Index

A

Anaconda environment, 472
Attributes, 439

B

Backward differentiation formula (BDF), 221
Bayesian statistics
  conditional probability, 385
  import modules, 384
  likelihood function, 385
  linear regression model, 393
    Boolean masks, 394
    dataset, 401
    GLM model, 399
    MCMC algorithm, 396
    MCMC sampling, 401
    Seaborn library, 397
    stochastic variable, 397
  Monte Carlo simulation methods, 386
  overview, 383
  posterior probability, 385
  random variables, 386
    mc.find_MAP function, 389
    mc.sample function, 387
    mc.traceplot function, 388
    PyMC models, 386
    SciPy stats module, 386
  sampling posterior distribution, 390
  statistical modeling, 385
  unconditional probabilities, 384
BDF. *See* Backward differentiation formula (BDF)
Butterworth filter, 422

C

Clustering, 378
  confusion_matrix function, 380
  K-means method, 378
  predict method, 379
  sklearn.cluster module, 378
Comma-separated
    values (CSV) format, 426
  csv.reader function, 427
  DataFrame instance, 429
  delimiter argument, 427
  loadtxt function, 428
  read_csv function, 428
  skiprows argument, 427
  TAB character, 426
  usecols argument, 428
Computing environments, 4
Conda, 472
  conda create command, 473
  conda info command, 473
  conda install PACKAGE, 475
  conda list command, 474
  conda update command, 475
  YAML format, 474
Conflicting objectives, 1
CSV format. *See* Comma-separated
    values (CSV) format
Cython library, 454
  cy_cumsum function, 465
  cy_julia_fractal function, 468
  cy_sum function, 465
  overview, 461

D

Delimiter-separated
    values (DSV), 426
Dense matrices, 235
Durbin–Watson statistical test, 360

E

Einstein summation convention, 61
Extrapolation, 169

© Robert Johansson 2015
R. Johansson, *Numerical Python*, DOI 10.1007/978-1-4842-0553-2

F

FDM. *See* Finite-difference method (FDM)
FEM. *See* Finite-element method (FEM)
FEniCS framework, 265
 CellFunction instance, 279
 DirichletBC class, 270, 274
 dolfin.Constant, 275
 dolfin.Expression object, 269
 dolfin.FunctionSpace class, 268
 dolfin.interactive, 271
 dolfin library, 266
 dolfin.MeshFunction object, 281
 dolfin.refine function, 279
 dolfin.solve, 282
 function instance, 270
 mesh object, 273
 RectangleMesh function, 267
 refined_mesh, 280
 vector method, 272
Finite-difference method (FDM), 257
 boundary-value problem, 257
 Dirichlet boundary conditions, 261
 eye function, 258
 ODE problem, 259
 reshape method, 260
 scipy.sparse module, 260, 262
 two-dimensional generalization, 259
Finite-element method (FEM), 262
Finite impulse response (FIR) filters, 419

G

Generalized least squares (GLS), 335
get_values method, 387

H

Hierarchical Data Format 5 (HDF5), 430
 attributes, 439
 datasets, 435
 files, 432
 flush method, 442
 group objects, 432
 HDFStore object, 444
 h5py library, 431
 iterrows method, 442
 PyTables library, 431, 440
 where method, 443
High-and low-level languages, trade-off, 1
Hypothesis testing, 325

I

Infinite impulse response (IIR) filters, 420
Installation commands, 477
Integral transforms, 202
 Fourier transform function, 205
 Laplace transform, 203
Interpolation, 169
 bivariate, 185
 Chebyshev polynomials, 170
 explicit matrix form, 170
 griddata function, 182
 implicit matrix form, 170
 import modules, 169
 Legendre polynomials, 170
 multivariate situations, 180
 polynomial, 173
 spline, 177
 Vandermonde matrix, 170
Interpreter, 5
IPython, 6
 autocompletion, 7
 command prompt, 6
 documentation, 7
 extension commands, 9
  debugger mode, 10
  file system navigation, 9
  %timeit and %time commands, 12
  profiler function, 12
  reset command, 11
  running scripts, 9
 input and output caching, 6
 notebook, 14
  cell-types, 16
  dashboard page, 14
  editing cells, 17
  features, 14
  HTML document, 19
  JSON-based file format, 15, 21
  markdown cells, 18
  PDF format, 20
 object introspection, 7
 Qt console, 13
 system shell, 8

