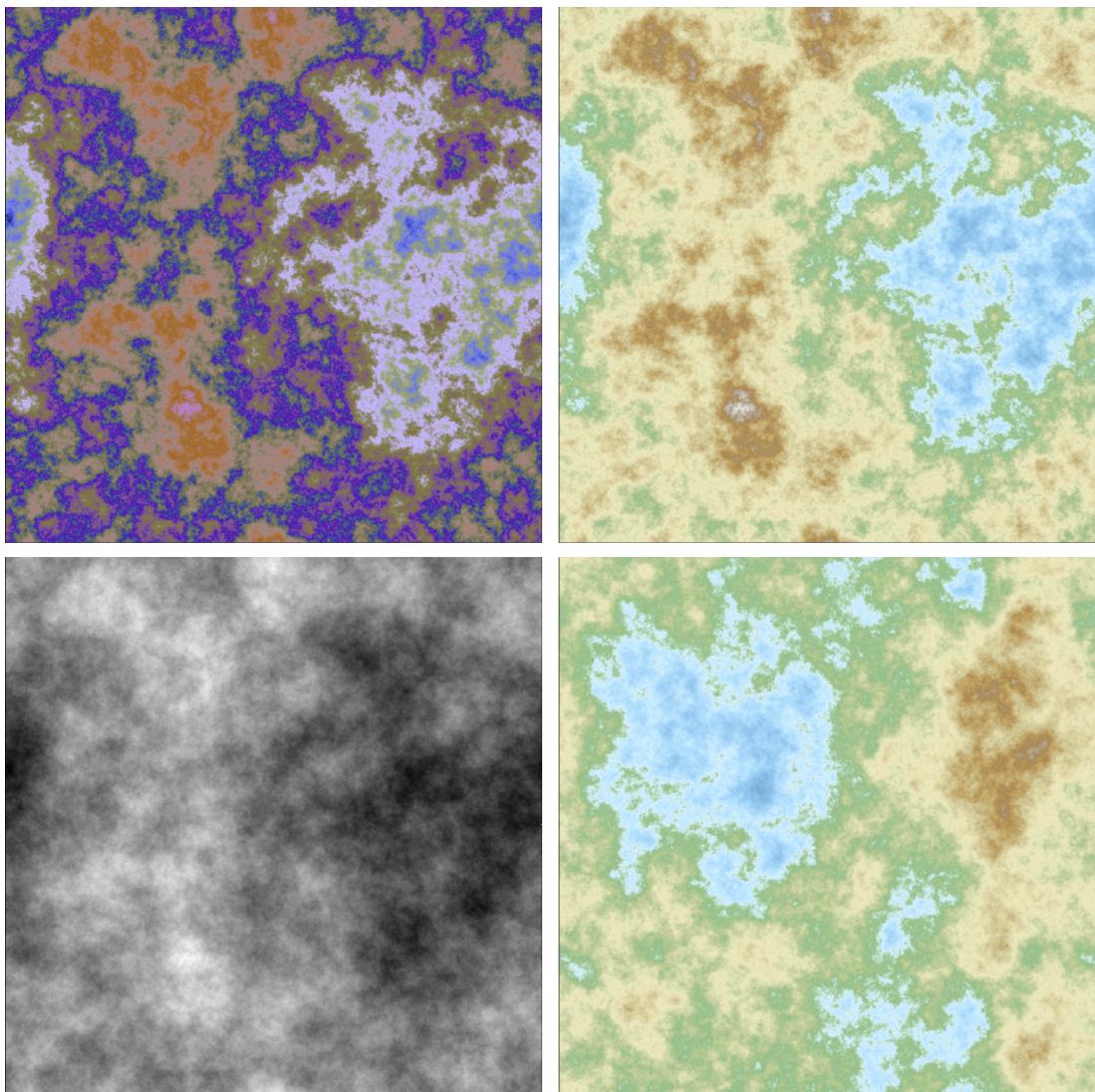

Synthetic Data and Generative AI



Preface

This book covers the foundations of machine learning, with modern approaches to solving complex problems and the systematic generation and use of synthetic data. Emphasis is on scalability, automation, testing, optimizing, and interpretability (explainable AI). For instance, regression techniques – including logistic and Lasso – are presented as a single method, without using advanced linear algebra. There is no need to learn 50 versions when one does it all and more. Confidence regions and prediction intervals are built using parametric bootstrap, without statistical models or probability distributions. Models (including generative models and mixtures) are mostly used to create rich synthetic data to test and benchmark various methods.

Topics covered include clustering and classification, GPU machine learning, ensemble methods including an original boosting technique, elements of graph modeling, deep neural networks, auto-regressive and non-periodic time series, Brownian motions and related processes, simulations, interpolation, random numbers, natural language processing (smart crawling, taxonomy creation and structuring unstructured data), computer vision (shapes generation and recognition), curve fitting, cross-validation, goodness-of-fit metrics, feature selection, curve fitting, gradient methods, optimization techniques and numerical stability.

Several chapters focus on synthetic data, agent-based modeling and GIS applications: fractal-like terrain generation with the diamond-square algorithm, disaggregation of ocean tides time series, geospatial interpolation of temperatures in the Chicago area, and synthetic star clusters evolving over time and bound by gravity. The latter provides great insights to explore the past and future of our universe or studying collision graphs. It also allows you to explore alternative universes, for instance with negative masses. Chapters 15 and 16 are more advanced and may be skipped in introductory classes. The former focuses on point process applications, while the later focuses on applications a machine learning methods to discover new insights in a famous mathematical conjecture: the Riemann Hypothesis. Section 17.7.2 illustrates the use of copulas to produce synthetic data, applied to a well-known insurance dataset.

Methods are accompanied by enterprise-grade Python code, replicable datasets and visualizations, including data animations (gifs, videos, even sound done in Python). The code uses various data structures and library functions sometimes with advanced options. It constitutes a solid introduction to scientific programming. The code, datasets, spreadsheets and data visualizations are also on GitHub, spread across the following repositories: [Machine Learning](#), [Point Processes](#), [Visualizations](#), and [Experimental Math](#). Chapters are mostly independent from each other, allowing you to read in random order. A glossary, index and numerous cross-references make the navigation easy and unify all the chapters.

The style is very compact, getting down to the point quickly, and suitable to business professionals. Jargon and arcane theories are absent, replaced by simple English to facilitate the reading by non-experts, and to help you discover topics usually made inaccessible to beginners. While state-of-the-art research is presented in all chapters, the prerequisites to read this book are minimal: an analytic professional background, or a first course in calculus and linear algebra. The original presentation avoids all unnecessary math and statistics, yet without eliminating advanced topics. Finally, this book is the main reference for my course on intuitive machine learning. For details about the classes, see [here](#).

About the Author

Vincent Granville is a pioneering data scientist and machine learning expert, co-founder of Data Science Central (acquired by a publicly traded company in 2020), founder of [MLTechniques.com](#), former VC-funded executive, author and patent owner. Vincent's past corporate experience includes Visa, Wells Fargo, eBay, NBC, Microsoft, and CNET.



Vincent is also a former post-doc at Cambridge University, and the National Institute of Statistical Sciences (NISS). He published in *Journal of Number Theory*, *Journal of the Royal Statistical Society* (Series B), and *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He is also the author of multiple books, available [here](#). He lives in Washington state, and enjoys doing research on stochastic processes, dynamical systems, experimental math and probabilistic number theory.

Data synthetization explained in one picture

The diagram is organized as follows. Dashed blue lines are associated to GANs ([generative adversarial networks](#)), where the goal is to produce a sequence of synthetic datasets that get better and better at mimicking the structure present in the real data, over successive iterations. The diagram features 5 such iterations, with the synthetized datasets denoted as S_1, S_2, \dots, S_5 . Typically, GANs follow the gradient of h to reach an optimum configuration q that can not be classified as non-real anymore. Synthetic data that gets closer to the real data gets rewarded in this reinforcement learning technique. Like any simulation-intensive method, training the neural network can be time-consuming, and this black-box approach may lack [explainability](#).

Dashed pink lines are associated to modeling techniques ([generative model](#), GMM) where synthetic data is obtained by simulating the underlying model using the parameter values estimated on the real data, that is, $q_k = p$ for all k . In case of GMM ([Gaussian mixture models](#)), the parameters are the cluster centers, the covariance matrix attached to each cluster, and the proportions of the mixture. For stationary time series, the parameter is typically the autocorrelation function (ACF). In some applications including when using [copulas](#), the EPDF (empirical probability density function) is used instead.

The goal is to mimic the structure in the real data, not the real data itself. The structure is represented by a parametric configuration denoted as p in the real data. I use the notation p_1, \dots, p_5 for the structures found in the 5 synthetic data sets. The quality h_k of the synthetic data set k is the distance between p_k and p , based on the [Hellinger metric](#) or some discriminating function in the case of GAN. It is assumed that the real data has been normalized (transformed) before synthesizing. “Estim. param.” stands for estimated parameters in the diagram, though sometimes the parameters can be a function or matrix rather than a set of elements.

