

INDEX

Page numbers in italics refer to figures and tables.

Symbols

- + (addition operator), *60*
- = (assignment operator)
 - arguments, 18, 146
 - assignment statements, 12, 27
- \ (backslash), 210, 286
- : (colon)
 - block delimiter, 68
 - indexing, 116, 119
 - key-value pairs, 107
 - statements, 70
- { } (curly brackets), 41
 - declaring dictionaries, 106, 131
 - renaming columns, 53
 - replacing values in columns, 55, 57
 - specifying column headers, 27
- ' ' (curly quotation marks), 21
- / (division operator), *60*
- \\ (double backslash), 210, 286
- " " (double quotation marks), 13, 20–21, *21*
- [[]] (double square brackets), 32, 41, 50, 113–114
- ... (ellipsis)
 - continued input lines, 5
 - hidden and truncated data, 128–129
- == (equality operator), *60*
 - comparing lists, 44–45
 - comparing values, 67
 - matching values in columns, 120
- \ (escape character), 19, 210, 286
- \\ (escape sequence for backslash), 19–20, 210, 286
- \' (escape sequence for quotation mark), 19
- / (forward slash), 210, 286
- > (greater than operator), *60*, 123
- >= (greater than or equal to operator), *60*, 123
- # (hash mark)
 - comment lines, 254
 - removing padding zeros from dates, 197
- < (less than operator), *60*, 123
- <= (less than or equal to operator), *60*, 123
- * (multiplication operator), *60*, 66
- != (not equal to operator), *60*, 121
- () (parentheses), 41
 - creating dataframes, 27–28
 - default settings, 31
 - display text, 12
 - equations, 15
 - escape characters, 19
 - functions and methods, 32
- % (percent sign; directives), 195
- >>> prompt, 5
- ' ' (single quotation marks)
 - blank values, 58
 - directive strings, 195
 - printing, 19
 - strings, 12–13
 - using consistently, 13, 20–21, *21*
- [] (square brackets), 41
 - attributes, 32
 - concatenating dataframes, 37
 - defining columns, 27, 50
 - list comprehensions, 73
 - lists, 40, 42, 51, 91, 123, 132
 - slicing, 88, 90
- (subtraction operator), *60*
- ~ (tilde) for the opposite, 122–123, 125

A

absolute values
 abs() function (values), 64–65
 abs() method (dataframes), 65

accessors, 29, 257
 dt for Datetime module, 190
 importing modules, 29
 np for NumPy, 44
 op for OpenPyXL module, 210
 pd for pandas module, 29
 st for statistics module, 57
 string accessor, str(), 57
 udf for DIY script, 256

addition operator (+), 60

%a directive (Sun), 195

%A directive (Sunday), 195

aggregate functions
 aggfunc parameter, 157
 calculating math values,
 161–162
 crosstab() method vs. pivot
 _table() method, 157
 functionality of, 147
 handling NaN objects, 164
 math operations, 147–150
 multiple aggregate functions,
 150–151
 pivot_table() method, 158–160
 calculating math values, 161–163
 defined, 144
 math operators, 148
 multiple, 150–152

alignment in Excel output cells
 alignment attribute, 238–239
 Alignment class, 238–239, 241–242

Anaconda distribution platform
 downloading, 4
 IDLE, 5
 packaging to allow module
 importing, 26
 Spyder, 6

append() method (lists), 4, 43

arguments, 18, 146

assignment operator (=)
 arguments, 18, 146
 assignment statements, 12, 27

astype() method (columns), 59,
 81–82

at[] attribute
 changing specific cell values,
 300–301
 selecting data by label, 95

attributes, 32

AutoFilters
 auto_filter.ref parameter, 234–236,
 242, 270
 Excel file formatting, 235–236, 236

automation, 249–283
 analyzing Excel workbooks, 250–251,
 250, 251
 dynamic pivot tables, 280–282
 emailing, 244–248
 exception reports and data
 management tasks, 262–263
 flowcharting manual process,
 251–252, 252
 modularity, 252–257
 saving UDFs in separate scripts,
 256–257, 256
 writing and calling UDFs, 252–253
 rolling over files, 257–262
 creating new file for today's date,
 260–262
 importing favorite modules, 257
 importing UDFs, 258
 printing headers, 258
 setting dataframe display
 preferences, 258–259
 setting date, 259–260
 Spyder as daily workflow GUI, 282
 updating Excel files, 276–279, 279

AVERAGE formula (Excel), 150

axis attribute (dataframes)
 axis = 0 (row), 36
 axis = 1 (column), 38

B

backslash (\), 210, 286

%b directive (Jan), 194, 195

%B directive (January), 194, 195

blocks of code, 68–74
 block delimiter (colon, :), 68

Boolean objects
 comparing lists, 45
 comparing values, 67–68
 converting objects to strings, 82

- source and definition, *14*
- True and False values, 45
- Border class (Excel formatting), 241–242
- bracketology, 41
 - { } (curly brackets), 41
 - declaring dictionaries, 106, 131
 - renaming columns, 53
 - replacing values in columns, 55, 57
 - specifying column headers, 27
 - [[]] (double square brackets), 32, 41, 50, 113–114
 - () (parentheses), 41
 - creating dataframes, 27–28
 - default settings, 31
 - display text, 12
 - equations, 15
 - escape characters, 19
 - functions and methods, 32
 - [] (square brackets), 41
 - attributes, 32
 - concatenating dataframes, 37
 - defining columns, 27, 50
 - list comprehensions, 73
 - lists, 40, 42, 51, 91, 123, 132
 - slicing, 88, 90
- C**
- cardinal relationships (`merge()` method), 183, *183*
- case sensitivity
 - dataframes, 29–30
 - ignoring, 22
 - methods, 96, 106–107
- cell values in dataframes, 77–105
- cell objects (Excel formatting), 237–238
- changing specific, 300–301
- converting, 79–82
 - entire dataframe columns, 81–82, *81*
 - individual variables, 79–81
 - objects to strings, 82
- indexing, 93–97
 - modifying existing cells, 96–97
 - pulling values by index position, 93–94
 - pulling values by unique index label, 94–95
 - pulling values from series, 95–96
 - targeting single values, 95
- manually inserting, 82–85
 - answering questions and saving answers, 83
 - pausing execution to review output, 85
 - selecting menu options, 84–85
- NaN objects, 86–88
 - manually creating, 86
 - manually entering into dataframe, 86–87
 - replacing, 87–88
- overview of values and variables, 78–79
- slicing, 88–93
 - lists from lists, 91–92
 - pulling select elements from lists, 92–93
 - single strings, 88–89
 - within entire columns, 90–91
- splitting, 98–101
 - columns in dataframes, 100–101
 - handling inconsistent delimiters, 99–100
 - single strings, 98–99
- classes (data types)
 - defined, 13
 - list of common, *14*
- `client.dispatch()` method (email), 245
- `close()` method (Excel export), 232, 302
- coding style
 - break lines, 19, 28
 - column lists, recycling, 115
 - flowcharting, 251–252
 - hard returns
 - in Excel, 216, 219, 288
 - for whitespace, 18, 28, 53, 205
 - indentation, 28, 53, 149
 - line continuation, 205
 - modules, loading favorites, 257
 - pathname convention, 211, 224, 230, 233, 261
 - `print()` statements, 16
 - recycling, 115, 256
 - testing your code in Spyder, 17

