Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 36, S. 55-60, Mayıs 2022

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 36, pp. 55-60, May 2022

Research Article

Recording Performances of Some File Types for Pandas Data

Hakan Temiz

Artvin Coruh University, Faculty of Engineering, Department of Computer Engineering, Artvin, Turkey, (ORCID: 0000-0002-1351-7565), htemiz@artvin.edu.tr

(1st International Conference on Engineering and Applied Natural Sciences ICEANS 2022, May 10-13, 2022)

(**DOI:** 10.31590/ejosat.1103499)

ATIF/REFERENCE: Temiz, H., (2022). Recording Performances of Some File Types for Pandas Data. *European Journal of Science and Technology*, (36), 55-60.

Abstract

Scientists, researchers, engineers, etc. almost everyone who works with data crosses paths with Pandas at some point. It is so powerful library that allows for easy, rapid and efficient manipulation of data. It can convert data it represent into various file types. Among these file types, the determination of the one which records the same Pandas data with the smallest size on the disk is an important issue considering the abundance of today's data. In this study, the file types that can save Pandas data with minimum size has been experimentally investigated from various perspectives. In this respect, the CSV, HDF, JSON, Excel and Pickle file types are involved in the experiments. The sizes of these files were benchmarked under several conditions such as the completeness or lack of data and type of variables that are contained in data. In addition, it was also examined that how file sizes vary as data increases.

Keywords: pandas, data, file sizes, file types, recording performance.

Bazı Dosya Türlerinin Pandas Verisini Kaydetmedeki Performansları

Öz

Bilim insanları, araştırmacılar, mühendisler vb. verilerle çalışan hemen hemen herkesin yolu bir noktada Pandas kütüphanesi ile kesişmektedir. Pandas, verilerin kolay, hızlı ve verimli bir şekilde işlenmesine izin veren çok güçlü bir kütüphanedir. Temsil ettiği verileri çeşitli dosya türlerine dönüştürebilme kabiliyetine sahiptir. Bu dosya türleri arasından, aynı Pandas verisini diske en küçük boyutta kaydeden dosya türünün tespiti günümüz verisinin bolluğu göz önüne alındığında önemli bir konudur. Bu çalışmada, Pandas verilerini minimum boyutta kaydedebilen dosya türleri deneysel olarak çeşitli açılardan incelenmiştir. Bu doğrultuda deneylerde CSV, HDF, JSON, Excel ve Pickle dosya türleri incelemeye alınmıştır. Bu dosyaların boyutları, verilerin tamlığı veya eksikliği ile verilerde bulunan değişkenlerin türü gibi çeşitli koşullar altında karşılaştırılmıştır. Ayrıca veriler arttıkça dosya boyutlarının nasıl değiştiği de bu çalışma kapsamında incelenmiştir.

Anahtar Kelimeler: pandas, veri, dosya boyutları, dosya türleri, kayıt performansı.

1. Introduction

Data is a central part of contemporary life today. Every second, an unprecedented amount of data is generated by machines or people. Machine generated data correspond to data produced from cameras, sensors, satellites, real-time or medical monitoring devices, trackers for personal health care, and so on. Humans act as another major source of data. We create a vast amount of data by tweeting, logging, blogging, or posting messages, images and other types of contents in social media. Organizations are another spring for cooking data especially with enterprise resource management (ERP) systems.

The amount of data generated through the sources mentioned above has pushed the capabilities and capabilities of data processing tools, devices and analytics towards new technologies and techniques. Newly developed tools have gained the ability to process and/or transfer high volumes of data very quickly. Millions of connected devices collects, transfers or stores data with high accuracy and efficiency. Analytics can seamlessly process and uncover ground-breaking insights from much more complex and vast amount of data to support decision makers to make better decisions.

All these developments have increased the need for talented individuals specialized in data science and engineering. These individuals fulfil the jobs under the names data scientist or data engineer. A data scientist or engineer (as data workers) typically should have proficiency in data science and related tools and software. Python is one of the most used platforms by data workers. Numerous software and tools have already been developed on both platforms and broadly used for data representation, manipulation and recording (Abeykoon et al., 2020),(Van Rossum & Drake, 2003). Both provide very rich sets of libraries for working on data. In Python side, the renowned library, Pandas (Reback et al., 2020),(Hoyer & Hamman, 2017) is the first choice for data.