J, K

JavaScript Object Notation (JSON) format, 445
 json.dump function, 446
 json.load function, 446–447

L

LUdecomposition method, 129

M

Machine learning, 363
- classification, 374, 377
  - classification_report function, 376
  - confusion_matrix function, 376
  - data and target attributes, 375
  - datasets module, 375
  - linear_model module, 375
  - load_iris function, 375
  - predict method, 376
  - sklearn.ensemble module, 377
  - sklearn.metrics module, 376
  - sklearn.tree module, 376
  - train_test_split function, 375
- clustering, 378
  - confusion_matrix function, 380
  - *K*-means method, 378
  - predict method, 379
  - sklearn.cluster module, 378
- cross-validation, 365
- dimensionality reduction, 366
- feature extraction, 365
- feature selection, 366
- regression, 366
  - ElasticNet class, 373
  - fit method, 369
  - LASSO method, 372
  - LinearRegression instance, 367
  - LinearRegression object, 368
  - make_regression function, 366
  - regularized, 369
  - score method, 367
  - SSE, 367
- sklearn modules, 364
- supervised learning, 364
- training, 364
- unsupervised learning, 365

Matplotlib
- annotations, 102
- axes layout managers, 113
  - GridSpec, 117
  - insets, 113
  - plt.subplot2grid function, 116
  - subplots, 114
- axes instances, 91, 96
- axis properties, 104
  - autoscale method, 105
  - axis ticks, 106
  - grid lines, 108
  - log-scale plots, 110
  - *set_title* method, 105
  - set_xlabel and set_ylabel methods, 104
  - set_xlim and set_ylim methods, 105
  - set_xticks and set_yticks methods, 107
  - spine attribute, 112
  - tick placements, 109
  - twinx method, 111
- color map graph, 119
- definition, 89
- figure instances, 91, 95
- import, 90
- interactive mode, 93
- legends, 101
- line properties, 98
- noninteractive nodes, 94
- NumPy arrays, 92
- plot types, 97
- plt.subplots function, 97
- text formatting, 102
- 3D graphs, 120

Matrix and vector operations, 57
- Einstein summation convention, 61
- elementwise multiplication, 57
- Kronecker product, 60
- matrix-vector multiplication, 58
- nontrivial matrix multiplication, 58

Mesh-grid arrays, 33

Miniconda environment, 472

Multiple integrals, 196

N

Nonparametric methods, 329

Numba library, 454–455
- Heaviside step function, 459–460
- imshow function, 458
- JIT-compiled function, 456
- jit_julia_fractal function, 458
- NumPy universal function, 461
- NumPy-array aware function, 460
- py_cumsum function, 456
- py_sum function, 456
- NumPy vectorize function, 459

Numerical integration methods, 188
- interpretation, 188
- midpoint rule, 189
- Newton–Cotes quadrature rule, 189
- quad function, 192
- quadrature rule, 190
- SciPy, 192
- Simpson's rule, 189
- sympy.Lambda function, 190
- tabulated integrand, 194
- trapezoid rule, 189

NumPy arrays
- attributes, 26
- Boolean-valued indexing, 39
- constant values, 32
- creation, 30

NumPy arrays (*cont.*)
- data types, 27
- Einstein summation convention, 61
- elementwise multiplication, 57
- fancy indexing, 39
- incremental sequences, 33
- indexing and slicing expressions, 36
- Kronecker product, 60
- logarithmic sequences, 33
- matrix creation, 35
  - arbitrary one-dimensional array, 35
  - nonzero diagonals, 35
- matrix-vector multiplication, 58
- memory data, 29
- mesh-grid arrays, 33
- multidimensional arrays, 37
- nontrivial matrix multiplication, 58
- properties, 34
- Python lists, 31
- real and imaginary parts, 29
- reshaping and resizing, 41
- slices, 36
- vectorized expressions, 44
  - aggregate function, 50
  - arithmetic operations, 46
  - array operations, 56
  - Boolean-valued arrays, 53
  - conditional expressions, 53
  - elementary mathematical function, 50
  - elementwise functions, 48
  - logical operations, 54
  - set operations, 55
- views, 38
- uninitialized values, 34

NumPy library, 26

O

ODEs. *See* Ordinary differential equations (ODEs)

