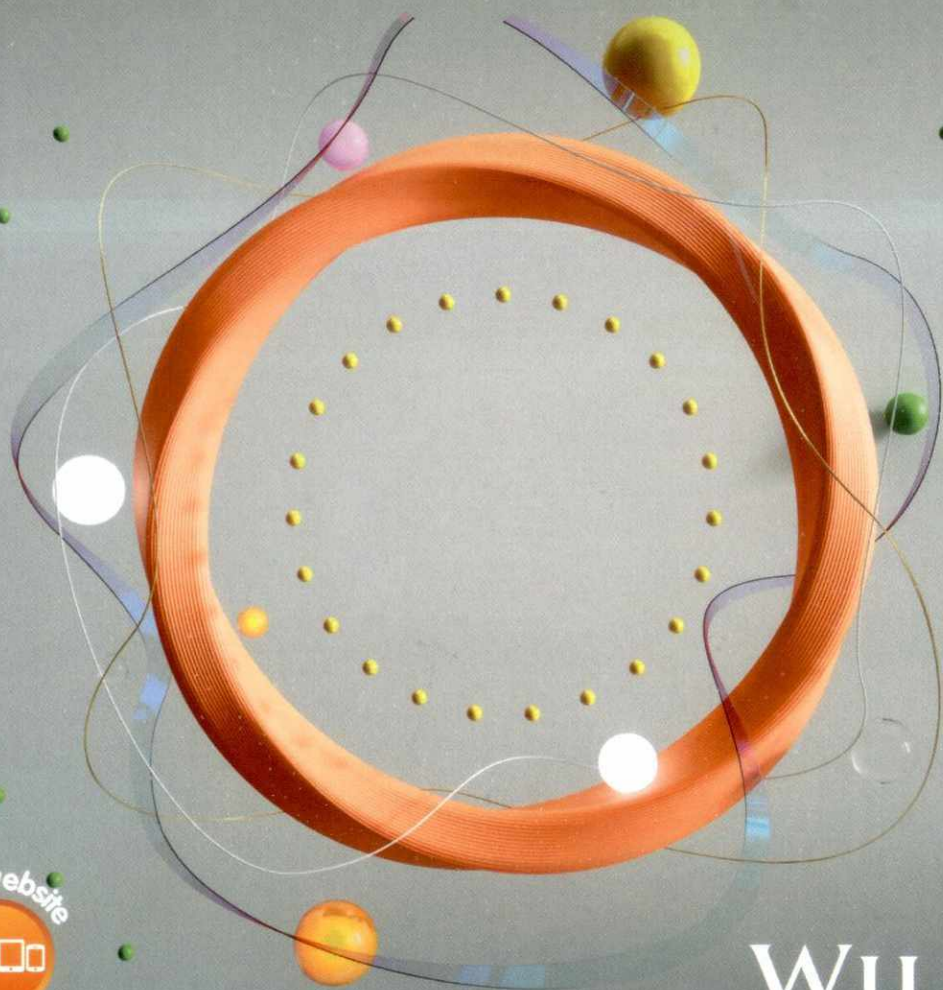


Data Science Fundamentals with **R, Python, and Open Data**

Marco Cremonini



WILEY

Contents

Preface *xiii*

About the Companion Website *xvii*

Introduction *xix*

1	Open-Source Tools for Data Science	1
1.1	R Language and RStudio	1
1.1.1	R Language	2
1.1.2	RStudio Desktop	2
1.1.3	Package Manager	2
1.1.4	Package Tidyverse	4
1.2	Python Language and Tools	5
1.2.1	Option A: Anaconda Distribution	6
1.2.2	Option B: Manual Installation	6
1.2.3	Google Colab	7
1.2.4	Packages NumPy and Pandas	7
1.3	Advanced Plain Text Editor	8
1.4	CSV Format for Datasets	8
	Questions	10
2	Simple Exploratory Data Analysis	13
2.1	Missing Values Analysis	13
2.2	R: Descriptive Statistics and Utility Functions	15
2.3	Python: Descriptive Statistics and Utility Functions	17
	Questions	19
3	Data Organization and First Data Frame Operations	23
3.1	R: Read CSV Datasets and Column Selection	24
3.1.1	Reading a CSV Dataset	26
3.1.1.1	Reading Errors	27
3.1.2	Selection by Column Name	29
3.1.3	Selection by Column Index Position	30
3.1.4	Selection by Range	31
3.1.5	Selection by Exclusion	32
3.1.6	Selection with Selection Helper	35
3.2	R: Rename and Relocate Columns	36