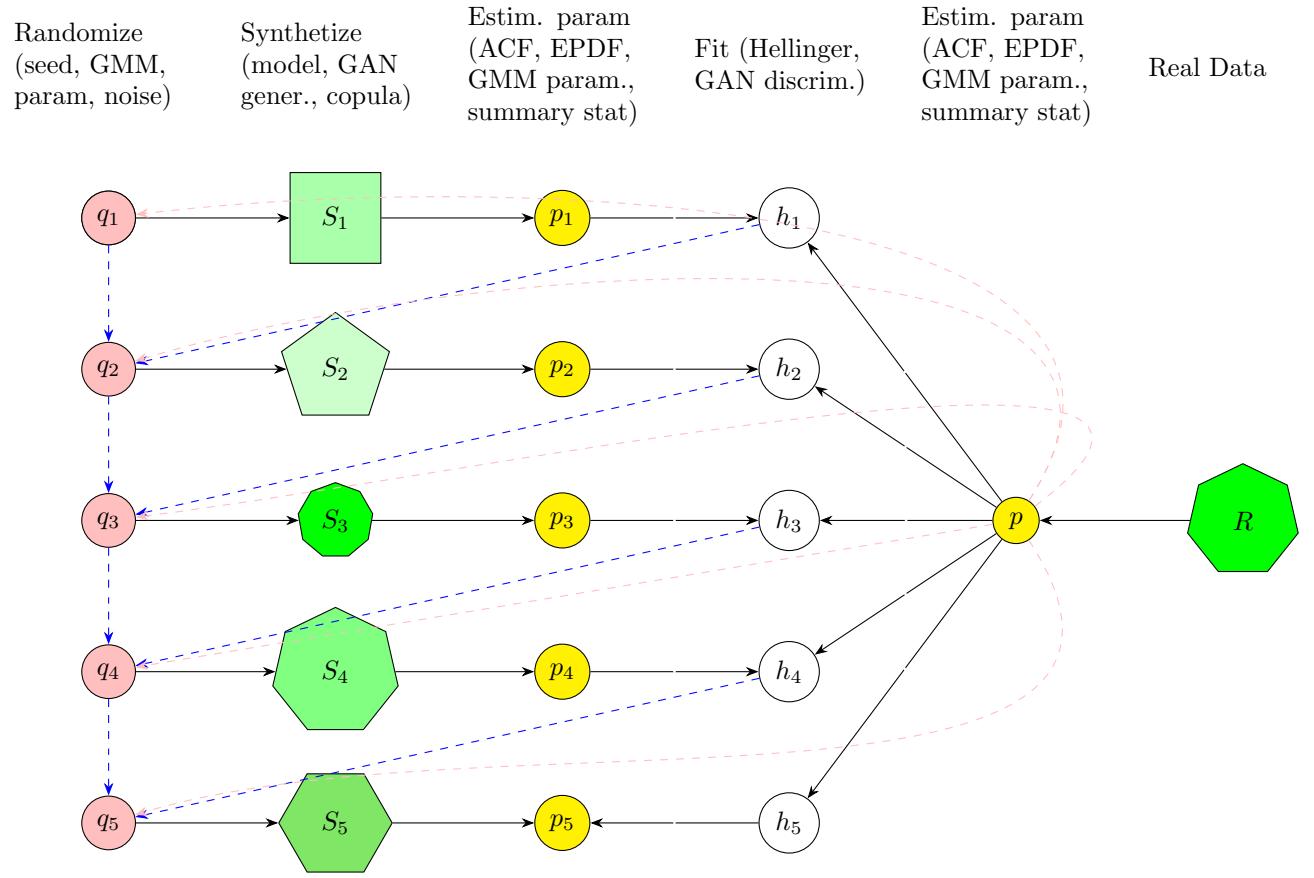


Figure 1: Data synthetization: general schema

Contents

List of Figures	10
List of Tables	12
1 Machine Learning Cloud Regression and Optimization	13
1.1 Introduction: circle fitting	13
1.1.1 Previous versions of my method	14
1.2 Methodology, implementation details and caveats	15
1.2.1 Solution, R-squared and backward compatibility	15
1.2.2 Upgrades to the model	16
1.3 Case studies	17
1.3.1 Logistic regression, two ways	17
1.3.2 Ellipsoid and hyperplane fitting	18
1.3.2.1 Curve fitting: 250 examples in one video	18
1.3.2.2 Confidence region for the fitted ellipse: application to meteorite shapes	19
1.3.2.3 Python code	20
1.3.3 Non-periodic sum of periodic time series: ocean tides	26
1.3.3.1 Numerical instability and how to fix it	27
1.3.3.2 Python code	28
1.3.4 Fitting a line in 3D, unsupervised clustering, and other generalizations	29
1.3.4.1 Example: confidence region for the cluster centers	30
1.3.4.2 Exact solution and caveats	31
1.3.4.3 Comparison with K-means clustering	32
1.3.4.4 Python code	34
1.4 Connection to synthetic data: meteorites, ocean tides	36
2 A Simple, Robust and Efficient Ensemble Method	37
2.1 Introduction	37
2.2 Methodology	38
2.2.1 How hidden decision trees (HDT) work	38
2.2.2 NLP case study: summary and findings	39
2.2.3 Parameters	40
2.2.4 Improving the methodology	40
2.3 Implementation details	40
2.3.1 Correcting for bias	40
2.3.1.1 Time-adjusted scores	41
2.3.2 Excel spreadsheet	41
2.3.3 Python code and dataset	41
2.4 Model-free confidence intervals and perfect nodes	45
2.4.1 Interesting asymptotic properties of confidence intervals	45
3 Gentle Introduction to Linear Algebra – Synthetic Time Series	47
3.1 Power of a matrix	47
3.2 Examples, generalization, and matrix inversion	48
3.2.1 Example with a non-invertible matrix	49
3.2.2 Fast computations	49
3.2.3 Square root of a matrix	49
3.3 Application to machine learning problems	50
3.3.1 Markov chains	50
3.3.2 Time series: auto-regressive processes	50

3.3.3	Linear regression	51
3.4	Mathematics of auto-regressive time series	51
3.4.1	Simulations: curious fractal time series	52
3.4.1.1	White noise: Fréchet, Weibull and exponential cases	52
3.4.1.2	Illustration	52
3.4.2	Solving Vandermonde systems: a numerically stable method	53
3.5	Math for Machine Learning: Must-Read Books	54
4	The Art of Visualizing High Dimensional Data	55
4.1	Introduction	55
4.2	Applications	56
4.2.1	Spatial time series	56
4.2.2	Prediction intervals in any dimensions	56
4.2.3	Supervised classification of an infinite dataset	57
4.2.3.1	Machine learning perspective	57
4.2.3.2	Six challenging problems	58
4.2.3.3	Mathematical background: the Riemann Hypothesis	58
4.2.3.4	Partial solutions to the six challenging problems	59
4.2.4	Algorithms with chaotic convergence	60
4.3	Python code	60
4.3.1	Path simulation	60
4.3.2	Visual convergence analysis in 2D	63
4.3.3	Supervised classification	64
4.4	Visualizations	67
5	Fast Classification and Clustering via Image Convolution Filters	70
5.1	Introduction	70
5.2	Generating the synthetic data	71
5.2.1	Simulations with logistic distribution	71
5.2.2	Mapping the raw observations onto an image bitmap	72
5.3	Classification and unsupervised clustering	72
5.3.1	Supervised classification based on convolution filters	73
5.3.2	Clustering based on histogram equalization	73
5.3.3	Fractal classification: deep neural network analogy	74
5.3.4	Generalization to higher dimensions	75
5.3.5	Towards a very fast implementation	75
5.4	Python code	76
5.4.1	Fractal classification	77
5.4.2	GPU classification and clustering	79
5.4.3	Home-made graphic library	81
6	Shape Classification and Synthetization via Explainable AI	84
6.1	Introduction	84
6.2	Mathematical foundations	84
6.3	Shape signature	85
6.3.1	Weighted centroid	85
6.3.2	Computing the signature	86
6.3.3	Example	87
6.4	Shape comparison	87
6.4.1	Shape classification	88
6.5	Application	88
6.6	Exercises	89
7	Synthetic Data, Interpretable Regression, and Submodels	90
7.1	Introduction	90
7.2	Synthetic data sets and the spreadsheet	91
7.2.1	Correlation structure	91
7.2.2	Standardized regression	92
7.2.3	Initial conditions	92
7.2.4	Simulations and Excel spreadsheet	92
7.3	Damping schedule and convergence acceleration	93
7.3.1	Spreadsheet implementation	93

7.3.2	Interpretable regression with no overfitting	94
7.3.3	Adaptive damping	94
7.4	Performance assessment on synthetic data	94
7.4.1	Results	95
7.4.2	Distribution-free confidence intervals	97
7.4.2.1	Parametric bootstrap	98
7.5	Feature selection	98
7.5.1	Combinatorial approach	98
7.5.2	Stepwise approach	99
7.6	Conclusion	100
8	From Interpolation to Fuzzy Regression	102
8.1	Introduction	102
8.2	Original version	103
8.3	Full, non-linear model in higher dimensions	103
8.3.1	Geometric proximity, weights, and numerical stability	104
8.3.2	Predicted values and prediction intervals	104
8.3.3	Illustration, with spreadsheet	105
8.3.3.1	Output fields	106
8.4	Results	106
8.4.1	Performance assessment	106
8.4.2	Visualization	107
8.4.3	Amplitude restoration	107
8.5	Exercises	108
8.6	Python source code and datasets	109
9	New Interpolation Methods for Synthetization and Prediction	113
9.1	First method	113
9.1.1	Example with infinite summation	114
9.1.2	Applications: ocean tides, planet alignment	115
9.1.3	Problem in two dimensions	116
9.1.4	Spatial interpolation of the temperature dataset	117
9.2	Second method	119
9.2.1	From unstable polynomial to robust orthogonal regression	120
9.2.2	Using orthogonal functions	120
9.2.3	Application to regression	120
9.3	Python code	121
9.3.1	Time series interpolation	121
9.3.2	Geospatial temperature dataset	124
9.3.3	Regression with Fourier series	127
10	High Quality Random Numbers for Simulations and Data Synthetization	129
10.1	Introduction	129
10.2	Pseudo-random numbers	130
10.2.1	Strong pseudo-random numbers	130
10.2.1.1	New test of randomness for PRNGs	131
10.2.1.2	Theoretical background: the law of the iterated logarithm	131
10.2.1.3	Connection to the Generalized Riemann Hypothesis	131
10.2.2	Testing well-known sequences	132
10.2.2.1	Reverse-engineering a pseudo-random sequence	133
10.2.2.2	Illustrations	134
10.3	Python code	136
10.3.1	Fixes to the faulty random function in Python	136
10.3.2	Prime test implementation to detect subtle flaws in PRNG's	136
10.3.3	Special formula to compute 10 million digits of $\sqrt{2}$	139
10.4	Military-grade PRNG Based on Quadratic Irrationals	142
10.4.1	Fast algorithm rooted in advanced analytic number theory	142
10.4.2	Fast PRNG: explanations	143
10.4.3	Python code	143
10.4.4	Computing a digit without generating the previous ones	145
10.4.5	Security and comparison with other PRNGs	145
10.4.5.1	Important comments	145

