

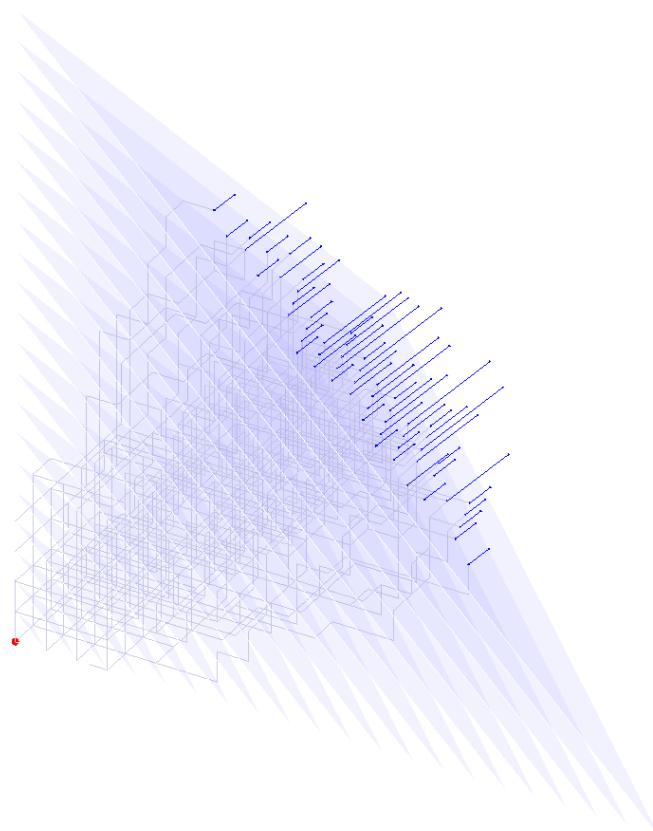
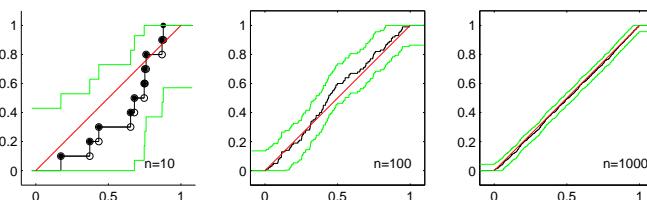
Computational Statistical Experiments in MATLAB

Raazesh Sainudiin and Dominic Lee,
Laboratory for Mathematical Statistical Experiments, Christchurch Centre
Christchurch, New Zealand

©2007–2016 Raazesh Sainudiin. ©2008–2013 Dominic Lee.

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 4.0 International License. To view a copy of this license, visit
<https://creativecommons.org/licenses/by-nc-sa/4.0/>.

This work was partially supported by NSF grant DMS-03-06497 and NSF/NIGMS grant DMS-02-01037.



Contents

1 Preliminaries	14
1.1 Elementary Set Theory	14
1.2 Natural Numbers, Integers and Rational Numbers	17
1.3 Real Numbers	21
1.4 Introduction to MATLAB	25
1.5 Permutations, Factorials and Combinations	28
1.6 Array, Sequence, Limit,	30
1.7 Elementary Number Theory	36
2 Probability Model	38
2.1 Experiments	38
2.2 Probability	40
2.2.1 Consequences of our Definition of Probability	42
2.2.2 Sigma Algebras of Typical Experiments*	44
2.3 Conditional Probability	46
2.3.1 Independence and Dependence	49
3 Random Variables	54
3.1 Basic Definitions	54
3.2 An Elementary Discrete Random Variable	56
3.3 An Elementary Continuous Random Variable	57
3.4 Expectations	60
3.5 Stochastic Processes	63
4 Random Numbers	65
4.1 Physical Random Number Generators	65
4.2 Pseudo-Random Number Generators	65
4.2.1 Linear Congruential Generators	66
4.2.2 Generalized Feedback Shift Register and the “Mersenne Twister” PRNG	69

CONTENTS	3
5 Statistics	72
5.1 Data and Statistics	72
5.2 Exploring Data and Statistics	79
5.2.1 Univariate Data	79
5.2.2 Bivariate Data	81
5.2.3 Trivariate Data	82
5.2.4 Multivariate Data	82
5.3 Loading and Exploring Real-world Data	84
5.3.1 Geological Data	84
5.3.2 Metereological Data	88
5.3.3 Textual Data	91
5.3.4 Machine Sensor Data	92
5.3.5 Biological Data	93
6 Common Random Variables	94
6.1 Inversion Sampler for Continuous Random Variables	94
6.2 Some Simulations of Continuous Random Variables	95
6.3 Continuous Random Variables	95
6.4 Discrete Random Variables	109
6.5 Inversion Sampler for Discrete Random Variables	109
6.6 Some Simulations of Discrete Random Variables	109
6.7 Sir Francis Galton's Quincunx	125
6.8 Random Vectors	131
6.9 von Neumann Rejection Sampler (RS)	138
6.10 Importance Resampler	143
6.11 Other Continuous Random Variables	144
6.12 Other Random Vectors	145
6.13 Problems	146
7 Statistical Experiments	148
7.1 Introduction	148
7.2 Some Common Experiments	148
7.3 Typical Decision Problems with Experiments	150
8 Limits of Random Variables	151
8.1 Convergence of Random Variables	151
8.2 Some Basic Limit Laws of Statistics	154

CONTENTS	4
9 Finite Markov Chains	157
9.1 Introduction	157
9.2 Random Mapping Representation and Simulation	165
9.3 Irreducibility and Aperiodicity	171
9.4 Stationarity	174
9.5 Reversibility	176
9.6 Metropolis-Hastings Markov chain	180
9.7 Glauber Dynamics	183
9.7.1 Random Walks on \mathbb{Z} and the reflection principle	188
9.8 Coupling from the past	188
9.8.1 <i>Algorithm – Coupling from the past.</i>	189
10 General Markov Chains	192
10.1 Markov Chain Monte Carlo	192
10.1.1 <i>Algorithm – Single-component Metropolis-Hastings sampler.</i>	196
10.1.2 <i>Algorithm – Gibbs sampler.</i>	197
10.2 Exercises	198
11 Fundamentals of Estimation	199
11.1 Introduction	199
11.2 Point Estimation	199
11.3 Some Properties of Point Estimators	200
11.4 Confidence Set Estimation	203
11.5 Likelihood	205
12 Maximum Likelihood Estimator	208
12.1 Introduction to Maximum Likelihood Estimation	208
12.2 Practical Excursion in One-dimensional Optimisation	209
12.3 Properties of the Maximum Likelihood Estimator	218
12.4 Fisher Information	218
12.5 Delta Method	224
13 Maximum Likelihood Estimation for Multiparameter Models	228
13.1 Introduction	228
13.2 Practical Excursion in Multi-dimensional Optimisation	228
13.3 Confidence Sets for Multiparameter Models	232
14 Non-parametric DF Estimation	236
14.1 Estimating DF	237
14.2 Plug-in Estimators of Statistical Functionals	242

CONTENTS	5
15 Bootstrap	244
15.1 Non-parametric Bootstrap for Confidence Sets	244
15.2 Parametric Bootstrap for Confidence Sets	247
15.3 Empirical distribution function	250
15.4 Nonparametric bootstrap	252
15.4.1 Bootstrap estimates of bias, variance and mean squared error	252
15.4.2 Percentile interval	254
15.4.3 <i>Properties of the percentile interval.</i>	255
15.4.4 Bias-corrected and accelerated (BCA) interval	255
15.4.5 Properties of the BCA interval	256
15.5 Extension to multivariate data and linear regression	256
15.5.1 Confidence intervals for regression coefficients	258
15.5.2 Alternative bootstrap method for regression	259
15.6 Extension to dependent data	261
15.6.1 Block bootstrap	261
15.7 Exercises	262
16 Monte Carlo Estimation	264
16.1 Monte Carlo Integral Estimation	264
16.2 Variance Reduction via Importance Sampling	268
16.3 Sequential Monte Carlo Methods	270
16.3.1 Sequential Importance Sampling	270
16.3.2 Population MCMC	270
16.3.3 Genetic Monte Carlo Algorithms	270
16.4 Monte Carlo Optimisation	270
17 Hypothesis Testing	271
17.1 Introduction	271
17.2 The Wald Test	272
17.3 A Composite Hypothesis Test	274
17.4 p-values	276
17.5 Permutation Test for the equality of any two DFs	278
17.6 Pearson's Chi-Square Test for Multinomial Trials	280

CONTENTS	6
18 Nonparametric Density Estimation	284
18.1 Histogram	284
18.1.1 <i>Definition.</i>	289
18.1.2 Drawbacks of the histogram	289
18.1.3 Selection of histogram bin width	289
18.2 Kernel density estimation	290
18.2.1 Examples of kernel functions	291
18.2.2 Bandwidth selection	292
18.2.3 Adjustment at boundaries	293
18.3 Extension to multivariate data	294
18.3.1 Bandwidth selection	294
18.4 Smoothed bootstrap	295
18.4.1 Generating from a kernel density	296
18.4.2 Smoothed bootstrap procedure for generating bootstrap samples	296
18.5 Exercises	297
19 Bayesian Experiments	300
19.1 A Bayesian Quincunx	300
19.2 Conjugate Families	300
19.3 Bayesian Model Selection	300
20 Statistical Learning	301
20.1 Supervised Learning	301
20.2 Unsupervised Learning	301
20.3 Classification	301
20.4 Regression	301
21 Appendix	317
21.1 Code	317
21.2 Data	325
22 Student Projects	328
22.1 Testing the Approximation of π by Buffon's Needle Test	329
22.1.1 Introduction & Motivation	329
22.1.2 Materials & Methods	330
22.1.3 Results	335
22.1.4 Conclusion	335
22.2 Estimating the Binomial probability p for a Galton's Quincunx	337
22.2.1 Motivation & Introduction	337

CONTENTS

7

22.2.2 Materials and Methods	338
22.2.3 Statistical Methodology	339
22.2.4 Results & Conclusion	340
22.3 Investigation of a Statistical Simulation from the 19th Century	341
22.3.1 Introduction and Motivation	341
22.3.2 Statistical Methodology	342
22.3.3 Results	344
22.3.4 Conclusion	346
22.4 Testing the average waiting time for the Orbiter Bus Service	349
22.4.1 Motivation	349
22.4.2 Method	350
22.4.3 Results	351
22.4.4 Discussion	351
22.4.5 Conclusion	354
22.5 Diameter of <i>Dosinia</i> Shells	359
22.5.1 Introduction and Objective	359
22.5.2 Materials and Methods	359
22.5.3 Results	360

List of Tables

1.1	Symbol Table: Sets and Numbers	24
6.1	Some continuous RVs that can be simulated from using Algorithm 3.	104
6.2	Random Variables with PDF, Mean and Variance	127
13.1	Summary of the Method of Moment Estimator (MME) and the Maximum Likelihood Estimator (MLE) for some IID Experiments.	232
17.1	Outcomes of an hypothesis test.	272
17.2	Some terminology in hypothesis testing.	272
17.3	Evidence scale against the null hypothesis in terms of the range of p – value.	277

List of Figures

1.1	Union and intersection of sets shown by Venn diagrams	15
1.2	These Venn diagram illustrate De Morgan’s Laws.	16
1.3	A function f (“father of”) from \mathbb{X} (a set of children) to \mathbb{Y} (their fathers) and its inverse (“children of”).	19
1.4	A pictorial depiction of addition and its inverse. The domain is plotted in orthogonal Cartesian coordinates	20
1.5	A depiction of the real line segment $[-10, 10]$	22
1.6	Point plot and stem plot of the finite sequence $\langle b_{1:10} \rangle$ declared as an array.	32
1.7	A plot of the sine wave over $[-2\pi, 2\pi]$	34
2.1	A binary tree whose leaves are all possible outcomes.	40
2.2	First ball number in 1114 NZ Lotto draws from 1987 to 2008.	43
3.1	The Indicator function of event $A \in \mathcal{F}$ is a RV $\mathbf{1}_A$ with DF F	55
3.2	A RV X from a sample space Ω with 8 elements to \mathbb{R} and its DF F	56
3.3	The Indicator Function $\mathbf{1}_{\mathbb{H}}$ of the event ‘Heads occurs’, for the experiment ‘Toss 1 times,’ \mathcal{E}_{θ}^1 , as the RV X from the sample space $\Omega = \{\mathbb{H}, \mathbb{T}\}$ to \mathbb{R} and its DF F . The probability that ‘Heads occurs’ and that ‘Tails occurs’ are $f(1; \theta) = \mathbf{P}_{\theta}(X = 1) = \mathbf{P}_{\theta}(\mathbb{H}) = \theta$ and $f(0; \theta) = \mathbf{P}_{\theta}(X = 0) = \mathbf{P}_{\theta}(\mathbb{T}) = 1 - \theta$, respectively.	58
3.4	A plot of the PDF and DF or CDF of the Uniform($0, 1$) continuous RV X	59
3.5	Mean ($\mathbf{E}_{\theta}(X)$), variance ($\mathbf{V}_{\theta}(X)$) and the rate of change of variance ($\frac{d}{d\theta} \mathbf{V}_{\theta}(X)$) of a Bernoulli(θ) RV X as a function of the parameter θ	62
4.1	The linear congruential sequence of <code>LinConGen(256, 137, 0, 123, 257)</code> with non-maximal period length of 32 as a line plot over $\{0, 1, \dots, 256\}$, scaled over $[0, 1]$ by a division by 256 and a histogram of the 256 points in $[0, 1]$ with 15 bins.	67
4.2	The LCG called <code>RANDU</code> with $(m, a, c) = (2147483648, 65539, 0)$ has strong correlation between three consecutive points as: $x_{i+2} = 6x_{k+1} - 9x_k$. The two plots are showing (x_i, x_{i+1}, x_{i+2}) from two different view points.	69
4.3	Triplet point clouds from the “Mersenne Twister” with two different seeds (see Lab-work 42).	71
5.1	Sample Space, Random Variable, Realisation, Data, and Data Space.	72
5.2	Data Spaces $\mathbb{X} = \{0, 1\}^2$ and $\mathbb{X} = \{0, 1\}^3$ for two and three Bernoulli trials, respectively.	73

5.3	Plot of the DF of Uniform(0, 1), five IID samples from it, and the ECDF \hat{F}_5 for these five data points $x = (x_1, x_2, x_3, x_4, x_5) = (0.5164, 0.5707, 0.0285, 0.1715, 0.6853)$ that jumps by $1/5 = 0.20$ at each of the five samples.	78
5.4	Frequency, Relative Frequency and Density Histograms	80
5.5	Frequency, Relative Frequency and Density Histograms	81
5.6	2D Scatter Plot	82
5.7	3D Scatter Plot	83
5.8	Plot Matrix of uniformly generated data in $[0, 1]^5$	83
5.9	Matrix of Scatter Plots of the latitude, longitude, magnitude and depth of the 22-02-2011 earth quakes in Christchurch, New Zealand.	86
5.10	Google Earth Visualisation of the earth quakes	87
5.11	Daily rainfalls in Christchurch since March 27 2010	89
5.12	Daily temperatures in Christchurch for one year since March 27 2010	90
5.13	Wordle of JOE 2010	91
5.14	Double Pendulum	92
6.1	A plot of the PDF, DF or CDF and inverse DF of the Uniform($-1, 1$) RV X	95
6.2	Visual Cognitive Tool GUI: Inversion Sampling from $X \sim \text{Uniform}(-5, 5)$	96
6.3	Density and distribution functions of $\text{Exponential}(\lambda)$ RVs, for $\lambda = 1, 10, 10^{-1}$, in four different axes scales.	97
6.4	Visual Cognitive Tool GUI: Inversion Sampling from $X \sim \text{Exponential}(0.5)$	98
6.5	The PDF f , DF F , and inverse DF $F^{[-1]}$ of the the Exponential($\lambda = 1.0$) RV.	99
6.6	Visual Cognitive Tool GUI: Inversion Sampling from $X \sim \text{Laplace}(5)$	101
6.7	Visual Cognitive Tool GUI: Inversion Sampling from $X \sim \text{Cauchy}$	103
6.8	Unending fluctuations of the running means based on n IID samples from the Standard Cauchy RV X in each of five replicate simulations (blue lines). The running means, based on n IID samples from the Uniform($0, 10$) RV, for each of five replicate simulations (magenta lines).	104
6.9	Density and distribution function of several $\text{Normal}(\mu, \sigma^2)$ RVs.	105
6.10	PDF and CDF of $X \sim \text{Gamma}(\beta = 0.1, \alpha)$ with $\alpha \in \{1, 2, 3, 4, 5\}$	109
6.11	The DF $F(x; 0.3, 0.7)$ of the de Moivre($0.3, 0.7$) RV and its inverse $F^{[-1]}(u; 0.3, 0.7)$	113
6.12	Mean and variance of a $\text{Geometric}(\theta)$ RV X as a function of the parameter θ	117
6.13	PDF of $X \sim \text{Geometric}(\theta = 0.5)$ and the relative frequency histogram based on 100 and 1000 samples from X	118
6.14	PDF of $X \sim \text{Binomial}(n = 10, \theta = 0.5)$ and the relative frequency histogram based on 100,000 samples from X	121
6.15	PDF of $X \sim \text{Poisson}(\lambda = 10)$ and the relative frequency histogram based on 1000 samples from X	123
6.16	Figures from Sir Francis Galton, F.R.S., <i>Natural Inheritance</i> , , Macmillan, 1889.	126

6.17 Quincunx on the Cartesian plane. Simulations of Binomial($n = 10, \theta = 0.5$) RV as the x-coordinate of the ordered pair resulting from the culmination of sample trajectories formed by the accumulating sum of $n = 10$ IID Bernoulli($\theta = 0.5$) random vectors over $\{(1, 0), (0, 1)\}$ with probabilities $\{\theta, 1 - \theta\}$, respectively. The blue lines and black asterisks perpendicular to and above the diagonal line, i.e. the line connecting $(0, 10)$ and $(10, 0)$, are the density histogram of the samples and the PDF of our Binomial($n = 10, \theta = 0.5$) RV, respectively.	133
6.18 Visual Cognitive Tool GUI: Quincunx.	135
6.19 Septcunx on the Cartesian co-ordinates. Simulations of Multinomial($n = 2, \theta_1 = 1/3, \theta_2 = 1/3, \theta_3 = 1/3$) \vec{R} as the sum of n IID de Moivre($\theta_1 = 1/3, \theta_2 = 1/3, \theta_3 = 1/3$) \vec{R} s over $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ with probabilities $\{\theta_1, \theta_2, \theta_3\}$, respectively. The blue lines perpendicular to the sample space of the Multinomial($3, \theta_1, \theta_2, \theta_3$) \vec{R} , i.e. the plane in \mathbb{R}^3 connecting $(n, 0, 0)$, $(0, n, 0)$ and $(0, 0, n)$, are the density histogram of the samples.	136
6.20 Visual Cognitive Tool GUI: Septcunx.	137
6.21 Visual Cognitive Tool GUI: Rejection Sampling from $X \sim \text{Normal}(0, 1)$ with PDF f based on proposals from $Y \sim \text{Laplace}(1)$ with PDF g	139
6.22 Rejection Sampling from $X \sim \text{Normal}(0, 1)$ with PDF f based on 100 proposals from $Y \sim \text{Laplace}(1)$ with PDF g	140
7.1 Geometry of the Θ 's for de Moivre[k] Experiments with $k \in \{1, 2, 3, 4\}$	149
8.1 Distribution functions of several $\text{Normal}(\mu, \sigma^2)$ RVs for $\sigma^2 = 1, \frac{1}{10}, \frac{1}{100}, \frac{1}{1000}$	151
9.1 Transition Diagram of Flippant Freddy's Jumps.	158
9.2 The probability of being back in rollovia in t time steps after having started there under transition matrix P with (i) $p = q = 0.5$ (blue line with asterisks), (ii) $p = 0.85, q = 0.35$ (black line with dots) and (iii) $p = 0.15, q = 0.95$ (red line with pluses).	160
9.3 Transition Diagram of Dry and Wet Days in Christchurch.	162
9.4 Transition diagram over six lounges (without edge probabilities).	170
9.5 Stochastic Optimization with Metropolis chain.	182
9.6 The sample at time step 10^6 from the Glauber dynamics for the hardcore model on 100×100 regular torus grid. A red site is occupied while a blue site is vacant.	186
11.1 Density and Confidence Interval of the Asymptotically Normal Point Estimator	204
11.2 Data Spaces $\mathbb{X}_1 = \{0, 1\}$, $\mathbb{X}_2 = \{0, 1\}^2$ and $\mathbb{X}_3 = \{0, 1\}^3$ for one, two and three IID Bernoulli trials, respectively and the corresponding likelihood functions.	207
11.3 100 realisations of $C_{10}, C_{100}, C_{1000}$ based on samples of size $n = 10, 100$ and 1000 drawn from the Bernoulli($\theta^* = 0.5$) RV as per Labwork 254. The MLE $\hat{\theta}_n$ (cyan dot) and the log-likelihood function (magenta curve) for each of the 100 replications of the experiment for each sample size n are depicted. The approximate normal-based 95% confidence intervals with blue boundaries are based on the exact $\text{se}_n = \sqrt{\theta^*(1 - \theta^*)/n} = \sqrt{1/4}$, while those with red boundaries are based on the estimated $\widehat{\text{se}}_n = \sqrt{\hat{\theta}_n(1 - \hat{\theta}_n)/n}$. The fraction of times the true parameter $\theta^* = 0.5$ was engulfed by the exact and approximate confidence interval (empirical coverage) over the 100 replications of the experiment for each of the three sample sizes are given by the numbers after $\text{Cvrg.} =$ and $\sim =$, above each sub-plot, respectively.	207

12.1 Plot of $\log(L(1, 0, 0, 0, 1, 1, 0, 0, 1, 0; \theta))$ as a function of the parameter θ over the parameter space $\Theta = [0, 1]$ and the MLE $\hat{\theta}_{10}$ of 0.4 for the coin-tossing experiment.	210
12.2 Plot of $\log(L(\lambda))$ as a function of the parameter λ and the MLE $\hat{\lambda}_{132}$ of 0.1102 for Fenemore-Wang Orbiter Waiting Times Experiment from STAT 218 S2 2007. The density or PDF and the DF at the MLE of 0.1102 are compared with a histogram and the empirical DF.	213
12.3 Comparing the Exponential($\hat{\lambda}_{6128} = 28.6694$) PDF and DF with a histogram and empirical DF of the times (in units of days) between earth quakes in NZ. The epicentres of 6128 earth quakes are shown in left panel.	214
12.4 The ML fitted Rayleigh($\hat{\alpha}_{10} = 2$) PDF and a histogram of the ocean wave heights.	216
12.5 Plot of $\log(L(\lambda))$ as a function of the parameter λ , the MLE $\hat{\lambda}_{132} = 0.1102$ and 95% confidence interval $C_n = [0.0914, 0.1290]$ for Fenemore-Wang Orbiter Waiting Times Experiment from STAT 218 S2 2007. The PDF and the DF at (1) the MLE 0.1102 (black), (2) lower 95% confidence bound 0.0914 (red) and (3) upper 95% confidence bound 0.1290 (blue) are compared with a histogram and the empirical DF.	223
13.1 Plot of Levy density as a function of the parameter $(x, y) \in [-10, 10]^2$ scripted in Labwork 251.	229
13.2 Plot of the “well-behaved” (uni-modal and non-spiky) $\log(L((x_1, x_2, \dots, x_{100}); \lambda, \zeta))$, based on 100 samples $(x_1, x_2, \dots, x_{100})$ drawn from the Lognormal($\lambda^* = 10.36, \zeta^* = 0.26$) as per Labwork 253.	231
14.1 Plots of ten distinct ECDFs \hat{F}_n based on 10 sets of n IID samples from Uniform(0, 1) RV X , as n increases from 10 to 100 to 1000. The DF $F(x) = x$ over $[0, 1]$ is shown in red. The script of Labwork 255 was used to generate this plot.	237
14.2 The empirical DFs $\hat{F}_n^{(1)}$ from sample size $n = 10, 100, 1000$ (black), is the point estimate of the fixed and known DF $F(x) = x, x \in [0, 1]$ of Uniform(0, 1) RV (red). The 95% confidence band for each \hat{F}_n are depicted by green lines.	239
14.3 The empirical DF \hat{F}_{6128} for the inter earth quake times and the 95% confidence bands for the non-parametric experiment.	239
14.4 The empirical DF \hat{F}_{132} for the Orbiter waiting times and the 95% confidence bands for the non-parametric experiment.	240
14.5 The empirical DFs $\hat{F}_{n_1}^{(1)}$ with $n_1 = 56485$, for the web log times starting October 1, and $\hat{F}_{n_2}^{(2)}$ with $n_2 = 53966$, for the web log times starting October 2. Their 95% confidence bands are indicated by the green.	241
15.1 Data from Bradley Efrons LSAT and GPA scores for fifteen individuals (left). The confidence interval of the sample correlation, the plug-in estimate of the population correlation, is obtained from the sample correlation of one thousand bootstrapped data sets (right).	248
17.1 Plot of power function $\beta(\lambda)$ for different values of the critical value c and the size α as function of the critical values.	275
17.2 The smallest α at which a size α test rejects the null hypothesis H_0 is the p – value.	276
18.1 Histogram estimates for the with nine different bin-widths of $\{2, 4, 6, 8, 11, 14, 17, 20, 35\}$. The fifth histogram with a bin-width of 11 is the optimal one with the right amount of smoothing. The ones with smaller bin-widths are under-smoothed and those with larger bin-widths are over-smoothed.	285

22.1 Example of Needle Tosses	331
22.2 Explaining the outcomes of the needle toss	331
22.3 Outcome Space of Buffon's Needle Experiment for General Case	332
22.4 Plot showing the midpoints mapping back to specific values on the x axis.	343
22.5 Plot showing both the GN and GDN CDFs. They are very similar.	344
22.6 Plot showing the empirical DF of our results against GDN. Our values take on a stair case appearance and are very close to GDN. The main deviations occur mostly in the tails.	345
22.7 Plot showing the Standard Normal Distribution against Galton's Normal Distribution.	346
22.8 From the graphs above, we can see that often, a short wait is followed by a long wait, in both directions. Also, the anticlockwise times are generally much closer to 10 minutes waiting time. It is also seen that around rush hour times (8:30, 15:00, 16:45), a pattern emerged where several buses in quick succession were followed by a long wait for the next bus to arrive. This could be because of the time taken for more passengers than usual to aboard and depart, and areas where traffic volume is greater at these times.	352
22.9 This graph shows the probability distribution function for the exponential function with the green line indicating a λ value of 0.1, the claimed λ . The red line indicates the value of λ estimated, 0.1105. From this graph, you can see the probability of getting a short waiting time is high - approximately 0.06, while the probability of a long waiting time is much much lower - approx imately 0.01. The Matlab code for this graph is shown in Appendix I.	353
22.10 This plot is the Empirical CDF plot(black), with a 95% confidence interval (red) and the actual CDF based on claimed $\lambda = 0.1$ (blue). The Matlab code for this graph is given in Appendix II. This graph shows the accuracy of the empirical distribution and hence the accuracy of the data we collected. There are some inconsistencies caused by the randomness of inter-arrival times but our empirical CDF is generally good as the actual CDF lies mostly within the interval lines. With more data points, our accuracy would greatly improve.	353

Chapter 1

Preliminaries

1.1 Elementary Set Theory

A **set** is a collection of distinct objects. We write a set by enclosing its elements with curly braces. For example, we denote a set of the two objects \circ and \bullet by:

$$\{\circ, \bullet\}.$$

Sometimes, we give names to sets. For instance, we might call the first example set A and write:

$$A = \{\circ, \bullet\}.$$

We do not care about the order of elements within a set, i.e. $A = \{\circ, \bullet\} = \{\bullet, \circ\}$. We do not allow a set to contain multiple copies of any of its elements unless the copies are distinguishable, say by labels. So, $B = \{\circ, \bullet, \bullet\}$ is not a set unless the two copies of \bullet in B are labelled or marked to make them distinct, e.g. $B = \{\circ, \tilde{\bullet}, \bullet'\}$. Names for sets that arise in a mathematical discourse are given upper-case letters (A, B, C, D, \dots). Special symbols are reserved for commonly encountered sets.

Here is the set \mathfrak{G} of twenty two Greek lower-case alphabets that we may encounter later:

$$\mathfrak{G} = \{ \alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \kappa, \lambda, \mu, \nu, \xi, \pi, \rho, \sigma, \tau, \upsilon, \varphi, \chi, \psi, \omega \}.$$

They are respectively named alpha, beta, gamma, delta, epsilon, zeta, eta, theta, kappa, lambda, mu, nu, xi, pi, rho, sigma, tau, upsilon, phi, chi, psi and omega. *LHS* and *RHS* are abbreviations for objects on the Left and Right Hand Sides, respectively, of some binary relation. By the notation:

$$LHS := RHS,$$

we mean that *LHS* is equal, by definition, to *RHS*.

The set which does not contain any element (the collection of nothing) is called the **empty set**:

$$\emptyset := \{ \}.$$

We say an element b belongs to a set B , or simply that b belongs to B or that b is an element of B , if b is one of the elements that make up the set B , and write:

$$b \in B.$$

When b does not belong to B , we write:

$$b \notin B.$$

For our example set $A = \{\circ, \bullet\}$, $\star \notin A$ but $\bullet \in A$.

We say that a set C is a **subset** of another set D and write:

$$C \subset D$$

if every element of C is also an element of D . By this definition, any set is a subset of itself.

We say that two sets C and D are **equal** (as sets) and write $C = D$ ‘if and only if’ (\iff) every element of C is also an element of D , and every element of D is also an element of C . This definition of set equality is notationally summarised as follows:

$$C = D \iff C \subset D, D \subset C .$$

When two sets C and D are not equal by the above definition, we say that C is **not equal** to D and write:

$$C \neq D .$$

The **union** of two sets C and D , written as $C \cup D$, is the set of elements that belong to C or D . We can formally express our definition of set union as:

$$C \cup D := \{x : x \in C \text{ or } x \in D\} .$$

When a colon (:) appears inside a set, it stands for ‘such that’. Thus, the above expression is read as ‘ C union D is equal by definition to the set of all elements x , such that x belongs to C or x belongs to D .’

Similarly, the **intersection** of two sets C and D , written as $C \cap D$, is the set of elements that belong to both C and D . Formally:

$$C \cap D := \{x : x \in C \text{ and } x \in D\} .$$

Venn diagrams are visual aids for set operations as in the diagrams below.

Figure 1.1: Union and intersection of sets shown by Venn diagrams

The set-difference or **difference** of two sets C and D , written as $C \setminus D$, is the set of elements in C that do not belong to D . Formally:

$$C \setminus D := \{x : x \in C \text{ and } x \notin D\} .$$

When a universal set, e.g. U is well-defined, the **complement** of a given set B denoted by B^c is the set of all elements of U that don’t belong to B , i.e.:

$$B^c := U \setminus B .$$

We say two sets C and D are **disjoint** if they have no elements in common, i.e. $C \cap D = \emptyset$.

By drawing Venn diagrams, let us check **De Morgan’s Laws**:

$$(A \cup B)^c = A^c \cap B^c \text{ and } (A \cap B)^c = A^c \cup B^c$$

Classwork 1 (Fruits and colours) Consider a set of fruits $F = \{\text{orange, banana, apple}\}$ and a set of colours $C = \{\text{red, green, blue, orange}\}$. Then,

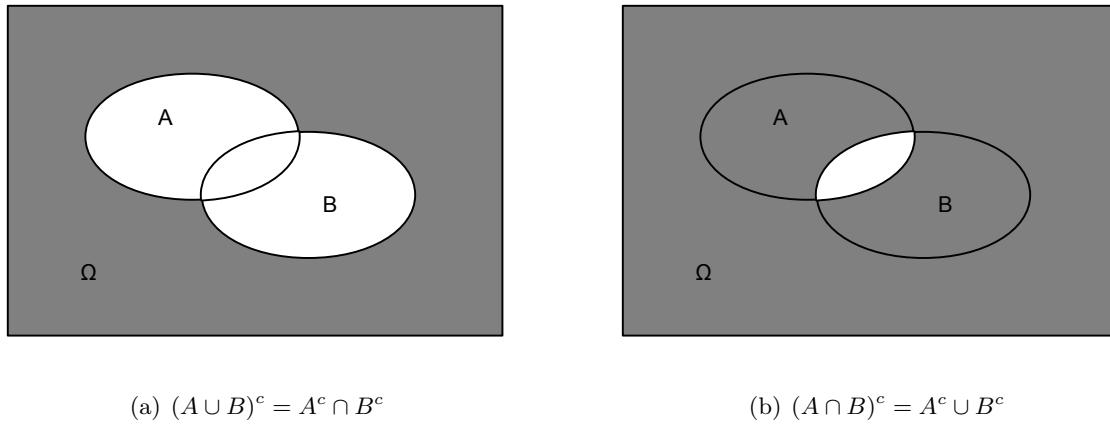


Figure 1.2: These Venn diagram illustrate De Morgan's Laws.

1. $F \cap C =$

2. $F \cup C =$

3. $F \setminus C =$

4. $C \setminus F =$

Classwork 2 (Subsets of a universal set) Suppose we are given a universal set U , and three of its subsets, A , B and C . Also suppose that $A \subset B \subset C$. Find the circumstances, if any, under which each of the following statements is true (T) and justify your answer:

- | | | | |
|---------------------------|--------------------------------|---------------------------|------------------------|
| (1) $C \subset B$ | T when $B = C$ | (2) $A \subset C$ | T by assumption |
| (3) $C \subset \emptyset$ | T when $A = B = C = \emptyset$ | (4) $\emptyset \subset A$ | T always |
| (5) $C \subset U$ | T by assumption | (6) $U \subset A$ | T when $A = B = C = U$ |

Exercises

Ex. 1.1 — Let Ω be the universal set of students, lecturers and tutors involved in a course. Now consider the following subsets:

- The set of 50 students, $S = \{S_1, S_2, S_3, \dots, S_{50}\}$.
- The set of 3 lecturers, $L = \{L_1, L_2, L_3\}$.
- The set of 4 tutors, $T = \{T_1, T_2, T_3, L_3\}$.

Note that one of the lecturers also tutors in the course. Find the following sets:

- | | |
|-----------------------|------------------|
| (a) $T \cap L$ | (f) $S \cap L$ |
| (b) $T \cap S$ | (g) $S^c \cap L$ |
| (c) $T \cup L$ | (h) T^c |
| (d) $T \cup L \cup S$ | (i) $T^c \cap L$ |
| (e) S^c | (j) $T^c \cap T$ |

Ex. 1.2 — Using Venn diagram, sketch and check the rule:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Ex. 1.3 — Using Venn diagram, sketch and check the rule:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

Ex. 1.4 — Using a Venn diagram, illustrate the idea that $A \subseteq B$ if and only if $A \cup B = B$.

SET SUMMARY

$\{a_1, a_2, \dots, a_n\}$	— a set containing the elements, a_1, a_2, \dots, a_n .
$a \in A$	— a is an element of the set A .
$A \subseteq B$	— the set A is a subset of B .
$A \cup B$	— “union”, meaning the set of all elements which are in A or B , or both.
$A \cap B$	— “intersection”, meaning the set of all elements in both A and B .
$\{\} \text{ or } \emptyset$	— empty set.
Ω	— universal set.
A^c	— the complement of A , meaning the set of all elements in Ω , the universal set, which are not in A .

1.2 Natural Numbers, Integers and Rational Numbers

We denote the number of elements in a set named B by:

$$\#B := \text{Number of elements in the set } B .$$

In fact, the Hindu-Arab numerals we have inherited are based on this intuition of the size of a collection. The elements of the set of **natural numbers**:

$$\mathbb{N} := \{1, 2, 3, 4, \dots\} , \text{ may be defined using } \# \text{ as follows:}$$

$$\begin{aligned} 1 &:= \#\{\star\} = \#\{\bullet\} = \#\{\alpha\} = \#\{\{\bullet\}\} = \#\{\{\bullet, \bullet'\}\} = \dots, \\ 2 &:= \#\{\star', \star\} = \#\{\bullet, \circ\} = \#\{\alpha, \omega\} = \#\{\{\circ\}, \{\alpha, \star, \bullet\}\} = \dots, \\ &\vdots \end{aligned}$$

For our example sets, $A = \{\circ, \bullet\}$ and the set of Greek alphabets \mathcal{G} , $\#A = 2$ and $\#\mathcal{G} = 22$. The number zero may be defined as the size of an empty set:

$$0 := \#\emptyset = \#\{\}$$

The set of **non-negative integers** is:

$$\mathbb{Z}_+ := \mathbb{N} \cup \{0\} = \{0, 1, 2, 3, \dots\} .$$

A **product set** is the **Cartesian product** (\times) of two or more possibly distinct sets:

$$A \times B := \{(a, b) : a \in A \text{ and } b \in B\}$$

For example, if $A = \{\circ, \bullet\}$ and $B = \{\star\}$, then $A \times B = \{(\circ, \star), (\bullet, \star)\}$. Elements of $A \times B$ are called **ordered pairs**.

The binary arithmetic operation of **addition** (+) between a pair of non-negative integers $c, d \in \mathbb{Z}_+$ can be defined via sizes of disjoint sets. Suppose, $c = \#C$, $d = \#D$ and $C \cap D = \emptyset$, then:

$$c + d = \#C + \#D := \#(C \cup D).$$

For example, if $A = \{\circ, \bullet\}$ and $B = \{\star\}$, then $A \cap B = \emptyset$ and $\#A + \#B = \#(A \cup B) \iff 2 + 1 = 3$.

The binary arithmetic operation of **multiplication** (\cdot) between a pair of non-negative integers $c, d \in \mathbb{Z}_+$ can be defined via sizes of product sets. Suppose, $c = \#C$, $d = \#D$, then:

$$c \cdot d = \#C \cdot \#D := \#(C \times D).$$

For example, if $A = \{\circ, \bullet\}$ and $B = \{\star\}$, then $\#A \cdot \#B = \#(A \times B) \iff 2 \cdot 1 = 2$.

More generally, a product set of A_1, A_2, \dots, A_m is:

$$A_1 \times A_2 \times \cdots \times A_m := \{(a_1, a_2, \dots, a_m) : a_1 \in A_1, a_2 \in A_2, \dots, a_m \in A_m\}$$

Elements of an m -product set are called **ordered m -tuples**. When we take the product of the same set we abbreviate as follows:

$$A^m := \underbrace{A \times A \times \cdots \times A}_{m \text{ times}} := \{(a_1, a_2, \dots, a_m) : a_1 \in A, a_2 \in A, \dots, a_m \in A\}$$

Classwork 3 (Cartesian product of sets) 1. Let $A = \{\circ, \bullet\}$. What are the elements of A^2 ? 2. Suppose $\#A = 2$ and $\#B = 3$. What is $\#(A \times B)$? 3. Suppose $\#A_1 = s_1, \#A_2 = s_2, \dots, \#A_m = s_m$. What is $\#(A_1 \times A_2 \times \cdots \times A_m)$?

Now, let's recall the definition of a function. A **function** is a “mapping” that associates each element in some set \mathbb{X} (the domain) to exactly one element in some set \mathbb{Y} (the range). Two different elements in \mathbb{X} can be mapped to or associated with the same element in \mathbb{Y} , and not every element in \mathbb{Y} needs to be mapped. Suppose $x \in \mathbb{X}$. Then we say $f(x) = y \in \mathbb{Y}$ is the **image** of x . To emphasise that f is a **function** from $\mathbb{X} \ni x$ to $\mathbb{Y} \ni y$, we write:

$$f(x) = y : \mathbb{X} \rightarrow \mathbb{Y}.$$

And for some $y \in \mathbb{Y}$, we call the set:

$$f^{[-1]}(y) := \{x \in \mathbb{X} : f(x) = y\} \subset \mathbb{X},$$

the **pre-image** or **inverse image** of y , and

$$f^{[-1]} := f^{[-1]}(y \in \mathbb{Y}) = X \subset \mathbb{X},$$

Figure 1.3: A function f (“father of”) from \mathbb{X} (a set of children) to \mathbb{Y} (their fathers) and its inverse (“children of”).

as the **inverse** of f .

We motivated the non-negative integers \mathbb{Z}_+ via the size of a set. With the notion of two directions (+ and -) and the magnitude of the current position from the origin zero (0) of a dynamic entity, we can motivate the set of **integers**:

$$\mathbb{Z} := \{\dots, -3, -2, -1, 0, +1, +2, +3, \dots\} .$$

The integers with a **minus** or **negative sign** (-) before them are called negative integers and those with a **plus** or **positive sign** (+) before them are called positive integers. Conventionally, + signs are dropped. Some examples of functions you may have encountered are **arithmetic operations** such as **addition** (+), **subtraction** (-), **multiplication** (\cdot) and **division** (/) of ordered pairs of integers. The reader is assumed to be familiar with such arithmetic operations with pairs of integers. Every integer is either positive, negative, or zero. In terms of this we define the notion of **order**. We say an integer a is **less than** an integer b and write $a < b$ if $b - a$ is positive. We say an integer a is **less than or equal to** an integer b and write $a \leq b$ if $b - a$ is positive or zero. Finally, we say that a is greater than b and write $a > b$ if $b < a$. Similarly, a is greater than equal to b , i.e. $a \geq b$, if $b \leq a$. The set of integers are **well-ordered**, i.e., for every integer a there is a next largest integer $a + 1$.

Classwork 4 (Addition over integers) Consider the set of integers $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$. Try to set up the arithmetic operation of addition as a function. The domain for addition is the Cartesian product of \mathbb{Z} :

$$\mathbb{Z}^2 := \mathbb{Z} \times \mathbb{Z} := \{(a, b) : a \in \mathbb{Z}, b \in \mathbb{Z}\}$$

What is its range ?

$$+ : \mathbb{Z} \times \mathbb{Z} \rightarrow$$

If the magnitude of the entity’s position is measured in units (e.g. meters) that can be rationally divided into q pieces with $q \in \mathbb{N}$, then we have the set of rational numbers:

$$\mathbb{Q} := \{p/q : p \in \mathbb{Z}, q \in \mathbb{Z} \setminus \{0\}\}$$

The expressions p/q and p'/q' denote the same rational number if and only if $p \cdot q' = p' \cdot q$. Every rational number has a unique irreducible expression p/q , where q is positive and as small as possible. For example, $1/2$, $2/4$, $3/6$, and $1001/2002$ are different expressions for the same rational number whose irreducible unique expression is $1/2$.

Figure 1.4: A pictorial depiction of addition and its inverse. The domain is plotted in orthogonal Cartesian coordinates.

Addition and multiplication are defined for rational numbers by:

$$\frac{p}{q} + \frac{p'}{q'} = \frac{p \cdot q' + p' \cdot q}{q \cdot q'} \quad \text{and} \quad \frac{p}{q} \cdot \frac{p'}{q'} = \frac{p \cdot p'}{q \cdot q'} .$$

The rational numbers form a **field** under the operations of addition and multiplication defined above in terms of addition and multiplication over integers. This means that the following properties are satisfied:

1. Addition and multiplication are each **commutative**

$$a + b = b + a, \quad a \cdot b = b \cdot a ,$$

and associative

$$a + (b + c) = (a + b) + c, \quad a \cdot (b \cdot c) = (a \cdot b) \cdot c .$$

2. Multiplication **distributes** over addition

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c) .$$

3. 0 is the **additive identity** and 1 is the multiplicative identity

$$0 + a = a \quad \text{and} \quad 1 \cdot a = a .$$

4. Every rational number a has a negative, $a + (-a) = 0$ and every non-zero rational number a has a reciprocal, $a \cdot 1/a = 1$.

The field axioms imply the usual laws of arithmetic and allow subtraction and division to be defined in terms of addition and multiplication as follows:

$$\frac{p}{q} - \frac{p'}{q'} := \frac{p}{q} + \frac{-p'}{q'} \quad \text{and} \quad \frac{p}{q} / \frac{p'}{q'} := \frac{p}{q} \cdot \frac{q'}{p'}, \quad \text{provided } p' \neq 0 .$$

We will see later that the theory of finite fields is necessary for the study of pseudo-random number generators (PRNGs) and PRNGs are the heart-beat of randomness and statistics with computers.

1.3 Real Numbers

Unlike rational numbers which are expressible in their reduced forms by p/q , it is fairly tricky to define or express real numbers. It is possible to define real numbers formally and constructively via equivalence classes of Cauchy sequence of rational numbers. For this all we need are notions of (1) infinity, (2) sequence of rational numbers and (3) distance between any two rational numbers in an infinite sequence of them. These are topics usually covered in an introductory course in real analysis and are necessary for a firm foundation in computational statistics. Instead of a formal constructive definition of real numbers, we give a more concrete one via decimal expansions. See Donald E. Knuth's treatment [*Art of Computer Programming, Vol. I, Fundamental Algorithms*, 3rd Ed., 1997, pp. 21-25] for a fuller story. A **real number** is a numerical quantity x that has a decimal expansion:

$$x = n + 0.d_1d_2d_3 \dots , \text{ where, each } d_i \in \{0, 1, \dots, 9\}, n \in \mathbb{Z} ,$$

and the sequence $0.d_1d_2d_3 \dots$ does not terminate with infinitely many consecutive 9s. By the above decimal representation, the following arbitrarily accurate enclosure of the real number x by rational numbers is implied:

$$n + \frac{d_1}{10} + \frac{d_2}{100} + \dots + \frac{d_k}{10^k} =: \underline{x}_k \leq x < \bar{x}_k := n + \frac{d_1}{10} + \frac{d_2}{100} + \dots + \frac{d_k}{10^k} + \frac{1}{10^{k+1}}$$

for every $k \in \mathbb{N}$. Thus, rational arithmetic $(+, -, \cdot, /)$ can be extended with arbitrary precision to any ordered pair of real numbers x and y by operations on their rational enclosures \underline{x}, \bar{x} and \underline{y}, \bar{y} .

Some examples of real numbers that are not rational (**irrational numbers**) are:

$$\sqrt{2} = 1.41421356237309 \dots \text{the side length of a square with area of 2 units}$$

$$\pi = 3.14159265358979 \dots \text{the ratio of the circumference to diameter of a circle}$$

$$e = 2.71828182845904 \dots \text{Euler's constant}$$

We can think of π as being enclosed by the following pairs of rational numbers:

$$\begin{aligned} 3 + \frac{1}{10} &=: \underline{\pi}_1 \leq \pi < \bar{\pi}_1 := 3 + \frac{1}{10} + \frac{1}{10^1} \\ 3 + \frac{1}{10} + \frac{4}{100} &=: \underline{\pi}_2 \leq \pi < \bar{\pi}_2 := 3 + \frac{1}{10} + \frac{4}{100} + \frac{1}{100} \\ 3 + \frac{1}{10} + \frac{4}{100} + \frac{1}{10^3} &=: \underline{\pi}_3 \leq \pi < \bar{\pi}_3 := 3 + \frac{1}{10} + \frac{4}{100} + \frac{1}{10^3} + \frac{1}{10^3} \\ &\vdots \\ 3.14159265358979 &=: \underline{\pi}_{14} \leq \pi < \bar{\pi}_{14} := 3.14159265358979 + \frac{1}{10^{14}} \\ &\vdots \end{aligned}$$

Think of the real number system as the continuum of points that make up a line, as shown in Figure 1.5.

Let y and z be two real numbers such that $y \leq z$. Then, the **closed interval** $[y, z]$ is the set of real numbers x such that $y \leq x \leq z$:

$$[y, z] := \{x : y \leq x \leq z\} .$$

Figure 1.5: A depiction of the real line segment $[-10, 10]$.

The **half-open interval** $(y, z]$ or $[y, z)$ and the **open interval** (y, z) are defined analogously:

$$\begin{aligned}(y, z] &:= \{x : y < x \leq z\} , \\ [y, z) &:= \{x : y \leq x < z\} , \\ (y, z) &:= \{x : y < x < z\} .\end{aligned}$$

We also allow y to be **minus infinity** (denoted $-\infty$) or z to be **infinity** (denoted ∞) at an open endpoint of an interval, meaning that there is no lower or upper bound. With this allowance we get the set of **real numbers** $\mathbb{R} := (-\infty, \infty)$, the **non-negative real numbers** $\mathbb{R}_+ := [0, \infty)$ and the **positive real numbers** $\mathbb{R}_{>0}(0, \infty)$ as follows:

$$\begin{aligned}\mathbb{R} &:= (-\infty, \infty) = \{x : -\infty < x < \infty\} , \\ \mathbb{R}_+ &:= [0, \infty) = \{x : 0 \leq x < \infty\} , \\ \mathbb{R}_{>0} &:= (0, \infty) = \{x : 0 < x < \infty\} .\end{aligned}$$

For a positive real number $b \in \mathbb{R}_{>0}$ and an integer $n \in \mathbb{Z}$, the n -th **power** or **exponent** of b is:

$$b^0 = 1, \quad b^n = b^{n-1} \cdot b \quad \text{if } n > 0, \quad b^n = b^{n+1}/b \quad \text{if } n < 0 .$$

The following **laws of exponents** hold by mathematical induction when $m, n \in \mathbb{Z}$:

$$b^{m+n} = b^m \cdot b^n, \quad (b^m)^n = b^{m \cdot n} .$$

If $y \in \mathbb{R}$ and $m \in \mathbb{N}$, the unique positive real number $z \in \mathbb{R}_{>0}$ such that $z^m = y$ is called the m -th **root of y** and denoted by $\sqrt[m]{y}$, i.e.,

$$z^m = y \implies z = \sqrt[m]{y} .$$

For a rational number $r = p/q \in \mathbb{Q}$, we define the r -th power of $b \in \mathbb{R}$ as follows:

$$b^r = b^{p/q} := \sqrt[q]{b^p}$$

The laws of exponents hold for this definition and different expressions for the same rational number $r = ap/aq$ yield the same power, i.e., $b^{p/q} = b^{ap/aq}$. Recall that a real number $x = n + 0.d_1d_2d_3\dots \in \mathbb{R}$ can be arbitrarily precisely enclosed by the rational numbers $\underline{x}_k := n + \frac{d_1}{10} + \frac{d_2}{100} + \dots + \frac{d_k}{10^k}$ and $\bar{x}_k := n + \frac{d_1}{10} + \frac{d_2}{100} + \dots + \frac{d_k}{10^k} + \frac{1}{10^k}$ by increasing k . Suppose first that $b > 1$. Then, using rational powers, we can enclose b^x ,

$$b^{n+\frac{d_1}{10}+\frac{d_2}{100}+\dots+\frac{d_k}{10^k}} =: b^{\underline{x}_k} \leq b^x < b^{\bar{x}_k} =: b^{n+\frac{d_1}{10}+\frac{d_2}{100}+\dots+\frac{d_k}{10^k}+\frac{1}{10^k}} ,$$

within an interval of width $b^{n+\frac{d_1}{10}+\frac{d_2}{100}+\dots+\frac{d_k}{10^k}} \left(b^{\frac{1}{10^k}} - 1 \right) < b^{n+1}(b-1)/10^k$. By taking a large enough k we can evaluate b^x to any accuracy. Finally, when $b < 1$ we define $b^x := (1/b)^{-x}$ and when $b = 0$, $b^x := 1$.

Suppose $y \in \mathbb{R}_{>0}$ and $b \in \mathbb{R} \setminus \{1\}$ then the real number x such that $y = b^x$ is called the **logarithm of y to the base b** and we write this as:

$$y = b^x \iff x = \log_b y$$

The definition implies:

$$x = \log_b(b^x) = b^{\log_b x},$$

and the laws of exponents imply:

$$\begin{aligned}\log_b(xy) &= \log_b x + \log_b y, \quad \text{if } x > 0, y > 0 \text{ and} \\ \log_b(c^y) &= y \log_b c, \quad \text{if } c > 0.\end{aligned}$$

The **common logarithm** is $\log_{10}(y)$, the **binary logarithm** is $\log_2(y)$ and the **natural logarithm** is $\log_e(y)$, where e is the Euler's constant. Since we will mostly work with $\log_e(y)$ we use $\log(y)$ to mean $\log_e(y)$. You are assumed to be familiar with trigonometric functions ($\sin(x)$, $\cos(x)$, $\tan(x)$, ...). We sometimes denote the special power function e^y by $\exp(y)$.

Familiar extremal elements of a set of real numbers, say A , are the following:

$$\boxed{\max A := \text{greatest element in } A}$$

For example, $\max\{1, 4, -9, 345\} = 345$, $\max[-93.8889, 1002.786] = 1002.786$.

$$\boxed{\min A := \text{least element in } A}$$

For example, $\min\{1, 4, -9, 345\} = -9$, $\min[-93.8889, 1002.786] = -93.8889$. We need a slightly more sophisticated notion for the extremal elements of a set A that may not belong to A . We say that a real number x is a **lower bound** for a non-empty set of real numbers A , provided $x \leq a$ for every $a \in A$. We say that the set A is **bounded below** if it has at least one lower bound. A lower bound is the **greatest lower bound** if it is at least as large as any other lower bound. The greatest lower bound of a set of real numbers A is called the **infimum** of A and is denoted by:

$$\boxed{\inf A := \text{greatest lower bound of } A}$$

For example, $\inf(0, 1) = 0$ and $\inf\{10.333 \cup [-99, 1001.33]\} = -99$. We similarly define the **least upper bound** of a non-empty set of real numbers A to be the **supremum** of A and denote it as:

$$\boxed{\sup A := \text{least upper bound of } A}$$

For example, $\sup(0, 1) = 1$ and $\sup\{10.333 \cup [-99, 1001.33]\} = 1001.33$. By convention, we define $\inf \emptyset := \infty$, $\sup \emptyset := -\infty$. Finally, if a set A is not bounded below then $\inf A := -\infty$ and if a set A is not bounded above then $\sup A := \infty$.

Symbol	Meaning
$A = \{\star, \circ, \bullet\}$	A is a set containing the elements \star, \circ and \bullet
$\circ \in A$	\circ belongs to A or \circ is an element of A
$A \ni \circ$	\circ belongs to A or \circ is an element of A
$\circ \notin A$	\circ does not belong to A
$\#A$	Size of the set A , for e.g. $\#\{\star, \circ, \bullet, \odot\} = 4$
\mathbb{N}	The set of natural numbers $\{1, 2, 3, \dots\}$
\mathbb{Z}	The set of integers $\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
\mathbb{D}_+	The set of non-negative integers $\{0, 1, 2, 3, \dots\}$
\emptyset	Empty set or the collection of nothing or $\{\}$
$A \subset B$	A is a subset of B or A is contained by B , e.g. $A = \{\circ\}, B = \{\bullet\}$
$A \supset B$	A is a superset of B or A contains B e.g. $A = \{\circ, \star, \bullet\}, B = \{\circ, \bullet\}$
$A = B$	A equals B , i.e. $A \subset B$ and $B \subset A$
$Q \implies R$	Statement Q implies statement R or If Q then R
$Q \iff R$	$Q \implies R$ and $R \implies Q$
$\{x : x \text{ satisfies property } R\}$	The set of all x such that x satisfies property R
$A \cup B$	A union B , i.e. $\{x : x \in A \text{ or } x \in B\}$
$A \cap B$	A intersection B , i.e. $\{x : x \in A \text{ and } x \in B\}$
$A \setminus B$	A minus B , i.e. $\{x : x \in A \text{ and } x \notin B\}$
$A := B$	A is equal to B by definition
$A =: B$	B is equal to A by definition
A^c	A complement, i.e. $\{x : x \in U, \text{ the universal set, but } x \notin A\}$
$A_1 \times A_2 \times \dots \times A_m$	The m -product set $\{(a_1, a_2, \dots, a_m) : a_1 \in A_1, a_2 \in A_2, \dots, a_m \in A_m\}$
A^m	The m -product set $\{(a_1, a_2, \dots, a_m) : a_1 \in A, a_2 \in A, \dots, a_m \in A\}$
$f := f(x) = y : \mathbb{X} \rightarrow \mathbb{Y}$	A function f from domain \mathbb{X} to range \mathbb{Y}
$f^{[-1]}(y)$	Inverse image of y
$f^{[-1]} := f^{[-1]}(y \in \mathbb{Y}) = X \subset \mathbb{X}$	Inverse of f
$a < b$ or $a \leq b$	a is less than b or a is less than or equal to b
$a > b$ or $a \geq b$	a is greater than b or a is greater than or equal to b
\mathbb{Q}	Rational numbers
(x, y)	the open interval (x, y) , i.e. $\{r : x < r < y\}$
$[x, y]$	the closed interval (x, y) , i.e. $\{r : x \leq r \leq y\}$
$(x, y]$	the half-open interval $(x, y]$, i.e. $\{r : x < r \leq y\}$
$[x, y)$	the half-open interval $[x, y)$, i.e. $\{r : x \leq r < y\}$
$\mathbb{R} := (-\infty, \infty)$	Real numbers, i.e. $\{r : -\infty < r < \infty\}$
$\mathbb{R}_+ := [0, \infty)$	Real numbers, i.e. $\{r : 0 \leq r < \infty\}$
$\mathbb{R}_{>0} := (0, \infty)$	Real numbers, i.e. $\{r : 0 < r < \infty\}$

Table 1.1: Symbol Table: Sets and Numbers

1.4 Introduction to MATLAB

We use MATLAB to perform computations and visualisations. MATLAB is a numerical computing environment and programming language that is optimised for vector and matrix processing. STAT 218/313 students will have access to Maths & Stats Department's computers that are licensed to run MATLAB . You can remotely connect to these machines from home by following instructions at <http://www.math.canterbury.ac.nz/php/resources/comdocs/remote>.

Labwork 5 (Basics of MATLAB) Let us familiarize ourselves with MATLAB in this session. First, you need to launch MATLAB from your terminal. Since this is system dependent, ask your tutor for help. The command window within the MATLAB window is where you need to type commands. Here is a minimal set of commands you need to familiarize yourself with in this session.

1. Type the following command to add 2 numbers in the command window right after the command prompt `>>` .

```
>> 13+24
```

Upon hitting **Enter** or **Return** on your keyboard, you should see:

```
ans =
37
```

The summand 37 of 13 and 24 is stored in the default variable called `ans` which is short for answer.

2. We can write **comments** in MATLAB following the % character. All the characters in a given line that follow the percent character % are ignored by MATLAB . It is very helpful to comment what is being done in the code. You won't get full credit without sufficient comments in your coding assignments. For example we could have added the comment to the previous addition. To save space in these notes, we suppress the blank lines and excessive line breaks present in MATLAB 's command window.

```
>> 13+24 % adding 13 to 24 using the binary arithmetic operator +
ans =      37
```

3. You can **create or reopen a diary file** in MATLAB to record your work. Everything you typed or input and the corresponding output in the command window will be recorded in the diary file. You can create or reopen a diary file by typing `diary filename.txt` in the command window. When you have finished recording, simply type `diary off` in the command window **to turn off the diary file**. The diary file with .txt extension is simply a text-file. It can be edited in different editors after the diary is turned off in MATLAB . You need to type `diary LabWeek1.txt` to start recording your work for electronic submission if needed.

```
>> diary blah.txt % start a diary file named blah.txt
>> 3+56
ans =      59
>> diary off % turn off the current diary file blah.txt
```

```
>> type blah.txt % this allows you to see the contents of blah.txt
3+56
ans =      59
diary off
>> diary blah.txt % reopen the existing diary file blah.txt
>> 45-54
ans =      -9
>> diary off % turn off the current diary file blah.txt again
>> type blah.txt % see its contents
3+56
ans =      59
diary off
45-54
ans =      -9
diary off
```

4. Let's learn to store values in variables of our choice. Type the following at the command prompt :

```
>> VariableCalledX = 12
```

Upon hitting enter you should see that the number 12 has been assigned to the variable named **VariableCalledX** :

```
VariableCalledX =      12
```

5. MATLAB stores default value for some variables, such as **pi** (π), **i** and **j** (complex numbers).

```
>> pi
ans =      3.1416
>> i
ans =      0 + 1.0000i
>> j
ans =      0 + 1.0000i
```

All predefined symbols (variables, constants, function names, operators, etc) in MATLAB are written in lower-case. Therefore, it is a good practice to name the variable you define using upper and mixed case letters in order to prevent an unintended overwrite of some predefined MATLAB symbol.

