data-viz-python

Norah Jones

2025-02-07

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

1 Reading data

1.1 Types of data - structured and unstructured

Reading data is the first step to extract information from it. Data can exist broadly in two formats:

(1) Structured data, and
(2) Unstructured data.

Structured data is typically stored in a tabular form, where rows in the data correspond to "observations" and columns correspond to "variables". For example, the following dataset contains 5 observations, where each observation (or row) consists of information about a movie. The variables (or columns) contain different pieces of information about a given movie. As all variables for a given row are related to the same movie, the data below is also called relational data.

```
ModuleNotFoundError: No module named 'pandas'
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 2
      1 #| echo: false
----> 2 import pandas as pd
      3 data = pd.read_csv('movies_sample_data.csv')
      4 data.head()
ModuleNotFoundError: No module named 'pandas'
```

Unstructured data is data that is not organized in any pre-defined manner. Examples of unstructured data can be text files, audio/video files, images, Internet of Things (IoT) data, etc. Unstructured data is relatively harder to analyze as most of the analytical methods and tools are oriented towards structured data. However, an unstructured data can be used to obtain structured data, which in turn can be analyzed. For example, an image can be converted to an array of pixels - which will be structured data. Machine learning algorithms can then be used on the array to classify the image as that of a dog or a cat.

In this course, we will focus on analyzing structured data.

1.2 Reading a *csv* file with *Pandas*

Structured data can be stored in a variety of formats. The most popular format is *data_file_name.csv*, where the extension *csv* stands for comma separated values. The variable values of each observation are separated by a comma in a *.csv* file. In other words, the **delimiter** is a comma in a *csv* file. However, the comma is not visible when a *.csv* file is opened with Microsoft Excel.

1.2.1 Using the *read_csv* function

We will use functions from the *Pandas* library of *Python* to read data. Let us import *Pandas* to use its functions.

```
import pandas as pd
```

Note that *pd* is the acronym that we will use to call a *Pandas* function. This acronym can be anything as desired by the user.

The function to read a *csv* file is read_csv(). It reads the dataset into an object of type Pandas DataFrame. Let us read the dataset *movie_ratings.csv* in Python.

```
movie_ratings = pd.read_csv('movie_ratings.csv')
```

The built-in python function `type` can be used to check the dataype of an object:

```
type(movie_ratings)
```

2 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

3 Summary

In summary, this book has no content whatsoever.

References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.