

INDEX

Symbols

- <- assignment operator, 11
- #| (hash pipes), 160
- () (parentheses), 7–8
- ... syntax, 207–209

A

- Aden-Buie, Garrick, 201, 218
- aesthetic properties
 - altering, 29–31, 34–36
 - mapping data to, 25–26, 33–34
- albersusa package, 68, 70, 79
- alignment
 - of content in slides, 129–131, 165–167
 - in tables, 90–92
- American Community Survey data, 194–199
- Analyzing US Census Data: Methods, Maps, and Models in R* (Walker), 189, 199
- application programming interface (API) keys, 188–189
- arguments
 - adding to custom functions, 203–204
 - in functions, 8
 - passing from one function to another, 207–209
- arithmetic operators, 6–7
- as.character() function, 185
- asides, 140
- automatically accessing online data, xxii, 181, 199
 - American Community Survey data, 194–199
 - Decennial Census data, 189–194

- with googlesheets4 package, 182–188
 - with tidycensus package, 188–189
 - automating report production.
 - See parameterized reporting
 - axes
 - in bbplot package, 55
 - mapping data to aesthetic properties, 25–26
 - tweaking appearance of, 35
- ## B
- background images, adding to slides, 131, 167
 - background in bbplot package, 56–57
 - bar charts, 29, 33–34. *See also* data visualization
 - base R, 12
 - BBC custom theme (bbplot package), 47–48, 59
 - axes, 55
 - background, 56–57
 - bbc_style() function, 48, 51–58
 - color, 58–59
 - formatting text, 52–54
 - function definition, 51–52
 - grid lines, 56
 - installing bbplot, 50
 - legend, 54
 - small multiples, 57–58
 - styling plots with, 48–51
 - bounding boxes, 66
 - boxr package, 153
 - Bryan, Jenny, 154, 215, 217
 - Build Website option, RStudio, 140

C

- Cascading Style Sheets (CSS)
 - applying to slides, 131–135
 - custom CSS
 - applying to slides, 132–133, 167, 168–169
 - applying to website, 141–143, 172–174
- case_when() function, 71
- cells_body() function, 94
- cells_column_labels() function, 90
- cell_text() function, 90
- census_api_key() function, 188–189
- Census Bureau data
 - accessing, 188
 - American Community Survey data, 194–199
 - connecting to with API key, 188–189
 - Decennial Census data, 189–194
- center alignment
 - in slides, 129, 167
 - in tables, 90–91, 92
- .center[] content class, 129–131
- Çetinkaya-Rundel, Mine, 16
- c() function, 8–9, 15, 191
- chartjunk, 20, 23
- charts. *See* data visualization
- clean_names() function, 144, 184, 206, 213
- cloud hosting, 153–154
- clutter
 - chartjunk, 20, 23
 - in graphs, 22–23
 - minimizing in tables, 87–89
- code chunks
 - in Quarto, 160–161
 - in R Markdown documents, 104–106, 109–111, 144
- color
 - bbplot package, 58–59
 - fill in data visualizations, 30–31, 45
 - intentional use in tables, 94
- colorspace package, 68–69, 78
- cols_align() function, 92
- columns, importing from Google Sheets, 187–188
- comma() function, 118
- command line, working with R on, 4
- comma-separated values (CSV) files, 9–10, 13–14
- comments, 16–17, 104
- comparison operators, 7–8
- console, RStudio, 5
- content, website
 - adding, 143–148
 - applying distill layouts to, 148
 - interactive, 148–153
- coordinate reference systems (CRSs), 66–67, 70, 81–82
- Council of State Governments Justice Center website, 155
- count() function, 49
- COVID-19 map, 61, 68
 - adding geospatial data, 73–74
 - calculating daily COVID-19 cases, 70–71
 - calculating incidence rates, 71–73
 - importing data, 69–70
 - importing packages, 68–69
 - making, 74–79
 - in New Zealand, xix–xx
- create_theme() function, 141
- CRSs (coordinate reference systems), 66–67, 70, 81–82
- crsuggest package, 81
- CSS. *See* Cascading Style Sheets
- CSV (comma-separated values) files, 9–10, 13, 14
- custom data visualization themes, xxi, 47–48, 59. *See also* BBC custom theme
 - styling plots with, 48–51
- custom fonts, 52
- custom functions, 201. *See also* packages
 - adding arguments, 203–204
 - formatting race and ethnicity data, 204–207
 - passing arguments to another function (... syntax), 207–209
 - writing simple functions, 202–203
- custom themes, 134
- cut() function, 72