6. We could have stored the sum of 13 and 24 in the variable **X**, by entering:

```
>> X = 13 + 24
X =      37
```

7. Similarly, you can store the outcome of multiplication (via operation *****), subtraction (via operation **-**), division (via **/**) and exponentiation (via **^**)of any two numbers of your choice in a variable name of your choice. Evaluate the following expressions in MATLAB :

$$\begin{aligned} p &= 45.89 * 1.00009 \\ m &= 5376.0 - 6.00 \end{aligned}$$

$$\begin{aligned} d &= 89.0 / 23.3454 \\ p &= 2^{0.5} \end{aligned}$$

8. You may compose the elementary operations to obtain rational expressions by using parenthesis to specify the order of the operations. To obtain $\sqrt{2}$, you can type the following into MATLAB 's command window.

```
>> 2^(1/2)
ans =      1.4142
```

The omission of parenthesis about $1/2$ means something else and you get the following output:

```
>> 2^1/2
ans =      1
```

MATLAB first takes the 1st power of 2 and then divides it by 2 using its default precedence rules for binary operators in the absence of parenthesis. The order of operations or default precedence rule for arithmetic operations is 1. brackets or parentheses; 2. exponents (powers and roots); 3. division and multiplication; 4. addition and subtraction. The mnemonic **bedmas** can be handy. When in doubt, use parenthesis to force the intended order of operations.

9. When you try to divide by 0, MATLAB returns **Inf** for infinity.

```
>> 10/0
ans =    Inf
```

10. We can clear the value we have assigned to a particular variable and reuse it. We demonstrate it by the following commands and their output:

```
>> X
X =      37
>> clear X
>> X
??? Undefined function or variable 'X'.
```

Entering **X** after clearing it gives the above self-explanatory error message preceded by **???**.

11. We can suppress the output on the screen by ending the command with a semi-colon. Take a look at the simple command that sets **X** to $\sin(3.145678)$ with and without the ‘;**;**’ at the end:

```
>> X = sin(3.145678)
X =    -0.0041
>> X = sin(3.145678);
```

12. If you do not understand a MATLAB function or command then type **help** or **doc** followed by the function or command. For example:

```
>> help sin
SIN    Sine of argument in radians.
SIN(X) is the sine of the elements of X.
See also asin, sind.
Overloaded methods:
darray/sin
Reference page in Help browser
    doc sin
>> doc sin
```

It is a good idea to use the help files before you ask your tutor.

13. Set the variable `x` to equal 17.13 and evaluate $\cos(x)$, $\log(x)$, $\exp(x)$, $\arccos(x)$, $\text{abs}(x)$, $\text{sign}(x)$ using the MATLAB commands `cos`, `log`, `exp`, `acos`, `abs`, `sign`, respectively. Read the help files to understand what each function does.
14. When we work with real numbers (floating-point numbers) or really large numbers, we might want the output to be displayed in concise notation. This can be controlled in MATLAB using the `format` command with the `short` or `long` options with/without `e` for scientific notation. `format compact` is used for getting compacted output and `format` returns the default format. For example:

```
>> format compact
>> Y=15;
>> Y = Y + acos(-1)
Y = 18.1416
>> format short
>> Y
Y = 18.1416
>> format short e
>> Y
Y = 1.8142e+001
>> format long
>> Y
Y = 18.141592653589793
>> format long e
>> Y
Y = 1.814159265358979e+001
>> format
>> Y
Y = 18.1416
```

15. Finally, to quit from MATLAB just type `quit` or `exit` at the prompt.

```
>> quit
```

16. An **M-file** is a special text file with a `.m` extension that contains a set of code or instructions in MATLAB . In this course we will be using two types of M-files: **script** and **function** files. A script file is simply a list of commands that we want executed and saves us from retyping code modules we are pleased with. A function file allows us to write specific tasks as functions with input and output. These functions can be called from other script files, function files or command window. We will see such examples shortly.

By now, you are expected to be familiar with arithmetic operations, simple function evaluations, format control, starting and stopping a diary file and launching and quitting MATLAB .

1.5 Permutations, Factorials and Combinations

Definition 1 (Permutations and Factorials) A **permutation** of n objects is an arrangement of n distinct objects in a row. For example, there are 2 permutations of the two objects $\{1, 2\}$:

$$12, \quad 21,$$

and 6 permutations of the three objects $\{a, b, c\}$:

$$abc, \quad acb, \quad bac, \quad bca, \quad cab, \quad cba.$$

Let the number of ways to choose k objects out of n and to arrange them in a row be denoted by $p_{n,k}$. For example, we can choose two ($k = 2$) objects out of three ($n = 3$) objects, $\{a, b, c\}$, and arrange them in a row in six ways ($p_{3,2}$):

$$ab, \quad ac, \quad ba, \quad bc, \quad ca, \quad cb.$$

Given n objects, there are n ways to choose the left-most object, and once this choice has been made there are $n - 1$ ways to select a different object to place next to the left-most one. Thus, there are $n(n - 1)$ possible choices for the first two positions. Similarly, when $n > 2$, there are $n - 2$ choices for the third object that is distinct from the first two. Thus, there are $n(n - 1)(n - 2)$ possible ways to choose three distinct objects from a set of n objects and arrange them in a row. In general,

$$p_{n,k} = n(n - 1)(n - 2) \dots (n - k + 1)$$

and the total number of permutations called ‘ n factorial’ and denoted by $n!$ is

$$n! := p_{n,n} = n(n - 1)(n - 2) \dots (n - n + 1) = n(n - 1)(n - 2) \dots (3)(2)(1) =: \prod_{i=1}^n i.$$

Some factorials to bear in mind

$$0! := 1 \quad 1! = 1, \quad 2! = 2, \quad 3! = 6, \quad 4! = 24, \quad 5! = 120 \quad 10! = 3,628,800.$$

When n is large we can get a good idea of $n!$ without laboriously carrying out the $n - 1$ multiplications via Stirling’s approximation (*Methodus Differentialis* (1730), p. 137) :

$$n! \cong \sqrt{2\pi n} \left(\frac{n}{e}\right)^n.$$

Definition 2 (Combinations) The combinations of n objects taken k at a time are the possible choices of k different elements from a collection of n objects, disregarding order. They are called the k -combinations of the collection. The combinations of the three objects $\{a, b, c\}$ taken two at a time, called the 2-combinations of $\{a, b, c\}$, are

$$ab, \quad ac, \quad bc,$$

and the combinations of the five objects $\{1, 2, 3, 4, 5\}$ taken three at a time, called the 3-combinations of $\{1, 2, 3, 4, 5\}$ are

$$123, \quad 124, \quad 125, \quad 134, \quad 135, \quad 145, \quad 234, \quad 235, \quad 245, \quad 345.$$

The total number of k -combination of n objects, called a **binomial coefficient**, denoted $\binom{n}{k}$ and read “ n choose k ,” can be obtained from $p_{n,k} = n(n - 1)(n - 2) \dots (n - k + 1)$ and $k! := p_{k,k}$. Recall that $p_{n,k}$ is the number of ways to choose the first k objects from the set of n objects and arrange them in a row with regard to order. Since we want to disregard order and each k -combination appears exactly $p_{k,k}$ or $k!$ times among the $p_{n,k}$ many permutations, we perform a division:

$$\binom{n}{k} := \frac{p_{n,k}}{p_{k,k}} = \frac{n(n - 1)(n - 2) \dots (n - k + 1)}{k(k - 1)(k - 2) \dots 2 \ 1}.$$

Binomial coefficients are often called “Pascal’s Triangle” and attributed to Blaise Pascal’s *Traité du Triangle Arithmétique* from 1653, but they have many “fathers”. There are earlier treatises of the binomial coefficients including Szu-yüan Yü-chien (“The Precious Mirror of the Four Elements”) by the Chinese mathematician Chu Shih-Chieh in 1303, and in an ancient Hindu classic, *Pingala’s Chandasāstra*, due to Halāyudha (10-th century AD).

1.6 Array, Sequence, Limit, ...

In this section we will study a basic data structure in MATLAB called an **array** of numbers. Arrays are finite sequences and they can be processed easily in MATLAB . The notion of infinite sequences lead to **limits**, one of the most fundamental concepts in mathematics.

For any natural number n , we write

$$\langle x_{1:n} \rangle := x_1, x_2, \dots, x_{n-1}, x_n$$

to represent the **finite sequence** of real numbers $x_1, x_2, \dots, x_{n-1}, x_n$. For two integers m and n such that $m \leq n$, we write

$$\langle x_{m:n} \rangle := x_m, x_{m+1}, \dots, x_{n-1}, x_n$$

to represent the **finite sequence** of real numbers $x_m, x_{m+1}, \dots, x_{n-1}, x_n$. In mathematical analysis, finite sequences and their countably infinite counterparts play a fundamental role in limiting processes. Given an integer m , we denote an **infinite sequence** or simply a sequence as:

$$\langle x_{m:\infty} \rangle := x_m, x_{m+1}, x_{m+2}, x_{m+3}, \dots$$

Given index set \mathcal{I} which may be finite or infinite in size, a sequence can either be seen as a set of ordered pairs:

$$\{(i, x_i) : i \in \mathcal{I}\},$$

or as a function that maps the index set to the set of real numbers:

$$x(i) = x_i : \mathcal{I} \rightarrow \{x_i : i \in \mathcal{I}\},$$

The finite sequence $\langle x_{m:n} \rangle$ has $\mathcal{I} = \{m, m+1, m+2, m+3, \dots, n\}$ as its index set while an infinite sequence $\langle x_{m:\infty} \rangle$ has $\mathcal{I} = \{m, m+1, m+2, m+3, \dots\}$ as its index set. A **sub-sequence** $\langle x_{j:k} \rangle$ of a finite sequence $\langle x_{m:n} \rangle$ or an infinite sequence $\langle x_{m:\infty} \rangle$ is:

$$\langle x_{j:k} \rangle = x_j, x_{j+1}, \dots, x_{k-1}, x_k \quad \text{where, } \quad m \leq j \leq k \leq n < \infty.$$

A rectangular arrangement of $m \cdot n$ real numbers in m rows and n columns is called an $m \times n$ **matrix**. The ' $m \times n$ ' represents the **size** of the matrix. We use bold upper-case letters to denote matrices, for e.g:

$$\mathbf{B}X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n-1} & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n-1} & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m-1,1} & x_{m-1,2} & \dots & x_{m-1,n-1} & x_{m-1,n} \\ x_{m,1} & x_{m,2} & \dots & x_{m,n-1} & x_{m,n} \end{bmatrix}$$

Matrices with only one row or only one column are called **vectors**. An $1 \times n$ matrix is called a **row vector** since there is only one row and an $m \times 1$ matrix is called a **column vector** since there is only one column. We use bold-face lowercase letters to denote row and column vectors.

$$\text{A row vector } \mathbf{B}x = [x_1 \ x_2 \ \dots \ x_n] = (x_1, x_2, \dots, x_n)$$

$$\text{and a column vector } \mathbf{B}y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{m-1} \\ y_m \end{bmatrix} = [y_1 \ y_2 \ \dots \ y_m]' = (y_1, y_2, \dots, y_m)'.$$

The superscripting by $'$ is the transpose operation and simply means that the rows and columns are exchanged. Thus the transpose of the matrix BX is:

$$BX' = \begin{bmatrix} x_{1,1} & x_{2,1} & \dots & x_{m-1,1} & x_{m,1} \\ x_{1,2} & x_{2,2} & \dots & x_{m-1,2} & x_{m,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{1,n-1} & x_{2,n-1} & \dots & x_{m-1,n-1} & x_{m,n-1} \\ x_{1,n} & x_{2,n} & \dots & x_{m-1,n} & x_{m,n} \end{bmatrix}$$

In linear algebra and calculus, it is natural to think of vectors and matrices as points (ordered m -tuples) and ordered collection of points in Cartesian co-ordinates. We assume that the reader has heard of operations with matrices and vectors such as matrix multiplication, determinants, transposes, etc. Such concepts will be introduced as they are needed in the sequel.

Finite sequences, vectors and matrices can be represented in a computer by an elementary data structure called an **array**.

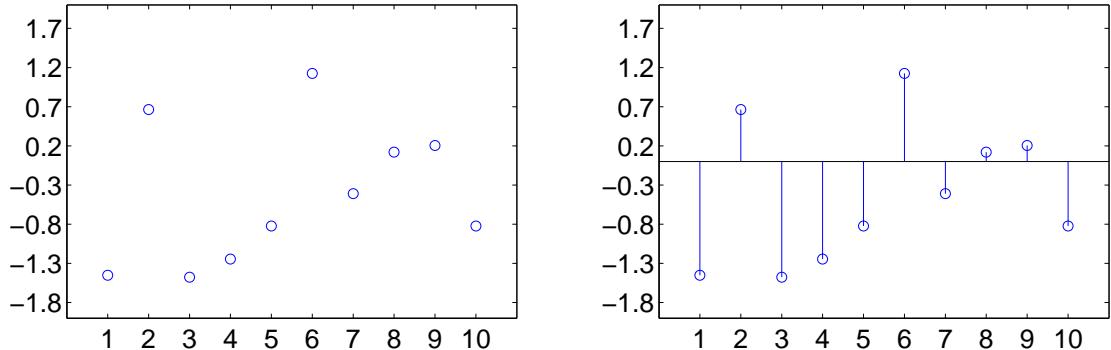
Labwork 6 (Sequences as arrays) Let us learn to represent, visualise and operate finite sequences as MATLAB arrays. Try out the commands and read the comments for clarification.

```
>> a = [17] % Declare the sequence of one element 17 in array a
a =
17
>> % Declare the sequence of 10 numbers in array b
>> b=[-1.4508 0.6636 -1.4768 -1.2455 -0.8235 1.1254 -0.4093 0.1199 0.2043 -0.8236]
b =
-1.4508    0.6636   -1.4768   -1.2455   -0.8235    1.1254   -0.4093    0.1199    0.2043   -0.8236
>> c = [1 2 3] % Declare the sequence of 3 consecutive numbers 1,2,3
c =
1     2     3
>> % linspace(x1, x2, n) generates n points linearly spaced between x1 and x2
>> r = linspace(1, 3, 3) % Declare sequence r = c using linspace
r =
1     2     3
>> s1 = 1:10 % declare an array s1 starting at 1, ending by 10, in increments of 1
s =
1     2     3     4     5     6     7     8     9     10
>> s2 = 1:2:10 % declare an array s2 starting at 1, ending by 10, in increments of 2
s =
1     3     5     7     9
>> s2(3) % obtain the third element of the finite sequence s2
ans =
5
>> s2(2:4) % obtain the subsequence from second to fourth elements of the finite sequence s2
ans =
3     5     7
```

We may visualise (as per Figure 1.6) the finite sequences $\langle b_{1:n} \rangle$ stored in the array **b** as the set of ordered pairs $\{(1, b_1), (2, b_2), \dots, (10, b_n)\}$ representing the function $b(i) = b_i : \{1, 2, \dots, n\} \rightarrow \{b_1, b_2, \dots, b_n\}$ via **point plot** and **stem plot** using Matlab's **plot** and **stem** commands, respectively.

```
>> display(b) % display the array b in memory
b =
-1.4508    0.6636   -1.4768   -1.2455   -0.8235    1.1254   -0.4093    0.1199    0.2043   -0.8236
>> plot(b,'o') % point plot of ordered pairs (1,b(1)), (2,b(2)), ..., (10,b(10))
>> stem(b) % stem plot of ordered pairs (1,b(1)), (2,b(2)), ..., (10,b(10))
>> plot(b,'-o') % point plot of ordered pairs (1,b(1)), (2,b(2)), ..., (10,b(10)) connected by lines
```

Labwork 7 (Vectors and matrices as arrays) Let us learn to represent, visualise and operate vectors as MATLAB arrays. Syntactically, a vector is stored in an array exactly in the same way we stored a finite sequence. However, mathematically, we think of a vector as an ordered m -tuple that can be visualised as a point in Cartesian co-ordinates. Try out the commands and read the comments for clarification.

Figure 1.6: Point plot and stem plot of the finite sequence $\langle b_{1:10} \rangle$ declared as an array.

```

>> a = [1 2]          % an 1 X 2 row vector
>> z = [1 2 3]        % Declare an 1 X 3 row vector z with three numbers
z =
    1      2      3
>> % linspace(x1, x2, n) generates n points linearly spaced between x1 and x2
>> r = linspace(1, 3, 3)           % Declare an 1 X 3 row vector r = z using linspace
r =
    1      2      3
>> c = [1; 2; 3]                % Declare a 3 X 1 column vector c with three numbers. Semicolons delineate columns
c =
    1
    2
    3
>> rT = r'                    % The column vector (1,2,3)' by taking the transpose of r via r'
rT =
    1
    2
    3
>> y = [1 1 1]                  % y is a sequence or row vector of 3 1's
y =
    1      1      1
>> ones(1,10)                  % ones(m,n) is an m X n matrix of ones. Useful when m or n is large.
ans =
    1      1      1      1      1      1      1      1      1      1

```

We can use two dimensional arrays to represent matrices. Some useful built-in commands to generate standard matrices are:

```

>> Z=zeros(2,10) % the 2 X 10 matrix of zeros
Z =
    0      0      0      0      0      0      0      0      0      0
    0      0      0      0      0      0      0      0      0      0
>> O=ones(4,5) % the 4 X 5 matrix of ones
O =
    1      1      1      1      1
    1      1      1      1      1
    1      1      1      1      1
    1      1      1      1      1
>> E=eye(4) % the 4 X 4 identity matrix
E =
    1      0      0      0
    0      1      0      0
    0      0      1      0
    0      0      0      1

```

We can also perform operations with arrays representing vectors, finite sequences, or matrices.

```

>> y % the array y is
y =      1      1      1
>> z % the array z is
z =      1      2      3
>> x = y + z
x =      2      3      4
                                % x is the sum of vectors y and z (with same size 1 X 3)
>> y = y * 2
y =      2      2      2
                                % y is updated to 2 * y (each term of y is multiplied by 2)
>> p = z .* y
p =      2      4      6
                                % p is the vector obtained by term-by-term product of z and y
>> d = z ./ y
d = 0.5000    1.0000    1.5000
                                % d is the vector obtained by term-by-term division of z and y
>> t=linspace(-10,10,4)
t = -10.0000   -3.3333   3.3333   10.0000
                                % t has 4 numbers equally-spaced between -10 and 10
>> s = sin(t)
s = 0.5440    0.1906   -0.1906   -0.5440
                                % s is a vector obtained from the term-wise sin of the vector t
>> sSq = sin(t) .^ 2
sSq = 0.2960    0.0363    0.0363    0.2960
                                % sSq is an array obtained from term-wise squaring (. ^ 2) of the sin(t) array
>> cSq = cos(t) .^ 2
cSq = 0.7040    0.9637    0.9637    0.7040
                                % cSq is an array obtained from term-wise squaring (. ^ 2) of the cos(t) array
>> sSq + cSq % we can add the two arrays sSq and cSq to get the array of 1's
ans =      1      1      1      1
>> n = sin(t) .^ 2 + cos(t) .^ 2
n =      1      1      1      1
                                % we can directly do term-wise operation sin^2(t) + cos^2(t) of t as well
>> t2 = (-10:6.666665:10)
t2 = -10.0000   -3.3333   3.3333   10.0000
                                % t2 is similar to t above but with ':' syntax of (start:increment:stop)

```

Similarly, operations can be performed with matrices.

```

>> (0+0) .^ (1/2) % term-by-term square root of the matrix obtained by adding 0=ones(4,5) to itself
ans =
  1.4142    1.4142    1.4142    1.4142    1.4142
  1.4142    1.4142    1.4142    1.4142    1.4142
  1.4142    1.4142    1.4142    1.4142    1.4142
  1.4142    1.4142    1.4142    1.4142    1.4142

```

We can access specific rows or columns of a matrix as follows:

```

>> % declare a 3 X 3 array A of row vectors
>> A = [0.2760    0.4984    0.7513; 0.6797    0.9597    0.2551; 0.1626    0.5853    0.6991]
A =
  0.2760    0.4984    0.7513
  0.6797    0.9597    0.2551
  0.1626    0.5853    0.6991
>> A(2,:) % access the second row of A
ans =
  0.6797    0.9597    0.2551
>> B = A(2:3,:)
B =
  0.6797    0.9597    0.2551
  0.1626    0.5853    0.6991
>> C = A(:,[1 3]) % store the first and third columns of A in matrix C
C =
  0.2760    0.7513
  0.6797    0.2551

```

Labwork 8 (Plotting a function as points of ordered pairs in two arrays) Next we plot the function $\sin(x)$ from several ordered pairs $(x_i, \sin(x_i))$. Here x_i 's are from the domain $[-2\pi, 2\pi]$.

We use the `plot` function in MATLAB . Create an M-file called `MySineWave.m` and copy the following commands in it. By entering `MySineWave` in the command window you should be able to run the script and see the figure in the figure window.

```
SineWave.m
x = linspace(-2*pi,2*pi,100);           % x has 100 points equally spaced in [-2*pi, 2*pi]
y = sin(x);                            % y is the term-wise sin of x, ie sin of every number in x is in y, resp.
plot(x,y,'.');                         % plot x versus y as dots should appear in the Figure window
xlabel('x');                           % label x-axis with the single quote enclosed string x
ylabel('sin(x)', 'FontSize',16);        % label y-axis with the single quote enclosed string
title('Sine Wave in [-2 pi, 2 pi]', 'FontSize',16);    % give a title; click Figure window to see changes
set(gca,'XTick',-8:1:8,'FontSize',16) % change the range and size of X-axis ticks
% you can go to the Figure window's File menu to print/save the plot
```

The plot was saved as an encapsulated postscript file from the File menu of the Figure window and is displayed below.

Figure 1.7: A plot of the sine wave over $[-2\pi, 2\pi]$.

Let us first recall some elementary ideas from real analysis.

Definition 3 (Convergent sequence of real numbers) A sequence of real numbers $\langle x_i \rangle_{i=1}^{\infty} := x_1, x_2, \dots$ is said to converge to a limit $a \in \mathbb{R}$ and denoted by:

$$\lim_{i \rightarrow \infty} x_i = a ,$$

if for every natural number $m \in \mathbb{N}$, a natural number $N_m \in \mathbb{N}$ exists such that for every $j \geq N_m$, $|x_j - a| \leq \frac{1}{m}$.

Example 9 (Limit of a sequence of 17s) Let $\langle x_i \rangle_{i=1}^{\infty} = 17, 17, 17, \dots$. Then $\lim_{i \rightarrow \infty} x_i = 17$. This is because for every $m \in \mathbb{N}$, we can take $N_m = 1$ and satisfy the definition of the limit, i.e.:

$$\text{for every } j \geq N_m = 1, |x_j - 17| = |17 - 17| = 0 \leq \frac{1}{m} .$$

Example 10 (Limit of $1/i$) Let $\langle x_i \rangle_{i=1}^{\infty} = \frac{1}{1}, \frac{1}{2}, \frac{1}{3}, \dots$, i.e. $x_i = \frac{1}{i}$, then $\lim_{i \rightarrow \infty} x_i = 0$. This is because for every $m \in \mathbb{N}$, we can take $N_m = m$ and satisfy the definition of the limit, i.e.:

$$\text{for every } j \geq N_m = m, |x_j - 0| = \left| \frac{1}{j} - 0 \right| = \frac{1}{j} \leq \frac{1}{m} .$$

However, several other sequences also approach the limit 0. Some such sequences that approach the limit 0 from the right are:

$$\langle x_{1:\infty} \rangle = \frac{1}{1}, \frac{1}{4}, \frac{1}{9}, \dots \quad \text{and} \quad \langle x_{1:\infty} \rangle = \frac{1}{1}, \frac{1}{8}, \frac{1}{27}, \dots ,$$

and some that approach the limit 0 from the left are:

$$\langle x_{1:\infty} \rangle = -\frac{1}{1}, -\frac{1}{2}, -\frac{1}{3}, \dots \quad \text{and} \quad \langle x_{1:\infty} \rangle = -\frac{1}{1}, -\frac{1}{4}, -\frac{1}{9}, \dots ,$$

and finally some that approach 0 from either side are:

$$\langle x_{1:\infty} \rangle = -\frac{1}{1}, +\frac{1}{2}, -\frac{1}{3}, \dots \quad \text{and} \quad \langle x_{1:\infty} \rangle = -\frac{1}{1}, +\frac{1}{4}, -\frac{1}{9}, \dots .$$

When we do not particularly care about the specifics of a sequence of real numbers $\langle x_{1:\infty} \rangle$, in terms of the exact values it takes for each i , but we are only interested that it converges to a limit a we write:

$$x \rightarrow a$$

and say that x approaches a . If we are only interested in those sequences that converge to the limit a from the right or left, we write:

$$x \rightarrow a^+ \quad \text{or} \quad x \rightarrow a^-$$

and say x approaches a from the right or left, respectively.

Definition 4 (Limits of Functions) We say a function $f(x) : \mathbb{R} \rightarrow \mathbb{R}$ has a **limit** $L \in \mathbb{R}$ as x approaches a and write:

$$\lim_{x \rightarrow a} f(x) = L ,$$

provided $f(x)$ is arbitrarily close to L for all values of x that are sufficiently close to, but not equal to, a . We say that f has a **right limit** L_R or **left limit** L_L as x approaches a from the left or right, and write:

$$\lim_{x \rightarrow a^+} f(x) = L_R \quad \text{or} \quad \lim_{x \rightarrow a^-} f(x) = L_L ,$$

provided $f(x)$ is arbitrarily close to L_R or L_L for all values of x that are sufficiently close to, but not equal to, a from the right of a or the left of a , respectively. When the limit is not an element of \mathbb{R} or when the left and right limits are distinct, we say that the limit does not exist.

Example 11 (Limit of $1/x^2$) Consider the function $f(x) = \frac{1}{x^2}$. Then

$$\lim_{x \rightarrow 1} f(x) = \lim_{x \rightarrow 1} \frac{1}{x^2} = 1$$

exists since the limit $1 \in \mathbb{R}$, and the right and left limits are the same:

$$\lim_{x \rightarrow 1^+} f(x) = \lim_{x \rightarrow 1^+} \frac{1}{x^2} = 1 \quad \text{and} \quad \lim_{x \rightarrow 1^-} f(x) = \lim_{x \rightarrow 1^-} \frac{1}{x^2} = 1 .$$

However, the following limit does not exist:

$$\lim_{x \rightarrow 0} f(x) = \lim_{x \rightarrow 0} \frac{1}{x^2} = \infty$$

since $\infty \notin \mathbb{R}$.

Example 12 (Limit of $(1+x)^{\frac{1}{x}}$) The limit of $f(x) = (1+x)^{\frac{1}{x}}$ as x approaches 0 exists and it is the Euler's constant e :

$$\lim_{x \rightarrow 0} f(x) = \lim_{x \rightarrow 0} (1+x)^{\frac{1}{x}} = e \approx 2.71828 .$$

Notice that the above limit exists despite the fact that $f(0) = (1+0)^{\frac{1}{0}}$ itself is undefined and does not exist.

Example 13 (Limit of $\frac{x^3-1}{x-1}$) For $f(x) = \frac{x^3-1}{x-1}$, this limit exists:

$$\lim_{x \rightarrow 1} f(x) = \lim_{x \rightarrow 1} \frac{x^3 - 1}{x - 1} = \lim_{x \rightarrow 1} \frac{(x-1)(x^2 + x + 1)}{(x-1)} = \lim_{x \rightarrow 1} x^2 + x + 1 = 3$$

despite the fact that $f(1) = \frac{1^3-1}{1-1} = \frac{0}{0}$ itself is undefined and does not exist.

Next we look at some examples of limits at infinity.

Example 14 (Limit of $(1 - \frac{\lambda}{n})^n$) The limit of $f(n) = (1 - \frac{\lambda}{n})^n$ as n approaches ∞ exists and it is $e^{-\lambda}$:

$$\lim_{n \rightarrow \infty} f(n) = \lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^n = e^{-\lambda}.$$

Example 15 (Limit of $(1 - \frac{\lambda}{n})^{-\alpha}$) The limit of $f(n) = (1 - \frac{\lambda}{n})^{-\alpha}$, for some $\alpha > 0$, as n approaches ∞ exists and it is 1 :

$$\lim_{n \rightarrow \infty} f(n) = \lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^{-\alpha} = 1.$$

Definition 5 (Continuity of a function) We say a real-valued function $f(x) : D \rightarrow \mathbb{R}$ with the domain $D \subset \mathbb{R}$ is **right continuous** or **left continuous** at a point $a \in D$, provided:

$$\lim_{x \rightarrow a^+} f(x) = f(a) \quad \text{or} \quad \lim_{x \rightarrow a^-} f(x) = f(a),$$

respectively. We say f is **continuous** at $a \in D$, provided:

$$\lim_{x \rightarrow a^+} f(x) = f(a) = \lim_{x \rightarrow a^-} f(x).$$

Finally, f is said to be continuous if f is continuous at every $a \in D$.

Example 16 (Discontinuity of $f(x) = (1+x)^{\frac{1}{x}}$ at 0) Let us reconsider the function $f(x) = (1+x)^{\frac{1}{x}} : \mathbb{R} \rightarrow \mathbb{R}$. Clearly, $f(x)$ is continuous at 1, since:

$$\lim_{x \rightarrow 1} f(x) = \lim_{x \rightarrow 1} (1+x)^{\frac{1}{x}} = 2 = f(1) = (1+1)^{\frac{1}{1}},$$

but it is not continuous at 0, since:

$$\lim_{x \rightarrow 0} f(x) = \lim_{x \rightarrow 0} (1+x)^{\frac{1}{x}} = e \approx 2.71828 \neq f(0) = (1+0)^{\frac{1}{0}}.$$

Thus, $f(x)$ is not a continuous function over \mathbb{R} .

1.7 Elementary Number Theory

We introduce basic notions that we need from elementary number theory here. These notions include integer functions and modular arithmetic as they will be needed later on.

For any real number x :

$\lfloor x \rfloor := \max\{y : y \in \mathbb{Z} \text{ and } y \leq x\}$, i.e., the greatest integer less than or equal to x (the **floor** of x),

$\lceil x \rceil := \min\{y : y \in \mathbb{Z} \text{ and } y \geq x\}$, i.e., the least integer greater than or equal to x (the **ceiling** of x).

Example 17 (Floors and ceilings)

$$\lfloor 1 \rfloor = 1, \quad \lceil 1 \rceil = 1, \quad \lfloor 17.8 \rfloor = 17, \quad \lceil -17.8 \rceil = -18, \quad \lceil \sqrt{2} \rceil = 2, \quad \lfloor \pi \rfloor = 3, \quad \lceil \frac{1}{10^{100}} \rceil = 1.$$

Labwork 18 (Floors and ceilings in MATLAB) We can use MATLAB functions `floor` and `ceil` to compute $\lfloor x \rfloor$ and $\lceil x \rceil$, respectively. Also, the argument x to these functions can be an array.

```
>> sqrt(2) % the square root of 2 is
ans =      1.4142
>> ceil(sqrt(2)) % ceiling of square root of 2
ans =      2
>> floor(-17.8) % floor of -17.8
ans =     -18
>> ceil([1 sqrt(2) pi -17.8 1/(10^100)]) %the ceiling of each element of an array
ans =      1      2      4     -17      1
>> floor([1 sqrt(2) pi -17.8 1/(10^100)]) % the floor of each element of an array
ans =      1      1      3    -18      0
```

Classwork 19 (Relations between floors and ceilings) Convince yourself of the following formulae. Use examples, plots and/or formal arguments.

$$\begin{aligned}\lceil x \rceil &= \lfloor x \rfloor \iff x \in \mathbb{Z} \\ \lceil x \rceil &= \lfloor x \rfloor + 1 \iff x \notin \mathbb{Z} \\ \lfloor -x \rfloor &= -\lceil x \rceil \\ x - 1 < \lfloor x \rfloor &\leq x \leq \lceil x \rceil < x + 1\end{aligned}$$

Let us define modular arithmetic next. Suppose x and y are any real numbers, i.e. $x, y \in \mathbb{R}$, we define the binary operation called “ $x \bmod y$ ” as:

$$x \bmod y := \begin{cases} x - y\lfloor x/y \rfloor & \text{if } y \neq 0 \\ x & \text{if } y = 0 \end{cases}$$

Chapter 2

Probability Model

2.1 Experiments

Ideas about chance events and random behaviour arose out of thousands of years of game playing, long before any attempt was made to use mathematical reasoning about them. Board and dice games were well known in Egyptian times, and Augustus Caesar gambled with dice. Calculations of odds for gamblers were put on a proper theoretical basis by Fermat and Pascal in the early 17th century.

Definition 6 An **experiment** is an activity or procedure that produces distinct, well-defined possibilities called **outcomes**. The set of all outcomes is called the **sample space**, and is denoted by Ω .

The subsets of Ω are called **events**. A single outcome, ω , when seen as a subset of Ω , as in $\{\omega\}$, is called a **simple event**.

Events, $E_1, E_2 \dots E_n$, that cannot occur at the same time are called **mutually exclusive** events, or **pair-wise disjoint** events. This means that $E_i \cap E_j = \emptyset$ where $i \neq j$.

Example 20 Some standard examples of experiments are the following:

- $\Omega = \{\text{Defective, Non-defective}\}$ if our experiment is to inspect a light bulb.

There are only two outcomes here, so $\Omega = \{\omega_1, \omega_2\}$ where $\omega_1 = \text{Defective}$ and $\omega_2 = \text{Non-defective}$.

- $\Omega = \{\text{Heads, Tails}\}$ if our experiment is to note the outcome of a coin toss.

This time, $\Omega = \{\omega_1, \omega_2\}$ where $\omega_1 = \text{Heads}$ and $\omega_2 = \text{Tails}$.

- If our experiment is to roll a die then there are six outcomes corresponding to the number that shows on the top. For this experiment, $\Omega = \{1, 2, 3, 4, 5, 6\}$.

Some examples of events are the set of odd numbered outcomes $A = \{1, 3, 5\}$, and the set of even numbered outcomes $B = \{2, 4, 6\}$.

The simple events of Ω are $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}$, and $\{6\}$.

The outcome of a random experiment is uncertain until it is performed and observed. Note that sample spaces need to reflect the problem in hand. The example below is to convince you that an experiment's sample space is merely a collection of distinct elements called outcomes and these outcomes have to be *discernible in some well-specified sense* to the experimenter!

Example 21 Consider a generic die-tossing experiment by a human experimenter. Here $\Omega = \{\omega_1, \omega_2, \omega_3, \dots, \omega_6\}$, but the experiment might correspond to rolling a die whose faces are:

1. sprayed with six different scents (nose!), or
2. studded with six distinctly flavoured candies (tongue!), or
3. contoured with six distinct bumps and pits (touch!), or
4. acoustically discernible at six different frequencies (ears!), or
5. painted with six different colours (eyes!), or
6. marked with six different numbers 1, 2, 3, 4, 5, 6 (eyes!), or , ...

These six experiments are equivalent as far as probability goes.

Definition 7 A **trial** is a single performance of an experiment and it results in an outcome.

Example 22 Some standard examples of a trial are:

- A roll of a die.
- A toss of a coin.
- A release of a chaotic double pendulum.

An experimenter often performs more than one trial. Repeated trials of an experiment forms the basis of science and engineering as the experimenter learns about the phenomenon by repeatedly performing the same mother experiment with possibly different outcomes. This repetition of trials in fact provides the very motivation for the definition of probability.

Definition 8 An **n-product experiment** is obtained by repeatedly performing n trials of some experiment. The experiment that is repeated is called the “mother” experiment.

Experiment 1 (The Bernoulli Product Experiment; Toss a coin n times) Suppose our experiment entails tossing a coin n times and recording H for Heads and T for Tails. When $n = 3$, one possible outcome of this experiment is HHT, ie. a Head followed by another Head and then a Tail. Seven other outcomes are possible.

The sample space for “toss a coin three times” experiment is:

$$\Omega = \{H, T\}^3 = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\},$$

with a particular sample point or outcome $\omega = HTH$, and another distinct outcome $\omega' = HHH$. An event, say A , that ‘at least two Heads occur’ is the following subset of Ω :

$$A = \{HHH, HHT, HTH, THH\}.$$

Another event, say B , that ‘no Heads occur’ is:

$$B = TTT$$

Note that the event B is also an outcome or sample point. Another interesting event is the empty set $\emptyset \subset \Omega$. The event that ‘nothing in the sample space occurs’ is \emptyset .

Figure 2.1: A binary tree whose leaves are all possible outcomes.

Classwork 23 (A thrice-bifurcating tree of outcomes) Can you think of a graphical way to enumerate the outcomes of the Experiment 1? Draw a diagram of this under the caption of Figure 2.1, using the caption as a hint (in other words, draw your own Figure 2.1).

EXPERIMENT SUMMARY

Experiment	–	an activity producing distinct outcomes.
Ω	–	set of all outcomes of the experiment.
ω	–	an individual outcome in Ω , called a simple event.
$A \subseteq \Omega$	–	a subset A of Ω is an event.
Trial	–	one performance of an experiment resulting in 1 outcome.

2.2 Probability

The mathematical model for probability or the probability model is an axiomatic system that may be motivated by the intuitive idea of ‘long-term relative frequency’. If the axioms and definitions are intuitively motivated, the probability model simply follows from the application of logic to these axioms and definitions. No attempt to define probability in the real world is made. However, the application of probability models to real-world problems through statistical experiments has a fruitful track record. In fact, you are here for exactly this reason.

Idea 9 (The long-term relative frequency (LTRF) idea) Suppose we are interested in the fairness of a coin, i.e. if landing Heads has the same “probability” as landing Tails. We can toss it n times and call $N(H, n)$ the fraction of times we observed Heads out of n tosses. Suppose that after conducting the tossing experiment 1000 times, we rarely observed Heads, e.g. 9 out of the 1000 tosses, then $N(H, 1000) = 9/1000 = 0.009$. Suppose we continued the number of tosses to a million and found that this number approached closer to 0.1, or, more generally, $N(H, n) \rightarrow 0.1$ as $n \rightarrow \infty$. We might, at least intuitively, think that the coin is unfair and has a lower “probability” of 0.1 of landing Heads. We might think that it is fair had we observed $N(H, n) \rightarrow 0.5$ as $n \rightarrow \infty$. Other crucial assumptions that we have made here are:

1. **Something Happens:** Each time we toss a coin, we are certain to observe Heads **or** Tails, denoted by $H \cup T$. The probability that “something happens” is 1. More formally:

$$N(H \cup T, n) = \frac{n}{n} = 1.$$

This is an intuitively reasonable assumption that simply says that one of the possible outcomes is certain to occur, provided the coin is not so thick that it can land on or even roll along its circumference.

2. **Addition Rule:** Heads and Tails are mutually exclusive events in any given toss of a coin, i.e. they cannot occur simultaneously. The intersection of mutually exclusive events is the empty set and is denoted by $H \cap T = \emptyset$. The event $H \cup T$, namely that the event that “coin lands Heads **or** coin lands Tails” satisfies:

$$N(H \cup T, n) = N(H, n) + N(T, n).$$

3. The coin-tossing experiment is repeatedly performed in an **independent** manner, i.e. the outcome of any individual coin-toss does not affect that of another. This is an intuitively reasonable assumption since the coin has no memory and the coin is tossed identically each time.

We will use the LTRF idea more generally to motivate a mathematical model of probability called probability model. Suppose A is an event associated with some experiment \mathcal{E} , so that A either does or does not occur when the experiment is performed. We want the probability that event A occurs in a specific performance of \mathcal{E} , denoted by $\mathbf{P}(A)$, to intuitively mean the following: if one were to perform a super-experiment \mathcal{E}^∞ by independently repeating the experiment \mathcal{E} and recording $N(A, n)$, the fraction of times A occurs in the first n performances of \mathcal{E} within the super-experiment \mathcal{E}^∞ . Then the LTRF idea suggests:

$$N(A, n) := \frac{\text{Number of times } A \text{ occurs}}{n = \text{Number of performances of } \mathcal{E}} \rightarrow \mathbf{P}(A), \text{ as } n \rightarrow \infty \quad (2.1)$$

Now, we are finally ready to define probability.

Definition 10 (Probability) Let \mathcal{E} be an experiment with sample space Ω . Let \mathcal{F} denote a suitable collection of events in Ω that satisfy the following conditions:

1. It (the collection) contains the sample space: $\boxed{\Omega \in \mathcal{F}}$.
2. It is closed under complementation: $\boxed{A \in \mathcal{F} \implies A^c \in \mathcal{F}}$.
3. It is closed under countable unions: $\boxed{A_1, A_2, \dots \in \mathcal{F} \implies \bigcup_i A_i := A_1 \cup A_2 \cup \dots \in \mathcal{F}}$.

Formally, this collection of events is called a **sigma field** or a **sigma algebra**. Our experiment \mathcal{E} has a sample space Ω and a collection of events \mathcal{F} that satisfy the three condition.

Given a double, e.g. (Ω, \mathcal{F}) , **probability** is just a function \mathbf{P} which assigns each event $A \in \mathcal{F}$ a number $\mathbf{P}(A)$ in the real interval $[0, 1]$, i.e. $\boxed{\mathbf{P} : \mathcal{F} \rightarrow [0, 1]}$, such that:

1. The ‘Something Happens’ axiom holds, i.e. $\boxed{\mathbf{P}(\Omega) = 1}$.
2. The ‘Addition Rule’ axiom holds, i.e. for events A and B :

$$\boxed{A \cap B = \emptyset \implies \mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B)}.$$

2.2.1 Consequences of our Definition of Probability

It is important to realize that we accept the ‘addition rule’ as an axiom in our mathematical definition of probability (or our probability model) and we do **not** prove this rule. However, the facts which are stated (with proofs) below, are logical consequences of our definition of probability:

1. For any event A , $\boxed{\mathbf{P}(A^c) = 1 - \mathbf{P}(A)}.$

Proof: One line proof.

$$\overbrace{\mathbf{P}(A) + \mathbf{P}(A^c)}^{LHS} = \underbrace{\mathbf{P}(A \cup A^c)}_{+ \text{ rule } \because A \cap A^c = \emptyset} = \underbrace{\mathbf{P}(\Omega)}_{A \cup A^c = \Omega} = \underbrace{1}_{\because \mathbf{P}(\Omega) = 1} \stackrel{RHS}{\Rightarrow} \mathbf{P}(A^c) = 1 - \mathbf{P}(A)$$

- If $A = \Omega$ then $A^c = \Omega^c = \emptyset$ and $\boxed{\mathbf{P}(\emptyset) = 1 - \mathbf{P}(\Omega) = 1 - 1 = 0}.$

2. For any two events A and B , we have the **inclusion-exclusion principle**:

$$\boxed{\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B) - \mathbf{P}(A \cap B)}.$$

Proof: Since:

$$\begin{aligned} A &= (A \setminus B) \cup (A \cap B) && \text{and} && (A \setminus B) \cap (A \cap B) = \emptyset, \\ A \cup B &= (A \setminus B) \cup B && \text{and} && (A \setminus B) \cap B = \emptyset \end{aligned}$$

the addition rule implies that:

$$\begin{aligned} \mathbf{P}(A) &= \mathbf{P}(A \setminus B) + \mathbf{P}(A \cap B) \\ \mathbf{P}(A \cup B) &= \mathbf{P}(A \setminus B) + \mathbf{P}(B) \end{aligned}$$

Substituting the first equality above into the second, we get:

$$\mathbf{P}(A \cup B) = \mathbf{P}(A \setminus B) + \mathbf{P}(B) = \mathbf{P}(A) - \mathbf{P}(A \cap B) + \mathbf{P}(B)$$

3. For a sequence of mutually disjoint events $A_1, A_2, A_3, \dots, A_n$:

$$\boxed{A_i \cap A_j = \emptyset \text{ for any } i, j \implies \mathbf{P}(A_1 \cup A_2 \cup \dots \cup A_n) = \mathbf{P}(A_1) + \mathbf{P}(A_2) + \dots + \mathbf{P}(A_n).}$$

Proof: If A_1, A_2, A_3 are mutually disjoint events, then $A_1 \cup A_2$ is disjoint from A_3 . Thus, two applications of the addition rule for disjoint events yields:

$$\mathbf{P}(A_1 \cup A_2 \cup A_3) = \mathbf{P}((A_1 \cup A_2) \cup A_3) \underset{+ \text{ rule}}{=} \mathbf{P}(A_1 \cup A_2) + \mathbf{P}(A_3) \underset{+ \text{ rule}}{=} \mathbf{P}(A_1) + \mathbf{P}(A_2) + \mathbf{P}(A_3)$$

The n -event case follows by mathematical induction.

We have formally defined the **probability model** specified by the **probability triple** $(\Omega, \mathcal{F}, \mathbf{P})$ that can be used to model an **experiment** \mathcal{E} .

Example 24 (First Ball out of NZ Lotto) Let us observe the number on *the first ball that pops out in a New Zealand Lotto trial*. There are forty balls labelled 1 through 40 for this experiment and so the sample space is

$$\Omega = \{1, 2, 3, \dots, 39, 40\}.$$

Because the balls are vigorously whirled around inside the Lotto machine before the first one pops out, we can model each ball to pop out first with the same probability. So, we assign each outcome $\omega \in \Omega$ the same probability of $\frac{1}{40}$, i.e., our probability model for this experiment is:

$$\mathbf{P}(\omega) = \frac{1}{40}, \text{ for each } \omega \in \Omega = \{1, 2, 3, \dots, 39, 40\} .$$

(Note: We sometimes abuse notation and write $\mathbf{P}(\omega)$ instead of the more accurate but cumbersome $\mathbf{P}(\{\omega\})$ when writing down probabilities of simple events.)

Figure 2.2 (a) shows the frequency of the first ball number in 1114 NZ Lotto draws. Figure 2.2 (b) shows the relative frequency, i.e., the frequency divided by 1114, the number of draws. Figure 2.2 (b) also shows the equal probabilities under our model.

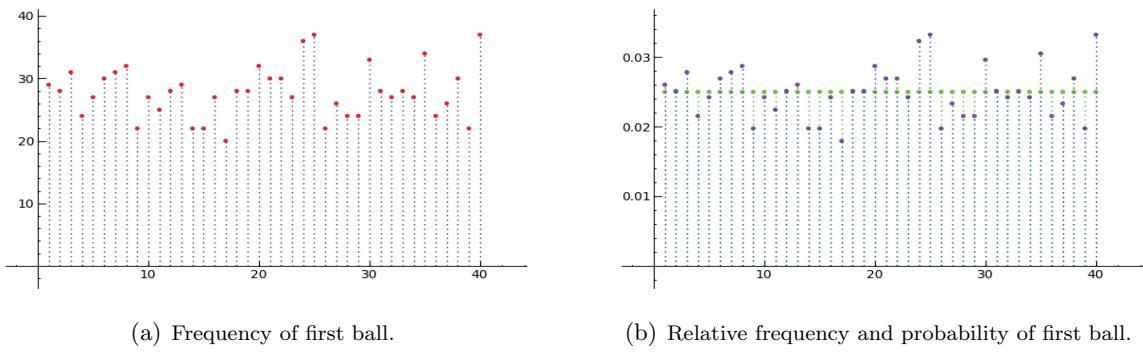


Figure 2.2: First ball number in 1114 NZ Lotto draws from 1987 to 2008.

Next, let us take a detour into how one might interpret it in the real world. The following is an adaptation from Williams D, *Weighing the Odds: A Course in Probability and Statistics*, Cambridge University Press, 2001, which henceforth is abbreviated as WD2001.

Probability Model

Sample space Ω

Sample point ω

(No counterpart)

Event A, a (suitable) subset of Ω

$\mathbf{P}(A)$, a number between 0 and 1

Real-world Interpretation

Set of all outcomes of an experiment

Possible outcome of an experiment

Actual outcome ω^* of an experiment

The real-world event corresponding to A

occurs if and only if $\omega^* \in A$

Probability that A will occur for an experiment yet to be performed

Events in Probability Model

Sample space Ω

The \emptyset of Ω

The intersection $A \cap B$

$A_1 \cap A_2 \cap \dots \cap A_n$

The union $A \cup B$

$A_1 \cup A_2 \cup \dots \cup A_n$

A^c , the complement of A

$A \setminus B$

$A \subset B$

Real-world Interpretation

The certain even ‘something happens’

The impossible event ‘nothing happens’

‘Both A and B occur’

‘All of the events A_1, A_2, \dots, A_n occur simultaneously’

‘At least one of A and B occurs’

‘At least one of the events A_1, A_2, \dots, A_n occurs’

‘A does not occur’

‘A occurs, but B does not occur’

‘If A occurs, then B must occur’

In the probability model of Example 24, show that for any event $E \subset \Omega$,

$$\mathbf{P}(E) = \frac{1}{40} \times \text{number of elements in } E .$$

Let $E = \{\omega_1, \omega_2, \dots, \omega_k\}$ be an event with k outcomes (simple events). Then by the addition rule for mutually exclusive events we get:

$$\mathbf{P}(E) = \mathbf{P}(\{\omega_1, \omega_2, \dots, \omega_k\}) = \mathbf{P}\left(\bigcup_{i=1}^k \{\omega_i\}\right) = \sum_{i=1}^k \mathbf{P}(\{\omega_i\}) = \sum_{i=1}^k \frac{1}{40} = \frac{k}{40} .$$

2.2.2 Sigma Algebras of Typical Experiments*

Example 25 ('Toss a fair coin once') Consider the 'Toss a fair coin once' experiment. What is its sample space Ω and a reasonable collection of events \mathcal{F} that underpin this experiment?

$$\Omega = \{\text{H}, \text{T}\}, \quad \mathcal{F} = \{\Omega, \emptyset, \{\text{H}\}, \{\text{T}\}\} ,$$

A function that will satisfy the definition of probability for this collection of events \mathcal{F} and assign $\mathbf{P}(\text{H}) = \frac{1}{2}$ is summarized below. First check that the above \mathcal{F} is a sigma-algebra. Draw a picture for \mathbf{P} with arrows that map elements in the domain \mathcal{F} given above to elements in its range.

Event $A \in \mathcal{F}$	$\mathbf{P} : \mathcal{F} \rightarrow [0, 1]$	$\mathbf{P}(A) \in [0, 1]$
$\Omega = \{\text{H}, \text{T}\} \bullet$	→	1
$\text{T} \bullet$	→	$1 - \frac{1}{2}$
$\text{H} \bullet$	→	$\frac{1}{2}$
$\emptyset \bullet$	→	0

Classwork 26 (The trivial sigma algebra) Note that $\mathcal{F}' = \{\Omega, \emptyset\}$ is also a sigma algebra of the sample space $\Omega = \{\text{H}, \text{T}\}$. Can you think of a probability for the collection \mathcal{F}' ?

Event $A \in \mathcal{F}'$	$\mathbf{P} : \mathcal{F}' \rightarrow [0, 1]$	$\mathbf{P}(A) \in [0, 1]$
$\Omega = \{\text{H}, \text{T}\} \bullet$	→	
$\emptyset \bullet$	→	

Thus, \mathcal{F} and \mathcal{F}' are two distinct sigma algebras over our $\Omega = \{\text{H}, \text{T}\}$. Moreover, $\mathcal{F}' \subset \mathcal{F}$ and is called a sub sigma algebra. Try to show that $\{\Omega, \emptyset\}$ is the smallest possible sigma algebra over all possible sigma algebras over any given sample space Ω (think of intersecting an arbitrary family of sigma algebras)?

Generally one encounters four types of sigma algebras and they are:

- When the sample space $\Omega = \{\omega_1, \omega_2, \dots, \omega_k\}$ is a finite set with k outcomes and $\mathbf{P}(\omega_i)$, the probability for each outcome $\omega_i \in \Omega$ is known, then one typically takes the sigma-algebra \mathcal{F} to be the set of all subsets of Ω called the **power set** and denoted by 2^Ω . The probability of each event $A \in 2^\Omega$ can be obtained by adding the probabilities of the outcomes in A , i.e., $\mathbf{P}(A) = \sum_{\omega_i \in A} \mathbf{P}(\omega_i)$. Clearly, 2^Ω is indeed a sigma-algebra and it contains $2^{\#\Omega}$ events in it.
- When the sample space $\Omega = \{\omega_1, \omega_2, \dots\}$ is a countable set then one typically takes the sigma-algebra \mathcal{F} to be the set of all subsets of Ω . Note that this is very similar to the case with finite Ω except now $\mathcal{F} = 2^\Omega$ could have uncountably many events in it.

3. If $\Omega = \mathbb{R}^d$ for finite $d \in \{1, 2, 3, \dots\}$ then the **Borel sigma-algebra** is the smallest sigma-algebra containing all **half-spaces**, i.e., sets of the form

$$\{x = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d : x_1 \leq c_1, x_2 \leq c_2, \dots, x_d \leq c_d\}, \quad \text{for any } c = (c_1, c_2, \dots, c_d) \in \mathbb{R}^d,$$

When $d = 1$ the half-spaces are the half-lines $\{(-\infty, c] : c \in \mathbb{R}\}$ and when $d = 2$ the half-spaces are the south-west quadrants $\{(-\infty, c_1] \times (-\infty, c_2] : (c_1, c_2) \in \mathbb{R}^2\}$, etc. (Equivalently, the Borel sigma-algebra is the smallest sigma-algebra containing all open sets in \mathbb{R}^d).

4. Given a finite set $\mathbb{S} = \{s_1, s_2, \dots, s_k\}$, let Ω be the sequence space $\mathbb{S}^\infty := \mathbb{S} \times \mathbb{S} \times \mathbb{S} \times \dots$, i.e., the set of sequences of infinite length that are made up of elements from \mathbb{S} . A set of the form

$$A_1 \times A_2 \times \dots \times A_n \times \mathbb{S} \times \mathbb{S} \times \dots, \quad A_k \subset \mathbb{S} \text{ for all } k \in \{1, 2, \dots, n\},$$

is called a **cylinder set**. The set of events in \mathbb{S}^∞ is the smallest sigma-algebra containing the cylinder sets.

Exercises

Ex. 2.1 — In English language text, the twenty six letters in the alphabet occur with the following frequencies:

E	13%	R	7.7%	A	7.3%	H	3.5%	F	2.8%	M	2.5%	W	1.6%	X	0.5%	J	0.2%
T	9.3%	O	7.4%	S	6.3%	L	3.5%	P	2.7%	Y	1.9%	V	1.3%	K	0.3%	Z	0.1%
N	7.8%	I	7.4%	D	4.4%	C	3%	U	2.7%	G	1.6%	B	0.9%	Q	0.3%		

Suppose you pick one letter at random from a randomly chosen English book from our central library with $\Omega = \{A, B, C, \dots, Z\}$ (ignoring upper/lower cases), then what is the probability of these events?

- (a) $\mathbf{P}(\{Z\})$
- (b) $\mathbf{P}(\text{'picking any letter'})$
- (c) $\mathbf{P}(\{E, Z\})$
- (d) $\mathbf{P}(\text{'picking a vowel'})$
- (e) $\mathbf{P}(\text{'picking any letter in the word WAZZZUP'})$
- (f) $\mathbf{P}(\text{'picking any letter in the word WAZZZUP or a vowel'})$.

Ex. 2.2 — Find the sample spaces for the following experiments:

1. Tossing 2 coins whose faces are sprayed with black paint denoted by B and white paint denoted by W.
2. Drawing 4 screws from a bucket of left-handed and right-handed screws denoted by L and R, respectively.
3. Rolling a die and recording the number on the upturned face until the first 6 appears.

Ex. 2.3 — Suppose we pick a letter at random from the word WAIMAKARIRI.

1. What is the sample space Ω ?
2. What probabilities should be assigned to the outcomes?
3. What is the probability of *not* choosing the letter R?

Ex. 2.4 — There are seventy five balls in total inside the Bingo Machine. Each ball is labelled by one of the following five letters: B, I, N, G, and O. There are fifteen balls labelled by each letter. The letter on the first ball that comes out of a BINGO machine after it has been well-mixed is the outcome of our experiment.

- (a) Write down the sample space of this experiment.
- (b) Find the probabilities of each simple event.
- (c) Show that $\mathbf{P}(\Omega)$ is indeed 1.
- (d) Check that the addition rule for mutually exclusive events holds for the simple events $\{B\}$ and $\{I\}$.
- (e) Consider the following events: $C = \{B, I, G\}$ and $D = \{G, I, N\}$. Using the addition rule for two arbitrary events, find $\mathbf{P}(C \cup D)$.

PROBABILITY SUMMARY

Axioms:

1. If $A \subseteq \Omega$ then $0 \leq \mathbf{P}(A) \leq 1$ and $\mathbf{P}(\Omega) = 1$.
2. If A, B are disjoint events, then $\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B)$.
[This is true only when A and B are disjoint.]
3. If A_1, A_2, \dots are disjoint then $\mathbf{P}(A_1 \cup A_2 \cup \dots) = \mathbf{P}(A_1) + \mathbf{P}(A_2) + \dots$

Rules:

$$\begin{aligned}\mathbf{P}(A^c) &= 1 - \mathbf{P}(A) \\ \mathbf{P}(A \cup B) &= \mathbf{P}(A) + \mathbf{P}(B) - \mathbf{P}(A \cap B) \quad [\text{always true}]\end{aligned}$$

2.3 Conditional Probability

Next, we define conditional probability and the notion of independence of events. We use the LTRF idea to motivate the definition.

Idea 11 (LTRF intuition for conditional probability) Let A and B be any two events associated with our experiment \mathcal{E} with $\mathbf{P}(A) \neq 0$. The ‘conditional probability that B occurs given that A occurs’ denoted by $\mathbf{P}(B|A)$ is again intuitively underpinned by the super-experiment \mathcal{E}^∞ which is the ‘independent’ repetition of our original experiment \mathcal{E} ‘infinitely’ often. The LTRF idea is that $\mathbf{P}(B|A)$ is the long-term proportion of those experiments on which A occurs that B also occurs.

Recall that $N(A, n)$ as defined in (2.1) is the fraction of times A occurs out of n independent repetitions of our experiment \mathcal{E} (ie. the experiment \mathcal{E}^n). If $A \cap B$ is the event that ‘ A and B occur simultaneously’, then we intuitively want

$$\mathbf{P}(B|A) \quad “\rightarrow” \quad \frac{N(A \cap B, n)}{N(A, n)} = \frac{N(A \cap B, n)/n}{N(A, n)/n} = \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(A)}$$

as our $\mathcal{E}^n \rightarrow \mathcal{E}^\infty$. So, we **define** conditional probability as we want.

Definition 12 (Conditional Probability) Suppose we are given an experiment \mathcal{E} with a triple $(\Omega, \mathcal{F}, \mathbf{P})$. Let A and B be events, ie. $A, B \in \mathcal{F}$, such that $\mathbf{P}(A) \neq 0$. Then, we define the **conditional probability** of B given A by,

$$\mathbf{P}(B|A) := \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(A)} . \quad (2.2)$$

Note that for a **fixed** event $A \in \mathcal{F}$ with $\mathbf{P}(A) > 0$ and **any** event $B \in \mathcal{F}$, the conditional probability $\mathbf{P}(B|A)$ is a probability as in Definition 10, ie. a function:

$$\mathbf{P}(B|A) : \mathcal{F} \rightarrow [0, 1]$$

that assigns to each $B \in \mathcal{F}$ a number in the interval $[0, 1]$, such that,

1. $\mathbf{P}(\Omega|A) = 1$ Meaning ‘Something Happens given the event A happens’
2. The ‘Addition Rule’ axiom holds, ie. for events $B_1, B_2 \in \mathcal{F}$,

$$B_1 \cap B_2 = \emptyset \text{ implies } \mathbf{P}(B_1 \cup B_2|A) = \mathbf{P}(B_1|A) + \mathbf{P}(B_2|A) .$$

3. For mutually exclusive or pairwise-disjoint events, B_1, B_2, \dots ,

$$\mathbf{P}(B_1 \cup B_2 \cup \dots |A) = \mathbf{P}(B_1|A) + \mathbf{P}(B_2|A) + \dots .$$

From the definition of conditional probability we get the following rules:

1. Complementation rule: $\mathbf{P}(B|A) = 1 - \mathbf{P}(B^c|A)$.
2. Addition rule for two arbitrary events B_1 and B_2 :

$$\mathbf{P}(B_1 \cup B_2|A) = \mathbf{P}(B_1|A) + \mathbf{P}(B_2|A) - \mathbf{P}(B_1 \cap B_2|A) .$$

3. Multiplication rule for two likely events:

If A and B are events, and if $\mathbf{P}(A) \neq 0$ and $\mathbf{P}(B) \neq 0$, then

$$\mathbf{P}(A \cap B) = \mathbf{P}(A)\mathbf{P}(B|A) = \mathbf{P}(B)\mathbf{P}(A|B) .$$

Example 27 (Wasserman03, p. 11) A medical test for a disease D has outcomes + and -. the probabilities are:

	Have Disease (D)	Don't have disease (D^c)
Test positive (+)	0.009	0.099
Test negative (-)	0.001	0.891

Using the definition of conditional probability, we can compute the conditional probability that you test positive given that you have the disease:

$$\mathbf{P}(+|D) = \frac{\mathbf{P}(+ \cap D)}{\mathbf{P}(D)} = \frac{0.009}{0.009 + 0.001} = 0.9 ,$$

and the conditional probability that you test negative given that you don't have the disease:

$$\mathbf{P}(-|D^c) = \frac{\mathbf{P}(- \cap D^c)}{\mathbf{P}(D^c)} = \frac{0.891}{0.099 + 0.891} \approx 0.9 .$$

Thus, the test is quite accurate since sick people test positive 90% of the time and healthy people test negative 90% of the time.