Pandas is a member of SciPy (Virtanen et al., 2020) library that is an open source software package tailored for scientific computing in Python. Pandas allows for a fast, easy, powerful and flexible way of data analysis and manipulation for multi-dimensional data. Thanks to its simplicity, rapidity, flexibility, and many other features, it has managed to become one of the most preferred and widely used tools. Overwhelming majority of data scientists, researchers and engineers prefer to use it in their works.

Pandas can easily convert data into various file types for storage and exchange. Since today's data sizes are very high, the space occupied by the same data on the recording media needs to be minimal in terms of storage costs, efficiency and speed. Therefore, we should find a way to store data in files with minimal space. The smaller the file size, the smaller the disk space it will occupy and the faster and less costly it will be able to transfer it through the network to other computers and devices.

In this study, various file types that are the pandas can record its data are compared in terms of occupied sizes on a storage device. The experiments have been performed for well-known five distinct file types, as per two separate datasets: small-size and large-size. The change in the file size in respect to the amount of the data was also examined by these two distinct size of data. The occupied file sizes are measured for Pandas data composed of a certain variable type and mixture of these. This study clearly revealed which file type is more suitable for certain conditions.

2. File Types

In essential, Pandas can easily write to or read from a variety of file types. It accompanies a variety of methods to convert its data into or to read from these files. Amongst these, only most common types of file formats are introduced in the experiments. These file types are Comma Separated Values file (CSV), Hierarchical Data Format (HDF) (Fortner, 1998), JavaScript Object Notation (JSON) (Pezoa, Reutter, Suarez, Ugarte, & Vrgoč, 2016) file, Microsoft Excel file, and Python's preferred serialization library, Pickle (Van Rossum, 2020). The storage formats of the files are given in Table 1. Next, a very brief information is provided about them.

Table 1. Storage Formats of files.

Storage Format	*CSV*	*JSON*	*Excel*	*HDF*	*Pickle*
Plain text	✓	✓			
Binary			✓	✓	✓

CSV is an ordinary plain text file where each line preserves a record of data whose, in default, fields are separated by commas. It is very prevalently used for storing tabular data thanks to its convenience in reading and writing. Other than commas, several delimiter characters, such as space, semicolons etc., are also used for separating values.

Excel is one of Microsoft products for Office Suite, which enables to efficiently work on spreadsheets. Excel provides a very reach of features and functions for computation, data manipulation and analysis, data exchanging, statistical analyzing, engineering, financial, plotting, etc.

JSON is another plain-text file format that enables computers to parse and generate data. It is very easy also for humans to read and write since it stores data as plain text. Its language-independent feature makes it a versatile data-interchange format. JSON, structures data in collections of name-value pairs or ordered list of values.

HDF is designed by The HDF Group, a non-profit corporation, to organize and store large amounts of data. HDF is supported by many programming languages and software. It uses B-trees for indexing the data which makes it faster than a SQL database to access stored data that is stored as arrays in binary format. The compression level of HDF file is set to 9 in all experiments in this study.

Pickle is designed to serialize Python objects as byte streams in binary files or in bytes-like objects. Pickle is a very widely used Python library to represent and store data. The serialization process whereby a Python object is transformed into a byte stream is called as "pickling", and the inverse operation is called as "unpickling".

3. Method

Pandas allows for storing a rich set of variable types in the same data set. In Pandas, data is often contained through DataFrame objects. A typical DataFrame represents a tabular form of data with certain rows and columns. Normally, every column comprise of a single type variable, such as integer, real number, text or categorical. In this context, a DataFrame can comprise of columns each having the same variable type, or possibly any combination of these. Surely, file sizes would vary with the variety of data types that make up a data set. Therefore, it is important to examine the change in file sizes relative to data constituted of various data types. In this respect, A DataFrame is created with random entries for each of the Integer, String, Float and Categorical variable types, and additional DataFrame for the mixture of these. The total size of each individual file type as which these DataFrames are saved on disk is then measured.

The randint() and random() functions under the random module of the numpy library were used for generating the DataFrames of integer and float variables, respectively. The uuid4() method from the uuid library was used to randomly fill in the DataFrames of string and categorical variables.

