
Practical Python Data Wrangling and Data Quality

Susan E. McGregor

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Table of Contents

Preface.....	ix
1. Introduction to Data Wrangling and Data Quality.....	1
What Is “Data Wrangling”?	2
What Is “Data Quality”?	3
Data Integrity	4
Data “Fit”	5
Why Python?	6
Versatility	6
Accessibility	7
Readability	7
Community	7
Python Alternatives	8
Writing and “Running” Python	8
Working with Python on Your Own Device	11
Getting Started with the Command Line	11
Installing Python, Jupyter Notebook, and a Code Editor	14
Working with Python Online	19
Hello World!	20
Using Atom to Create a Standalone Python File	20
Using Jupyter to Create a New Python Notebook	21
Using Google Colab to Create a New Python Notebook	22
Adding the Code	23
In a Standalone File	23
In a Notebook	23
Running the Code	23
In a Standalone File	23
In a Notebook	24

Documenting, Saving, and Versioning Your Work	24
Documenting	24
Saving	25
Versioning	26
Conclusion	35
2. Introduction to Python.....	37
The Programming “Parts of Speech”	38
Nouns ≈ Variables	39
Verbs ≈ Functions	42
Cooking with Custom Functions	46
Libraries: Borrowing Custom Functions from Other Coders	47
Taking Control: Loops and Conditionals	47
In the Loop	48
One Condition...	51
Understanding Errors	55
Syntax Snafus	56
Runtime Runaround	58
Logic Loss	60
Hitting the Road with Citi Bike Data	62
Starting with Pseudocode	63
Seeking Scale	69
Conclusion	70
3. Understanding Data Quality.....	71
Assessing Data Fit	73
Validity	74
Reliability	76
Representativeness	77
Assessing Data Integrity	79
Necessary, but Not Sufficient	81
Important	82
Achievable	85
Improving Data Quality	88
Data Cleaning	88
Data Augmentation	89
Conclusion	90
4. Working with File-Based and Feed-Based Data in Python.....	91
Structured Versus Unstructured Data	93
Working with Structured Data	97
File-Based, Table-Type Data—Take It to Delimit	97

Wrangling Table-Type Data with Python	99
Real-World Data Wrangling: Understanding Unemployment	105
XLSX, ODS, and All the Rest	107
Finally, Fixed-Width	114
Feed-Based Data—Web-Driven Live Updates	118
Wrangling Feed-Type Data with Python	120
Working with Unstructured Data	134
Image-Based Text: Accessing Data in PDFs	134
Wrangling PDFs with Python	135
Accessing PDF Tables with Tabula	139
Conclusion	140
5. Accessing Web-Based Data	141
Accessing Online XML and JSON	143
Introducing APIs	145
Basic APIs: A Search Engine Example	146
Specialized APIs: Adding Basic Authentication	148
Getting a FRED API Key	149
Using Your API key to Request Data	150
Reading API Documentation	151
Protecting Your API Key When Using Python	153
Creating Your “Credentials” File	155
Using Your Credentials in a Separate Script	155
Getting Started with .gitignore	157
Specialized APIs: Working With OAuth	159
Applying for a Twitter Developer Account	160
Creating Your Twitter “App” and Credentials	162
Encoding Your API Key and Secret	167
Requesting an Access Token and Data from the Twitter API	168
API Ethics	172
Web Scraping: The Data Source of Last Resort	173
Carefully Scraping the MTA	176
Using Browser Inspection Tools	178
The Python Web Scraping Solution: Beautiful Soup	180
Conclusion	184
6. Assessing Data Quality	185
The Pandemic and the PPP	187
Assessing Data Integrity	187
Is It of Known Pedigree?	188
Is It Timely?	189
Is It Complete?	189

Is It Well-Annotated?	201
Is It High Volume?	206
Is It Consistent?	208
Is It Multivariate?	211
Is It Atomic?	213
Is It Clear?	213
Is It Dimensionally Structured?	215
Assessing Data Fit	215
Validity	216
Reliability	219
Representativeness	220
Conclusion	222
7. Cleaning, Transforming, and Augmenting Data.....	225
Selecting a Subset of Citi Bike Data	226
A Simple Split	227
Regular Expressions: Supercharged String Matching	229
Making a Date	233
De-crufting Data Files	236
Decrypting Excel Dates	239
Generating True CSVs from Fixed-Width Data	242
Correcting for Spelling Inconsistencies	244
The Circuitous Path to “Simple” Solutions	250
Gotchas That Will Get Ya!	252
Augmenting Your Data	253
Conclusion	256
8. Structuring and Refactoring Your Code.....	257
Revisiting Custom Functions	258
Will You Use It More Than Once?	258
Is It Ugly and Confusing?	258
Do You Just Really Hate the Default Functionality?	259
Understanding Scope	259
Defining the Parameters for Function “Ingredients”	262
What Are Your Options?	263
Getting Into Arguments?	263
Return Values	264
Climbing the “Stack”	265
Refactoring for Fun and Profit	267
A Function for Identifying Weekdays	267
Metadata Without the Mess	270
Documenting Your Custom Scripts and Functions with pydoc	277

The Case for Command-Line Arguments	281
Where Scripts and Notebooks Diverge	284
Conclusion	285
9. Introduction to Data Analysis.....	287
Context Is Everything	288
Same but Different	289
What's Typical? Evaluating Central Tendency	290
What's That Mean?	290
Embrace the Median	291
Think Different: Identifying Outliers	292
Visualization for Data Analysis	292
What's Our Data's Shape? Understanding Histograms	296
The Significance of Symmetry	297
Counting "Clusters"	305
The \$2 Million Question	306
Proportional Response	317
Conclusion	321
10. Presenting Your Data.....	323
Foundations for Visual Eloquence	324
Making Your Data Statement	326
Charts, Graphs, and Maps: Oh My!	327
Pie Charts	328
Bar and Column Charts	330
Line Charts	335
Scatter Charts	339
Maps	342
Elements of Eloquent Visuals	345
The "Finicky" Details Really Do Make a Difference	345
Trust Your Eyes (and the Experts)	345
Selecting Scales	347
Choosing Colors	347
Above All, Annotate!	348
From Basic to Beautiful: Customizing a Visualization with seaborn and matplotlib	349
Beyond the Basics	354
Conclusion	355
11. Beyond Python.....	357
Additional Tools for Data Review	358
Spreadsheet Programs	358

OpenRefine	359
Additional Tools for Sharing and Presenting Data	361
Image Editing for JPGs, PNGs, and GIFs	361
Software for Editing SVGs and Other Vector Formats	362
Reflecting on Ethics	363
Conclusion	364
A. More Python Programming Resources.....	365
B. A Bit More About Git.....	369
C. Finding Data.....	375
D. Resources for Visualization and Information Design.....	381
Index.....	383