Optimization, 147
- bisection method, 150
- constraints, 161
  - cvxopt library, 166
  - inequality function, 164
  - Lagrangian function, 162
  - L-BFGS-B method, 161
  - linear programming, 165
  - objective function, 163–164
  - optimize.minimize function, 163
  - SciPy SLSQP solver, 163
- continuous and smooth functions, 149
- convex problems, 149
- feasible method, 150
- import libraries, 147
- minimization problem, 148
- multivariate optimization
  - BFGS method, 155–156
  - brute force, 157
  - Hessian evaluations, 155
  - Newton's method, 153
  - objective function, 157
  - optimize.minimize function, 158
  - slice objects, 157
  - steepest descent method, 153
  - vectorized functions, 154
- nonlinear least square problem
  - Levenberg–Marquardt method, 159
  - model function, 160
- nonlinear programming problem, 148
- univariate optimization, 150

Ordinary differential equations (ODEs), 207–208
- boundary value conditions, 209
- canonical form, 208
- direction field graph, 214
- dsolve function, 214
- homogeneous, 208
- import modules, 207
- initial value conditions, 209
- Laplace transformation, 217
- nonhomogeneous, 208
- numerical methods, 220
  - Adams methods, 222
  - adaptive stepsize/stepsize control, 222
  - Euler's method, 220
  - Runge-Kutta method, 221
  - SciPy (*see* SciPy)
- source term, 208
- standard form, 208
- symbolic solution, 209

Ordinary least squares (OLS), 335

P, Q

Pandas library, 285
- DataFrame object, 289
  - apply method, 293
  - columns attribute, 289
  - drop method, 296
  - groupby method, 296
  - index attributes, 290
  - info method, 292
  - ix indexer attribute, 290
  - read_csv function, 291
  - sort_index method, 294
  - sortlevel method, 294
  - sort method, 295
  - sum method, 296
  - value_counts method, 295
  - values attribute, 290

import modules, 286
seaborn graphics library, 306
boxplot function, 308
dropna method, 308
heatmap, 310
jointplot function, 308
kdeplot function, 308
sns.set function, 306
violinplot functions, 308
Series object, 286
describe method, 288
index attribute, 287
kind argument, 288
plot method, 289
time series, 297
concat function, 305
DataFrame.plot method, 301
date_range function, 297
DateTimeIndex instance, 301
DatetimeIndex object, 298
freq keyword, 297
groupby methods, 303
info method, 301
join method, 303
mean function, 303
PeriodIndex class, 299
resample method, 304
reset_index method, 302
to_period method, 299
to_pydatetime method, 298
UNIX timestamps, 300
using read_csv, 299
Partial differential equations (PDEs), 255, 257
FDM, 257
boundary-value problem, 257
Dirichlet boundary conditions, 261
eye function, 258
ODE problem, 259
reshape method, 260
scipy.sparse module, 260, 262
two-dimensional generalization, 259
FEM
Dirichlet/Neumann type, 263
libraries, 264
strong form, 262
test function, 262
trial function, 262
FEniCS framework, 265
DirichletBC class, 270, 274
dolfin.CellFunction, 279
dolfin.Constant, 275
dolfin.Expression object, 269
dolfin.FunctionSpace class, 268
dolfin.interactive, 271
dolfin library, 266
dolfin.MeshFunction object, 281
dolfin.refine function, 279
dolfin.solve, 282
function instance, 270
mesh object, 273
RectangleMesh function, 267
refined_mesh, 280
vector method, 272
import modules, 256
Poisson model, 355
Polynomials, 171
Probability density function (PDF), 387

R

RandomState object, 317
Regression, 366
ElasticNet class, 373
fit method, 369
LASSO method, 373
LinearRegression instance, 367
LinearRegression object, 368
make_regression function, 366
regularized, 369
score method, 367
SSE, 367

S, T

SciPy
args argument, 227
definition, 125
double pendulum, dynamics, 233
eigenvalue equation, 134
import modules, 126
integrate.odeint, 224
linear equation system, 126
condition number, 127
higher-order polynomial model, 133
SciPy la.lstsq method, 132
LUdecomposition method, 129
parameters/constant values, 126
rectangular systems, 131
square systems, 127
symbolic variables, 130
sympy.solve function, 132
unique solution, 127
unknown model parameters, 132
Lokta–Volterra equation, 226
nonlinear equations, 136
bisection method, 138
Broyden's method, 142
multivariate equation systems, 142
Newton's method, 139
numerical techniques, 137