In reality, data persist in the form of a mixture of different types of variables rather than a single type. Therefore, it is much more meaningful to examine the sizes of particular file types for such data sets. To observe required space to store such data for certain file types, an additional dataset consisting of an equal mix of different data types was included in the experiments as well. Eventually, 5 distinct data sets were established for experiments.

Additionally, in order to observe how file sizes vary according to data size, these DataFrames were created in total of two different number of records. The former and latter consist of 1000 and 10.000 records, respectively. So, the second data is ten times the first. The number of columns remains the same in both versions. The DataFrames comprised of a single variable type only have 10 columns, while the DataFrame with compounded variables has 40. The details of two data sets are given in Table 2.

Table 2. Details of Data Sets.

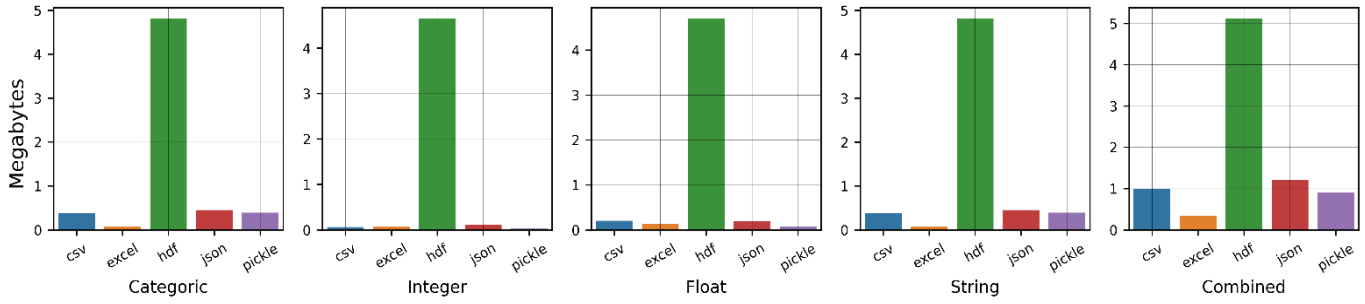
Data Size	#	Variable Types: Integer	Float	String	Categorical	Combined
Small	Rows	1.000				
	Columns	10				40
Large	Rows	10.000				
	Columns	10				40

4. Results

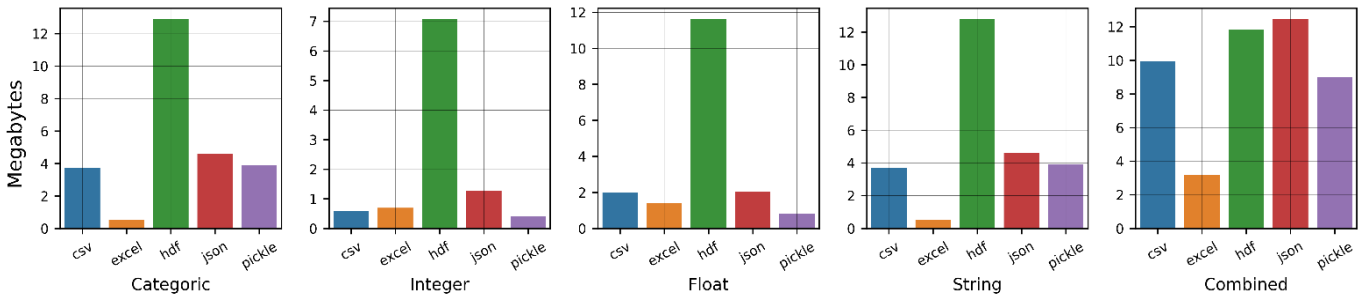
The whole experiment was performed on two alternatives of the data sets: on the complete and 20% missing data. The following sections describe the results of the experiments for both cases.

4.1. Complete Data Case

In this case, data set was complete. The results are shown in Fig 1. Subfigure (a) graphically present the file sizes for small-size data of particular variable(s), whereas (b) for large-size data.