- coding style (*continued*)
 - top-level code, 257
 - visual preferences, dataframes, 125–130, 258–259
 - whitespace, 19–28, 80, 99, 149
 - workflow design, 251–252
- colon (:)
- block delimiter, 68
- indexing, 116, 119
- key-value pairs, 107
- statements, 70
- column_dimensions[.width parameter, 238–239
- columns, 49–75
 - adding with single value, 58
 - borders (Excel export), 241–242, 242
 - changing values in, 55–57
 - overwriting all values, 55
 - replacing particular values, 55–57
 - replacing substrings, 57
 - conditional logic within, 68–74
 - controlling execution flow, 68
 - handling exceptions, 73–74
 - iterating through dataframes, 70–71
 - iterating through lists, 69
 - list comprehensions, 72–73
 - repeating processes, 69–70
 - replicating SUMIF formula, 71–72
 - storing values in new columns, 72
 - converting
 - into lists, 46
 - from one data type to another, 81–82
 - counting
 - every value in a column, 140–141
 - list of specific values in a column, 141–142
 - with shape attribute, 34
 - specific values in a column, 141
 - creating empty dataframes with column lists, 108
 - customizing labels, 297
 - defining, 50
 - deleting, 51–52, 182, 300
 - duplicating, 58–59
 - fonts (Excel export), 241–242, 242
 - formatting datetime objects in, 240–241, 241
 - math methods and operators, 60–68
 - average, maximum, or minimum values, 61
 - comparing values, 67–68
 - converting to absolute values, 64–65
 - median values, 61
 - rounding, 63–64
 - storing calculated values in new columns, 66–67
 - stringing together multiple methods, 65
 - sum() method, 60–61
 - moving values between indexes and columns, 112–113
 - printing
 - lists of labels and their data types, 62
 - rightmost, 126–127
 - selected, 54–55
 - renaming (relabeling), 52–53, 181, 218, 299–300
 - reordering, 50–51, 300
 - setting column labels, 217–218
 - setting Excel input columns as dataframe indexes, 218–219
 - setting number to display, 127–128
 - setting width
 - Excel export, 238–239, 239
 - Spyder display, 128–129
 - slicing, 90–91
 - sorting dataframes by values, 53–54
 - specifying as indexes (Excel export), 297–298
 - splitting, 100–101, 299
 - subsetting dataframes
 - by column list, 113–117
 - by excluding values in columns, 121
 - by matching values in columns, 120–121
- columns parameter
 - creating empty dataframes, 108–109
 - crosstab() method vs. pivot_table() method, 157
 - explicitly defining, 145–146, 149
 - grouping unique values, 161

- comma-separated value (CSV) files,
 - converting to dataframes, 220
- concatenating
 - concat() method (dataframes)
 - functionality of, 35
 - left/right convention, 168
 - merge() method vs., 168
 - columns, 59–60
 - dataframes, 4, 35–40
 - by column, different index values, 39–40
 - by column, same index values, 38–39
 - by row, different columns, 38
 - by row, same columns, 36–37
 - data types, 15–16
 - lists, 43
 - strings, 12
- conditional logic, 68–74
 - controlling execution flow, 68
 - handling exceptions, 73–74
 - iterating
 - through dataframes, 70–71
 - through lists, 69
 - list comprehensions, 72–73
 - repeating processes, 69–70
 - replicating SUMIF formula, 71–72
 - storing values in new columns, 72
- contains() method (subsetting dataframes), 121–122
- control flow structure, 68
- converting
 - Boolean to string, 82
 - column
 - to index, 112
 - to list, 46
 - dataframe to HTML code, 245–246, 267
 - datetime to string
 - column, 193, 196–197
 - value, 196
 - float to integer
 - column, 81–82
 - value, 80
 - float to string
 - column, 81–82
 - value, 80
 - float to timedelta, 199
 - index to column, 112
 - integer to float
 - column, 81–82
 - value, 80
 - integer to string
 - column, 81–82
 - value, 80
 - set to list, 42–43
 - string to float
 - column, 81–82
 - value, 80
 - string to integer
 - column, 81–82
 - value, 80
 - timedelta to integer
 - value, 200
- copying
 - dataframes. *See* copy() method
 - Excel and other files, 290–291
 - index labels to new columns, 112
 - shutil.copy() method, 261, 291, 294
- copy() method
 - copying dataframes, 31
 - entanglement (avoidance), 114
 - filtering values before counting, 142
 - preserving dataframe instead of overwriting, 142
 - subsetting dataframes
 - by column, 31
 - by column list, 113
 - by relative index location, 117
 - utility of, 114
- copy parameter (merge() method), 174
- counting and summing, 139–166
 - count aggregate function, 159–160, 162
 - counters, 69–70
 - COUNTIF formula replication, 140–141, 141, 147, 151, 151
 - COUNTIFS formula replication, 145
 - crosstab() method, 144–156
 - adding row and column totals, 152–153
 - creating basic cross-tabulations, 144–147
 - creating multilevel cross-tabulations, 146–147
 - handling missing values, 153–155

- counting and summing (*continued*)
 - crosstab() method (*continued*)
 - normalizing results, 155–156
 - performing math operations, 147–152
 - pivot_table() method, 157–164
 - basic form of, 158–161
 - calculating math values, 161–163
 - grouping unique values, 161
 - handling missing values, 163–164
 - organizing dataframe values, 164
 - value_counts() method, 140–143
 - counting every value in columns, 140–141
 - counting lists of specific values in columns, 141–142
 - counting specific values in columns, 141
 - normalizing results, 143
 - CreateItem() method (email), 245
 - cross-tabulation, 140, 144–156
 - adding row and column totals, 152–153
 - creating
 - basic, 144–147
 - multilevel, 146–147
 - crosstab() method, 140, 144–156
 - adding row and column totals, 152–153
 - crashing execution, 152
 - creating basic cross-tabulations, 144–147
 - creating multilevel cross-tabulations, 146–147
 - handling missing values, 153–155
 - normalizing results, 155–156
 - parameters, 157
 - performing math operations, 147–152
 - pivot_table() method vs., 157–163, 157
 - defined, 144
 - handling missing values, 153–155
 - normalizing results, 155–156
 - performing math operations, 147–152
- CSV (comma-separated value) files, converting to dataframes, 220
- curly brackets ({}), 41
 - declaring dictionaries, 106, 131
 - renaming columns, 53
 - replacing values in columns, 55, 57
 - specifying column headers, 27
- curly quotation marks (‘ ’), 21
- ## D
- data filters, *See* AutoFilters
 - dataframes, xxi, 13, 25–48, 105–136
 - controlling appearance (display), 125–130
 - global display settings, 127–130
 - printing first or last few rows, 125–126
 - printing rightmost columns, 126–127
 - printing specific rows, 126
 - copying, 31
 - counting rows and columns with shape attribute, 34
 - counting rows using len() function, 33, 33
 - specifying index labels, 30–31
 - viewing output, 28–29, 29
 - DataFrame() method, 27–28, 106–108
 - arguments, 107–108
 - cleaning up formatting, 28
 - creating empty dataframes, 45–46, 108
 - creating from Excel import, 209–222
 - creating manually, 27
 - creating with column lists, 45–46, 108
 - functionality of, 107
 - specifying index labels, 30–31
 - viewing, 28–29, 29
 - data types, listing of, 62
 - datetime operations, 201–206
 - calculating timedeltas in new column, 203–204
 - common date string formats, 202
 - customizing results, 203
 - subsetting dataframes using datetime objects, 204–205
 - defined, 26
 - deleting rows with specific values, 33, 34–35

