

Contents

Acknowledgment — V

1 Presenting the challenge — 1

2 Python quick start — 4

- 2.1 Why Python? — 4
- 2.2 Starting up — 4
- 2.3 Essential data types — 8
 - 2.3.1 Integer, float and string — 8
 - 2.3.2 List — 11
 - 2.3.3 Tuple — 13
 - 2.3.4 Dictionary — 14
 - 2.3.5 Data frame — 16
- 2.4 Essential structures — 23
 - 2.4.1 if-conditions — 23
 - 2.4.2 for-loops — 24
 - 2.4.3 zip — 25
 - 2.4.4 Functions — 26
 - 2.4.5 try and except — 29
 - 2.4.6 User defined packages and modules — 30
- 2.5 Wrap up — 33

3 The steps of data processing — 34

- 3.1 Introductory data pyramid example — 35
- 3.2 Misconceptions about the wisdom level — 36
- 3.3 Navigating through the levels of the data pyramid — 37
- 3.4 Wrap up — 39

4 From experimental files to data — 40

- 4.1 Challenges — 40
- 4.2 Introducing the exemplary results file structure — 41
- 4.3 Pseudo-code — 46
- 4.4 Shortcut for accessing data and information: pandas-functions — 47
- 4.5 Reading file contents — 48
- 4.6 Regular expressions for accessing data and information — 49
 - 4.6.1 Accessing data from the *Collected Data* section — 51
 - 4.6.2 Accessing data from the *General Information* section — 57
- 4.7 Building the custom `file_to_data` module — 64
- 4.8 Data and information extraction via a custom module — 70
- 4.9 Further considerations — 71

4.10	Wrap up — 72
5	From data to information — 73
5.1	Suggested folder organization — 73
5.2	Pseudo-code — 75
5.3	Building on already existing features — 76
5.4	Basic data visualization to identify parameters — 76
5.5	Parameter extraction step by step — 80
5.6	Selecting the CMC from experimentally determined values — 84
5.7	Building the <code>data_to_information</code> module — 86
5.8	Using the custom <code>data_to_information</code> module — 89
5.9	Wrap up — 91
6	Where to put data and information — 92
6.1	The minimal option: organized folder structure — 93
6.2	Adapting module functions — 98
6.3	Just one data point... — 100
6.4	Collecting <i>data</i> and <i>information</i> from multiple files — 102
6.5	More advanced: using SQLite — 106
6.6	Database creation — 107
6.7	Create tables — 107
6.8	Filling the tables — 111
6.9	Benefits of a relational structure — 114
6.10	Wrap up — 117
7	How to visualize data and information — 119
7.1	Source and visualization options — 119
7.2	Python script + matplotlib and/or seaborn — 122
7.2.1	Sourcing and selection via SQLAlchemy — 122
7.2.2	Sourcing and selection via pandas — 124
7.2.3	Plotting via matplotlib — 127
7.2.4	Plotting via seaborn — 129
7.3	Dedicated visualization software — 132
7.3.1	Connecting to data via Microsoft® Power BI Desktop® — 133
7.3.2	Building a data model — 136
7.3.3	Visualizing — 138
7.4	Wrap up — 141
8	Responding to lessons learned — 142
8.1	Taking another perspective — 142
8.2	Extracting and ingesting additional parameters from data — 143
8.3	Extending the existing database table <code>table_parameters</code> — 148

8.4	Writing to the extended table <code>table_parameters</code> —	149
8.5	Wrap up —	152
9	Where to go from here —	154
9.1	Linear models —	155
9.2	Causal analysis —	156
9.3	dowhy? —	157
9.3.1	Model the problem as a causal graph —	161
9.3.2	Identify the causal effect —	162
9.3.3	Estimate the causal effect —	163
9.3.4	Refuting the estimate —	166
9.4	Wrap up —	167
10	Conclusion —	168
A	Packaging a custom module —	173
B	Comments to tables and columns via SQLAlchemy —	177
C	A word on version control systems —	179
D	Overview of utilized Python and package versions —	181
E	Extracting data via <code>pd.read_csv</code> —	183
F	Installation of an ODBC driver —	187
	List of Figures —	189
	Bibliography —	195
	Index —	197