10.4.6 Curious application: a new type of lottery	146
11 Some Unusual Random Walks	147
11.1 Symmetric unbiased constrained random walks	147
11.1.1 Three fundamental properties of pure random walks	147
11.1.2 Random walks with more entropy than pure random signal	148
11.1.2.1 Applications	148
11.1.2.2 Algorithm to generate quasi-random sequences	149
11.1.2.3 Variance of the modified random walk	149
11.1.3 Random walks with less entropy than pure random signal	150
11.2 Related stochastic processes	151
11.2.1 From Brownian motions to clustered Lévy flights	151
11.2.2 Integrated Brownian motions and special auto-regressive processes	152
11.3 Python code	153
11.3.1 Computing probabilities and variances attached to S_n	153
11.3.2 Path simulations	154
12 Divergent Optimization Algorithm and Synthetic Functions	156
12.1 Introduction	156
12.1.1 The problem, with illustration	157
12.2 Non-converging fixed-point algorithm	158
12.2.1 Trick leading to intuitive solution	158
12.2.2 Root detection: method and parameters	158
12.2.3 Case study: factoring a product of two large primes	159
12.3 Generalization with synthetic random functions	159
12.3.1 Example	161
12.3.2 Connection to the Poisson-binomial distribution	162
12.3.2.1 Location of next root: guesstimate	162
12.3.2.2 Integer sequences with high density of primes	162
12.3.3 Python code: finding the optimum	163
12.4 Smoothing highly chaotic curves	164
12.4.1 Python code: smoothing	164
12.5 Connection to synthetic data: random functions	167
13 Synthetic Terrain Generation and AI-generated Art	168
13.1 Introduction	168
13.2 Terrain generation and the evolutionary process	170
13.2.1 Morphing and non-linear palette operations	170
13.2.2 The diamond-square algorithm	170
13.2.3 The evolutionary process	171
13.2.4 Finding optimum parameters	171
13.2.5 Mimicking real terrain: the synthesis step	171
13.3 Python code	172
13.3.1 Producing data videos with four sub-videos in parallel	172
13.3.2 Main program	173
13.4 AI-generated art with 3D contours	177
13.4.1 Python code using Matplotlib	178
13.4.2 Python code using Plotly	179
13.4.3 Tips to quickly solve new problems	180
14 Synthetic Star Cluster Generation with Collision Graphs	181
14.1 Introduction	181
14.2 Model parameters and simulation results	182
14.2.1 Explanation of color codes	182
14.2.2 Detailed description of top parameters	182
14.2.3 Interesting parameter sets	183
14.3 Analysis of star collisions and collision graph	184
14.3.1 Weighted directed graphs: visualization with NetworkX	185
14.3.2 Interesting findings: how the universe got started	185
14.4 Animated data visualizations	186
14.5 Python code and computational issues	187
14.5.1 Simulating the real and synthetic universes	187

14.5.2	Visualizing collision graphs	191
15	Perturbed-Lattice Point Process: Inference, Nearest Neighbor Graph	193
15.1	Perturbed lattices: definition and properties	193
15.1.1	Point counts distribution	194
15.1.2	Periodicity and amplitude of point count expectations	194
15.1.3	Testing the independence of point counts	195
15.1.3.1	Results and Interpretation	196
15.1.3.2	About the Spreadsheet	197
15.2	Cluster processes and nearest neighbor graphs	197
15.2.1	Synthetic, semi-rigid cluster structures	197
15.2.2	Python code to generate cluster processes	199
15.2.3	References on cluster processes	199
15.2.4	Superimposed perturbed lattices: an alternative to mixture models	200
15.2.4.1	Hexagonal lattice, nearest neighbors	201
15.2.4.2	Exercises: nearest neighbor graphs, size of connected components	202
15.2.4.3	Python code to compute connected components	203
15.3	Statistical inference for point processes	205
15.3.1	Estimation of Core Parameters	205
15.3.1.1	Intensity	206
15.3.1.2	Scaling factor	206
15.3.1.3	Alternative estimation method	206
15.3.2	Spatial statistics, nearest neighbors, clustering	207
15.3.2.1	Inference for two-dimensional processes	207
15.3.2.2	Other possible tests	207
15.3.2.3	Rayleigh test	208
15.3.2.4	Exercises	209
15.4	Special topics	210
15.4.1	Minimum contrast estimation and explainable AI	210
15.4.2	Model identifiability, hard-to-detect patterns	211
15.4.2.1	Stochastic residues	211
15.4.3	Hidden model and random permutations	211
15.4.4	Retrieving the F distribution	213
15.4.4.1	Theoretical values obtained by simulations	213
15.4.4.2	Retrieving F from the interarrival times distribution	214
15.4.5	Record distances between an observed point and its vertex	214
15.4.5.1	Distribution of records	215
15.4.5.2	Distribution of arrival times for records	216
16	New Perspective on the Riemann Hypothesis	217
16.1	Introduction	217
16.1.1	Key concepts and terminology	218
16.1.2	Orbits and holes	218
16.1.3	Industrial Applications	218
16.2	Euler products	219
16.2.1	Finite Euler Products	219
16.2.1.1	Generalization using Dirichlet characters	220
16.2.2	Infinite Euler products	221
16.2.2.1	Special products	221
16.2.2.2	Probabilistic properties and conjectures	222
16.3	Finite Dirichlet series and generalizations	223
16.3.1	Finite Dirichlet series	223
16.3.2	Non-trivial cases with infinitely many primes and a hole	225
16.3.2.1	Sums of two cubes, or cuban primes	225
16.3.2.2	Primes associated to elliptic curves	225
16.3.2.3	Analytic continuation, convergence, and functional equation	226
16.3.2.4	Hybrid Dirichlet-Taylor series	226
16.3.3	Riemann Hypothesis with cosines replaced by wavelets	227
16.3.4	Riemann Hypothesis for Beurling primes	228
16.3.5	Stochastic Euler products	229
16.4	Exercises	230

16.5	Python code	233
16.5.1	Computing the orbit of various Dirichlet series	233
16.5.2	Creating videos of the orbit	236
17	Misc Topics Including Copulas to Synthetize Data	239
17.1	The sound that data makes	239
17.1.1	From data visualizations to videos to data music	239
17.1.2	References	240
17.1.3	Python code	240
17.2	Data videos and enhanced visualizations in R	241
17.2.1	Cairo library to produce better charts	241
17.2.2	AV library to produce videos	242
17.3	Dual confidence regions	243
17.3.1	Case study	243
17.3.2	Standard confidence region	243
17.3.3	Dual confidence region	244
17.3.4	Simulations	244
17.3.5	Original problem with minimum contrast estimators	245
17.3.6	General shape of confidence regions	246
17.4	Fast feature selection based on predictive power	247
17.4.1	How cross-validation works	248
17.4.2	Measuring the predictive power of a feature	248
17.4.3	Efficient implementation	249
17.5	NLP: taxonomy creation and text generation	250
17.5.1	Designing a keyword taxonomy	250
17.5.2	Fast clustering algorithm for keyword data	251
17.5.2.1	Computational complexity	251
17.5.2.2	Smart crawling of the whole Internet and a bit of graph theory	252
17.6	Automated detection of outliers and number of clusters	253
17.6.1	Black-box elbow rule to detect outliers	253
17.7	Copulas, Hellinger distance and more about synthetic data	254
17.7.1	Sensitivity analysis, bias reduction and other uses of synthetic data	255
17.7.2	Using copulas to generate synthetic data	255
17.7.2.1	The insurance dataset: Python code and results	256
17.7.2.2	Potential improvements	258
17.8	Advice to beginners	259
17.8.1	Getting started and learning how to learn	259
17.8.1.1	Getting help	260
17.8.1.2	Beyond Python	260
17.8.2	Automated data cleaning and exploratory analysis	260
17.8.3	Example of simple analysis: marketing attribution	261
Glossary		262
Bibliography		265
Index		270

List of Figures

1	Data synthetization: general schema	3
1.1	Fitted ellipse (blue), given the training set (red) distributed around a partial arc	19
1.2	Confidence region in blue, $n = 30$ training set points; 50 training sets (left) vs 150 (right)	20
1.3	Three non-periodic time series made of periodic terms (see section 16.2.2.1)	26
1.4	Training set (red), validation set (orange), fitted curve (blue) and model (gray)	27
1.6	Biased confidence region for (θ_A^*, θ_B^*) ; same example as in Figure 1.5; true value is $(0.5, 1.0)$	30
1.5	Finding the two centers θ_A^*, θ_B^* in sample 39; $n = 1000$	31
1.7	Challenging mixture, requiring $p_A = 3, p_B = 1$ to identify the two cluster centers	32
2.1	Output from the Excel version of HDT	42
3.1	AR models, classified based on the types of roots of the characteristic polynomial	53
4.1	Scatterplot observations vs. predicted values, with prediction intervals (in any dimension)	67
4.2	Comets orbiting the sun: simulation	67
4.3	Comets orbiting the sun: snapshot in time	68
4.4	Three orbits of $\eta(\sigma + it)$: $\sigma = 0.5$ (red), 0.75 (blue) and 1.25 (yellow)	68
4.5	Sample orbit points of $\eta(\sigma + it)$: $\sigma = 0.5$ (red), 0.75 (blue) and 1.25 (yellow)	68
4.6	Sample orbit points of $\eta(\sigma + it)$: $\sigma = 0.5$ (red), 0.75 (blue) and 1.25 (yellow)	69
4.7	Raw orbit points of $\eta(\sigma + it)$: $\sigma = 0.5$ (red), 0.75 (blue) and 1.25 (yellow)	69
4.8	Convergence of partial sums of $\eta(z)$, for six $z = \sigma + it$ in the complex plane	69
5.1	Special interlacing of 4 lattice processes with $s = 0$	72
5.2	Classification of left dataset; $s = 0.15, w = 10$. One loop (middle) vs 3 (right).	73
5.3	Clustering of left dataset; $s = 0.15$, 3 loops, $w = 10$ (middle) vs 20 (right).	74
5.4	Classification ($w = 10$) and clustering ($w = 20$); $s = 0.05$, three loops.	74
5.5	Fractal classification, $s = 0.15$. Loop 6, 250 and 400.	75
5.6	Fractal classification, $s = 0.05$. Loop: 6 and 60.	75
5.7	Fast (left) vs standard method (right), 3 loops, $s = 0.15, w = 10$	76
5.8	Fast method, $s = 0.05, w = 20$. Three loops (middle), one loop (right).	76
6.1	Comparing two shapes	85
6.2	Weighted centroid, shape signature	86
6.3	Weight function used in Figure 6.2	87
6.4	Another interesting shape	88
7.1	Regression coefficients oscillating when using adaptive damping	95
7.2	Convergence of regression coefficients (left) and distribution of residual error (right)	96
7.3	Goodness-of-fit: training set (right) versus validation set (left)	96
8.1	Fuzzy regression with prediction intervals, original version, 1D	103
8.2	Fuzzy regression with prediction intervals, full model, 2D	105
8.3	Scatterplots: median vs weighted method, on validation (left) vs training set (right)	107
8.4	Dirichlet eta function (real part, bottom) and interpolation error (top)	109
9.1	Interpolating the real part of $\zeta(\frac{1}{2} + it)$ based on orange points	114
9.2	Tides at Dublin (5-min data), with 80 mins between interpolating nodes	117
9.3	Temperature data: interpolation with my method (observed values at dots)	118
9.4	My method: round dots represent observed values, “+” are interpolated	118
9.5	Temperature dataset: interpolation using ordinary kriging	119