- dtypes attribute, 62
- Excel files, exporting, 223–248
 - exporting single tabs, 224
 - formatting, 234–243
- Excel files, importing, 209–222
 - complex spreadsheets, 216–219
 - creating or downloading
 - spreadsheets, 210, 210
 - filtering source data, 214
 - importing all tabs at once, 213
 - importing specific tabs from
 - workbooks, 212–213
 - parsing input spreadsheets,
 - 214–219, 215
 - reading in CSV files, 220
 - sheet_name parameter, 212–213,
 - 226, 227, 231, 302
- Excel spreadsheets vs., 27–28, 27
- exporting
 - to CSV files, 220
 - to Excel files, 228–244
 - to HTML code, 245–246
- identifying and dropping duplicated
 - rows, 33, 35
- indexes, 109–113
 - moving values between columns
 - and, 112–113
 - naming and renaming, 109
 - renumbering or resetting, 110–111
 - sorting by, 111–112
- iteration through, 70–71
- manually entering NaN objects into,
 - 86–87
- merging, 167–188
 - basics of, 168–170
 - orphaned keys, 170–172, 171
 - quality control with shape
 - attribute, 185
 - syntax, 172–185
- naming, 29–30
- passing variables into, 30
- setting display preferences, 258–259
- simple, xxvii
- structure of, 106, 107
- subsetting, 113–125
 - by column, 31–32
 - by column labels in lists, 122
 - by column list, 113–117
 - by datetime objects, 204–205
 - by excluding values in columns, 121
 - by exclusion, 123
 - by inclusion, 122–123
 - by index label, 118–119
 - by matching values in columns,
 - 120–121
 - by mathematical condition, 123–124
 - by NaN objects, 124–125
 - by relative index location, 116–117
 - by substring values, 121–122
- virtual spreadsheets, xxi, xxvii,
 - 27–28, 33
- data parameter (DataFrame() method),
 - 108, 157, 159
- data types, 12–17
 - classes, 13
 - concatenating, 15–16
 - converting, 79–82
 - defined, 12
 - displaying, 78
 - dtypes attribute (dataframes), 62
 - failure to standardize, 16–17
 - info() method (dataframes), 62
 - knowing your data, 22
 - list of common, 14
 - native, 13
 - type() function (values), 12, 14, 78
- dates and times, 190–206
 - comparing and calculating,
 - 199–200
 - converting to other data types,
 - 81–82
 - creating new file for today's date,
 - 260–262
 - DateOffset() method, 261
 - datetime64[ns] objects, 201–204
 - datetime() method
 - creating datetime objects,
 - 190–191, 196
 - re-creating datetime objects, 192
 - Datetime module, 14, 189–190,
 - 203, 264
 - datetime objects, 189–197
 - dt accessor, 190
 - formatting in single column,
 - 240–241, 241
 - importing, 190

- dates and times (*continued*)
 - Datetime module (*continued*)
 - isolating units of time as integers, 191–192
 - source and definition, 14
 - `datetime.now()` method, 189–191, 260
 - datetime operations, 201–206
 - calculating timedeltas in new column, 203–204
 - customizing results, 203
 - subsetting dataframes using datetime objects, 204–205
 - isolating units of time as integers, 191–192
 - `strftime()` method, 189
 - time durations, 198–199
- `%d` directive (day of month), 195
- debugging
 - commenting, value of, 254
 - defined, 5–6
 - IDLE, 6, 16, 182, 289
 - printing list of column labels, 62
 - printing variable type, 47
 - Spyder, 6, 17, 20–21
 - status messages, 259
 - temporarily pausing execution, 269
- deleting
 - columns, 51–52, 182, 300
 - files, 292
 - folders, 293
 - indexes, 110
 - rows. *See also* subsetting dataframes
 - by index label, 119–120, 298
 - with specific values, 33, 34–35
- delimiters
 - directive strings, 195
 - inconsistent, 99–100, 203
 - splitting, 98–99, 299
- deprecated methods and techniques, 4
- `df` prefix for dataframes, 29
- dictionary objects, 130–134
 - accessing and modifying contents, 132–133
 - creating new key-value pairs, 133
 - curly brackets (`{}`), 41
 - declaring lists of dictionaries, 132
 - declaring single dictionaries, 106, 131
 - defined, 130
 - DNA within dataframes, 106
 - key-value pairs, 107, 131, 133
 - list of dictionaries, 132–133
 - source and definition, 14
 - storing in dataframes, 134
- directives for timestamp formatting
 - commonly used, 194–195
 - converting datetime objects to strings, 196–197
 - converting strings to datetime objects, 301
 - customizing results, 203
 - directive strings, 195, 196
 - functionality of, 189, 193
 - isolating units of time as strings, 197
 - percent sign (`%`), 195
 - timestamps, 194–195
- directories. *See* folders
- display controls, 111. *See also*
 - `print()` function
 - column number, 111
 - column width, 112
 - `display.precision` argument, 130
 - float precision, 113
 - `reset_option()` method, 111
 - rows displayed, 113
- distribution platform (Anaconda), 4
- division operator (`/`), 60
- dot notation, 32, 109
- double backslash (`\\`), 210, 286
- double quotation marks (`" "`), 13, 20–21, 21
- double square brackets (`[[]]`), 32, 41, 50, 113–114
- `drop` argument (`reset_index` attribute), 110
- `drop_duplicates()` method, 33, 35
- `drop()` method
 - cleaning up results, 153, 182
 - deleting columns, 51–52, 182, 300
 - dropping rows by index label, 119–120, 298
- `dropna()` method (subsetting dataframes), 124–125
- `dropna` parameter (`crosstab()` method), 153–154, 157, 163
- `dt` accessor (Datetime module), 190
- `dtypes` attribute, 62

- Ducks module (DIY), 258–271, 305–309
 - auto_email() UDF, 263–265, 267, 306
 - downloading script, 258, 306
 - importing, 258, 306
 - pause() UDF, 266–267, 269, 308
 - set_date() UDF, 259–260, 308
- duplicates
 - dataframes, 33, 35
 - index labels, 30, 37
 - lists, 42–43