Now, suppose you go for a test and test positive. What is the probability that you have the disease ?

$$\mathbf{P}(D|+) = \frac{\mathbf{P}(D \cap +)}{\mathbf{P}(+)} = \frac{0.009}{0.009 + 0.099} \approx 0.08$$

Most people who are not used to the definition of conditional probability would intuitively associate a number much bigger than 0.08 for the answer. Interpret conditional probability in terms of the meaning of the numbers that appear in the numerator and denominator of the above calculations.

Next we look at one of the most elegant applications of the definition of conditional probability along with the addition rule for a partition of Ω .

Proposition 13 (Bayes' Theorem, 1763) Suppose the events $A_1, A_2, \dots, A_k \in \mathcal{F}$, with $\mathbf{P}(A_h) > 0$ for each $h \in \{1, 2, \dots, k\}$, partition the sample space Ω , ie. they are mutually exclusive (disjoint) and exhaustive events with positive probability:

$$A_i \cap A_j = \emptyset, \text{ for any distinct } i, j \in \{1, 2, \dots, k\}, \quad \bigcup_{h=1}^k A_h = \Omega, \quad \mathbf{P}(A_h) > 0$$

Thus, precisely one of the A_h 's will occur on any performance of our experiment \mathcal{E} .

Let $B \in \mathcal{F}$ be some event with $\mathbf{P}(B) > 0$, then

$$\mathbf{P}(A_h|B) = \frac{\mathbf{P}(B|A_h)\mathbf{P}(A_h)}{\sum_{h=1}^k \mathbf{P}(B|A_h)\mathbf{P}(A_h)} \quad (2.3)$$

Proof: We apply elementary set theory, the definition of conditional probability $k + 2$ times and the addition rule once:

$$\begin{aligned} \mathbf{P}(A_h|B) &= \frac{\mathbf{P}(A_h \cap B)}{\mathbf{P}(B)} = \frac{\mathbf{P}(B \cap A_h)}{\mathbf{P}(B)} = \frac{\mathbf{P}(B|A_h)\mathbf{P}(A_h)}{\mathbf{P}(B)} \\ &= \frac{\mathbf{P}(B|A_h)\mathbf{P}(A_h)}{\mathbf{P}\left(\bigcup_{h=1}^k (B \cap A_h)\right)} = \frac{\mathbf{P}(B|A_h)\mathbf{P}(A_h)}{\sum_{h=1}^k \mathbf{P}(B \cap A_h)} \\ &= \frac{\mathbf{P}(B|A_h)\mathbf{P}(A_h)}{\sum_{h=1}^k \mathbf{P}(B|A_h)\mathbf{P}(A_h)} \end{aligned}$$

The operations done to the denominator in the proof above:

$$\mathbf{P}(B) = \sum_{h=1}^k \mathbf{P}(B|A_h)\mathbf{P}(A_h) \quad (2.4)$$

is also called ‘the law of total probability’ or ‘the total probability theorem’. We call $\mathbf{P}(A_h)$ the **prior probability of A_h** and $\mathbf{P}(A_h|B)$ the **posterior probability of A_h** .

Example 28 (Wasserman2003 p.12) Suppose Larry divides his email into three categories: A_1 = “spam”, A_2 = “low priority”, and A_3 = “high priority”. From previous experience, he finds that $\mathbf{P}(A_1) = 0.7$, $\mathbf{P}(A_2) = 0.2$ and $\mathbf{P}(A_3) = 0.1$. Note that $\mathbf{P}(A_1 \cup A_2 \cup A_3) = \mathbf{P}(\Omega) = 0.7 + 0.2 + 0.1 = 1$. Let B be the event that the email contains the word “free.” From previous experience, $\mathbf{P}(B|A_1) = 0.9$, $\mathbf{P}(B|A_2) = 0.01$ and $\mathbf{P}(B|A_3) = 0.01$. Note that $\mathbf{P}(B|A_1) + \mathbf{P}(B|A_2) + \mathbf{P}(B|A_3) = 0.9 + 0.01 + 0.01 \neq 1$. Now, suppose Larry receives an email with the word “free.” What is the probability that it is “spam,” “low priority,” and “high priority” ?

$$\mathbf{P}(A_1|B) = \frac{\mathbf{P}(B|A_1)\mathbf{P}(A_1)}{\mathbf{P}(B|A_1)\mathbf{P}(A_1) + \mathbf{P}(B|A_2)\mathbf{P}(A_2) + \mathbf{P}(B|A_3)\mathbf{P}(A_3)} = \frac{0.9 \times 0.7}{(0.9 \times 0.7) + (0.01 \times 0.2) + (0.01 \times 0.1)} = \frac{0.63}{0.633} \approx 0.995$$

$$\mathbf{P}(A_2|B) = \frac{\mathbf{P}(B|A_2)\mathbf{P}(A_2)}{\mathbf{P}(B|A_1)\mathbf{P}(A_1) + \mathbf{P}(B|A_2)\mathbf{P}(A_2) + \mathbf{P}(B|A_3)\mathbf{P}(A_3)} = \frac{0.01 \times 0.2}{(0.9 \times 0.7) + (0.01 \times 0.2) + (0.01 \times 0.1)} = \frac{0.002}{0.633} \approx 0.003$$

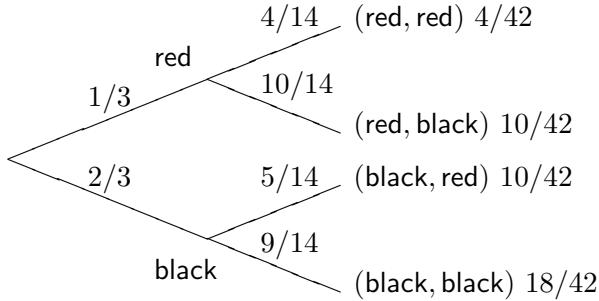
$$\mathbf{P}(A_3|B) = \frac{\mathbf{P}(B|A_3)\mathbf{P}(A_3)}{\mathbf{P}(B|A_1)\mathbf{P}(A_1) + \mathbf{P}(B|A_2)\mathbf{P}(A_2) + \mathbf{P}(B|A_3)\mathbf{P}(A_3)} = \frac{0.01 \times 0.1}{(0.9 \times 0.7) + (0.01 \times 0.2) + (0.01 \times 0.1)} = \frac{0.001}{0.633} \approx 0.002$$

Note that $\mathbf{P}(A_1|B) + \mathbf{P}(A_2|B) + \mathbf{P}(A_3|B) = 0.995 + 0.003 + 0.002 = 1$.

Example 29 (Urn with red and black balls) A well-mixed urn contains five red and ten black balls. We draw two balls from the urn without replacement. What is the probability that the second ball drawn is red?

This is easy to see if we draw a probability tree diagram. The first split in the tree is based on the outcome of the first draw and the second on the outcome of the last draw. The outcome of the first draw dictates the probabilities for the second one since we are sampling without replacement. We multiply the probabilities on the edges to get probabilities of the four endpoints, and then sum the ones that correspond to red in the second draw, that is

$$P(\text{second ball is red}) = 4/42 + 10/42 = 1/3 .$$



Alternatively, use the total probability theorem to break the problem down into manageable pieces. Let $R_1 = \{(\text{red}, \text{red}), (\text{red}, \text{black})\}$ and $R_2 = \{(\text{red}, \text{red}), (\text{black}, \text{red})\}$ be the events corresponding to a red ball in the 1st and 2nd draws, respectively, and let $B_1 = \{(\text{black}, \text{red}), (\text{black}, \text{black})\}$ be the event of a black ball on the first draw.

Now R_1 and B_1 partition Ω so we can write:

$$\begin{aligned} P(R_2) &= \mathbf{P}(R_2 \cap R_1) + \mathbf{P}(R_2 \cap B_1) \\ &= \mathbf{P}(R_2|R_1)\mathbf{P}(R_1) + \mathbf{P}(R_2|B_1)\mathbf{P}(B_1) \\ &= (4/14)(1/3) + (5/14)(2/3) = 1/3 . \end{aligned}$$

2.3.1 Independence and Dependence

Definition 14 (Independence of two events) Any two events A and B are said to be **independent** if and only if

$$\mathbf{P}(A \cap B) = \mathbf{P}(A)\mathbf{P}(B) . \quad (2.5)$$

Let us make sense of this definition in terms of our previous definitions. When $\mathbf{P}(A) = 0$ or $\mathbf{P}(B) = 0$, both sides of the above equality are 0. If $\mathbf{P}(A) \neq 0$, then rearranging the above equation we get:

$$\frac{\mathbf{P}(A \cap B)}{\mathbf{P}(A)} = \mathbf{P}(B) .$$

But, the LHS is $\mathbf{P}(B|A)$ by definition 2.2, and thus for independent events A and B , we get:

$$\mathbf{P}(B|A) = \mathbf{P}(B) .$$

This says that information about the occurrence of A does not affect the occurrence of B . If $\mathbf{P}(B) \neq 0$, then an analogous argument:

$$\mathbf{P}(A \cap B) = \mathbf{P}(A)\mathbf{P}(B) \iff \mathbf{P}(B \cap A) = \mathbf{P}(A)\mathbf{P}(B) \iff \frac{\mathbf{P}(B \cap A)}{\mathbf{P}(B)} = \mathbf{P}(A) \iff \mathbf{P}(A|B) = \mathbf{P}(A) ,$$

says that information about the occurrence of B does not affect the occurrence of A . Therefore, the probability of their joint occurrence $\mathbf{P}(A \cap B)$ is simply the product of their individual probabilities $\mathbf{P}(A)\mathbf{P}(B)$.

Definition 15 (Independence of a sequence of events) We say that a finite or infinite sequence of events A_1, A_2, \dots are independent if whenever i_1, i_2, \dots, i_k are distinct elements from the set of indices \mathbb{N} , such that $A_{i_1}, A_{i_2}, \dots, A_{i_k}$ are defined (elements of \mathcal{F}), then

$$\mathbf{P}(A_{i_1} \cap A_{i_2} \dots \cap A_{i_k}) = \mathbf{P}(A_{i_1})\mathbf{P}(A_{i_2}) \cdots \mathbf{P}(A_{i_k})$$

Example 30 (Some Standard Examples) A sequence of events in a sequence of independent trials is independent.

(a) Suppose you toss a fair coin twice such that the first toss is independent of the second. Then,

$$\mathbf{P}(\text{Heads on the first toss} \cap \text{Tails on the second toss}) = \mathbf{P}(\mathsf{H})\mathbf{P}(\mathsf{T}) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} .$$

(b) Suppose you independently toss a fair die three times. Let E_i be the event that the outcome is an even number on the i -th trial. The probability of getting an even number in all three trials is:

$$\begin{aligned} \mathbf{P}(E_1 \cap E_2 \cap E_3) &= \mathbf{P}(E_1)\mathbf{P}(E_2)\mathbf{P}(E_3) \\ &= (\mathbf{P}(\{2, 4, 6\}))^3 \\ &= (\mathbf{P}(\{2\}) \cup \{4\} \cup \{6\}))^3 \\ &= (\mathbf{P}(\{2\}) + \mathbf{P}(\{4\}) + \mathbf{P}(\{6\}))^3 \\ &= \left(\frac{1}{6} + \frac{1}{6} + \frac{1}{6}\right)^3 = \left(\frac{1}{2}\right)^3 = \frac{1}{8} . \end{aligned}$$

(c) Suppose you toss a fair coin independently m times. Then each of the 2^m possible outcomes in the sample space Ω has equal probability of $\frac{1}{2^m}$ due to independence.

Example 31 (dependence and independence) Suppose we toss two fair dice. Let A denote the event that the sum of the dice is six and B denote the event that the first die equals four. The sample space encoding the thirty six ordered pairs of outcomes for the two dice is $\Omega =$

$\{(1,1), (1,2), \dots, (1,6), (2,1), \dots, (2,6), \dots, (5,6), (6,6)\}$ and due to independence $\mathbf{P}(\omega) = 1/36$ for each $\omega \in \Omega$. Then

$$\mathbf{P}(A \cap B) = \mathbf{P}(\{(4,2)\}) = \frac{1}{36},$$

but

$$\begin{aligned}\mathbf{P}(A)\mathbf{P}(B) &= \mathbf{P}(\{(1,5), (2,4), (3,3), (4,2), (5,1)\})\mathbf{P}(\{(4,1), (4,2), (4,3), (4,4), (4,5), (4,6)\}) \\ &= \frac{5}{36} \times \frac{6}{36} = \frac{5}{36} \times \frac{1}{6} = \frac{5}{216},\end{aligned}$$

and therefore A and B are not independent. The reason for the events A and B being dependent is clear because the chance of getting a total of six depends on the outcome of the first die (not being six).

Now, let C be the event that the sum of the two dice equals seven. Then

$$\mathbf{P}(C \cap B) = \mathbf{P}(\{(4,3)\}) = \frac{1}{36},$$

while

$$\begin{aligned}\mathbf{P}(C \cap B) &= \mathbf{P}(\{(1,6), (2,5), (3,4), (4,3), (5,2), (6,1)\})\mathbf{P}(\{(4,1), (4,2), (4,3), (4,4), (4,5), (4,6)\}) \\ &= \frac{6}{36} \times \frac{6}{36} = \frac{1}{36},\end{aligned}$$

and therefore C and B are independent events. Once again this is clear because the chance of getting a total of seven does not depend any more on the outcome of the first die (it is allowed to be any one of the six possible outcomes).

Example 32 (Pairwise independent events that are not jointly independent) Let a ball be drawn from an well-stirred urn containing four balls labelled 1,2,3,4. Consider the events $A = \{1,2\}$, $B = \{1,3\}$ and $C = \{1,4\}$. Then,

$$\begin{aligned}\mathbf{P}(A \cap B) &= \mathbf{P}(A)\mathbf{P}(B) = \frac{2}{4} \times \frac{2}{4} = \frac{1}{4}, \\ \mathbf{P}(A \cap C) &= \mathbf{P}(A)\mathbf{P}(C) = \frac{2}{4} \times \frac{2}{4} = \frac{1}{4}, \\ \mathbf{P}(B \cap C) &= \mathbf{P}(B)\mathbf{P}(C) = \frac{2}{4} \times \frac{2}{4} = \frac{1}{4},\end{aligned}$$

but,

$$\frac{1}{4} = \mathbf{P}(\{1\}) = \mathbf{P}(A \cap B \cap C) \neq \mathbf{P}(A)\mathbf{P}(B)\mathbf{P}(C) = \frac{2}{4} \times \frac{2}{4} \times \frac{2}{4} = \frac{1}{8}.$$

Therefore, inspite of being pairwise independent, the events A , B and C are not jointly independent.

Exercises

Ex. 2.5 — What gives the greater probability of hitting some target at least once:

- 1.hitting in a shot with probability $\frac{1}{2}$ and firing 1 shot, or
- 2.hitting in a shot with probability $\frac{1}{3}$ and firing 2 shots?

First guess. Then calculate.

Ex. 2.6 — Suppose we independently roll two fair dice each of whose faces are marked by numbers 1,2,3,4, 5 and 6.

- 1.List the sample space for the experiment if we note the numbers on the 2 upturned faces.
- 2.What is the probability of obtaining a sum greater than 4 but less than 7?

Ex. 2.7 — Based on past experience, 70% of students in a certain course pass the midterm test. The final exam is passed by 80% of those who passed the midterm test, but only by 40% of those who fail the midterm test. What fraction of students pass the final exam?

Ex. 2.8 — A small brewery has two bottling machines. Machine 1 produces 75% of the bottles and machine 2 produces 25%. One out of every 20 bottles filled by machine 1 is rejected for some reason, while one out of every 30 bottles filled by machine 2 is rejected. What is the probability that a randomly selected bottle comes from machine 1 given that it is accepted?

Ex. 2.9 — A process producing micro-chips, produces 5% defective, at random. Each micro-chip is tested, and the test will correctly detect a defective one $4/5$ of the time, and if a good micro-chip is tested the test will declare it is defective with probability $1/10$.

- (a)If a micro-chip is chosen at random, and tested to be good, what was the probability that it was defective anyway?
- (b)If a micro-chip is chosen at random, and tested to be defective, what was the probability that it was good anyway?
- (c)If 2 micro-chips are tested and determined to be good, what is the probability that at least one is in fact defective?

Ex. 2.10 — Suppose that $\frac{2}{3}$ of all gales are force 1, $\frac{1}{4}$ are force 2 and $\frac{1}{12}$ are force 3. Furthermore, the probability that force 1 gales cause damage is $\frac{1}{4}$, the probability that force 2 gales cause damage is $\frac{2}{3}$ and the probability that force 3 gales cause damage is $\frac{5}{6}$.

- (a)If a gale is reported, what is the probability of it causing damage?
- (b)If the gale caused damage, find the probabilities that it was of: force 1; force 2; force 3.
- (c)If the gale did NOT cause damage, find the probabilities that it was of: force 1; force 2; force 3.

Ex. 2.11 — **The sensitivity and specificity of a medical diagnostic test for a disease are defined as follows:

$$\begin{aligned}\text{sensitivity} &= \mathbf{P}(\text{test is positive} \mid \text{patient has the disease}) , \\ \text{specificity} &= \mathbf{P}(\text{test is negative} \mid \text{patient does not have the disease}) .\end{aligned}$$

Suppose that a medical test has a sensitivity of 0.7 and a specificity of 0.95. If the prevalence of the disease in the general population is 1%, find

- (a)the probability that a patient who tests positive actually has the disease,
- (b)the probability that a patient who tests negative is free from the disease.

Ex. 2.12 — **The detection rate and false alarm rate of an intrusion sensor are defined as

$$\begin{aligned}\text{detection rate} &= \mathbf{P}(\text{detection declared} \mid \text{intrusion}) , \\ \text{false alarm rate} &= \mathbf{P}(\text{detection declared} \mid \text{no intrusion}) .\end{aligned}$$

If the detection rate is 0.999 and the false alarm rate is 0.001, and the probability of an intrusion occurring is 0.01, find

- (a)the probability that there is an intrusion when a detection is declared,
- (b)the probability that there is no intrusion when no detection is declared.

Ex. 2.13 — **Let A and B be events such that $\mathbf{P}(A) \neq 0$ and $\mathbf{P}(B) \neq 0$. When A and B are disjoint, are they also independent? Explain clearly why or why not.

CONDITIONAL PROBABILITY SUMMARY

$\mathbf{P}(A|B)$ means the probability that A occurs given that B has occurred.

$$\mathbf{P}(A|B) = \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(B)} = \frac{\mathbf{P}(A)\mathbf{P}(B|A)}{\mathbf{P}(B)} \quad \text{if } \mathbf{P}(B) \neq 0$$

$$\mathbf{P}(B|A) = \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(A)} = \frac{\mathbf{P}(B)\mathbf{P}(A|B)}{\mathbf{P}(A)} \quad \text{if } \mathbf{P}(A) \neq 0$$

Conditional probabilities obey the axioms and rules of probability.

Chapter 3

Random Variables

It can be inconvenient to work with a set of outcomes Ω upon which arithmetic is not possible. We are often measuring our outcomes with subsets of real numbers. Some examples include:

Experiment	Possible measured outcomes
Counting the number of typos up to now	$\mathbb{Z}_+ := \{0, 1, 2, \dots\} \subset \mathbb{R}$
Length in centi-meters of some shells on New Brighton beach	$(0, +\infty) \subset \mathbb{R}$
Waiting time in minutes for the next Orbiter bus to arrive	$\mathbb{R}_+ := [0, \infty) \subset \mathbb{R}$
Vertical displacement from current position of a pollen on water	\mathbb{R}

3.1 Basic Definitions

To take advantage of our measurements over the real numbers, in terms of its metric structure and arithmetic, we need to formally define this measurement process using the notion of a random variable.

Definition 16 (Random Variable) Let (Ω, \mathcal{F}, P) be some probability triple. Then, a **Random Variable (RV)**, say X , is a function from the sample space Ω to the set of real numbers \mathbb{R}

$$X : \Omega \rightarrow \mathbb{R}$$

such that for every $x \in \mathbb{R}$, the inverse image of the half-open real interval $(-\infty, x]$ is an element of the collection of events \mathcal{F} , i.e.:

$$\text{for every } x \in \mathbb{R}, \quad X^{[-1]}((-\infty, x]) := \{\omega : X(\omega) \leq x\} \in \mathcal{F}.$$

This definition can be summarised by the statement that a RV is an \mathcal{F} -measurable map. We assign probability to the RV X as follows:

$$\mathbf{P}(X \leq x) = \mathbf{P}(X^{[-1]}((-\infty, x])) := \mathbf{P}(\{\omega : X(\omega) \leq x\}). \quad (3.1)$$

Definition 17 (Distribution Function) The **Distribution Function (DF)** or **Cumulative Distribution Function (CDF)** of any RV X , over a probability triple (Ω, \mathcal{F}, P) , denoted by F is:

$$F(x) := \mathbf{P}(X \leq x) = \mathbf{P}(\{\omega : X(\omega) \leq x\}), \quad \text{for any } x \in \mathbb{R}. \quad (3.2)$$

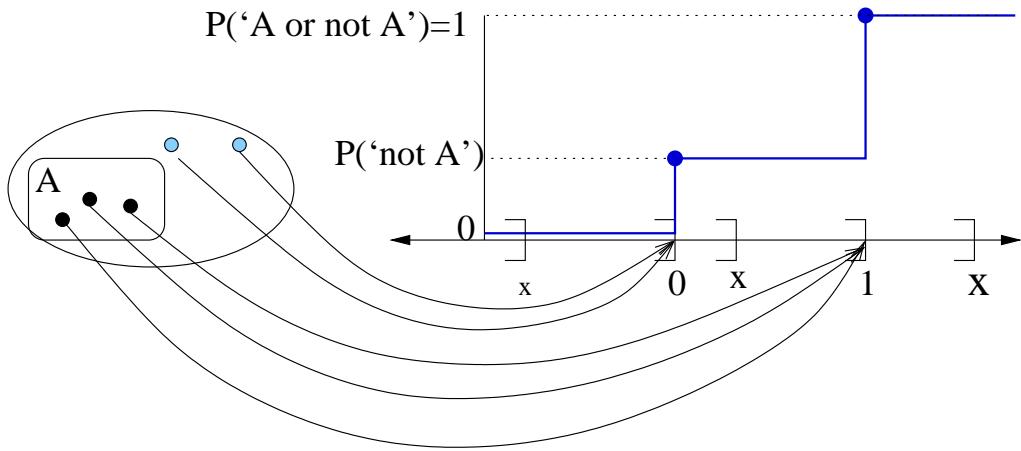
Thus, $F(x)$ or simply F is a non-decreasing, right continuous, $[0, 1]$ -valued function over \mathbb{R} . When a RV X has DF F we write $X \sim F$.

A special RV that often plays the role of ‘building-block’ in Probability and Statistics is the indicator function of an event A that tells us whether the event A has occurred or not. Recall that an event belongs to the collection of possible events \mathcal{F} for our experiment.

Definition 18 (Indicator Function) The **Indicator Function** of an event A denoted $\mathbb{1}_A$ is defined as follows:

$$\mathbb{1}_A(\omega) := \begin{cases} 1 & \text{if } \omega \in A \\ 0 & \text{if } \omega \notin A \end{cases} \quad (3.3)$$

Figure 3.1: The Indicator function of event $A \in \mathcal{F}$ is a RV $\mathbb{1}_A$ with DF F



Classwork 33 (Indicator function is a random variable) Let us convince ourselves that $\mathbb{1}_A$ is really a RV. For $\mathbb{1}_A$ to be a RV, we need to verify that for any real number $x \in \mathbb{R}$, the inverse image $\mathbb{1}_A^{[-1]}((-\infty, x])$ is an event, ie :

$$\mathbb{1}_A^{[-1]}((-\infty, x]) := \{\omega : \mathbb{1}_A(\omega) \leq x\} \in \mathcal{F} .$$

All we can assume about the collection of events \mathcal{F} is that it contains the event A and that it is a sigma algebra. A careful look at the Figure 3.1 yields:

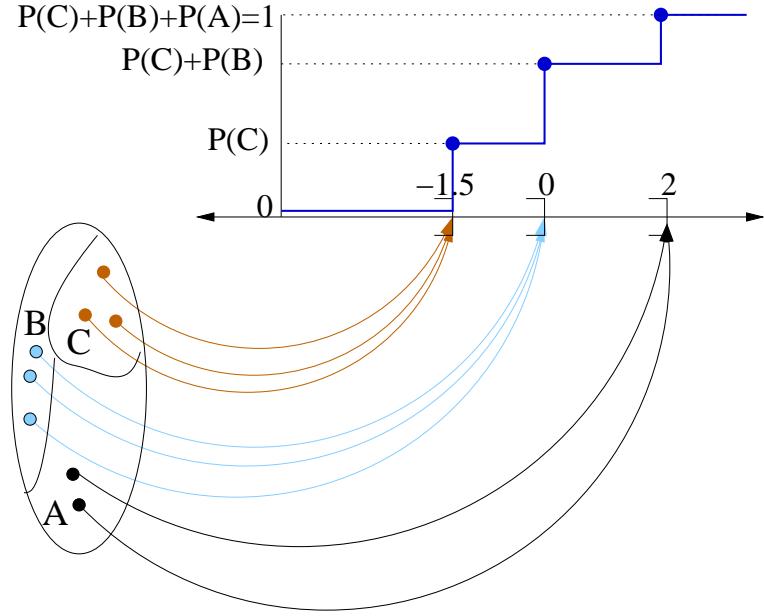
$$\mathbb{1}_A^{[-1]}((-\infty, x]) := \{\omega : \mathbb{1}_A(\omega) \leq x\} = \begin{cases} \emptyset & \text{if } x < 0 \\ A^c & \text{if } 0 \leq x < 1 \\ A \cup A^c = \Omega & \text{if } 1 \leq x \end{cases}$$

Thus, $\mathbb{1}_A^{[-1]}((-\infty, x])$ is one of the following three sets that belong to \mathcal{F} ; (1) \emptyset , (2) A^c and (3) Ω depending on the value taken by x relative to the interval $[0, 1]$. We have proved that $\mathbb{1}_A$ is indeed a RV.

Some useful properties of the Indicator Function are:

$$\mathbb{1}_{A^c} = 1 - \mathbb{1}_A, \quad \mathbb{1}_{A \cap B} = \mathbb{1}_A \mathbb{1}_B, \quad \mathbb{1}_{A \cup B} = \mathbb{1}_A + \mathbb{1}_B - \mathbb{1}_A \mathbb{1}_B$$

We slightly abuse notation when A is a single element set by ignoring the curly braces.

Figure 3.2: A RV X from a sample space Ω with 8 elements to \mathbb{R} and its DF F 

Classwork 34 (A random variable with three values and eight sample points) Consider the RV X of Figure 3.2. Let the events $A = \{\omega_1, \omega_2\}$, $B = \{\omega_3, \omega_4, \omega_5\}$ and $C = \{\omega_6, \omega_7, \omega_8\}$. Define the RV X formally. What sets should \mathcal{F} minimally include? What do you need to do to make sure that \mathcal{F} is a sigma algebra?

3.2 An Elementary Discrete Random Variable

When a RV takes at most countably many values from a discrete set $\mathbb{D} \subset \mathbb{R}$, we call it a **discrete** RV. Often, \mathbb{D} is the set of integers \mathbb{Z} .

Definition 19 (probability mass function (PMF)) Let X be a discrete RV over a probability triple (Ω, \mathcal{F}, P) . We define the **probability mass function** (PMF) f of X to be the function $f : \mathbb{D} \rightarrow [0, 1]$ defined as follows:

$$f(x) := \mathbf{P}(X = x) = \mathbf{P}(\{\omega : X(\omega) = x\}), \quad \text{where } x \in \mathbb{D}.$$

The DF F and PMF f for a discrete RV X satisfy the following:

1. For any $x \in \mathbb{R}$,

$$\mathbf{P}(X \leq x) = F(x) = \sum_{\mathbb{D} \ni y \leq x} f(y) := \sum_{y \in \mathbb{D} \cap (-\infty, x]} f(y).$$

2. For any $a, b \in \mathbb{D}$ with $a < b$,

$$\mathbf{P}(a < X \leq b) = F(b) - F(a) = \sum_{y \in \mathbb{D} \cap (a, b]} f(y) .$$

In particular, when $\mathbb{D} = \mathbb{Z}$ and $a = b - 1$,

$$\mathbf{P}(b - 1 < X \leq b) = F(b) - F(b - 1) = f(b) = \mathbf{P}(\{\omega : X(\omega) = b\}) .$$

3. And of course

$$\sum_{x \in \mathbb{D}} f(x) = 1$$

The Indicator Function $\mathbf{1}_A$ of the event that ‘ A occurs’ for the θ -specific experiment \mathcal{E} over some probability triple $(\Omega, \mathcal{F}, \mathbf{P}_\theta)$, with $A \in \mathcal{F}$, is the Bernoulli(θ) RV. The parameter θ denotes the probability that ‘ A occurs’ (see Figure 3.3 when A is the event that ‘ H occurs’). This is our first example of a discrete RV.

Model 2 (Bernoulli(θ)) Given a parameter $\theta \in [0, 1]$, the probability mass function (PMF) for the Bernoulli(θ) RV X is:

$$f(x; \theta) = \theta^x (1 - \theta)^{1-x} \mathbf{1}_{\{0,1\}}(x) = \begin{cases} \theta & \text{if } x = 1, \\ 1 - \theta & \text{if } x = 0, \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

and its DF is:

$$F(x; \theta) = \begin{cases} 1 & \text{if } 1 \leq x, \\ 1 - \theta & \text{if } 0 \leq x < 1, \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

We emphasise the dependence of the probabilities on the parameter θ by specifying it following the semicolon in the argument for f and F and by subscripting the probabilities, i.e. $\mathbf{P}_\theta(X = 1) = \theta$ and $\mathbf{P}_\theta(X = 0) = 1 - \theta$.

3.3 An Elementary Continuous Random Variable

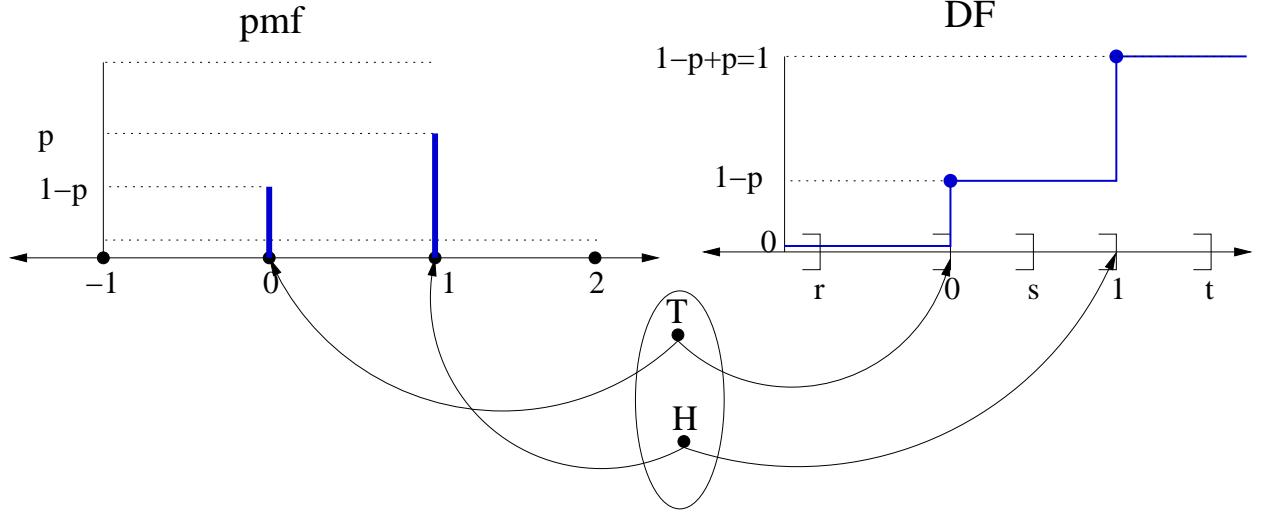
When a RV takes values in the continuum we call it a **continuous** RV. An example of such a RV is the vertical position (in micro meters) since the original release of a pollen grain on water. Another example of a continuous RV is the volume of water (in cubic meters) that fell on the southern Alps last year.

Definition 20 (probability density function (PDF)) A RV X is said to be ‘continuous’ if there exists a piecewise-continuous function f , called the probability density function (PDF) of X , such that for any $a, b \in \mathbb{R}$ with $a < b$,

$$\mathbf{P}(a < X \leq b) = F(b) - F(a) = \int_a^b f(x) dx .$$

The following hold for a continuous RV X with PDF f :

Figure 3.3: The Indicator Function $\mathbb{1}_H$ of the event ‘Heads occurs’, for the experiment ‘Toss 1 times,’ \mathcal{E}_θ^1 , as the RV X from the sample space $\Omega = \{H, T\}$ to \mathbb{R} and its DF F . The probability that ‘Heads occurs’ and that ‘Tails occurs’ are $f(1; \theta) = \mathbf{P}_\theta(X = 1) = \mathbf{P}_\theta(H) = \theta$ and $f(0; \theta) = \mathbf{P}_\theta(X = 0) = \mathbf{P}_\theta(T) = 1 - \theta$, respectively.



1. For any $x \in \mathbb{R}$, $\mathbf{P}(X = x) = 0$.
2. Consequentially, for any $a, b \in \mathbb{R}$ with $a \leq b$,

$$\mathbf{P}(a < X < b) = \mathbf{P}(a < X \leq b) = \mathbf{P}(a \leq X \leq b) = \mathbf{P}(a \leq X < b) .$$

3. By the fundamental theorem of calculus, except possibly at finitely many points (where the continuous pieces come together in the piecewise-continuous f):

$$f(x) = \frac{d}{dx} F(x)$$

4. And of course f must satisfy:

$$\int_{-\infty}^{\infty} f(x) dx = \mathbf{P}(-\infty < X < \infty) = 1 .$$

An elementary and fundamental example of a continuous RV is the Uniform(0, 1) RV of Model 3. It forms the foundation for random variate generation and simulation. In fact, it is appropriate to call this the fundamental model since every other experiment can be obtained from this one.

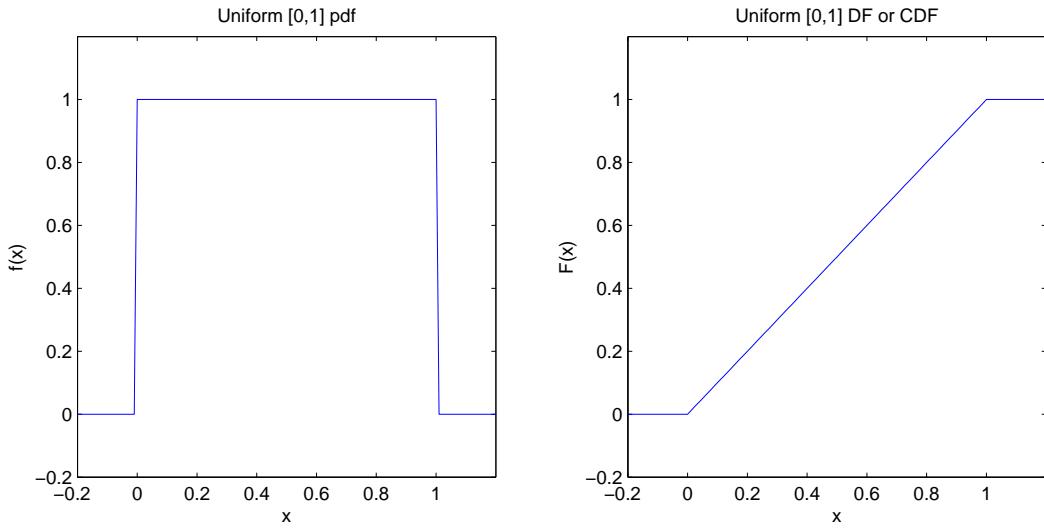
Model 3 (The Fundamental Model) The probability density function (PDF) of the fundamental model or the Uniform(0, 1) RV is

$$f(x) = \mathbb{1}_{[0,1]}(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1, \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

and its distribution function (DF) or cumulative distribution function (CDF) is:

$$F(x) := \int_{-\infty}^x f(y) dy = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } 0 \leq x \leq 1, \\ 1 & \text{if } x > 1 \end{cases} \quad (3.7)$$

Note that the DF is the identity map in $[0, 1]$. The PDF and DF are depicted in Figure 3.4.

Figure 3.4: A plot of the PDF and DF or CDF of the Uniform(0, 1) continuous RV X .****tossing a fair coin infinitely often and the fundamental model**

- The fundamental model is equivalent to infinite tosses of a fair coin (see using binary expansion of any $x \in (0, 1)$)
- The fundamental model has infinitely many copies of itself within it!

****universality of the fundamental model**

- one can obtain any other random object from the fundamental model!

3.4 Expectations

It is convenient to summarise a RV by a single number. This single number can be made to represent some average or expected feature of the RV via an integral with respect to the density of the RV.

Definition 21 (Expectation of a RV) The **expectation**, or **expected value**, or **mean**, or **first moment**, of a random variable X , with distribution function F and density f , is defined to be

$$\mathbf{E}(X) := \int x dF(x) = \begin{cases} \sum_x x f(x) & \text{if } X \text{ is discrete} \\ \int x f(x) dx & \text{if } X \text{ is continuous,} \end{cases} \quad (3.8)$$

provided the sum or integral is well-defined. We say the expectation exists if

$$\int |x| dF(x) < \infty. \quad (3.9)$$

Sometimes, we denote $\mathbf{E}(X)$ by $\mathbf{E}X$ for brevity. Thus, the expectation is a single-number summary of the RV X and may be thought of as the average. We subscript E to specify the parameter $\theta \in \Theta$ with respect to which the integration is undertaken.

$$\mathbf{E}_\theta X := \int x dF(x; \theta)$$

Definition 22 (Variance of a RV) Let X be a RV with mean or expectation $\mathbf{E}(X)$. Variance of X denoted by $\mathbf{V}(X)$ or VX is

$$\mathbf{V}(X) := \mathbf{E}((X - \mathbf{E}(X))^2) = \int (x - \mathbf{E}(X))^2 dF(x),$$

provided this expectation exists. The **standard deviation** denoted by $\text{sd}(X) := \sqrt{\mathbf{V}(X)}$. Thus variance is a measure of “spread” of a distribution.

Definition 23 (k -th moment of a RV) We call

$$\mathbf{E}(X^k) = \int x^k dF(x)$$

as the k -th moment of the RV X and say that the k -th moment exists when $\mathbf{E}(|X|^k) < \infty$. We call the following expectation as the k -th central moment:

$$\mathbf{E}((X - \mathbf{E}(X))^k).$$

Properties of Expectations

1. If the k -th moment exists and if $j < k$ then the j -th moment exists.
2. If X_1, X_2, \dots, X_n are RVs and a_1, a_2, \dots, a_n are constants, then

$$\mathbf{E}\left(\sum_{i=1}^n a_i X_i\right) = \sum_{i=1}^n a_i \mathbf{E}(X_i). \quad (3.10)$$

3. Let X_1, X_2, \dots, X_n be independent RVs, then

$$\mathbf{E} \left(\prod_{i=1}^n X_i \right) = \prod_{i=1}^n \mathbf{E}(X_i) . \quad (3.11)$$

4. $\mathbf{V}(X) = \mathbf{E}(X^2) - (\mathbf{E}(X))^2$. [prove by completing the square and applying (3.10)]

5. If a and b are constants then:

$$\mathbf{V}(aX + b) = a^2 \mathbf{V}(X) . \quad (3.12)$$

6. If X_1, X_2, \dots, X_n are independent and a_1, a_2, \dots, a_n are constants, then:

$$\mathbf{V} \left(\sum_{i=1}^n a_i X_i \right) = \sum_{i=1}^n a_i^2 \mathbf{V}(X_i) . \quad (3.13)$$

Mean and variance of Bernoulli(θ) RV: Let $X \sim \text{Bernoulli}(\theta)$. Then,

$$\begin{aligned} \mathbf{E}(X) &= \sum_{x=0}^1 xf(x) = (0 \times (1-\theta)) + (1 \times \theta) = 0 + \theta = \theta , \\ \mathbf{E}(X^2) &= \sum_{x=0}^1 x^2 f(x) = (0^2 \times (1-\theta)) + (1^2 \times \theta) = 0 + \theta = \theta , \\ \mathbf{V}(X) &= \mathbf{E}(X^2) - (\mathbf{E}(X))^2 = \theta - \theta^2 = \theta(1-\theta) . \end{aligned}$$

Parameter specifically,

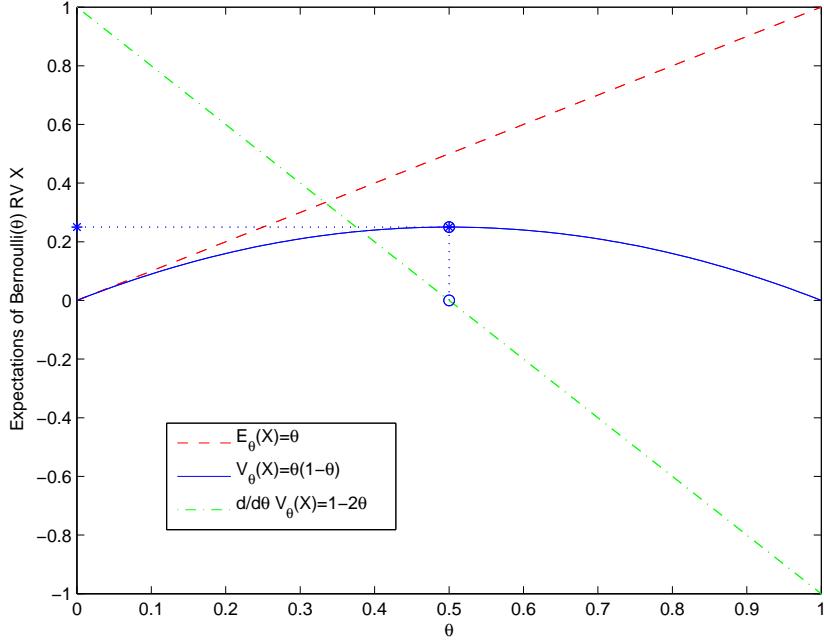
$$\mathbf{E}_\theta(X) = \theta \quad \text{and} \quad \mathbf{V}_\theta(X) = \theta(1-\theta) .$$

Maximum of the variance $\mathbf{V}_\theta(X)$ is found by setting the derivative to zero, solving for θ and showing the second derivative is locally negative, i.e. $\mathbf{V}_\theta(X)$ is concave down:

$$\begin{aligned} \mathbf{V}'_\theta(X) &:= \frac{d}{d\theta} \mathbf{V}_\theta(X) = 1 - 2\theta = 0 \iff \theta = \frac{1}{2} , & \mathbf{V}''_\theta(X) &:= \frac{d}{d\theta} \left(\frac{d}{d\theta} \mathbf{V}_\theta(X) \right) = -2 < 0 , \\ \max_{\theta \in [0,1]} \mathbf{V}_\theta(X) &= \frac{1}{2} \left(1 - \frac{1}{2} \right) = \frac{1}{4} , \text{ since } \mathbf{V}_\theta(X) \text{ is maximized at } \theta = \frac{1}{2} \end{aligned}$$

The plot depicting these expectations as well as the rate of change of the variance are depicted in Figure 3.5. Note from this Figure that $\mathbf{V}_\theta(X)$ attains its maximum value of $1/4$ at $\theta = 0.5$ where $\frac{d}{d\theta} \mathbf{V}_\theta(X) = 0$. Furthermore, we know that we don't have a minimum at $\theta = 0.5$ since the second derivative $\mathbf{V}''_\theta(X) = -2$ is negative for any $\theta \in [0, 1]$. This confirms that $\mathbf{V}_\theta(X)$ is concave down and therefore we have a maximum of $\mathbf{V}_\theta(X)$ at $\theta = 0.5$. We will revisit this example when we employ a numerical approach called Newton-Raphson method to solve for the maximum of a differentiable function by setting its derivative equal to zero.

Figure 3.5: Mean ($\mathbf{E}_\theta(X)$), variance ($\mathbf{V}_\theta(X)$) and the rate of change of variance ($\frac{d}{d\theta}\mathbf{V}_\theta(X)$) of a Bernoulli(θ) RV X as a function of the parameter θ .



Mean and variance of Uniform(0, 1) RV: Let $X \sim \text{Uniform}(0, 1)$. Then,

$$\mathbf{E}(X) = \int_{x=0}^1 x f(x) dx = \int_{x=0}^1 x 1 dx = \frac{1}{2} (x^2) \Big|_{x=0}^{x=1} = \frac{1}{2} (1 - 0) = \frac{1}{2},$$

$$\mathbf{E}(X^2) = \int_{x=0}^1 x^2 f(x) dx = \int_{x=0}^1 x^2 1 dx = \frac{1}{3} (x^3) \Big|_{x=0}^{x=1} = \frac{1}{3} (1 - 0) = \frac{1}{3},$$

$$\mathbf{V}(X) = \mathbf{E}(X^2) - (\mathbf{E}(X))^2 = \frac{1}{3} - \left(\frac{1}{2}\right)^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}.$$

Proposition 24 (Winnings on Average) Let $Y = r(X)$. Then

$$\mathbf{E}(Y) = \mathbf{E}(r(X)) = \int r(x) dF(x).$$

Think of playing a game where we draw $x \sim X$ and then I pay you $y = r(x)$. Then your average income is $r(x)$ times the chance that $X = x$, summed (or integrated) over all values of x .

Example 35 (Probability is an Expectation) Let A be an event and let $r(X) = \mathbb{1}_A(x)$. Recall $\mathbb{1}_A(x)$ is 1 if $x \in A$ and $\mathbb{1}_A(x) = 0$ if $x \notin A$. Then

$$\mathbf{E}(\mathbb{1}_A(X)) = \int \mathbb{1}_A(x) dF(x) = \int_A f(x) dx = \mathbf{P}(X \in A) = \mathbf{P}(A) \quad (3.14)$$

Thus, probability is a special case of expectation. Recall our LTRF motivation for the definition of probability and make the connection.

3.5 Stochastic Processes

Definition 25 (Independence of RVs) A finite or infinite sequence of RVs X_1, X_2, \dots is said to be independent or independently distributed if

$$\mathbf{P}(X_{i_1} \leq x_{i_1}, X_{i_2} \leq x_{i_2}, \dots, X_{i_k} \leq x_{i_k}) = \mathbf{P}(X_{i_1} \leq x_{i_1})\mathbf{P}(X_{i_2} \leq x_{i_2}) \cdots \mathbf{P}(X_{i_k} \leq x_{i_k})$$

for any distinct subset $\{i_1, i_2, \dots, i_l\}$ of indices of the sequence of RVs and any sequence of real numbers $x_{i_1}, x_{i_2}, \dots, x_{i_k}$.

By the above definition, the sequence of **discrete** RVs X_1, X_2, \dots taking values in an at most countable set \mathbb{D} are said to be independently distributed if for any distinct subset of indices $\{i_1, i_2, \dots, i_k\}$ such that the corresponding RVs $X_{i_1}, X_{i_2}, \dots, X_{i_k}$ exists as a distinct subset of our original sequence of RVs X_1, X_2, \dots and for any elements $x_{i_1}, x_{i_2}, \dots, x_{i_k}$ in \mathbb{D} , the following equality is satisfied:

$$\mathbf{P}(X_{i_1} = x_{i_1}, X_{i_2} = x_{i_2}, \dots, X_{i_k} = x_{i_k}) = \mathbf{P}(X_{i_1} = x_{i_1})\mathbf{P}(X_{i_2} = x_{i_2}) \cdots \mathbf{P}(X_{i_k} = x_{i_k})$$

For an independent sequence of RVs $\{X_1, X_2, \dots\}$, we have

$$\begin{aligned} & \mathbf{P}(X_{i+1} \leq x_{i+1} | X_i \leq x_i, X_{i-1} \leq x_{i-1}, \dots, X_1 \leq x_1) \\ &= \frac{\mathbf{P}(X_{i+1} \leq x_{i+1}, X_i \leq x_i, X_{i-1} \leq x_{i-1}, \dots, X_1 \leq x_1)}{\mathbf{P}(X_i \leq x_i, X_{i-1} \leq x_{i-1}, \dots, X_1 \leq x_1)} \\ &= \frac{\mathbf{P}(X_{i+1} \leq x_{i+1})\mathbf{P}(X_i \leq x_i)\mathbf{P}(X_{i-1} \leq x_{i-1}) \cdots \mathbf{P}(X_1 \leq x_1)}{\mathbf{P}(X_i \leq x_i)\mathbf{P}(X_{i-1} \leq x_{i-1}) \cdots \mathbf{P}(X_1 \leq x_1)} \\ &= \mathbf{P}(X_{i+1} \leq x_{i+1}) \end{aligned}$$

The above equality that

$$\mathbf{P}(X_{i+1} \leq x_{i+1} | X_i \leq x_i, X_{i-1} \leq x_{i-1}, \dots, X_1 \leq x_1) = \mathbf{P}(X_{i+1} \leq x_{i+1})$$

simply says that the conditional distribution of the RV X_{i+1} given all previous RVs X_i, X_{i-1}, \dots, X_1 is simply determined by the distribution of X_{i+1} .

When a sequence of RVs are not independent they are said to be **dependent**.

Definition 26 (Stochastic Process) A collection of RVs

$$(X_\alpha)_{\alpha \in N} := (\ X_\alpha : \alpha \in \mathbb{A} \)$$

is called a **stochastic process**. Thus, for every $\alpha \in \mathbb{A}$, the index set of the stochastic process, X_α is a RV. If the index set $\mathbb{A} \subset \mathbb{Z}$ then we have a **discrete time stochastic process**, typically denoted by

$$(X_i)_{i \in \mathbb{Z}} := \dots, X_{-2}, X_{-1}, X_0, X_1, X_2, \dots, \text{ or}$$

$$(X_i)_{i \in \mathbb{N}} := X_1, X_2, \dots, \text{ or}$$

$$(X_i)_{i \in [n]} := X_1, X_2, \dots, X_n, \text{ where, } [n] := \{1, 2, \dots, n\} .$$

If $\mathbb{A} \subset \mathbb{R}$ then we have a **continuous time stochastic process**, typically denoted by $\{X_t\}_{t \in \mathbb{R}}$, etc.

Definition 27 (Independent and Identically Distributed (IID)) The finite or infinite sequence of RVs or the stochastic process X_1, X_2, \dots is said to be independent and identically distributed or IID if :

- they are independently distributed according to Definition 25, and
- $F(X_1) = F(X_2) = \dots$, ie. all the X_i 's have the same DF $F(X_1)$.

This is perhaps the most elementary class of stochastic processes and we succinctly denote it by

$$(X_i)_{i \in [n]} := X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F, \quad \text{or} \quad (X_i)_{i \in \mathbb{N}} := X_1, X_2, \dots \stackrel{\text{IID}}{\sim} F.$$

We sometimes replace the DF F above by the name of the RV.

Definition 28 (Independently Distributed) The sequence of RVs or the stochastic process $(X_i)_{i \in \mathbb{N}} := X_1, X_2, \dots$ is said to be independently distributed if :

- X_1, X_2, \dots is independently distributed according to Definition 25.

This is a class of stochastic processes that is more general than the IID class.

Chapter 4

Random Numbers

“Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.” — John von Neumann (1951)

4.1 Physical Random Number Generators

Physical devices such as the BINGO machine demonstrated in class can be used to produce an integer uniformly at random from a finite set of possibilities. Such “ball bouncing machines” used in the British national lottery as well as the New Zealand LOTTO are complex nonlinear systems that are extremely sensitive to initial conditions (“chaotic” systems) and are physical approximations of the probability model called a “well-stirred urn” or an equi-probable de Moivre($1/k, \dots, 1/k$) random variable.

Let us look at the New Zealand LOTTO draws at <http://lotto.nzpages.co.nz/statistics.html> and convince ourselves that all fourty numbers $\{1, 2, \dots, 39, 40\}$ seem to be drawn uniformly at random. The British lottery animation at <http://understandinguncertainty.org/node/39> shows how often each of the 49 numbers came up in the first 1240 draws. Are these draws really random? We will answer these questions in the sequel (see <http://understandinguncertainty.org/node/40> if you can’t wait).

4.2 Pseudo-Random Number Generators

Our probability model and the elementary continuous Uniform(0, 1) RV are built from the abstract concept of a random variable over a probability triple. A direct implementation of these ideas on a computing machine is not possible. In practice, random variables are typically simulated by **deterministic** methods or algorithms. Such algorithms generate sequences of numbers whose behavior is virtually indistinguishable from that of truly random sequences. In computational statistics, simulating realisations from a given RV is usually done in two distinct steps. First, sequences of numbers that imitate independent and identically distributed (IID) Uniform(0, 1) RVs are generated. Second, appropriate transformations are made to these imitations of IID Uniform(0, 1) random variates in order to imitate IID random variates from other random variables or other random structures. These two steps are essentially independent and are studied by two non-overlapping groups of researchers. The formal term **pseudo-random number generator** (PRNG) or simply **random number generator** (RNG) usually refers to some deterministic algorithm used in the first step to produce pseudo-random numbers (PRNs) that imitate IID Uniform(0, 1) random variates.

In the following chapters, we focus on transforming IID Uniform(0, 1) variates to other non-uniform variates. In this chapter, we focus on the art of imitating IID Uniform(0, 1) variates using simple deterministic rules.

4.2.1 Linear Congruential Generators

The following procedure introduced by D. H. Lehmer in 1949 [*Proc. 2nd Symp. on Large-Scale Digital Calculating Machinery, Harvard Univ. Press, Cambridge, Mass., 1951, 141–146*] gives the simplest popular PRNG that can be useful in many statistical situations if used wisely.

Algorithm 1 Linear Congruential Generator (LCG)

1: *input:* five suitable integers:

1. m , the modulus; $0 < m$
2. a , the multiplier; $0 \leq a < m$
3. c , the increment; $0 \leq c < m$
4. x_0 , the seed; $0 \leq x_0 < m$
5. n , the number of desired pseudo-random numbers

2: *output:* $(x_0, x_1, \dots, x_{n-1})$, the linear congruential sequence of length n

3: **for** $i = 1$ to $n - 1$ **do**

4: $x_i \leftarrow (ax_{i-1} + c) \bmod m$

5: **end for**

6: *return:* (x_1, x_2, \dots, x_n)

In order to implement LCGs we need to be able to do high precision exact integer arithmetic in MATLAB. We employ the Module `vpi` to implement variable precision integer arithmetic. You need to download this module for the next Labwork.

Labwork 36 (Generic Linear Congruential Sequence) Let us implement Algorithm 1 in MATLAB as follows.

```

function x = LinConGen(m,a,c,x0,n)
% Returns the linear congruential sequence
% Needs variable precision integer arithmetic in MATLAB!!!
% Usage: x = LinConGen(m,a,c,x0,n)
% Tested:
% Knuth3.3.4Table1.Line1: LinConGen(100000001,23,0,01234,10)
% Knuth3.3.4Table1.Line5: LinConGen(256,137,0,01234,10)
% Knuth3.3.4Table1.Line20: LinConGen(2147483647,48271,0,0123456,10)
% Knuth3.3.4Table1.Line21: LinConGen(2147483399,40692,0,0123456,10)

x=zeros(1,n); % initialize an array of zeros
X=vpi(x0); % X is a variable precision integer seed
x(1) = double(X); % convert to double
A=vpi(a); M=vpi(m); C=vpi(c); % A,M,C as variable precision integers
for i = 2:n % loop to generate the Linear congruential sequence
    % the linear congruential operation in variable precision integer
    % arithmetic
    % comment out the next ';' to get integer output
    X=mod(A * X + C, M);

```

```
x(i) = double(X); % convert to double
end
```

We can call it for some arbitrary input arguments as follows:

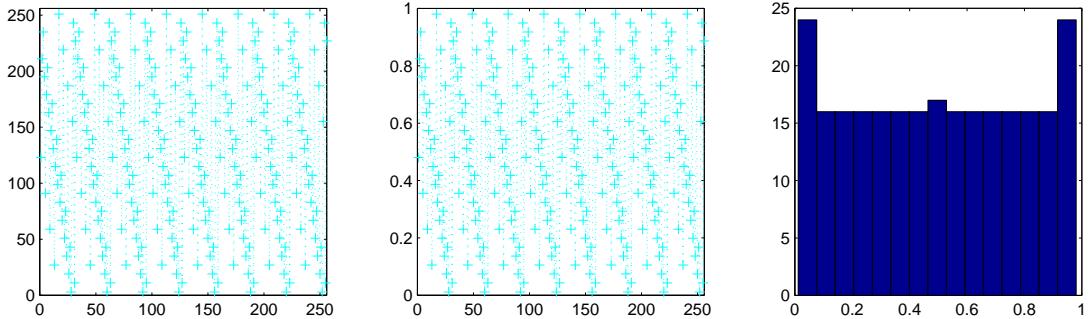
```
>> LinConGen(13,12,11,10,12)
ans =
    10     1     10     1     10     1     10     1     10     1     10     1
>> LinConGen(13,10,9,8,12)
ans =
     8     11     2     3     0     9     8     11     2     3     0     9
```

and observe that the generated sequences are not “random” for input values of (m, a, c, x_0, n) equalling $(13, 12, 11, 10, 12)$ or $(13, 10, 9, 8, 12)$. Thus, we need to do some work to determine the *suitable* input integers (m, a, c, x_0, n) .

Labwork 37 (LCG with period length of 32) Consider the linear congruential sequence with $(m, a, c, x_0, n) = (256, 137, 0, 123, 257)$ with period length of only $32 < m = 256$. We can visualise the sequence as plots in Figure 4.1 after calling the following M-file.

```
LinConGenKnuth334T1L5Plots.m
LCGSeq=LinConGen(256,137,0,123,257)
subplot(1,3,1)
plot(LCGSeq,'+')
axis([0 256 0 256]); axis square
LCGSeqIn01=LCGSeq ./ 256
subplot(1,3,2)
plot(LCGSeqIn01,'+')
axis([0 256 0 1]); axis square
subplot(1,3,3)
hist(LCGSeqIn01,15)
axis square
```

Figure 4.1: The linear congruential sequence of $\text{LinConGen}(256, 137, 0, 123, 257)$ with non-maximal period length of 32 as a line plot over $\{0, 1, \dots, 256\}$, scaled over $[0, 1]$ by a division by 256 and a histogram of the 256 points in $[0, 1]$ with 15 bins.



Choosing the *suitable* magic input (m, a, c, x_0, n)

The linear congruential generator is a special case of a *discrete dynamical system*:

$$x_i = f(x_{i-1}), \quad f : \{0, 1, 2, \dots, m-1\} \rightarrow \{0, 1, 2, \dots, m-1\} \text{ and } f(x_{i-1}) = (ax_{i-1} + c) \pmod{m}.$$

Since f maps a the finite set $\{1, 2, \dots, m-1\}$ into itself, such systems are bound to have a repeating cycle of numbers called the **period**. In Labwork 36, the generator `LinConGen(13,12,11,10,12)` has period $(10, 1)$ of length 2, the generator `LinConGen(13,10,9,8,12)` has period $(8, 11, 2, 3, 0, 9)$ of length 6 and the generator `LinConGen(256,137,0,123,257)` has a period of length 32. All these generators have a non-maximal period length less than their modulus m . A good generator should have a maximal period of m . Let us try to implement a generator with a maximal period of $m = 256$.

The period of a general LCG is at most m , and for some choices of a the period can be much less than m as shown in the examples considered earlier. The LCG will have a full period if and only if:

1. c and m are relatively prime,
2. $a - 1$ is divisible by all prime factors of m ,
3. $a - 1$ is a multiple of 4 if m is a multiple of 4

Labwork 38 (LCG with maximal period length of 256) Consider the linear congruential sequence with $(m, a, c, x_0, n) = (256, 137, 123, 13, 256)$. First check that these parameters do indeed satisfy the three condition above and therefore can produce the maximal period length of only $m = 256$. Modify the input parameter to `LinConGen` and repeat Labwork 37 in order to first produce a sequence of length 257. Do you see that the period is of maximal length of 256 as opposed to the generator of Labwork 37? Next produce a Figure to visualise the sequence as done in Figure 4.1.

A useful sequence should clearly have a relatively long period, say at least 2^{30} . Therefore, the **modulus m has to be rather large** because the **period** cannot have more than m elements. Moreover, the quality of pseudo-random numbers of a LCG is extremely sensitive to the choice of m , a and c even if the maximal period is attained. The next example illustrates this point.

Labwork 39 (The infamous RANDU) RANDU is an infamous LCG, which has been used since the 1960s. It is widely considered to be one of the most ill-conceived random number generators designed. Notably, it fails the **spectral test** badly for dimensions greater than 2. The following commands help visualise the sequence of first 5001/3 triplets (x_i, x_{i+1}, x_{i+2}) seeded from $x_0 = 1$ (Figure 4.2). Read `help reshape` and `help plot3`.