10.1	Orbit of $L(z, \chi)$ at $\sigma = \frac{1}{2}$, with $0 < t < 200$ and $\chi = \chi_4$ (left) versus pseudo-random χ (right)	132
10.2	$L_3^*(n)$ test statistic for four sequences: Python[200] and SQRT[90,91] fail	134
10.3	$ L_3(n) $ test statistic for four sequences: Python[200] and SQRT[90,91] fail	134
10.4	Correlations are computed on sequences consisting of 300 binary digits	146
11.1	Typical path S_n with $0 \leq n \leq 50,000$ for four types of random walks	148
11.2	$\delta_n = 1 - \text{Var}[S_{n+1}] + \text{Var}[S_n]$ for four types of random walks, with $0 \leq n \leq 5000$	149
11.3	Same as Figure 11.2, using a more aesthetic but less meaningful chart type	150
11.4	Clustered Brownian process	152
11.5	AR models, classified based on the types of roots of the characteristic polynomial	153
12.1	Function $f(b)$ as a better alternative to $g(b)$ in Figure 12.2. Root at $b = 3083$	157
12.2	Function $g(b) = 2 - \cos(2\pi b) - \cos(2\pi a/b)$, with $a = 3083 \times 7919$	157
12.3	Transformed function f_3 , amplifying the root at $b = 3083$	158
12.4	Signal strength ρ_n , first 130 fixed-point iterations; $n = 31$ leads to a root.	161
12.5	(b_n, ρ_n) plot. Yellow and orange dots linked to roots.	161
12.6	Signal strength ρ_n , first 130 fixed-point iterations; $n = 87$ leads to a root.	161
12.7	Random function from section 12.3.1, with root at $b = 5646$	164
13.1	Six frames from the terrain video, each one containing four images	169
13.2	Contour plot, 3D mixture model, produced with Plotly	177
13.3	Same as Figure 13.2, produced with Matplotlib	178
14.1	Collisions graph for the biggest star eater (star 47) in video 7	185
14.2	Summary statistics for the whole collision structure: the X axis represents the time	186
14.3	Snapshots of universe 4 (left) and universe 7 (right)	187
15.1	Period and amplitude of $\phi_\tau(t)$; here $\tau = 1, \lambda = 1.4, s = 0.3$	195
15.2	A new test of independence (R-squared version)	195
15.3	Radial cluster process ($s = 0.2, \lambda = 1$) with centers in blue; zoom in on the left	198
15.4	Radial cluster process ($s = 2, \lambda = 1$) with centers in blue; zoom in on the left	198
15.5	Manufactured marble lacking true lattice randomness (left)	198
15.6	Four superimposed Poisson-binomial processes: $s = 0$ (left), $s = 5$ (right)	201
15.7	Rayleigh test to assess if a point distribution matches that of a Poisson process	209
15.8	Realization of a 5-interlacing with $s = 0.15$ and $\lambda = 1$: original (left), modulo $2/\lambda$ (right)	212
15.9	Locally random permutation σ ; $\tau(k)$ is the index of X_k 's closest neighbor to the right	212
15.10	Each arrow links a point (blue) to its vertex (red): $s = 0.2$ (left), $s = 1$ (right)	215
15.11	Distance between a point and its vertex ($\lambda = s = 1$)	216
16.1	Three orbits ($\sigma = 0.5, 0.75, 1.25$) with finite Euler product: $P = \{2, 3\}$ (left) vs $\{2, 3, 5\}$ (right)	220
16.2	Distance between orbit and location $(c, 0)$ depending on t on the X-axis	222
16.3	Distance between orbit and location $(c, 0)$ depending on t on the X-axis	222
16.4	Distance between orbit and location $(c, 0)$ depending on t on the X-axis	222
16.5	Four orbits where the “hole” (repulsion basin) is apparent	224
16.6	Three orbits with “hole” closer to the origin, showing impact of $\beta > \frac{1}{2}$ and larger n	224
16.7	Orbit of Dirichlet eta $\eta(z)$ when cosines are replaced by other periodic functions	228
17.1	Data linked to the melody: red curve for note frequencies, blue curve for note durations	240
17.2	R plot before Cairo (left), and after (right)	241
17.3	Intermediate (left) and last frame (right) of the video	242
17.4	Example of 90% dual confidence region for (p, q)	244
17.5	Minimum contrast estimation for (λ, s) using (p, q) as proxy stats	245
17.6	Non-elliptic confidence regions with various confidence levels	246
17.7	Elbow rule (right) finds $m = 3$ clusters in Brownian motion (left)	254

List of Tables

1.1	Estimated ellipse parameters vs true values ($n = 30$), for shape in Figure 1.2	20
1.2	First and last step of <code>curve_fitting</code> , approaching the model.	28
1.3	MSE for different methods and θ s, same data set as in Figure 1.5	33
1.4	MSE for different methods and θ s, same data set as in Figure 1.7	33
2.1	List of potential features to use in the model	38
2.2	Statistics for selected HDT nodes (Excel version)	41
2.3	Order of magnitude for the expectation and standard deviation of the range R_n	45
3.1	Characteristic polynomials used in the simulations	52
7.1	Regression coefficients and performance metrics r, s based on methodology	97
7.2	Correlation matrix	97
7.3	Best performance given m (number of features)	98
7.4	Feature comparison table (top 32 feature combinations)	100
7.5	Feature comparison table (bottom 31 feature combinations)	101
8.1	R -squared ρ^2 and slope β , on training and validation sets, median vs weighted	107
10.1	$L_3^*(n)$, for various sequences ($n = 20,000$); “Fail” means failing the prime test	135
12.1	High ρ_n at iterations $n = 31$ and $n = 127$ points to roots 3083 and 7919	160
14.1	Description of top parameters used in the star cluster simulator	183
14.2	Eight selected parameter sets covering various situations	184
15.1	Variance attached to F_s , as a function of s	194
15.2	Poisson process ($s = \infty$) versus $s = 39.85$	214
17.1	Extract of the mapping table used to recover (λ, s) from (p, q)	246
17.2	Eight bins: 2 features (A, B) times 2 outcomes (Good/Bad)	248
17.3	Amount of data collected at each level, when crawling the Internet	252
17.4	Comparing real data with two different synthetic copies	257

Glossary

Autoregressive process	Auto-correlated time series, as described in section 3.4. Time-continuous versions include Gaussian processes and Brownian motions, while random walks are a discrete example; two-dimensional versions exist. These processes are essentially integrated white noise. See pages 50, 98, 152
Binning	Feature binning consists of aggregating the values of a feature into a small number of bins, to avoid overfitting and reduce the number of nodes in methods such as naive Bayes, neural networks, or decision trees. Binning can be applied to two or more features simultaneously. I discuss optimum binning in this book. See pages 38, 74, 248
Boosted model	Blending of several models to get the best of each one, also referred to as ensemble methods. The concept is illustrated with hidden decision trees in this book. Other popular examples are gradient boosting and AdaBoost. See pages 37, 262
Bootstrapping	A data-driven, model-free technique to estimate parameter values, to optimize goodness-of-fit metrics. Related to resampling in the context of cross-validation. In this book, I discuss parametric bootstrap on synthetic data that mimics the actual observations. See pages 16, 97, 209, 262
Confidence Region	A confidence region of level γ is a 2D set of minimum area covering a proportion γ of the mass of a bivariate probability distribution. It is a 2D generalization of confidence intervals. In this book, I also discuss dual confidence regions – the analogous of credible regions in Bayesian inference. See pages 13, 16, 19, 21, 30, 206, 207, 243, 246
Cross-validation	Standard procedure used in bootstrapping, and to test and validate a model, by splitting your data into training and validation sets. Parameters are estimated based on training set data. An alternative to cross-validation is testing your model on synthetic data with known response. See pages 16, 38, 94, 100, 184, 248, 262
Decision trees	A simple, intuitive non-linear modeling techniques used in classification problems. It can handle missing and categorical data, as well as a large number of features, but requires appropriate feature binning. Typically one blends multiple binary trees each with a few nodes, to boost performance. See pages 37, 38, 40, 42, 262, 263
Dimension reduction	A technique to reduce the number of features in your dataset while minimizing the loss in predictive power. The most well known are principal component analysis and feature selection to maximize goodness-of-fit metrics. See pages 13, 17, 263, 264
Empirical distribution	Cumulative frequency histogram attached to a statistic (for instance, nearest neighbor distances), and based on observations. When the number of observations tends to infinity and the bin sizes tend to zero, this step function tends to the theoretical cumulative distribution function of the statistic in question. See pages 17, 97, 121, 130, 193, 196, 202, 208, 213, 215, 227, 256
Ensemble methods	A technique consisting of blending multiple models together, such as many decision trees with logistic regression, to get the best of each method and outperform each method taken separately. Examples include boosting, bagging, and AdaBoost. In this book, I discuss hidden decision trees. See pages 37, 84, 262
Explainable AI	Automated machine learning techniques that are easy to interpret are referred to as interpretable machine learning or explainable artificial intelligence. As much as possible, the methods discussed in this book belong to that category. The goal is to design black-box systems less likely to generate unexpected results with unintended consequences. See pages 14, 36, 70, 75, 84, 91, 156, 172, 210, 254