E

- elif statements
 - control flow structure, 68
 - selecting menu options, 85
- ellipsis (...)
 - continued input lines, 5
 - hidden and truncated data, 128–129
- else statements, 68
- emailing from Python, 244–248
 - auto_email() UDF (Ducks module), 263–265, 267–269, 269, 306
 - automating email, 263–265, 267–269, 269, 306
 - converting dataframes to HTML code, 245–246
 - sending basic email, 244–245
 - sending email containing HTML, 246–247, 247
- equality operator (==), 60
 - comparing lists, 44–45
 - comparing values, 67
 - matching values in columns, 120
- error messages
 - Excel file permission error, 232–233, 244, 27
 - FileExistsError: [WinError 183], 289
 - merge() method errors, 182, 184, 185
 - pandas.errors.MergeError: No common columns, 182
 - SettingWithCopyWarning, 114
 - SyntaxError: EOL while scanning string literal, 20–21
 - TypeError: can only concatenate str..., 16–17
 - value_counts() method Traceback error, 142

- escape character (\), 19, 210, 286
- escape sequences, 18–20, 210, 219, 286–287
- Excel, 223–248. *See also* automation
 - cleaning up messy spreadsheets, 295–303, 303
 - changing specific cell values, 300–301
 - converting strings to datetime objects, 301
 - customizing column labels, 297
 - dropping records by index label, 298
 - exporting cleaned-up dataframes back to Excel, 301–302
 - formatting clean Excel files, 302
 - reading into dataframes, 296–297, 296
 - renaming columns, 299–300
 - reordering columns, 300
 - sorting by index label, 298
 - specifying columns as indexes, 297–298
 - splitting columns, 299
- functionality replicated within Python
 - AVERAGE, cross-tabulation, 61, 150
 - COUNTIF, 140–141, 145, 147, 151
 - custom date feature, 193–194
 - data filter, 219, 224, 234–236
 - freeze frame, 219, 224, 228, 231
 - justification, 239
 - LEFT, 88–89
 - macros, 219, 252
 - MAX, 149
 - MEDIAN, 150
 - MIN, 149
 - #N/A and #NAME errors, 86, 147
 - pivot tables, 159–160
 - RIGHT, 88–89
 - SUM, 147–148, 152
 - SUMIF, 71, 147–148, 151
 - VLOOKUP, 144, 148, 151, 152
 - wrap text, 236
- exporting dataframes to Excel, 228–244, 229
 - adding multiple dataframes to one workbook, 231
 - closing workbook objects and Excel files, 243–244, 244

- Excel (*continued*)
 - exporting dataframes to Excel
 - (*continued*)
 - closing writer objects and Excel files, 232, 232
 - complex dataframes, 225–226, 226
 - creating workbook objects and Excel files, 233–234
 - creating writer objects and Excel files, 230–231
 - designating tab names, 226–227, 227
 - excluding dataframe index, 227, 227
 - freezing the view, 228
 - simple dataframes, 224, 224
 - single tabs, 224–228
 - formatting, 234–243
 - borders, fonts, and other styles in columns, 241–242, 242
 - column width, 238–239, 239
 - combining formatting, 242–243
 - data filters, 235–236, 236
 - formatting datetime objects in single column, 240–241, 241
 - horizontal justification for single row, 239–240, 240
 - wrapping text, 236–238, 237, 238
 - importing Excel files into dataframes, 209–222
 - complex spreadsheets, 216–219, 216
 - creating or downloading spreadsheets, 210, 210
 - filtering source data, 214
 - importing all tabs at once, 213
 - importing specific tabs from workbooks, 212–213
 - parsing input spreadsheets, 214–219, 215
 - reading in CSV files, 220
 - Excel function replication
 - VLOOKUP formula, 167–172, 168, 171, 174, 175, 176, 183, 186
 - Excel–Python–Excel workflow
 - automation, 249–283
 - cleaning up messy spreadsheets, 295–303, 303
 - exporting dataframes to Excel files, 223–248
 - reading Excel files into dataframes, 209–222
 - ExcelWriter() method, 229
 - adding multiple dataframes to one workbook, 231
 - creating and populating workbook objects and Excel files, 233
 - creating writer objects and Excel files, 230
 - exception reports, 262–264
 - extend() function (lists), 43
- F**
 - %f directive (microseconds), 195
 - File Explorer, Microsoft
 - OS module, 267, 284–294
 - pathnames, 286–287, 287
 - shell utilities, 285
 - files
 - copying, 290–291
 - deleting, 292
 - moving, 291–292
 - quick reference for managing, 294
 - renaming, 291–292
 - fillna() method (dataframes), 87–88, 148, 155
 - fill_value parameter (pivot_table() method), 157, 163–164
 - filter formatted on Excel output file, 234–236
 - auto_filter.ref parameter, 234–236, 242, 270
 - filtering dataframes. *See* *subsetting dataframes*
 - floats
 - controlling display of, 130
 - converting, 79–82, 81
 - float() function, 79, 84
 - printing, 14–15
 - rounding columns or dataframes, 63
 - source and definition, 14
 - storing calculated values in new columns, 66
 - floor() method (Datetime module), 268

- folders
 - creating new, 289
 - deleting, 293
 - listing contents of, 288
 - moving, 291–292
 - OS module, 267, 284–294
 - renaming, 291–292
 - shutil module, 281–282, 293
 - viewing and changing working
 - directory, 287–288
 - Font class (Excel formatting), 241–242
 - for loops
 - analyzing merge results, 178
 - counters, 69–70
 - creating dictionary key-value pairs, 133
 - declaring lists of dictionaries, 132
 - iterating through lists, 69
 - normalizing `crosstab()` data, 155
 - storing dictionaries in dataframes, 134
 - forward slash (/), 210, 286
 - `freeze_panes` parameter (`to_excel()` method), 228, 231, 302
 - functions. *See also* aggregate functions
 - attributes, comparison with, 32
 - case sensitivity, 22
 - defined, 32
 - methods vs., 106
 - native functionality, xxiv, 4, 13–14
 - user-defined functions (UDFs), 252–275
- G**
- graphical user interface (GUI)
 - `input()` function, 82–85
 - Spyder, 84
 - greater than operator (>), 60, 123
 - greater than or equal to operator (>=), 60, 123
- H**
- hard returns
 - artifact, removal from Excel input, 218–219
 - break lines, 148
 - in code, 18, 28, 53
 - line continuation, 206
 - in source Excel file, 216, 219
 - hash mark (#)
 - comment lines, 254
 - removing padding zeros from dates, 197
 - `%H` directive (24-hour clock), 195
 - `head()` method
 - limiting number of rows to display, 129
 - printing first few rows, 125–126
 - `hello_world()` UDF, 253–254
 - horizontal parameter, 239–240
 - `how` parameter (`merge()` method), 173, 178, 185
 - HTML (hypertext markup language)
 - converting dataframes to, 244–246
 - `HTML_Object` parameter, 263
 - sending email containing, 246–247, 247
 - Hungarian notation, 29
- I**
- `iat[]` attribute (indexing), 95
 - IDEs (integrated development environments), 4–8. *See also* IDLE; Spyder
 - `%I` directive (12-hour clock), 195
 - IDLE (Integrated Development and Learning Environment), 3, 4–6
 - basic dataframe, 29
 - code appearance in, 27–29
 - error messages, 6
 - input/output, 5, 5
 - launching, 5
 - pros and cons of, 6
 - support for modules, 8
 - if logic and statements
 - control flow structure, 68
 - iterating through dataframes, 69
 - iterating through lists, 69
 - list comprehensions, 72
 - selecting menu options, 85
 - `iloc[]` attribute (indexing), 93–96
 - functionality of, 92–97
 - pulling values
 - by index position, 93–94
 - from series, 95–96
 - targeting single values, 95
 - by unique index label, 94
 - replacing individual cell values, 96
 - subsetting dataframes, 115–118

