# INDEX

## Symbols

+ (addition operator) methods, 231
;, ;;, ;;;, etc. (array concatenation operators), 143
#= (begin multiline comment), 59
& (bitwise AND), 31
# (comment line), 59
~ (distribution assertion), 414
÷ (division operator), 28
;; (double semicolon), 191
.. (ellipsis operator), 453
=# (end multiline comment), 59
± (error operator), 281
: (expression quotation), 168
== (generic equality operator), 31
> (greater-than operator), 31
>= (greater-than-or-equal-to operator), 31
=== (identity operator), 31
oo (infinity), 393
⊆ (issubset operator), 136
⊇ (issuperset operator), 136
\ (left division operator), 398
< (less-than operator), 31
<= (less-than-or-equal-to operator), 31
&& (logical AND), 31
|| (logical OR), 31
∈ (membership operator), 43
% (modular arithmetic operator), 29
∉ (not-a-member operator), 43
π (pi, mathematical constant), 218–219, 245
--> (plot recipe assignment), 256
:= (plot recipe assignment), 256
: (range operator), 35
// (rational operator), 28
; (semicolon), 29
-p (startup flag), 480
-t (startup flag), 469
<: (subtype operator), 233–234
:: (type assertion or declaration), 225–229

## A

abmplot() function, 374
abmvideo() function, 375–376
AbstractFloat type, 234
abstract types, 223, 229
  creating, 234
Action() Javis function, 205
act!() Javis function, 204
Adams, Ansel, 83
add_edge!() Luxor function, 193, 196
addition operator (+) methods, 231
adjacency matrix, 196
adjoint() function, 145
agent-based modeling, 362–379
@agent macro, 364
Agents package, 363
  multiple agent types, 367
  spaces, 363
allagents() function, 368
all() function, 166
allowmissing() function, 331
animation, 198–208
  with Reel, 206
anim_rotate_around() Javis function, 204
anim_rotate() Javis function, 205
ANSI color codes, 23
any() function, 166
Any type, 221–224
APL, 10
Aquinas, Thomas, 153
arbitrary precision arithmetic, 216–218
args property, 264
arguments
  keyword, 96–97
  optional, 96–97
Arizona, 430
array comprehensions, 125–127
array concatenation operators (;, ;;, ;;;, etc.), 143

arrays, 33–35
  adjoints and transposes, 144–146
  concatenation, 142–143
    using a newline, 142
    using semicolons, 143
  Hermitian adjoint, 145
  indexing, 34
  initialization, 139–142
  logical indexing, 143–144
  multidimensional, 36
  mutability with `fill()` and `repeat()`, 140–141
  planes, 143
  reshaping, 141–142
  sparse, 196
artificial life, 380
`as` import keyword, 65
atomic variables, 472–474
automatic differentiation, 406–413
*Avengers*, *The*, 189
`AxisArrays` package, 452–454
`Axis()` function, 453

## B

`Background()` `Javis` function, 204
baseball, 312
`Base` module, 62, 232
`batch_size` keyword argument for `pmap()`, 485
`begin...end` block, 30–31
begin multiline comment (`#=`), 59
`@belapsed` macro, 470, 474
`BenchmarkTools` package, 470, 474
Bessel function, 207, 384–388, 393–395
  series representation, 384
  symbolic derivatives, 386–388
Bessel's equation, 393–395
`Between()` function, 339
`BigFloat` type, 217–218
`big()` function, 217
`BigInt` type, 217–218
bimodal distribution, 320
binary operators, 159
  defining custom, 159–161
`@bind` macro, 208–209
binomial coefficient, 313, 317
`binomial()` function, 313
bioinformatics, 361
BioJulia, 361
biology, Julia ecosystem, 361
`BitArray` type, 144
bitwise AND (`&`), 31
BLAS, 404, 468
blood, counting cells, 442–449
`blue()` function, 443
`Bool` type, 31
`break` keyword, 124–125
broadcasting, 51
BSD, 4, 5
bugs, 357
byte array literal, 130–131