3.3	R: Slicing, Column Creation, and Deletion	38
3.3.1	Subsetting and Slicing	39
3.3.2	Column Creation	42
3.3.3	Column Deletion	43
3.3.4	Calculated Columns	44
3.3.5	Function <code>mutate()</code> and Data Masking	44
3.4	R: Separate and Unite Columns	45
3.4.1	Separation	46
3.4.2	Union	48
3.5	R: Sorting Data Frames	49
3.5.1	Sorting by Multiple Columns	50
3.5.2	Sorting by an External List	51
3.6	R: Pipe	55
3.6.1	Forward Pipe	55
3.6.2	Pipe in Base R	57
3.6.2.1	Variant	57
3.6.3	Parameter Placeholder	58
3.7	Python: Column Selection	59
3.7.1	Selecting Columns from Dataset Read	61
3.7.2	Selecting Columns from a Data Frame	62
3.7.3	Selection by Positional Index, Range, or with Selection Helper	63
3.7.4	Selection by Exclusion	64
3.8	Python: Rename and Relocate Columns	67
3.8.1	Standard Method	67
3.8.2	Functions <code>rename()</code> and <code>reindex()</code>	67
3.9	Python: NumPy Slicing, Selection with Index, Column Creation and Deletion	69
3.9.1	NumPy Array Slicing	69
3.9.2	Slicing of Pandas Data Frames	70
3.9.3	Methods <code>.loc</code> and <code>.iloc</code>	73
3.9.4	Selection with Selection Helper	77
3.9.5	Creating and Deleting Columns	79
3.9.6	Functions <code>insert()</code> and <code>assign()</code>	80
3.10	Python: Separate and Unite Columns	81
3.10.1	Separate	81
3.10.2	Unite	84
3.11	Python: Sorting Data Frame	85
3.11.1	Sorting Columns	85
3.11.2	Sorting Index Levels	86
3.11.3	From Indexed to Non-indexed Data Frame	88
3.11.4	Sorting by an External List	89
	Questions	91
4	Subsetting with Logical Conditions	99
4.1	Logical Operators	99
4.2	R: Row Selection	101
4.2.1	Operator <code>%in%</code>	104
4.2.2	Boolean Mask	105

4.2.3	Examples	106
4.2.3.1	Wrong Disjoint Condition	107
4.2.4	Python: Row Selection	114
4.2.5	Boolean Mask, Base Selection Method	115
4.2.6	Row Selection with <code>query()</code>	119
	Questions	121
5	Operations on Dates, Strings, and Missing Values	127
5.1	R: Operations on Dates and Strings	129
5.1.1	Date and Time	129
5.1.1.1	Datetime Data Type	129
5.1.2	Parsing Dates	130
5.1.3	Using Dates	132
5.1.4	Selection with Logical Conditions on Dates	133
5.1.5	Strings	136
5.2	R: Handling Missing Values and Data Type Transformations	141
5.2.1	Missing Values as Replacement	142
5.2.1.1	Keywords for Missing Values	142
5.2.2	Introducing Missing Values in Dataset Reads	143
5.2.3	Verifying the Presence of Missing Values	144
5.2.3.1	Functions <code>any()</code> , <code>all()</code> , and <code>colSums()</code>	146
5.2.4	Replacing Missing Values	147
5.2.5	Omit Rows with Missing Values	149
5.2.6	Data Type Transformations	150
5.3	R: Example with Dates, Strings, and Missing Values	154
5.3.1	When an Invisible Hand Mess with Your Data	158
5.3.2	Base Method	159
5.3.3	A Better Heuristic	162
5.3.4	Specialized Functions	162
5.3.4.1	Function <code>parse_date_time()</code>	162
5.3.5	Result Comparison	165
5.4	Python: Operations on Dates and Strings	165
5.4.1	Date and Time	165
5.4.1.1	Function <code>pd.to_datetime()</code>	165
5.4.1.2	Function <code>datetime.datetime.strptime()</code>	167
5.4.1.3	Locale Configuration	168
5.4.1.4	Function <code>datetime.datetime.strptime()</code>	169
5.4.1.5	Pandas Timestamp Functions	169
5.4.2	Selection with Logical Conditions on Dates	171
5.4.3	Strings	172
5.5	Python: Handling Missing Values and Data Type Transformations	173
5.5.1	Missing Values as Replacement	173
5.5.1.1	Function <code>pd.replace()</code>	175
5.5.2	Introducing Missing Values in Dataset Reads	175
5.5.3	Verifying the Presence of Missing Values	176
5.5.4	Selection with Missing Values	178
5.5.5	Replacing Missing Values with Actual Values	179