- importing Excel files. *See* Excel files,
 - importing, *under* dataframes
- importing modules
 - accessors, 29, 257
 - import keyword, 26, 29
- index_col parameter (read_excel() method), 218, 297–298, 303
- indexes in dataframes, 93–97, 109–113
 - concatenating dataframes by column
 - different index values, 39–40
 - same index values, 38–39
 - creating dataframes with specific
 - index labels, 30–31
 - defined, 26
 - deleting, 110
 - dropping rows by index label,
 - 119–120, 298
 - duplicate labels, 30, 37
 - excluding when exporting single
 - tabs, 227, 227
 - index.name attribute, 109
 - modifying existing cells, 96–97
 - moving values between columns and
 - indexes, 112–113
 - MultiIndexes, 147, 151, 163
 - naming, 109
 - negative indexing, 89, 91–92
 - pulling values
 - by index position, 93–94
 - from series, 95–96
 - targeting single values, 95
 - by unique index label, 94–95
 - renaming, 109
 - renumbering, 110–111
 - reordering columns in dataframes, 51
 - resetting, 110–111
 - sequential or range, 26
 - setting Excel columns as dataframe
 - indexes, 218–219
 - slicing strings and lists, 88–89
 - sorting by, 111–112, 298
 - specifying columns as, 297–298
 - subsetting dataframes
 - by index label, 118–119
 - by relative index location, 116–117
 - zero-based indexing
 - adding rows at end of dataframes, 97
 - defined, 26
 - excluding index when exporting
 - single tabs, 227
 - handling inconsistent
 - delimiters, 100
 - importing specific tabs from
 - workbooks, 213
 - index parameter vs., 108
 - pulling values by index position, 93
 - setting column labels, 218
 - slicing strings and lists, 88–89,
 - 91–92
 - specifying columns as indexes, 297
 - subsetting dataframes, 116–117
- indexing, 92–97
- index parameter
 - crosstab() method vs. pivot_table() method, 157
 - excluding indexes from exports,
 - 227, 227
 - explicitly defining, 145–146, 149
 - functionality of, 108–109
- indicator parameter (merge() method),
 - 172, 174, 177
- info() method, 62
- inner joins (merge() method), 175,
 - 176–177, 176, 179, 184, 186
- inplace argument (reset_index attribute), 110
- input() function
 - entering email addresses, 265
 - functionality of, 82–83
 - manually inserting values with, 82–85
 - answering questions and saving
 - answers, 83
 - pausing execution to review
 - output, 85
 - selecting menu options, 84–85
 - menus, 84–85
 - pausing execution, 85, 266
 - rolling over files, 259
 - updating and formatting Excel files
 - in Spyder, 273
- integers
 - converting, 79–82, 81
 - defined, 13
 - isolating units of time as, 191–192
 - printing, 14–15
 - source and definition, 14

Integrated Development and Learning Environment. *See* IDLE

integrated development environments (IDEs), 4–8. *See also* IDLE; Spyder

int() function, 63, 79–80, 84, 200, 277

isin() method

- filtering values before counting, 142
- subsetting dataframes
 - by column labels in lists, 122
 - by exclusion, 123
 - by inclusion, 122–123

isnull() method (NaN objects), 124–125

iteration

- through dataframes, 70–71
- defined, 69
- iterrows() method
 - iterating through dataframes, 70
 - storing values in new columns, 72
- through lists, 69

J

%j directive (day of year), 195

joins (merging)

- analyzing merge results, 177–179
- defining keys, 179–181
- merge() method, 167–188
- returned records, 176
- types of, 175–176, 175, 176

K

key-value pairs

- dataframe structure, 107
- dictionary objects, 131, 131, 133
- renaming columns, 53
- replacing particular values in
 - dataframes and target columns, 55–57

L

leading zeros (padding in dates), 195, 197

left joins (merge() method), 175, 176, 176

- left_index parameter, 173
- left_on parameter, 173, 182
- left parameter, 172, 173

len() function

- calculating length of hyphens, 274
- counting number of elements in lists, 42
- counting rows, 33, 97

less than operator (<), 60, 123

less than or equal to operator (<=), 60, 123

line continuation, 18, 28, 205

lists, 40–46

- adding lists to, 43
- appending single items to, 43
- comparing, 44–45
- converting columns into lists, 46
- converting lists into series, 46–47
- counting a list of specific values in a column, 141–142
- counting number of elements in, 42
- creating, 42
- creating empty dataframes with, 45–46, 108
- declaring lists of dictionaries, 132
- DNA within dataframes, 40
- functionality of, 40
- identifying minimum, maximum, and mean values, 44
- isolating unique values, 42–43
- iteration through, 69
- list comprehensions, 72–73
- pulling select elements from, 92–93
- removing elements of, 44
- slicing lists from lists, 91–92
- sorting, 43–44
- source and definition, 14
- square brackets, 41
- subsetting dataframes by column list, 113–117

load_workbook() method (Excel export), 229, 233–234, 302

loc[] attribute (indexing), 93–97

- absolute location, 93
- adding rows at end of dataframes, 97
- functionality of, 93