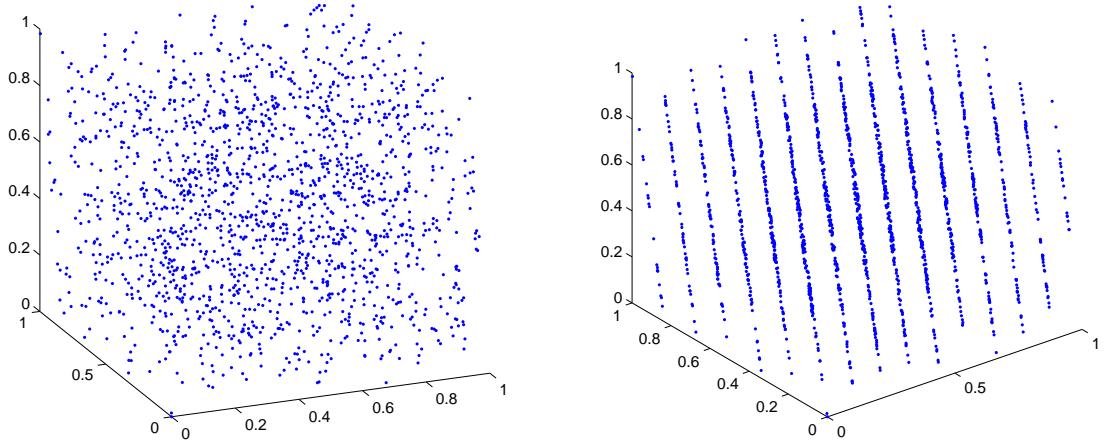
```
>> x=reshape( (LinConGen(2147483648,65539,0,1,5001)./ 2147483648) ,3,[]);
>> plot3(x(1,:),x(2,:),x(3,:),'.'
```

Labwork 40 (Fishman20 and Lecuyer21 LCGs) The following two LCGs are recommended in Knuth's Art of Computer Programming, vol. 2, for generating pseudo-random numbers for simple simulation tasks.

```
>> LinConGen(2147483647,48271,0,08787458,10) ./ 2147483647
ans =    0.0041    0.5239    0.0755    0.7624    0.6496    0.0769    0.9030    0.4259    0.9948    0.8868

>> LinConGen(2147483399,40692,0,01234567,10) ./ 2147483399
ans =    0.0006    0.3934    0.4117    0.7893    0.3913    0.6942    0.6790    0.3337    0.2192    0.1883
```

Figure 4.2: The LCG called RANDU with $(m, a, c) = (2147483648, 65539, 0)$ has strong correlation between three consecutive points as: $x_{i+2} = 6x_{k+1} - 9x_k$. The two plots are showing (x_i, x_{i+1}, x_{i+2}) from two different view points. .



The number of random numbers n should at most be about $m/1000$ in order to avoid the future sequence from behaving like the past. Thus, if $m = 2^{32}$ then a new generator, with a new suitable set of (m, a, c, x_0, n) should be adopted after the consumption of every few million pseudo-random numbers.

The LCGs are the least sophisticated type of PRNGs. They are easier to understand but are not recommended for intensive simulation purposes. The next section briefly introduces a more sophisticated PRNG we will be using in this course. Moreover our implementation of LCGs using the variable precision integer package is extremely slow in MATLAB and is only of pedagogical interest.

4.2.2 Generalized Feedback Shift Register and the “Mersenne Twister” PRNG

The following generator termed `twister` in MATLAB is recommended for use in simulation. It has extremely long periods, low correlation and passes most statistical tests (the DIEHARD statistical tests). The `twister` random number generator of Makoto Matsumoto and Takuji Nishimura is a variant of the twisted generalized feedback shift-register algorithm, and is known as the “Mersenne Twister” generator [Makoto Matsumoto and Takuji Nishimura, *Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator*, ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1 (Jan. 1998), Pages 3–30]. It has a Mersenne prime period of $2^{19937} - 1$ (about 10^{6000}) and is **equi-distributed** in 623 dimensions. It uses 624 words of state per generator and is comparable in speed to the other generators. The recommended default seed is 5489. See <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html> and http://en.wikipedia.org/wiki/Mersenne_twister for details.

Let us learn to implement the MATLAB function that generates PRNs. In MATLAB the function `rand` produces a deterministic PRN sequence. First, read `help rand`. We can generate PRNs as follows.

Labwork 41 (Calling PRNG in MATLAB) In MATLAB `rand` is basic PRNG command.

```
>> rand(1,10) % generate a 1 X 10 array of PRNs
ans =
    0.8147    0.9058    0.1270    0.9134    0.6324    0.0975    0.2785    0.5469    0.9575    0.9649
>> rand(1,10) % generate another 1 X 10 array of PRNs
ans =
    0.1576    0.9706    0.9572    0.4854    0.8003    0.1419    0.4218    0.9157    0.7922    0.9595
>> rand('twister',5489) % reset the PRNG to default state Mersenne Twister with seed=5489
>> rand(1,10) % reproduce the first array
ans =
    0.8147    0.9058    0.1270    0.9134    0.6324    0.0975    0.2785    0.5469    0.9575    0.9649
>> rand(1,10) % reproduce the second array
ans =
    0.1576    0.9706    0.9572    0.4854    0.8003    0.1419    0.4218    0.9157    0.7922    0.9595
```

In general, you can use any seed value to initiate your PRNG. You may use the `clock` command to set the seed:

```
>> SeedFromClock=sum(100*clock); % save the seed from clock
>> rand('twister',SeedFromClock) % initialize the PRNG
>> rand(1,10)
ans =
    0.3696    0.3974    0.6428    0.6651    0.6961    0.7311    0.8982    0.6656    0.6991    0.8606
>> rand(2,10)
ans =
    0.3432    0.9511    0.3477    0.1007    0.8880    0.0853    0.6067    0.6976    0.4756    0.1523
    0.5827    0.5685    0.0125    0.1555    0.5551    0.8994    0.2502    0.5955    0.5960    0.5700
>> rand('twister',SeedFromClock) % initialize the PRNG to same SeedFromClock
>> rand(1,10)
ans =
    0.3696    0.3974    0.6428    0.6651    0.6961    0.7311    0.8982    0.6656    0.6991    0.8606
```

Labwork 42 (3D plots of triplets generated by the “Mersenne Twister”) Try to find any correlation between triplets generated by the “Mersenne Twister” by rotating the 3D plot generated by the following code:

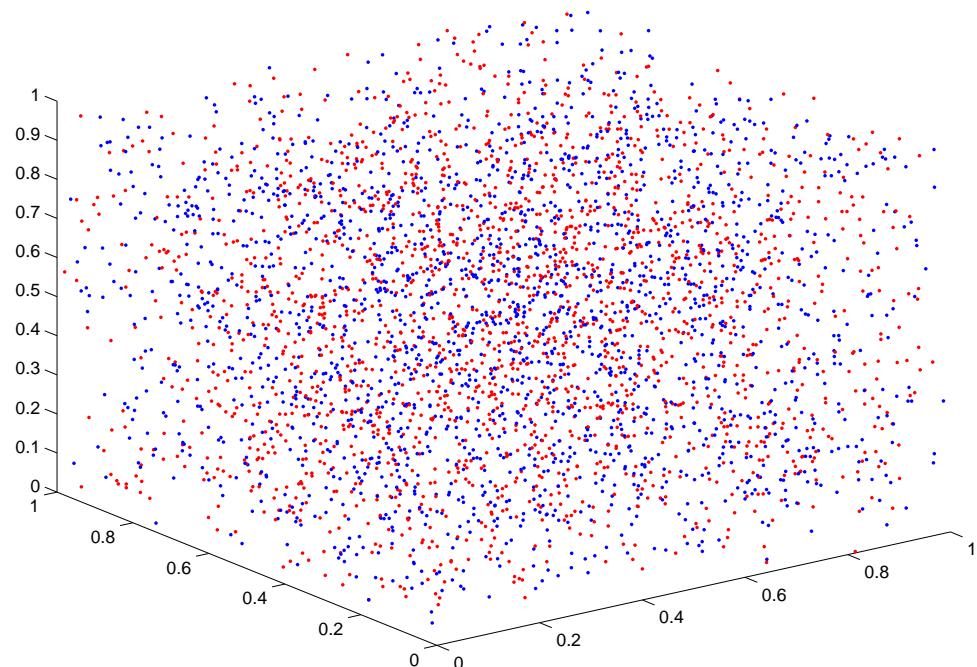
```
>> rand('twister',1234)
>> x=rand(3,2000); % store PRNs in a 3X2000 matrix named x
>> plot3(x(1,:),x(2,:),x(3,:),'.'
```

Compare this with the 3D plot of triplets from RANDU of Labwork 39. Which of these two PRNGs do you think is “more random” looking? and why?

Change the seed value to the recommended default by the authors and look at the point cloud (in red) relative to the previous point cloud (in blue). Rotate the plots to visualise from multiple angles. Are they still random looking?

```
>> rand('twister',1234)% same seed as before
>> x=rand(3,2000); % store PRNs in a 3X2000 matrix named x
>> rand('twister',5489)% the recommended default seed
>> y=rand(3,2000);% store PRNs seeded by 5489 in a 3X2000 matrix named y
>> plot3(x(1,:),x(2,:),x(3,:),'b.') % plot triplets as blue dots
>> hold on;
>> plot3(y(1,:),y(2,:),y(3,:),'r.') % plot triplets as red dots
```

Figure 4.3: Triplet point clouds from the “Mersenne Twister” with two different seeds (see Lab-work 42). .



Chapter 5

Statistics

5.1 Data and Statistics

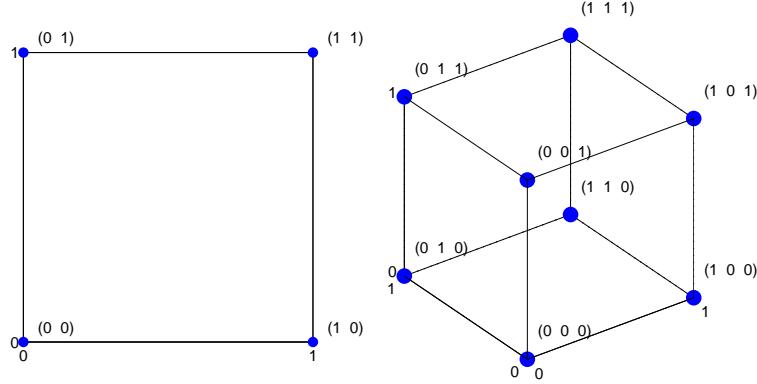
Definition 29 (Data) The function X measures the outcome ω of an experiment with sample space Ω [Often, the sample space is also denoted by S]. Formally, X is a random variable [or a random vector $X = (X_1, X_2, \dots, X_n)$, i.e. a vector of random variables] taking values in the **data space** \mathbb{X} :

$$X(\omega) : \Omega \rightarrow \mathbb{X}.$$

The realisation of the RV X when an experiment is performed is the observation or data $x \in \mathbb{X}$. That is, when the experiment is performed once and it yields a specific $\omega \in \Omega$, the data $X(\omega) = x \in \mathbb{X}$ is the corresponding realisation of the RV X .

Figure 5.1: Sample Space, Random Variable, Realisation, Data, and Data Space.

Example 43 (Tossing a coin n times) For some given parameter $\theta \in \Theta := [0, 1]$, consider n IID Bernoulli(θ) trials, i.e. $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta)$. Then the random vector $X = (X_1, X_2, \dots, X_n)$, which takes values in the data space $\mathbb{X} = \{0, 1\}^n := \{(x_1, x_2, \dots, x_n) : x_i \in \{0, 1\}, i = 1, 2, \dots, n\}$, made up of vertices of the n -dimensional hyper-cube, measures the outcomes of this experiment. A particular realisation of X , upon performance of this experiment, is the observation, data or data vector (x_1, x_2, \dots, x_n) . For instance, if we observed $n - 1$ tails and 1 heads, in that order, then our data vector $(x_1, x_2, \dots, x_{n-1}, x_n) = (0, 0, \dots, 0, 1)$.

Figure 5.2: Data Spaces $\mathbb{X} = \{0, 1\}^2$ and $\mathbb{X} = \{0, 1\}^3$ for two and three Bernoulli trials, respectively.

Definition 30 (Statistic) A **statistic** T is any function of the data:

$$T(x) : \mathbb{X} \rightarrow \mathbb{T} .$$

Thus, a statistic T is also an RV that takes values in the space \mathbb{T} . When $x \in \mathbb{X}$ is the realisation of an experiment, we let $T(x) = t$ denote the corresponding realisation of the statistic T . Sometimes we use $T_n(X)$ and \mathbb{T}_n to emphasise that X is an n -dimensional random vector, i.e. $\mathbb{X} \subset \mathbb{R}^n$

Classwork 44 (Is data a statistic?) Is the RV X , for which the realisation is the observed data $X(\omega) = x$, a statistic? In other words, is the data a statistic? [Hint: consider the identity map $T(x) = x : \mathbb{X} \rightarrow \mathbb{T} = \mathbb{X}$.]

Next, we define two important statistics called the **sample mean** and **sample variance**. Since they are obtained from the sample data, they are called **sample moments**, as opposed to the **population moments**. The corresponding population moments are $\mathbf{E}(X_1)$ and $\mathbf{V}(X_1)$, respectively.

Definition 31 (Sample Mean) From a given a sequence of RVs X_1, X_2, \dots, X_n , we may obtain another RV called the n -samples mean or simply the sample mean:

$$T_n((X_1, X_2, \dots, X_n)) = \bar{X}_n((X_1, X_2, \dots, X_n)) := \frac{1}{n} \sum_{i=1}^n X_i . \quad (5.1)$$

For brevity, we write

$$\bar{X}_n((X_1, X_2, \dots, X_n)) \quad \text{as} \quad \bar{X}_n ,$$

and its realisation

$$\bar{X}_n((x_1, x_2, \dots, x_n)) \quad \text{as} \quad \bar{x}_n .$$

Note that the expectation and variance of \bar{X}_n are:

$$\begin{aligned} \mathbf{E}(\bar{X}_n) &= \mathbf{E}\left(\frac{1}{n} \sum_{i=1}^n X_i\right) && [\text{by definition (5.1)}] \\ &= \frac{1}{n} \sum_{i=1}^n \mathbf{E}(X_i) && [\text{by property (3.10)}] \end{aligned}$$

Furthermore, if every X_i in the original sequence of RVs X_1, X_2, \dots is **identically** distributed with the same expectation, by convention $\mathbf{E}(X_1)$, then:

$$\mathbf{E}(\bar{X}_n) = \frac{1}{n} \sum_{i=1}^n \mathbf{E}(X_i) = \frac{1}{n} \sum_{i=1}^n \mathbf{E}(X_1) = \frac{1}{n} n \mathbf{E}(X_1) = \mathbf{E}(X_1) . \quad (5.2)$$

Similarly, we can show that:

$$\begin{aligned} \mathbf{V}(\bar{X}_n) &= \mathbf{V}\left(\frac{1}{n} \sum_{i=1}^n X_i\right) && [\text{by definition (5.1)}] \\ &= \left(\frac{1}{n}\right)^2 \mathbf{V}\left(\sum_{i=1}^n X_i\right) && [\text{by property (3.12)}] \end{aligned}$$

Furthermore, if the original sequence of RVs X_1, X_2, \dots is **independently** distributed then:

$$\mathbf{V}(\bar{X}_n) = \left(\frac{1}{n}\right)^2 \mathbf{V}\left(\sum_{i=1}^n X_i\right) = \frac{1}{n^2} \sum_{i=1}^n \mathbf{V}(X_i) \quad [\text{by property (3.13)}]$$

Finally, if the original sequence of RVs X_1, X_2, \dots is **independently and identically** distributed with the same variance ($\mathbf{V}(X_1)$ by convention) then:

$$\mathbf{V}(\bar{X}_n) = \frac{1}{n^2} \sum_{i=1}^n \mathbf{V}(X_i) = \frac{1}{n^2} \sum_{i=1}^n \mathbf{V}(X_1) = \frac{1}{n^2} n \mathbf{V}(X_1) = \frac{1}{n} \mathbf{V}(X_1) . \quad (5.3)$$

Labwork 45 (Sample mean) After initializing the fundamental sampler, we draw five samples and then obtain the sample mean using the MATLAB function `mean`. In the following, we will reuse the samples stored in the array `XsFromUni01Twstr101`.

```
>> rand('twister',101); % initialise the fundamental Uniform(0,1) sampler
>> XsFromUni01Twstr101=rand(1,5); % simulate n=5 IID samples from Uniform(0,1) RV
>> SampleMean=mean(XsFromUni01Twstr101);% find sample mean
>> disp(XsFromUni01Twstr101); % The data-points x_1,x_2,x_3,x_4,x_5 are:
    0.5164    0.5707    0.0285    0.1715    0.6853
>> disp(SampleMean); % The Sample mean is :
    0.3945
```

We can thus use `mean` to obtain the sample mean \bar{x}_n of n sample points x_1, x_2, \dots, x_n .

We may also obtain the sample mean using the `sum` function and a division by sample size:

```
>> sum(XsFromUni01Twstr101) % take the sum of the elements of the XsFromUni01Twstr101 array
ans =      1.9723
>> sum(XsFromUni01Twstr101) / 5 % divide the sum by the sample size 5
ans =      0.3945
```

We can also obtain the sample mean via matrix product or multiplication as follows:

```
>> size(XsFromUni01Twstr101) % size(SomeArray) gives the size or dimensions of the arrar SomeArray
ans =      1      5
>> ones(5,1) % here ones(5,1) is an array of 1's with size or dimension 5 X 1
ans =
    1
    1
```

```

1
1
1
>> XsFromUni01Twstr101 * ones(5,1) % multiplying an 1 X 5 matrix with a 5 X 1 matrix of Ones
ans = 1.9723
>> XsFromUni01Twstr101 * ( ones(5,1) * 1/5) % multiplying an 1 X 5 matrix with a 5 X 1 matrix of 1/5 's
ans = 0.3945

```

Definition 32 (Sample Variance & Standard Deviation) From a given a sequence of random variables X_1, X_2, \dots, X_n , we may obtain another statistic called the n -samples variance or simply the sample variance :

$$T_n((X_1, X_2, \dots, X_n)) = S_n^2((X_1, X_2, \dots, X_n)) := \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2 . \quad (5.4)$$

For brevity, we write $S_n^2((X_1, X_2, \dots, X_n))$ as S_n^2 and its realisation $S_n^2((x_1, x_2, \dots, x_n))$ as s_n^2 .

Sample standard deviation is simply the square root of sample variance:

$$S_n((X_1, X_2, \dots, X_n)) = \sqrt{S_n^2((X_1, X_2, \dots, X_n))} \quad (5.5)$$

For brevity, we write $S_n((X_1, X_2, \dots, X_n))$ as S_n and its realisation $S_n((x_1, x_2, \dots, x_n))$ as s_n .

Once again, if $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} X_1$, the expectation of the sample variance is:

$$\mathbf{E}(S_n^2) = \mathbf{V}(X_1) .$$

Labwork 46 (Sample variance and sample standard deviation) We can compute the sample variance and sample standard deviation for the five samples stored in the array `XsFromUni01Twstr101` from Labwork 45 using MATLAB's functions `var` and `std`, respectively.

```

>> disp(XsFromUni01Twstr101); % The data-points x_1,x_2,x_3,x_4,x_5 are :
    0.5164    0.5707    0.0285    0.1715    0.6853
>> SampleVar=var(XsFromUni01Twstr101);% find sample variance
>> SampleStd=std(XsFromUni01Twstr101);% find sample standard deviation
>> disp(SampleVar) % The sample variance is:
    0.0785
>> disp(SampleStd) % The sample standard deviation is:
    0.2802

```

It is important to bear in mind that the statistics such as sample mean and sample variance are random variables and have an underlying distribution.

Definition 33 (Order Statistics) Suppose $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F$, where F is the DF from the set of all DFs over the real line. Then, the n -sample **order statistics** $X_{([n])}$ is:

$$X_{([n])}((X_1, X_2, \dots, X_n)) := (X_{(1)}, X_{(2)}, \dots, X_{(n)}) , \text{ such that, } X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)} . \quad (5.6)$$

For brevity, we write $X_{([n])}((X_1, X_2, \dots, X_n))$ as $X_{([n])}$ and its realisation $X_{([n])}((x_1, x_2, \dots, x_n))$ as $x_{([n])} = (x_{(1)}, x_{(2)}, \dots, x_{(n)})$.

Without going into the details of how to sort the data in ascending order to obtain the order statistics (an elementary topic of an Introductory Computer Science course), we simply use MATLAB's function `sort` to obtain the order statistics, as illustrated in the following example.

Labwork 47 (Order statistics and sorting) The order statistics for the five samples stored in `XsFromUni01Twstr101` from Labwork 45 can be computed using `sort` as follows:

```
>> disp(XsFromUni01Twstr101); % display the sample points
    0.5164    0.5707    0.0285    0.1715    0.6853
>> SortedXsFromUni01Twstr101=sort(XsFromUni01Twstr101); % sort data
>> disp(SortedXsFromUni01Twstr101); % display the order statistics
    0.0285    0.1715    0.5164    0.5707    0.6853
```

Therefore, we can use `sort` to obtain our order statistics $x_{(1)}, x_{(2)}, \dots, x_{(n)}$ from n sample points x_1, x_2, \dots, x_n .

Next, we will introduce a family of common statistics, called the q^{th} quantile, by first defining the function:

Definition 34 (Inverse DF or Inverse CDF or Quantile Function) Let X be an RV with DF F . The **inverse DF** or **inverse CDF** or **quantile function** is:

$$F^{[-1]}(q) := \inf \{x : F(x) > q\}, \quad \text{for some } q \in [0, 1]. \quad (5.7)$$

If F is strictly increasing and continuous then $F^{[-1]}(q)$ is the unique $x \in \mathbb{R}$ such that $F(x) = q$.

A **functional** is merely a function of another function. Thus, $T(F) : \{\text{All DFs}\} \rightarrow \mathbb{T}$, being a map or function from the space of DFs to its range \mathbb{T} , is a functional. Some specific examples of functionals we have already seen include:

1. The **mean** of RV $X \sim F$ is a function of the DF F :

$$T(F) = \mathbf{E}(X) = \int x dF(x).$$

2. The **variance** of RV $X \sim F$ is a function of the DF F :

$$T(F) = \mathbf{E}(X - \mathbf{E}(X))^2 = \int (x - \mathbf{E}(X))^2 dF(x).$$

3. The **value of DF at a given $x \in \mathbb{R}$** of RV $X \sim F$ is also a function of DF F :

$$T(F) = F(x).$$

Other functionals of F that depend on the quantile function $F^{[-1]}$ are:

1. The q^{th} **quantile** of RV $X \sim F$:

$$T(F) = F^{[-1]}(q) \quad \text{where } q \in [0, 1].$$

2. The **first quartile** or the 0.25^{th} **quantile** of the RV $X \sim F$:

$$T(F) = F^{[-1]}(0.25).$$

3. The **median** or the **second quartile** or the 0.50^{th} **quantile** of the RV $X \sim F$:

$$T(F) = F^{[-1]}(0.50) .$$

4. The **third quartile** or the 0.75^{th} **quantile** of the RV $X \sim F$:

$$T(F) = F^{[-1]}(0.75) .$$

Definition 35 (Empirical Distribution Function (EDF or ECDF)) Suppose we have n IID RVs, $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F$, where F is a DF from the set of all DFs over the real line. Then, the n -sample empirical distribution function (EDF or ECDF) is the discrete distribution function \hat{F}_n that puts a probability mass of $1/n$ at each sample or data point x_i :

$$\hat{F}_n(x) = \frac{\sum_{i=1}^n \mathbf{1}(X_i \leq x)}{n}, \quad \text{where} \quad \mathbf{1}(X_i \leq x) := \begin{cases} 1 & \text{if } x_i \leq x \\ 0 & \text{if } x_i > x \end{cases} \quad (5.8)$$

Labwork 48 (Plot of empirical CDF) Let us plot the ECDF for the five samples drawn from the $\text{Uniform}(0, 1)$ RV in Labwork 45 using the MATLAB function `ECDF` (given in Labwork 247). Let us super-impose the samples and the true DF as depicted in Figure 5.3 with the following script:

```
plotunifecdf.m
xs = -1:0.01:2; % vector xs from -1 to 2 with increment .05 for x values
% get the [0,1] uniform DF or cdf of xs in vector cdf
cdf=zeros(size(xs));% initialise cdf as zero
indices = find(xs>=1); cdf(indices) = 1; % set cdf as 1 when xs >= 1
indices = find(xs>=0 & xs<=1); cdf(indices)=xs(indices); % cdf=xs when 0 <= xs <= 1
plot(xs,cdf,'r') % plot the DF
hold on; title('Uniform [0,1] DF and ECDF'); xlabel('x'); axis([-0.2 1.2 -0.2 1.2])
x=[0.5164, 0.5707, 0.0285, 0.1715, 0.6853]; % five samples
plot(x,zeros(1,5),'r+','LineWidth',2,'MarkerSize',10)% plot the data as red + marks
hold on; grid on; % turn on grid
ECDF(x,1,.2,.6);% ECDF (type help ECDF) plot is extended to left and right by .2 and .4, respectively.
```

Definition 36 (q^{th} Sample Quantile) For some $q \in [0, 1]$ and n IID RVs $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F$, we can obtain the ECDF \hat{F}_n using (5.8). The q^{th} **sample quantile** is defined as the statistic (statistical functional):

$$T(\hat{F}_n) = \hat{F}_n^{[-1]}(q) := \inf \{x : \hat{F}_n^{[-1]}(x) \geq q\} . \quad (5.9)$$

By replacing q in this definition of the q^{th} sample quantile by 0.25, 0.5 or 0.75, we obtain the first, second (**sample median**) or third **sample quartile**, respectively.

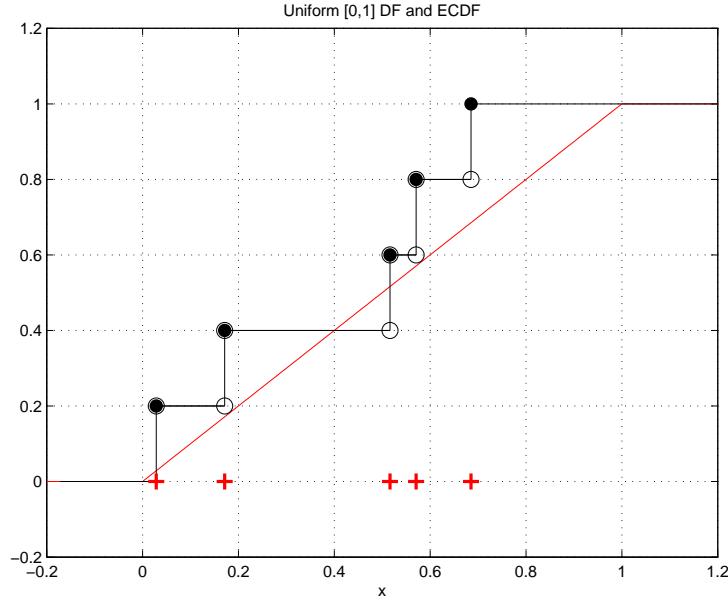
The following algorithm can be used to obtain the q^{th} sample quantile of n IID samples (x_1, x_2, \dots, x_n) on the basis of their order statistics $(x_{(1)}, x_{(2)}, \dots, x_{(n)})$.

The q^{th} sample quantile, $\hat{F}_n^{[-1]}(q)$, is found by interpolation from the order statistics $(x_{(1)}, x_{(2)}, \dots, x_{(n)})$ of the n data points (x_1, x_2, \dots, x_n) , using the formula:

$$\hat{F}_n^{[-1]}(q) = (1 - \delta)x_{(i+1)} + \delta x_{(i+2)}, \quad \text{where,} \quad i = \lfloor (n-1)q \rfloor \quad \text{and} \quad \delta = (n-1)q - \lfloor (n-1)q \rfloor .$$

Thus, the **sample minimum** of the data points (x_1, x_2, \dots, x_n) is given by $\hat{F}_n^{[-1]}(0)$, the **sample maximum** is given by $\hat{F}_n^{[-1]}(1)$ and the **sample median** is given by $\hat{F}_n^{[-1]}(0.5)$, etc.

Figure 5.3: Plot of the DF of Uniform(0, 1), five IID samples from it, and the ECDF \hat{F}_5 for these five data points $x = (x_1, x_2, x_3, x_4, x_5) = (0.5164, 0.5707, 0.0285, 0.1715, 0.6853)$ that jumps by $1/5 = 0.20$ at each of the five samples.



Algorithm 2 q^{th} Sample Quantile of Order Statistics

1: *input:*

1. q in the q^{th} sample quantile, i.e. the argument q of $\hat{F}_n^{[-1]}(q)$,
 2. order statistic $(x_{(1)}, x_{(2)}, \dots, x_{(n)})$, i.e. the sorted (x_1, x_2, \dots, x_n) , where $n > 0$.
- 2: *output:* $\hat{F}_n^{[-1]}(q)$, the q^{th} sample quantile
- 3: $i \leftarrow \lfloor (n-1)q \rfloor$
 - 4: $\delta \leftarrow (n-1)q - i$
 - 5: **if** $i = n-1$ **then**
 - 6: $\hat{F}_n^{[-1]}(q) \leftarrow x_{(i+1)}$
 - 7: **else**
 - 8: $\hat{F}_n^{[-1]}(q) \leftarrow (1-\delta)x_{(i+1)} + \delta x_{(i+2)}$
 - 9: **end if**
- 10: *return:* $\hat{F}_n^{[-1]}(q)$
-

Labwork 49 (The q^{th} sample quantile) Use the implementation of Algorithm 2 in Labwork 248 as the MATLAB function `qthSampleQuantile` to find the q^{th} sample quantile of two simulated data arrays:

1. `SortedXsFromUni01Twstr101`, the order statistics that was constructed in Labwork 47 and
2. Another sorted array of 7 samples called `SortedXs`

```
>> disp(SortedXsFromUni01Twstr101)
    0.0285    0.1715    0.5164    0.5707    0.6853
>> rand('twister',420);
>> SortedXs=sort(rand(1,7));
>> disp(SortedXs)
    0.1089    0.2670    0.3156    0.3525    0.4530    0.6297    0.8682
>> for q=[0, 0.25, 0.5, 0.75, 1.0]
    disp([q, qthSampleQuantile(q,SortedXsFromUni01Twstr101) ...
           qthSampleQuantile(q,SortedXs)])
end
      0    0.0285    0.1089
    0.2500    0.1715    0.2913
    0.5000    0.5164    0.3525
    0.7500    0.5707    0.5414
    1.0000    0.6853    0.8682
```

5.2 Exploring Data and Statistics

5.2.1 Univariate Data

A **histogram** is a graphical representation of the frequency with which elements of a data array:

$$x = (x_1, x_2, \dots, x_n) ,$$

of real numbers fall within each of the m intervals or **bins** of some **interval partition**:

$$b := (b_1, b_2, \dots, b_m) := ([\underline{b}_1, \bar{b}_1], [\underline{b}_2, \bar{b}_2], \dots, [\underline{b}_m, \bar{b}_m])$$

of the **data range** of x given by the closed interval:

$$\mathcal{R}(x) := [\min\{x_1, x_2, \dots, x_n\}, \max\{x_1, x_2, \dots, x_n\}] .$$

Elements of this partition b are called bins, their mid-points are called **bin centres**:

$$c := (c_1, c_2, \dots, c_m) := ((\underline{b}_1 + \bar{b}_1)/2, (\underline{b}_2 + \bar{b}_2)/2, \dots, (\underline{b}_m + \bar{b}_m)/2)$$

and their overlapping boundaries, i.e. $\bar{b}_i = \underline{b}_{i+1}$ for $1 \leq i < m$, are called **bin edges**:

$$d := (d_1, d_2, \dots, d_{m+1}) := (\underline{b}_1, \underline{b}_2, \dots, \underline{b}_{m-1}, \underline{b}_m, \bar{b}_m) .$$

For a given partition of the data range $\mathcal{R}(x)$ or some superset of $\mathcal{R}(x)$, three types of histograms are possible: frequency histogram, relative frequency histogram and density histogram. Typically, the partition b is assumed to be composed of m overlapping intervals of the same width $w = \bar{b}_i - \underline{b}_i$ for all $i = 1, 2, \dots, m$. Thus, a histogram can be obtained by a set of bins along with their corresponding **heights**:

$$h = (h_1, h_2, \dots, h_m) , \text{ where } h_k := g(\#\{x_i : x_i \in b_k\})$$

Thus, h_k , the height of the k -th bin, is some function g of the number of data points that fall in the bin b_k . Formally, a histogram is a sequence of ordered pairs:

$$((b_1, h_1), (b_2, h_2), \dots, (b_m, h_m)) .$$

Given a partition b , a **frequency histogram** is the histogram:

$$((b_1, h_1), (b_2, h_2), \dots, (b_m, h_m)) , \text{ where } h_k := \#\{x_i : x_i \in b_k\} ,$$

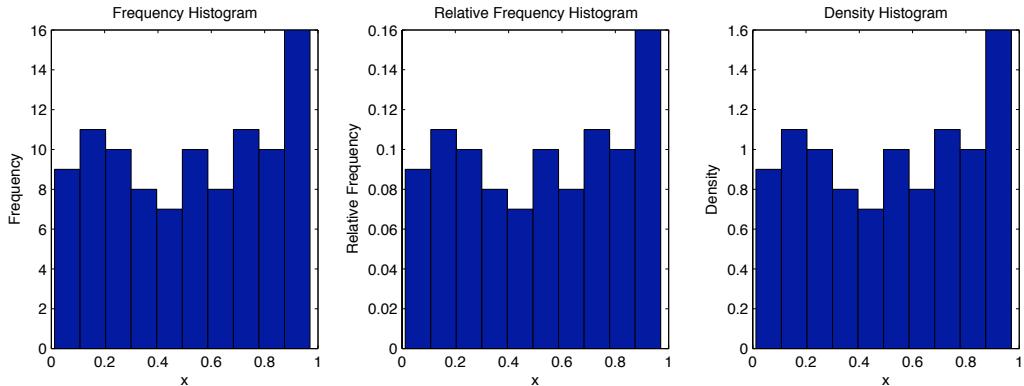
a **relative frequency histogram** is the histogram:

$$((b_1, h_1), (b_2, h_2), \dots, (b_m, h_m)) , \text{ where } h_k := n^{-1} \#\{x_i : x_i \in b_k\} ,$$

and a **density histogram** is the histogram:

$$((b_1, h_1), (b_2, h_2), \dots, (b_m, h_m)) , \text{ where } h_k := (w_k n)^{-1} \#\{x_i : x_i \in b_k\} , w_k := \bar{b}_k - \underline{b}_k .$$

Figure 5.4: Frequency, Relative Frequency and Density Histograms



Labwork 50 (Histograms with specified number of bins for univariate data) Let us use samples from the `rand('twister',5489)` as our data set x and plot various histograms. Let us use `hist` function (read `help hist`) to make a default histogram with ten bins. Then we can make three types of histograms as shown in Figure 5.4 as follows:

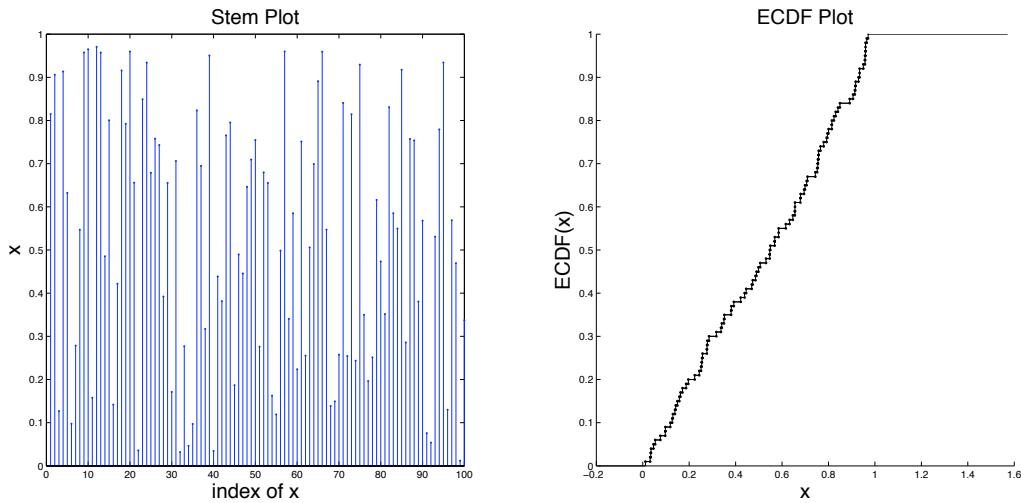
```
>> rand('twister',5489);
>> x=rand(1,100); % generate 100 PRNs
>> hist(x) % see what default hist does in Figure Window
>> % Now let us look deeper into the last hist call
>> [Fs, Cs] = hist(x) % Cs is the bin centers and Fs is the frequencies of data set x
Fs =
    9     11     10      8      7     10      8     11     10     16
Cs =
    0.0598    0.1557    0.2516    0.3474    0.4433    0.5392    0.6351    0.7309    0.8268    0.9227
>> % produce a histogram plot the last argument 1 is the width value for immediately adjacent bars -- help bar
>> bar(Cs,Fs,1) % create a frequency histogram
>> bar(Cs,Fs/100,1) % create a relative frequency histogram
>> bar(Cs,Fs/(0.1*100),1) % create a density histogram (area of bars sum to 1)
>> sum(Fs/(0.1*100) .* ones(1,10)*0.1) % checking if area does sum to 1
>> ans = 1
```

Try making a density histogram with 1000 samples from `rand` with 15 bins. You can specify the number of bins by adding an extra argument to `hist`, for e.g. `[Fs, Cs] = hist(x,15)` will produce 15 bins of equal width over the data range $\mathcal{R}(x)$.

Labwork 51 (Stem plots and ECDF plots for univariate data) We can also visualise the 100 data points in the array x using stem plot and ECDF plot as shown in Figure 5.5 as follows:

```
>> rand('twister',5489);
>> x=rand(1,100); % produce 100 samples with rand
>> stem(x,'.') % make a stem plot of the 100 data points in x (the option '.' gives solid circles for x)
>>% ECDF (type help ECDF) plot is extended to left and right by .2 and .6, respectively
>>% (second parameter 6 makes the dots in the plot smaller).
>> ECDF(x,6,.2,.6);
```

Figure 5.5: Frequency, Relative Frequency and Density Histograms



We can also visually summarise univariate data using the **box plot** or **box-whisker plot** available in the Stats Toolbox of MATLAB. These family of plots display a set of sample quantiles, typically they are include, the median, the first and third quartiles and the minimum and maximum values of our data array x .

5.2.2 Bivariate Data

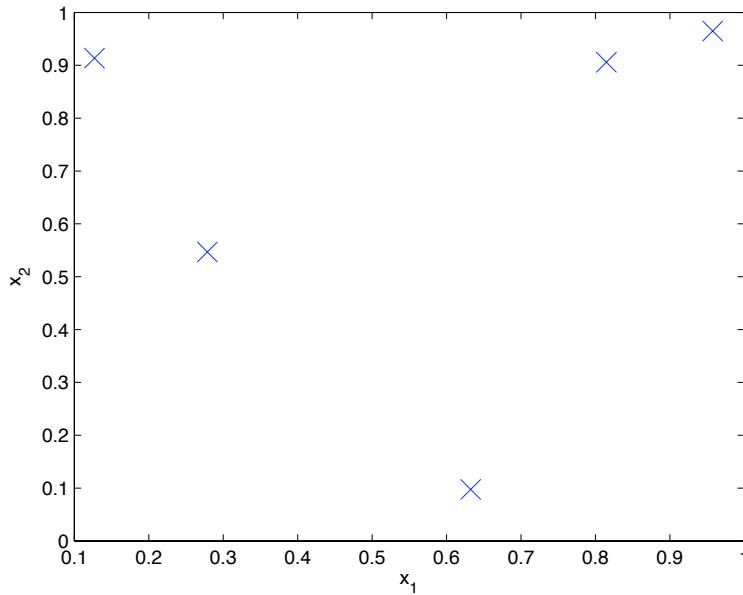
By bivariate data array x we mean a $2 \times n$ matrix of real numbers or equivalently n ordered pairs of points $(x_{1,i}, x_{2,i})$ as $i = 1, 2, \dots, n$. The most elementary visualisation of these n ordered pairs is in orthogonal Cartesian co-ordinates. Such plots are termed **2D scatter plots** in statistics.

Labwork 52 (Visualising bivariate data) Let us generate a 2×5 array representing samples of 5 ordered pairs sampled uniformly at random over the unit square $[0, 1] \times [0, 1]$. We can make 2D scatter plot as shown in Figure 5.6 as follows:

```
>> rand('twister',5489);
>> x=rand(2,5)% create a sequence of 5 ordered pairs uniformly from unit square [0,1]X[0,1]
x =
    0.8147    0.1270    0.6324    0.2785    0.9575
    0.9058    0.9134    0.0975    0.5469    0.9649
>> plot(x(1,:),x(2,:),'x') % a 2D scatter plot with marker cross or 'x'
>> plot(x(1,:),x(2,:),'x', 'MarkerSize',15) % a 2D scatter plot with marker cross or 'x' and larger Marker size
>> xlabel('x_1'); ylabel('x_2'); % label the axes
```

There are several other techniques for visualising bivariate data, including, 2D histograms, surface plots, heat plots, and we will encounter some of them in the sequel.

Figure 5.6: 2D Scatter Plot



5.2.3 Trivariate Data

Trivariate data is more difficult to visualise on paper but playing around with the rotate 3D feature in MATLAB's Figure window can help bring a lot more perspective.

Labwork 53 (Visualising trivariate data) We can make **3D scatter plots** as shown in Figure 5.7 as follows:

```
>> rand('twister',5489);
>> x=rand(3,5)% create a sequence of 5 ordered triples uniformly from unit cube [0,1]X[0,1]X[0,1]
x =
    0.8147    0.9134    0.2785    0.9649    0.9572
    0.9058    0.6324    0.5469    0.1576    0.4854
    0.1270    0.0975    0.9575    0.9706    0.8003
>> plot3(x(1,:),x(2,:),x(3,:),'x') % a simple 3D scatter plot with marker 'x'
>>% a more interesting one with options that control marker type, line-style,
>>% colour in [Red Green Blue] values and marker size - read help plot3 for more options
>> plot3(x(1,:),x(2,:),x(3,:),'Marker','*','LineStyle','none','Color',[1 0 1],'MarkerSize',15)
>> plot3(x(1,:),x(2,:),x(3,:),'m*','MarkerSize',15) % makes same figure as before but shorter to write
>> box on % turn on the box and see the effect on the Figure
>> grid on % turn on the grid and see the effect on the Figure
>> xlabel('x_1'); ylabel('x_2'); zlabel('x_3'); % assign labels to x,y and z axes
```

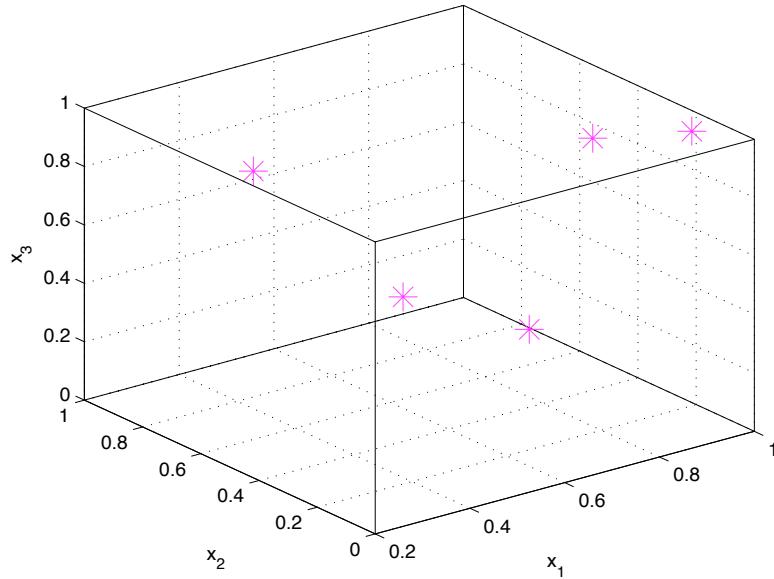
Repeat the visualisation below with a larger array, say $x=\text{rand}(3,1000)$, and use the rotate 3D feature in the Figure window to visually explore the samples in the unit cube. Do they seem to be uniformly distributed inside the unit cube?

There are several other techniques for visualising trivariate data, including, iso-surface plots, moving surface or heat plots, and you will encounter some of them in the future.

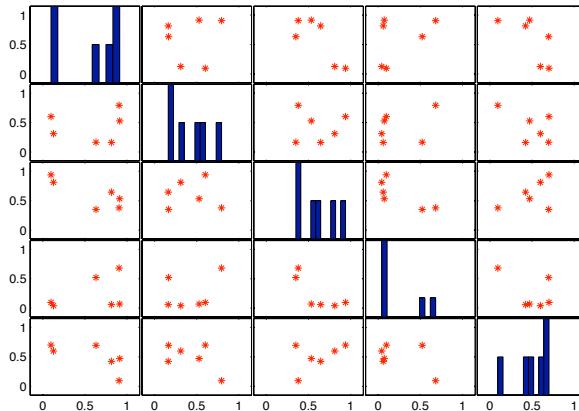
5.2.4 Multivariate Data

For high-dimensional data in d -dimensional space \mathbb{R}^d with $d \geq 3$ you have to look at several lower dimensional projections of the data. We can simultaneously look at 2D scatter plots for every pair of

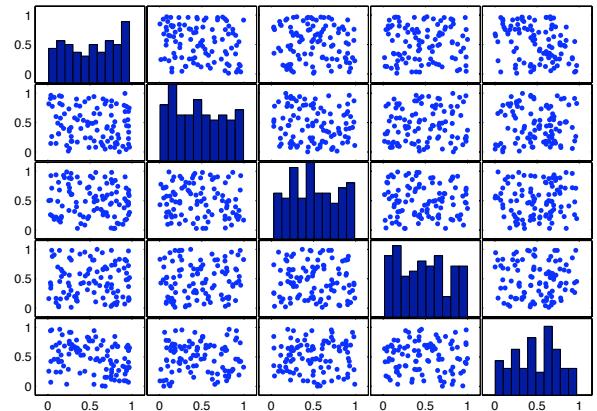
Figure 5.7: 3D Scatter Plot



co-ordinates $\{(i, j) \in \{1, 2, \dots, d\}^2 : i \neq j\}$ and at histograms for every co-ordinate $i \in \{1, 2, \dots, d\}$ of the n data points in \mathbb{R}^d . Such a set of low-dimensional projections can be conveniently represented in a $d \times d$ matrix of plots called a **matrix plot**.

Figure 5.8: Plot Matrix of uniformly generated data in $[0, 1]^5$ 

(a) First six samples



(b) All thousand samples

Labwork 54 Let us make matrix plots from a uniformly generated sequence of 100 points in 5D unit cube $[0, 1]^5$ as shown in Figure 5.8.

```
>> rand('twister',5489);
>> % generate a sequence of 1000 points uniformly distributed in 5D unit cube [0,1]X[0,1]X[0,1]X[0,1]X[0,1]
>> x=rand(1000,5);
>> x(1:6,:) % first six points in our 5D unit cube, i.e., the first six rows of x
ans =
```

```

0.8147    0.6312    0.7449    0.3796    0.4271
0.9058    0.3551    0.8923    0.3191    0.9554
0.1270    0.9970    0.2426    0.9861    0.7242
0.9134    0.2242    0.1296    0.7182    0.5809
0.6324    0.6525    0.2251    0.4132    0.5403
0.0975    0.6050    0.3500    0.0986    0.7054
>> plotmatrix(x(1:5,:),'r*') % make a plot matrix
>> plotmatrix(x) % make a plot matrix of all 1000 points

```

5.3 Loading and Exploring Real-world Data

All of the data we have played with so far were computer-generated. It is time to get our hands dirty with real-world data. The first step is to obtain the data. Often, publicly-funded institutions allow the public to access their databases. Such data can be fetched from appropriate URLs in one of the two following ways:

Method A: Manually download by filling the appropriate fields in an online request form.

Method B: Automagically download directly from your MATLAB session.

Then we want to inspect it for inconsistencies, missing values and replace them with `NaN` values in MATLAB that stand for not-any-number. Finally, we can visually explore, transform and interact with the data to discover interesting patterns that are hidden in the data. This process is called *exploratory data analysis* and is the foundational first step towards subsequent computational statistical experiments [*John W. Tukey, Exploratory Data Analysis, Addison-Wesely, New York, 1977*].

5.3.1 Geological Data

Let us focus on the data of earth quakes that heavily damaged Christchurch on February 22 2011. This data can be fetched from the URL <http://magma.geonet.org.nz/resources/quakesearch/> by Method A and loaded into MATLAB for exploratory data analysis as done in Labwork 55.

Labwork 55 Let us go through the process one step at a time using Method A.

1. Download the data as a CSV or *comma separated variable* file in plain ASCII text (this has been done for this data already for you and saved as `NZ20110222earthquakes.csv` in the `CSEMatlabScripts` directory).
2. Open the file in a simple text editor such as `Note Pad` in Windows or one of the following editors in OS X, Unix, Solaris, Linux/GNU variants such as Ubuntu, SUSE, etc: `vi`, `vim`, `emacs`, `geany`, etc. The first three and last two lines of this file look as follows:

```

CUSP_ID,LAT,LONG,NZMGE,NZMGN,ORI_YEAR,ORI_MONTH,ORI_DAY,ORI_HOUR,ORI_MINUTE,ORI_SECOND,MAG,DEPTH
3481751,-43.55432,172.68898,2484890,5739375,2011,2,22,0,0,31.27814,3.79,5.8559,
3481760,-43.56579,172.70621,2486287,5738106,2011,2,22,0,0,43.70276,3.76,5.4045,
.
.
.
3469114,-43.58007,172.67126,2483470,5736509,2011,2,22,23,28,11.1014,3.117,3,
3469122,-43.55949,172.70396,2486103,5738805,2011,2,22,23,50,1.06171,3.136,12,

```

The thirteen columns correspond to fairly self-descriptive features of each measured earth quake given in the first line or row. They will become clear in the sequel. Note that the comma character (',') separates each unit or measurement or description in any CSV file.

3. The next set of commands show you how to load, manipulate and visually explore this data.

```
%>> %% Load the data from the comma delimited text file 'NZ20110222earthquakes.csv' with
%% the following column IDs
%% CUSP_ID,LAT,LONG,NZMGE,NZMGN,ORI_YEAR,ORI_MONTH,ORI_DAY,ORI_HOUR,ORI_MINUTE,ORI_SECOND,MAG,DEPTH
%% Using MATLAB's dlmread command we can assign the data as a matrix to EQ;
%% note that the option 1,0 to dlmread skips first row of column descriptors
%
% the variable EQall is about to be assigned the data as a matrix
EQall = dlmread('NZ20110222earthquakes.csv', ',', 1, 0);
size(EQall) % report the dimensions or size of the matrix EQall
ans =
    145      14
```

4. In order to understand the syntax in detail get help from MATLAB !

```
>> help dlmread
DLMREAD Read ASCII delimited file.
.
.
.
```

5. When there are units in the CSV file that can't be converted to floating-point numbers, it is customary to load them as a `NaN` or *Not-a-Number* value in MATLAB . So, let's check if there are any rows with `NaN` values and remove them from our analysis. Note that this is not the only way to deal with missing data! After that let's remove any locations outside Christchurch and its suburbs (we can find the latitude and longitude bounds from online resources easily) and finally view the 4-tuples of (latitude, longitude, magnitude, depth) for each measured earth quake in Christchurch on February 22 of 2011 as a scatter plot shown in Figure 5.9 (the axes labels were subsequently added from clicking <Edit> and <Figure Properties...> tabs of the output Figure Window).

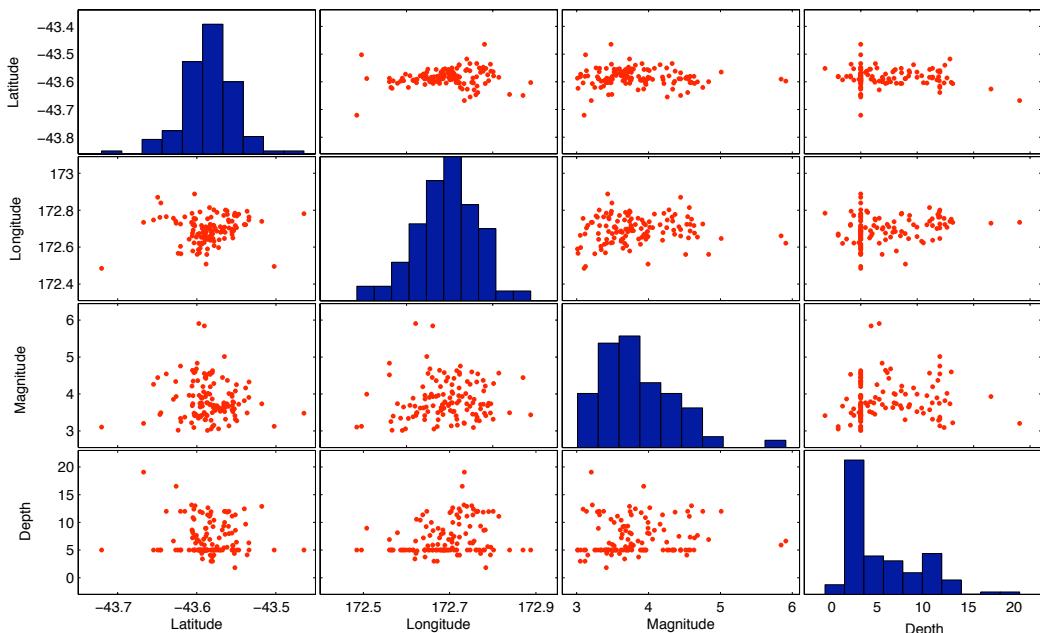
```
>> EQall(any(isnan(EQall),2),:) = []; %Remove any rows containing NaNs from the matrix EQall
>> % report the size of EQall and see if it is different from before we removed and NaN containing rows
>> size(EQall)
ans = 145 14
>> % remove locations outside Chch and assign it to a new variable called EQ
>> EQ = EQall(-43.75<EQall(:,2) & EQall(:,2)<-43.45 ...
& 172.45<EQall(:,3) & EQall(:,3)<172.9 & EQall(:,12)>3, :);
>> % now report the size of the earthquakes in Christchurch in variable EQ
>> size(EQ)
ans = 124 14
>> % assign the four variables of interest
>> LatData=EQ(:,2); LonData=EQ(:,3); MagData=EQ(:,12); DepData=EQ(:,13);
>> % finally make a plot matrix of these 124 4-tuples as red points
>> plotmatrix([LatData,LonData,MagData,DepData], 'r.');
```

All of these commands have been put in a script M-file `NZEQChCch20110222.m` in Labwork 249 and you can simply call it from the command window to automatically load the data and assign it to the variables `EQAll` `EQ`, `LatData`, `LonData`, `MagData` and `DepData`, instead of retyping each command above every time you need these matrices in MATLAB , as follows:

```
>> NZEQChCch20110222
ans =    145    14
ans =    145    14
ans =    124    14
```

In fact, we will do exactly this to conduct more exploratory data analysis with these earth quake measurements in Labwork 56.

Figure 5.9: Matrix of Scatter Plots of the latitude, longitude, magnitude and depth of the 22-02-2011 earth quakes in Christchurch, New Zealand.



Labwork 56 Try to understand how to manipulate time stamps of events in MATLAB and the Figures being output by following the comments in the script file `NZEQChCch20110222EDA.m`.

```
>> NZEQChCch20110222
ans =    145    14
ans =    145    14
ans =    124    14
ans =    145    14
ans =    145    14
ans =    124    14
ans = 22-Feb-2011 00:00:31
ans = 22-Feb-2011 23:50:01
```

```
----- NZEQChCch20110222EDA.m -----
%% Load the data from the comma delimited text file 'NZ20110222earthquakes.csv'
% using the script M-file NZEQChCch20110222.m
NZEQChCch20110222
%% working with time stamps is tricky
%% time is encoded by columns 6 through 11
%% as origin of earthquake in year, month, day, hour, minute, sec:
```

```

%% ORI_YEAR,ORI_MONTH,ORI_DAY,ORI_HOUR,ORI_MINUTE,ORI_SECOND
%% datenum is Matlab's date encoding function see help datenum
TimeData=datenum(EQ(:,6:11)); % assign origin times of earth quakes in datenum coordinates
MaxD=max(TimeData); % get the latest time of observation in the data
MinD=min(TimeData); % % get the earliest time of observation in the data
datestr(MinD) % a nice way to conver to calendar time!
datestr(MaxD) % ditto

% recall that there four variables were assigned in NZEQChCch20110222.m
% LatData=EQ(:,2); LonData=EQ(:,3); MagData=EQ(:,12); DepData=EQ(:,13);

%clear any existing Figure windows
clf
plot(TimeData,MagData,'o-') % plot origin time against magnitude of each earth quake

figure % tell matlab you are about to make another figure
plotmatrix([LatData,LonData,MagData,DepData],'r.');

figure % tell matlab you are about to make another figure
scatter(LonData,LatData,'.') % plot the LONGitude Vs. LATtitude

figure % tell matlab you are about to make another figure
% relative frequency histogram of magnitudes from 0 to 12 on Richter Scale with 15 bins
hist(MagData,15)

%max(MagData)

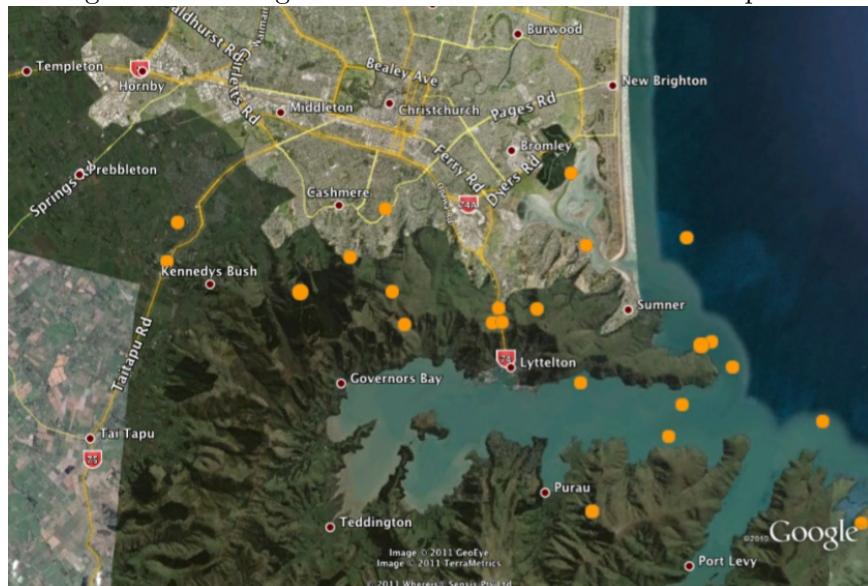
figure % tell matlab you are about to make another figure
semilogx(DepData,MagData,'.') % see the depth in log scale

%%%%%
% more advanced topic - uncomment and read help if bored
%tri = delaunay(LatData,LonData);
%triplot(tri,LatData,LonData,DepData);

```

Geostatistical exploratory data analysis with Google Earth

Figure 5.10: Google Earth Visualisation of the earth quakes



A global search at <http://neic.usgs.gov/cgi-bin/epic/epic.cgi> with the following parame-

ters:

Date Range: 2011 2 22 to 2011 2 22

Catalog: USGS/NEIC (PDE-Q)

produced 43 earth quakes world-wide, including those in Christchurch as shown in Figure 5.10. One can do a lot more than a mere visualisation with the USGS/NEIC database of earth-quakes worldwide, the freely available Google earth software bundle <http://www.google.com/earth/index.html> and the freely available MATLAB package googleearth from http://www.mathworks.com/matlabcentral/fx_files/12954/4/content/googleearth/html/html_product_page.html.

5.3.2 Metereological Data

New Zealand's meteorological service NIWA provides weather data under its TERMS AND CONDITIONS FOR ACCESS TO DATA (See http://cliflo.niwa.co.nz/doc/terms_print.html). We will explore some data of rainfall and temperatures from NIWA.

Daily Rainfalls in Christchurch

Automagic downloading of the data by Method B can be done if the data provider allows automated queries. It can be accomplished by `urlread` for instance.