D

data. *See also* automatically accessing online data; geospatial data analyzing with `tidyverse`, 14–16 importing to create map, 69–70 working with in R, 9–13

data frames, 11

- for creating drought visualization, 32
- for creating tables, 86–87
- vs. tibbles, 13
- wrangling geospatial data, 82–83

data visualization, xxi, 19–20, 45–46. *See also* custom data visualization themes; maps; programming with R; tables

- complete drought visualization code, 42–45
- drought visualization effectiveness, 20–23
- grammar of graphics, 23–25
- including parameters in code for, 118–119
- interactive, on websites, 150–153
- re-creating drought visualization, 32–42
- with `tidycensus` package, 195–196
- in website, 146–147
- working with `ggplot2` package, 25–32

Data Visualization (Healy), 17

Decennial Census data, 189–194

dependencies, package, 211–213

DESCRIPTION file, 216

`devtools` package, 210–211, 214–217

dimensions, geospatial data, 66

`dir_ls()` function, 154

`distill_article` format, 140

`distill` package, 138, 155, 157

- adding website content, 143–148
- applying custom CSS, 141–143
- applying layouts, 148
- building website, 140–141
- creating new project, 138
- hosting website, 153–155

- interactive content, 148–153
- project files, 138–140

`distinct()` function, 144

`div` tags, 165–167, 175

`dk` package, 209, 211–213, 217

documentation

- accessing, 17–18
- for creating packages, 214–215

`dplyr` package, 49, 183

Dropbox, sharing website through, 153–154

`drop_na()` function, 72

drought data visualization, 19–20, 45

- adding final polishes, 38–42
- altering aesthetic properties, 34–36
- effectiveness of, 20–23
- faceting, 36–38
- plotting one region and year, 32–34
- re-creating, 32–42

`DT` package, 148, 149

E

`element_functions`, 53

environment pane, RStudio, 5

examples, adding to documentation, 215

execute field, Quarto YAML section, 159–160

exported functions, 215

F

`facet_grid()` function, 36

faceting data visualizations

- drought data visualization, 36–38, 41, 45
- in maps, 76
- reducing clutter by, 23
- in websites, 146

`facet_wrap()` function, 58, 76

`fct_inorder()` function, 74

`fig-height` option, Quarto, 160–161, 163

`fig.height` option, R Markdown, 128

`file_show()` function, 203

files pane, RStudio, 5

`fill` aesthetic property, 29–31, 34, 59, 83, 198

`filter()` function, 16, 32, 145, 192

finalise_plot() function, 48, 53
 FiveThirtyEight website, tables on,
 96–97
 fmt_currency() function, 93
 fonts
 custom, 52–53
 in slides, 132–133, 135
 format field, Quarto YAML section, 159
 format() function, 74
 format section, 172
 fs package, 154, 202–203
 functions. *See also* packages
 accessing documentation for, 17
 accessing geospatial data with,
 79–81
 adding to arguments, 203–204
 adding to packages, 210, 217
 basic R syntax, 8–9
 creating, xxii, 201–209
 definition of in *bbplot* package,
 51–52
 exported, 215
 internal, 215
 referring to correctly in packages,
 213–214
 tidyverse package, 15–16
 writing simple, 202–203
Fundamentals of Data Visualization
 (Wilke), 85–86