5.5.6	Modifying Values <i>by View</i> or <i>by Copy</i>	180
5.5.7	Data Type Transformations	182
5.6	Python: Examples with Dates, Strings, and Missing Values	182
5.6.1	Example 1: Eurostat	182
5.6.2	Example 2: Open Data Berlin	186
	Questions	190
6	Pivoting and Wide-long Transformations	195
6.1	R: Pivoting	197
6.1.1	From Long to Wide	197
6.1.2	From Wide to Long	199
6.1.3	GOV.UK: Gender Pay Gap	200
6.2	Python: Pivoting	202
6.2.1	From Wide to Long with Columns	203
6.2.2	From Long to Wide with Columns	204
6.2.3	Wide-long Transformation with Index Levels	206
6.2.4	Indexed Data Frame	207
6.2.4.1	Function <code>unstack()</code>	208
6.2.4.2	Function <code>stack()</code>	211
6.2.5	From Long to Wide with Elements of Numeric Type	213
	Questions	216
7	Groups and Operations on Groups	221
7.1	R: Groups	222
7.1.1	Groups and Group Indexes	224
7.1.1.1	Function <code>group_by()</code>	224
7.1.1.2	Index Details	226
7.1.2	Aggregation Operations	227
7.1.2.1	Functions <code>group_by()</code> and <code>summarize()</code>	227
7.1.2.2	Counting Rows: function <code>n()</code>	228
7.1.2.3	Arithmetic Mean: function <code>mean()</code>	228
7.1.2.4	Maximum and Minimum Values: Functions <code>max()</code> and <code>min()</code>	230
7.1.2.5	Summing Values: function <code>sum()</code>	231
7.1.2.6	List of Aggregation Functions	232
7.1.3	Sorting Within Groups	232
7.1.4	Creation of Columns in Grouped Data Frames	234
7.1.5	Slicing Rows on Groups	236
7.1.5.1	Functions <code>slice_*</code>	236
7.1.5.2	Combination of Functions <code>filter()</code> and <code>rank()</code>	238
7.1.6	Calculated Columns with Group Values	242
7.2	Python: Groups	244
7.2.1	Group Index and Aggregation Operations	247
7.2.1.1	Functions <code>groupby()</code> and <code>aggregate()</code>	247
7.2.1.2	Counting Rows, Computing Arithmetic Means, and Sum for Each Group	247
7.2.2	Names on Columns with Aggregated Values	251
7.2.3	Sorting Columns	252
7.2.4	Sorting on Index Levels	254