(a) Small-size data (one thousand records without missing entries)



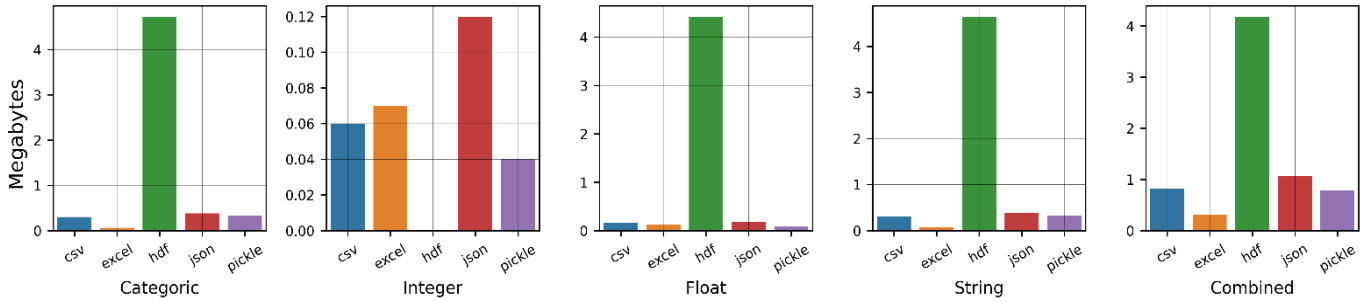
(b) Large-size data (ten thousand records without missing entries)

Fig. 1. Sizes on disk of some file types recorded for the same complete Pandas data consisting of certain variable type(s). The amount of records in large-size data (b) is ten times the amount of record in small-size data (a).

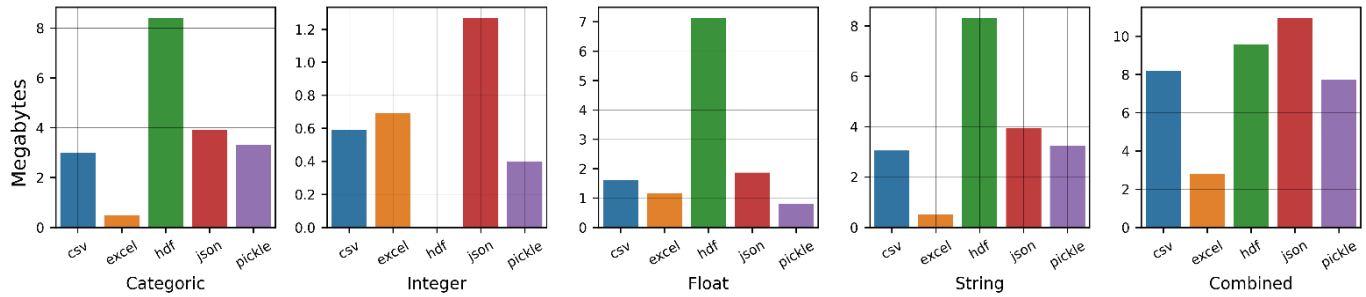
For small-size data sets consisting of other variable types than integer and float, Pandas achieved the minimum file size when saving them as Excel file type. It stores data consisting of categorical, string and float variable types very efficiently. For data of integer and float variables, Pickle file type offers the smallest sizes. On the other hand, HDF file occupies the largest space for any data of any variable type. For large-size data sets, the ratio between the different file sizes is quite similar to those of small-size data sets. The only difference is that the file sizes have increased depending on the number of records. The results from mixed data are much more important, as a typical data today usually comprise of multiple different types of variables. The both figures tell us that the file size is still minimum when Pandas saves the combined data as an Excel file. The second best smallest file size is achieved when it stores the data as a Pickle file. The HDF file type still occupies the largest size on disk.

4.2. Missing Data Case

It is very likely that data present a considerable amount of missing entries. Although missing data is not desired, it is a frequently encountered situation. Hence, it is also very important to observe how the file sizes vary in presence of missing data. For this purpose, the same experiment repeated again but this time with missing data. In order to simulate the real-world, the lost data rate were chosen as 20%, which can be considered moderate. The results are shown in Fig. 2. Subfigure (a) graphically present the file sizes for small-size data of a particular variable(s), whereas (b) for large-size data.



(a) Small-size data (one thousand records with 20% missing entries)



(b) Large-size data (ten thousand records with 20% missing entries)

Fig. 2. Size on disk of some file types recorded for the same Pandas data with 20% missing entries, consisting of certain variable type(s). The amount of records in large-size data (b) is ten times the amount of record in small-size data (a)

In this experiment, the files are reduced in size by a relatively small amount as the datasets lack 20% of data. This change in the file size is best seen in CSV file. Its size dramatically dropped compared to the others. When we examine the file sizes for both size data sets comprised of combined variables, CSV file size is approximately 10MB for the complete data, while it drops to around 8MB given the 20% lack of data. The drop in sizes is clearly observed in the other file types as well, though relatively small. Another observation is that the order of file sizes does not change regardless of whether the data is complete or incomplete. An interesting finding is that although the CSV file stores data in plain text format, the size of the integer data type is smaller than the Excel file in both cases, data is complete or 20% missing.