G

gapminder package, 25, 86
 gdp data frame, 86–87
Geocomputation with R (Lovelace,
 Nowosad, and Muenchow),
 79, 81–82
 GeoJSON files, 79
 geom_col() function, 29, 33–34, 146
 geometric objects (geoms), 26–29, 40
 geom_line() function, 27–28
 geom_point() function, 27
 geom_rect() function, 40–42
 geom_sf() function, 62–63, 76, 198
 geospatial data, 62
 accessing, 79–81
 adding to COVID-19 map, 73–74
 bounding box, 66
 coordinate reference system, 66–67
 dimensions, 66
 geometry column, 62, 67
 geometry type, 62–66
 wrangling, 82–83
 Gerke, Travis, 201, 218
 get_acs() function, 194–199, 207, 213
 get_decennial() function, 189–191,
 193–194, 199
 ggplot2 package, 20, 25. *See also* custom
 data visualization themes
 altering aesthetic properties,
 29–31
 choosing geometric objects, 26–29
 complete themes, 38
 documentation for, 53
 entry themes, 31–32
 faceting, 36–38
 and grammar of graphics, 24–25
 mapping data to aesthetic
 properties, 25–26, 33–34
 and tidycensus package,
 195–198
 ggplot() function, 26, 33, 195, 198
 ggplotly() function, 150–153
 GitHub, sharing packages on, 217
 GitHub Pages hosting, 138, 154–155,
 175–176
 glimpse() function, 183
 googledrive package, 153
 googlesheets4 package, 181–182, 199
 connecting to Google, 182
 importing only certain columns,
 187–188
 reading data from sheet, 182–183
 using data in R Markdown,
 183–187
 grammar of graphics, 20, 23–25,
 31–32. *See also* data
 visualization
Grammar of Graphics, The (Wilkinson), 24
 grid lines, 56, 87–89
 Grolemond, Garrett, 16
 group_by() function, 71
 group_by(NAME) function, 192
 gs4_auth() function, 182
 gtExtras package, 86, 95–97
 gt() function, 88
 gt package, 86, 88–94, 97, 145

`gt_plt_sparkline()` function, 95–96
`gt_theme_538()` function, 96–97
Guibourg, Clara, 47, 59
`guide_legend()` function, 78

H

Happy Git and GitHub for the useR
(Bryan), 154, 217
Harris, Meghan, 182–187
hash pipe (`#|`), 160
Healy, Kieran, 17
hello.R file, 210
help resources, 17–18
Herman, Matt, 137–138, 148–149,
155, 188
Hill, Alison, 110–111
hosting, website, 153–155, 175–177
HTML documents
 slides as, 126
 for websites, 138, 154–155

I

IDE (integrated development
 environment), 4–5
`if_else()` function, 82–83
`import()` function, 32
importing data
 to create map, 69–70
 from CSV file, 9–10
 with `googlesheets4` package,
 182–188
 raw geospatial, 79
importing files into projects, 14
incidence rates, calculating, 71–73
incremental reveal in slideshows,
 128–129, 165
index.qmd file, 170–171
inline R code, in R Markdown,
 108–109, 117
`install.packages()` function, 12, 211
integrated development environment
 (IDE), 4–5
interactive tables, 97
interactive tooltip, 148, 150–153
interactive website content, 148–153
internal functions, 215
Ismay, Chester, 4, 12

J

janitor package, 144, 184, 206,
 212–213
JavaScript, 148
JavaScript Object Notation (JSON)
 format, 32

K

Karamanis, Georgios, 19–20, 32,
 34–42
Kim, Albert, 4, 12
KnitR package, 158, 169
`knitr::opts_chunk$set()` function, 106
knitting, 103–107, 109–111, 120

L

`labs()` function, 78
`lag()` function, 71
layouts, `distill` package, 148
left alignment in tables, 90–92
 `.left-column[]` content class, 130
legends
 `bbplot` package, 54
 drought data visualization, 42
license for packages, 215–217
line charts, 27–28
LINESTRING geometry type, 64
lists
 revealing incrementally in
 slideshows, 165
 in R Markdown documents, 107
`load_variables()` function, 190
Lovelace, Robin, 79, 81–82
1-screen-inset layout, 148
`lubridate` package, 144

M

Madjid, Abdoul, 61, 68–69, 83–84
`map()` function, 121
mapping data to aesthetic properties,
 25–26, 33–34
maps, xxi, 61, 83–84
 geospatial data
 accessing simple features,
 79–81
 primer, 62–67
 wrangling, 82–83