Paul Brouwers has a basic CliFlo datafeed on <http://www.math.canterbury.ac.nz/php/lib/cliflo/rainfall.php>. This returns the date and rainfall in milli meters as measured from the CHCH aeroclub station. It is assumed that days without readings would not be listed. The data doesn't go back much before 1944.

Labwork 57 Understand how Figure 5.11 is obtained by the script file `RainFallsInChch.m` by typing and following the comments:

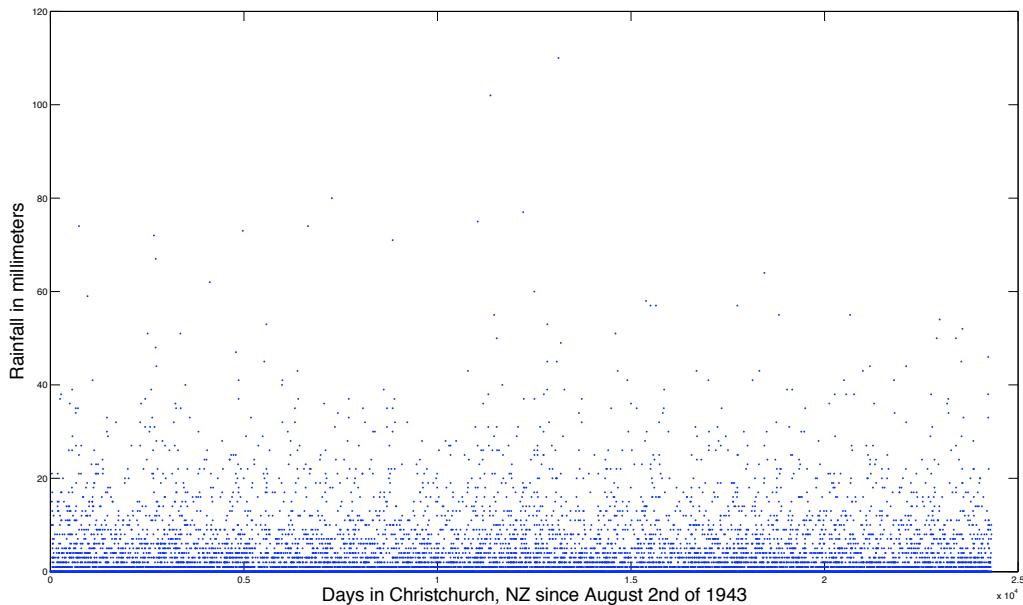
```
>> RainFallsInChch
RainFallsChch = [24312x1 int32] [24312x1 double]
ans = 24312 2
FirstDayOfData = 19430802
LastDayOfData = 20100721
```

```
RainFallsInChch.m
%% How to download data from an URL directly without having to manually
%% fill out forms
% first make a string of the data using urlread (read help urlread if you want details)
StringData = urlread('http://www.math.canterbury.ac.nz/php/lib/cliflo/rainfall.php');
RainFallsChch = textscan(StringData, '%d %f', 'delimiter', ',')
RC = [RainFallsChch{1} RainFallsChch{2}]; % assign Matlab cells as a matrix
size(RC) % find the size of the matrix

FirstDayOfData = min(RC(:,1))
LastDayOfData = max(RC(:,1))

plot(RC(:,2),'.')
xlabel('Days in Christchurch, NZ since August 2nd of 1943','FontSize',20);
ylabel('Rainfall in millimeters','FontSize',20)
```

Figure 5.11: Daily rainfalls in Christchurch since March 27 2010



Daily Temperatures in Christchurch

Labwork 58 Understand how Figure 5.12 is being generated by following the comments in the script file ChchTempsLoad.m by typing:

```
>> ChchTempsLoad
```

```
ChchTempsLoad.m
%% Load the data from the comma delimited text file 'NIWACliFloChchAeroClubStationTemps.txt'
%% with the following column IDs
%% Max_min: Daily Temperature in Christchurch New Zealand
%% Stationate(NZST),Tmax(C),Period(Hrs),Tmin(C),Period(Hrs),Tgmin(C),Period(Hrs),Tmean(C),RHmean(%),Period(Hrs)

% the matrix T is about to be assigned the data as a matrix; the option [27,1,20904,5] to
% specify the upper-left and lower-right corners of an imaginary rectangle
% over the text file 'NIWACliFloChchAeroClubStationTemps.txt'.
% here we start from line number 27 and end at the last line number 20904
% and we read only columns NZST,Tmax(C),Period(Hrs),Tmin(C),Period(Hrs)

T = dlmread('NIWACliFloChchAeroClubStationTemps.txt',',',[27,1,20904,5]);
% just keep column 1,2 and 4 named NZST,Tmax(C),Period(Hrs),Tmin(C),
% i.e. date in YYYYMMDD foramt, maximum temperature, minimum temperature
T = T(:,[1,2,4]); % just pull the time
% print size before removing missig data rows are removed
size(T) % report the dimensions or size of the matrix T

% This file has a lot of missing data points and they were replaced with
% NaN values - see the file for various manipulations that were done to the
% raw text file from NIWA (Copyright NIWA 2011 Subject to NIWA's Terms and
% Conditions. See: http://cliflo.niwa.co.nz/pls/niwp/doc/terms.html)
T(any(isnan(T),2),:) = [];% Remove any rows containing NaNs from a matrix

size(T) % if the matrix has a different size now then the data-less days now!
```

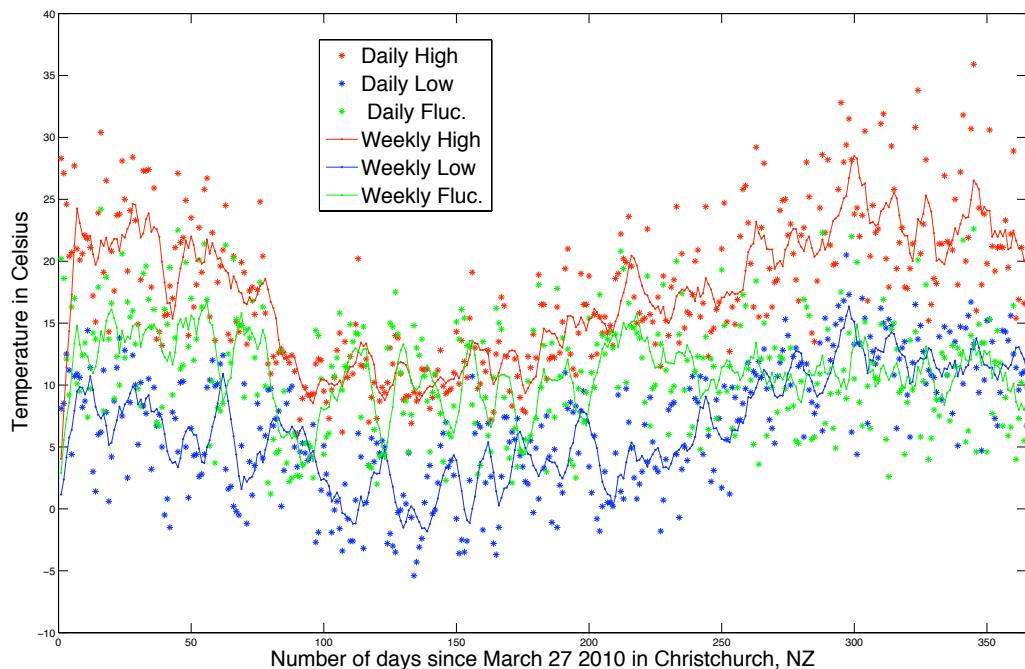
```

clf % clears all current figures

% Daily max and min temperature in the 100 days with good data
% before last date in this data, i.e., March 27 2011 in Christchurch NZ
H365Days = T(end-365:end,2);
L365Days = T(end-365:end,3);
F365Days = H365Days-L365Days; % assign the maximal fluctuation, i.e. max-min
plot(H365Days,'r*') % plot daily high or maximum temperature = Tmax
hold on; % hold the Figure so that we can overlay more plots on it
plot(L365Days,'b*') % plot daily low or minimum temperature = Tmin
plot(F365Days, 'g*') % plot daily Fluctuation = Tmax - Tmin
% filter for running means
windowSize = 7;
WeeklyHighs = filter(ones(1,windowSize)/windowSize,1,H365Days);
plot(WeeklyHighs,'r.-')
WeeklyLows = filter(ones(1,windowSize)/windowSize,1,L365Days);
plot(WeeklyLows,'b.-')
WeeklyFlucs = filter(ones(1,windowSize)/windowSize,1,F365Days);
plot(WeeklyFlucs,'g.-')
xlabel('Number of days since March 27 2010 in Christchurch, NZ','FontSize',20);
ylabel('Temperature in Celsius','FontSize',20)
MyLeg = legend('Daily High','Daily Low','Daily Fluc.', 'Weekly High','Weekly Low',...
    'Weekly Fluc.', 'Location','NorthEast')
% Create legend
% legend1 = legend(axes1,'show');
set(MyLeg,'FontSize',20);
xlim([0 365]); % set the limits or boundary on the x-axis of the plots
hold off % turn off holding so we stop overlaying new plots on this Figure

```

Figure 5.12: Daily temperatures in Christchurch for one year since March 27 2010



5.3.3 Textual Data

Processing and analysing textual data to make a decision is another important computational statistical experiment. An obvious example is machine translation and a less obvious one is exploratory data analysis of the textual content of

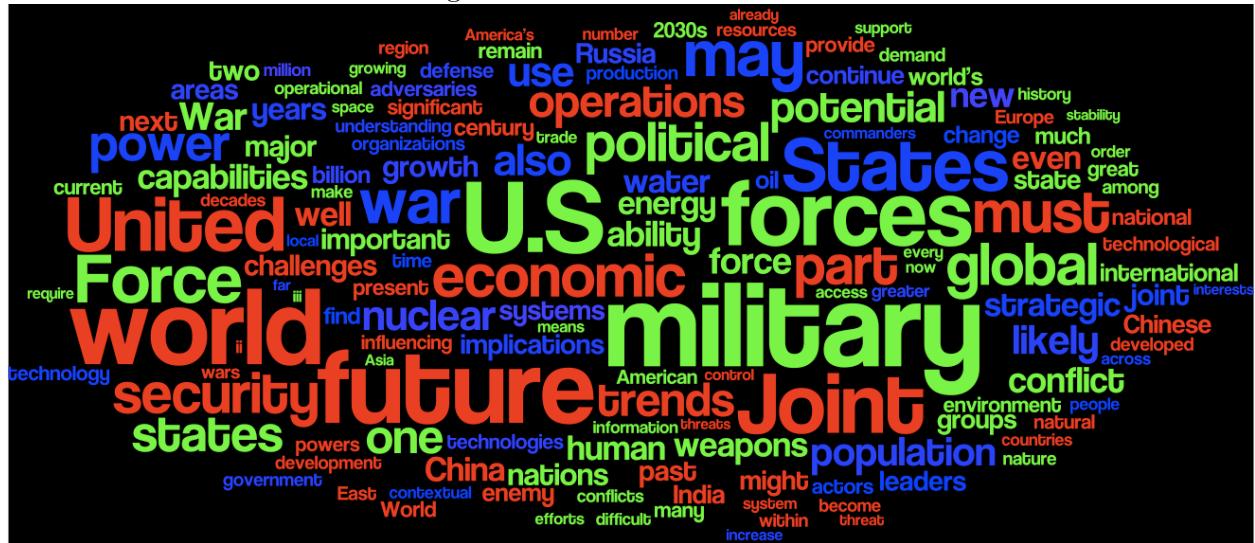
- a large document
 - twitter messages within an online social network of interest
 - etc.

An interesting document with a current affairs projection is the Joint Operating Environment 2010 Report by the US Department of Defense. This document was downloaded from http://www.jfcom.mil/newslink/storyarchive/2010/JOE_2010_o.pdf. The first paragraph of this 74 page document (JOE 2010 Report) reads:

ABOUT THIS STUDY The Joint Operating Environment is intended to inform joint concept development and experimentation throughout the Department of Defense. It provides a perspective on future trends, shocks, contexts, and implications for future joint force commanders and other leaders and professionals in the national security field. This document is speculative in nature and does not suppose to predict what will happen in the next twenty-five years. Rather, it is intended to serve as a starting point for discussions about the future security environment at the operational level of war. Inquiries about the Joint Operating Environment should be directed to USJFCOM Public Affairs, 1562 Mitscher Avenue, Suite 200, Norfolk, VA 23551-2488, (757) 836-6555.

Distribution Statement A: Approved for Public Release

Figure 5.13: Wordle of JOE 2010



We can try to produce a statistic of this document by recording the frequency of words in its textual content. Then we can produce a “word histogram” or “word cloud” to explore the document visually at one of the coarsest possible resolutions of the textual content in the JOE 2010 Report. The “word cloud” shown in Figure 5.13 was produced by Phillip Wilson using *wordle* from <http://www.wordle.net/>. A description from the wordle URL says:

Wordle is a toy for generating word clouds from text that you provide. The clouds give greater prominence to words that appear more frequently in the source text. You can tweak your clouds with different fonts, layouts, and color schemes. The images you create with Wordle are yours to use however you like. You can print them out, or save them to the Wordle gallery to share with your friends.

Labwork 59 (favourite word cloud) This is just for fun. Produce a “word cloud” of your honours thesis or summer project or any other document that fancies your interest by using *wordle* from <http://www.wordle.net/>. Play with the aesthetic features to change colour, shapes, etc.

5.3.4 Machine Sensor Data

Instrumentation of modern machines, such as planes, rockets and cars allow the sensors in the machines to collect live data and dynamically take *decisions* and subsequent *actions* by executing algorithms to drive their devices in response to the data that is streaming into their sensors. For example, a rocket may have to adjust its boosters to compensate for the prevailing directional changes in wind in order to keep going up and launch a satellite. These types of decisions and actions, theorised by *controlled Markov processes*, typically arise in various fields of engineering such as, aerospace, civil, electrical, mechanical, robotics, etc.

In an observational setting, without an associated control problem, one can use machine sensor data to get information about some state of the system or phenomenon, i.e., what is it doing? or where is it?, etc. Sometimes sensors are attached to a sample of individuals from a wild population, say Emperor Penguins in Antarctica where the phenomenon of interest may be the diving habits of this species after the eggs hatch. As an other example we can attach sensors to a double pendulum and find what it is doing when we give it a spin.

Based on such observational data the experimenter typically tries to learn about the behaviour of the system from the sensor data to estimate parameters, test hypotheses, etc. Such types of experiments are typically performed by scientists in various fields of science, such as, astronomy, biology, chemistry, geology, physics, etc.

Chaotic Time Series of a Double Pendulum

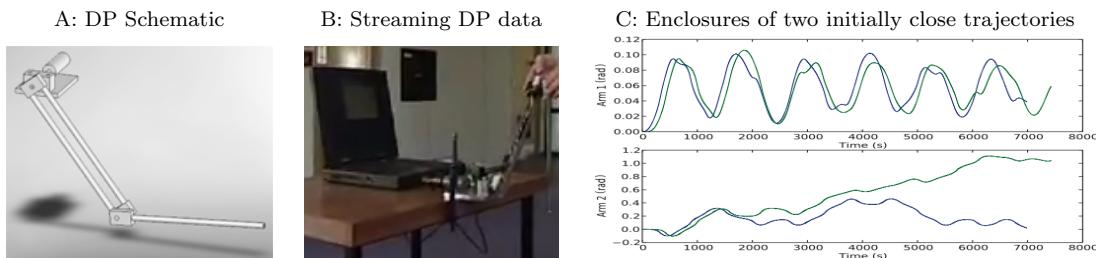


Figure 5.14: Double Pendulum

Sensors called *optical encoders* have been attached to the top end of each arm of a chaotic double pendulum in order to obtain the angular position of each arm through time as shown in Figure 5.14. Time series of the angular position of each arm for two trajectories that were initialized very similarly, say the angles of each arm of the double pendulum are almost the same at the initial time of release. Note how quickly the two trajectories diverge! System with such a sensitivity to initial conditions are said to be *chaotic*.

Labwork 60 (A Challenging Task) Try this if you are interested. Read any of the needed details about the design and fabrication of the double pendulum at <http://www.math.canterbury.ac.nz/~r.sainudiin/lmse/double-pendulum/>. Then use MATLAB to generate a plot similar to Figure 5.14(C) using time series data of trajectory 1 and trajectory 2 linked from the bottom of the above URL.

5.3.5 Biological Data



Chapter 6

Common Random Variables

The Uniform(0, 1) RV of Model 3 forms the foundation for random variate generation and simulation. This is appropriately called the fundamental model or experiment, since every other experiment can be obtained from this one.

Next, we simulate or generate samples from other RVs by making the following two assumptions:

1. independent samples from the Uniform(0, 1) RV can be generated, and
2. real arithmetic can be performed exactly in a computer.

Both these assumptions are, in fact, not true and require a more careful treatment of the subject. We may return to these careful treatments later on.

6.1 Inversion Sampler for Continuous Random Variables

Proposition 37 (Inversion sampler) Let $F(x) := \int_{-\infty}^x f(y) dy : \mathbb{R} \rightarrow [0, 1]$ be a continuous DF with density f , and let its inverse $F^{[-1]} : [0, 1] \rightarrow \mathbb{R}$ be:

$$F^{[-1]}(u) := \inf\{x : F(x) = u\}.$$

Then, $F^{[-1]}(U)$ has the distribution function F , provided U is a Uniform(0, 1) RV. Recall $\inf(A)$ or infimum of a set A of real numbers is the greatest lower bound of every element of A .

Proof: The “one-line proof” of the proposition is due to the following equalities:

$$\mathbf{P}(F^{[-1]}(U) \leq x) = \mathbf{P}(\inf\{y : F(y) = U\} \leq x) = \mathbf{P}(U \leq F(x)) = F(x), \quad \text{for all } x \in \mathbb{R}.$$

This yields the inversion sampler or the inverse (C)DF sampler, where we (i) *generate* $u \sim \text{Uniform}(0, 1)$ and (ii) *return* $x = F^{[-1]}(u)$, as formalised by the following algorithm.

This algorithm emphasises the fundamental sampler’s availability in an *input* step, and its set-up needs in an *initialise* step. In the following sections, we will not mention these universal steps; they will be taken for granted. The direct applicability of Algorithm 3 is limited to univariate densities for which the inverse of the cumulative distribution function is explicitly known. The next section will consider some examples.

Algorithm 3 Inversion Sampler or Inverse (C)DF Sampler

-
- 1: *input*: (1) $F^{[-1]}(x)$, inverse of the DF of the target RV X , (2) the fundamental sampler
 - 2: *initialise*: set the seed, if any, for the fundamental sampler
 - 3: *output*: a sample from X distributed according to F
 - 4: *draw* $u \sim \text{Uniform}(0, 1)$
 - 5: *return*: $x = F^{[-1]}(u)$
-

6.2 Some Simulations of Continuous Random Variables

6.3 Continuous Random Variables

Model 4 ($\text{Uniform}(\theta_1, \theta_2)$) Given two real parameters $\theta_1, \theta_2 \in \mathbb{R}$, such that $\theta_1 < \theta_2$, the PDF of the $\text{Uniform}(\theta_1, \theta_2)$ RV X is:

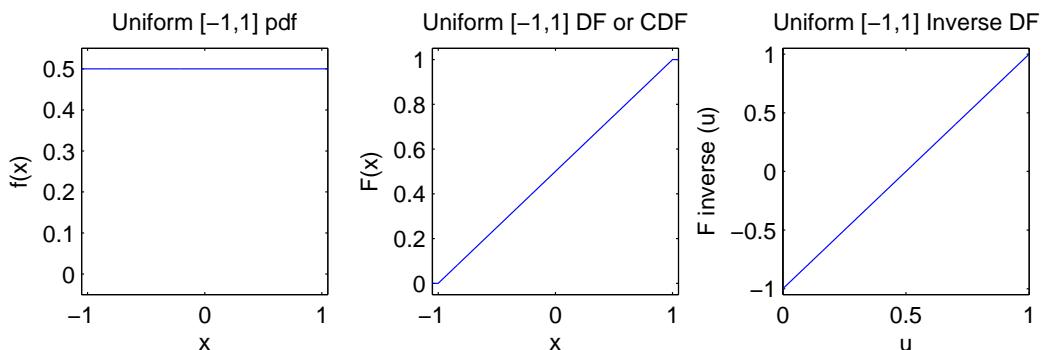
$$f(x; \theta_1, \theta_2) = \begin{cases} \frac{1}{\theta_2 - \theta_1} & \text{if } \theta_1 \leq x \leq \theta_2, \\ 0 & \text{otherwise} \end{cases} \quad (6.1)$$

and its DF given by $F(x; \theta_1, \theta_2) = \int_{-\infty}^x f(y; \theta_1, \theta_2) dy$ is:

$$F(x; \theta_1, \theta_2) = \begin{cases} 0 & \text{if } x < \theta_1 \\ \frac{x - \theta_1}{\theta_2 - \theta_1} & \text{if } \theta_1 \leq x \leq \theta_2, \\ 1 & \text{if } x > \theta_2 \end{cases} \quad (6.2)$$

Recall that we emphasise the dependence of the probabilities on the two parameters θ_1 and θ_2 by specifying them following the semicolon in the argument for f and F .

Figure 6.1: A plot of the PDF, DF or CDF and inverse DF of the $\text{Uniform}(-1, 1)$ RV X .

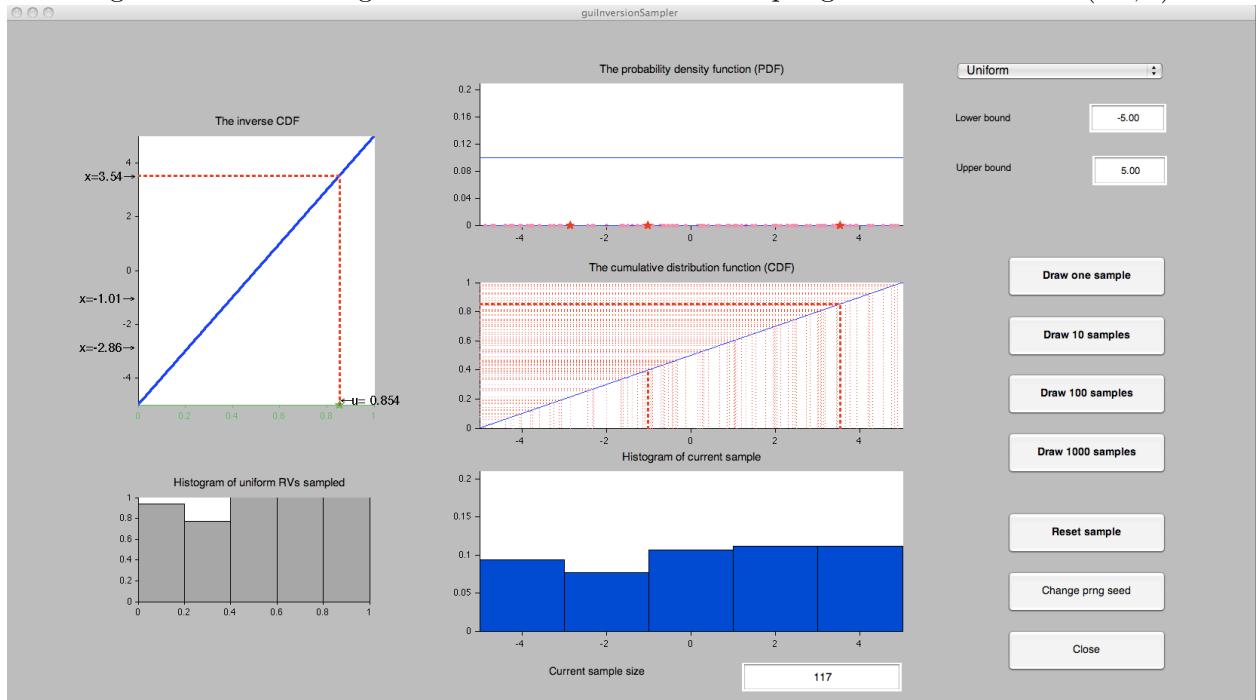


Labwork 61 (Inversion Sampler Demo – $\text{Uniform}(-5, 5)$) Let us comprehend the inversion sampler by calling the interactive visual cognitive tool built by Jennifer Harlow under a grant from University of Canterbury's Centre for Teaching and Learning (UCTL):

```
>> guiInversionSampler
```

The M-file `guiInversionSampler.m` will bring a graphical user interface (GUI) as shown in Figure 6.2. The default target distribution is $\text{Uniform}(-5, 5)$. Now repeatedly push the “Draw one sample” button several times and comprehend the simulation process. You can press “Draw 100 samples” to really comprehend the inversion sampler in action after 100 samples are drawn and depicted in the density histogram of the accumulating samples. Next try changing the numbers in the “Lower bound” and “Upper bound” boxes in order to alter the parameters θ_1 and θ_2 of $\text{Uniform}(\theta_1, \theta_2)$ RV.

Figure 6.2: Visual Cognitive Tool GUI: Inversion Sampling from $X \sim \text{Uniform}(-5, 5)$.



Simulation 62 ($\text{Uniform}(\theta_1, \theta_2)$) To simulate from $\text{Uniform}(\theta_1, \theta_2)$ RV X using the Inversion Sampler, we first need to find $F^{[-1]}(u)$ by solving for x in terms of $u = F(x; \theta_1, \theta_2)$:

$$u = \frac{x - \theta_1}{\theta_2 - \theta_1} \iff x = (\theta_2 - \theta_1)u + \theta_1 \iff F^{[-1]}(u; \theta_1, \theta_2) = \theta_1 + (\theta_2 - \theta_1)u$$

Here is a simple implementation of the Inversion Sampler for the $\text{Uniform}(\theta_1, \theta_2)$ RV in MATLAB :

```
>> rand('twister',786); % initialise the fundamental sampler for Uniform(0,1)
>> theta1=-1; theta2=1; % declare values for parameters theta1 and theta2
>> u=rand; % rand is the Fundamental Sampler and u is a sample from it
>> x=theta1+(theta2 - theta1)*u; % sample from Uniform(-1,1] RV
>> disp(x); % display the sample from Uniform[-1,,1] RV
0.5134
```

It is just as easy to draw n IID samples from $\text{Uniform}(\theta_1, \theta_2)$ RV X by transforming n IID samples from the $\text{Uniform}(0, 1)$ RV as follows:

```
>> rand('twister',786543); % initialise the fundamental sampler
>> theta1=-83; theta2=1004; % declare values for parameters a and b
>> u=rand(1,5); % now u is an array of 5 samples from Uniform(0,1)
```

```
>> x=theta1+(theta2 - theta1)*u; % x is an array of 5 samples from Uniform(-83,1004]) RV
>> disp(x); % display the 5 samples just drawn from Uniform(-83,1004) RV
465.3065 111.4994 14.3535 724.8881 254.0168
```

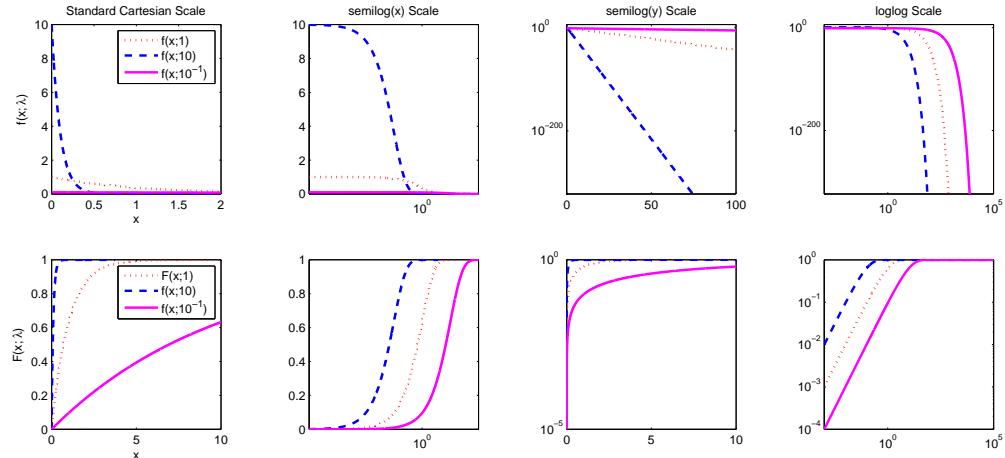
Model 5 ($\text{Exponential}(\lambda)$) For a given $\lambda > 0$, an $\text{Exponential}(\lambda)$ RV has the following PDF f and DF F :

$$f(x; \lambda) = \lambda e^{-\lambda x} \quad F(x; \lambda) = 1 - e^{-\lambda x}. \quad (6.3)$$

This distribution is fundamental because of its property of **memorylessness** and plays a fundamental role in continuous time processes as we will see later.

We encode the PDF and DF of the $\text{Exponential}(\lambda)$ RV as MATLAB functions `ExponentialPdf` and `ExponentialCdf` and use them to produce Figure 6.3 in Labwork 246.

Figure 6.3: Density and distribution functions of $\text{Exponential}(\lambda)$ RVs, for $\lambda = 1, 10, 10^{-1}$, in four different axes scales.



Mean and Variance of $\text{Exponential}(\lambda)$: Show that the mean of an $\text{Exponential}(\lambda)$ RV X is:

$$\mathbf{E}_\lambda(X) = \int_0^\infty x f(x; \lambda) dx = \int_0^\infty x \lambda e^{-\lambda x} dx = \frac{1}{\lambda},$$

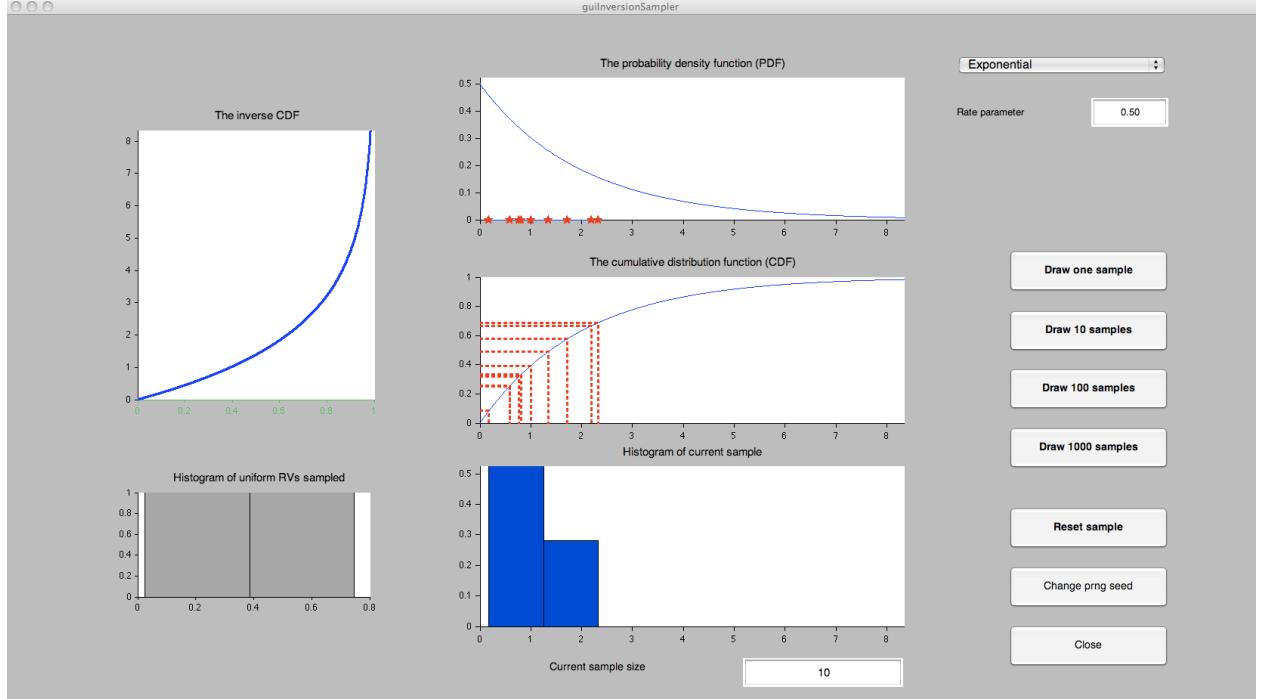
and the variance is:

$$\mathbf{V}_\lambda(X) = \left(\frac{1}{\lambda}\right)^2.$$

Labwork 63 (Inversion Sampler Demo – $\text{Exponential}(0.5)$) Let us understand the inversion sampler by calling the interactive visual cognitive tool:

```
>> guiInversionSampler
```

The M-file `guiInversionSampler.m` will bring a graphical user interface (GUI) as shown in Figure 6.4. First change the target distribution from the default $\text{Uniform}(-5, 5)$ to $\text{Exponential}(0.5)$ from the drop-down menu. Now push the “Draw 10 samples” button and comprehend the simulation process. Next try changing the “Rate Parameter” from 0.5 to 10.0 for example and generate several inversion samples and see the density histogram of the accumulating samples. You can press “Draw one sample” to really comprehend the inversion sampler in action one step at a time.

Figure 6.4: Visual Cognitive Tool GUI: Inversion Sampling from $X \sim \text{Exponential}(0.5)$.

Let us consider the problem of simulating from an $\text{Exponential}(\lambda)$ RV with realisations in $\mathbb{R}_+ := [0, \infty) := \{x : x \geq 0, x \in \mathbb{R}\}$ to model the waiting time for a bus at a bus stop.

Simulation 64 ($\text{Exponential}(\lambda)$) For a given $\lambda > 0$, an $\text{Exponential}(\lambda)$ RV has the following PDF f , DF F and inverse DF $F^{[-1]}$:

$$f(x; \lambda) = \lambda e^{-\lambda x} \quad F(x; \lambda) = 1 - e^{-\lambda x} \quad F^{[-1]}(u; \lambda) = \frac{-1}{\lambda} \log_e(1 - u) \quad (6.4)$$

We write the natural logarithm \log_e as log for notational simplicity. An implementation of the Inversion Sampler for $\text{Exponential}(\lambda)$ as a function in the M-file:

```
function x = ExpInvCDF(u,lambda);
% Return the Inverse CDF of Exponential(lambda) RV X
% Call Syntax: x = ExpInvCDF(u,lambda);
%             ExpInvCDF(u,lambda);
% Input      : lambda = rate parameter,
%             u = array of numbers in [0,1]
% Output     : x
x=-(1/lambda) * log(1-u);
```

We can simply call the function to draw a sample from, say the $\text{Exponential}(\lambda = 1.0)$ RV by:

```
lambda=1.0; % some value for lambda
u=rand; % rand is the Fundamental Sampler
ExpInvCDF(u,lambda) % sample from Exponential(1) RV via function in ExpInvCDF.m
```

Because of the following:

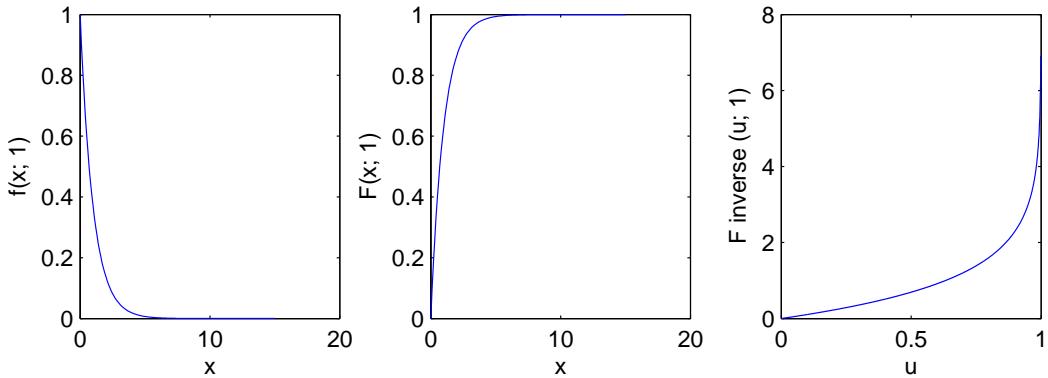
$$U \sim \text{Uniform}(0, 1) \implies -U \sim \text{Uniform}(-1, 0) \implies 1 - U \sim \text{Uniform}(0, 1),$$

we could save a subtraction operation in the above algorithm by replacing $-(1/\lambda) * \log(1-u)$ by $-(1/\lambda) * \log(u)$. This is implemented as the following function.

```
function x = ExpInvSam(u,lambda);
% Return the Inverse CDF based Sample from Exponential(lambda) RV X
% Call Syntax: x = ExpInvSam(u,lambda);
%               or ExpInvSam(u,lambda);
% Input      : lambda = rate parameter,
%               u = array of numbers in [0,1] from Uniform[0,1] RV
% Output     : x
x=-(1/lambda)*log(u);
```

```
>> rand('twister',46678); % initialise the fundamental sampler
>> Lambda=1.0; % declare Lambda=1.0
>> x=ExpInvSam(rand(1,5),Lambda); % pass an array of 5 Uniform(0,1) samples from rand
>> disp(x); % display the Exponential(1.0) distributed samples
    0.5945    2.5956    0.9441    1.9015    1.3973
```

Figure 6.5: The PDF f , DF F , and inverse DF $F^{[-1]}$ of the the Exponential($\lambda = 1.0$) RV.



It is straightforward to do replicate experiments. Consider the experiment of drawing five independent samples from the Exponential($\lambda = 1.0$) RV. Suppose we want to repeat or replicate this experiment seven times and find the sum of the five outcomes in each of these replicates. Then we may do the following:

```
>> rand('twister',1973); % initialise the fundamental sampler
>> % store 7 replications of 5 IID draws from Exponential(1.0) RV in array a
>> lambda=1.0; a= -1/lambda * log(rand(5,7)); disp(a);
    0.7267    0.3226    1.2649    0.4786    0.3774    0.0394    1.8210
    1.2698    0.4401    1.6745    1.4571    0.1786    0.4738    3.3690
    0.4204    0.1219    2.2182    3.6692    0.9654    0.0093    1.7126
    2.1427    0.1281    0.8500    1.4065    0.1160    0.1324    0.2635
    0.6620    1.1729    0.6301    0.6375    0.3793    0.6525    0.8330
>> %sum up the outcomes of the sequence of 5 draws in each replicate
>> s=sum(a); disp(s);
    5.2216    2.1856    6.6378    7.6490    2.0168    1.3073    7.9990
```

Labwork 65 (Next seven buses at your bus-stop) Consider the problem of modelling the arrival of buses at a bus stop. Suppose that the time between arrivals is an Exponential($\lambda = 0.1$) RV X with a mean inter-arrival time of $1/\lambda = 10$ minutes. Suppose you go to your bus stop and

zero a stop-watch. Simulate the times of arrival for the next seven buses as indicated by your stop-watch. Seed the fundamental sampler by your Student ID (eg. if your ID is 11424620 then type `rand('twister', 11424620)`; just before the simulation). Hand in the code with the arrival times of the next seven buses at your ID-seeded bus stop.

The support of the $\text{Exponential}(\lambda)$ RV is $\mathbb{R}_+ := [0, \infty)$. Let us consider a RV built by mirroring the $\text{Exponential}(\lambda)$ RV about the origin with the entire real line as its support.

Model 6 ($\text{Laplace}(\lambda)$ or Double $\text{Exponential}(\lambda)$ RV) If a RV X is equally likely to be either positive or negative with an exponential density, then the $\text{Laplace}(\lambda)$ or Double $\text{Exponential}(\lambda)$ RV, with the rate parameter $\lambda > 0, \lambda \in \mathbb{R}$, may be used to model it. The density function for the $\text{Laplace}(\lambda)$ RV given by $f(x; \lambda)$ is

$$f(x; \lambda) = \frac{\lambda}{2} e^{-\lambda|x|} = \begin{cases} \frac{\lambda}{2} e^{\lambda x} & \text{if } x < 0 \\ \frac{\lambda}{2} e^{-\lambda x} & \text{if } x \geq 0 \end{cases}. \quad (6.5)$$

Let us define the sign of a real number x by

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

Then, the DF of the $\text{Laplace}(\lambda)$ RV X is

$$F(x; \lambda) = \int_{-\infty}^x f(y; \lambda) dy = \frac{1}{2} \left(1 + \text{sign}(x) \left(1 - e^{-\lambda|x|} \right) \right), \quad (6.6)$$

and its inverse DF is

$$F^{[-1]}(u; \lambda) = -\frac{1}{\lambda} \text{sign} \left(u - \frac{1}{2} \right) \log \left(1 - 2 \left| u - \frac{1}{2} \right| \right), \quad u \in [0, 1] \quad (6.7)$$

Mean and Variance of $\text{Laplace}(\lambda)$ RV X : Show that the mean of a $\text{Laplace}(\lambda)$ RV X is

$$\mathbf{E}(X) = \int_0^\infty x f(x; \lambda) dx = \int_0^\infty x \frac{\lambda}{2} e^{-\lambda|x|} dx = 0,$$

and the variance is

$$\mathbf{V}(X) = \left(\frac{1}{\lambda} \right)^2 + \left(\frac{1}{\lambda} \right)^2 = 2 \left(\frac{1}{\lambda} \right)^2.$$

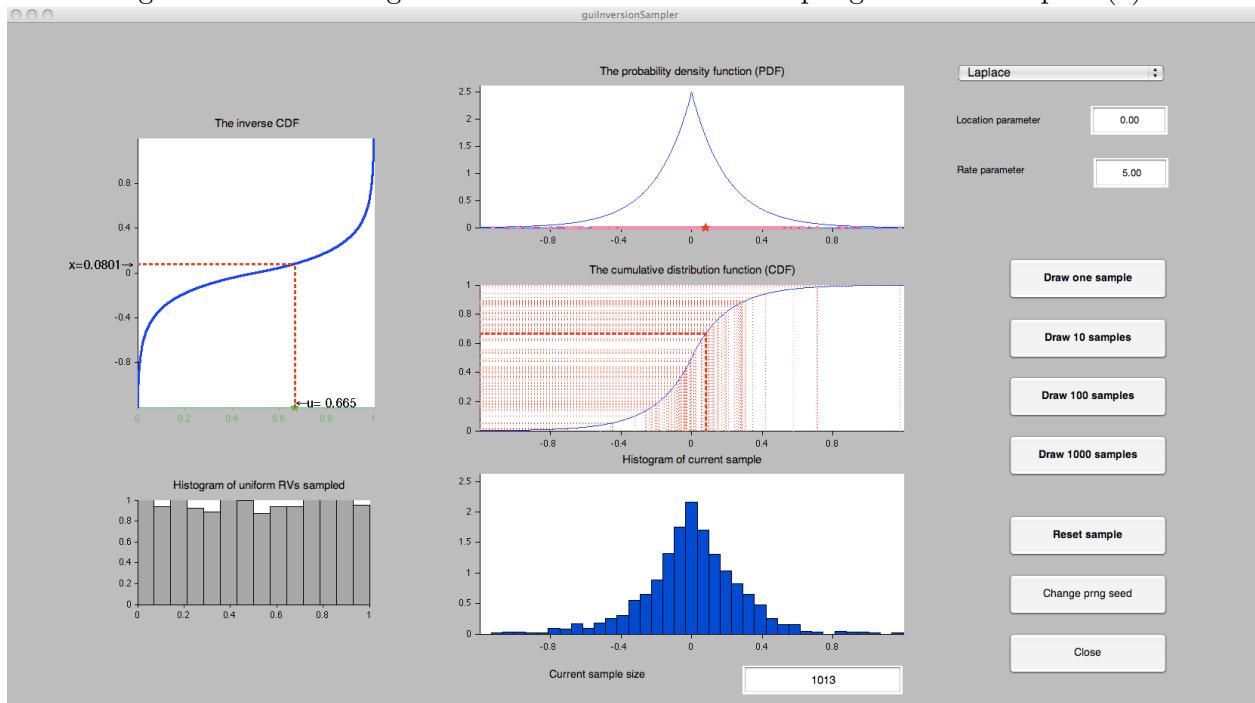
Note that the mean is 0 due to the symmetry of the density about 0 and the variance is twice that of the $\text{Exponential}(\lambda)$ RV.

Labwork 66 (Rejection Sampler Demo – $\text{Laplace}(5)$) Let us comprehend the rejection sampler by calling the interactive visual cognitive tool:

```
>> guiInversionSampler
```

The M-file `guiInversionSampler.m` will bring a graphical user interface (GUI) as shown in Figure 6.6. Using the drop-down menu change from the default target distribution $\text{Uniform}(-5, 5)$ to $\text{Laplace}(5)$. Now repeatedly push the “Draw one sample” button several times and comprehend the simulation process. You can also press “Draw 1000 samples” and see the density histogram of the generated samples. Next try changing the numbers in the “Rate parameter” box from 5.00 to 1.00 in order to alter the parameter λ of $\text{Laplace}(\lambda)$ RV. If you are more adventurous then try to alter the number in the “Location parameter” box from 0.00 to some thing else, say 10.00. Although our formulation of $\text{Laplace}(\lambda)$ implicitly had a location parameter of 0.00, we can easily introduce a location parameter μ into the PDF. With a pencil and paper try to rewrite the PDF in (6.5) with an additional location parameter μ .

Figure 6.6: Visual Cognitive Tool GUI: Inversion Sampling from $X \sim \text{Laplace}(5)$.



Simulation 67 ($\text{Laplace}(\lambda)$) Here is an implementation of an inversion sampler to draw IID samples from a $\text{Laplace}(\lambda)$ RV X by transforming IID samples from the $\text{Uniform}(0, 1)$ RV U :

```
LaplaceInvCDF.m
function x = LaplaceInvCDF(u,lambda);
% Call Syntax: x = LaplaceInvCDF(u,lambda);
%               or LaplaceInvCDF(u,lambda);
%
% Input      : lambda = rate parameter > 0,
%               u = an 1 X n array of IID samples from Uniform[0,1] RV
% Output     : an 1Xn array x of IID samples from Laplace(lambda) RV
%               or Inverse CDF of Laplace(lambda) RV
x=-(1/lambda)*sign(u-0.5) .* log(1-2*abs(u-0.5));
```

We can simply call the function to draw a sample from, say the $\text{Laplace}(\lambda = 1.0)$ RV by

```
>> lambda=1.0; % some value for lambda
>> rand('twister',6567); % initialize the fundamental sampler
```

```

>> u=rand(1,5); % draw 5 IID samples from Uniform(0,1) RV
>> disp(u);
0.6487 0.9003 0.3481 0.6524 0.8152

>> x=LaplaceInvCDF(u,lambda); % draw 5 samples from Laplace(1) RV using inverse CDF
>> disp(x);
0.3530 1.6127 -0.3621 0.3637 0.9953

```

Next, let us become familiar with an RV for which the expectation does not exist. This will help us appreciate the phrase “none of which is dominant” in the informal statement of the CLT later.

Model 7 (Cauchy) The density of the Cauchy RV X is:

$$f(x) = \frac{1}{\pi(1+x^2)}, \quad -\infty < x < \infty , \quad (6.8)$$

and its DF is:

$$F(x) = \frac{1}{\pi} \tan^{-1}(x) + \frac{1}{2} . \quad (6.9)$$

Randomly spinning a LASER emitting improvisation of “Darth Maul’s double edged lightsaber” that is centered at $(1, 0)$ in the plane \mathbb{R}^2 and recording its intersection with the y -axis, in terms of the y coordinates, gives rise to the *Standard Cauchy* RV.

Mean of Cauchy RV: The expectation of the Cauchy RV X , obtained via integration by parts (set $u = x$ and $v = \tan^{-1}(x)$) does not exist , since:

$$\int |x| dF(x) = \frac{2}{\pi} \int_0^\infty \frac{x}{1+x^2} dx = (x \tan^{-1}(x)]_0^\infty - \int_0^\infty \tan^{-1}(x) dx = \infty . \quad (6.10)$$

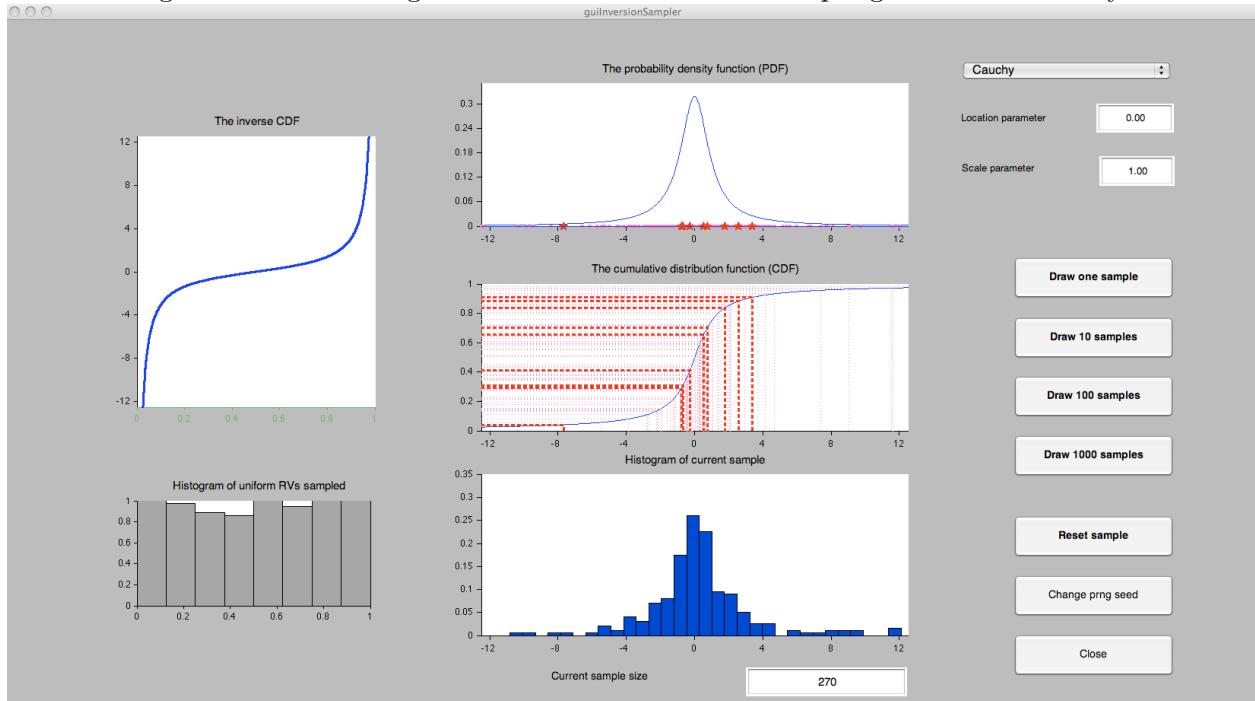
Variance and higher moments cannot be defined when the expectation itself is undefined.

Labwork 68 (Inversion Sampler Demo – Cauchy) Let us comprehend the inversion sampler by calling the interactive visual cognitive tool:

```
>> guiInversionSampler
```

The M-file `guiInversionSampler.m` will bring a graphical user interface (GUI) as shown in Figure 6.7. Using the drop-down menu change from the default target distribution $\text{Uniform}(-5, 5)$ to Cauchy. Now repeatedly push the “Draw one sample” button several times and comprehend the simulation process. You can also press “Draw 10 samples” several times and see the density histogram of the generated samples. Next try changing the numbers in the “Scale parameter” and “Location Parameter” boxes from the default values of 1.00 and 0.00, respectively. Although our formulation of Cauchy RV is also called *Standard Cauchy* as it implicitly had a location parameter of 0.00 and scale parameter of 1. With a pencil and paper (in conjunction with a wikipedia search if you have to) try to rewrite the PDF in (6.8) with an additional location parameter μ and scale parameter σ .

Simulation 69 (Cauchy) We can draw n IID samples from the Cauchy RV X by transforming n IID samples from $\text{Uniform}(0, 1)$ RV U using the inverse DF as follows:

Figure 6.7: Visual Cognitive Tool GUI: Inversion Sampling from $X \sim \text{Cauchy}$.

```
>> rand('twister',2435567); % initialise the fundamental sampler
>> u=rand(1,5); % draw 5 IID samples from Uniform(0,1) RV
>> disp(u);
    0.7176    0.6655    0.9405    0.9198    0.2598
>> x=tan(pi * u); % draw 5 samples from Standard cauchy RV using inverse CDF
>> disp(x); % display the samples in x
   -1.2272   -1.7470   -0.1892   -0.2575    1.0634
```

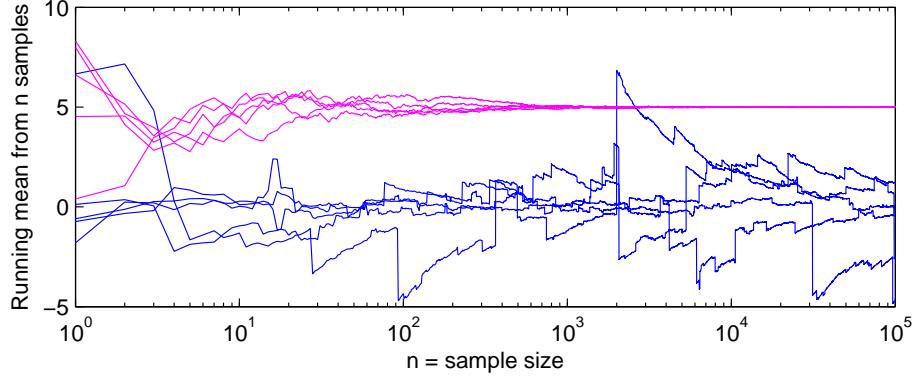
Recall that the mean of the Cauchy RV X does not exist since $\int |x| dF(x) = \infty$ (6.10). We will investigate this in Labwork 70.

Labwork 70 (Running mean of the Standard Cauchy RV) Let us see what happens when we plot the running sample mean for an increasing sequence of IID samples from the Standard Cauchy RV X by implementing the following script file:

```
% script to plot the oscillating running mean of Std Cauchy samples
% relative to those for the Uniform(0,10) samples
rand('twister',25567); % initialize the fundamental sampler
for i=1:5
N = 10^5; % maximum sample size
u=rand(1,N); % draw N IID samples from Uniform(0,1)
x=tan(pi * u); % draw N IID samples from Standard cauchy RV using inverse CDF
n=1:N; % make a vector n of current sample size [1 2 3 ... N-1 N]
CSx=cumsum(x); % CSx is the cumulative sum of the array x (type 'help cumsum')
% Running Means <- vector division of cumulative sum of samples by n
RunningMeanStdCauchy = CSx ./ n; % Running Mean for Standard Cauchy samples
RunningMeanUnif010 = cumsum(u*10.0) ./ n; % Running Mean for Uniform(0,10) samples
semilogx(n, RunningMeanStdCauchy) %
hold on;
semilogx(n, RunningMeanUnif010, 'm')
end
```

```
xlabel('n = sample size');
ylabel('Running mean from n samples')
```

Figure 6.8: Unending fluctuations of the running means based on n IID samples from the Standard Cauchy RV X in each of five replicate simulations (blue lines). The running means, based on n IID samples from the Uniform(0, 10) RV, for each of five replicate simulations (magenta lines).



The resulting plot is shown in Figure 6.8. Notice that the running means or the sample mean of n samples as a function of n , for each of the five replicate simulations, never settles down to a particular value. This is because of the “thick tails” of the density function for this RV which produces extreme observations. Compare them with the running means, based on n IID samples from the Uniform(0, 10) RV, for each of five replicate simulations (magenta lines). The latter sample means have settled down stably to the mean value of 5 after about 700 samples.

For a continuous RV X with a closed-form expression for the inverse DF $F^{[-1]}$, we can employ Algorithm 3 to draw samples from X . Table 6.1 summarises some random variables that are amenable to Algorithm 3.

Table 6.1: Some continuous RVs that can be simulated from using Algorithm 3.

Random Variable X	$F(x)$	$X = F^{[-1]}(U)$, $U \sim \text{Uniform}(0, 1)$	Simplified form
Uniform(a, b)	(6.2)	$a + (b - a)U$	–
Exponential(λ)	(6.3)	$\frac{-1}{\lambda} \log(1 - U)$	$\frac{-1}{\lambda} \log(U)$
Laplace(λ)	(6.7)	$-\frac{1}{\lambda} \text{sign}(U - \frac{1}{2}) \log(1 - 2 U - \frac{1}{2})$	–
Cauchy	(6.9)	$\tan(\pi(U - \frac{1}{2}))$	$\tan(\pi U)$

Next, we familiarise ourselves with the Gaussian or Normal RV.

Model 8 (Normal(μ, σ^2)) X has a Normal(μ, σ^2) or Gaussian(μ, σ^2) distribution with the location parameter $\mu \in \mathbb{R}$ and the scale or variance parameter $\sigma^2 > 0$, if:

$$f(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right), \quad x \in \mathbb{R} \quad (6.11)$$

Normal($0, 1$) distributed RV, which plays a fundamental role in asymptotic statistics, is conventionally denoted by Z . Z is said to have the **Standard Normal** distribution with PDF $f(z; 0, 1)$ and DF $F(z; 0, 1)$ conventionally denoted by $\varphi(z)$ and $\Phi(z)$, respectively.

There is no closed form expression for $\Phi(z)$ or $F(x; \mu, \sigma)$. The latter is simply defined as:

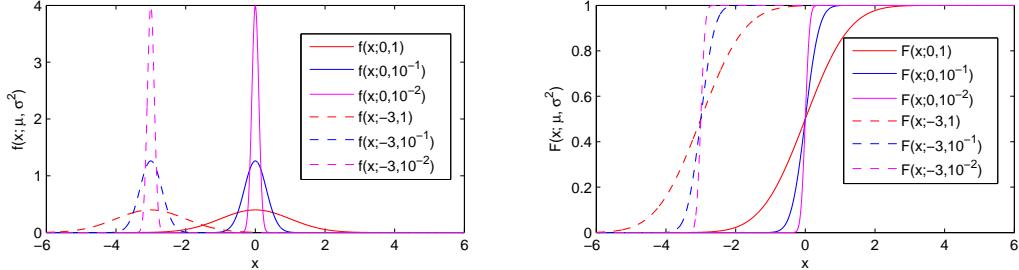
$$F(x; \mu, \sigma^2) = \int_{-\infty}^x f(y; \mu, \sigma) dy$$

We can express $F(x; \mu, \sigma^2)$ in terms of the error function (erf) as follows:

$$F(x; \mu, \sigma^2) = \frac{1}{2} \operatorname{erf}\left(\frac{x - \mu}{\sqrt{2\sigma^2}}\right) + \frac{1}{2} \quad (6.12)$$

We implement the PDF (6.11) and DF (6.12) for a $\text{Normal}(\mu, \sigma^2)$ RV X as MATLAB functions `NormalPdf` and `NormalCdf`, respectively, in Labwork 245, and then produce plots for various $\text{Normal}(\mu, \sigma^2)$ RVs, shown in Figure 6.9. Observe the concentration of probability mass, in terms of the PDF and DF plots, about the location parameter μ as the variance parameter σ^2 decreases.

Figure 6.9: Density and distribution function of several $\text{Normal}(\mu, \sigma^2)$ RVs.



Mean and Variance of $\text{Normal}(\mu, \sigma^2)$: The mean of a $\text{Normal}(\mu, \sigma^2)$ RV X is:

$$\mathbf{E}(X) = \int_{-\infty}^{\infty} xf(x; \mu, \sigma^2) dx = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} x \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right) dx = \mu ,$$

and the variance is:

$$\mathbf{V}(X) = \int_{-\infty}^{\infty} (x - \mu)^2 f(x; \mu, \sigma^2) dx = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} (x - \mu)^2 \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right) dx = \sigma^2 .$$

Labwork 71 (Compute the the $\mathbf{P}(X \in (a, b))$ for the $\text{Normal}(0, 1)$ RV X) Write a function to evaluate the $\mathbf{P}(X \in (a, b))$ for the $\text{Normal}(0, 1)$ RV X for user-specified values of a and b . [Hint: one option is by making two calls to `NormalCdf` and doing one arithmetic operation.]

Simulations 62 and 64, 67 and 69 produce samples from a continuous RV X with a closed-form expression for the inverse DF $F^{[-1]}$ via Algorithm 3 (Table 6.1). But only a few RVs have an explicit $F^{[-1]}$. For example, $\text{Normal}(0, 1)$ RV does not have an explicit $F^{[-1]}$. Algorithm 4 is a more general but inexact method that relies on an approximate numerical solution of x , for a given u , that satisfies the equation $F(x) = u$.

Simulation 72 ($\text{Normal}(\mu, \sigma^2)$) We may employ Algorithm 4 to sample from the $\text{Normal}(\mu, \sigma^2)$ RV X using the following function.