Feature selection	Features – as opposed to the model response – are also called independent variables or predictors. Feature selection, akin to dimensionality reduction , aims at finding the minimum subset of variables with enough predictive power . It is also used to eliminate redundant features and find causality (typically using hierarchical Bayesian models), as opposed to mere correlations. Sometimes, two features have poor predictive power when taken separately, but provide improved predictions when combined together. See pages 13 , 16 , 38 , 95 , 98 , 239 , 247 , 262 , 264
Generative model	Bayesian Gaussian mixtures (GMM) combined with kernel density estimation and the EM algorithm is a classic modeling tool. In this book, I used <i>m</i>-interlacings instead. Generative adversarial networks (GAN) work as follows: the generator creates new observations and the discriminator tests whether the new observations are statistically indistinguishable from training set data. When this goal is achieved, the new observations is your synthetic data. In this book, new observations are generated with parametric bootstrap instead. See pages 3 , 36 , 53 , 100 , 167 , 168 , 170 , 177 , 184 , 264
Goodness-of-fit	A model fitting criterion or metric to assess how a model or sub-model fits to a dataset, or to measure its predictive power on a validation set . Examples include R-squared , Chi-squared, Kolmogorov-Smirnov, error rate such as false positives and other metrics discussed in this book. See pages 16 , 57 , 94 , 95 , 248 , 262 , 264
Gradient methods	Iterative optimization techniques to find the minimum of maximum of a function, such as the maximum likelihood . When there are numerous local minima or maxima, use swarm optimization . Gradient methods (for instance, stochastic gradient descent or Newton's method) assume that the function is differentiable. If not, other techniques such as Monte Carlo simulations or the fixed-point algorithm can be used. Constrained optimization involves using Lagrange multipliers . See pages 16 , 32 , 56 , 90
Graph structures	Graphs are found in decision trees , in neural networks (connections between neurons), in nearest neighbors methods (NN graphs), in hierarchical Bayesian models , and more. See pages 71 , 75 , 185 , 251 , 252
Hyperparameter	An hyperparameter is used to control the learning process: for instance, the dimension, the number of features, parameters, layers (neural networks) or clusters (clustering problem), or the width of a filtering window in image processing. By contrast, the values of other parameters (typically node weights in neural networks or regression coefficients) are derived via training. See pages 30 , 57 , 71 , 76 , 102 , 171 , 263
Link function	A link function maps a nonlinear relationship to a linear one so that a linear model can be fit, and then mapped back to the original form using the inverse function. For instance, the logit link function is used in logistic regression . Generalizations include quantile functions and inverse sigmoids in neural network to work with additive (linear) parameters. See pages 14 , 17 , 263
Logistic regression	A generalized linear regression method where the binary response (fraud/non-fraud or cancer/non-cancer) is modeled as a probability via the logistic link function. Alternatives to the iterative maximum likelihood solution are discussed in this book. See pages 17 , 34 , 37 , 41 , 262 , 263
Neural network	A blackbox system used for predictions, optimization, or pattern recognition especially in computer vision. It consists of layers, neurons in each layer, link functions to model non-linear interactions, parameters (weights associated to the connections between neurons) and hyperparameters . Networks with several layers are called deep neural networks . Also, neurons are sometimes called nodes. See pages 70 , 74 , 76 , 84 , 102 , 262 , 263
NLP	Natural language processing is a set of techniques to deal with unstructured text data, such as emails, automated customer support, or webpages downloaded with a crawler. The example discussed in section 17.5 deals with creating a keyword taxonomy based on parsing Google search result pages. Text generation is referred to as NLG or natural language generation , using large language models (LLM). See pages 37 , 250

Numerical stability	This issue occurring in unstable optimization problems typically with multiple minima or maxima, is frequently overlooked and leads to poor predictions or high volatility. It is sometimes referred to as ill-conditioned problems . I explain how to fix it in several examples in this book, for instance in section 3.4.2. Not to be confused with numerical precision. See pages 13, 15, 60
Overfitting	Using too many unstable parameters resulting in excellent performance on the training set , but poor performance on future data or on the validation set . It typically occurs with numerically unstable procedures such as regression (especially polynomial regression) when the training set is not large enough, or in the presence of wide data (more features than observations) when using a method not suited to this situation. The opposite is underfitting. See pages 16, 93, 102, 255, 256, 262, 264
Predictive power	A metric to assess the goodness-of-fit or performance of a model or subset of features, for instance in the context of dimensionality reduction or feature selection . Typical metrics include R-squared , or confusion matrices in classification. See pages 39, 41, 45, 247, 249, 254, 263
R-squared	A goodness-of-fit metric to assess the predictive power of a model, measured on a validation set . Alternatives include adjusted R-squared, mean absolute error and other metrics discussed in this book. See pages 13, 16, 36, 57, 91, 94, 96, 98, 105, 263, 264
Random number	Pseudo-random numbers are sequences of binary digits, usually grouped into blocks, satisfying properties of independent Bernoulli trials. In this book, the concept is formally defined, and strong pseudo-number generators are built and used in computer-intensive simulations. See pages 30, 129, 136, 253
Regression methods	I discuss a unified approach to all regression problems in chapter 1. Traditional techniques include linear, logistic, Bayesian, polynomial and Lasso regression (to deal with numerical instability and overfitting), solved using optimization techniques, maximum likelihood methods, linear algebra (eigenvalues and singular value decomposition) or stepwise procedures. See pages 13, 14, 16, 17, 20, 28, 37, 41, 47, 51, 53, 57, 90, 96, 102, 109, 263, 264
Supervised learning	Techniques dealing with labeled data (classification) or when the response is known (regression). The opposite is unsupervised learning , for instance clustering problems. In-between, you have semi-supervised learning and reinforcement learning (favoring good decisions). The technique described in chapter 1 fits into unsupervised regression. Adversarial learning is testing your model against extreme cases intended to make it fail, to build better models. See pages 264
Synthetic data	Artificial data simulated using a generative model , typically a mixture model , to enrich existing datasets and improve the quality of training sets . Called augmented data when blended with real data. See pages 13, 14, 16, 18, 28, 30, 34, 36, 49, 53, 56, 70, 71, 76, 89, 95, 106, 113, 119, 129, 142, 148, 156, 170, 177, 184, 244, 253, 255, 262
Tensor	Matrix generalization with three or more dimensions. A matrix is a two-dimensional tensor. A triple summation with three indices is represented by a three-dimensional tensor, while a double summation involves a standard matrix. See pages 70, 75
Training set	Dataset used to train your model in supervised learning . Typically, a portion of the training set is used to train the model, the other part is used as validation set . See pages 14, 16, 18, 21, 30, 37, 41, 57, 73, 89, 96, 102, 106, 184, 248, 262, 264
Validation set	A portion of your training set , typically 20%, used to measure the actual performance of your predictive algorithm outside the training set. In cross-validation and bootstrapping, the training and validation sets are split into multiple subsets to get a better sense of variations in the predictions. See pages 16, 28, 42, 57, 94, 102, 184, 248, 255, 256, 262, 263, 264

Bibliography

- [1] Weighted percentiles using numpy. *Forum discussion*, 2020. StackOverflow [\[Link\]](#). 102
- [2] Jan Ackmann et al. Machine-learned preconditioners for linear solvers in geophysical fluid flows. *Preprint*, pages 1–19, 2020. arXiv:2010.02866 [\[Link\]](#). 94
- [3] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley, fourth edition, 2016. [\[Link\]](#). 202
- [4] José M. Amigó, Roberto Dale, and Piergiulio Tempesta. A generalized permutation entropy for random processes. *Preprint*, pages 1–9, 2012. arXiv:2003.13728 [\[Link\]](#). 213
- [5] Luc Anselin. *Point Pattern Analysis: Nearest Neighbor Statistics*. The Center for Spatial Data Science, University of Chicago, 2016. Slide presentation [\[Link\]](#). 200
- [6] Adrian Baddeley. Spatial point processes and their applications. In Weil W., editor, *Stochastic Geometry. Lecture Notes in Mathematics*, pages 1–75. Springer, Berlin, 2007. [\[Link\]](#). 199
- [7] David Bailey and Richard Crandall. Random generators and normal numbers. *Experimental Mathematics*, 11, 2002. Project Euclid [\[Link\]](#). 145
- [8] N. Balakrishnan and C.R. Rao (Editors). *Order Statistics: Theory and Methods*. North-Holland, 1998. 202, 216
- [9] Christopher Beckham and Christopher Pal. A step towards procedural terrain generation with GANs. *Preprint*, pages 1–5, 2017. arXiv:1707.03383 [\[Link\]](#). 169
- [10] Rabi Bhattacharya and Edward Waymire. *Random Walk, Brownian Motion, and Martingales*. Springer, 2021. 147
- [11] Barbara Bogacka. *Lecture Notes on Time Series*. 2008. Queen Mary University of London [\[Link\]](#). 50
- [12] B. Bollobas and P. Erdős. Cliques in random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 80(3):419–427, 1976. [\[Link\]](#). 203
- [13] Miklos Bona. *Combinatorics of Permutations*. Routledge, second edition, 2012. 213
- [14] Peter Borwein, Stephen K. Choi, and Michael Coons. Completely multiplicative functions taking values in $\{-1, 1\}$. *Transactions of the American Mathematical Society*, 362(12):6279–6291, 2010. [\[Link\]](#). 221
- [15] Peter Borwein and Michael Coons. Transcendence of power series for some number theoretic functions. *Proceedings of the American Mathematical Society*, 137(4):1303–1305, 2009. [\[Link\]](#). 223
- [16] Oliver Bröker and Marcus J. Groteb. Sparse approximate inverse smoothers for geometric and algebraic multigrid. *Applied Numerical Mathematics*, 41(1):61–80, 2002. 91
- [17] H. M. Bui and M. B. Milinovich. Gaps between zeros of the Riemann zeta-function. *Quarterly Journal of Mathematics*, 69(2):402–423, 2018. [\[Link\]](#). 233
- [18] Bartłomiej Błaszczyzyn and Dhandapani Yogeshwaran. Clustering and percolation of point processes. *Preprint*, pages 1–20, 2013. Project Euclid [\[Link\]](#). 199
- [19] Bartłomiej Błaszczyzyn and Dhandapani Yogeshwaran. On comparison of clustering properties of point processes. *Preprint*, pages 1–26, 2013. arXiv:1111.6017 [\[Link\]](#). 199
- [20] Bartłomiej Błaszczyzyn and Dhandapani Yogeshwaran. Clustering comparison of point processes with applications to random geometric models. *Preprint*, pages 1–44, 2014. arXiv:1212.5285 [\[Link\]](#). 199
- [21] Oliver Chikumbo and Vincent Granville. Optimal clustering and cluster identity in understanding high-dimensional data spaces with tightly distributed points. *Machine Learning and Knowledge Extraction*, 1(2):715–744, 2019. 254
- [22] Keith Conrad. *L-functions and the Riemann Hypothesis*. 2018. 2018 CTNT Summer School [\[Link\]](#). 132, 218, 221, 226
- [23] Noel Cressie. *Statistic for Spatial Data*. Wiley, revised edition, 2015. 199