7.2.5	Slicing Rows on Groups	255
7.2.5.1	Functions <code>nlargest()</code> and <code>nsmallest()</code>	259
7.2.6	Calculated Columns with Group Values	259
7.2.7	Sorting Within Groups	261
	Questions	265
8	Conditions and Iterations	271
8.1	R: Conditions and Iterations	272
8.1.1	Conditions	272
8.1.1.1	Function <code>if_else()</code>	275
8.1.1.2	Function <code>case_when()</code>	276
8.1.1.3	Function <code>if()</code> and Constructs <code>If-else</code> and <code>If-else If-else</code>	277
8.1.2	Iterations	278
8.1.2.1	Function <code>for()</code>	278
8.1.2.2	Function <code>Foreach()</code>	280
8.1.3	Nested Iterations	280
8.1.3.1	Replacing a Single-Element Value	282
8.1.3.2	Iterate on the First Column	283
8.1.3.3	Iterate on all Columns	283
8.2	Python: Conditions and Iterations	284
8.2.1	Conditions	284
8.2.1.1	Function <code>if()</code>	285
8.2.1.2	Constructs <code>If-else</code> and <code>If-elif-else</code>	285
8.2.1.3	Function <code>np.where()</code>	286
8.2.1.4	Function <code>np.select()</code>	287
8.2.1.5	Functions <code>pd.where()</code> and <code>pd.mask()</code>	289
8.2.2	Iterations	291
8.2.2.1	Functions <code>for()</code> and <code>while()</code>	291
8.2.3	Nested Iterations	294
8.2.3.1	Execution Time	296
8.2.4	Iterating on Multi-index	297
8.2.4.1	Function <code>join()</code>	300
8.2.4.2	Function <code>items()</code>	301
	Questions	302
9	Functions and Multicolumn Operations	307
9.1	R: User-defined Functions	308
9.1.1	Using Functions	309
9.1.2	Data Masking	312
9.1.3	Anonymous Functions	315
9.2	R: Multicolumn Operations	316
9.2.1	Base Method	316
9.2.1.1	Functions <code>apply()</code> , <code>lapply()</code> , and <code>sapply()</code>	316
9.2.1.2	Mapping	319
9.2.2	Mapping and Anonymous Functions: <i>purrr</i> -style Syntax	321
9.2.3	Conditional Mapping	321
9.2.4	Subsetting Rows with Multicolumn Logical Condition	323

9.2.4.1	Combination of Functions <code>filter()</code> and <code>if_any()</code>	323
9.2.5	Multicolumn Transformations	324
9.2.5.1	Combination of Functions <code>mutate()</code> and <code>across()</code>	324
9.2.6	Introducing Missing Values	325
9.2.7	Use Cases and Execution Time Measurement	326
9.2.7.1	Case 1	327
9.2.7.2	Case 2	328
9.3	Python: User-defined and Lambda Functions	330
9.3.1	User-defined Functions	330
9.3.1.1	Lambda Functions	333
9.3.2	Python: Multicolumn Operations	334
9.3.2.1	Execution Time	336
9.3.3	General Case	337
9.3.3.1	Function <code>apply()</code>	337
	Questions	342

10 Join Data Frames 347

10.1	Basic Concepts	348
10.1.1	Keys of a Join Operation	349
10.1.2	Types of Join	350
10.1.3	R: Join Operation	351
10.1.4	Join Functions	354
10.1.4.1	Function <code>inner_join()</code>	354
10.1.4.2	Function <code>full_join()</code>	356
10.1.4.3	Functions <code>left_join()</code> and <code>right_join()</code>	357
10.1.4.4	Function <code>merge()</code>	357
10.1.5	Duplicated Keys	358
10.1.6	Special Join Functions	363
10.1.6.1	Semi Join	363
10.1.6.2	Anti Join	365
10.2	Python: Join Operations	369
10.2.1.1	Function <code>merge()</code>	371
10.2.1.2	Inner Join	372
10.2.1.3	Outer/Full Join	374
10.2.2	Join Operations with Indexed Data Frames	375
10.2.3	Duplicated Keys	378
10.2.4	Special Join Types	384
10.2.4.1	Semi Join: Function <code>isin()</code>	384
10.2.4.2	Anti Join: Variants	386
	Questions	389

11 List/Dictionary Data Format 393

11.1	R: List Data Format	395
11.1.1	Transformation of List Columns to Ordinary Rows and Columns	401
11.1.1.1	Other Options	403
11.1.2	Function <code>map</code> in List Column Transformations	406
11.2	R: JSON Data Format and Use Cases	410

11.2.1	Memory Problem when Reading Very Large Datasets	421
11.3	Python: Dictionary Data Format	422
11.3.1	Methods	424
11.3.2	From Dictionary to Data Frame With a Single Level of Nesting	427
11.3.2.1	Functions <code>pd.DataFrame()</code> and <code>pd.DataFrame.from_dict()</code>	427
11.3.3	From Dictionary to Data Frame with Several Levels of Nesting	429
11.3.3.1	Function <code>pd.json_normalize()</code> and Join Operation	429
11.3.4	Python: Use Cases with JSON Datasets	436
	Questions	443

Index 447