loc[len()] construct, 97

pulling values

- from series, 95–96
- by unique index label, 94

replacing individual cell values, 96

storing returned values in new columns, 72

subsetting dataframes, 118–119

M

- margin labels in `crosstab()` method
 - `margins_name` parameter, 153, 157
 - `margins` parameter, 152–154, 157
- math methods and operators, 60–68
 - average, maximum, or minimum values, 61
 - comparing values, 67–68
 - converting to absolute values, 64–65
 - cross-tabulation, 147–152
 - list of, 60
 - median values, 61
 - rounding, 63–64
 - storing calculated values in new columns, 66–67
 - stringing together multiple methods, 65
 - `sum()` method, 60–61
- `max_colwidth` option (`dataframe` display), 128–129
- maximum, finding
 - `max` argument (aggregate function), 149
 - MAX formula (Excel), 149
 - `max()` function, 44, 60–61, 65
 - `max()` method
 - calculating column maximum value, 61
 - identifying maximum list values, 44
 - subsetting dataframes by mathematical condition, 124
- McKinney, Wes, 26, 130
- `%m` directive (number of month), 194, 195
- `%M` directive (minutes), 195
- mean, calculating
 - `mean` argument (aggregate function), 150, 158–161
 - `mean()` function
 - calculating column average value, 61
 - identifying mean list values, 44
- median, calculating
 - `median` argument, 150
 - MEDIAN formula (Excel), 150
 - `median()` method, 61
- menu design and usage, 84–85
- `merge()` method, 167–188
 - alternative calling method, 170
 - complicated nature of, 167
 - `concat()` method vs., 168
 - error statements, 182, 184, 185
 - orphaned keys, 170–172, 171
 - quality control with shape attribute, 185
 - syntax, 172–185
 - analyzing merge results, 177–179
 - checking match expectations, 183–185, 183
 - defining keys, 179–181
 - handling different key column labels, 181–182
 - parameters, 173–174
 - specifying join type, 174–177, 175, 176
 - VLOOKUP formula vs., 167–172, 168, 174, 175, 176, 183, 186
- merging dataframes, 167–188. *See also* `merge()` method
 - basics of, 168–170
 - orphaned keys, 170–172, 171
- `pandas.errors.MergeError: No common columns`, 182
- quality control with shape attribute, 185
- syntax, 172–185
 - analyzing merge results, 177–179
 - checking match expectations, 183–185, 183
 - defining keys, 179–181
 - handling different key column labels, 181–182
 - parameters, 173–174
 - specifying join type, 174–177, 175, 176
- messy spreadsheet cleanup, 100, 295–303
- methods, 32
 - accessors, 29, 257
 - functions vs., 32, 106
 - modules, 27, 29
 - stringing together multiple methods, 65
 - user-defined functions (UDFs), 252–275

- Microsoft
 - curly quotation marks, 21
 - Excel, *xxi*
 - File Explorer, 210, 231, 285–287
 - pathnames vs. Python, 210, 231, 285–287
 - Word, 210, 236
 - VB, *xxiii*
- minimum, finding
 - `min` argument (aggregate function), 149
 - `min()` function, 44, 60–61, 65
 - calculating column minimum value, 61
 - identifying minimum list values, 44
 - MIN function (Excel), 149
 - `min()` method, 65
 - subsetting dataframes, 124, 204–205
- modularity, 252
- modules, *xxiv*
 - Datetime (`dt`), 189–205
 - Do-It-Yourself (DIY) modules, 256–257, 305
 - Ducks (DIY), 258–271, 305–309
 - importing, 8, 15, 26–27, 256–258
 - modularity, 252
 - Numpy (`np`), 44, 61, 86–87, 124, 141, 153, 257
 - OS (`os`), 225, 287–294
 - pandas (`pd`), *xxiv*, *xxvii*, 25–48, 105–136
 - `shutil` (`shutil`), 257, 261, 286, 290–291, 293
 - statistics (`st`), 61
- Monty Python, *xxiii*, 5
- moving
 - files, 291–292
 - folders, 291–292
 - values between indexes and columns, 112–113
- MultiIndexes, 147, 151, 163
- multimetric tables, 150–151, *151*
- multiplication operator (`*`), 60, 66

N

- `\n` (escape sequence for newline), 18–19
- #N/A Excel error message
 - merging vs. VLOOKUP, 179
- NaN objects
 - differences from, 86, 153
 - equivalence to, 38, 54, 77
- naming
 - columns, 52–53, 181, 218, 299–300
 - dataframes, 29–30
 - files, 291–292
 - folders, 291–292
 - indexes, 109
- NaN (Not a Number) objects, 86–88
 - complex spreadsheets, 217
 - concatenation, 38–40
 - `crosstab()` method and missing values, 153–155
 - handling and replacing, 87–88, 148, 153–155, 163–164, 300–301
 - joins, 176, 179
 - knowing your data, 22
 - manually creating, 86
 - manually entering into dataframe, 86–87
 - messy spreadsheets, 296–297, 300–301
 - #N/A Excel error message, 35
 - `pivot_table()` method and missing values, 162–163
 - sorting dataframes by column values, 54
 - source and definition, *14*
 - subsetting dataframes by, 124–125
- native functionality, *xxiv*, *xxvi*, 4, 8, 13–14, 26, 32, 41, 46, 60–63, 79, 83, 98
- negative indexing, 89, 91–92. *See also* zero-based indexing
- normalization
 - converting time units to standard format, 199
 - defined, 143
 - normalize parameter
 - `crosstab()` method results, 155–156
 - `value_counts()` method results, 143, *143*
- Not a Number objects. *See* NaN objects

- not equal to operator (`!=`), 60, 121
- `nrows` parameter (`read_excel()` method), 214
- `number_format` attribute (Excel output), 240–241
- NumPy library, 14, 44, 61, 86
 - importing, 44
 - lists, unique elements, 264
 - `median()` method for dataframes, 61
 - NaN objects, 86, 124, 153–154
 - `np` accessor, 44

O

- observed parameter (`pivot_table()` method), 157
- `on` parameter (`merge()` method), 173, 179–181
- OpenPyXL module
 - common object types, 14
 - exporting and formatting Excel files, 229–238, 240, 242, 302
 - importing, 210–211
 - `op` accessor, 211, 238
- orphaned keys when merging, 170–172, 171
- OS module, 267, 284–294
 - importing, 257, 287
 - `os` accessor, 257, 287
 - `os.chdir()` function, 287–288, 294
 - `os.getcwd()` function, 287, 294
 - `os.listdir()` function, 288, 294
 - `os.mkdir()` function, 289, 294
 - `os.path.exists()` function, 289–290, 294
 - `os.path.isdir()` function, 290, 294
 - `os.path.isfile()` function, 290, 294
 - `os.remove()` function, 292, 294
 - `os.rename()` function, 261–262, 291, 294
 - `os.rmdir()` function, 293, 294
 - `shutil` module, 281–282, 293
- outer joins (`merge()` method), 175, 176, 176, 179
- overdue invoice process, 264–266, 266
- overwriting
 - dataframes, 28, 35
 - Excel file output (no warning), 225, 230
- folder deletion, prevention of, 244
- index in dataframe, 110–111
- overwrite warnings
 - copy file, 291
 - create new folder, 289
 - delete file (none), 293
 - file access conflicts, 225, 230, 292
 - `pd.ExcelWriter()` (none), 230
- no warning, 98

P

- padding (leading zeros in dates), 195, 197
- pandas module, xxiv, xxvii. *See also* dataframes
 - common data and object types, 14
 - failure to preserve some elements from Excel source files, 219
 - importing, 26, 29
 - origin of, 26
 - `pd` accessor, 29
- parameters, 18, 146
- parentheses (`()`), 41
 - creating dataframes, 27–28
 - default settings, 31
 - display text, 12
 - equations, 15
 - escape characters, 19
 - functions and methods, 32
- pathnames
 - checking existence of, 289–290
 - in File Explorer vs. Python, 286–287
 - pathname convention, 211
- parameters, 157
- `%p` directive (AM/PM), 195
- percent sign (`%`), 195
- pivot tables
 - Excel pivot table replications, 144, 157–160, 159, 160
 - `pivot_table()` method
 - basic form of, 158–161
 - calculating math values, 161–163
 - `crosstab()` method vs., 157–163, 157
 - example, 280–282
 - grouping unique values, 161
 - handling missing values, 163–164
 - organizing dataframe values, 164