- [24] D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer, second edition, 2002. Volume 1 – Elementary Theory and Methods. [151](#)
- [25] D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer, second edition, 2014. Volume 2 – General Theory and Structure. [151](#)
- [26] Tilman M. Davies and Martin L. Hazelton. Assessing minimum contrast parameter estimation for spatial and spatiotemporal log-Gaussian Cox processes. *Statistica Neerlandica*, 67(4):355–389, 2013. [245](#)
- [27] Marc Deisenroth, A. Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020. [\[Link\]](#). [54](#)
- [28] Harold G. Diamond and Wen-Bin Zhang. *Beurling Generalized Numbers*. American Mathematical Society, 2016. Mathematical Surveys and Monographs, Volume 213 [\[Link\]](#). [133](#), [229](#)
- [29] D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes – Volume I: Elementary Theory and Methods*. Springer, second edition, 2013. [200](#)
- [30] D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes – Volume II: General Theory and Structure*. Springer, second edition, 2014. [200](#)
- [31] David Coupier (Editor). *Stochastic Geometry: Modern Research Frontiers*. Wiley, 2019. [210](#)
- [32] Ding-Geng Chen (Editor), Jianguo Sun (Editor), and Karl E. Peace (Editor). *Interval-Censored Time-to-Event Data: Methods and Applications*. Chapman and Hall/CRC, 2012. [201](#)
- [33] Khaled Emam, Lucy Mosquera, and Richard Hoptroff. *Practical Synthetic Data Generation*. O'Reilly, 2020. [100](#)
- [34] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, volume 5, pages 17–61, 1960. [\[Link\]](#). [203](#)
- [35] Achim Zeileis et al. Colorspace: A toolbox for manipulating and assessing colors and palettes. *Preprint*, pages 1–45, 2019. arXiv:1903.06490 [\[Link\]](#) [\[R Library\]](#). [169](#)
- [36] Arash Farahmand. *Math 55 Lecture Notes*. 2021. University of Berkeley [\[Link\]](#). [49](#), [54](#)
- [37] W. Feller. On the Kolmogorov-Smirnov limit theorems for empirical distributions. *Annals of Mathematical Statistics*, 19(2):177–189, 1948. [\[Link\]](#). [202](#), [209](#)
- [38] Nikos Frantzikinakis. Ergodicity of the Liouville system implies the Chowla conjecture. *Preprint*, pages 1–41, 2016. arXiv [\[Link\]](#). [223](#)
- [39] P. M. Gauthier. Approximating the Riemann zeta-function by polynomials with restricted zeros. *Canadian Mathematical Bulletin*, 62(3):475–478, 2018. [\[Link\]](#). [233](#)
- [40] P. A. Van Der Geest. The binomial distribution with dependent Bernoulli trials. *Journal of Statistical Computation and Simulation*, pages 141–154, 2004. [\[Link\]](#). [148](#)
- [41] Stamatia Giannarou and Tania Stathaki. Shape signature matching for object identification invariant to image transformations and occlusion. 2007. ResearchGate [\[Link\]](#). [85](#)
- [42] Minas Gjoka, Emily Smith, and Carter Butts. Estimating clique composition and size distributions from sampled network data. *Preprint*, pages 1–9, 2013. arXiv:1308.3297 [\[Link\]](#). [203](#)
- [43] B.V. Gnedenko and A. N. Kolmogorov. *Limit Distributions for Sums of Independent Random Variables*. Addison-Wesley, 1954. [152](#)
- [44] Manuel González-Navarrete and Rodrigo Lambert. Non-markovian random walks with memory lapses. *Preprint*, pages 1–14, 2018. arXiv [\[Link\]](#). [147](#)
- [45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. [\[Link\]](#). [54](#)
- [46] Vincent Granville. Estimation of the intensity of a Poisson point process by means of nearest neighbor distances. *Statistica Neerlandica*, 52(2):112–124, 1998. [\[Link\]](#). [200](#)
- [47] Vincent Granville. *Applied Stochastic Processes, Chaos Modeling, and Probabilistic Properties of Numeration Systems*. MLTechniques.com, 2018. [\[Link\]](#). [133](#)
- [48] Vincent Granville. *Stochastic Processes and Simulations: A Machine Learning Perspective*. MLTechniques.com, 2022. [\[Link\]](#). [52](#), [60](#), [152](#), [162](#), [193](#), [194](#), [195](#), [197](#), [201](#), [203](#), [229](#), [233](#), [247](#)
- [49] Vincent Granville, Mirko Krivanek, and Jean-Paul Rasson. Simulated annealing: A proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:652–656, 1996. [73](#)
- [50] Vincent Granville and Richard L Smith. Disaggregation of rainfall time series via Gibbs sampling. *NISS Technical Report*, pages 1–21, 1996. [\[Link\]](#). [108](#)
- [51] Kristen Grauman. Shape matching. 2008. University of Texas, Austin [\[Link\]](#). [88](#)
- [52] Hui Guo et al. Eyes tell all: Irregular pupil shapes reveal gan-generated faces. *Preprint*, pages 1–7, 2021. arXiv:2109.00162 [\[Link\]](#). [255](#)

- [53] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. O'Reilly, third edition, 2023. [36](#)
- [54] Radim Halíř and Jan Flusser. Numerically stable direct least squares fitting of ellipses. *Preprint*, pages 1–8, 1998. [\[Link\]](#). [18](#), [20](#)
- [55] Peter Hall. *Introduction to the theory of coverage processes*. Wiley, 1988. [210](#)
- [56] Adam J. Harper. Moments of random multiplicative functions, II: High moments. *Algebra and Number Theory*, 13(10):2277–2321, 2019. [\[Link\]](#). [129](#), [229](#)
- [57] Adam J. Harper. Moments of random multiplicative functions, I: Low moments, better than squareroot cancellation, and critical multiplicative chaos. *Forum of Mathematics, Pi*, 8:1–95, 2020. [\[Link\]](#). [129](#), [131](#), [229](#)
- [58] Adam J. Harper. Almost sure large fluctuations of random multiplicative functions. *Preprint*, pages 1–38, 2021. arXiv [\[Link\]](#). [131](#), [223](#), [229](#)
- [59] K. Hartmann, J. Krois, and B. Waske. *Statistics and Geospatial Data Analysis*. Freie Universität Berlin, 2018. E-Learning Project SOGA [\[Link\]](#). [196](#)
- [60] D. R. Heath-Brown. Primes represented by $x^3 + 2y^3$. *Acta Mathematica*, 186:1–84, 2001. [\[Link\]](#). [225](#)
- [61] T. W. Hilberdink and M. L. Lapidus. Beurling Zeta functions, generalised primes, and fractal membranes. *Preprint*, pages 1–31, 2004. arXiv [\[Link\]](#). [132](#), [133](#), [229](#)
- [62] Christian Hill. *Learning Scientific Programming with Python*. Cambridge University Press, 2016. [\[Link\]](#). [20](#)
- [63] Robert V. Hogg, Joseph W. McKean, and Allen T. Craig. *Introduction to Mathematical Statistics*. Pearson, eighth edition, 2016. [\[Link\]](#). [54](#)
- [64] Zhiqiu Hu and Rong-Cai Yang. A new distribution-free approach to constructing the confidence region for multiple parameters. *PLOS One*, pages 1–13, 2013. [\[Link\]](#). [244](#)
- [65] Peter Humphries. The distribution of weighted sums of the Liouville function and Pólya's conjecture. *Preprint*, pages 1–33, 2011. arXiv [\[Link\]](#). [230](#)
- [66] Timothy D. Johnson. Introduction to spatial point processes. *Preprint*, page 2008. NeuroImaging Statistics Oxford (NISOx) group [\[Link\]](#)[\[Mirror\]](#). [200](#)
- [67] Chigozie Kelechi. Towards efficiency in the residual and parametric bootstrap techniques. *American Journal of Theoretical and Applied Statistics*, 5(5), 2016. [\[Link\]](#). [98](#)
- [68] Denis Kojevnikov, Vadim Marmer, and Kyungchul Song. Limit theorems for network dependent random variables. *Journal of Econometrics*, 222(2):419–427, 2021. [\[Link\]](#). [200](#)
- [69] Samuel Kotz, Tomasz Kozubowski, and Krzysztof Podgorski. *The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering, and Finance*. Springer, 2001. [214](#)
- [70] Faraj Lagum. *Stochastic Geometry-Based Tools for Spatial Modeling and Planning of Future Cellular Networks*. PhD thesis, Carleton University, 2018. [\[Link\]](#). [199](#)
- [71] Günther Last and Mathew Penrose. *Lectures on the Poisson Process*. Cambridge University Press, 2017. [199](#)
- [72] Yuk-Kam Lau, Gerald Tenenbaum, and Jie Wu. On mean values of random multiplicative functions. *Proceedings of the American Mathematical Society*, 142(2):409–420, 2013. [\[Link\]](#). [129](#), [131](#)
- [73] Gary R. Lawlor. A l'Hospital's rule for multivariable functions. *Preprint*, pages 1–13, 2013. arXiv:1209.0363 [\[Link\]](#). [114](#)
- [74] Jing Lei et al. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113:1094–1111, 2018. [\[Link\]](#). [98](#)
- [75] G. Last M.A. Klatt and D. Yogeshwaran. Hyperuniform and rigid stable matchings. *Random Structures and Algorithms*, 2:439–473, 2020. [\[Link\]](#)[\[PowerPoint\]](#). [199](#)
- [76] Jorge Mateu, Frederic P Schoenberg, and David M Diez. On distances between point patterns and their applications. *Preprint*, pages 1–29, 2010. [\[Link\]](#). [200](#)
- [77] Natarajan Meghanathan. Distribution of maximal clique size of the vertices for theoretical small-world networks and real-world networks. *Preprint*, pages 1–20, 2015. arXiv:1508.01668 [\[Link\]](#). [203](#)
- [78] Masahiro Mine. Probability density functions attached to random Euler products for automorphic L-functions. *Preprint*, pages 1–38, 2020. arXiv [\[Link\]](#). [229](#), [230](#)
- [79] Christoph Molnar. *Interpretable Machine Learning*. ChristophMolnar.com, 2022. [\[Link\]](#). [98](#), [254](#)