4.3. Increase Rate in File Sizes

This section presents and discusses the changes in file sizes when the amount of data is increased by 10 times.

Fig. 3 presents the rate of increase in file sizes for both versions (complete and missing) of data. The lowness of the bars means that although the amount of data increases at the same rate, less file size is obtained.

It can easily be said that, generally, the file sizes increased at almost the same rate as the amount of data increased. Most file formats show a linear relationship between data increment and file size, except the HDF. Additionally, except for the HDF file, missing data does not cause a significant change in file sizes. There has been a fairly small amount of fluctuations in most cases.

On the other hand, the size of the HDF file increased more or less at a much lower rate than the increase in data. The HDF file have increased in size by 2 to 3 times, while the size of other files has increased by 8 to 10 times. However, it is clearly seen from the figure that the HDF file is very sensitive to missing data. The ratio of HDF file for missing data comprised of integer variables cannot be measured since Pandas does not allow to store such DataFrame object. However, its size remains almost the same for combined data. For other data types, its size decreases in moderate amount in the presence of missing data.

Amongst the data types, the HDF file is best when saving integer data, and then combined data in the complete case. On the other hand, it is best when recording float data, and then categorical data in the missing case.

The second smallest inflation rate in size has been seen for the Excel file. Its size has increased less than the increment of the data when Pandas data contains categorical or string variables. For the both missing and complete data cases, there is hardly change in its size for integer data. Its size would remain nearly the same. Much more interesting outcome is seen for string data. The increase rate in file size is much higher for the missing data case than the size of complete case.

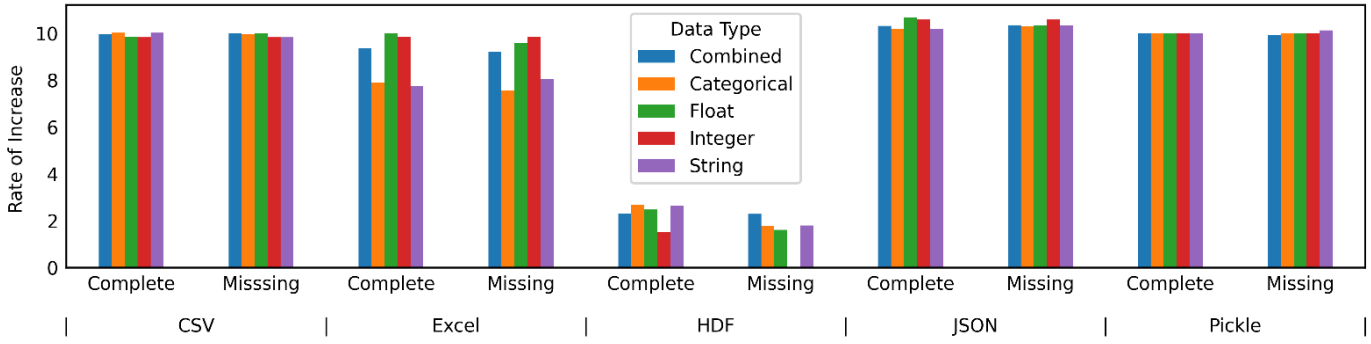


Fig. 3. Rate of increase in file sizes for the complete data and incomplete data with 20% missing entries when data increases by 10 times.

The pickle file's bars rich exactly the same high for all data types in the complete case. Which means that the pickle file stores data in such a way that the file size increase linearly. On the other hand, the same thing cannot be said for the missing case. The file sizes inflate nonlinearly in the missing case. Even much higher compared to the complete case. This dramatic increase can be seen much clearly from the bar of string data. This obviously indicates that the pickle cannot store string data as much efficiently as it records the other types. On the contrary, it is much more successful at saving combined data in the presence of missing data than saving other types of data.