## C

C, xxi, 25
C++, xxii, 20
cactus ferruginous pygmy owl, 430
`CairoMakie` package, 376
California, 213
call stack, 179–181
`canny()` function, 447
cartesian indices, 459–464
`cdf()` function, 417
central limit theorem, 417
`Chain` package, 174–175
characters, 44–46
Chesapeake Bay, 193–195
Chocolately, 9
closure, 199–200
`coalesce()` function, 330
`@code_warntype` macro, 243–245
coding environments, 23
coin, fairness, 413
`collect()` function, 36, 128
collection
  defined, 123
  flattening, 369
  and loops, 123
  as parametric type, 248
  type of elements, 221
`Cols()` function, 340
`cols()` function, 345
combinations, 313
combinatorics, 312
`combine()` function, 346–347, 350
  `renamecols()` function, 347
comment line (`#`), 59
comments, 59

- complex numbers, 27
  - plotting, 264
- composite types
  - creating, 234–236
  - immutable, 236
  - mutable, 236
  - properties of, 235
- comprehensions, array, 125–127
- computer algebra, 382–395
- computer languages, 26
- concatenation, 41–43
- concrete types, 222
- concurrency, 467–486
- `conj()` function, 146
- `const` keyword, 52
- constructors, 235
  - inner, 240
  - methods, 235–236
  - outer, 240
- `continue` statement, 125
- `ContinuousAgent` notation, 364
- contour plots, 110–112, 116
- converting units, 272–274
- cooperative multitasking, 477
- core, CPU, 468
- `Core` module, 62
- correlation, 357
  - and crime, 354, 357
  - Pearson, 354
- `corrplot()` plotting function, 357
- counting, 312
- COVID-19, 334
  - models, 359
- crime, 352–353
- cryptography, 307
- CSV files, 332
- `CSV` package, 332, 334
  - `normalizenames()` function, 334

## D

- data
  - noisy, 421
  - out of core, 358
- dataframe, 333–337
  - `Between()` function, 339
  - columns
    - mutating, 341
    - references and copies, 341
  - display, 335
  - filtering, 337, 339
  - grouping, 349–350
  - indexing, 336–337
    - `Cols()` function, 340
    - using regular expressions, 340
  - `missing` values, 337
  - `names` function, 342
  - `Not()` function, 339
  - plotting, 338–339
  - `show()` function, 335
  - summary statistics, 348
  - transposing, 342–343
- `DataFrames` package, 333–337
- Dawkins, Richard, 361
- `DefaultSymbols` submodule, 271
- dependency hell, 11
- `describe()` function, 348, 355
- destructuring, 96, 157–159
  - `struct` keyword arguments, 158–159
- determinant, 399
- `det()` function, 399
- `@df` macro, 335, 353
- `Dict()` function, 134–135
- dictionaries, 134–135
  - indexing, 134–135
- `diff()` (`SymPy` differential operator), 394
- `DifferentialEquations` package, 294–303, 362, 408–413
  - combining with `Measurements`, 302–303
  - controlling error, 301, 412
  - plotting solutions, 297–298
  - time-dependent parameters, 299–301
- differential equations with `SymPy`, 393–395
- `Differential()` function, 387
- Dijkstra, Edsger, 213
- dimensions, 270
- directed graph, 193
- `disallowmissing()` function, 331
- `@distributed` macro, 484–485
- `Distributed` package, 480
- distribution, 321–322
  - bimodal, 320
  - empirical, 322
  - Gaussian, 324–326
  - normal, 324–326
  - theoretical, 322

- distribution assertion (~), 414
- `Distributions` package, 324–326
- `DivideError`, 179
- division, 27
- division operator (÷), 28
- DNA, 361
- `do` blocks, 166–167
- docstrings, 67
- documentation
  - with docstrings, 67–68
  - and Markdown, 68–69
- `doit()` function, 393
- double semicolon (`;;`), 191
- drugs, 362
- drum, modes of vibration, 206–207
- `dsolve()` `SymPy` function, 394
- `DSP` package, 435–437
- `duration()` function, 432