- [80] Marc-Andreas Muendler. Linear difference equations and autoregressive processes. 2000. University of Berkeley [Link]. 50
- [81] V. Kumar Murty. Seminar on Fermat's last theorem. In *Canadian Mathematical Society – Conference Proceedings*, volume 17, Toronto, Canada, 1995. [Link]. 226
- [82] Peter Mörters and Yuval Peres. *Brownian Motion*. Cambridge University Press, 2010. Cambridge Series in Statistical and Probabilistic Mathematics, Volume 30 [Link]. 147, 151
- [83] Jesper Møller. Introduction to spatial point processes and simulation-based inference. In *International Center for Pure and Applied Mathematics (Lecture Notes)*, Lomé, Togo, 2018. [Link][Mirror]. 200, 213, 245
- [84] Jesper Møller and Rasmus P. Waagepetersen. *An Introduction to Simulation-Based Inference for Spatial Point Processes*. Springer, 2003. 200
- [85] Jesper Møller and Rasmus P. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. CRC Press, 2007. 200
- [86] S. Ghosh N., Miyoshi, and T. Shirai. Disordered complex networks: energy optimal lattices and persistent homology. *Preprint*, pages 1–44, 2020. arXiv:2009.08811. 193
- [87] Saralees Nadarajah. A modified Bessel distribution of the second kind. *Statistica*, 67(4):405–413, 2007. [Link]. 214
- [88] Hasan Nasab, Mahdi Tavana, and Mohsen Yousefu. A new heuristic algorithm for the planar minimum covering circle problem. *Production and Manufacturing Research*, pages 142–155, 2014. [Link]. 210
- [89] Guillermo Navas-Palencia. Optimal binning: mathematical programming formulation. *Preprint*, pages 1–21, 2020. arXiv:2001.08025 [Link]. 38
- [90] Nathan Ng. Large gaps between the zeros of the Riemann zeta function. *Journal of Number Theory*, 128(3):509–556, 2007. [Link]. 233
- [91] Yosihiko Ogata. Cluster analysis of spatial point patterns: posterior distribution of parents inferred from offspring. *Japanese Journal of Statistics and Data Science*, 3:367–390, 2020. 199
- [92] Fred Park. Shape descriptor / feature extraction techniques. 2011. UCI iCAMP 2011 [Link]. 85
- [93] Yuval Peres and Allan Sly. Rigidity and tolerance for perturbed lattices. *Preprint*, pages 1–20, 2020. arXiv:1409.4490 [Link]. 193, 199
- [94] Carl Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. [Link]. 53
- [95] Alfred R.Osborne. Multidimensional Fourier series. *International Geophysics*, 97:115–145, 2010. [Link]. 121
- [96] Kamron Saniee. A simple expression for multivariate Lagrange interpolation. *SIAM Undergraduate Research Online*, 2007. SIURO [Link]. 104
- [97] Mahesh Shivanand and all. Fitting random regression models with Legendre polynomial and B-spline to model the lactation curve for Indian dairy goat of semi-arid tropic. *Journal of Animal Breeding and Genetics*, pages 414–422, 2022. [Link]. 121
- [98] Karl Sigman. Notes on the Poisson process. New York NY, 2009. IEOR 6711: Columbia University course [Link]. 199
- [99] Joshua Snoke et al. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society Series A*, 181:663–688, 2018. arXiv:1604.06651 [Link]. 36
- [100] Luuk Spreeuwiers. *Image Filtering with Neural Networks: Applications and Performance Evaluation*. PhD thesis, University of Twente, 1992. 74
- [101] J. Michael Steele. Le Cam's inequality and Poisson approximations. *The American Mathematical Monthly*, 101(1):48–54, 1994. [link]. 162
- [102] Dietrich Stoyan, Wilfrid S. Kendall, Sung Nok Chiu, and Joseph Mecke. *Stochastic Geometry and Its Applications*. Wiley, 2013. 210
- [103] E.C. Titchmarsh and D.R. Heath-Brown. *The Theory of the Riemann Zeta-Function*. Oxford Science Publications, second edition, 1987. 59, 132, 218
- [104] Chris Tofallis. Fitting equations to data with the perfect correlation relationship. *Preprint*, pages 1–11, 2015. Hertfordshire Business School Working Paper[Link]. 14
- [105] D. Umbach and K.N. Jones. A few methods for fitting circles to data. *IEEE Transactions on Instrumentation and Measurement*, 52(6):1881–1885, 2003. [Link]. 15, 18

- [106] D. A. Vaccari and H. K. Wang. Multivariate polynomial regression for identification of chaotic time series. *Mathematical and Computer Modelling of Dynamical Systems*, 13(4):1–19, 2007. [\[Link\]](#). 18
- [107] Remco van der Hofstad. *Random Graphs and Complex Networks*. Cambridge University Press, 2016. [\[Link\]](#). 202
- [108] Yu Vizilter and Sergey Zheltov. Geometrical correlation and matching of 2D image shapes. 2012. ResearchGate [\[Link\]](#). 87
- [109] Fengyun Wang and all. Bivariate Fourier-series-based prediction of surface residual stress fields using stresses of partial points. *Mathematics and Mechanics of Solids*, 2018. [\[Link\]](#). 121
- [110] Luyao Wang and Hai Cheng. Pseudo-random number generator based on logistic chaotic system. *Entropy*, 21, 2019. [\[Link\]](#). 145
- [111] Mingguang Wu, Yanjie Sun, and Yaqian Li. Adaptive transfer of color from images to maps and visualizations. *Cartography and Geographic Information Science*, pages 289–312, 2021. [\[Link\]](#). 169
- [112] Lan Wua, Yongcheng Qi, and Jingping Yang. Asymptotics for dependent Bernoulli random variables. *Statistics and Probability Letters*, pages 455–463, 2012. [\[Link\]](#). 147
- [113] Oren Yakir. Recovering the lattice from its random perturbations. *Preprint*, pages 1–18, 2020. arXiv:2002.01508 [\[Link\]](#). 199
- [114] Ruqiang Yan, Yongbin Liub, and Robert Gao. Permutation entropy: A nonlinear statistical measure for status characterization of rotary machines. *Mechanical Systems and Signal Processing*, 29:474–484, 2012. 213
- [115] Shaohong Yan, Aimin Yang, et al. Explicit algorithm to the inverse of Vandermonde matrix. In *2009 International Conference on Test and Measurement*, 2009. IEEE [\[Link\]](#). 48
- [116] D. Yogeshwaran. Geometry and topology of the boolean model on a stationary point processes : A brief survey. *Preprint*, pages 1–13, 2018. Researchgate [\[Link\]](#). 200
- [117] Tonglin Zhang. A Kolmogorov-Smirnov type test for independence between marks and points of marked point processes. *Electronic Journal of Statistics*, 8(2):2557–2584, 2014. 195

Index

- α -compositing, 58
 m -interlacing, 72, 200, 208, 211, 263
- A/B testing, 261
AdaBoost, 37, 262
additive number theory, 225, 232
adversarial learning, 255, 264
agent-based modeling, 168, 181
AI art, 177, 255
algebraic number, 134
algorithmic bias, 255
analytic continuation, 219, 226
analytic function, 132
anisotropy, 186, 207, 213
anti-aliasing, 56, 60, 236, 241
association rule, 249
attraction (point process), 198
attraction basin, 59
attractor distribution, 152, 202, 208, 215
augmented data, 36, 89, 168, 170, 264
auto-correlation, 50, 133
auto-regressive process, 50, 152, 262
- Bailey–Borwein–Plouffe formulas, 134
Bayesian classification, 75
Bayesian inference, 45
 - hierarchical models, 263
 - naive Bayes, 249Bernoulli trials, 243
Berry-Esseen inequality, 130
Bessel function, 214
Beurling primes, 133, 229
binning, 248
 - optimum binning, 38, 262binomial distribution, 208
bisection method (root finding), 159
boosted trees, 256
bootstrapping, 16, 97, 209, 255
 - percentile method, 102boundary effect, 195, 196, 201, 202, 206–208, 212
Brownian motion, 52, 147, 151, 168, 169, 184, 253, 262
 - Lévy flight, 152Brun’s theorem, 225
- Cauchy distribution, 152
Cauchy-Riemann equations, 132
causality, 263
Cayley-Hamilton theorem, 48
CDF regression, 18
censored data, 201
central limit theorem, 152, 193, 243
- chaotic dynamical system, 168, 184
character
 - principal, 221characteristic function, 214
characteristic polynomial, 48, 50–52, 153
ChatGPT, 250
Chebyshev’s bias (prime numbers), 132, 221
checksum, 260
Chi-squared test, 195
Chowla conjecture, 223
classification, 264
clique (graph theory), 203
cluster process, 197, 201, 208
clustering, 264
Collatz conjecture, 140
collision graph, 184
color model
 - RGB, 56, 242
 - RGBA, 56, 57, 77, 242color opacity, 178
color transparency, 21, 170, 242
complex random variable, 129, 229
computational complexity, 142, 250
computer vision, 13, 84
confidence band, 209
confidence interval, 45, 206, 262
confidence level, 244, 246
confidence region, 16, 30, 206, 207, 243, 255
 - dual region, 45, 244, 262conformal map, 15
confusion matrix, 248, 264
connected components, 185, 201–203, 207, 208, 251
contour level, 177, 246
contour plot, 246
convergence
 - abscissa, 221, 226
 - absolute, 113, 218, 219
 - alternating series, 219
 - conditional, 113, 131, 221
 - Dirichlet test, 219convergence acceleration, 60
convex linear combination, 109
convolution of distributions, 214
copula, 3, 100, 239, 255
cosine distance, 251
counting measure, 194
covariance matrix, 91, 243
covering (stochastic), 210
covering problem, 209
credible interval, 45

credible region (Bayesian), 244, 262
 critical line (number theory), 114
 cross-validation, 16, 184, 248
 cuban primes, 225
 curse of dimensionality, 116
 curve fitting, 27

 data video, 168, 177
 decision tree, 37
 Dedekind zeta function, 132, 229
 deep neural network, 74, 263
 dense set (topology), 223
 density estimation, 200
 diamond-square algorithm, 169
 Diehard tests of randomness, 131
 dimensionality reduction, 17
 Dirichlet character, 132, 133, 220, 225
 modulo 4, 221, 226, 227
 Dirichlet eta function, 233
 Dirichlet functional equation, 132, 226, 227
 Dirichlet series, 129
 Dirichlet's theorem, 132, 221, 223, 227
 Dirichlet- L function, 132, 220, 227
 disaggregation, 115
 discrete Fourier series, 120
 discrete orthogonal functions, 120
 dissimilarity metric, 251
 distributed architecture, 247
 distribution
 Cauchy, 152
 Fréchet, 52, 152
 Gaussian, 243
 generalized logistic, 91, 197
 Hotelling, 244
 Laplace, 214
 logistic, 17
 Lévy, 152
 modified Bessel, 214
 Poisson-binomial, 162, 216
 Poisson-exponential, 193
 Rademacher, 129, 130
 Rayleigh, 208, 209, 215
 Weibull, 52, 152, 208
 domain of attraction, 202
 dot product, 15
 dummy variable, 37
 dyadic map, 133
 dynamical systems, 133, 202
 chaotic systems, 168, 184
 dyadic map, 133
 ergodicity, 133
 logistic map, 133
 shift map, 133
 stochastic, 169

 edge effect (statistics), 201
 eigenvalue, 14, 53, 91, 264
 power iteration, 93
 elbow rule, 156, 201, 208, 253
 elliptic curve, 225
 EM algorithm, 36, 263

 empirical distribution, 17, 97, 121, 193, 196, 202, 208, 213, 215, 227, 256
 multivariate, 130
 empirical quantiles, 102
 ensemble methods, 37, 84, 256
 entropy, 187, 213, 249
 equidistribution modulo 1, 136
 equilibrium distribution, 169
 Erdős-Rényi model, 203
 ergodicity, 133, 169, 206, 208, 214
 Euler product, 129, 219, 226
 random, 229
 Euler's transform, 233
 evolutionary process, 169
 experimental design, 261
 experimental math, 57, 217
 explainable AI, 3, 14, 36, 75, 84, 91, 156, 172, 210, 254
 exploratory analysis, 260
 exponential decay, 41
 exponential sums, 226
 extrapolation, 109
 extreme value theory, 152, 215

 feature attribution, 254
 feature importance, 254
 feature selection, 16, 98, 247
 Fermat's last theorem, 226
 fixed-point algorithm, 60, 90, 156, 263
 flag vector, 249, 261
 Fourier series, 120
 Fourier transform, 214
 fractal dimension, 52
 fractional part function, 135
 Frobenius norm, 91
 Fruchterman and Rheingold algorithm, 252
 Fréchet distribution, 52, 152
 fuzzy classification, 57