- print() function, 5, 11
 - dataframe display controls, 125–130
 - data types, 78–79
 - empty lines, 18
 - escape characters and escape sequences, 19–20
 - functionality of, 12
 - lists of column labels and their data types, 62
 - numbers, 14
 - numeric variables, 14–15
 - parameters, 18–20
 - select columns, 54–55
 - spellcheck feature, 12
 - strings of strings, 13
 - string variables, 12

Python

- add-on modules, xxiv, xxvii
- attributes, 32
- common pitfalls, 20–22
 - ignoring case sensitivity, 22
 - inconsistent use of quotation marks, 20–21
 - not knowing your data, 22
- dataframes, xxi, xxvii
- Excel, compatibility with, xxiii–xxiv
- functions, 32
- methods, 32, 106
- modules, role of, xxiv, xxvii
- Monty Python, xxiii, 5
- native functionality, xxiv
- object-oriented language, xxiv, 4, 13, 78
- open source, xxiii–xxiv
- origin of, xxiii, 26
- pathnames, 286–287
- Python in Excel, 5
- quick reference, 311–312
- technical considerations, 4–8
 - distribution platforms, 4
 - IDEs, 4–8
 - versioning, 4
- user-friendly nature of, xxiii, 26
- van Rossum, Guido, xxiii, 26

Q

- quality controls
 - DIY, 185–186
 - merge() method, 177–179, 183

- shape attribute, 31, 185
- try-except blocks, 73–74
- quotation marks
 - blank values, 13
 - curly quotation marks, 21
 - directive strings, 195
 - printing, 19
 - single vs. double consistency, 13, 20–21, 21
 - strings, 12–13

R

- read_csv() method, 220
- reader engines (Excel import), 211, 220
- read_excel() method, 210–214
 - complex spreadsheets, 216–219, 216
 - handling hard returns in Excel data, 219
 - setting column labels, 217–218
 - setting Excel columns as dataframe indexes, 218–219
 - filtering source data, 214
 - importing all tabs at once, 213
 - importing specific tabs from workbooks, 212–213, 212
 - parsing input spreadsheets, 214–215, 215
 - reading messy spreadsheets into dataframes, 296–297, 296
 - specifying columns as indexes, 297–298
- records. *See* rows
- remove() function (lists), 44
- renaming
 - columns, 52–53, 181, 218, 299–300
 - files, 291–292
 - folders, 291–292
 - indexes, 109
 - rename() method, 52–53, 181, 218, 299–300
- replace() method
 - handling hard returns in Excel data, 219
 - replacing particular values in dataframes and target columns, 55–57
 - replacing substrings, 57
- reports, 262–266

- reset_index() method
 - renumbering/resetting indexes, 110–111
 - subsetting dataframes, 268
- reset_option() method (dataframe display), 127–128, 127, 130
- return keyword (UDFs), 254
- return statements (UDFs), 254–255
- right_index parameter, 173
- right_joins (merge() method), 175, 176, 176
 - right_on parameter, 173, 182
 - right parameter, 172, 173
- rolling over files, 257–262
 - creating new file for today's date, 260–262
 - importing favorite modules, 257
 - importing UDFs, 258
 - printing headers, 258
 - setting dataframe display preferences, 258–259
 - setting date, 259–260
- rounding
 - controlling how floats are displayed, 130
 - round() function (values), 63
 - round() method (dataframes), 63–64
- rows (records), 26
 - adding at end of dataframes, 97
 - concatenating dataframes by different columns, 38
 - same columns, 36–37
 - counting
 - using len() function, 33, 97
 - using shape attribute, 34
 - deleting with specific values, 33, 34–35
 - dropping by index label, 119–120, 298
 - horizontal justification for single, 239–240, 240
 - identifying and dropping duplicated, 35
 - indexing, 92–97
 - limiting number to display, 129
 - printing
 - first or last few, 125–126
 - specific, 126
 - subsetting dataframes
 - by index label, 118–119
 - by relative index location, 116–117

S

- save() method (Excel export), 302
- scripts, 6, 20
 - commenting within, 20
 - display standardizing, 127
 - Do-It-Yourself modules and UDFs, 256
 - hard returns and line indentation, 50
 - saving in Spyder, 7
 - temporary lines, 62
- %S directive (second), 195
- Send() method (emailing), 245
- series objects
 - converting lists into, 46–47
 - dataframes, DNA component of, 46
 - defined, 26, 40
 - pulling values from, 95–96
 - Series() method, 46, 96
 - source and definition, 14
- set() function (for unique lists), 42–43
- set_index() method
 - converting preexisting columns to indexes, 112–113
 - specifying columns as indexes, 297–298
- set_option() method (dataframe display), 127
- shape attribute
 - counting rows and columns, 34
 - testing whether record counts match, 185–186
- sheet_name parameter (Excel files), 212–213, 226, 227, 231, 302
- sheetnames attribute (Excel export), 233
- sheet objects (Excel export), 229, 234
- shell utilities, 286, 290–291, 293
- shutil module
 - OS module, 267, 284–294
 - shutil.copy() function, 261, 291, 294
 - shutil.rmtree() function, 293, 294
- single quotation marks (' ')
 - blank values, 58
 - directive strings, 195
 - printing, 19

- strings, 12–13
 - using consistently, 13, 20–21, 21
- skipfooter parameter (`read_excel()` method), 214
- slicing, 88–93
 - defined, 88
 - within entire columns, 90–91
 - lists from lists, 91–92
 - pulling select elements from lists, 92–93
 - single strings, 88–89
- sorting
 - ascending parameter, 53–54, 111, 298
 - `sorted()` function
 - slicing lists from lists, 92
 - sorting lists, 43–44
 - `sort_index()` method, 111–112, 298
 - sort parameter, 157, 164, 174
 - `sort_values()` method
 - sorting dataframes by column values, 53–54
 - sorting rows, 179
- `split()` function
 - functionality of, 98
 - handling inconsistent delimiters, 99–100
 - splitting columns, 100–101, 299
 - splitting single strings, 98–99
- Spyder, 3–8
 - Anaconda, launching from, 6
 - arguments, 107
 - basic dataframe, 29
 - code appearance in, 28, 29
 - Console pane, 6–7, 7
 - as daily workflow GUI, 282, 282
 - Editor pane, 6–7, 7
 - error messages, 16–17, 17, 20, 21
 - launching, 6
 - manually inserting values, 84
 - open source, 6
 - saving UDFs in separate scripts, 256
 - support for modules, 8
 - working directory, 288, 288
- square brackets (`[]`), 41
 - attributes, 32
 - concatenating dataframes, 37
 - defining columns, 27, 50
 - double square brackets (`[[]]`), 32, 41, 50, 113–114
 - list comprehensions, 73
 - lists, 40, 42, 51, 91, 123, 132
 - slicing, 88, 90
- Star Trek*, 193
- statistics module
 - importing, 61
 - `median()` method, 61
 - `st` accessor, 61
- `str` accessor (`replace()` method), 57
- `str[]` attribute (slicing), 90
- `strftime()` method, 189. *See also* dates and times
- strings
 - concatenating, 12
 - converting between integers, floats, and strings, 79–82, 81
 - converting datetime objects to, 193–197
 - converting single objects to custom-formatted strings, 196–197
 - isolating units of time as strings, 197
 - removing leading zeros from single-digit time elements, 197
 - transforming timestamps, 193–196, 194–195, 196
 - converting to datetime objects, 301
 - printing strings of strings, 13
 - printing string variables, 12
 - replacing substrings, 57
 - slicing, 88–89
 - source and definition, 15–16
 - splitting, 98–99
 - stringing together multiple methods, 65
 - `strftime()` method, 189
 - `str()` function
 - converting objects to strings, 15–16, 45, 82
 - functionality of, 79
 - `str[]` attribute vs., 90
 - subsetting dataframes by substring values, 121–122
- subsetting dataframes, 113–125
 - by column labels in lists, 122–123
 - by column lists, 113–117