## E

- *e* (Euler's number), 219
- eccentricity in the Ptolemaic system, 201
- editors, 5, 14–15
- eigenvalues, 399
- eigenvectors, 399
- `eigvals()` function, 399–400
  - of symmetric matrix, 401
  - of triangular matrix, 402
- `eigvecs()` function, 399
- Einstein, Albert, 381
- Elixir, 169
- ellipsis operator (`..`) , 453
- `eltype()` function, 221
- Emacs, 15
  - REPL interaction, 15
- end multiline comment (`=#`), 59
- entropy sources, 307
- `enumerate()` function, 147–148
- epicycles, 201–205
- `Eq()` `SymPy` function, 394
- `erf()` (error function), 393
- error operator (±), 281
- error propagation, 280–284
- errors, 178–186
  - combining with units, 283
- Euler's number, 219
- `@everywhere` macro, 480
- evolution, simulated, 362–379
- `EvolutionaryModelingTools` package, 362
- exceptions, 178–186
  - types of, 178–179
- executive process, 480
- `expand_derivatives()` function, 387
- `expint()` function, 264
- exponential integral, 264
- expression, 26
  - from string, 168
- expression objects, 168–170
  - interpolation of values, 169–170
- expression quotation (`:`), 168
- `Expr` type, 168
- `:extra_kwargs` entry, 257

## F

- `f0` numerical suffix, 281
- factorial, 216
- `factorial()` function, 217, 313
- factorization, matrix, 402–403
- `factorize()` function, 402–403
- factor trees, 197–198
- `@fastmath` performance macro, 176–177
- FedEx, 61
- `fetch()` function, 476
- `fill()` function, 139
- `fillrange` plotting attribute, 257
- `filter()` function, 163–164
- `finally` keyword, 186
- `findfirst()` function, 131–132
- `findlast()` function, 131–132
- `findnext()` function, 132
- `fir()` function, 435–437
- fish, 193
- flattening a collection, 369
- `Float16` type, 215
- `Float32` type, 215, 281
- `Float64` type, 27, 214
- `floatmax()` function, 216
- `floatmin()` function, 216
- fluid dynamics, 284–294
- `foldl()` function, 164
- `foldr()` function, 164
- `Folds` package, 469–470
- football fields, 270
- force from potential energy, 408–413
- Fortran, xxi–xxii, 404, 454
- `ForwardDiff` package, 406–413
- Fourier transform, 433

Fox, Professor L., 395
framerate() function, 432
France, 350
FreeUnits Unitful type, 272
frequency analysis, 433–441
frequency filtering, 435–441
function, 48–51
  anonymous, 51
  composing, 50
  difference from macros, 172
  extending, 232
  higher-order, 161
  keyword arguments
    concise syntax, 154
  return value, 50
functional languages, 229
fundraising, 269

## G

gas, 306
Gaston plotting backend, 116, 278
Gaussian distribution, 109, 324–326, 392–393
generator expression, 127–128
generic equality operator (==), 31
gingerbread man, 95–96
*Glaucidium brasilianum cactorum*, 430
GLMakie package, 376
global keyword, 226–227
global variables, 52
gnuplot, 116, 278
goats, 310
gradient, 406
graphplot() Luxor function, 195–198
GraphRecipes package, 193
graphs (node-edge diagrams), 192–199
Graphs package, 193–198
  layout methods
    :buchheim, 198
    :tree, 198
  layout quality, 195
Gray() function, 443
greater-than operator (>), 31
greater-than-or-equal-to operator (>=), 31
Greek letters, 13
green() function, 443
grid() function, 118
GroupedDataFrame type, 349–350
GR plotting backend, 115, 277–278
gui() function, 86

## H

hardware requirements, 4
HDF5 plotting back end, 116
heatmaps, 110
Hermitian adjoint, 145
higher-order functions, 161
high-performance computing, 485–486
high school, bad memories, 197
histogram2d() function, 355
histograms, 322, 353
  bins, 323
  2D, 355
  using :scatterhist, 325
Hopper, Grace, 467
hough_circle_gradient() function, 446
Hough transform, 446–449
hyperthreading, 469
hypothesis testing, 358
HypothesisTests package, 358

## I

identity matrix, 399
identity operator (===), 31
IDEs, 20–22
idxs plotting keyword, 297
if blocks, 33
ImageBinarization package, 444
image blurring, 463
ImageFeatures package, 446–449
image processing, 442–464
image reduction, 463
Images package, 442–449
  RGB type, 442
ImageView package, 442
import statement, 63–64
imshow() function, 442
@inbounds performance macro, 176
indexing of arrays, 34, 38–39
  with arrays, 39–41
InexactError, 179
Inf type, 215
Inf16 type, 215
Inf64 type, 215