- maps (*continued*)
 - making, 79–83
 - re-creating COVID-19 map, 68–79
 - with tidy census package, 196–199
 - using appropriate projections, 81–82
 - in website, 145–147
 - margin() function, 53
 - markdown text, 106–108
 - mean() function, 8–9, 15
 - metadata, package, 215–217
 - meta descriptions, 139
 - middle content class, 131
 - Mock, Tom, 86, 90, 95
 - Moffitt Cancer Center, 201, 218
 - moon_reader output format,
 - R Markdown, 126
 - Mucciolo, Livia, 113
 - Muenchow, Jannes, 79, 81–82
 - MULTILINESTRING geometry type, 65
 - multiple reports, generating
 - simultaneously.
 - See* parameterized reporting
 - MULTIPOINT geometry type, 64
 - MULTIPOLYGON geometry type, 65–66
 - multi-tool workflow for reports, 101–102, 114
 - mutate() function, 82–83, 185, 192
- N**
- named lists, 121–122
 - NAMESPACE file, 215
 - ne_countries() function, 80
 - Nowosad, Jakub, 79, 81–82
- O**
- objects, saving data as, 11
 - online data. *See* automatically accessing online data
 - options in Quarto, 160–161
 - output directory for website, 140
 - output widths in Quarto websites, 175
- P**
- packages
 - adding functions, 210, 217
 - adding license and metadata, 215–217
 - checking with devtools, 211–212
 - creating, xxii, 201–202, 209–218
 - creating documentation, 214–215
 - dependencies, 211, 212–213
 - documentation websites associated with, 18
 - importing to create COVID-19 map, 68–69
 - installing, 12–13, 217
 - loading, 12
 - referring to functions correctly, 213–214
 - palmerpenguins package, 48–49, 103
 - parameterized reporting, xxi–xxii, 113, 124
 - best practices, 124
 - creating R script, 119–123
 - with Quarto, 161–163
 - report templates in R Markdown, 114–119
 - params variable, 121–122
 - parentheses (())
 - in functions, 8
 - using with arithmetic operators, 7
 - plotly package, 148, 150–153
 - plots. *See* data visualization
 - POINT geometry type, 63
 - points, adding to graphs, 27
 - POLYGON geometry type, 62–63
 - precision, 93
 - presentations. *See* slideshow presentations
 - print() function, 8
 - programming with R, xxi, 3, 18, 218
 - basic syntax, 6–9
 - comments, 16–17
 - help resources, 17–18
 - installing, 4
 - R script files, 5–6, 104, 119–123
 - RStudio projects, 13–14
 - setting up, 4–5
 - working with data, 9–13
 - analysis with tidyverse, 14–16
 - projections, 66–67, 81–82
 - projects, RStudio, 13–14
 - pull() function, 118, 120, 144–145

`.pull-left[]` content class, 129–130
`.pull-right[]` content class, 129–130
purrr package, 121–123
pwalk() function, 122–123

Q

`.qmd` files, 169–171
Quarto, xxii, 157–158, 177
 creating document, 158–159
 making slideshows, 163–169
 making websites, 169–177
 parameterized reporting, 161–163
 Quarto Pub, 176–177
 vs. R Markdown, 111, 157,
 159–161, 177
`_quarto.yml` file, 169, 171–172, 174

R

ragg package, 52
raster data, 62
rbind() function, 120
R CMD check, 211–213
rdrop2 package, 153
reactable package, 97, 149–150
read_csv() function, 12–13, 202
read.csv() function, 10, 17
read_sf() function, 79, 81
read_sheet() function, 183, 187–188
rectangular data, 13
reknitting R Markdown documents,
 108–109
relative paths, 13
relocate() function, 74, 185
`.remark-slide-content`, 132
remotes package, 50
rename() function, 185
Render button, Quarto, 161
render() function, 120–123, 163
Render Website option, RStudio, 170
replace_na() function, 71
reports, xxi–xxii. *See also*
 parameterized reporting;
 R Markdown
 and googlesheets4 package,
 183–187
 multi-tool workflow for,
 101–102, 114
 templates for, 114–119