- subsetting dataframes (*continued*)
 - by excluding values in columns, 121
 - by exclusion in list, 123
 - by exclusion using not equal to operator (!=), 35, 71, 121
 - by inclusion in list, 122–123
 - by index label, 118–119
 - by matching values in columns, 120–121
 - by mathematical conditions, 123–124
 - by NaN objects, 124–125
 - by relative index location, 116–117
 - by substring values, 121–122
 - tilde (~) for the opposite, 122–123, 125
 - by using datetime objects, 204–205
- substrings
 - replacing, 57
 - splitting strings into, 98–99, 299
 - subsetting dataframes by substring values, 121–122
- subtraction operator (-), 60
- suffixes parameter (merge() method), 174, 181
- summation
 - SUM formula (Excel replication), 152
 - SUMIF formula (Excel replication), 71–72, 147, 151, 151
 - sum() method
 - applying to columns, 60–61
 - replicating SUMIF formula with, 71–72

T

- \t (escape sequence), 19
- tail() method
 - limiting number of rows to display, 129
 - printing last few rows, 125–126
- text functions, replicating Excel formulas
 - concatenating strings, 12–13, 59
 - LEFT (slicing), 88–89
 - RIGHT (slicing), 88–89
- tilde (~) for the opposite, 122–123, 125
- timedelta objects
 - calculating in new column, 203–204
 - comparing and calculating dates and times, 199–200
 - source and definition, 14

- timedelta() method, 198–199
- time durations, 198–199
- timestamps, 193–196, 194–195, 196.
 - See also* datetime objects
- to_datetime() method
 - converting strings to datetime objects, 301
 - functionality of, 189, 202–203
 - using directives to customize results, 203
- to_excel() method, 223–244, 302
 - freeze frame, 219, 224, 228, 231
 - index exclusion, 172–174, 227
 - open file errors, 225
 - overwrites, 225
- tab export
 - multiple, 228–233
 - single, 224–226
- to_html() method, 245–246
- tolist() method (dataframes), 46
- to_list() method (series splitting), 101, 264
- to_timedelta() method, 274
- Traceback statements, 16, 142
- trend analysis, 281
- try-except blocks, 73–74
- tuples
 - lists, comparison to, 34
 - parentheses, 41
 - shape attribute returning data as, 34
 - source and definition, 14
- TypeError statements. *See* error messages
- type() function, 12, 14, 78

U

- unique values
 - drop_duplicates() method (dataframes), 33–35
 - index labels, not required, 26, 30
 - lists, 42–43, 264
- unpacking sequences, 255
- user-defined functions (UDFs), 252–275
 - defined, 252
 - defining required and optional parameters, 253–254
 - Ducks module, 305–309
 - importing UDFs, 258

- returning one or more values, 254–255
- saving in separate scripts, 256–257
- writing and calling, 252–253

V

- validate parameter (`merge()` method), 174, 183–185
- value_counts() method, 140–143
 - counting a list of specific values in a column, 141–142
 - counting every value in a column, 140–141
 - counting specific values in a column, 141
 - crashing execution, 142
 - normalizing results, 143
- values parameter
 - `crossstab()` method, 147–149
 - `pivot_table()` method, 157, 159, 163
- van Rossum, Guido, xxiii, 26
- variables
 - defined, 12
 - printing numeric variables, 14–15
- versioning, 4
- vet's office scenarios, 250–252
 - Ducks module, 258, 305–310
 - New_App.py*, 269–275
 - Overdue.py*, 264–266
 - Pivot_Analysis.py*, 280–282
 - Rollover.py*, 260–262
 - Today.py*, 267–269
 - Update.py*, 276–279
- virtual Excel spreadsheets (dataframes), xxi, xxvii, 27–28, 33
- Visual Basic (VB), xxiii
- VLOOKUP formula, 167–172, 168, 171, 174, 175, 176, 183, 186
- keys, 168
 - defining, 179–181
 - handling different key column labels, 181–182
 - orphaned, 170–172, 171
- `merge()` method, differences from, 174, 183, 186
- `merge()` method, similarities to, 73, 168–170

W

- %w directive (numbered day of week), 195
- website, author's (<https://www.johnwengler.com>), xix, 210, 251, 258, 296, 306
- where() statements, 71–72
- while loops, 69–70
- workbook objects (Excel export)
 - closing, 243–244, 244
 - creating and populating, 233–234, 302
 - formatting Excel files, 235
 - replicating Excel processes, 229, 229
 - source and definition, 14
- workflow design, 45, 250. *See also* Excel–Python–Excel workflow
- working directory, viewing and changing, 287–288
- working environments, 4–8
 - Anaconda, packaged by, 4
 - IDLE, 5–6
 - Spyder, 6–8
- worksheet objects (Excel export), 235
- wrap_text parameter (Excel export), 236–238, 237, 240
- writer engines (Excel export), 230
- writer objects (Excel export)
 - adding multiple dataframes to one workbook, 231
 - closing, 232, 232
 - creating, 230–231
 - exporting dataframes to Excel, 302
 - replicating Excel processes, 228, 229
 - source and definition, 14

Y

- %y directive (YY), 194
- %Y directive (YYYY), 194

Z

- zero-based indexing. *See also* indexes
 - adding rows at end of dataframes, 97
 - defined, 26
 - Excel counting, differences from, 26, 88, 93, 213, 218
 - excluding index when exporting single tabs, 227

- zero-based indexing (*continued*)
 - handling inconsistent delimiters, 100
 - importing Excel tab by relative location, 213
 - index parameter vs., 108
 - negative indexing, 89, 91–92
 - pulling values by index position, 93
 - setting column labels, 218
 - slicing strings and lists, 88–89, 91–92
 - specifying columns as indexes, 297
 - start counting with a 0, 26
 - subsetting dataframes, 116–117