Algorithm 4 Inversion Sampler by Numerical Solution of $F(X) = U$ via Newton-Raphson Method

-
- 1: *input:* $F(x)$, the DF of the target RV X
 - 2: *input:* $f(x)$, the density of X
 - 3: *input:* A reasonable **Stopping Rule**,
e.g. a specified tolerance $\epsilon > 0$ and a maximum number of iterations **MAX**
 - 4: *input:* a careful mechanism to specify x_0
 - 5: *output:* a sample from X distributed according to F
 - 6: *draw:* $u \sim \text{Uniform}(0, 1)$
 - 7: *initialise:* $i \leftarrow 0$, $x_i \leftarrow x_0$, $x_{i+1} \leftarrow x_0 - \frac{F(x_0) - u}{f(x_0)}$
 - 8: **while** Stopping Rule is not satisfied,
e.g. $|F(x_i) - F(x_{i-1})| > \epsilon$ AND $i < \text{MAX}$ **do**
 - 9: $x_i \leftarrow x_{i+1}$
 - 10: $x_{i+1} \leftarrow \left(x_i - \frac{F(x_i) - u}{f(x_i)} \right)$
 - 11: $i \leftarrow i + 1$
 - 12: **end while**
 - 13: *return:* $x \leftarrow x_i$
-

```
function x = Sample1NormalByNewRap(u,Mu,SigmaSq)
% Returns a sample from Normal(Mu, SigmaSq)
% Newton-Raphson numerical solution of F(x)=u
% Input: u = one random Uniform(0,1) sample
%         Mu = Mean of Normal(Mu, SigmaSq)
%         SigmaSq = Variance of Normal(Mu, SigmaSq)
% Usage: x = Sample1NormalByNewRap(u,Mu,SigmaSq)
% To transform an array Us of uniform samples to array Xs of Normal samples via arrayfun
%         Xs = arrayfun(@(u)(Sample1NormalByNewRap(u,-100.23,0.01)),Us);
Epsilon=1e-5; % Tolerance in stopping rule
MaxIter=10000; % Maximum allowed iterations in stopping rule
x=0; % initialize the output x as 0
% initialize i, xi, and xii
i=0; % Mu is an ideal initial condition since F(x; Mu, SigmaSq)
xi = Mu; % is convex when x < Mu and concave when x > Mu and the
% Newton-Raphson method started at Mu converges
xii = xi - (NormalCdf(xi,Mu,SigmaSq)-u)/NormalPdf(xi,Mu,SigmaSq);
% Newton-Raphson Iterations
while (abs(NormalCdf(xii,Mu,SigmaSq)-NormalCdf(xi,Mu,SigmaSq))...
    > Epsilon & i < MaxIter),
    xi = xii;
    xii = xii - (NormalCdf(xii,Mu,SigmaSq)-u)/NormalPdf(xii,Mu,SigmaSq);
    i=i+1;
end
x=xii; % record the simulated x from the j-th element of u
```

We draw five samples from the $\text{Normal}(0, 1)$ RV Z and store them in z as follows. The vector z can be obtained by a Newton-Raphson-based numerical transformation of the vector u of 5 IID samples from the $\text{Uniform}(0, 1)$ RV. We simply need to apply the function `Sample1NormalByNewRap` to each element of an array of $\text{Uniform}(0, 1)$ samples. MATLAB's `arrayfun` command can be used to apply `@(u)(Sample1NormalByNewRap(u,0,1))` (i.e., `Sample1NormalByNewRap` as a function of u) to every element of our array of $\text{Uniform}(0, 1)$ samples, say `Us`. Note that $F(z)$ is the same as the drawn u from U at least up to four significant digits.

```
>> rand('twister',563987);
>> Us=rand(1,5); % store 5 samples from Uniform(0,1) RV in array Us
```

```

>> disp(Us); % display Us
    0.8872    0.2569    0.5275    0.8650    0.8517
>> z=Sample1NormalByNewRap(Us(1),0,1); %transform Us(1) to a Normal(0,1) sample z
>> disp(z); % display z
    1.2119
>> z = arrayfun(@(u)(Sample1NormalByNewRap(u,0,1)),Us); %transform array Us via arrayfun
>> % display array z obtained from applying Sample1NormalByNewRap to each element of Us
>> disp(z);
    1.2119   -0.6530    0.0691    1.1031    1.0439
>> % check that numerical inversion of F worked, i.e., is F(z)=u ?
>> disp(NormalCdf(z,0,1));
    0.8872    0.2569    0.5275    0.8650    0.8517

```

Next we draw five samples from the $\text{Normal}(-100.23, 0.01)$ RV X , store it in an array x and observe that the numerical method is reasonably accurate by the equality of u and $F(x)$.

```

>> rand('twister',563987);
>> disp(Us); % display Us
    0.8872    0.2569    0.5275    0.8650    0.8517
>> % transform array Us via arrayfun
>> x = arrayfun(@(u)(Sample1NormalByNewRap(u,-100.23,0.01)),Us);
>> disp(x);
    -100.1088  -100.2953  -100.2231  -100.1197  -100.1256
>> disp(NormalCdf(x,-100.23,0.01));
    0.8872    0.2569    0.5275    0.8650    0.8517

```

One has to be extremely careful with this approximate simulation algorithm implemented in floating-point arithmetic. More robust samplers for the $\text{Normal}(\mu, \sigma^2)$ RV exist. However, Algorithm 4 is often the only choice when simulating from an arbitrary RV with an unknown closed-form expression for its $F^{[-1]}$.

Next, we use our simulation capability to gain an informal and intuitive understanding of one of the most elementary theorems in probability and statistics, namely, the Central Limit Theorem (CLT). We will see a formal treatment of CLT later.

Informally, the CLT can be stated as follows:

“The sample mean of a large number of IID samples, none of which is dominant, tends to the Normal distribution as the number of samples increases.”

Labwork 73 (Investigating the Central Limit Theorem with IID Exponential($\lambda = 0.1$) RVs)
 Let us investigate the histograms from 10000 simulations of the sample mean of $n = 10, 100, 1000$ IID Exponential($\lambda = 0.1$) RVs as follows:

```

>> rand('twister',1973); % initialise the fundamental sampler
>> % a demonstration of Central Limit Theorem (CLT) -- Details of CLT are in the sequel
>> % the sample mean should be a Normal(1/lambda,lambda/n) RV
>> lambda=0.1; Reps=10000; n=10; hist(sum(-1/lambda * log(rand(n,Reps)))/n)
>> lambda=0.1; Reps=10000; n=100; hist(sum(-1/lambda * log(rand(n,Reps)))/n,20)
>> lambda=0.1; Reps=10000; n=1000; hist(sum(-1/lambda * log(rand(n,Reps)))/n,20)

```

Do you see a pattern in the histograms?

See the histograms generated from the following code that produces sample means from the Cauchy RV:

```
>> Reps=10000; n=1000; hist(sum(tan(pi * rand(n,Reps)))/n,20)
>> Reps=10000; n=1000; hist(sum(tan(pi * rand(n,Reps)))/n,20)
>> Reps=10000; n=1000; hist(sum(tan(pi * rand(n,Reps)))/n,20)
```

Classwork 74 (Why doesn't the sample mean of the Cauchy RV ever settle down?) Explain in words why the mean of n IID samples from the Cauchy RV “is **not** obeying” the Central Limit Theorem. Also relate it to Figure 6.8 of Labwork 70.

Model 9 (Gamma(λ, k) RV) Given a shape parameter $\alpha > 0$ and a rate parameter $\beta > 0$, the RV X is said to be Gamma(α, β) distributed if its PDF is:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x), \quad x > 0 ,$$

where, the gamma function which interpolates the factorial function is:

$$\Gamma(\alpha) := \int_0^\infty \exp(-y) y^{\alpha-1} dy .$$

When $k \in \mathbb{N}$, then $\Gamma(k) = (k - 1)!$. The DF of X is:

$$F(x; \lambda, k) = \mathbf{1}_{\mathbb{R}_{>0}}(x) \frac{\beta^\alpha}{\Gamma(\alpha)} \int_0^x y^{\alpha-1} \exp(-\beta y) dy = \begin{cases} 0 & \text{if } x \leq 0 \\ \frac{\gamma(\alpha, \beta x)}{\Gamma(\alpha)} & \text{if } x > 0 \end{cases}$$

where $\gamma(\alpha, \beta x)$ is called the lower incomplete Gamma function.

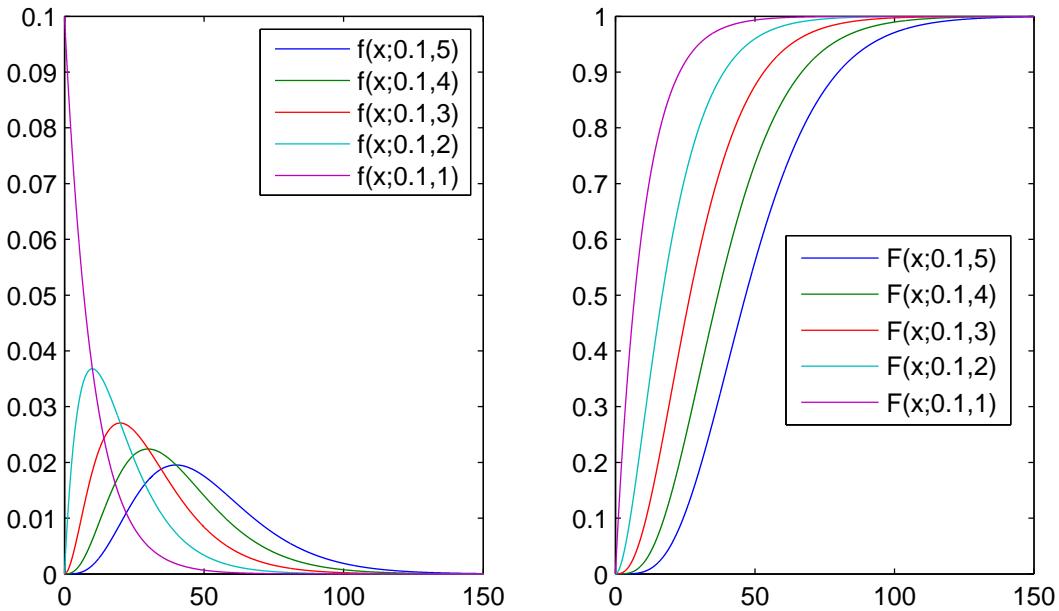
The expectation and variance of a Gamma(α, β) RV are α/β and α/β^2 , respectively. The Gamma function and the incomplete Gamma function are available as MATLAB functions `gamma` and `gammainc`, respectively. Thus, `gamma(k)` returns $\Gamma(k)$ and `gammainc(lambda*x, k)` returns $F(x; \lambda, k)$. Using these functions, it is straightforward to evaluate the PDF and CDF of $X \sim \text{Gamma}(\lambda, k)$. We use the following script to get a sense for the impact upon the PDF and CDF of the shape parameter k as it ranges in $\{1, 2, 3, 4, 5\}$ for a given scale parameter $\lambda = 0.1$.

PlotPdfCdfGamma.m

```
lambda=0.1; % choose some scale parameter
Xs=0:0.01:150; % choose some x values
% Plot PDFs for k=5,4,3,2,1
k=5; fXsk5=(1/gamma(k))*(lambda*exp(-lambda*Xs).*(lambda*Xs).^(k-1));% PDF for k=5
k=4; fXsk4=(1/gamma(k))*(lambda*exp(-lambda*Xs).*(lambda*Xs).^(k-1));% PDF for k=4
k=3; fXsk3=(1/gamma(k))*(lambda*exp(-lambda*Xs).*(lambda*Xs).^(k-1));% PDF for k=3
k=2; fXsk2=(1/gamma(k))*(lambda*exp(-lambda*Xs).*(lambda*Xs).^(k-1));% PDF for k=2
k=1; fXsk1=(1/gamma(k))*(lambda*exp(-lambda*Xs).*(lambda*Xs).^(k-1));% PDF for k=1
clf; % clear any previous figures
subplot(1,2,1); % make first PDF plot
plot(Xs,fXsk5, Xs, fXsk4, Xs, fXsk3, Xs, fXsk2, Xs, fXsk1)
legend('f(x;0.1,5)', 'f(x;0.1,4)', 'f(x;0.1,3)', 'f(x;0.1,2)', 'f(x;0.1,1)')
subplot(1,2,2) % make second CDF plots using MATLAB's gammaintc (incomplete gamma function)
plot(Xs,gammaintc(lambda*Xs,5), Xs,gammaintc(lambda*Xs,4), Xs,gammaintc(lambda*Xs,3),...
Xs,gammaintc(lambda*Xs,2), Xs,gammaintc(lambda*Xs,1))
legend('F(x;0.1,5)', 'F(x;0.1,4)', 'F(x;0.1,3)', 'F(x;0.1,2)', 'F(x;0.1,1)')
```

Note that if $X \sim \text{Gamma}(1, \beta)$ then $X \sim \text{Exponential}(\beta)$, since:

$$f(x; 1, \beta) = \frac{1}{(1-1)!} \beta \exp(-\beta x) = \beta \exp(-\beta x) .$$

Figure 6.10: PDF and CDF of $X \sim \text{Gamma}(\beta = 0.1, \alpha)$ with $\alpha \in \{1, 2, 3, 4, 5\}$.

More generally, if $X \sim \text{Gamma}(\alpha, \beta)$ and $\alpha \in \mathbb{N}$, then $X \sim \sum_{i=1}^{\alpha} Y_i$, where $Y_i \stackrel{IID}{\sim} \text{Exponential}(\beta)$ RVS, i.e. the sum of α IID $\text{Exponential}(\beta)$ RVs forms the model for the $\text{Gamma}(\alpha, \beta)$ RV. If you model the inter-arrival time of buses at a bus-stop by IID $\text{Exponential}(\beta)$ RV, then you can think of the arrival time of the k^{th} bus as a $\text{Gamma}(\alpha, \beta)$ RV.

6.4 Discrete Random Variables

6.5 Inversion Sampler for Discrete Random Variables

Next, consider the problem of **sampling from a random variable X with a discontinuous or discrete DF** using the inversion sampler. We need to define the inverse more carefully here.

Proposition 38 (Inversion sampler with compact support) Let the support of the RV X be over some real interval $[a, b]$ and let its inverse DF be defined as follows:

$$F^{[-1]}(u) := \inf\{x \in [a, b] : F(x) \geq u, 0 \leq u \leq 1\}.$$

If $U \sim \text{Uniform}(0, 1)$ then $F^{[-1]}(U)$ has the DF F , i.e. $F^{[-1]}(U) \sim F \sim X$.

Proof: The proof is a consequence of the following equalities:

$$\mathbf{P}(F^{[-1]}(U) \leq x) = \mathbf{P}(U \leq F(x)) = F(x) := \mathbf{P}(X \leq x)$$

6.6 Some Simulations of Discrete Random Variables

Simulation 75 (Bernoulli(θ)) Consider the problem of simulating from a $\text{Bernoulli}(\theta)$ RV based on an input from a $\text{Uniform}(0, 1)$ RV. Recall that $\lfloor x \rfloor$ (called the ‘floor of x ’) is the largest integer

that is smaller than or equal to x , e.g. $\lfloor 3.8 \rfloor = 3$. Using the floor function, we can simulate a Bernoulli(θ) RV X as follows:

```
>> theta = 0.3; % set theta = Prob(X=1)
% return x -- floor(y) is the largest integer less than or equal to y
>> x = floor(rand + theta); % rand is the Fundamental Sampler
>> disp(x) % display the outcome of the simulation
0
>> n=10; % set the number of IID Bernoulli(theta=0.3) trials you want to simulate
>> x = floor(rand(1,10)+theta); % vectorize the operation
>> disp(x) % display the outcomes of the simulation
0 0 1 0 0 0 0 0 1 1
```

Again, it is straightforward to do replicate experiments, e.g. to demonstrate the Central Limit Theorem for a sequence of n IID Bernoulli(θ) trials.

```
>> % a demonstration of Central Limit Theorem --
>> % the sample mean of a sequence of n IID Bernoulli(theta) RVs is Gaussian(theta,theta*(1-theta)/n)
>> theta=0.5; Reps=10000; n=10; hist(sum(floor(rand(n,Reps)+theta))/n)
>> theta=0.5; Reps=10000; n=100; hist(sum(floor(rand(n,Reps)+theta))/n,20)
>> theta=0.5; Reps=10000; n=1000; hist(sum(floor(rand(n,Reps)+theta))/n,30)
```

Consider the class of discrete RVs with distributions that place all probability mass on a single real number. This is the probability model for the deterministic real variable.

Model 10 (Point Mass(θ)) Given a specific point $\theta \in \mathbb{R}$, we say an RV X has point mass at θ or is Point Mass(θ) distributed if the DF is:

$$F(x; \theta) = \begin{cases} 0 & \text{if } x < \theta \\ 1 & \text{if } x \geq \theta \end{cases} \quad (6.13)$$

and the PMF is:

$$f(x; \theta) = \begin{cases} 0 & \text{if } x \neq \theta \\ 1 & \text{if } x = \theta \end{cases} \quad (6.14)$$

Thus, Point Mass(θ) RV X is deterministic in the sense that every realisation of X is exactly equal to $\theta \in \mathbb{R}$. We will see that this distribution plays a central limiting role in asymptotic statistics.

Mean and variance of Point Mass(θ) RV: Let $X \sim \text{Point Mass}(\theta)$. Then:

$$\mathbf{E}(X) = \sum_x x f(x) = \theta \times 1 = \theta, \quad \mathbf{V}(X) = \mathbf{E}(X^2) - (\mathbf{E}(X))^2 = \theta^2 - \theta^2 = 0.$$

Simulation 76 (Point Mass(θ)) Let us simulate a sample from the Point Mass(θ) RV X . Since this RV produces the same realisation θ we can implement it via the following M-file:

```
function x = Sim1PointMass(u,theta)
% Returns one sample from the Point Mass(theta) RV X
% Call Syntax: x = SimPointMass(u,theta);
% Input      : u = one uniform random number eg. rand()
%               theta = a real number (scalar)
% Output     : x = sample from X
x=theta;
```

Here is call to the function.

```
>> Sim1PointMass(rand(),2)
ans =
2
>> % we can use arrayfun to apply Sim1Pointmass to any array of Uniform(0,1) samples
>> arrayfun(@(u)(Sim1PointMass(u,17)),rand(2,10))
ans =
17    17    17    17    17    17    17    17    17    17
17    17    17    17    17    17    17    17    17    17
```

Note that it is not necessary to have input IID samples from Uniform(0, 1) RV via `rand` in order to draw samples from the Point Mass(θ) RV. For instance, an input matrix of zeros can do the job:

```
>> arrayfun(@(u)(Sim1PointMass(u,17)),zeros(2,8))
ans =
17    17    17    17    17    17    17    17
17    17    17    17    17    17    17    17
```

Next let us consider a natural generalization of the Bernoulli(θ) RV with more than two outcomes.

Model 11 (de Moivre($\theta_1, \theta_2, \dots, \theta_k$)) Given a specific point $(\theta_1, \theta_2, \dots, \theta_k)$ in the k -Simplex:

$$\Delta_k := \{ (\theta_1, \theta_2, \dots, \theta_k) : \theta_1 \geq 0, \theta_2 \geq 0, \dots, \theta_k \geq 0, \sum_{i=1}^k \theta_i = 1 \},$$

we say that an RV X is de Moivre($\theta_1, \theta_2, \dots, \theta_k$) distributed if its PMF is:

$$f(x; \theta_1, \theta_2, \dots, \theta_k) = \begin{cases} 0 & \text{if } x \notin [k] := \{1, 2, \dots, k\}, \\ \theta_x & \text{if } x \in [k]. \end{cases}$$

The DF for de Moivre($\theta_1, \theta_2, \dots, \theta_k$) RV X is:

$$F(x; \theta_1, \theta_2, \dots, \theta_k) = \begin{cases} 0 & \text{if } -\infty < x < 1 \\ \theta_1 & \text{if } 1 \leq x < 2 \\ \theta_1 + \theta_2 & \text{if } 2 \leq x < 3 \\ \vdots & \\ \theta_1 + \theta_2 + \dots + \theta_{k-1} & \text{if } k-1 \leq x < k \\ \theta_1 + \theta_2 + \dots + \theta_{k-1} + \theta_k = 1 & \text{if } k \leq x < \infty \end{cases} \quad (6.15)$$

The de Moivre($\theta_1, \theta_2, \dots, \theta_k$) RV can be thought of as a probability model for “the outcome of rolling a polygonal cylindrical die with k rectangular faces that are marked with $1, 2, \dots, k$ ”. The parameters $\theta_1, \theta_2, \dots, \theta_k$ specify how the die is loaded and may be idealised as specifying the cylinder’s centre of mass with respect to the respective faces. Thus, when $\theta_1 = \theta_2 = \dots = \theta_k = 1/k$, we have a probability model for the outcomes of a fair die.

Mean and variance of de Moivre($\theta_1, \theta_2, \dots, \theta_k$) RV: The not too useful expressions for the first two moments of $X \sim \text{de Moivre}(\theta_1, \theta_2, \dots, \theta_k)$ are,

$$\mathbf{E}(X) = \sum_{x=1}^k x\theta(x) = \theta_1 + 2\theta_2 + \dots + k\theta_k , \text{ and}$$

$$\mathbf{V}(X) = \mathbf{E}(X^2) - (\mathbf{E}(X))^2 = (\theta_1 + 2^2\theta_2 + \dots + k^2\theta_k) - (\theta_1 + 2\theta_2 + \dots + k\theta_k)^2 .$$

However, if $X \sim \text{de Moivre}(1/k, 1/k, \dots, 1/k)$, then the mean and variance for the fair k -faced die based on Faulhaber's formula for $\sum_{i=1}^k i^m$, with $m \in \{1, 2\}$, are,

$$\mathbf{E}(X) = \frac{1}{k} (1 + 2 + \dots + k) = \frac{1}{k} \frac{k(k+1)}{2} = \frac{k+1}{2} ,$$

$$\mathbf{E}(X^2) = \frac{1}{k} (1^2 + 2^2 + \dots + k^2) = \frac{1}{k} \frac{k(k+1)(2k+1)}{6} = \frac{2k^2 + 3k + 1}{6} ,$$

$$\begin{aligned} \mathbf{V}(X) &= \mathbf{E}(X^2) - (\mathbf{E}(X))^2 = \frac{2k^2 + 3k + 1}{6} - \left(\frac{k+1}{2}\right)^2 = \frac{2k^2 + 3k + 1}{6} - \left(\frac{k^2 + 2k + 1}{4}\right) \\ &= \frac{8k^2 + 12k + 4 - 6k^2 - 12k - 6}{24} = \frac{2k^2 - 2}{24} = \frac{k^2 - 1}{12} . \end{aligned}$$

Next we simulate from $\text{de Moivre}(\theta_1, \theta_2, \dots, \theta_k)$ RV X via its inverse DF

$$F^{[-1]} : [0, 1] \rightarrow [k] := \{1, 2, \dots, k\} ,$$

given by:

$$F^{[-1]}(u; \theta_1, \theta_2, \dots, \theta_k) = \begin{cases} 1 & \text{if } 0 \leq u < \theta_1 \\ 2 & \text{if } \theta_1 \leq u < \theta_1 + \theta_2 \\ 3 & \text{if } \theta_1 + \theta_2 \leq u < \theta_1 + \theta_2 + \theta_3 \\ \vdots & \\ k & \text{if } \theta_1 + \theta_2 + \dots + \theta_{k-1} \leq u < 1 \end{cases} \quad (6.16)$$

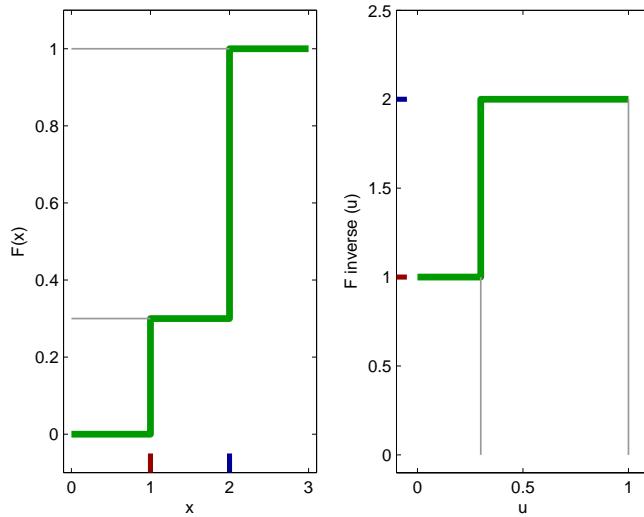
When $k = 2$ in the $\text{de Moivre}(\theta_1, \theta_2)$ model, we have an RV that is similar to the Bernoulli($p = \theta_1$) RV. The DF F and its inverse $F^{[-1]}$ for a specific $\theta_1 = 0.3$ are depicted in Figure 6.11.

First we simulate from an equi-probable special case of the $\text{de Moivre}(\theta_1, \theta_2, \dots, \theta_k)$ RV, with $\theta_1 = \theta_2 = \dots = \theta_k = 1/k$.

Simulation 77 ($\text{de Moivre}(1/k, 1/k, \dots, 1/k)$) The equi-probable $\text{de Moivre}(1/k, 1/k, \dots, 1/k)$ RV X with a discrete uniform distribution over $[k] = \{1, 2, \dots, k\}$ can be efficiently sampled using the ceiling function. Recall that $\lceil y \rceil$ is the smallest integer larger than or equal to y , eg. $\lceil 13.1 \rceil = 14$. Algorithm 5 produces samples from the $\text{de Moivre}(1/k, 1/k, \dots, 1/k)$ RV X .

The M-file implementing Algorithm 5 is:

```
function x = SimdeMoivreEqui(u,k);
% return samples from de Moivre(1/k,1/k,...,1/k) RV X
% Call Syntax: x = SimdeMoivreEqui(u,k);
% Input      : u = array of uniform random numbers eg. rand
%               k = number of equi-probable outcomes of X
% Output     : x = samples from X
x = ceil(k * u); % ceil(y) is the smallest integer larger than y
%x = floor(k * u); if outcomes are in {0,1,...,k-1}
```

Figure 6.11: The DF $F(x; 0.3, 0.7)$ of the de Moivre(0.3, 0.7) RV and its inverse $F^{-1}(u; 0.3, 0.7)$.**Algorithm 5** Inversion Sampler for de Moivre($1/k, 1/k, \dots, 1/k$) RV1: *input:*

1. k in de Moivre($1/k, 1/k, \dots, 1/k$) RV X
2. $u \sim \text{Uniform}(0, 1)$

2: *output:* a sample from X 3: *return:* $x \leftarrow \lceil ku \rceil$

Let us use the function `SimdeMoivreEqui` to draw five samples from a fair seven-faced cylindrical dice.

```
>> k=7; % number of faces of the fair dice
>> n=5; % number of trials
>> rand('twister',78657); % initialise the fundamental sampler
>> u=rand(1,n); % draw n samples from Uniform(0,1)
>> % inverse transform samples from Uniform(0,1) to samples
>> % from de Moivre(1/7,1/7,1/7,1/7,1/7,1/7,1/7)
>> outcomes=SimdeMoivreEqui(u,k); % save the outcomes in an array
>> disp(outcomes);
   6      5      5      5      2
```

Now, let us consider the more general problem of implementing a sampler for an arbitrary but specified de Moivre($\theta_1, \theta_2, \dots, \theta_k$) RV. That is, the values of θ_i need not be equal to $1/k$.

Simulation 78 (de Moivre($\theta_1, \theta_2, \dots, \theta_k$)) We can generate samples from a de Moivre($\theta_1, \theta_2, \dots, \theta_k$) RV X when $(\theta_1, \theta_2, \dots, \theta_k)$ are specifiable as an input vector via the following algorithm.

The M-file implementing Algorithm 6 is:

```
function x = SimdeMoivreOnce(u,thetas)
% Returns a sample from the de Moivre(thetas=(theta_1,...,theta_k)) RV X
% Call Syntax: x = SimdeMoivreOnce(u,thetas);
```

Algorithm 6 Inversion Sampler for de Moivre($\theta_1, \theta_2, \dots, \theta_k$) RV X 1: *input:*1. parameter vector $(\theta_1, \theta_2, \dots, \theta_k)$ of de Moivre($\theta_1, \theta_2, \dots, \theta_k$) RV X .2. $u \sim \text{Uniform}(0, 1)$ 2: *output:* a sample from X 3: *initialise:* $F \leftarrow \theta_1, i \leftarrow 1$ 4: **while** $u > F$ **do**5: $i \leftarrow i + 1$ 6: $F \leftarrow F + \theta_i$ 7: **end while**8: *return:* $x \leftarrow i$

```
%           deMoivreEqui(u,thetas);
% Input      : u = a uniform random number eg. rand
%             : thetas = an array of probabilities thetas=[theta_1 ... theta_k]
% Output     : x = sample from X
x=1; % initial index is 1
cum_theta=thetas(x);
while u > cum_theta;
    x=x+1;
    cum_theta = cum_theta + thetas(x);
end
```

Let us use the function `deMoivreEqui` to draw five samples from a fair seven-faced dice.

```
>> k=7; % number of faces of the fair dice
>> n=5; % number of trials
>> rand('twister',78657); % initialise the fundamental sampler
>> Us=rand(1,n); % draw n samples from Uniform(0,1)
>> disp(Us);
    0.8330    0.6819    0.6468    0.6674    0.2577
>> % inverse transform samples from Uniform(0,1) to samples
>> % from de Moivre(1/7,1/7,1/7,1/7,1/7,1/7,1/7)
>> f=[1/7 1/7 1/7 1/7 1/7 1/7 1/7];
>> disp(f);
    0.1429    0.1429    0.1429    0.1429    0.1429    0.1429
>> % use funarray to apply function-handled SimdeMoivreOnce to
>> % each element of array Us and save it in array outcomes2
>> outcomes2=arrayfun(@(u)(SimdeMoivreOnce(u,f)),Us);
>> disp(outcomes2);
    6      5      5      5      2
>> disp(SimdeMoivreEqui(u,k)); % same result using the previous algorithm
    6      5      5      5      2
```

Clearly, Algorithm 6 may be used to sample from any de Moivre($\theta_1, \dots, \theta_k$) RV X . We demonstrate this by producing five samples from a randomly generated PMF `f2`.

```
>> rand('twister',1777); % initialise the fundamental sampler
>> f2=rand(1,10); % create an arbitrary array
>> f2=f2/sum(f2); % normalize to make a probability mass function
>> disp(f2); % display the weights of our 10-faced die
    0.0073    0.0188    0.1515    0.1311    0.1760    0.1121    ...
    0.1718    0.1213    0.0377    0.0723
```

```
>> disp(sum(f2)); % the weights sum to 1
    1.0000
>> disp(arrayfun(@(u)(SimdeMoivreOnce(u,f2)),rand(5,5))) % the samples from f2 are
    4     3     4     7     3
    6     7     4     5     3
    5     8     7    10     6
    2     3     5     7     7
    6     5     9     5     7
```

Note that the principal work here is the sequential search, in which the mean number of comparisons until success is:

$$1\theta_1 + 2\theta_2 + 3\theta_3 + \dots + k\theta_k = \sum_{i=1}^k i\theta_i$$

For the de Moivre($1/k, 1/k, \dots, 1/k$) RV, the right-hand side of the above expression is:

$$\sum_{i=1}^k i \frac{1}{k} = \frac{1}{k} \sum_{i=1}^k i = \frac{1}{k} \frac{k(k+1)}{2} = \frac{k+1}{2},$$

indicating that the average-case efficiency is linear in k . This linear dependence on k is denoted by $O(k)$. In other words, as the number of faces k increases, one has to work linearly harder to get samples from de Moivre($1/k, 1/k, \dots, 1/k$) RV using Algorithm 6. Using the simpler Algorithm 5, which exploits the fact that all values of θ_i are equal, we generated samples in constant time, which is denoted by $O(1)$. Let us consider a RV that arises from an IID stochastic process of Bernoulli(θ) RVs $\{X_i\}_{i \in \mathbb{N}}$, ie.

$$\{X_i\}_{i \in \mathbb{N}} := \{X_1, X_2, \dots\} \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta).$$

When we consider the number of IID Bernoulli(θ) trials before the first ‘Head’ occurs we get the following discrete RV.

Model 12 (Geometric(θ) RV) Given a parameter $\theta \in (0, 1)$, the PMF of the Geometric(θ) RV X is

$$f(x; \theta) = \begin{cases} \theta(1 - \theta)^x & \text{if } x \in \mathbb{Z}_+ := \{0, 1, 2, \dots\} \\ 0 & \text{otherwise} \end{cases} \quad (6.17)$$

It is straightforward to verify that $f(x; \theta)$ is indeed a PDF :

$$\sum_{x=0}^{\infty} f(x; \theta) = \sum_{x=0}^{\infty} \theta(1 - \theta)^x = \theta \left(\frac{1}{1 - (1 - \theta)} \right) = \theta \left(\frac{1}{\theta} \right) = 1$$

The above equality is a consequence of the geometric series identity (6.18) with $a = \theta$ and $\vartheta := 1 - \theta$:

$$\sum_{x=0}^{\infty} a\vartheta^x = a \left(\frac{1}{1 - \vartheta} \right), \text{ provided, } 0 < \vartheta < 1. \quad (6.18)$$

Proof:

$$a + a\vartheta + a\vartheta^2 + \dots + a\vartheta^n = \sum_{0 \leq x \leq n} a\vartheta^x = a + \sum_{1 \leq x \leq n} a\vartheta^x = a + \vartheta \sum_{1 \leq x \leq n} a\vartheta^{x-1} = a + \vartheta \sum_{0 \leq x \leq n-1} a\vartheta^x = a + \vartheta \sum_{0 \leq x \leq n} a\vartheta^x - a\vartheta^{n+1}$$

Therefore,

$$\begin{aligned}
 \sum_{0 \leq x \leq n} a\vartheta^x &= a + \vartheta \sum_{0 \leq x \leq n} a\vartheta^x - a\vartheta^{n+1} \\
 \left(\sum_{0 \leq x \leq n} a\vartheta^x \right) - \left(\vartheta \sum_{0 \leq x \leq n} a\vartheta^x \right) &= a - a\vartheta^{n+1} \\
 \left(\sum_{0 \leq x \leq n} a\vartheta^x \right) (1 - \vartheta) &= a(1 - \vartheta^{n+1}) \\
 \sum_{0 \leq x \leq n} a\vartheta^x &= a \left(\frac{1 - \vartheta^{n+1}}{1 - \vartheta} \right) \\
 \sum_{x=0}^{\infty} a\vartheta^x := \lim_{n \rightarrow \infty} \sum_{0 \leq x \leq n} a\vartheta^x &= a \left(\frac{1}{1 - \vartheta} \right), \text{ provided, } 0 < \vartheta < 1
 \end{aligned}$$

The outcome of a Geometric(θ) RV can be thought of as “the number of tosses needed before the appearance of the first ‘Head’ when tossing a coin with probability of ‘Heads’ equal to θ in a independent and identical manner.”

Mean and variance of Geometric(θ) RV: Let $X \sim \text{Geometric}(\theta)$ RV. Then,

$$\mathbf{E}(X) = \sum_{x=0}^{\infty} x\theta(1-\theta)^x = \theta \sum_{x=0}^{\infty} x(1-\theta)^x$$

In order to simplify the RHS above, let us employ differentiation with respect to θ :

$$\frac{-1}{\theta^2} = \frac{d}{d\theta} \left(\frac{1}{\theta} \right) = \frac{d}{d\theta} \sum_{x=0}^{\infty} (1-\theta)^x = \sum_{x=0}^{\infty} -x(1-\theta)^{x-1}$$

Multiplying the LHS and RHS above by $-(1-\theta)$ and substituting in $\mathbf{E}(X) = \theta \sum_{x=0}^{\infty} x(1-\theta)^x$, we get a much simpler expression for $\mathbf{E}(X)$:

$$\frac{1-\theta}{\theta^2} = \sum_{x=0}^{\infty} x(1-\theta)^x \implies \mathbf{E}(X) = \theta \left(\frac{1-\theta}{\theta^2} \right) = \frac{1-\theta}{\theta}.$$

Similarly, it can be shown that

$$\mathbf{V}(X) = \frac{1-\theta}{\theta^2}.$$

Simulation 79 (Geometric(θ)) We can simulate a sample x from a Geometric(θ) RV X using the following simple algorithm:

$$x \leftarrow \lfloor \log(u)/\log(1-\theta) \rfloor, \quad \text{where, } u \sim \text{Uniform}(0, 1).$$

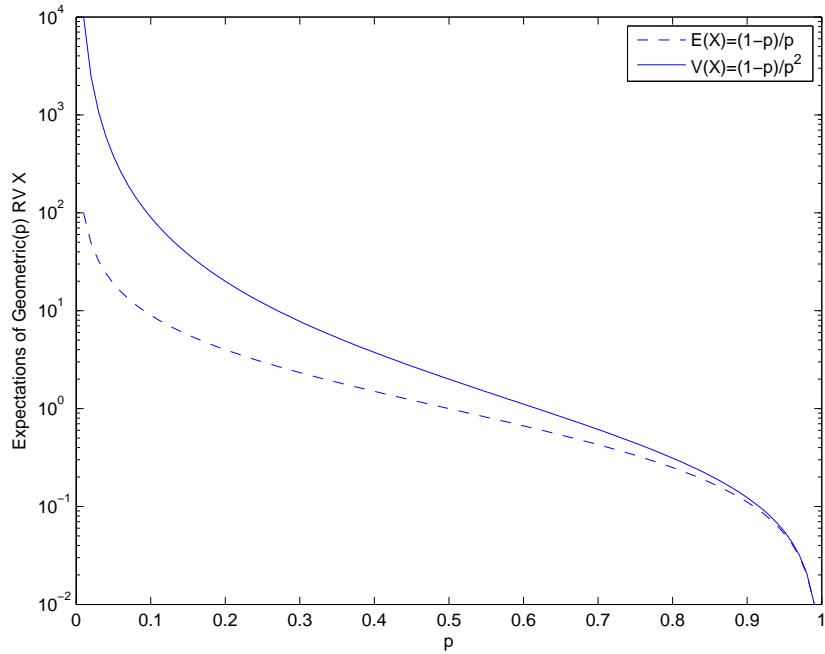
To verify that the above procedure is valid, note that:

$$\begin{aligned}
 \lfloor \log(U)/\log(1-\theta) \rfloor = x &\iff x \leq \log(U)/\log(1-\theta) < x+1 \\
 &\iff x \leq \log_{1-\theta}(U) < x+1 \\
 &\iff (1-\theta)^x \geq U > (1-\theta)^{x+1}
 \end{aligned}$$

The inequalities are reversed since the base being exponentiated is $1-\theta \leq 1$. The uniform event $(1-\theta)^x \geq U > (1-\theta)^{x+1}$ happens with the desired probability:

$$(1-\theta)^x - (1-\theta)^{x+1} = (1-\theta)^x(1 - (1-\theta)) = \theta(1-\theta)^x =: f(x; \theta), \quad X \sim \text{Geometric}(\theta).$$

We implement the sampler to generate samples from Geometric(θ) RV with $\theta = 0.5$, for instance:

Figure 6.12: Mean and variance of a Geometric(θ) RV X as a function of the parameter θ .

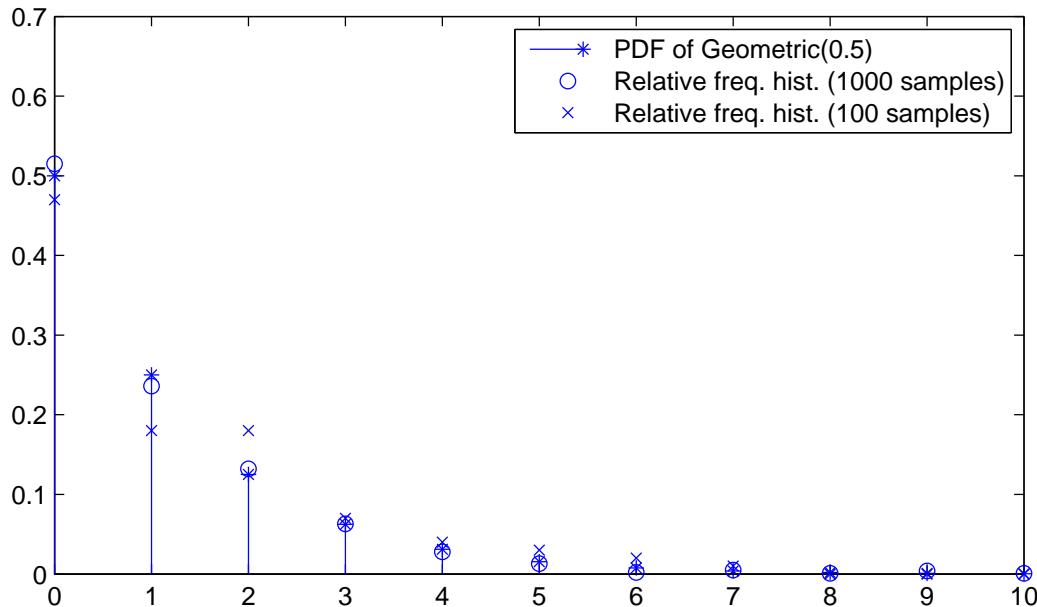
```
>> theta=0.5; u=rand(); % choose some theta and uniform(0,1) variate
>> % Simulate from a Geomertic(theta) RV
>> floor(log (u) / log (1 - theta))
ans =
    0
>> floor(log ( rand(1,10) ) / log (1 - 0.5)) % theta=0.5, 10 samples
ans =
    0    0    1    0    2    1    0    0    0    0
```

Labwork 80 (Compare PDF to the relative frequency histogram of simulated Geometric(θ) RV)

It is a good idea to make a relative frequency histogram of a simulation algorithm and compare that to the PDF of the discrete RV we are simulating from. We use the following script to create Figure 6.13:

```
theta=0.5;
SampleSize=1000;
% simulate 1000 samples from Geometric(theta) RV
Samples=floor(log(rand(1,SampleSize))/ log (1-theta));
Xs = 0:10; % get some values for x
RelFreqs=hist(Samples,Xs)/SampleSize; % relative frequencies of Samples
stem(Xs,theta*((1-theta) .^ Xs),'*')% PDF of Geometric(theta) over Xs
hold on;
plot(Xs,RelFreqs,'o')% relative frequence histogram
RelFreqs100=hist(Samples(1:100),Xs)/100; % Relative Frequencies of first 100 samples
plot(Xs,RelFreqs100,'x')
legend('PDF of Geometric(0.5)', 'Relative freq. hist. (1000 samples)', ...
'Relative freq. hist. (100 samples)')
```

Figure 6.13: PDF of $X \sim \text{Geometric}(\theta = 0.5)$ and the relative frequency histogram based on 100 and 1000 samples from X .



The RV Y in Table ?? may be generalized to an experiment \mathcal{E}_θ^n with n coin tosses. Let X_i be the Indicator function of the event ‘Heads on the i -th toss’ as before. Then Y defined by,

$$Y := \sum_{i=1}^n X_i := X_1 + X_2 + \cdots + X_n ,$$

is the number of ‘Heads’ in n tosses. Akin to the second row of Table ??, for the ‘Toss n times’ experiment \mathcal{E}_θ^n the RV Y as defined above will take values in $\{0, 1, 2, \dots, n\}$ and is therefore a discrete RV. This is called the Binomial RV as defined next. But, first we remind ourselves of some elementary definitions involving arrangements of objects from a collection (recall Section 1.5).

Model 13 (Binomial(n, θ) RV) Let the RV $X = \sum_{i=1}^n X_i$ be the sum of n independent and identically distributed Bernoulli(θ) RVs, i.e.:

$$X = \sum_{i=1}^n X_i, \quad X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta) .$$

Given two parameters n and θ , the PMF of the Binomial(n, θ) RV X is:

$$f(x; n, \theta) = \begin{cases} \binom{n}{x} \theta^x (1 - \theta)^{n-x} & \text{if } x \in \{0, 1, 2, 3, \dots, n\} , \\ 0 & \text{otherwise} \end{cases} \quad (6.19)$$

where, $\binom{n}{x}$ is:

$$\binom{n}{x} = \frac{n(n-1)(n-2)\dots(n-x+1)}{x(x-1)(x-2)\dots(2)(1)} = \frac{n!}{x!(n-x)!} .$$

$\binom{n}{x}$ is read as “ n choose x .”

Proof: Observe that for the Binomial(n, θ) RV X , $\mathbf{P}(X = x) = f(x; n, \theta)$ is the probability that x of the n Bernoulli(θ) trials result in an outcome of 1's. Next note that if all $n X_i$'s are 0's, then $X = 0$, and if all $n X_i$'s are 1's, then $X = n$. In general, if some of the $n X_i$'s are 1's and the others are 0, then X can only take values in $\{0, 1, 2, \dots, n\}$ and therefore $f(x; n, \theta) = 0$ if $x \notin \{0, 1, 2, \dots, n\}$.

Now, let us compute $f(x; n, \theta)$ when $x \in \{0, 1, 2, \dots, n\}$. Consider the set of indices $\{1, 2, 3, \dots, n\}$ for the n IID Bernoulli(θ) RVs $\{X_1, X_2, \dots, X_n\}$. Now choose x indices from $\{1, 2, \dots, n\}$ to mark those trials in a particular realization of $\{x_1, x_2, \dots, x_n\}$ with the Bernoulli outcome of 1. The probability of each such event is $\theta^x(1 - \theta)^{n-x}$ due to the IID assumption. For each realization $\{x_1, x_2, \dots, x_n\} \in \{0, 1\}^n := \{\text{all binary } (0-1) \text{ strings of length } n\}$, specified by a choice of x trial indices with Bernoulli outcome 1, the binomial RV $X = \sum_{i=1}^n X_i$ takes the value x . Since there are exactly $\binom{n}{x}$ many ways in which we can choose x trial indices (with outcome 1) from the set of n trial indices $\{1, 2, \dots, n\}$, we get the desired product for $f(x; n, \theta) = \binom{n}{x} \theta^x (1 - \theta)^{n-x}$ when $x \in \{0, 1, \dots, n\}$.

Mean and variance of Binomial(n, θ) RV: Let $X \sim \text{Binomial}(n, \theta)$. Based on the definition of expectation:

$$\mathbf{E}(X) = \int x dF(x; n, \theta) = \sum_x x f(x; n, \theta) = \sum_{x=0}^n x \binom{n}{x} \theta^x (1 - \theta)^{n-x} .$$

However, this is a nontrivial sum to evaluate. Instead, we may use (3.10) and (3.13) by noting that $X = \sum_{i=1}^n X_i$, where the $\{X_1, X_2, \dots, X_n\} \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta)$, $\mathbf{E}(X_i) = \theta$ and $\mathbf{V}(X_i) = \theta(1 - \theta)$:

$$\begin{aligned} \mathbf{E}(X) &= \mathbf{E}(X_1 + X_2 + \dots + X_n) = \mathbf{E}\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n \mathbf{E}(X_i) = n\theta , \\ \mathbf{V}(X) &= \mathbf{V}\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n \mathbf{V}(X_i) = \sum_{i=1}^n \theta(1 - \theta) = n\theta(1 - \theta) . \end{aligned}$$

Labwork 81 (Binomial coefficient) We may implement the MATLAB function `BinomialCoefficient` to compute:

$$\binom{n}{x} = \frac{n!}{x!(n-x)!} = \frac{n(n-1)(n-2)\dots(n-x+1)}{x(x-1)(x-2)\dots(2)(1)} = \frac{\prod_{i=(n-x+1)}^n i}{\prod_{i=2}^x} ,$$

with the following M-file:

```
function BC = BinomialCoefficient(n,x)
% returns the binomial coefficient of n choose x
% i.e. the combination of n objects taken x at a time
% x and n are scalar integers and 0 <= x <= n
NminusX = n-x;
NumeratorPostCancel = prod(n:-1:(max([NminusX,x])+1)) ;
DenominatorPostCancel = prod(2:min([NminusX, x]));
BC = NumeratorPostCancel/DenominatorPostCancel;
```

and call `BinomialCoefficient` in the function `BinomialPdf` to compute the PDF $f(x; n, \theta)$ of the Binomial(n, θ) RV X as follows:

```
BinomialPdf.m
```

```
function fx = BinomialPdf(x,n,theta)
% Binomial probability mass function. Needs BinomialCoefficient(n,x)
% f = BinomialPdf(x,n,theta)
% f is the prob mass function for the Binomial(x;n,theta) RV
% and x can be array of samples.
% Values of x are integers in [0,n] and theta is a number in [0,1]
fx = zeros(size(x));
fx = arrayfun(@(xi)(BinomialCoefficient(n,xi)),x);
fx = fx .* (theta .^ x) .* (1-theta) .^ (n-x);
```

For example, we can compute the desired PDF for an array of samples x from $\text{Binomial}(8, 0.5)$ RV X , as follows:

```
>> x=0:1:8
x =      0      1      2      3      4      5      6      7      8
>> BinomialPdf(x,8,0.5)
ans =    0.0039    0.0312    0.1094    0.2188    0.2734    0.2188    0.1094    0.0312    0.0039
```

Simulation 82 ($\text{Binomial}(n, \theta)$ as $\sum_{i=1}^n \text{Bernoulli}(\theta)$) Since the $\text{Binomial}(n, \theta)$ RV X is the sum of n IID $\text{Bernoulli}(\theta)$ RVs we can also simulate from X by first simulating n IID $\text{Bernoulli}(\theta)$ RVs and then adding them up as follows:

```
>> rand('twister',17678);
>> theta=0.5; % give some desired theta value, say 0.5
>> n=5; % give the parameter n for Binomial(n,theta) RV X, say n=5
>> xis=floor(rand(1,n)+theta) % produce n IID samples from Bernoulli(theta=0.5) RVs X1,X2,...Xn
xis =      1      1      0      0      0
>> x=sum(xis) % sum up the xis to get a sample from Binomial(n=5,theta=0.5) RV X
x =      2
```

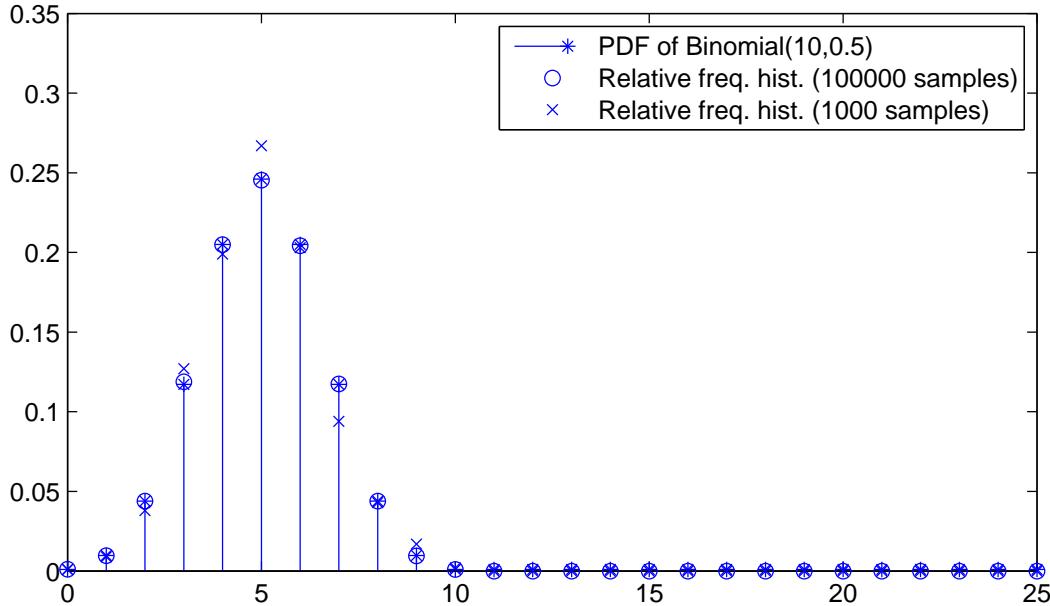
It is straightforward to produce more than one sample from X by exploiting the column-wise summing property of MATLAB's `sum` function when applied to a two-dimensional array:

```
>> rand('twister',17);
>> theta=0.25; % give some desired theta value, say 0.25 this time
>> n=3; % give the parameter n for Binomial(n,theta) RV X, say n=3 this time
>> xis10 = floor(rand(n,10)+theta) % produce an n by 10 array of IID samples from Bernoulli(theta=0.25) RVs
xis10 =
  0     0     0     0     1     0     0     0     0     0
  0     1     0     1     1     0     0     0     0     0
  0     0     0     0     0     0     0     1     0     0
>> x=sum(xis10) % sum up the array column-wise to get 10 samples from Binomial(n=3,theta=0.25) RV X
x =      0      1      0      1      2      0      0      1      0      0
```

In Simulation 82, the number of IID $\text{Bernoulli}(\theta)$ RVs needed to simulate one sample from the $\text{Binomial}(n, \theta)$ RV is exactly n . Thus, as n increases, the amount of time needed to simulate from $\text{Binomial}(n, \theta)$ is $O(n)$, i.e. linear in n . We can simulate more efficiently by exploiting a simple relationship between the $\text{Geometric}(\theta)$ RV and the $\text{Binomial}(n, \theta)$ RV.

The $\text{Binomial}(n, \theta)$ RV X is related to the IID $\text{Geometric}(\theta)$ RV Y_1, Y_2, \dots : X is the number of successful $\text{Bernoulli}(\theta)$ outcomes (outcome is 1) that occur in a total of n $\text{Bernoulli}(\theta)$ trials, with the number of trials between consecutive successes distributed according to IID $\text{Geometric}(\theta)$ RV.

Figure 6.14: PDF of $X \sim \text{Binomial}(n = 10, \theta = 0.5)$ and the relative frequency histogram based on 100,000 samples from X .



Simulation 83 (Binomial(θ) from IID Geometric(θ) RVs) By this principle, we can simulate from the Binomial(θ) X by Step 1: generating IID Geometric(θ) RVs Y_1, Y_2, \dots , Step 2: stopping as soon as $\sum_{i=1}^k (Y_i + 1) > n$ and Step 3: setting $x \leftarrow k - 1$.

We implement the above algorithm via the following M-file:

```
function x = Sim1BinomByGeoms(n,theta)
% Simulate one sample from Binomial(n,theta) via Geometric(theta) RVs
YSum=0; k=0; % initialise
while (YSum <= n),
    TrialsToSuccess=floor(log(rand)/log (1-theta)) + 1; % sample from Geometric(theta)+1
    YSum = YSum + TrialsToSuccess; % total number of trials
    k=k+1; % number of Bernoulli successes
end
x=k-1; % return x
```

Here is a call to simulate 12 samples from Binomial($n = 10, \theta = 0.5$) RV:

```
>> theta=0.5; % declare theta
>> n=10; % say n=10
>> SampleSize=12;% say you want to simulate 12 samples
>> rand('twister',10001) % seed the fundamental sampler
>> Samples=arrayfun(@(T)Sim1BinomByGeoms(n,T),theta*ones(1,SampleSize))
Samples = 7 5 8 8 4 1 4 8 2 4 6 5
```

Figure 6.14 depicts a comparison of the PDF of $\text{Binomial}(n = 10, \theta = 0.5)$ RV and a relative frequency histogram based on 100,000 simulations from it.

In several situations it becomes cumbersome to model the events using the $\text{Binomial}(n, \theta)$ RV, especially when when the parameter $\theta \propto 1/n$ and the events become rare. However, for some real

parameter $\lambda > 0$, the $\text{Binomial}(n, \lambda/n)$ RV with probability of the number of successes in n trials, with per-trial success probability λ/n , approaches the Poisson distribution with expectation λ , as n approaches ∞ (actually, it converges in distribution as defined later). The $\text{Poisson}(\lambda)$ RV is much simpler to work with than the combinatorially laden $\text{Binomial}(n, \theta = \lambda/n)$ RV. We sketch the details of this next.

Let $X \sim \text{Binomial}(n, \theta = \lambda/n)$, then for any $x \in \{0, 1, 2, 3, \dots, n\}$,

$$\begin{aligned}\mathbf{P}(X = x) &= \binom{n}{x} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^{n-x} \\ &= \frac{n(n-1)(n-2)\cdots(n-x+1)}{x(x-1)(x-2)\cdots(2)(1)} \left(\frac{\lambda^x}{n^x}\right) \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-x} \\ &= \overbrace{\left(\frac{n}{n}\right) \left(\frac{n-1}{n}\right) \left(\frac{n-2}{n}\right) \cdots \left(\frac{n-x+1}{n}\right)}^{\text{underbrace}} \overbrace{\left(\frac{\lambda^x}{x!}\right)}^{\text{underbrace}} \underbrace{\left(1 - \frac{\lambda}{n}\right)^n}_{\text{overbrace}} \underbrace{\left(1 - \frac{\lambda}{n}\right)^{-x}}_{\text{overbrace}}\end{aligned}\tag{6.20}$$

As $n \rightarrow \infty$, the expression below the first overbrace $\rightarrow 1$, while that below the second overbrace, being independent of n remains the same. By the elementary examples of limits 14 and 15, as $n \rightarrow \infty$, the expression over the first underbrace approaches $e^{-\lambda}$ while that over the second underbrace approaches 1. Finally, we get the desired limit:

$$\lim_{n \rightarrow \infty} \mathbf{P}(X = x) = \frac{e^{-\lambda} \lambda^x}{x!} .$$

Model 14 (Poisson(λ) RV) Given a real parameter $\lambda > 0$, the discrete RV X is said to be Poisson(λ) distributed if X has PDF:

$$f(x; \lambda) = \begin{cases} \frac{e^{-\lambda} \lambda^x}{x!} & \text{if } x \in \mathbb{Z}_+ := \{0, 1, 2, \dots\} , \\ 0 & \text{otherwise .} \end{cases}\tag{6.21}$$

Note that the PDF integrates to 1:

$$\sum_{x=0}^{\infty} f(x; \lambda) = \sum_{x=0}^{\infty} \frac{e^{-\lambda} \lambda^x}{x!} = e^{-\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} = e^{-\lambda} e^{\lambda} = 1 ,$$

where we exploit the Taylor series of e^{λ} to obtain the second-last equality above.

Mean and variance of Poisson(λ) RV: Let $X \sim \text{Poisson}(\lambda)$. Then:

$$\mathbf{E}(X) = \sum_{x=0}^{\infty} x f(x; \lambda) = \sum_{x=0}^{\infty} x \frac{e^{-\lambda} \lambda^x}{x!} = e^{-\lambda} \sum_{x=0}^{\infty} x \frac{\lambda^x}{x!} = e^{-\lambda} \sum_{x-1=0}^{\infty} \frac{\lambda \lambda^{x-1}}{(x-1)!} = e^{-\lambda} \lambda e^{\lambda} = \lambda .$$

Similarly,

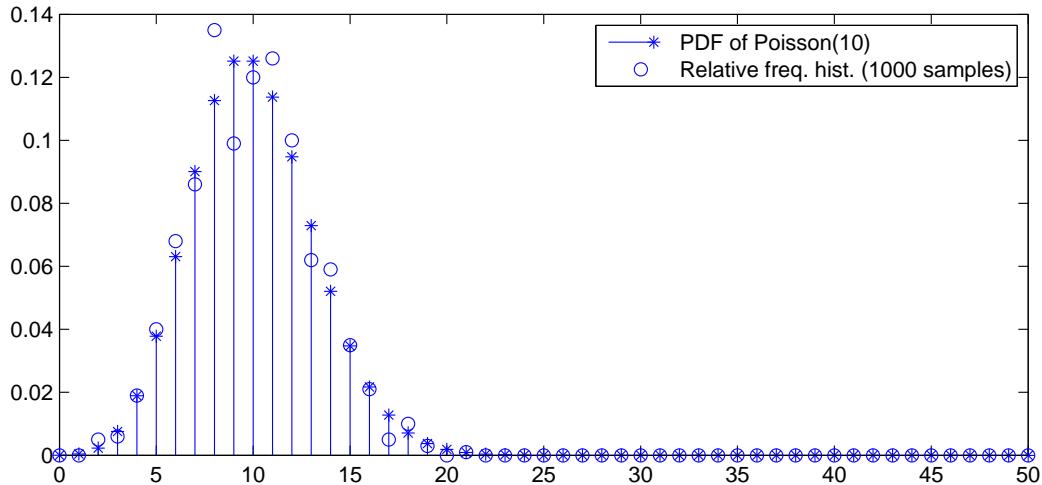
$$\mathbf{V}(X) = \mathbf{E}(X^2) - (\mathbf{E}(X))^2 = \lambda + \lambda^2 - \lambda^2 = \lambda .$$

since

$$\begin{aligned}
 \mathbf{E}(X^2) &= \sum_{x=0}^{\infty} x^2 \frac{e^{-\lambda} \lambda^x}{x!} = \lambda e^{-\lambda} \sum_{x=1}^{\infty} \frac{x \lambda^{x-1}}{(x-1)!} = \lambda e^{-\lambda} \left(1 + \frac{2\lambda}{1} + \frac{3\lambda^2}{2!} + \frac{4\lambda^3}{3!} + \dots \right) \\
 &= \lambda e^{-\lambda} \left(\left(1 + \frac{\lambda}{1} + \frac{\lambda^2}{2!} + \frac{\lambda^3}{3!} + \dots \right) + \left[\frac{\lambda}{1} + \frac{2\lambda^2}{2!} + \frac{3\lambda^3}{3!} + \dots \right] \right) \\
 &= \lambda e^{-\lambda} \left((e^\lambda) + \lambda \left(1 + \frac{2\lambda}{2!} + \frac{3\lambda^2}{3!} + \dots \right) \right) = \lambda e^{-\lambda} \left(e^\lambda + \lambda \left(1 + \lambda + \frac{\lambda^2}{2!} + \dots \right) \right) \\
 &= \lambda e^{-\lambda} (e^\lambda + \lambda e^\lambda) = \lambda e^{-\lambda} (e^\lambda + \lambda e^\lambda) = \lambda(1 + \lambda) = \lambda + \lambda^2
 \end{aligned}$$

Note that $\text{Poisson}(\lambda)$ distribution is one whose mean and variance are the same, namely λ .