- infinity, 28, 215
- infinity (`oo`), 393
- infix operators, 159
  - defining custom, 159–161
- `in` membership operator, 43
- inner constructors, 240
- instability, 288
- installation, 3–5
  - on BSD, 6–7
  - on Docker, 9
  - on Linux, 6–7
  - on macOS, 7–8
  - on Windows, 8–9
- `Int16` type, 226
- `Int32` type, 27, 247
- `Int64` type, 27, 215, 247
- `Int128` type, 216
- `Integral()` `SymPy` function, 393
- `InteractiveDynamics` package, 373
- interprocess communication, 483
- `Int` type, 247
- inverse, matrix, 398
- `inv()` function, 398
- irrational numbers, 218–219
- `Irrational` type, 218–219 , 233
- `isa()` function, 216
- `isascii()` function, 163
- `ishermitian()` function, 400
- `issubset()` function, 136
- issubset operator (⊆), 136
- issuperset operator (⊇), 136
- `issymmetric()` function, 400
- iterated map, 95–96

## J

- Java, 20
- JavaScript, 25, 211
- `Javis` package, 198–205
  - motions, 205
- Johns Hopkins, 334
- `join()` function, 58
  - optional delimiter, 132
- Julia, features of
  - big, 151
  - composability, 303, 383–385, 388
  - introduction, xxi
  - no classes, 241
  - not functional, 229
  - not object oriented, 229
  - unusual, 213
- `JuliaDB` package, 358
- `julia-emacs`, 15
- `julia-repl`, 15
- Julia versions, 5
- `julia-vim`, 14–15
- Jupyter, 16, 17
  - plotting with, 17

## K

- keyword arguments, 96–97
- `kill_agent()` function, 372
- `@kwdef` macro, 241–242

## L

- LAPACK, 404
- LaTeX, 13, 274
  - strings, 129
- `latexify()` function, 274
- `Latexify` package, 274–276, 385
- `LaTeXStrings` package, 104–105
- lava lamps, 307
- Lederman, Leon M., 269
- left division operator (`\`), 398
- Leibniz, 245
- `lens()` function, 106–107
- less-than operator (`<`), 31
- less-than-or-equal-to operator (`<=`), 31
- *Let's Make a Deal*, 310
- `libblastrampoline` package, 404
- linear algebra, 233, 395–403
- `LinearAlgebra` package, 399–400
- linear equations, system, 397–399
- linear regression, 105
- `LinearSolve` package, 404
- line plot, 86
- Linux, 5, 404
- Lisp, 25, 169
- load balancing, 479
- `local` keyword, 226–227
- logic, 31
  - three-valued, 331
- logical AND (`&&`), 31
- logical indexing, 143–144
- logical OR (`||`), 31
- looping, 46–47
  - over strings, 48

lowered form, 243
lscpu command, 469
LuaLaTeX, 274, 279
Luxor package, 190–192, 239, 251, 285, 294
  coordinate system, 192
  defaults, 191
  fonts, 192
  scale factor, 191

## M

machine file, 481
macOS, 4–5
@macroexpand macro, 177
macros, 170–177
  adding syntax to Julia, 171–173
  for broadcasting, 173–174
  for chaining functions, 174–175
  collision avoidance, 171
  creating, 171
  difference from functions, 171
  for information, 177
  invocation syntax, 171
  for performance, 175–177
  for string formatting, 177
  for timing, 175
map() operator, 161–163
  and broadcasting, 162–163
mapreduce() operator, 166
marginalhist() plotting function, 356
Marx, Groucho, 123
MathJax, 274–275, 389
math symbols, 13
MATLAB, xxii
matrix, 37
  identity, 399
  special types, 400–402
  triangular, 401
matrix factorization, 402–403
matrix inverse, 147
matrix multiplication, 146–147
maximum() function, 166
mean() function, 319–321
measurement() function, 282
Measurements package, 280–284
  combining with DifferentialEquations, 302–303
median() function, 319
membership, 43
membership operator (∈), 43
Meta.parse() function, 168
metaprogramming, 167–177
MethodError, 179
methods, 229–233
methods() function, 231
minimum() function, 166
missing() function, 330
Missings package, 331
missing type, 328–330
  and logic, 331
  and Plots, 329
MIT, xxi
mode() function, 320
modular arithmetic operator (%), 29
module paths, 66
modules
  creating, 65–67
  current, 66
  exported names, 63
  naming, 63
  paths and dots, 66–67
  renaming imported, 65
Monk, Thelonious, 462
Monty Hall problem, 310–311
mosaicview() function, 443
MP3, 430
multiple dispatch, xxii, 229–233, 241
multiplication by juxtaposition, 29–30
multiprocessing, 468, 479–485
multithreading, 468–479
mutable keyword, 236
mutation, 55–59
  arrays, 55–56
  by functions, 56–57
  strings, 58
mutually assured destruction, 457
myid() function, 483