reproducibility, 102
revealing content incrementally in
 slideshows, 128–129, 165
reveal.js JavaScript library, 163, 169
R for Data Science (Wickham, Çetinkaya-
 Rundel, and Golemund), 16
right alignment in tables, 90–91
`.right-column[]` content class, 130
rio package, 32
R Markdown, xxi–xxii, 102, 111
 code chunks in documents,
 104–106
 creating document, 102–103
 creating R script, 119–123
 creating tables, 86
 formatting text, 107
 generating numbers with
 parameters, 117–118
 and googlesheets4 package,
 183–187
 headings, 107
 inline R code, 108–109
 markdown text, 106–108
 vs. Quarto, 111, 157, 159–161, 177
 report templates in, 114–119
 running code chunks interactively,
 109–111
 setup code chunk, 105–106
 website project files, 138–139
 xaringan package, working with,
 126–127
 YAML section in documents, 104
rmarkdown package, 120–123
rnaturalearth package, 80
Rodrigues, Bruno, 203–204
rolling averages, 71–73
rollmean() function, 72
row_number() function, 184–185
Roxygen, 214–215
R Packages (Wickham and Bryan), 215
R programming language, xx, 3,
 18, 218
 basic syntax, 6–9
 comments, 16–17
 help resources, 17–18
 installing, 4
 R script files, 5–6, 104, 119–123
 RStudio projects, 13–14

- R programming language (*continued*)
 - setting up, 4–5
 - working with data, 9–13
 - analysis with `tidyverse`, 14–16
 - R script files, 5–6, 104, 119–123
 - RStudio
 - building websites, 140, 170
 - creating documents
 - Quarto, 158–159
 - R Markdown, 102–103
 - exploring interface, 4–5
 - installing, 4
 - packages
 - creating, 209–210
 - installing, 12–13
 - projects in, 13–14
 - creating new `distill` project, 138
 - publishing Quarto website, 176
 - R script files, 5–6
 - script file pane, 5–6
 - working with custom fonts, 52
 - working with data, 9–13
- S**
- Sass (style sheet language), 168–169, 173–174
 - `scss:defaults` section, 168–169
 - `scss:rules` section, 169
 - Sayed, Safia, 113
 - `scale_fill_discrete_sequential()` function, 78
 - `scale_fill_viridis_c()` function, 30–31, 198
 - `scale_fill_viridis_d()` function, 34, 42
 - scales package, 118
 - Scherer, Cédric, 19–20, 32, 34–42
 - Schwabish, Jon, 91, 93
 - Scientific American*. *See* drought data visualization
 - `sf` package, 68–70, 79, 81
 - `shift_geometry()` function, 198–199
 - `show_in_excel()` function, 203–204
 - `show_in_excel_penguins()` function, 202–203
 - simple features (`sf`) data, 62
 - accessing geospatial data, 79–81
 - bounding box, 66
 - coordinate reference system, 66–67
 - dimensions, 66
 - fields, 62
 - geometry column, 62, 67
 - geometry type, 62–66
 - wrangling geospatial data, 82–83
 - `_site.yml` file, 139–140, 143
 - `slice_max()` function, 144, 145
 - slideshow presentations, xxii, 125, 135
 - adding background images, 131
 - adjusting size of figures, 128
 - aligning content with content classes, 129–131
 - applying CSS to slides, 131–135
 - creating new slides, 127–128
 - getting started, 126–127
 - with Quarto, 163–169
 - revealing content incrementally, 128–129
 - two-column layouts, 129–130, 165–167
 - `xaringan` package, advantages of, 125–126
 - small multiples charts, 57–58. *See also* faceting data visualizations
 - snake case, 206
 - sparklines, 95–96
 - `states()` function, 79–80
 - statistical analysis, xx
 - Statistical Inference via Data Science* (Ismay and Kim), 4, 12
 - `str_glue()` function, 144, 202
 - strip text, modifying with `bbplot` package, 57–58
 - `st_transform()` function, 70, 81
 - `styles.css` file, 169, 172
 - `styles.scss` file, 173–174
 - `style_xaringan()` function, 135
 - Stylianou, Nassos, 47, 59
 - `suggest_top_crs()` function, 81
 - `summarize()` function, 15–16
 - syntax, R, 6–9
 - `systemfonts` package, 52
- T**
- tables, xxi, 85–86, 97
 - adding data visualizations, 95–97
 - alignment in, 90–92