Figure 6.15: PDF of $X \sim \text{Poisson}(\lambda = 10)$ and the relative frequency histogram based on 1000 samples from X .



The $\text{Poisson}(\lambda)$ RV X is also related to the IID $\text{Exponential}(\lambda)$ RVs Y_1, Y_2, \dots : X is the number of occurrences, per unit time, of an instantaneous event whose inter-occurrence time is the IID $\text{Exponential}(\lambda)$ RV. For example, the number of buses arriving at our bus-stop in the next minute, with exponentially distributed inter-arrival times, has a Poisson distribution.

Simulation 84 ($\text{Poisson}(\lambda)$ from IID $\text{Exponential}(\lambda)$ RVs) By this principle, we can simulate from the $\text{Poisson}(\lambda)$ X by Step 1: generating IID $\text{Exponential}(\lambda)$ RVs Y_1, Y_2, \dots , Step 2: stopping as soon as $\sum_{i=1}^k Y_i \geq 1$ and Step 3: setting $x \leftarrow k - 1$.

We implement the above algorithm via the following M-file:

```

function x = Sim1Poisson(lambda)
% Simulate one sample from Poisson(lambda) via Exponentials
YSum=0; k=0; % initialise
while (YSum < 1),
    YSum = YSum + -(1/lambda) * log(rand);
    k=k+1;
end
x=k-1; % return x

```

Here is a call to simulate 10 samples from $\text{Poisson}(\lambda = 10.0)$ and $\text{Poisson}(\lambda = 0.1)$ RVs:

```
>> arrayfun(@(lambda)Sim1Poisson(lambda),10.0*ones(1,10)) % lambda=10.0
ans =
    14     7    10    13    11     3     6     5     8     5
>> arrayfun(@(lambda)Sim1Poisson(lambda),0.1*ones(1,10)) % lambda=0.1
ans =
     2     0     0     0     0     0     0     0     0     0
```

Figure 6.15 depicts a comparison of the PDF of $\text{Poisson}(\lambda = 10)$ RV and a relative frequency histogram based on 1000 simulations from it.

Simulating from a $\text{Poisson}(\lambda)$ RV is also a special case of simulating from the following more general RV.

Model 15 ($GD(\theta_0, \theta_1, \dots)$) We say X is a General Discrete($\theta_0, \theta_1, \dots$) or $GD(\theta_0, \theta_1, \dots)$ RV over the countable discrete state space $\mathbb{Z}_+ := \{0, 1, 2, \dots\}$ with parameters $(\theta_0, \theta_1, \dots)$ if the PMF of X is defined as follows:

$$f(X = x; \theta_0, \theta_1, \dots) = \begin{cases} 0, & \text{if } x \notin \{0, 1, 2, \dots\} \\ \theta_0, & \text{if } x = 0 \\ \theta_1, & \text{if } x = 1 \\ \vdots \end{cases}$$

Algorithm 7 allows us to simulate from any member of the class of non-negative discrete RVs as specified by the probabilities $(\theta_0, \theta_1, \dots)$. When an RV X takes values in another countable set $\mathbb{X} \neq \mathbb{Z}_+$, then we can still use the above algorithm provided we have a one-to-one and onto mapping D from \mathbb{Z}_+ to \mathbb{X} that allows us to think of $\{0, 1, 2, \dots\}$ as indices of an array D .

Algorithm 7 Inversion Sampler for $GD(\theta_0, \theta_1, \dots)$ RV X

1: *input:*

1. θ_0 and $\{C(i) = \theta_i / \theta_{i-1}\}$ for any $i \in \{1, 2, 3, \dots\}$.
2. $u \sim \text{Uniform}(0, 1)$

2: *output:* a sample from X

3: *initialise:* $p \leftarrow \theta_0$, $q \leftarrow \theta_0$, $i \leftarrow 0$

4: **while** $u > q$ **do**

5: $i \leftarrow i + 1$, $p \leftarrow p C(i)$, $q \leftarrow q + p$

6: **end while**

7: *return:* $x = i$

Simulation 85 (Binomial(n, θ)) To simulate from a $\text{Binomial}(n, \theta)$ RV X , we can use Algorithm 7 with:

$$\theta_0 = (1 - \theta)^n, \quad C(x + 1) = \frac{\theta(n - x)}{(1 - \theta)(x + 1)}, \quad \text{Mean Efficiency: } O(1 + n\theta).$$

Similarly, with the appropriate θ_0 and $C(x + 1)$, we can also simulate from the $\text{Geometric}(\theta)$ and $\text{Poisson}(\lambda)$ RVs.

Labwork 86 This is a challenging exercise for the student who is finding the other Labworks too easy. So those who are novice to MATLAB may skip this Labwork.

1. Implement Algorithm 7 via a function named `MyGenDiscInvSampler` in MATLAB. Hand in the M-file named `MyGenDiscInvSampler.m` giving detailed comments explaining your understanding of each step of the code. [Hint: $C(i)$ should be implemented as a function (use function handles via @) that can be passed as a parameter to the function `MyGenDiscInvSampler`].
2. Show that your code works for drawing samples from a $\text{Binomial}(n, p)$ RV by doing the following:
 - (a) Seed the fundamental sampler by your Student ID (if your ID is 11424620 then type `rand('twister', 11424620);`)
 - (b) Draw 100 samples from the $\text{Binomial}(n = 20, p = 0.5)$ RV and report the results in an 2×2 table with column headings x and No. of observations. [Hint: the inputs θ_0 and $C(i)$ for the $\text{Binomial}(n, p)$ RV is given above].
3. Show that your code works for drawing samples from a $\text{Geometric}(p)$ RV by doing the following:
 - (a) Seed the fundamental sampler by your Student ID.
 - (b) Set the variable `Mytheta=rand`.
 - (c) Draw 100 samples from the $\text{Geometric}(\text{Mytheta})$ RV and report the sample mean. [Note: the inputs θ_0 and $C(i)$ for the $\text{Geometric}(\theta)$ RV should be derived and the workings shown].

To make concrete sense of the $\text{Binomial}(n, \theta)$ and other more sophisticated concepts in the sequel, let us take a historical detour into some origins of statistical thinking in 19th century England.

6.7 Sir Francis Galton's Quincunx

This section is introduced to provide some forms for a kinesthetic (hands-on) and visual understanding of some elementary statistical distributions and laws. The following words are from Sir Francis Galton, F.R.S., *Natural Inheritance*, pp. 62-65, Macmillan, 1889. In here you will already find the kernels behind the construction of $\text{Binomial}(\theta)$ RV as sum of IID $\text{Bernoulli}(\theta)$ RVs, Weak Law of Large Numbers, Central Limit Theorem, and more. We will mathematically present these concepts in the sequel as a way of giving precise meanings to Galton's observations with his Quincunx. *'The Charms of Statistics.—It is difficult to understand why statisticians commonly limit their inquiries to Averages, and do not revel in more comprehensive views. Their souls seem as dull to the charm of variety as that of the native of one of our flat English counties, whose retrospect of Switzerland was that, if its mountains could be thrown into its lakes, two nuances would be got rid of at once. An Average is but a solitary fact, whereas if a single other fact be added to it, an entire Normal Scheme, which nearly corresponds to the observed one, starts potentially into existence.'*

Some people hate the very name of statistics, but I find them full of beauty and interest. Whenever they are not brutalised, but delicately handled by the higher methods, and are warily interpreted, their power of dealing with complicated phenomenon is extraordinary. They are the only tools by

Figure 6.16: Figures from Sir Francis Galton, F.R.S., *Natural Inheritance*, , Macmillan, 1889.

FIG. 7.

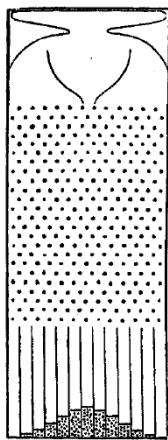


FIG. 8.

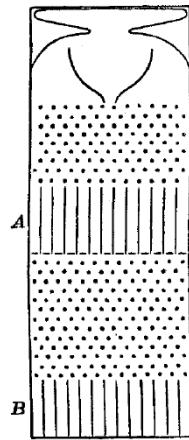


FIG. 9.

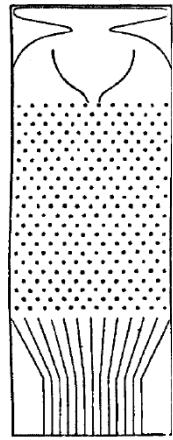


FIG. 2.

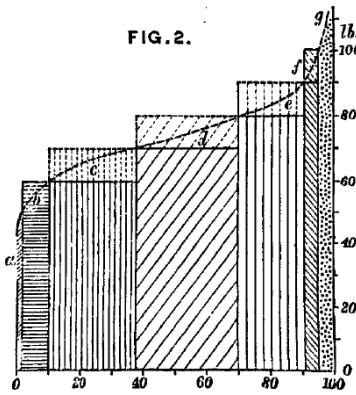
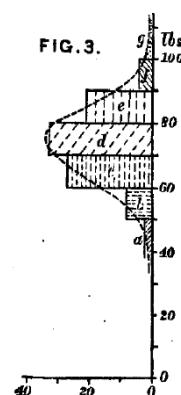


FIG. 3.



(a) FIG. 7, FIG. 8, and FIG. 9 (p. 63)

(b) FIG. 2 and FIG. 3 (p. 38)

which an opening can be cut through the formidable thicket of difficulties that bars the path of those who pursue the Science of man.

Mechanical Illustration of the Cause of the Curve of Frequency.—*The Curve of Frequency, and that of Distribution, are convertible : therefore if the genesis of either of them can be made clear, that of the other also becomes intelligible. I shall now illustrate the origin of the Curve of Frequency, by means of an apparatus shown in Fig. 7, that mimics in a very pretty way the conditions on which Deviation depends. It is a frame glazed in front, leaving a depth of about a quarter of an inch behind the glass. Strips are placed in the upper part to act as a funnel. Below the outlet of the funnel stand a succession of rows of pins stuck squarely into the backboard, and below these again are a series of vertical compartments. A charge of small shot is inclosed. When the frame is held topsy-turvy, all the shot runs to the upper end; then, when it is turned back into its working position, the desired action commences. Lateral strips, shown in the diagram, have the effect of directing all the shot that had collected at the upper end of the frame to run into the wide mouth of the funnel. The shot passes through the funnel and issuing from its narrow end, scampers deviously down through the pins in a curious and interesting way; each of them darting a step to the right or left, as the case may be, every time it strikes a pin. The pins are disposed in a quincunx fashion, so that every descending shot strikes against a pin in each successive row. The cascade issuing from the funnel broadens as it descends, and, at length, every shot finds itself caught in a compartment immediately after freeing itself from the last row of pins. The outline of the columns of shot that accumulate in the successive compartments approximates to the Curve of Frequency (Fig. 3, p. 38), and is closely of the same shape however often the experiment is repeated. The outline of the columns would become more nearly identical with the Normal Curve of Frequency, if the rows of pins were much more numerous, the shot smaller, and the compartments narrower; also if a larger quantity of shot was used.*

The principle on which the action of the apparatus depends is, that a number of small and independent accidents befall each shot in its career. In rare cases, a long run of luck continues to favour the course of a particular shot towards either outside place, but in the large majority of instances the number of accidents that cause Deviation to the right, balance in a greater or less degree those that cause Deviation to the left. Therefore most of the shot finds its way into the compartments that are situated near to a perpendicular line drawn from the outlet of the funnel, and the Frequency with which shots stray to different distances to the right or left of that line diminishes in a much

faster ratio than those distances increase. This illustrates and explains the reason why mediocrity is so common.”

Summary of Random Variables

Model	PDF	Mean	Variance
Bernoulli(θ)	$\theta^x(1-\theta)^{1-x}\mathbb{1}_{\{0,1\}}(x)$	θ	$\theta(1-\theta)$
Binomial(n, θ)	$\binom{n}{\theta} \theta^x(1-\theta)^{n-x}\mathbb{1}_{\{0,1,\dots,n\}}(x)$	$n\theta$	$n\theta(1-\theta)$
Geometric(θ)	$\theta(1-\theta)^x\mathbb{1}_{\mathbb{Z}_+}(x)$	$\frac{1}{\theta} - 1$	$\frac{1-\theta}{\theta^2}$
Poisson(λ)	$\frac{\lambda^x e^{-\lambda}}{x!} \mathbb{1}_{\mathbb{Z}_+}(x)$	λ	λ
Uniform(θ_1, θ_2)	$\mathbb{1}_{[\theta_1, \theta_2]}(x)/(\theta_2 - \theta_1)$	$\frac{\theta_1 + \theta_2}{2}$	$\frac{(\theta_2 - \theta_1)^2}{12}$
Exponential(λ)	$\lambda e^{-\lambda x}$	λ^{-1}	λ^{-2}
Normal(μ, σ^2)	$\frac{1}{\sigma\sqrt{2\pi}} e^{(x-\mu)^2/(2\sigma^2)}$	μ	σ^2
Gamma(α, β)	$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$	α/β	α/β^2

Table 6.2: Random Variables with PDF, Mean and Variance

Exercises

Ex. 6.1 — One number in the following table for the probability function of a random variable X is incorrect. Which is it, and what should the correct value be?

x	1	2	3	4	5
$\mathbf{P}(X = x)$	0.07	0.10	1.10	0.32	0.40

Ex. 6.2 — Let X be the number of years before a particular type of machine will need replacement. Assume that X has the probability function $f(1) = 0.1$, $f(2) = 0.2$, $f(3) = 0.2$, $f(4) = 0.2$, $f(5) = 0.3$.

1. Find the distribution function, F , for X , and graph both f and F .
2. Find the probability that the machine needs to be replaced during the first 3 years.
3. Find the probability that the machine needs no replacement during the first 3 years.

Ex. 6.3 — Of 200 adults, 176 own one TV set, 22 own two TV sets, and 2 own three TV sets. A person is chosen at random. What is the probability mass function of X , the number of TV sets owned by that person?

Ex. 6.4 — Suppose a discrete random variable X has probability function give by

x	3	4	5	6	7	8	9	10	11	12	13
$\mathbf{P}(X = x)$	0.07	0.01	0.09	0.01	0.16	0.25	0.20	0.03	0.02	0.11	0.05

- (a) Construct a row of cumulative probabilities for this table, that is, find the distribution function of X .
- (b) Find the following probabilities.

(i) $\mathbf{P}(X \leq 5)$	(iii) $\mathbf{P}(X > 9)$	(v) $\mathbf{P}(4 < X \leq 9)$
(ii) $\mathbf{P}(X < 12)$	(iv) $\mathbf{P}(X \geq 9)$	(vi) $\mathbf{P}(4 < X < 11)$

Ex. 6.5 — A box contains 4 right-handed and 6 left-handed screws. Two screws are drawn at random without replacement. Let X be the number of left-handed screws drawn. Find the probability mass function for X , and then calculate the following probabilities:

1. $\mathbf{P}(X \leq 1)$
2. $\mathbf{P}(X \geq 1)$
3. $\mathbf{P}(X > 1)$

Ex. 6.6 — Suppose that a random variable X has geometric probability mass function,

$$f(x) = \frac{k}{2^x} \quad (x = 0, 1, 2, \dots).$$

1. Find the value of k .
2. What is $\mathbf{P}(X \geq 4)$?

Ex. 6.7 — Four fair coins are tossed simultaneously. If we count the number of heads that appear then we have a binomial random variable, $X = \text{the number of heads}$.

1. Find the probability mass function of X .

2. Compute the probabilities of obtaining no heads, precisely 1 head, at least 1 head, not more than 3 heads.

Ex. 6.8 — The distribution of blood types in a certain population is as follows:

Blood type	Type O	Type A	Type B	Type AB
Proportion	0.45	0.40	0.10	0.05

A random sample of 15 blood donors is observed from this population. Find the probabilities of the following events.

1. Only one type AB donor is included.
2. At least three of the donors are type B.
3. More than ten of the donors are *either* type O *or* type A.
4. Fewer than five of the donors are *not* type A.

Ex. 6.9 — If the probability of hitting a target in a single shot is 10% and 10 shots are fired independently, what is the probability that the target will be hit at least once?

Ex. 6.10 — Consider the probability density function

$$f(x) = \begin{cases} k & -4 \leq x \leq 4 \\ 0 & \text{otherwise} \end{cases}.$$

1. Find the value of k .
2. Find the distribution function, F .
3. Graph f and F .

Ex. 6.11 — Assume that a new light bulb will burn out at time t hours according to the probability density function given by

$$f(t) = \begin{cases} \lambda e^{-\lambda t} & \text{if } t > 0 \\ 0 & \text{otherwise} \end{cases}.$$

In this context, λ is often called the failure rate of the bulb.

- (a) Assume that $\lambda = 0.01$, and find the probability that the bulb will not burn out before τ hours.
 This τ -specific probability is often called the reliability of the bulb.
 Hint: Use the distribution function for an Exponential(λ) random variable (recall, $F(\tau; \lambda) = \int_{-\infty}^{\tau} f(t)dt$!).

- (b) For what value of τ is the reliability of the bulb exactly $\frac{1}{2}$?

Ex. 6.12 — Feller discusses the probability and statistics of flying bomb hits in an area of southern London during II world war. The area in question was partitioned into $24 \times 24 = 576$ small squares. The total number of hits was 537. There were 229 squares with 0 hits, 211 with 1 hit, 93 with 2 hits, 35 with 3 hits, 7 with 4 hits and 1 with 5 or more hits. Assuming the hits were purely random, use the Poisson approximation to find the probability that a particular square would have exactly k hits. Compute the expected number of squares that would have 0, 1, 2, 3, 4, and 5 or more hits and compare this with the observed results (Snell 9.2.14).

Ex. 6.13 — Suppose that a certain type of magnetic tape contains, on the average, 2 defects per 100 meters. What is the probability that a roll of tape 300 meters long will contain no defects?

Ex. 6.14 — In 1910, E. Rutherford and H. Geiger showed experimentally that the number of alpha particles emitted per second in a radioactive process is a random variable X having a Poisson distribution. If the average number of particles emitted per second is 0.5, what is the probability of observing two or more particles during any given second?

Ex. 6.15 — The number of lacunae (surface pits) on specimens of steel, polished and examined in a metallurgical laboratory, is known to have Poisson distribution.

1. Write down the formula for the probability that a specimen has x defects, explaining the meanings of the symbols you use.
2. Simplify the formula in the case $x = 0$.
3. In a large homogeneous collection of specimens, 10% have one or more lacunae. Find (approximately) the percentage having exactly two.
4. Why might the Poisson distribution not apply in this situation?

Ex. 6.16 — Find the probability that none of the three bulbs in a traffic signal need to be replaced during the first 1200 hours of operation if the length of time before a single bulb needs to be replaced is a continuous random variable X with density

$$f(x) = \begin{cases} 6(0.25 - (x - 1.5)^2) & 1 < x < 2 \\ 0 & \text{otherwise} \end{cases} .$$

Note: X is measured in multiples of 1000 hours.

Ex. 6.17 — Let the random variable X be the time after which certain ball bearings wear out, with density

$$f(x) = \begin{cases} ke^{-x} & 0 \leq x \leq 2 \\ 0 & \text{otherwise} \end{cases} .$$

Note: X is measured in years.

1. Find k .
2. Find the probability that a bearing will last at least 1 year.

Ex. 6.18 — **Starting from the definition of the variance of a random variable (Definition 22) show that

$$\mathbf{V}(X) = \mathbf{E}(X^2) - (\mathbf{E}(X))^2 .$$

Ex. 6.19 — **Let X be a discrete random variable with PMF given by

$$f(x) = \begin{cases} \frac{x}{10} & \text{if } x \in \{1, 2, 3, 4\}, \\ 0 & \text{otherwise.} \end{cases}$$

(a) Find:

- (i) $\mathbf{P}(X = 0)$
- (ii) $\mathbf{P}(2.5 < X < 5)$
- (iii) $\mathbf{E}(X)$
- (iv) $\mathbf{V}(X)$

(b) Write down the DF (or CDF) of X .

(c) Plot the PMF and CDF of X .

6.8 Random Vectors

Let us try to relate some discrete probability models to the Quincunx. First, we need to introduce simple random vectors (\vec{RV}), i.e. ordered pairs, ordered triples, or more generally ordered m -tuples of random variables (X_1, X_2, \dots, X_m) . We focus on elementary definitions needed to define bivariate \vec{RV} obtained from a pair of RVs. Here is a simple example of a discrete bivariate \vec{RV} that illustrates the notions of joint and marginal probabilities.

Ex. 6.20 — Example 87 (Pair of Bernoulli(1/2) RVs) Let X_1 and X_2 be a pair of IID Bernoulli(1/2) RVs each taking values in the set $\{0, 1\}$ with the following joint probabilities:

	$X_2 = 0$	$X_2 = 1$	
$X_1 = 0$	1/4	1/4	1/2
$X_1 = 1$	1/4	1/4	1/2
	1/2	1/2	1

From the above Table we can read for instance that the joint probability $\mathbf{P}((X_1, X_2) = (0, 0)) = 1/4$ and that the marginal probability $\mathbf{P}(X_1 = 0) = 1/2$.

Definition 39 (Joint PDF, PMF, CDF) A function $f(x_1, x_2)$ is called a **joint PDF (or PMF)** for the ordered pair of random variables (X_1, X_2) if:

1. $f(x_1, x_2) \geq 0$ for all $(x_1, x_2) \in \mathbb{R}^2$

- 2.

$$1 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dF(x_1, x_2) = \begin{cases} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2) dx_1 dx_2 & \text{if } (X_1, X_2) \text{ are continuous} \\ \sum_{x_1} \sum_{x_2} f(x_1, x_2) & \text{if } (X_1, X_2) \text{ are discrete} \end{cases}$$

3. for any event $A \subset \mathbb{R}^2$,

$$\mathbf{P}(A) = \int \int_A dF(x_1, x_2) = \begin{cases} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{1}_A((x_1, x_2)) f(x_1, x_2) dx_1 dx_2 & \text{if } (X_1, X_2) \text{ are continuous} \\ \sum_{x_1} \sum_{x_2} \mathbf{1}_A((x_1, x_2)) f(x_1, x_2) & \text{if } (X_1, X_2) \text{ are discrete} \end{cases}$$

The **joint CDF or joint DF** for discrete or continuous \vec{RV} (X_1, X_2) is:

$$F(x_1, x_2) := \mathbf{P}(X_1 \leq x_1, X_2 \leq x_2).$$

Definition 40 (Marginal PDF or PMF) If the \vec{RV} (X_1, X_2) has $f(x_1, x_2)$ as its joint density, i.e. joint PDF or joint PMF, then the **marginal PDF or PMF** of X_1 is defined by:

$$f(x_1) = \mathbf{P}(X_1 = x_1) = \begin{cases} \int_{-\infty}^{\infty} f(x_1, x_2) dx_2 & \text{if } (X_1, X_2) \text{ are continuous} \\ \sum_{x_2} f(x_1, x_2) & \text{if } (X_1, X_2) \text{ are discrete} \end{cases}$$

and the **marginal PDF or PMF** of X_2 is defined by:

$$f(x_2) = \mathbf{P}(X_2 = x_2) = \begin{cases} \int_{-\infty}^{\infty} f(x_1, x_2) dx_1 & \text{if } (X_1, X_2) \text{ are continuous} \\ \sum_{x_1} f(x_1, x_2) & \text{if } (X_1, X_2) \text{ are discrete} \end{cases}$$

Example 88 (Bivariate Uniform) Let (X_1, X_2) be uniformly distributed on the square $[0, 1]^2 := [0, 1] \times [0, 1]$. Then,

$$f(x_1, x_2) = \mathbf{1}_{[0,1]^2}(x_1, x_2) .$$

Let the rectangular event $A = \{X_1 < 1/3, Y < 1/2\} \subset [0, 1]^2$. By integrating the joint PDF over A , which amounts here to finding the area of A , we compute $\mathbf{P}(A) = (1/3)(1/2) = 1/6$. Note that the marginal PDF of X_1 or X_2 is the PDF of the Uniform(0, 1) RV.

Definition 41 (Conditional PDF or PMF) Let (X_1, X_2) be a discrete bivariate RV. The conditional PMF of $X_1|X_2 = x_2$, where $f(X_2 = x_2) := \mathbf{P}(X_2 = x_2) > 0$ is:

$$f(x_1|x_2) := \mathbf{P}(X_1 = x_1|X_2 = x_2) = \frac{\mathbf{P}(X_1 = x_1, X_2 = x_2)}{\mathbf{P}(X_2 = x_2)} = \frac{f(x_1, x_2)}{f(x_2)} .$$

Similarly, if $f(X_1 = x_1) > 0$, then the conditional PMF of $X_2|X_1 = x_1$ is:

$$f(x_2|x_1) := \mathbf{P}(X_2 = x_2|X_1 = x_1) = \frac{\mathbf{P}(X_1 = x_1, X_2 = x_2)}{\mathbf{P}(X_1 = x_1)} = \frac{f(x_1, x_2)}{f(x_1)} .$$

If (X_1, X_2) are continuous RVs such that $f(x_2) > 0$, then the conditional PDF of $X_1|X_2 = x_2$ is:

$$f(x_1|x_2) = \frac{f(x_1, x_2)}{f(x_2)}, \quad \mathbf{P}(X_1 \in A|X_2 = x_2) = \int_A f(x_1|x_2) dx_1 .$$

Similarly, if $f(x_1) > 0$, then the conditional PDF of $X_2|X_1 = x_1$ is:

$$f(x_2|x_1) = \frac{f(x_1, x_2)}{f(x_1)}, \quad \mathbf{P}(X_2 \in A|X_1 = x_1) = \int_A f(x_2|x_1) dx_2 .$$

We need a new notion for the variance of two RVs.

Definition 42 (Covariance) Suppose X_1 and X_2 are random variables, such that $\mathbf{E}(X_1^2) < \infty$ and $\mathbf{E}(X_2)^2 < \infty$. Then, $\mathbf{E}(|X_1 X_2|) < \infty$ and $\mathbf{E}(|(X_1 - \mathbf{E}(X_1))(X_2 - \mathbf{E}(X_2))|) < \infty$. We therefore define the covariance $\mathbf{Cov}(X_1, X_2)$ of X_1 and X_2 as:

$$\mathbf{Cov}(X_1, X_2) := \mathbf{E}((X_1 - \mathbf{E}(X_1))(X_2 - \mathbf{E}(X_2))) = \mathbf{E}(X_1 X_2) - \mathbf{E}(X_1)\mathbf{E}(X_2)$$

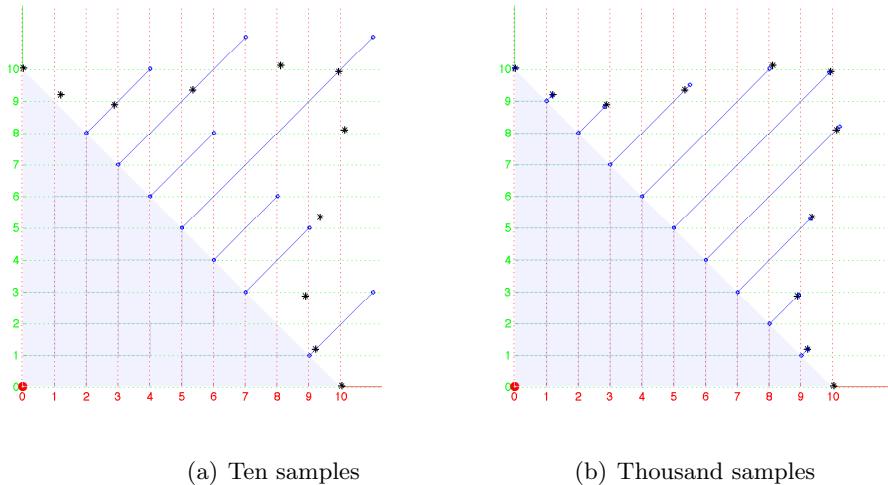
Let us consider the natural two-dimensional analogue of the Bernoulli(θ) RV in the real plane $\mathbb{R}^2 := (-\infty, \infty)^2 := (-\infty, \infty) \times (-\infty, \infty)$. A natural possibility is to use the **ortho-normal basis vectors** in \mathbb{R}^2 :

$$\boxed{e_1 := (1, 0), \quad e_2 := (0, 1)} .$$

Recall that vector addition and subtraction are done component-wise, i.e. $(x_1, x_2) \pm (y_1, y_2) = (x_1 \pm y_1, x_2 \pm y_2)$.

Classwork 89 (Geometry of Vector Addition) Recall elementary vector addition in the plane. What is $(1, 0) + (1, 0)$, $(1, 0) + (0, 1)$, $(0, 1) + (0, 1)$? What is the relationship between $(1, 0)$, $(0, 1)$ and $(1, 1)$ geometrically? How does the diagonal of the parallelogram relate to its two sides in the geometry of addition in the plane? What is $(1, 0) + (0, 1) + (1, 0)$?

Figure 6.17: Quincunx on the Cartesian plane. Simulations of $\text{Binomial}(n = 10, \theta = 0.5)$ RV as the x-coordinate of the ordered pair resulting from the culmination of sample trajectories formed by the accumulating sum of $n = 10$ IID $\text{Bernoulli}(\theta = 0.5)$ random vectors over $\{(1, 0), (0, 1)\}$ with probabilities $\{\theta, 1 - \theta\}$, respectively. The blue lines and black asterisks perpendicular to and above the diagonal line, i.e. the line connecting $(0, 10)$ and $(10, 0)$, are the density histogram of the samples and the PDF of our $\text{Binomial}(n = 10, \theta = 0.5)$ RV, respectively.



Model 16 ($\text{Bernoulli}(\theta)$ $\vec{\text{RV}}$) Given a parameter $\theta \in [0, 1]$, we say that $X := (X_1, X_2)$ is a $\text{Bernoulli}(\theta)$ random vector ($\vec{\text{RV}}$) if it has only two possible outcomes in the set $\{e_1, e_2\} \subset \mathbb{R}^2$, i.e. $x := (x_1, x_2) \in \{(1, 0), (0, 1)\}$. The PMF of the $\vec{\text{RV}}$ $X := (X_1, X_2)$ with realisation $x := (x_1, x_2)$ is:

$$f(x; \theta) := \mathbf{P}(X = x) = \theta \mathbf{1}_{\{e_1\}}(x) + (1 - \theta) \mathbf{1}_{\{e_2\}}(x) = \begin{cases} \theta & \text{if } x = e_1 := (1, 0) \\ 1 - \theta & \text{if } x = e_2 := (0, 1) \\ 0 & \text{otherwise} \end{cases}$$

Classwork 90 (Expectation and Variance of $\text{Bernoulli}(\theta)$ $\vec{\text{RV}}$) What is the Expectation of $\text{Bernoulli}(\theta)$ $\vec{\text{RV}}$?

$$\mathbf{E}_\theta(X) = \mathbf{E}_\theta((X_1, X_2)) = \sum_{(x_1, x_2) \in \{e_1, e_2\}} (x_1, x_2) f((x_1, x_2); \theta) = (1, 0)\theta + (0, 1)(1 - \theta) = (\theta, 1 - \theta).$$

How about the variance? [Hint: Use the definitions of $\mathbf{E}(X)$ and $\mathbf{V}(X)$ for the $\vec{\text{RV}}$ X . $\mathbf{E}(X^2)$ is not a single number and you may need new words such as covariance to deal with terms like $\mathbf{E}(X_1 X_2)$.]

We can write the $\text{Binomial}(n, \theta)$ RV Y as a $\text{Binomial}(n, \theta)$ $\vec{\text{RV}}$ $X := (Y, n - Y)$. In fact, this is the underlying model and the **bi** in the $\text{Binomial}(n, \theta)$ does refer to two in Latin. In the coin-tossing context this can be thought of keeping track of the number of Heads and Tails out of an IID sequence of n tosses of a coin with probability θ of observing Heads. In the Quincunx context, this amounts to keeping track of the number of right and left turns made by the ball as it drops through n levels of pegs where the probability of a right turn at each peg is independently and identically θ . In other words, the $\text{Binomial}(n, \theta)$ $\vec{\text{RV}}$ $(Y, n - Y)$ is the sum of n IID $\text{Bernoulli}(\theta)$

\vec{RV} s $X_1 := (X_{1,1}, X_{1,2}), X_2 := (X_{2,1}, X_{2,2}), \dots, X_n := (X_{n,1}, X_{n,2})$:

$$(Y, n - Y) = X_1 + X_2 + \dots + X_n = (X_{1,1}, X_{1,2}) + (X_{2,1}, X_{2,2}) + \dots + (X_{n,1}, X_{n,2})$$

Go the Biomathematics Research Centre on the 6th floor of Erskine to play with the Quincunx built by Ryan Lawrence in 2007 (See the project by Ashman and Lawrence at <http://www.math.canterbury.ac.nz/~r.sainudiin/courses/STAT218/projects/Stat218StudentProjects2007.pdf> for details). It is important to gain a physical intimacy with the Quincunx to appreciate the following model of it. We can make a statistical model of Galton's observations earlier regarding the dynamics of lead shots through the Quincunx as the sum of n IID Bernoulli(0.5) \vec{RV} s, where n is number of pegs that each ball bounces on before making a left or right turn with equal probability.

Exercise 91 (Number of paths and the binomial coefficient) How does the number of paths that lead to a bucket (x_1, x_2) with $x_1 + x_2 = n$ relate to the binomial coefficient $\binom{n}{x_1}$?

Labwork 92 (Quincunx Sampler Demo – Sum of n IID Bernoulli(1/2) \vec{RV} s) Let us understand the Quincunx construction of the Binomial($n, 1/2$) $\vec{RV} X$ as the sum of n independent and identical Bernoulli(1/2) \vec{RV} s by calling the interactive visual cognitive tool as follows:

```
>> guiMultinomial
```

The M-file `guiMultinomial.m` will bring a graphical user interface (GUI) as shown in Figure 6.18. Using the drop-down menu at “How many levels?” change the number of levels to 2 ($n = 2$). Now click the “Do one” button as many times as you like and comprehend the simulation process – the path taken by the ball as it falls through two levels. Next, from the drop-down menu at “How many Replication?” change it from 10 to 100. You can press “Do all” to watch all 100 balls drop into their possible values at level 2. Change the number of levels or n in Binomial($n, 1/2$) \vec{RV} to 3 or 5 or 10 and do more simulations until you are comfortable with the construction that the sum of n IID Bernoulli(1/2) \vec{RV} s is the Binomial($n, 1/2$) \vec{RV} .

When we drop 1000 balls into the simulated Quincunx the density histogram is much closer to the PDF of Binomial($n = 10, \theta = 0.5$) RV than when we only drop 10 balls. See Figure 6.17 for a description of the simulations. Try to replicate such a simulation on your own.

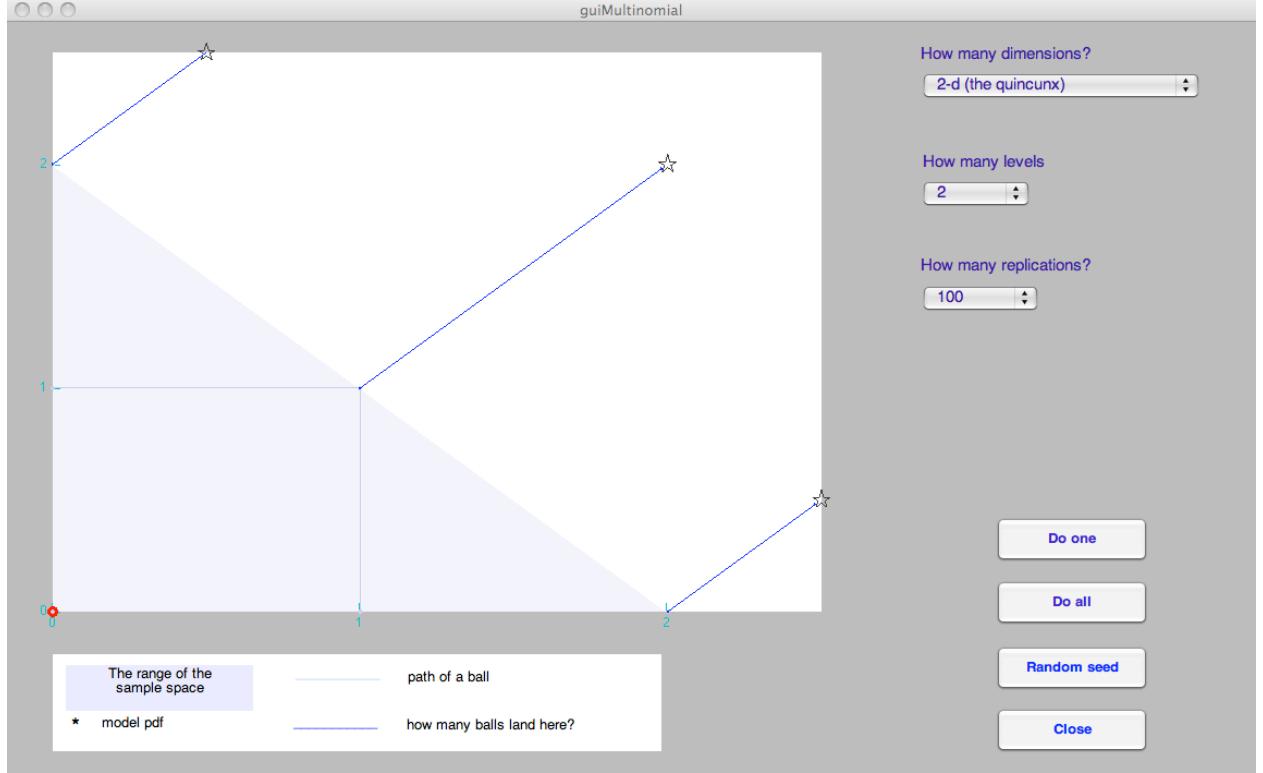
We are now ready to extend the Binomial(n, θ) RV or \vec{RV} to its multivariate version called the Multinomial($n, \theta_1, \theta_2, \dots, \theta_k$) \vec{RV} . We develop this \vec{RV} as the sum of n IID de Moivre($\theta_1, \theta_2, \dots, \theta_k$) \vec{RV} that is defined next.

Model 17 (de Moivre($\theta_1, \theta_2, \dots, \theta_k$) \vec{RV}) The PMF of the de Moivre($\theta_1, \theta_2, \dots, \theta_k$) $\vec{RV} X := (X_1, X_2, \dots, X_k)$ taking value $x := (x_1, x_2, \dots, x_k) \in \{e_1, e_2, \dots, e_k\}$, where the e_i 's are orthonormal basis vectors in \mathbb{R}^k is:

$$f(x; \theta_1, \theta_2, \dots, \theta_k) := \mathbf{P}(X = x) = \sum_{i=1}^k \theta_i \mathbf{1}_{\{e_i\}}(x) = \begin{cases} \theta_1 & \text{if } x = e_1 := (1, 0, \dots, 0) \in \mathbb{R}^k \\ \theta_1 & \text{if } x = e_1 := (0, 1, \dots, 0) \in \mathbb{R}^k \\ \vdots & \\ \theta_k & \text{if } x = e_k := (0, 0, \dots, 1) \in \mathbb{R}^k \\ 0 & \text{otherwise} \end{cases}$$

Of course, $\sum_{i=1}^k \theta_i = 1$.

Figure 6.18: Visual Cognitive Tool GUI: Quincunx.



When we add n IID $\text{de Moivre}(\theta_1, \theta_2, \dots, \theta_k)$ R \vec{V} s together, we get the $\text{Multinomial}(n, \theta_1, \theta_2, \dots, \theta_k)$ R \vec{V} s as defined below.

Model 18 ($\text{Multinomial}(n, \theta_1, \theta_2, \dots, \theta_k)$ R \vec{V}) We say that a R \vec{V} $Y := (Y_1, Y_2, \dots, Y_k)$ obtained from the sum of n IID $\text{de Moivre}(\theta_1, \theta_2, \dots, \theta_k)$ R \vec{V} s with realisations

$$y := (y_1, y_2, \dots, y_k) \in \mathbb{Y} := \{(y_1, y_2, \dots, y_k) \in \mathbb{Z}_+^k : \sum_{i=1}^k y_i = n\}$$

has the PMF given by:

$$f(y; n, \theta) := f(y; n, \theta_1, \theta_2, \dots, \theta_k) := \mathbf{P}(Y = y; n, \theta_1, \theta_2, \dots, \theta_k) = \binom{n}{y_1, y_2, \dots, y_k} \prod_{i=1}^k \theta_i^{y_i},$$

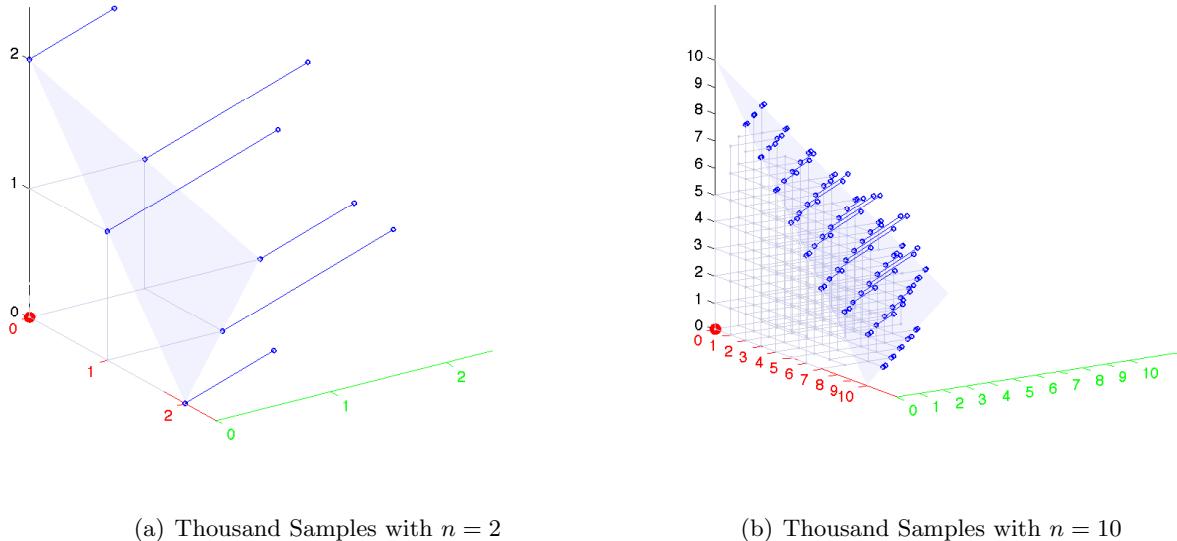
where, the multinomial coefficient:

$$\binom{n}{y_1, y_2, \dots, y_k} := \frac{n!}{y_1! y_2! \cdots y_k!}.$$

Note that the marginal PMF of Y_j is $\text{Binomial}(n, \theta_j)$ for any $j = 1, 2, \dots, k$.

We can visualise the $\text{Multinomial}(n, \theta_1, \theta_2, \theta_3)$ process as a sum of n IID $\text{de Moivre}(\theta_1, \theta_2, \theta_3)$ R \vec{V} s via a three dimensional extension of the Quincunx called the “Septcunx” and relate the number of paths that lead to a given trivariate sum (y_1, y_2, y_3) with $\sum_{i=1}^3 y_i = n$ as the multinomial coefficient $\frac{n!}{y_1! y_2! y_3!}$. In the Septcunx, balls choose from one of three paths along e_1 , e_2 and e_3 with probabilities θ_1 , θ_2 and θ_3 , respectively, in an IID manner at each of the n levels, before they collect at buckets placed at the integral points in the 3-simplex, $\mathbb{Y} = \{(y_1, y_2, y_3) \in \mathbb{Z}_+^3 : \sum_{i=1}^3 y_i = n\}$

Figure 6.19: Septcunx on the Cartesian co-ordinates. Simulations of Multinomial($n = 2, \theta_1 = 1/3, \theta_2 = 1/3, \theta_3 = 1/3$) \vec{RV} as the sum of n IID de Moivre($\theta_1 = 1/3, \theta_2 = 1/3, \theta_3 = 1/3$) \vec{RV} s over $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ with probabilities $\{\theta_1, \theta_2, \theta_3\}$, respectively. The blue lines perpendicular to the sample space of the Multinomial($3, \theta_1, \theta_2, \theta_3$) \vec{RV} , i.e. the plane in \mathbb{R}^3 connecting $(n, 0, 0)$, $(0, n, 0)$ and $(0, 0, n)$, are the density histogram of the samples.



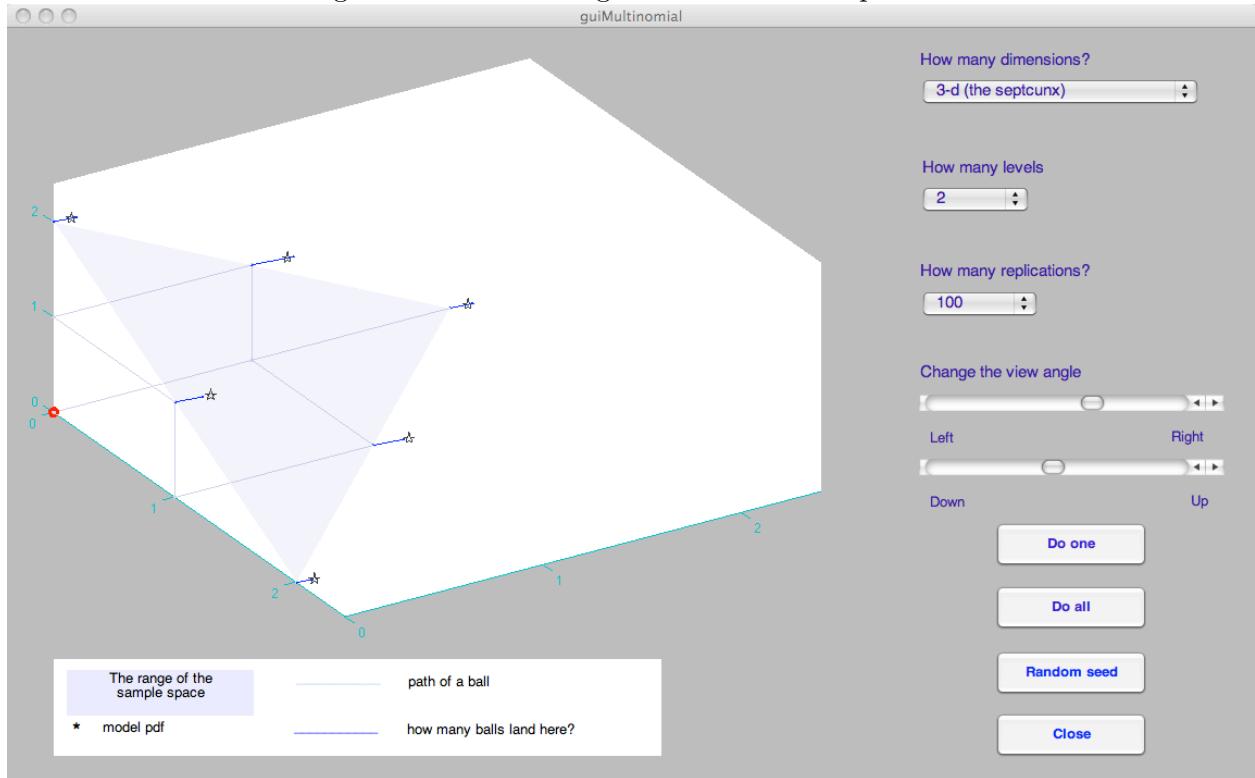
$n\}$. Once again, we can visualise that the sum of n IID de Moivre($\theta_1, \theta_2, \theta_3$) \vec{RV} s constitute the Multinomial($n, \theta_1, \theta_2, \theta_3$) \vec{RV} as depicted in Figure 6.19.

Labwork 93 (Septcunx Sampler Demo – Sum of n IID de Moivre(1/3, 1/3, 13/) \vec{RV} s) Let us understand the Septcunx construction of the Multinomial($n, 1/3, 1/3, 1/3, 1/3$) \vec{RV} X as the sum of n independent and identical de Moivre($1/3, 1/3, 13/$) \vec{RV} s by calling the interactive visual cognitive tool as follows:

```
>> guiMultinomial
```

The M-file `guiMultinomial.m` will bring a GUI as shown in Figure 6.18. Using the drop-down menu at “How many dimensions?” change to “3-d (the septcunx)” and you will see a septcunx as shown in Figure 6.20. Next, using the drop-down menu at “How many levels?” change the number of levels to 2 ($n = 2$). Now click the “Do one” button as many times as you like and comprehend the simulation process – the path taken by the ball as it falls through two levels in three dimensional space. Feel free to change the up-down and left-right sliders for the view angles. Next, from the drop-down menu at “How many Replication?” change it from 10 to 100. You can press “Do all” to watch all 100 balls drop into their possible values at level 2. Change the number of levels or n in Multinomial($n, 1/3, 1/3, 1/3, 1/3$) \vec{RV} to 5 or 10 and do more simulations until you are comfortable with the construction that the sum of n IID de Moivre($1/3, 1/3, 1/3$) \vec{RV} s is the Multinomial($n, 1/3, 1/3, 1/3, 1/3$) \vec{RV} .

Figure 6.20: Visual Cognitive Tool GUI: Septcunx.



Labwork 94 (PDF of Multinomial(n, θ) \vec{R}) We can implement the following MATLAB function `MultinomialPdf` to compute the PDF of the Multinomial(n, θ) \vec{R} where $\theta := (\theta_1, \theta_2, \dots, \theta_k)$ is a point in the k -simplex Δ_k as follows:

```
function MP = MultinomialPdf(x,n,theta)
% returns the multinomial Pdf of x(1),x(2),...,x(k) given
% theta(1),...,theta(k). x and theta are vectors and sum to
% the scalars n and 1, respectively and 0 <= x(i) <= n
% Since double precision numbers only have about 15 digits, the answer is
% only accurate for n <= 21 in factorial function.
NonZeroXs = find(x>0);
MP=exp(log(factorial(n))+sum((log(theta(NonZeroXs)) .* x(NonZeroXs)) ...
 - log(factorial(x(NonZeroXs)))));
```

We can call this function to evaluate the PDF at a specific sample $x = (x_1, x_2, \dots, x_k)$ as follows:

```
>> MultinomialPdf([2 0 0],2,[1/3 1/3 1/3])
ans =    0.1111
>> MultinomialPdf([0 2 0],2,[1/3 1/3 1/3])
ans =    0.1111
>> MultinomialPdf([0 0 2],2,[1/3 1/3 1/3])
ans =    0.1111
>> MultinomialPdf([1 1 0],2,[1/3 1/3 1/3])
ans =    0.2222
>> MultinomialPdf([1 0 1],2,[1/3 1/3 1/3])
ans =    0.2222
>> MultinomialPdf([0 1 1],2,[1/3 1/3 1/3])
ans =    0.2222
```

Simulation 95 (A simple multinomial simulation) Using the identity matrix I in \mathbb{R}^3 that can be created in MATLAB using the `eye(3)` command, and the `de Moivre(1/3, 1/3, 1/3)` RV sampler, simulate vector-valued samples from `de Moivre(1/3, 1/3, 1/3)` \vec{RV} . Finally add up $n = 10$ samples from `de Moivre(1/3, 1/3, 1/3)` \vec{RV} to produce samples from `Multinomial(10, 1/3, 1/3, 1/3)` \vec{RV} .

6.9 von Neumann Rejection Sampler (RS)

Rejection sampling [John von Neumann, 1947, in *Stanislaw Ulam 1909-1984*, a special issue of Los Alamos Science, Los Alamos National Lab., 1987, p. 135-136] is a Monte Carlo method to draw independent samples from a target RV X with probability density $f(x)$, where $x \in \mathbb{X} \subset \mathbb{R}^k$. Typically, the target density f is only known up to a constant and therefore the (normalised) density f itself may be unknown and it is difficult to generate samples directly from X .

Suppose we have another density or mass function g for which the following are true:

- (a) we can generate random variables from g ;
- (b) the support of g contains the support of f , i.e. $\mathbb{Y} \supset \mathbb{X}$;
- (c) a constant $a > 1$ exists, such that:

$$f(x) \leq ag(x). \quad (6.22)$$

for any $x \in \mathbb{X}$, the support of X . Then x can be generated from Algorithm 8.

Algorithm 8 Rejection Sampler (RS) of von Neumann

1: *input*:

- (1) a target density $f(x)$,
- (2) a proposal density $g(x)$ satisfying (a), (b) and (c) above.

2: *output*: a sample x from RV X with density f

3: **repeat**

4: Generate $y \sim g$ and $u \sim \text{Uniform}(0, 1)$

5: **until** $u \leq \frac{f(y)}{ag(y)}$

6: *return*: $x \leftarrow y$

Proposition 43 (Fundamental Theorem of Simulation) The von Neumann rejection sampler of Algorithm 8 produces a sample x from the random variable X with density $f(x)$.

Proof: We shall prove the result for the continuous case. For any real number t :

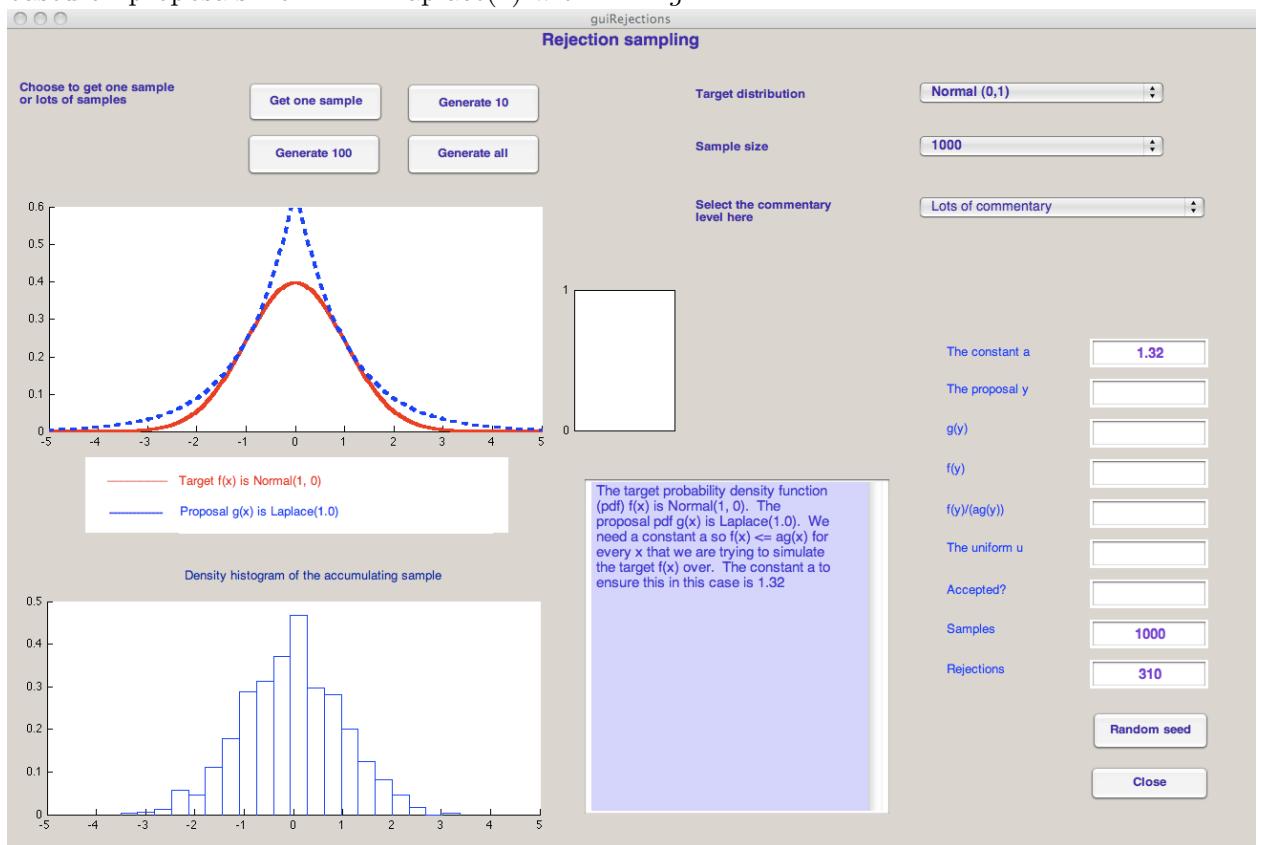
$$\begin{aligned} F(t) &= \mathbf{P}(X \leq t) = \mathbf{P}\left(Y \leq t \mid U \leq \frac{f(Y)}{ag(Y)}\right) = \frac{\mathbf{P}\left(Y \leq t, U \leq \frac{f(Y)}{ag(Y)}\right)}{\mathbf{P}\left(U \leq \frac{f(Y)}{ag(Y)}\right)} \\ &= \frac{\int_{-\infty}^t \left(\int_0^{f(y)/ag(y)} 1 du\right) g(y) dy}{\int_{-\infty}^{\infty} \left(\int_0^{f(y)/ag(y)} 1 du\right) g(y) dy} = \frac{\int_{-\infty}^t \left(\frac{f(y)}{ag(y)}\right) g(y) dy}{\int_{-\infty}^{\infty} \left(\frac{f(y)}{ag(y)}\right) g(y) dy} \\ &= \int_{-\infty}^t f(y) dy \end{aligned}$$

Labwork 96 (Rejection Sampler Demo) Let us understand the rejection sampler by calling the interactive visual cognitive tool:

```
>> guiRejections
```

The M-file `guiRejections.m` will bring a graphical user interface (GUI) as shown in Figure 6.21. Try various buttons and see how the output changes with explanations. Try switching the “Target distribution” to “Mywavy4” and generate several rejection samples and see the density histogram of the accumulating samples.

Figure 6.21: Visual Cognitive Tool GUI: Rejection Sampling from $X \sim \text{Normal}(0, 1)$ with PDF f based on proposals from $Y \sim \text{Laplace}(1)$ with PDF g .



Simulation 97 (Rejection Sampling Normal(0, 1) with Laplace(1) proposals) Suppose we wish to generate from $X \sim \text{Normal}(0, 1)$. Consider using the rejection sampler with proposals from $Y \sim \text{Laplace}(1)$ (using inversion sampler of Simulation 67). The support of both RVs is $(-\infty, \infty)$. Next:

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \text{ and } g(x) = \frac{1}{2} \exp(-|x|)$$

and therefore:

$$\frac{f(x)}{g(x)} = \sqrt{\frac{2}{\pi}} \exp\left(|x| - \frac{x^2}{2}\right) \leq \sqrt{\frac{2}{\pi}} \exp\left(\frac{1}{2}\right) = a \approx 1.3155 .$$

Hence, we can use the rejection method with:

$$\frac{f(y)}{ag(y)} = \frac{f(y)}{g(y)a} = \sqrt{\frac{2}{\pi}} \exp\left(|y| - \frac{y^2}{2}\right) \frac{1}{\sqrt{\frac{2}{\pi}} \exp\left(\frac{1}{2}\right)} = \exp\left(|y| - \frac{y^2}{2} - \frac{1}{2}\right)$$

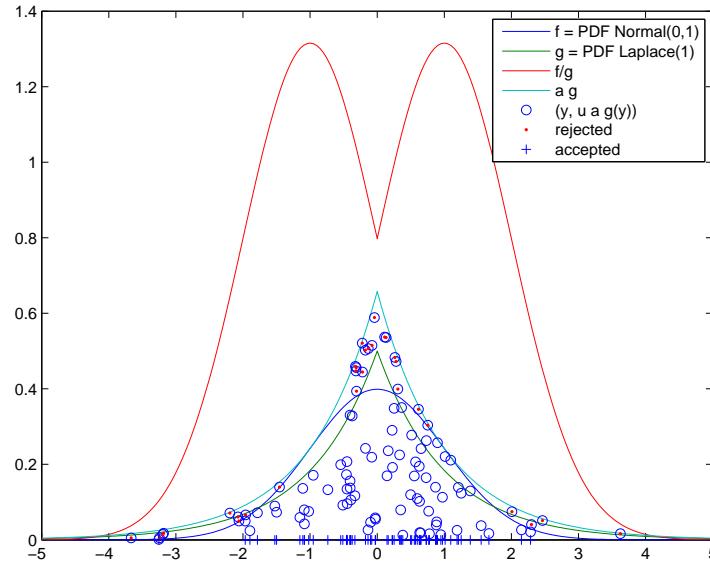
Let us implement a rejection sampler as a function in the M-file `RejectionNormalLaplace.m` by reusing the function in `LaplaceInvCDF.m`.

```
function x = RejectionNormalLaplace()
Accept = 0; % a binary variable to indicate whether a proposed point is accepted
while ~Accept % ~ is the logical NOT operation
    y = LaplaceInvCDF(rand(),1); % sample Laplace(1) RV
    Bound = exp( abs(y) - (y*y+1)/2 );
    u = rand();
    if u <= Bound
        x = y;
        Accept = 1;
    end % if
end % while
```

We may obtain a large number of samples and plot them as a histogram using the following commands:

```
>> % use funarray to convert 1000 zeros into samples from the Normal(0,1)
>> y=arrayfun(@(x)(RejectionNormalLaplace()),zeros(1,1000));
>> hist(y,20) % histogram with 20 bins
```

Figure 6.22: Rejection Sampling from $X \sim \text{Normal}(0, 1)$ with PDF f based on 100 proposals from $Y \sim \text{Laplace}(1)$ with PDF g .