JSON file ensures its minimum size for categorical and string data in the complete case when the amount of data increased. The worst performance is achieved for data of float type. This consequence is understandable considering the fact that float data accompany varying number of fractions. Contrary, for the missing data case, the highest inflate in the file size occurs integer data. This is a very odd outcome because normally float data is expected to have the highest increase compared to the others. There are very slight differences between the increase rates of the other data types. However, the categorical data has achieved the lowest increase rate for the missing data.

As for the CSV file, it is typically expected to have the lowest rate for the categorical or string data types. But the bars on the figure say the opposite. The lowest increase rate was obtained from integer and float data in the complete data case. However, in the missing data case, the integer and string data provided the lowest inflation in the file sizes, whereas the categorical and float data delivered the highest rates.

5. Discussion

When we analyzed the file sizes only and only in terms of the increase rate, it was observed that Excel was more successful than other files, that is, the file size increased more slowly than the amount of data increase. In terms of increase rate, there are no significant differences between other files, except for the JSON. On the other hand, the inflation rate in the size of JSON, generally was higher than the inflation rate of the data.

It is not possible to examine the effective performance of HDF file in terms of increase rate. Because it allocates extra space for the data that will probably be added in the future within the HDF file. Due to the redundant space allocated, the increase in the actual size of the file could not be fully measured. However, it can be easily said that the file occupies more space than the actual data in terms of the space it takes in the recording media.

Experiments have shown that Pandas data containing only integer or float variables will have the smallest file size when saved as a Pickle file. However, the Excel file ensures the smallest file size given that the data only comprise of categorical or string variables, or a mix of all variable types. On the other hand, amongst the file types examined, the HDF file occupies the largest disk space.

6. Conclusion

The Pandas is a widely used library by those who extensively deal with data. It allows one to work with the data in a very easy, rapid and efficient way. The Pandas enables the data to be exported to or read from various file formats. An important issue in this context is which file format is more convenient to represent the Pandas data in terms of occupied space on the storage media. In order to find out a clue for this question, the storage space of some file types to which the Pandas data can be converted has been evaluated from various aspects such as data size and lack or completeness of data.

For the experiments, two different sized datasets, one relatively small and the other 10 times more than the small dataset, were created synthetically. Both data sets were prepared according to two different alternatives in order to allow experimental analysis under the complete or incomplete data conditions. All data sets were examined separately and 10 times for each variable type, consisting only of float, integer, string, categorical and also a mixture of these variable types. The average file sizes that were obtained from the examinations were benchmarked. As a result of the comparisons, the files with the most successful performance were determined for the cases where the data sets are incomplete by twenty percent or complete for each variable data type. For both full and missing data cases, the

rate of increase in file sizes according to the increase in the amount of data was also examined.

In future studies, the capabilities of the files for much larger data capacities will be analyzed. It will also examine other aspects such as required memory consumption and time to read and/or write files.

References

Abeykoon, V., Perera, N., Widanage, C., Kamburugamuve, S., Kanewala, T. A., Maithree, H., … Fox, G. (2020). Data Engineering for HPC with Python. In *2020 IEEE/ACM 9th Workshop on Python for High-Performance and Scientific Computing (PyHPC)* (pp. 13–21). https://doi.org/10.1109/PyHPC51966.2020.00007

Fortner, B. (1998). HDF: The hierarchical data format. *Dr Dobb's J Software Tools Prof Program*, *23*(5), 42.

Hoyer, S., & Hamman, J. (2017). xarray: ND labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1).

Kişisel Verilerin Korunması Kanunu. (n.d.). Retrieved from https://www.mevzuat.gov.tr/mevzuatmetin/1.5.6698.pdf

Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M., & Vrgoč, D. (2016). Foundations of JSON schema. In *Proceedings of the 25th International Conference on World Wide Web* (pp. 263–273).

Reback, J., McKinney, W., jbrockmendel, den Bossche, J. Van, Augspurger, T., Cloud, P., … Mehyar, M. (2020). pandas-dev/pandas: Pandas 1.0.3. Zenodo. https://doi.org/10.5281/zenodo.3715232

Van Rossum, G. (2020). *The Python Library Reference, release 3.8.2*. Python Software Foundation.

Van Rossum, G., & Drake, F. L. (2003). *An introduction to Python*. Network Theory Ltd. Bristol.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … SciPy 1.0 Contributors. (2020). {SciPy} 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2