- creating data frames, 86–87
- differentiating header from body, 89–90
- intentional use of color, 94
- interactive, on websites, 149–150
- minimizing clutter, 87–89
- using correct level of precision, 93
- in website, 145, 147
- `tab_style()` function, 89–90, 94
- `tempfile()` function, 202
- templates, report, 114–119
- text
 - aligning in tables, 91–92
 - formatting, with `bbplot` package, 52–54
- `theme.css` file, 141–143
- themes, BBC. *See* BBC custom themes
- themes, `ggplot2` package, 38
 - `theme()` function, 38–40, 52, 58–59, 78
 - `theme_light()` function, 38
 - `theme_minimal()` function, 31, 146
- themes, Quarto, 167–168, 172
- themes, `xaringan` package, 133–134
- tibbles, 13, 120–123
- `tidycensus` package, 181, 188, 199
 - adding dependency packages, 212–213
 - connecting to Census Bureau with API key, 188–189
 - custom functions, 204–205, 207
 - visualizing American Community Survey data, 194–199
 - working with Decennial Census data, 189–194
- `tidyverse` package, 3, 12
 - and `bbplot` package, 48–49
 - creating maps, 68–69
 - data analysis with, 14–16
 - functions, 15–16
 - loading, 14
 - pipe, 15–16
 - R Markdown documents, errors when knitting, 110
 - tables, 86
 - websites, 144
 - wrangling geospatial data, 82–83
 - writing simple functions, 202

- `tigris` package, 79–80, 144, 196–199
- tooltip, interactive, 148, 150–153
- Tufte, Edward, 20

U

- `ungroup()` function, 71, 192
- Urban Institute parameterized reporting, 113, 124, 161–163
 - county-level reports, 124
 - fiscal briefs, 113, 124
- `urbnthemes` package, 114
- `url()` function, 131
- `usa_sf()` function, 70, 79
- US Census Bureau data, 188
 - American Community Survey data, 194–199
 - connecting to with API key, 188–189
 - Decennial Census data, 189–194
- `use_mit_license()` function, 216
- `use_package()` function, 212
- `use_r()` function, 210
- `usethis` package, 210, 216

V

- variables
 - data frame, 11
 - Decennial Census data, working with, 189–194
 - mapping to aesthetic properties, 25–26, 33–34
 - Sass, 168–169, 173–174
- vector geospatial data, 62
- vectors, creating, 120–121
- Visual Display of Quantitative Information, The* (Tufte), 20
- visualization of data. *See* data visualization

W

- Walker, Kyle, 188, 189, 193, 196, 199
- websites, xxii, 137–138, 155
 - applying custom CSS, 141–143
 - applying layouts, 148
 - building, 140–141
 - creating new `distill` project, 138
 - hosting, 153–155
 - interactive content, 148–153

- websites (*continued*)
 - making with Quarto, 169–177
 - navigation bars, 140, 174–175
 - pagination, 149–150
 - project files, 138–140
 - table of contents, 174
 - titles, 174
 - vignettes on documentation sites, 18
 - working with content, 143–148
- Westchester County COVID-19 website
 - project, 137–138, 148–149
- Wickham, Hadley, 16, 24–25, 202, 215
- Wilke, Claus, 85–86
- Wilkinson, Leland, 24
- Williams, Aaron, 113
- `write_csv()` function, 204
- `write_sheet()` function, 188

X

- xaringan package, 125, 135, 157
 - adding background images to slides, 131
 - adjusting size of figures, 128
 - advantages of, 125–126

- aligning content with content classes, 129–131
- applying CSS to slides, 131–135
- creating new slides, 127–128
- getting started with, 126–127
- revealing content incrementally, 128–129

- xaringanthemr package, 134–135

- x-axis

- mapping data to aesthetic properties, 25–26
- tweaking appearance of, 35

Y

- YAML section

- Quarto documents, 159–160, 163, 165, 167–168, 170–171
- R Markdown documents, 104, 117–119, 138–139

- y-axis

- mapping data to aesthetic properties, 25–26
- tweaking appearance of, 35

Z

- zoo package, 68–69, 72