## N

N0f8 type, 442
named tuples, 138–139
names() function, 342
namespaces, 62, 66
NASA, 191, 270
native types, 216
Netflix, 406
networked computing, 481–482, 484
nframes() function, 432

Node, 10
nonstandard string literals, 128–129
normal distribution, 324–326
Normal() function, 324
not-a-member operator (∉), 43
notebooks
  Jupyter, 16–17
  Pluto, 17–20
Not() function, 339
nouns, 213, 234
nsolve() SymPy function, 392
nthreads() function, 469
numbers, 26
  complex, 27
  irrational, 218–219
  rational, 28
  types of, 26–27
Number type, 222
numerical precision, 216
numeric-symbolic modeling, 384
numeric types, 214
nworkers() function, 480

## O

Object() Javis function, 204
object-oriented programming, 213, 229, 240
occursin() function, 131
Oceananigans, 284–294
  boundary conditions, 286
  computational grid, 285
  diffusivities, 286
  equation of state, 287
  initial conditions, 288
  the model, 287
  precompiling, 284
  running a simulation, 287–290
  visualization, 290, 292, 294
Octave, xxii
ODEProblem() function, 296
OffsetArrays.center() function, 458
OffsetArrays package, 454–459
ones() function, 141
OpenStreetMap, 363
operating systems, 4–5
optional arguments, 96–97
OSCAR, 404
outer constructors, 240

## P

-p (julia startup flag), 480
packages, 69–81
  privacy, 9
pairs() function, 148–150
pandemic
  simulation, 313, 316–318
    boundary conditions, 317
parametric instability, 300–302, 422–426
parametric plots, 86, 93–94
  3D, 112–113
parametric types, 214, 248–252
PCM, 430
pdf() function, 325
@pdf Luxor macro, 192
Pearson correlation, 354
Peel, Emma, 189
pendulum, 294–302, 408–409, 422–426
  finite angle, 298–299
  parametric instability, 300–302, 422–426
performance and type stability, 242–247
Perl, 133
permode keyword argument, 275–276
permutations, 313
permutedims() function, 145–146, 343
petaflop club, xxii
PGFPlots plotting backend, 116
PGFPlotsX plotting backend, 116, 279
physics, 269–304
pi, mathematical constant ($\pi$), 218–219, 245
PI, SymPy constant, 393
pixel type from Images package, 442
plot
  aspect ratio, 101
  attributes, 98–99
    fonts, 100
    frame styles, 100
  color palette, 109
  components of, 98
  contour, 110–112
    filled, 111–112
    labeled, 111

