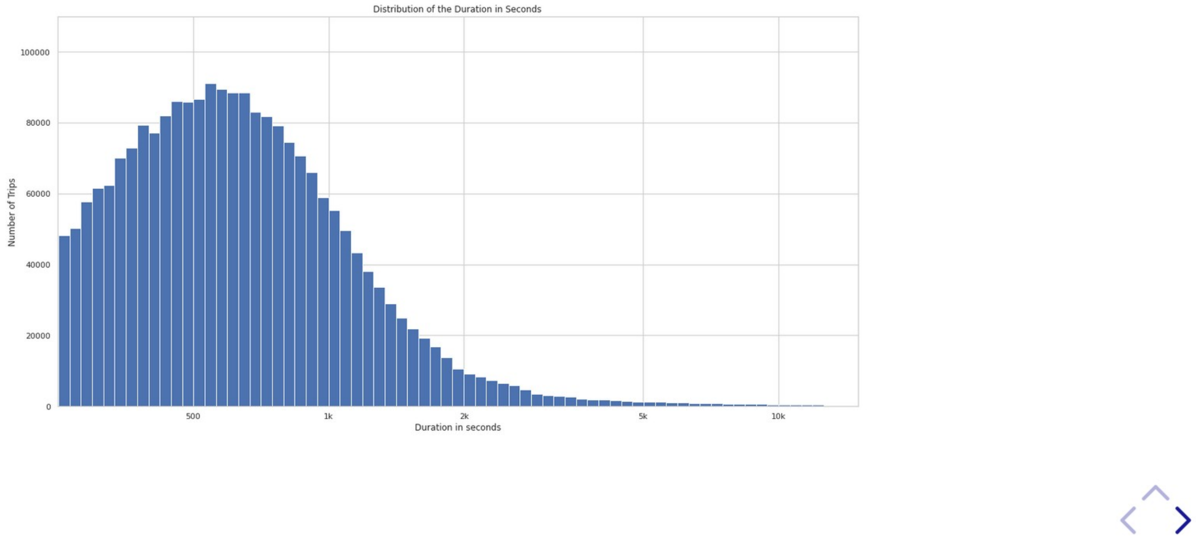
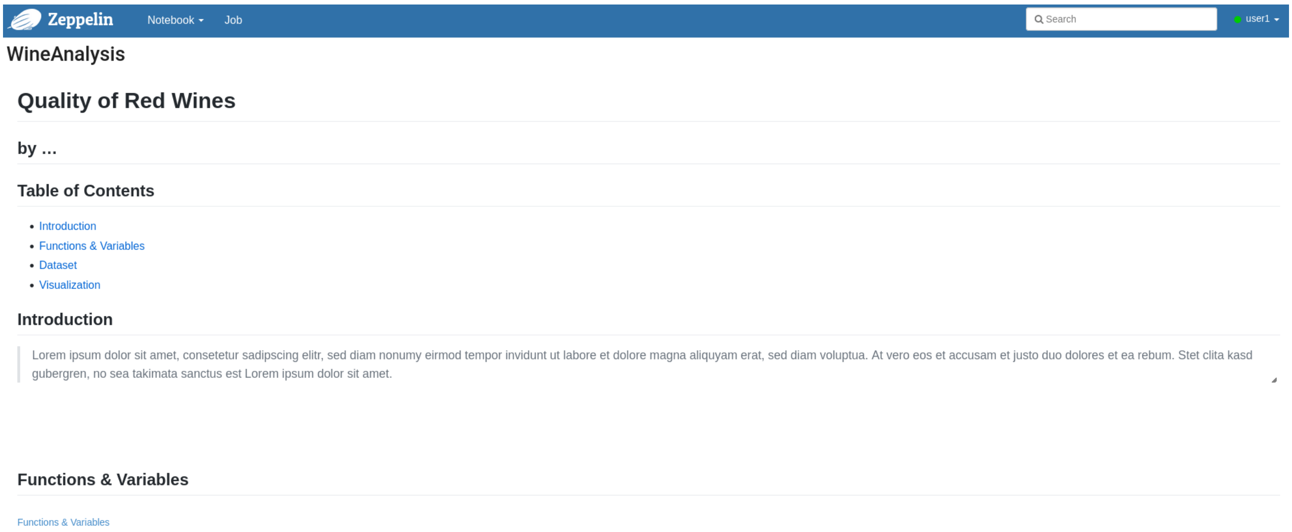


Figure VIII: The slideshow realized by Jupyter Notebook highlights the findings of the report

Using Apache Zeppelin, you can create a report based on your data analysis in which the code will be removed from the final report. This makes it easier for you or your audience to follow your report.



WineAnalysis

Quality of Red Wines

by ...

Table of Contents

- Introduction
- Functions & Variables
- Dataset
- Visualization

Introduction

> Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Functions & Variables

Functions & Variables

Figure IX: Report created with Apache Zeppelin