 Gamma function, 52, 152
 GAN (generative adversarial networks), 3, 36, 259, 263
 Gaussian circle problem, 232
 Gaussian distribution, 243
 Gaussian mixture, 36, 71
 Gaussian primes, 132, 228
 Gaussian process, 50, 262
 general linear model, 14
 generalized linear model, 14, 49
 generalized logistic distribution, 91, 207
 generative adversarial networks, 3, 36, 172, 255, 259, 263
 generative AI, 168, 181
 generative model, 36, 53, 100, 167, 168, 170, 177, 184, 264
 geostatistics, 103
 GIS, 117
 Glivenko-Cantelli theorem, 227
 GMM (Gaussian mixture model), 3, 36, 70, 259, 263
 Goldbach's conjecture, 225
 goodness-of-fit, 57, 248
 GPU-based clustering, 72
 gradient (optimization), 156

gradient boosting, 262
 gradient operator, 16
 Gram-Schmidt orthogonalization, 120
 graph, 201
 collision graph, 184
 connected components, 185, 207, 251
 directed, 185
 edge, 201
 Fruchterman-Reingold, 185
 nearest neighbor graph, 203, 207
 node, 201, 203
 random graph, 202
 random nearest neighbor graph, 202
 tree, 185
 undirected, 201–203, 208
 vertex, 201
 graph database, 252
 graph theory, 201
 GraphViz, 185
 greedy algorithm, 113, 232
 grid search, 156, 171
 half-tone (music), 239
 Hartman–Wintner theorem, 147
 hash table, 143, 187, 212, 249, 250
 sparse, 250
 Hausdorff distance, 88
 Hellinger distance, 3, 256
 Hermite polynomials, 120
 hexagonal lattice, 201
 hidden decision trees, 37, 38, 262
 hidden layer, 74
 hidden process, 193, 211, 215
 hierarchical clustering, 74, 250
 Hilbert primes, 228
 histogram equalization, 72, 74
 Hoeffding inequality, 150
 homogeneity (point process), 162, 200
 Hotelling distribution, 244
 Hurst exponent, 52
 hyperparameter, 30, 57, 104, 171
 identifiability, 211, 213
 ill-conditioned problem, 27, 53, 93, 264
 image segmentation, 74
 imputation (missing values), 255
 index
 index discrepancy, 213
 intensity (stochastic process), 193, 200, 207
 interarrival times, 151, 193, 202, 206, 213
 standardized, 214
 interlaced processes, 200
 Internet of Things, 193
 inverse distance weighting, 105
 inverse square law, 181
 iterated logarithm, 130, 131, 147
 Itô integral, 53
 K-means clustering, 32, 33
 key-value pair, 38, 249
 Kolmogorov-Smirnov test, 130, 195, 202, 256
 kriging, 113
 Kronecker's theorem, 223, 231
 Lagrange interpolation, 53
 Lagrange multiplier, 16, 263
 Laplace distribution, 214
 large language models, 250, 263
 Lasso regression, 16, 264
 lattice, 199
 perturbed lattice, 193
 shifted, 201
 stretched, 201
 law of the iterated logarithm, 130, 131, 147, 223, 229
 Le Cam's theorem, 162, 194
 least absolute residuals, 102
 link function, 14, 17
 Liouville function, 220, 231
 LLM (large language model), 250, 263
 log-polar map, 15
 logistic distribution, 17, 200
 logistic map, 133
 logistic regression, 17
 unsupervised, 34
 logit function, 263
 Lévy distribution, 152
 Lévy flight, 152
 Map-reduce, 247
 marketing attribution, 261
 Markov chain, 50
 MCMC, 129
 Mathematica, 246
 MaxCliqueDyn algorithm, 203
 maximum likelihood estimation, 245, 263, 264
 mean squared error, 16, 31
 medoid, 32
 Mersenne twister, 30, 133, 136, 149
 Mertens function, 220
 minimum contrast estimation, 171, 210, 213, 245
 mixture model, 30, 46, 169, 177, 193, 200, 201, 208,
 246, 264
 blending, 169
 model fitting, 57, 263
 model identifiability, 16
 modulus (complex number), 153, 219
 Monte Carlo simulations, 129, 263
 morphing (computer vision), 168
 moving average, 158
 multidimensional Fourier series, 121
 multiple root, 114
 multiplicative function
 completely multiplicative, 129, 131, 220, 221, 232
 Rademacher, 129
 Möbius function, 220
 N-body problem, 181
 n-gram (NLP), 250
 naive Bayes, 249, 262
 natural language generation, 250, 263
 natural language processing, 37, 250
 nearest neighbor interpolation, 102, 105

nearest neighbors, 193, 203, 209, 263
 nearest neighbor distances, 207–209, 211, 215
 nearest neighbor graph, 207
 NetworkX, 185
 neural network, 74
 hidden layer, 74
 hyperparameter, 76
 neuron, 74, 263
 seq2seq, 167
 sparse, 70
 very deep, 74
 Newton's method, 156
 NLG (natural language generation), 250, 263
 node (decision tree), 38, 256, 262
 perfect node, 45
 usable node, 39
 node (interpolation), 114
 normal number, 130, 223, 227
 strongly normal, 131
 numerical stability, 48

 Omega function, 220, 226
 order statistics, 215
 ordinary least squares, 51, 102, 120
 orthogonal function, 120
 Otsuka–Ochiai coefficient, 251
 outliers, 215, 253
 overfitting, 16, 213, 255, 256, 262

 palette, 168, 242
 parametric bootstrap, 21, 30, 36, 98, 209, 255, 262, 263
 partial derivative, 114
 partial least squares, 14
 path (graph theory), 201
 percentile bootstrap, 102
 permutation
 entropy, 213
 random permutation, 212
 perturbed lattices, 193
 Plotly, 177
 point count distribution, 194, 207, 210
 point process
 attractive, 208
 cluster process
 Matérn, 199
 Neyman–Scott, 199
 non-homogeneous, 162, 200
 perturbed lattice process, 199
 radial, 200
 renewal process, 199
 repulsive, 198
 Poisson point process, 151, 162, 193, 207
 Poisson-binomial distribution, 162, 193, 216
 Poisson-exponential distribution, 193
 positive semidefinite (matrix), 49, 92
 power iteration, 93
 preconditioning, 93
 prediction interval, 16, 97, 102
 predictive power, 38, 45, 248, 249, 254
 prime test (of randomness), 131, 142, 148
 principal component analysis, 49, 254, 262
 probability generating function, 148
 proxy space, 246
 pseudo-inverse matrix, 49
 pseudo-random numbers, 149, 253
 combined generators, 145
 congruential generator, 136
 Diehard tests, 131, 143
 Mersenne twister, 136, 149, 170
 prime test, 131, 148
 strongly random, 131, 134
 TestU01, 131
 Pólya conjecture, 222

 quadratic irrational, 133, 136, 142
 quantile, 244, 263
 empirical, 102, 256
 weighted, 102
 quantile function, 100, 121, 193, 197, 208
 quantile regression, 16

 R-squared, 16, 36, 171
 Rademacher distribution, 130
 Rademacher function, 129, 223, 229
 random, 131
 random function, 161
 random graph, 202, 203
 random multiplicative function, 129
 Rademacher, 131
 random permutation, 212
 random variable
 complex, 129
 random walk, 147, 171, 262
 first hitting time, 148, 151
 zero crossing, 147
 Rayleigh distribution, 208, 209, 215
 Rayleigh test, 208
 records, 215
 regression splines, 14
 regular expression, 250, 260
 reinforcement learning, 264
 rejection sampling, 256
 renewal process, 199
 repulsion (point process), 198, 209
 repulsion basin, 219
 resampling, 97, 209
 Riemann Hypothesis, 108, 114
 Generalized, 131, 221, 225, 227
 Riemann zeta function, 114, 129, 132, 227, 229
 root mean squared error, 57

 scaling factor, 207, 215
 seed (random number generator), 143, 170, 256
 semi-supervised learning, 264
 shape signature, 85
 Shapley value, 254
 Shepard's method, 105
 shift map, 133
 sigmoid function, 263
 simplex, 230
 singular value decomposition, 14, 264
 singularity, 187

six degrees of separation, 252
Sklar's theorem, 256
smoothing parameter, 104
spatial statistics, 103, 199
spectral domain, 169
spline regression, 121
square root (matrix), 49, 92
square-free integer, 130, 143, 223
stable distribution, 152, 170, 214
state space, 169
stationary distribution, 53
stationary process, 50, 152, 169, 184, 195, 200, 207
stepwise regression, 99
stochastic convergence, 169
stochastic function, 52
stochastic geometry, 210
stochastic process, 193
stochastic residues, 211
stop word (NLP), 250
stretching (point process), 201
Sturm-Liouville theory, 120
superimposition (point processes), 200
supervised classification, 72
surface plot, 177
swarm optimization, 28, 263
synthetic data, 14, 28, 30, 53, 89, 91, 113, 119, 129, 142, 148, 156, 170, 177, 184, 217, 244, 255
synthetic metric, 249

Tarjan's algorithm, 251
tensor, 75
text normalization, 250
Theil-Sen estimator, 102
time series, 51
 auto-regressive, 52, 152
 disaggregation, 108
 Hurst exponent, 52
 non-periodic, 26
total least squares, 14
training set, 102, 184, 248
transcendental number, 134
transformer, 74, 167
tree (graph theory), 185
twin primes, 225

universality property, 219, 223, 225
unsupervised clustering, 72
unsupervised learning, 34, 264

validation set, 16, 57, 102, 184, 248, 255, 256
Vandermonde matrix, 48, 53
vertex, 193, 201, 202, 215
video compression
 FFmpeg, 56, 60

Waring's problem, 225
Watts and Strogatz model, 252
Weibull distribution, 52, 152, 208, 215
weighted least squares, 14
weighted quantiles, 102
weighted regression, 17
white noise, 28, 50, 152, 262