current, 91
of a damped oscillator, 91
with errors, 284
functions, 88–89
gnuplot, 116
inset, 106–107, 121
interactive, 116
label position, 103–104
labels, 102
layout, 117–121
legend position, 102–103
mutating, 92
parametric, 93–94
3D, 112–113
polar, 86
polar coordinates, 86, 94–95
for publication, 277–280
recipes, 252–264
pipeline, 254–255
plot, 259–260
series, 255–258
type, 260–262
user, 262–263
saving, 106
scatter, 95–96
with singularities, 89
3D, 114–115
subplot, 100
surface, 108–109
settings, 108
in the terminal, 115
vector, 113
for the web, 116
plotlyjs plotting backend, 116
plot recipe assignment (-->), 256
Plots, 84
backends, 84, 115–116
activation, 84
and dependencies, 86
Gaston, 116
GR, 115
HDF5, 116
installation, 84
names of, 85
PGFPlots, 116
PlotlyJS, 116
PyPlot, 116
UnicodePlots, 115
closing windows, 86
displaying from programs, 86
plot settings, 98
plotting pipeline, 254–255
plot_title plotting argument, 102
plumbing, 406
Pluto, 16–18, 20, 23, 191
dependency graph, 18–20
interactive controls, 208–210
interface, 18–20
and LaTeX math, 274
and MathJax, 274–275
module files, 20
with SymPy, 388–395
PlutoUI package, 208–209
pmap() function, 480–481, 484
@png Luxor macro, 191
point (unit), 191
polar plots, 94–95
power spectral density, 433–435
precision() function, 218
prediction, 322
prerequisites, xxiii
prime factorization, 197–198
prime numbers, 125
@printf macro, 177, 214
Printf package, 177, 214
println() function, 32
multi-argument, 124
privacy with the package system, 9
probabilistic programming, 413–426
probabilistic simulation, 310–313, 316–318
probability, 306
combining, 317
frequency interpretation, 306
philosophy of, 414
probability density function, 325
probability theory, 359
prod() function, 165
psd() function, 433–435
pseudorandom numbers, 307
normally distributed, 326–327
Ptolemy, 201
Pumas package, 362
p-values, 358
pyplot plotting backend, 116
Python, xxii, 10, 25, 240, 388

## Q

`Quantity` `Unitful` type, 272
`quiver()` function, 113–114
`quote` blocks, 168
quoting expressions, 169

## R

R, 305, 358
`rand()` function, 307–309, 323–324
`randn()` function, 326–327
`random_agent()` function, 372
random events
  disjunction, 317
  in programs, 309
randomness and computers, 306
random number generators, 359
  seeding, 309
random numbers, 307
range operator (`:`), 35
ranges, 35–36
rational numbers, 28
rational operator (`//`), 28
raw strings, 129
`RCall` package, 358
`readchomp()` function, 482
readline, 11
`readline()` function, 124
`@recipe` macro, 255–256
`RecipesBase` package, 255–256
`red()` function, 443
`reduce()` function, 164–166
  and non-associative operators, 164–165
reducing functions
  and empty collections, 165
  `init` argument, 165
  neutral element, 165
`Reel` package, 206–208, 292
regression lines, 357
regular expressions, 132–133
  nonstandard string literal, 132
  for selecting dataframe columns, 340
religion, 269
`reltol` parameter, 301, 412
renaming imported modules, 65
`render()` function, and LaTeX, 274
`repeat()` function, 128, 139
REPL
  colors, 12–13
  command recall, 11
  entering Unicode characters, 13
  help mode, 11
  initialization, 12–13
  introduction, 10
  modes, 10, 12
  package mode, 11
  paste mode, 11
  and readline, 11
  shell mode, 11
  tab completion, 12
`replace()` function, 131–132
`reshape()` function, 141–142
RGB type from `Images` package, 442
`roll()` `Reel` function, 206
rotation matrix, 382–383
`run!()` (`Agents` function), 373
`run()` function, 481

## S

saving signals, 441
scatterplots, 95–96, 354
  3D, 114
  voluminous, 355
scientific communication, 210
scientific machine learning, 405–427
scientific notation, 27
scientists, 23
SciML, 405–427
scope, 52
  and `begin` blocks, 52
  and `if` blocks, 52
  and functions, 52–53
  and loops, 54
  modification in interactive contexts, 54
ScreenSend, 15
secrets, 63
semantic version strings, 129–130
semicolon (`;`), 29
`@series` macro, 257
`:series_plotindex` attribute, 257
series recipes, 255–258
`setprecision()` function, 218
sets, 135–137
  adding elements, 137
  difference, 136