SciPy (*cont.*)
optimize module, 141
sympy.solve function, 136
trigonometric equations, 136
univariate function, 136
univariate systems, 143
vector-valued function, 142
visualization, 144
odeint function, 223
odeint solver, 227
set_integrator method, 229
set_jac_params method, 230
sympy.lambdify, 232
Serialization, 449
Signal processing, 405
import modules, 405
signal files, 418
convolution filters, 418
FIR and IIR filters, 420
spectral analysis, 406
Fourier transform, 406
frequency-domain filter, 410
spectogram, 414
window function, 411
Slices, 36
Social component, 3
Sparse matrices, 235
eigen value problems, 245
graphs and networks, 247
add_edges_from, 248
degree method, 251
edge_color argument, 250
Tokyo Metro graph, 252
transfer attribute, 252
import modules, 235
linear algebra functions, 242
linear equation systems, 242
pyplot module, 240
in SciPy, 236
sp.sparse module, 240
Spline interpolation, 177
Spyder IDE
object inspector, 23
overview, 21
panes, 21
Python and IPython consoles, 23
shell prompt, 22
source code editor, 22
Spyder Integrated
Development Environment, 4
Statistics, 313
import modules, 313
ndarray methods, 314
population, 314
random module, 315
choice function, 317
randint function, 315–316
RandomState class, 318
RandomState instance, 317
seed function, 317
random variable and distributions, 318
discrete Poisson distribution, 321
interval method, 320, 323
moment method, 320
SciPy stats module, 318–319, 324
stats method, 320
var and std methods, 315
statsmodels library, 333
datasets, 349
discrete regression, 351
logistic regression, 351
Poisson model, 355
import module, 334
linear regression, 343
mathematical model, 335
multivariate linear regression, 335
patsy library, 334, 336
binary-valued treatment fields, 342
DataFrame objects, 338
design_info attribute, 340
design matrix, 338, 342
formula syntax, 337
function call notation, 341
np.linalg.lstsq function, 339
simple linear regression, 335
time-series analysis, 358
ARMA model, 358
AR model, 360
fit method, 360
plot_acf function, 359–360
Steepest descent method, 153
Sum of squared errors (SSE), 367
Symbolic and arbitrary-precision integration, 200
Symbolic computing.
See Symbolic Python (SymPy)
Symbolic Python (SymPy), 63
equation solving, 83
linear algebra, 85
LUsolve method, 87
manipulating expressions, 72
substitutions, 75
sympy.apart function, 75
sympy.cancel function, 75
sympy.collect function, 75
sympy.expand function, 73
sympy.factor function, 74
sympy.simplify function, 72
sympy.together function, 75

mathematical expressions, 70
mathematical symbols, 64
  arbitrary function, 69
  constants and
    special symbols, 69
  floating-point number, 68
  integer class, 67
  lambda functions, 70
  rational number, 68
  sin function, 70
  unapplied function, 69
  undefined functions, 70
  unevaluated function, 69
numerical evalution, 76
symbolic calculus, 77
  derivatives, 77
  integrals, 79
  limits, 82
  series expansions, 80
  sums and products, 83
sympy.init_printing function, 64

U

Univariate optimization, 150

V

Vectorized expressions
  aggregate function, 50
  arithmetic operations, 46
  array operations, 56
  Boolean-valued arrays, 53
  conditional expressions, 53
  elementary mathematical function, 50
  elementwise functions, 48
  logical operations, 54
  set operations, 55
Visualization. *See* Matplotlib

W, X, Y, Z

Weighted least squares (WLS), 335

Get the eBook for only $5!

Why limit yourself?

Now you can take the weightless companion with you wherever you go and access your content on your PC, phone, tablet, or reader.

Since you've purchased this print book, we're happy to offer you the eBook in all 3 formats for just $5.

Convenient and fully searchable, the PDF version enables you to easily find and copy code—or perform examples by quickly toggling between instructions and applications. The MOBI format is ideal for your Kindle, while the ePUB can be utilized on a variety of mobile devices.

To learn more, go to www.apress.com/companion or contact support@apress.com.

Apress®
THE EXPERT'S VOICE™

All Apress eBooks are subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.