intersection, 136
subset, 136
superset, 136
Set type, 135–137, 221
sfilt() function, 438–441
@shorthands macro, 257
Shostak, Seth, 429
show() function for dataframes, 335
SignalAnalysis package, 431–441
signal() function, 431
signal processing, 430–441
signals, saving, 441
significant digits, 281–282
simulated evolution, 362–379
simulation, probabilistic, 310–313, 316–318
SI units, 271–273
skipmissing() function, 330
slurping, 156–157
Smith, Frederick W., 61
smooth plot setting, 105
Socrates, 405
solar system, 190
solve() function
in DifferentialEquations, 296
in SymPy, 390
source code for Julia, 6
sparse array, 196
SparseArrays package, 196, 233
@spawnat macro, 483
@spawn macro, 474–475
SpecialFunctions package, 207, 264
spectrogram, 433
splatting, 154–155
split() function, 58
spreadsheet, 19
@sprintf macro, 177
standard deviation, 321
standard library, 62
statistics, 305, 322, 358–359
Julia packages, 359
standard library, 319
Statistics package, common functions, 320
StatsBase package, 320
StatsPlots package, 335, 356–357
std() function, 321
Steed, John, 189
string() function, 128
string interpolation, 133–134
string literals, nonstandard, 128–129
strings, raw, 129
strings and characters, 44–46
struct, 137–138
constuctor, 137
mutable, 138
properties, 138
struct keyword, 235
SubDataFrame type, 349
substitute() function, 383
subtype, 222
subtype operator (<:), 233–234
subtypes() function, 224
supertype, 222
supertype() function, 222
supertypes() function, 224
surface plots, 108–109
@svg Luxor macro, 192
symbolic mathematics, 382–395, 404
Symbolics package, 382–388
tracing to generate expressions, 386
symbols, 167
defined, 170, 187
Symbol type, 168
SymPy package, 388–395
differential equations, 393–395
evaluating integrals, 392–393
numerical root finding, 391–392
with Pluto, 388–395
@syms macro, 389
@sync macro, 474–475

## T

-t (julia startup flag), 469
thermal convection, 288
thermal diffusivity, 287
thickness_scaling plot setting, 104
@threads macro, 470–474
3D plot, 108–112
throw() function, 183–185
TikZ, 278
time() function, 478
timing, 475
tracing
to generate Symbolics expression, 386
of a matrix, 399
transpose() function, 145

transposing dataframes, 342–343
tr() function, 399
triangular matrix, 401
trigonometry, 160
try...catch blocks, 181–182
tuples, 42–43
  named, 138–139
turbulent convection, 292
Turing package, 413–426
twinx() function, 300
2D plot, 86
two language problem, xxii
type aliases, 247
  for collections, 248
type assertion, 224–228
type assertion or declaration operator (::), 225–229
type declaration, 224–228
  of global variables, 226
  purpose, 228
  and scope blocks, 228
typemax() function, 215
typemin() function, 215
typeof() function, 214, 222
type piracy, 233
type promotion, 27, 219–220
type recipes, 260–262
types, 26, 213–265
  abstract, 223, 229
    creating, 234
  concrete, 222
  creating composite, 234–236
  irrational, 218–219
  native, 216
  numeric, 214
  parametric, 214, 248–252
  user-defined, 234–242
typesetting units, 274–276
type stability, 228, 242–247

## U

u (nonstandard string literal), 270
uconvert() function, 273
Ulm, 381
uncertainty() function, 282
undirected graph, 193
Unicode characters, 13
  code point, 130
  entering in the REPL, 13
unicodeplots plotting backend, 115
uniform distribution, 309
UnionAll type, 249
Union types, 233–234
unitformat plotting keyword, 278
UnitfulLatexify package,
Unitful package, 270–280
unit() function, 274, 279
units
  combining with errors, 283
  converting, 272–274
  physical, 270–280
  in plots, 276–280
  stripping from Unitful expressions, 272–274
  typesetting, 274–276
uparse() function, 271
upreferred() function, 273
US Census, 351–353
user-defined types, 234–242
  performance, 242
@userplot macro, 263–264
user recipes, 262–263
using keyword, 63
ustrip() function, 273, 279

## V

value() function, 282
var() function, 321
@variables macro, 382
variance, 321
vector plots, 113
vectors of vectors, 37
verbs, 213, 234
versions of Julia, 5–6
@view macro, 395–396
views, 395–397, 450–452
@views macro, 395–396
Vim, 14–15, 23
  REPL interaction, 14–15
viscosity, 287
VS Code, 20–22

## W

wait() function, 476
WAV file, 430
WAV package, 441
weather, 253

where keyword, 249
while block, 32
whitespace, 33
Windows, 4–5
worker processes, 480
write() method from Reel, 208

Y

yield() function, 477–479

Z

zeros() function, 141
zip() function, 150–151