Classwork 98 (A note on the proposal's tail in rejection sampling) The condition $f(x) \leq ag(x)$ is equivalent to $f(x)/g(x) \leq a$, which says that $f(x)/g(x)$ must be bounded; therefore, g must have higher tails than f . The rejection method cannot be used to generate from a Cauchy distribution using a normal distribution, because the latter has lower tails than the former.

The next result tells us how many iterations of the algorithm are needed, on average, to get a sample value from a RV with PDF f .

Proposition 44 (Acceptance Probability of RS) The expected number of iterations of the rejection algorithm to get a sample x is the constant a .

Proof: For the continuous case:

$$\mathbf{P}(\text{'accept } y') = \mathbf{P}\left(u \leq \frac{f(y)}{ag(y)}\right) = \int_{-\infty}^{\infty} \left(\int_0^{f(y)/ag(y)} du \right) g(y) dy = \int_{-\infty}^{\infty} \frac{f(y)}{ag(y)} g(y) dy = \frac{1}{a}.$$

And the number of proposals before acceptance is the Geometric($1/a$) RV with expectation $\frac{1}{1/a} = a$.

The closer $ag(x)$ is to $f(x)$, especially in the tails, the closer a will be to 1, and hence the more efficient the rejection method will be.

The rejection method can still be used only if the un-normalised form of f or g (or both) is known. In other words, if we use:

$$f(x) = \frac{\tilde{f}(x)}{\int \tilde{f}(x) dx} \text{ and } g(x) = \frac{\tilde{g}(x)}{\int \tilde{g}(x) dx}$$

we know only $\tilde{f}(x)$ and/or $\tilde{g}(x)$ in closed-form. Suppose the following are satisfied:

- (a) we can generate random variables from g ;
- (b) the support of g contains the support of f , i.e. $\mathbb{Y} \supset \mathbb{X}$;
- (c) a constant $\tilde{a} > 0$ exists, such that:

$$\tilde{f}(x) \leq \tilde{a}\tilde{g}(x), \quad (6.23)$$

for any $x \in \mathbb{X}$, the support of X . Then x can be generated from Algorithm 9.

Algorithm 9 Rejection Sampler (RS) of von Neumann – target shape

1: *input*:

- (1) shape of a target density $\tilde{f}(x) = \left(\int \tilde{f}(x) dx \right) f(x)$,
- (2) a proposal density $g(x)$ satisfying (a), (b) and (c) above.

2: *output*: a sample x from RV X with density f

3: **repeat**

4: Generate $y \sim g$ and $u \sim \text{Uniform}(0, 1)$

5: **until** $u \leq \frac{\tilde{f}(y)}{\tilde{a}\tilde{g}(y)}$

6: *return*: $x \leftarrow y$

Now, the expected number of iterations to get an x is no longer \tilde{a} but rather the integral ratio:

$$\left(\frac{\int_{\mathbb{X}} \tilde{f}(x) dx}{\int_{\mathbb{Y}} \tilde{a}\tilde{g}(y) dy} \right)^{-1}.$$

The **Ziggurat Method** [G. Marsaglia and W. W. Tsang, SIAM Journal of Scientific and Statistical Programming, volume 5, 1984] is a rejection sampler that can efficiently draw samples from the $Z \sim \text{Normal}(0, 1)$ RV. The MATLAB function `randn` uses this method to produce samples from Z . See http://www.mathworks.com/company/newsletters/news_notes/clevescorner/spring01_cleve.html or http://en.wikipedia.org/wiki/Ziggurat_algorithm for more details.

Labwork 99 (Gaussian Sampling with randn) We can use MATLAB function `randn` that implements the Ziggurat method to draw samples from an RV $Z \sim \text{Normal}(0, 1)$ as follows:

```
>> randn('state',67678); % initialise the seed at 67678 and method as Ziggurat -- TYPE help randn
>> randn % produce 1 sample from Normal(0,1) RV
ans = 1.5587
>> randn(2,8) % produce an 2 X 8 array of samples from Normal(0,1) RV
ans =
1.2558 0.7834 0.6612 0.3247 0.1407 1.0562 0.8034 1.2970
-0.5317 0.0417 -0.3454 0.6182 -1.4162 0.4796 -1.5015 0.3718
```

If we want to produce samples from $X \sim \text{Normal}(\mu, \sigma^2)$ with some user-specified μ and σ , then we can use the following relationship between X and $Z \sim \text{Normal}(0, 1)$:

$$X \leftarrow \mu + \sigma Z, \quad Z \sim \text{Normal}(0, 1).$$

Suppose we want samples from $X \sim \text{Normal}(\mu = \pi, \sigma^2 = 2)$, then we can do the following:

```
>> randn('state',679); % initialise the seed at 679 and method as Ziggurat -- TYPE help randn
>> mu=pi % set the desired mean parameter mu
mu = 3.1416
>> sigma=sqrt(2) % set the desired standard deviation parameter sigma
sigma = 1.4142
>> mu + sigma * randn(2,8) % produces a 2 X 8 array of samples from Normal(3.1416,1.4.42)
ans =
1.3955 1.7107 3.9572 3.2618 6.1652 2.6971 2.4940 4.5928
0.8442 4.7617 3.5397 5.0282 1.6139 5.0977 2.0477 2.3286
```

Labwork 100 (Sampling from truncated normal distributions) [Christian P. Robert, Simulation of truncated normal variables, Statistics and Computing (1995) 5, 121-125] Let $N_+(\mu, \tau, \sigma^2)$ denote the left-truncated normal distribution with truncation point τ and density given by

$$f(x|\mu, \tau, \sigma^2) = \frac{\exp(-(x-\mu)^2/2\sigma^2)}{\sqrt{2\pi}\sigma[1 - \Phi((\tau-\mu)/\sigma)]} \mathbb{1}_{x \geq \tau}.$$

When $\tau < \mu$, the rejection sampler can readily be used to simulate from $N_+(\mu, \tau, \sigma^2)$ by simulating from $\text{Normal}(\mu, \sigma^2)$ until a number larger than τ is obtained. When $\tau > \mu$, however, this can be inefficient and increasingly so as τ gets further out into the right tail. In this case, a more efficient approach is to use the rejection sampler with the following translated exponential distribution as the proposal distribution:

$$g(y|\lambda, \tau) = \lambda \exp(-\lambda(y-\tau)) \mathbb{1}_{y \geq \tau}.$$

1. Show that for simulating from $N_+(\mu = 0, \tau, \sigma^2 = 1)$ when $\tau \geq 0$, the best choice of λ that maximizes the expected acceptance probability for the rejection sampler is given by

$$\lambda = \frac{\tau + \sqrt{\tau^2 + 4}}{2}$$

2. Find the maximum expected acceptance probabilities for the following truncation points, $\tau = 0, 0.5, 1, 1.5, 2, 2.5$ and 3 . What can you conclude about efficiency as τ gets further out into the right tail?

3. Describe how samples from $N_+(\mu, \tau, \sigma^2)$ can be obtained by simulating from $N_+(\mu = 0, \tau, \sigma^2 = 1)$ and using location-scale transformation.
4. A related distribution, denoted by $N_-(\mu, \tau, \sigma^2)$, is the right-truncated normal distribution truncated on the right at τ . Describe how samples from $N_-(\mu, \tau, \sigma^2)$ can be obtained by simulating from an appropriate left-truncated normal distribution.
5. Write a MATLAB function that provides samples from a truncated normal distribution. The function should have the following inputs: number of samples required, left or right truncation, μ , σ^2 and τ .

6.10 Importance Resampler

The rejection method cannot be used when the constant a or \tilde{a} that guarantees the envelope condition cannot be found. The importance resampler, also known as the method of sampling/importance resampling, does not require the constant, but it produces a random variable that is only approximately distributed according to f . As for the rejection method, we need a density/mass function g that we can generate from and that has support at least as large as the support of f .

Algorithm 10 Importance Resampler

1: *input:*

- (1) shape of a target density $\tilde{f}(x) = \left(\int \tilde{f}(x) dx \right) f(x)$,
- (2) a proposal density $g(x)$ satisfying only (a) and (b) above.
- (3) a large enough integer m .

2: *output:* a sample x' from RV X' with density f' that is close to f

3: Generate $y_1, \dots, y_m \sim g$

4: Compute

$$w_i = \frac{f(y_i)/g(y_i)}{\sum_{j=1}^m f(y_j)/g(y_j)}, i = 1, \dots, m .$$

5: Resample x' from $\{y_1, \dots, y_m\}$ with weights $\{w_1, \dots, w_m\}$

Proposition 45 The Importance Resampler of Algorithm 10 produces samples from a variable X' that is approximately distributed according to f , in the sense that:

$$\lim_{m \rightarrow \infty} \mathbf{P}(X' \leq t) = \int_{-\infty}^t f(x) dx \quad (6.24)$$

for any real number t .

Proof:

$$\begin{aligned} \mathbf{P}(X' \leq t) &= \sum_{i=1}^m w_i I_{(-\infty, t]}(y_i) = \frac{\frac{1}{m} \sum_{i=1}^m \frac{f(y_i)}{g(y_i)} I_{(-\infty, t]}(y_i)}{\frac{1}{m} \sum_{i=1}^m \frac{f(y_i)}{g(y_i)}} \\ &\xrightarrow{m \rightarrow \infty} \frac{\mathbb{E}\left[\frac{f(y)}{g(y)} I_{(-\infty, t]}(y)\right]}{\mathbb{E}\left[\frac{f(y)}{g(y)}\right]} = \frac{\int_{-\infty}^t f(y) dy}{\int_{-\infty}^{\infty} f(y) dy} = \int_{-\infty}^t f(y) dy \end{aligned}$$

Let us visualise the Importance Resampler in action from Labwork 250.

Labwork 101 (Cauchy RV via Importance Resampler) Use the sampling/importance resampling method to generate 1000 approximate Cauchy samples by using the $\text{Normal}(0, 1)$ samples:

$$f(x) = \frac{1}{\pi(1+x^2)} \text{ and } g(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

```
n = 1000;
m = 10000;
y = randn(1,m); % randn is the N(0,1) generator in Matlab
y2 = y .* y;
w = exp(0.5 * y2) ./ (1 + y2);
w = w / sum(w);
x = randsample(y,n,true,w); % resample n values from y weighted by w
```

Note that to get n sample points from f using sampling/importance resampling, we must start with a sample from g of size m larger than n .

As for the rejection method, the sampling/importance resampling method can still be used if only the un-normalised form of f or g (or both) is known, simply by using the un-normalised densities/mass functions to compute the weights.

6.11 Other Continuous Random Variables

Here, we see other common continuous RVs that can be simulated from transforming RVs we have already encountered.

Simulation 102 (Gamma(λ, k) for integer k) Using this relationship we can simulate from $X \sim \text{Gamma}(\lambda, k)$, for an integer-valued k , by simply summing k IID samples from $\text{Exponential}(\lambda)$ RV as follows:

```
>> lambda=0.1; %declare some lambda parameter
>> k=5; % declare some k parameter (has to be integer)
>> rand('twister',7267); % initialise the fundamental sampler
>> % sum k IID Exponential(lambda) samples for one desired sample from Gamma(lambda,k)
>> x= sum(-1/lambda*log(rand(k,1)))
x =
    28.1401
>> % sum the 10 columns of k X 10 IID Exponential(lambda) samples for 10 desired samples from Gamma(lambda,k)
>> x= sum(-1/lambda*log(rand(k,10)))
x =
    83.8150    61.2674    80.3683   103.5748    48.4454    20.2269    93.8310    56.1909    77.0656    29.0851
```

Model 19 (Lognormal(λ, ζ)) X has a Lognormal(λ, ζ) distribution if $\log(X)$ has a $\text{Normal}(\lambda, \zeta^2)$ distribution. The location parameter $\lambda = \mathbf{E}(\log(X)) > 0$ and the scale parameter $\zeta > 0$. The PDF is:

$$f(x; \lambda, \zeta) = \frac{1}{\sqrt{2\pi}\zeta x} \exp\left(-\frac{1}{2\zeta^2}(\log(x) - \lambda)^2\right), \quad x > 0 \quad (6.25)$$

No closed form expression for $F(x; \lambda, \zeta)$ exists and it is simply defined as:

$$F(x; \lambda, \zeta) = \int_0^x f(y; \lambda, \zeta) dy$$

We can express $F(x; \lambda, \zeta)$ in terms of Φ (and, in turn, via the associated error function erf) as follows:

$$F(x; \lambda, \zeta) = \Phi\left(\frac{\log(x) - \lambda}{\zeta}\right) = \frac{1}{2} \text{erf}\left(\frac{\log(x) - \lambda}{\sqrt{2}\zeta}\right) + \frac{1}{2} \quad (6.26)$$

Labwork 103 (Simulations with the Lognormal(λ_C, ζ_C) RV) Transform a sequence of samples obtained from the fundamental sampler to those from the Lognormal(λ_C, ζ_C) RV C by using only Algorithm 4 or MATLAB's `randn` as an intermediate step. [Hint: If Y is a Normal(λ, ζ^2) RV, then $Z = e^Y$ is said to be a Lognormal(λ, ζ) RV.]

1. Seed the fundamental sampler by your Student ID,
2. generate 1000 samples from an RV $C \sim \text{Lognormal}(\lambda = 10.36, \zeta = 0.26)$ by exponentiating the samples from the Normal(10.36, 0.26²) RV and
3. and report:
 - (a) how many of the samples are larger than 35000,
 - (b) the sample mean, and
 - (c) the sample standard deviation.

Beta RV

Chi-Square

F distribution

t-distribution

Weibul

Heavy-tail family

6.12 Other Random Vectors

Multivariate Normal

Uniform Distribution on Sphere

Dirichlet Distribution

Ex. 6.21 — **The covariance of two random variables X and Y is defined as

$$\mathbf{Cov}(X, Y) := \mathbf{E}((X - \mathbf{E}(X))(Y - \mathbf{E}(Y))) = \mathbf{E}(XY) - \mathbf{E}(X)\mathbf{E}(Y) .$$

(a) Show, starting from the definition, that $\mathbf{Cov}(X, Y) = \mathbf{E}(XY) - \mathbf{E}(X)\mathbf{E}(Y)$.

(b) When $\mathbf{Cov}(X, Y) = 0$, X and Y are said to be “uncorrelated”. Show that if X and Y are independent, then they are also uncorrelated.

Ex. 6.22 — ** Let X_1, X_2, \dots, X_n be random variables. Their joint CDF is defined as

$$F(x_1, x_2, \dots, x_n) := \mathbf{P}(X_1 \leq x_1, X_2 \leq x_2, \dots, X_n \leq x_n) .$$

By repeated application of the definition of conditional probability, show that the joint CDF admits the following “telescopic” representation:

$$\begin{aligned} F(x_1, x_2, \dots, x_n) &= F(x_n | x_1, \dots, x_{n-1})F(x_{n-1} | x_1, \dots, x_{n-2}) \cdots F(x_2 | x_1)F(x_1) \\ &= F(x_1) \prod_{i=2}^n F(x_i | x_1, \dots, x_{i-1}), \end{aligned}$$

where, $F(x_i | x_1, \dots, x_{i-1})$ denotes the conditional probability, $\mathbf{P}(X_i \leq x_i | X_1 \leq x_1, \dots, X_{i-1} \leq x_{i-1})$.

6.13 Problems

Exercise 104 If $u \sim U[0, 1]$, show that the distribution of $1 - u$ is also $U[0, 1]$.

Exercise 105 Write a Matlab function to generate n random variables from the distribution with the following mass function:

x	1.7	3.4	5.9	7.2	9.6
$f(x)$	0.15	0.4	0.05	0.1	0.3

Use your Matlab function to generate 1000 sample values from the distribution, and compare the relative frequencies obtained with the mass function probabilities.

Exercise 106 The Laplacian distribution is also called the double exponential distribution because it can be regarded as the extension of the exponential distribution for both positive and negative values. An easy way to generate a Laplacian(0, 1) random variable is to generate an exponential(1) random variable and then change its sign to negative with probability 0.5. Write a Matlab function to generate n Laplacian(0, 1) random variables using the `exprnd` function from Exercise 2.6.5. Call your function `laprnd`. It should take n as input and produce a row vector containing the n Laplacian(0, 1) random variables as output.

Exercise 107 (a) Referring to Example 2.2.3, write a MATLAB function to generate n $N(0, 1)$ random variables using the rejection method with the Laplacian(0, 1) distribution. Include a counter for the number of iterations in your function.

(b) Use your Matlab function to generate 1000 $N(0, 1)$ random variables. Plot the density histogram for your generated values and superimpose the $N(0, 1)$ density onto it. Compare the average number of iterations to get a single $N(0, 1)$ random variable with the constant a .

(c) Now suppose that we know only the un-normalised $N(0, 1)$ and Laplacian(0, 1) densities, i.e.:

$$\tilde{f}(x) = \exp\left(-\frac{x^2}{2}\right) \text{ and } \tilde{g}(x) = \exp(-|x|)$$

What is the constant \tilde{a} for the rejection method in this case? Implement the rejection method in Matlab, including a counter for the number of iterations, and use it to generate 1000 $N(0, 1)$ random variables. Compare the average number of iterations to get a single $N(0, 1)$ random variable with a and \tilde{a} .

Exercise 108 Consider (Ross, p.64.) the use of the rejection method to generate from the density:

$$f(x) = 20x(1-x)^3.$$

for $0 \leq x \leq 1$, using the $U(0, 1)$ distribution as proposal distribution.

- (a) Show that the constant for using the rejection method is $a = 2.1094$.
- (b) Write a MATLAB function to generate n random variables from f using the rejection method. Include a counter for the number of iterations in your function.
- (c) Use your MATLAB function to generate 1000 random variables from f . Plot the density histogram for your generated values and superimpose the density curve onto it. Compare the average number of iterations to get a single random variable with the constant a .

Exercise 109 Consider (Ross, .p65.) the use of the rejection method to generate from the density:

$$f(x) = \frac{2}{\sqrt{\pi}} x^{1/2} e^{-x}$$

for $x \geq 0$, and using the exponential distribution with mean m as proposal distribution.

- (a) Show that the constant for using the rejection method is:

$$a = \sqrt{\frac{2}{\pi e}} \frac{m^{3/2}}{(m-1)^{1/2}}$$

- (b) Show that the best exponential distribution to use is the one with a mean of $3/2$.
- (c) Write a MATLAB function to generate n random variables from f using the rejection method. Include a counter for the number of iterations in your function.
- (d) Use your MATLAB function to generate 1000 random variables from f . Plot the density histogram for your generated values and superimpose the density curve onto it. Compare the average number of iterations to get a single random variable with the constant a .

Exercise 110 (a) Referring to Example 2.3.3, implement the Matlab function to generate 1000 approximate Cauchy($0, 1$) random variables using sampling/importance resampling, starting with $m = 10,000 N(0, 1)$ sample values. Plot the density histogram for your generated values and superimpose the Cauchy($0, 1$) density onto it.

- (b) Explore what happens if you start with (i) $m = 1000N(0, 1)$ sample values, (ii) $m = 100000N(0, 1)$ sample values.

Exercise 111 Write a MATLAB function to generate 1000 approximate Laplacian($0, 1$) random variables using sampling/importance resampling with the $N(0, 1)$ distribution. Plot the density histogram for your generated values and superimpose the Laplacian($0, 1$) density onto it.

Exercise 112 Implement the RWMH sampler in Example 2.4.8. Perform 10,000 iterations and plot the outputs sequentially. Comment on the appearance of the plot with regard to convergence to the target density. Plot the density histogram for the last 5000 iterations and superimpose the target density onto it. Investigate what happens when $g(\cdot|x) = U(x - c, x + c)$ is used as the proposal density with different values of c that are smaller or larger than 1. (Note: In MATLAB , the modified Bessel function of the first kind is available as `besseli`.)

Chapter 7

Statistical Experiments

7.1 Introduction

We formalize the notion of a staistical experiment. Let us first motivate the need for a statistical experiment. Recall that statistical inference or learning is the process of using observations or data to infer the distribution that generated it. A generic question is:

Given realizations from $X_1, X_2, \dots, X_n \sim$ some unknown DF F , how do we infer F ?

However, to make this question tractable or even sensible it is best to restrict ourselves to a particular class or family of DFs that may be assumed to contain the unknown DF F .

Definition 46 (Experiment) A statistical experiment \mathcal{E} is a set of probability distributions (DFs, PDFs or PMFs) $\mathbb{P} := \{P_\theta : \theta \in \Theta\}$ associated with a RV X and indexed by the set Θ . We refer to Θ as the parameter space or the index set and $d : \Theta \rightarrow \mathbb{P}$ that associates to each $\theta \in \Theta$ a probability $P_\theta \in \mathbb{P}$ as the index map.

7.2 Some Common Experiments

Next, let's formally consider some experiments we have already encountered.

Experiment 20 (The Fundamental Experiment) The ‘uniformly pick a number in the interval $[0, 1]$ ’ experiment is the following singleton family of DFs :

$$\mathbb{P} = \{ F(x) = x\mathbf{1}_{[0,1]}(x) \}$$

where, the only distribution $F(x)$ in the family \mathbb{P} is a re-expression of (3.7) using the indicator function $\mathbf{1}_{[0,1]}(x)$. The parameter space of the fundamental experiment is a singleton whose DF is its own inverse, ie. $F(x) = F^{[-1]}(x)$.

Experiment 21 (Bernoulli) The ‘toss 1 times’ experiment is the following family of densities (PMFs) :

$$\mathbb{P} = \{ f(x; p) : p \in [0, 1] \}$$

where, $f(x; p)$ is given in (3.4). The one dimensional parameter space or index set for this experiment is $\Theta = [0, 1] \subset \mathbb{R}$.

Figure 7.1: Geometry of the Θ 's for de Moivre[k] Experiments with $k \in \{1, 2, 3, 4\}$.

Experiment 22 (Point Mass) The ‘deterministically choose a specific real number’ experiment is the following family of DFs :

$$\mathbb{P} = \{ F(x; a) : a \in \mathbb{R} \}$$

where, $F(x; a)$ is given in (6.13). The one dimensional parameter space or index set for this experiment is $\Theta = \mathbb{R}$, the entire real line.

Note that we can use the PDF's or the DF's to specify the family \mathbb{P} of an experiment. When an experiment can be parametrized by finitely many parameters it is said to be a **parametric** experiment. Experiment 21 involving discrete RVs as well as Experiment 22 are **parametric** since they both have only one parameter (the parameter space is one dimensional for Experiments 21 and 22). The Fundamental Experiment 20 involving the continuous RV of Model 3 is also parametric since its parameter space, being a point, is zero-dimensional. The next example is also parametric and involves $(k - 1)$ -dimensional families of discrete RVs.

Experiment 23 (de Moivre[k]) The ‘pick a number from the set $[k] := \{1, 2, \dots, k\}$ somehow’ experiment is the following family of densities (PMFs) :

$$\mathbb{P} = \{ f(x; \theta_1, \theta_2, \dots, \theta_k) : (\theta_1, \theta_2, \dots, \theta_k) \in \Delta_k \}$$

where, $f(x; \theta_1, \theta_2, \dots, \theta_k)$ is any PMF such that

$$f(x; \theta_1, \theta_2, \dots, \theta_k) = \theta_x, \quad x \in \{1, 2, \dots, k\} .$$

The $k - 1$ dimensional parameter space Θ is the k -Simplex Δ_k . This as an ‘exhaustive’ experiment since all possible densities over the finite set $[k] := \{1, 2, \dots, k\}$ are being considered that can be thought of as “the outcome of rolling a convex polyhedral die with k faces and an arbitrary center of mass specified by the θ_i 's.”

An experiment with infinite dimensional parameter space Θ is said to be **nonparametric**. Next we consider two nonparametric experiments.

Experiment 24 (All DFs) The ‘pick a number from the Real line in an arbitrary way’ experiment is the following family of distribution functions (DFs) :

$$\mathbb{P} = \{ F(x; F) : F \text{ is a DF} \} = \Theta$$

where, the DF $F(x; F)$ is indexed or parameterized by itself. Thus, the parameter space

$$\Theta = \mathbb{P} = \{\text{all DFs}\}$$

is the infinite dimensional space of **All DFs**”.

Next we consider a **nonparametric** experiment involving continuous RVs.

Experiment 25 (Sobolev Densities) The ‘pick a number from the Real line in some reasonable way’ experiment is the following family of densities (pdfs) :

$$\mathbb{P} = \left\{ f(x; f) : \int (f''(x))^2 < \infty \right\} = \Theta$$

where, the density $f(x; f)$ is indexed by itself. Thus, the parameter space $\Theta = \mathbb{P}$ is the infinite dimensional **Sobolev space** of “not too wiggly functions”.

7.3 Typical Decision Problems with Experiments

Some of the concrete problems involving experiments include:

- **Simulation:** Often it is necessary to simulate a RV with some specific distribution to gain insight into its features or simulate whole systems such as the air-traffic queues at ‘London Heathrow’ to make better management decisions.
- **Estimation:**
 1. **Parametric Estimation:** Using samples from some unknown DF F parameterized by some unknown θ , we can estimate θ from a statistic T_n called the estimator of θ using one of several methods (maximum likelihood, moment estimation, or parametric bootstrap).
 2. **Nonparametric Estimation of the DF:** Based on n IID observations from an unknown DF F , we can estimate it under the general assumption that $F \in \{\text{all DFs}\}$.
 3. **Confidence Sets:** We can obtain a $1 - \alpha$ confidence set for the point estimates, of the unknown parameter $\theta \in \Theta$ or the unknown DF $F \in \{\text{all DFs}\}$
- **Hypothesis Testing:** Based on observations from some DF F that is hypothesized to belong to a subset Θ_0 of Θ called the space of null hypotheses, we will learn to test (attempt to reject) the falsifiable null hypothesis that $F \in \Theta_0 \subset \Theta$.
- ...

Chapter 8

Limits of Random Variables

8.1 Convergence of Random Variables

This important topic is concerned with the limiting behavior of sequences of RVs

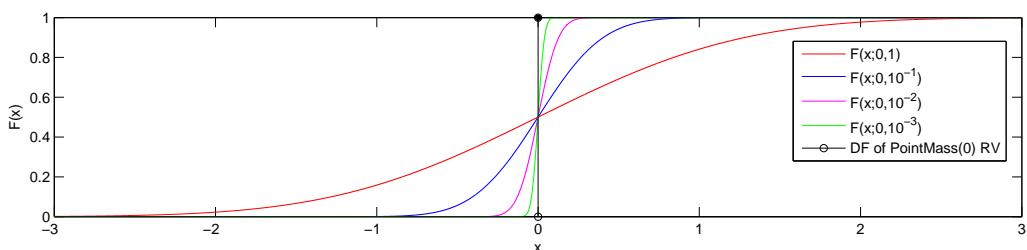
$$\{X_i\}_{i=1}^n := X_1, X_2, X_3, \dots, X_{n-1}, X_n \quad \text{as } n \rightarrow \infty.$$

From a statistical viewpoint $n \rightarrow \infty$ is associated with the amount of data or information $\rightarrow \infty$. Refresh yourself with notions of convergence, limits and continuity in the real line (**S 1.6**) before proceeding further.

Classwork 113 (Convergence of $X_i \sim \text{Normal}(0, 1/i)$) Suppose you are given an independent sequence of RVs $\{X_i\}_{i=1}^n$, where $X_i \sim \text{Normal}(0, 1/i)$. How would you talk about the convergence of $X_n \sim \text{Normal}(0, 1/n)$ as n approaches ∞ ? Take a look at Figure 8.1 for insight. The probability mass of X_n increasingly concentrates about 0 as n approaches ∞ and the variance $1/n$ approaches 0, as depicted in Figure 8.1. Based on this observation, can we expect $\lim_{n \rightarrow \infty} X_n = X$, where the limiting RV $X \sim \text{Point Mass}(0)$?

The answer is **no**. This is because $\mathbf{P}(X_n = X) = 0$ for any n , since $X \sim \text{Point Mass}(0)$ is a discrete RV with exactly one outcome 0 and $X_n \sim \text{Normal}(0, 1/n)$ is a continuous RV for every n , however large. In other words, a continuous RV, such as X_n , has 0 probability of realizing any single real number in its support, such as 0.

Figure 8.1: Distribution functions of several $\text{Normal}(\mu, \sigma^2)$ RVs for $\sigma^2 = 1, \frac{1}{10}, \frac{1}{100}, \frac{1}{1000}$.



Thus, we need more sophisticated notions of convergence for sequences of RVs. Two such notions are formalized next as they are minimal prerequisites for a clear understanding of three basic propositions in Statistics :

1. Weak Law of Large Numbers,
2. Central Limit Theorem,
3. Gilvenko-Cantelli Theorem.

Definition 47 (Convergence in Distribution) Let X_1, X_2, \dots , be a sequence of RVs and let X be another RV. Let F_n denote the DF of X_n and F denote the DF of X . Then we say that X_n converges to X in distribution, and write:

$$X_n \rightsquigarrow X$$

if for any real number t at which F is continuous,

$$\lim_{n \rightarrow \infty} F_n(t) = F(t) \quad [\text{in the sense of Definition 4}].$$

The above limit, by (3.2) in our Definition 17 of a DF, can be equivalently expressed as follows:

$$\begin{aligned} \lim_{n \rightarrow \infty} \mathbf{P}(\{\omega : X_n(\omega) \leq t\}) &= \mathbf{P}(\{\omega : X(\omega) \leq t\}), \\ \text{i.e. } \mathbf{P}(\{\omega : X_n(\omega) \leq t\}) &\rightarrow \mathbf{P}(\{\omega : X(\omega) \leq t\}), \quad \text{as } n \rightarrow \infty. \end{aligned}$$

Definition 48 (Convergence in Probability) Let X_1, X_2, \dots , be a sequence of RVs and let X be another RV. Let F_n denote the DF of X_n and F denote the DF of X . Then we say that X_n converges to X in probability, and write:

$$X_n \xrightarrow{P} X$$

if for every real number $\epsilon > 0$,

$$\lim_{n \rightarrow \infty} \mathbf{P}(|X_n - X| > \epsilon) = 0 \quad [\text{in the sense of Definition 4}].$$

Once again, the above limit, by (3.1) in our Definition 16 of a RV, can be equivalently expressed as follows:

$$\lim_{n \rightarrow \infty} \mathbf{P}(\{\omega : |X_n(\omega) - X(\omega)| > \epsilon\}) = 0, \quad \text{ie, } \mathbf{P}(\{\omega : |X_n(\omega) - X(\omega)| > \epsilon\}) \rightarrow 0, \quad \text{as } n \rightarrow \infty.$$

Let us revisit the problem of convergence in Classwork 113 armed with our new notions of convergence.

Example 114 (Convergence in distribution) Suppose you are given an independent sequence of RVs $\{X_i\}_{i=1}^n$, where $X_i \sim \text{Normal}(0, 1/i)$ with DF F_n and let $X \sim \text{Point Mass}(0)$ with DF F . We can formalize our observation in Classwork 113 that X_n is concentrating about 0 as $n \rightarrow \infty$ by the statement:

$$X_n \text{ is converging in distribution to } X, \text{ ie, } X_n \rightsquigarrow X.$$

Proof: To check that the above statement is true we need to verify that the definition of convergence in distribution is satisfied for our sequence of RVs X_1, X_2, \dots and the limiting RV X . Thus, we need to verify that for any continuity point t of the Point Mass(0) DF F , $\lim_{n \rightarrow \infty} F_n(t) = F(t)$. First note that

$$X_n \sim \text{Normal}(0, 1/n) \implies Z := \sqrt{n}X_n \sim \text{Normal}(0, 1),$$

and thus

$$F_n(t) = \mathbf{P}(X_n < t) = \mathbf{P}(\sqrt{n}X_n < \sqrt{nt}) = \mathbf{P}(Z < \sqrt{nt}).$$

The only discontinuous point of F is 0 where F jump from 0 to 1.

When $t < 0$, $F(t)$, being the constant 0 function over the interval $(-\infty, 0)$, is continuous at t . Since $\sqrt{nt} \rightarrow -\infty$, as $n \rightarrow \infty$,

$$\lim_{n \rightarrow \infty} F_n(t) = \lim_{n \rightarrow \infty} \mathbf{P}(Z < \sqrt{nt}) = 0 = F(t) .$$

And, when $t > 0$, $F(t)$, being the constant 1 function over the interval $(0, \infty)$, is again continuous at t . Since $\sqrt{nt} \rightarrow \infty$, as $n \rightarrow \infty$,

$$\lim_{n \rightarrow \infty} F_n(t) = \lim_{n \rightarrow \infty} \mathbf{P}(Z < \sqrt{nt}) = 1 = F(t) .$$

Thus, we have proved that $X_n \rightsquigarrow X$ by verifying that for any t at which the Point Mass(0) DF F is continuous, we also have the desired equality: $\lim_{n \rightarrow \infty} F_n(t) = F(t)$.

However, note that

$$F_n(0) = \frac{1}{2} \neq F(0) = 1 ,$$

and so convergence fails at 0, i.e. $\lim_{n \rightarrow \infty} F_n(t) \neq F(t)$ at $t = 0$. But, $t = 0$ is not a continuity point of F and the definition of convergence in distribution only requires the convergence to hold at continuity points of F .

For the same sequence of RVs in Classwork 113 and Example 114 we are tempted to ask whether $X_n \sim \text{Normal}(0, 1/n)$ converges in probability to $X \sim \text{Point Mass}(0)$, i.e. whether $X_n \xrightarrow{P} X$. We need some elementary inequalities in Probability to help us answer this question. We visit these inequalities next.

Proposition 49 (Markov's Inequality) Let (Ω, \mathcal{F}, P) be a probability triple and let $X = X(\omega)$ be a non-negative RV. Then,

$$\mathbf{P}(X \geq \epsilon) \leq \frac{\mathbf{E}(X)}{\epsilon}, \quad \text{for any } \epsilon > 0 . \quad (8.1)$$

Proof:

$$\begin{aligned} X &= X\mathbf{1}_{\{y:y \geq \epsilon\}}(x) + X\mathbf{1}_{\{y:y < \epsilon\}}(x) \\ &\geq X\mathbf{1}_{\{y:y \geq \epsilon\}}(x) \\ &\geq \epsilon\mathbf{1}_{\{y:y \geq \epsilon\}}(x) \end{aligned} \quad (8.2)$$

Finally, taking expectations on both sides of the above inequality and then using the fact that the expectation of an indicator function of an event is simply the probability of that event (3.14), we get the desired result:

$$\mathbf{E}(X) \geq \epsilon\mathbf{E}(\mathbf{1}_{\{y:y \geq \epsilon\}}(x)) = \epsilon\mathbf{P}(X \geq \epsilon) .$$

Let us look at some immediate consequences of Markov's inequality.

Proposition 50 (Chebychev's Inequality) For any RV X and any $\epsilon > 0$,

$$\mathbf{P}(|X| > \epsilon) \leq \frac{\mathbf{E}(|X|)}{\epsilon} \quad (8.3)$$

$$\mathbf{P}(|X| > \epsilon) = \mathbf{P}(X^2 \geq \epsilon^2) \leq \frac{\mathbf{E}(X^2)}{\epsilon^2} \quad (8.4)$$

$$\mathbf{P}(|X - \mathbf{E}(X)| \geq \epsilon) = \mathbf{P}((X - \mathbf{E}(X))^2 \geq \epsilon^2) \leq \frac{\mathbf{E}(X - \mathbf{E}(X))^2}{\epsilon^2} = \frac{\mathbf{V}(X)}{\epsilon^2} \quad (8.5)$$

Proof: All three forms of Chebychev's inequality are mere corollaries (careful reapplications) of Markov's inequality.

Armed with Markov's inequality we next enquire the convergence in probability for the sequence of RVs in Classwork 113 and Example 114.

Example 115 (Convergence in probability) Does the sequence of RVs $\{X_n\}_{n=1}^{\infty}$, where $X_n \sim \text{Normal}(0, 1/n)$, converge in probability to $X \sim \text{Point Mass}(0)$, i.e. does $X_n \xrightarrow{P} X$?

To find out if $X_n \xrightarrow{P} X$, we need to show that for any $\epsilon > 0$, $\lim_{n \rightarrow \infty} \mathbf{P}(|X_n - X| > \epsilon) = 0$.

Let ϵ be any real number greater than 0, then

$$\begin{aligned}\mathbf{P}(|X_n| > \epsilon) &= \mathbf{P}(|X_n|^2 > \epsilon^2) \\ &= \frac{\mathbf{E}(X_n^2)}{\epsilon^2} \quad [\text{by Markov's Inequality (8.1)}] \\ &= \frac{\frac{1}{n}}{\epsilon^2} \rightarrow 0, \quad \text{as } n \rightarrow \infty \quad [\text{in the sense of Definition 4].}\end{aligned}$$

Hence, we have shown that for any $\epsilon > 0$, $\lim_{n \rightarrow \infty} \mathbf{P}(|X_n - X| > \epsilon) = 0$ and therefore by Definition 48, $X_n \xrightarrow{P} X$ or $X_n \xrightarrow{P} 0$.

Convention: When X has a Point Mass(θ) distribution and $X_n \xrightarrow{P} X$, we simply write $X_n \xrightarrow{P} \theta$.

Now that we have been introduced to two notions of convergence for sequences of RVs we can begin to appreciate the statements of the basic limit theorems of Statistics.

8.2 Some Basic Limit Laws of Statistics

Proposition 51 (Weak Law of Large Numbers (WLLN)) If we are given a sequence of independent and identically distributed RVs, $X_1, X_2, \dots \stackrel{\text{IID}}{\sim} X_1$ and if $\mathbf{E}(X_1)$ exists, as per (3.9), then the sample mean \bar{X}_n converges in probability to the expectation of any one of the IID RVs, say $\mathbf{E}(X_1)$ by convention. More formally, we write:

$$\text{If } X_1, X_2, \dots \stackrel{\text{IID}}{\sim} X_1 \text{ and if } \mathbf{E}(X_1) \text{ exists, then } \bar{X}_n \xrightarrow{P} \mathbf{E}(X_1).$$

Proof: For simplicity, we will prove a slightly weaker result by assuming finite variance of X_1 . Suppose $\mathbf{V}(X_1) < \infty$, then:

$$\begin{aligned}\mathbf{P}(|\bar{X}_n - \mathbf{E}(\bar{X}_n)| \geq \epsilon) &= \frac{\mathbf{V}(\bar{X}_n)}{\epsilon^2} \quad [\text{by applying Chebychev's inequality (8.5) to the RV } \bar{X}_n] \\ &= \frac{\frac{1}{n}\mathbf{V}(X_1)}{\epsilon^2} \quad [\text{by the IID assumption of } X_1, X_2, \dots \text{ we can apply (5.3)}]\end{aligned}$$

Therefore, for any given $\epsilon > 0$,

$$\begin{aligned}\mathbf{P}(|\bar{X}_n - \mathbf{E}(\bar{X}_n)| \geq \epsilon) &= \mathbf{P}(|\bar{X}_n - \mathbf{E}(\bar{X}_n)| \geq \epsilon) \quad [\text{by the IID assumption of } X_1, X_2, \dots, \mathbf{E}(\bar{X}_n) = \mathbf{E}(X_1), \text{ as per (5.2)}] \\ &= \frac{\frac{1}{n}\mathbf{V}(X_1)}{\epsilon^2} \rightarrow 0, \quad \text{as } n \rightarrow \infty,\end{aligned}$$

or equivalently, $\lim_{n \rightarrow \infty} \mathbf{P}(|\bar{X}_n - \mathbf{E}(\bar{X}_n)| \geq \epsilon) = 0$. And the last statement is the definition of the claim made by the weak law of large numbers (WLLN), namely that $\bar{X}_n \xrightarrow{P} \mathbf{E}(X_1)$.

Heuristic Interpretation of WLLN: The distribution of the sample mean RV \bar{X}_n obtained from an independent and identically distributed sequence of RVs X_1, X_2, \dots [i.e. all the RVs X_i 's are independent of one another and have the same distribution function, and thereby the same expectation, variance and higher moments], concentrates around the expectation of any one of the RVs in the sequence, say that of the first one $\mathbf{E}(X_1)$ [without loss of generality], as n approaches infinity.

Example 116 (Bernoulli WLLN and Galton's Quincunx) We can appreciate the WLLN for $\bar{X}_n = n^{-1}S_n = \sum_{i=1}^n X_i$, where $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Bernoulli}(p)$ using the paths of balls dropped into a device built by Galton called the Quincunx.

Proposition 52 (Central Limit Theorem (CLT)) Let $X_1, X_2, \dots \stackrel{\text{IID}}{\sim} X_1$ and suppose $\mathbf{E}(X_1)$ and $\mathbf{V}(X_1)$ exists, then

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i \rightsquigarrow X \sim \text{Normal}\left(\mathbf{E}(X_1), \frac{\mathbf{V}(X_1)}{n}\right), \quad (8.6)$$

$$\bar{X}_n - \mathbf{E}(X_1) \rightsquigarrow X - \mathbf{E}(X_1) \sim \text{Normal}\left(0, \frac{\mathbf{V}(X_1)}{n}\right), \quad (8.7)$$

$$\sqrt{n}(\bar{X}_n - \mathbf{E}(X_1)) \rightsquigarrow \sqrt{n}(X - \mathbf{E}(X_1)) \sim \text{Normal}(0, \mathbf{V}(X_1)), \quad (8.8)$$

$$Z_n := \frac{\bar{X}_n - \mathbf{E}(\bar{X}_n)}{\sqrt{\mathbf{V}(\bar{X}_n)}} = \frac{\sqrt{n}(\bar{X}_n - \mathbf{E}(X_1))}{\sqrt{\mathbf{V}(X_1)}} \rightsquigarrow Z \sim \text{Normal}(0, 1), \quad (8.9)$$

$$\lim_{n \rightarrow \infty} P\left(\frac{\bar{X}_n - \mathbf{E}(\bar{X}_n)}{\sqrt{\mathbf{V}(\bar{X}_n)}} \leq z\right) = \lim_{n \rightarrow \infty} \mathbf{P}(Z_n \leq z) = \Phi(z) := \int_{-\infty}^z \left(\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right)\right) dx. \quad (8.10)$$

Thus, for sufficiently large n (say $n > 30$) we can make the following approximation:

$$P\left(\frac{\bar{X}_n - \mathbf{E}(\bar{X}_n)}{\sqrt{\mathbf{V}(\bar{X}_n)}} \leq z\right) \approx \mathbf{P}(Z \leq z) = \Phi(z) := \int_{-\infty}^z \left(\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right)\right) dx. \quad (8.11)$$

Proof: See any intermediate to advanced undergraduate text in Probability. Start from the index looking for "Central Limit Theorem" to find the page number for the proof

Heuristic Interpretation of CLT: Probability statements about the sample mean RV \bar{X}_n can be approximated using a Normal distribution.

Here is a simulation showing CLT in action.

```
>> % a demonstration of Central Limit Theorem --
>> % the sample mean of a sequence of n IID Exponential(lambda) RVs
>> % itself a Gaussian(1/lambda,lambda/n) RV
>> lambda=0.1; Reps=10000; n=10; hist(sum(-1/lambda * log(rand(n,Reps))/n))
>> lambda=0.1; Reps=10000; n=100; hist(sum(-1/lambda * log(rand(n,Reps))/n,20)
>> lambda=0.1; Reps=10000; n=1000; hist(sum(-1/lambda * log(rand(n,Reps))/n,20)
```

Let us look at an example that makes use of the CLT next.

Example 117 (Errors in computer code (Wasserman03, p. 78)) Suppose the collection of RVs X_1, X_2, \dots, X_n model the number of errors in n computer programs named $1, 2, \dots, n$, respectively. Suppose that the RV X_i modeling the number of errors in the i -th program is the $\text{Poisson}(\lambda = 5)$ for any $i = 1, 2, \dots, n$. Further suppose that they are independently distributed. Succinctly, we suppose that

$$X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Poisson}(\lambda = 5).$$

Suppose we have $n = 125$ programs and want to make a probability statement about \bar{X}_n which is the average error per program out of these 125 programs. Since $\mathbf{E}(X_i) = \lambda = 5$ and $\mathbf{V}(X_i) = \lambda = 5$,

we may want to know how often our sample mean \bar{X}_{125} differs from the expectation of 5 errors per program. Using the CLT we can approximate $\mathbf{P}(\bar{X}_n < 5.5)$, for instance, as follows:

$$\begin{aligned}\mathbf{P}(\bar{X}_n < 5.5) &= P\left(\frac{\sqrt{n}(\bar{X}_n - \mathbf{E}(X_1))}{\sqrt{\mathbf{V}(X_1)}} < \frac{\sqrt{n}(5.5 - \mathbf{E}(X_1))}{\sqrt{\mathbf{V}(X_1)}}\right) \\ &\approx P\left(Z < \frac{\sqrt{n}(5.5 - \lambda)}{\sqrt{\lambda}}\right) \quad [\text{by (8.11), and } \mathbf{E}(X_1) = \mathbf{V}(X_1) = \lambda] \\ &= P\left(Z < \frac{\sqrt{125}(5.5 - 5)}{\sqrt{5}}\right) \quad [\text{Since, } \lambda = 5 \text{ and } n = 125 \text{ in this Example}] \\ &= \mathbf{P}(Z \leq 2.5) = \Phi(2.5) = \int_{-\infty}^{2.5} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \right) dx \approx 0.993790334674224.\end{aligned}$$

The last number above needed the following:

Labwork 118 (Numerical approximation of $\Phi(2.5)$) The numerical approximation of $\Phi(2.5)$ was obtained via the following call to our erf-based `NormalCdf` function from 245.

```
>> format long
>> disp(NormalCdf(2.5,0,1))
0.993790334674224
```

The CLT says that if $X_1, X_2, \dots \stackrel{\text{IID}}{\sim} X_1$, then $Z_n := \sqrt{n}(\bar{X}_n - \mathbf{E}(X_1))/\sqrt{\mathbf{V}(X_1)}$ is approximately distributed as $\text{Normal}(0, 1)$. In Example 117, we knew $\sqrt{\mathbf{V}(X_1)}$. However, in general, we may not know $\sqrt{\mathbf{V}(X_1)}$. The next proposition says that we may estimate $\sqrt{\mathbf{V}(X_1)}$ using the sample standard deviation S_n of X_1, X_2, \dots, X_n , according to (5.5), and still make probability statements about the sample mean \bar{X}_n using a Normal distribution.

Proposition 53 (CLT based on Sample Variance) Let $X_1, X_2, \dots \stackrel{\text{IID}}{\sim} X_1$ and suppose $\mathbf{E}(X_1)$ and $\mathbf{V}(X_1)$ exists, then

$$\frac{\sqrt{n}(\bar{X}_n - \mathbf{E}(X_1))}{S_n} \rightsquigarrow \text{Normal}(0, 1). \quad (8.12)$$

We will use (8.12) for statistical estimation in the sequel.

Chapter 9

Finite Markov Chains

When a stochastic process $(X_\alpha)_{\alpha \in \mathbb{A}}$ is not independent it is said to be dependent. So far we have mostly concerned ourselves with independent processes. In this chapter we introduce finite Markov chains and their simulation methods. Finite Markov chains are among the simplest stochastic processes with a ‘first-order’ dependence called Markov dependence.

9.1 Introduction

A finite Markov chain is a stochastic process that moves among elements in a finite set \mathbb{X} as follows: when at $x \in \mathbb{X}$ the next position is chosen at random according to a fixed probability distribution $P(\cdot|x)$. We define such a process more formally below.

Definition 54 (Finite Markov Chain) A stochastic sequence,

$$(X_n)_{n \in \mathbb{Z}_+} := (X_0, X_1, \dots),$$

is a homogeneous **Markov chain** with **state space** \mathbb{X} and **transition matrix** $P := (P(x, y))_{(x,y) \in \mathbb{X}^2}$ if for all pair of **states** $(x, y) \in \mathbb{X}^2 := \mathbb{X} \times \mathbb{X}$, all integers $t \geq 1$, and all probable historical events $H_{t-1} := \bigcap_{n=0}^{t-1} \{X_n = x_n\}$ with $\mathbf{P}(H_{t-1} \cap \{X_t = x\}) > 0$, the following **Markov property** is satisfied:

$$\mathbf{P}(X_{t+1} = y | H_{t-1} \cap \{X_t = x\}) = \mathbf{P}(X_{t+1} = y | X_t = x) =: P(x, y). \quad (9.1)$$

The Markov property means that the conditional probability of going to state y at time $t + 1$ from state x at current time t is always given by the (x, y) -th entry $P(x, y)$ of the transition matrix P , no matter what sequence of states $(x_0, x_1, \dots, x_{t-1})$ preceded the current state x . Thus, the $|\mathbb{X}| \times |\mathbb{X}|$ matrix P is enough to obtain the state transitions since the x -th row of P is the probability distribution $P(x, \cdot) := (P(x, y))_{y \in \mathbb{X}}$. For this reason P is called a **stochastic matrix**, i.e.,

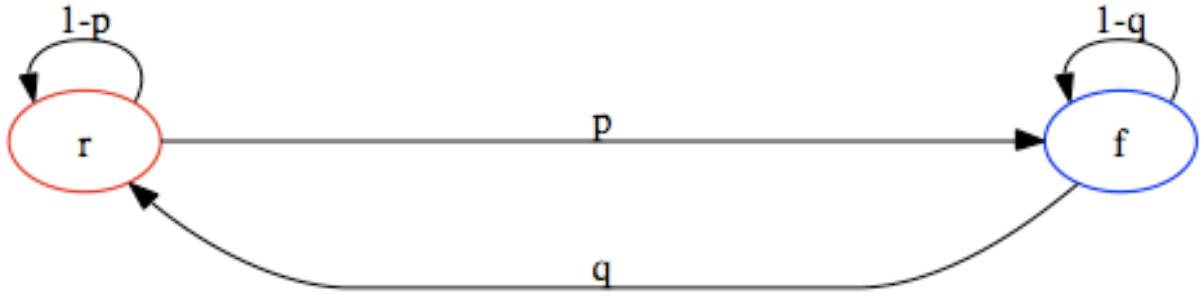
$$P(x, y) \geq 0 \quad \text{for all } (x, y) \in \mathbb{X}^2 \quad \text{and} \quad \sum_{y \in \mathbb{X}} P(x, y) = 1 \quad \text{for all } x \in \mathbb{X}. \quad (9.2)$$

Thus, for a Markov chain $(X_n)_{n \in \mathbb{Z}_+}$, the distribution of X_{t+1} given X_0, \dots, X_t depends on X_t alone. Because of this dependence on the previous state, the stochastic sequence, (X_0, X_1, \dots) , are *not* independent. We introduce the most important concepts using a simple example.

Example 119 (Flippant Freddy) Freddy the flippant frog lives in an enchanted pond with only two lily pads, *rollopia* and *flipopia*. A wizard gave a die and a silver coin to help flippant Freddy decide where to jump next. Freddy left the die on rollopia and the coin on flipopia. When Freddy got restless in rollopia he would roll the die and if the die landed odd he would leave the die behind and jump to flipopia, otherwise he would stay put. When Freddy got restless in flipopia he would flip the coin and if it landed Heads he would leave the coin behind and jump to rollopia, otherwise he would stay put.

Let the state space $\mathbb{X} = \{r, f\}$, and let (X_0, X_1, \dots) be the sequence of lily pads occupied by Freddy after his restless moments. Say the die on rollopia r has probability p of turning up odd and the coin on flipopia f has probability q of turning up heads. We can visualise the rules of Freddy's jumps by the following **transition diagram**:

Figure 9.1: Transition Diagram of Flippant Freddy's Jumps.



Then Freddy's sequence of jumps (X_0, X_1, \dots) is a Markov chain on \mathbb{X} with transition matrix:

$$P = \begin{matrix} r & f \\ \begin{matrix} r \\ f \end{matrix} & \begin{pmatrix} P(r,r) & P(r,f) \\ P(f,r) & P(f,f) \end{pmatrix} \end{matrix} = \begin{matrix} r & f \\ \begin{matrix} r \\ f \end{matrix} & \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix} \end{matrix}. \quad (9.3)$$

Suppose we first see Freddy in rollopia, i.e., $X_0 = r$. When he gets restless for the first time we know from the first row of P that he will leave to flipopia with probability p and stay with probability $1 - p$, i.e.,

$$\mathbf{P}(X_1 = f | X_0 = r) = p, \quad \mathbf{P}(X_1 = r | X_0 = r) = 1 - p. \quad (9.4)$$

What happens when he is restless for the second time? By considering the two possibilities for X_1 ,

Definition of conditional probability and the Markov property, we see that,

$$\begin{aligned}
 \mathbf{P}(X_2 = f | X_0 = r) &= \mathbf{P}(X_2 = f, X_1 = f | X_0 = r) + \mathbf{P}(X_2 = f, X_1 = r | X_0 = r) \\
 &= \frac{\mathbf{P}(X_2 = f, X_1 = f, X_0 = r)}{\mathbf{P}(X_0 = r)} + \frac{\mathbf{P}(X_2 = f, X_1 = r, X_0 = r)}{\mathbf{P}(X_0 = r)} \\
 &= \mathbf{P}(X_2 = f | X_1 = f, X_0 = r) \frac{\mathbf{P}(X_1 = f, X_0 = r)}{\mathbf{P}(X_0 = r)} \\
 &\quad + \mathbf{P}(X_2 = f | X_1 = r, X_0 = r) \frac{\mathbf{P}(X_1 = r, X_0 = r)}{\mathbf{P}(X_0 = r)} \\
 &= \mathbf{P}(X_2 = f | X_1 = f, X_0 = r) \mathbf{P}(X_1 = f | X_0 = r) \\
 &\quad + \mathbf{P}(X_2 = f | X_1 = r, X_0 = r) \mathbf{P}(X_1 = r | X_0 = r) \\
 &= \mathbf{P}(X_2 = f | X_1 = f) \mathbf{P}(X_1 = f | X_0 = r) \\
 &\quad + \mathbf{P}(X_2 = f | X_1 = r) \mathbf{P}(X_1 = r | X_0 = r) \\
 &= P(f, f)P(r, f) + P(r, f)P(r, r) \\
 &= (1 - q)p + p(1 - p)
 \end{aligned} \tag{9.5}$$

Similarly,

$$\mathbf{P}(X_2 = r | X_0 = r) = P(f, r)P(r, f) + P(r, r)P(r, r) = qp + (1 - p)(1 - p) \tag{9.6}$$

Instead of elaborate computations of the probabilities of being in a given state after Freddy's t -th restless moment, we can store the state probabilities at time t in a row vector:

$$\mu_t := (\mathbf{P}(X_t = r | X_0 = r), \mathbf{P}(X_t = f | X_0 = r)) ,$$

Now, we can conveniently represent Freddy starting in rollopia by the **initial distribution** $\mu_0 = (1, 0)$ and obtain the 1-step **state probability vector** in (9.4) from $\mu_1 = \mu_0 P$ and the 2-step state probabilities in (9.5) and (9.6) by $\mu_2 = \mu_1 P = \mu_0 P P = \mu_0 P^2$. In general, multiplying μ_t , the state probability vector at time t , by the transition matrix P on the right updates the state probabilities by another step:

$$\mu_t = \mu_{t-1} P \quad \text{for all } t \geq 1 .$$

And for any initial distribution μ_0 ,

$$\mu_t = \mu_0 P^t \quad \text{for all } t \geq 0 .$$

This can be easily implemented in MATLAB as follows:

```

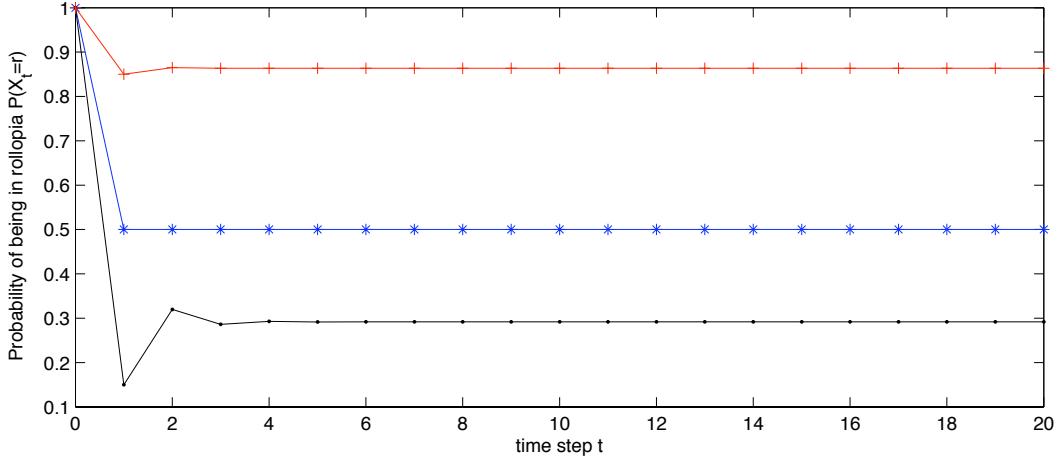
>> p=0.85; q=0.35; P = [1-p p; q 1-q] % assume an unfair coin and an unfair die
P =
    0.1500    0.8500
    0.3500    0.6500
>> mu0 = [1, 0] % initial state vector since Freddy started in rollopia
mu0 =
    1         0
>> mu0*P^0    % initial state distribution at t=0 is just mu0
ans =
    1         0
>> mu0*P^1    % state distribution at t=1
ans =
    0.1500    0.8500
>> mu0*P^2    % state distribution at t=2
ans =
    0.3200    0.6800
>> mu0*P^3    % state distribution at t=3
ans =
    0.2860    0.7140

```

Now, let us compute and look at the probability of being in rollopia after having started there for three values of p and q according to the following script:

```
FlippantFreddyRollopiaProbs.m
p=0.5; q=0.5; P = [1-p p; q 1-q]; % assume a fair coin and a fair die
mu0 = [1, 0]; % initial state vector since Freddy started in rollopia
for t = 1: 1: 21, mut(t,:)= mu0*P^(t-1); end
t=0:1:20; % vector of time steps t
plot(t,mut(:,1)', 'b*-')
hold on;
p=0.85; q=0.35; P = [1-p p; q 1-q]; % assume an unfair coin and an unfair die
for t = 1: 1: 21, mut(t,:)= mu0*P^(t-1); end
t=0:1:20; % vector of time steps t
plot(t,mut(:,1)', 'k.-')
p=0.15; q=0.95; P = [1-p p; q 1-q]; % assume another unfair coin and another unfair die
for t = 1: 1: 21, mut(t,:)= mu0*P^(t-1); end
t=0:1:20; % vector of time steps t
plot(t,mut(:,1)', 'r+-')
xlabel('time step t'); ylabel('Probability of being in rollopia $P(X_t=r)$')
xlabel('time step t'); ylabel('Probability of being in rollopia $P(X_t=r)$')
```

Figure 9.2: The probability of being back in rollopia in t time steps after having started there under transition matrix P with (i) $p = q = 0.5$ (blue line with asterisks), (ii) $p = 0.85, q = 0.35$ (black line with dots) and (iii) $p = 0.15, q = 0.95$ (red line with pluses).



It is evident from Figure 9.2 that as $t \rightarrow \infty$, μ_t approaches a distribution, say π , that depends on p and q in P . Such a limit distribution is called the **stationary distribution** and must satisfy the fixed point condition:

$$\pi P = \pi ,$$

that gives the solution:

$$\pi(r) = \frac{q}{p+q}, \quad \pi(f) = \frac{p}{p+q} .$$

In Figure 9.2 we see that $\mathbf{P}(X_t = r)$ approaches $\pi(r) = \frac{q}{p+q}$ for the three cases of p and q :

$$\begin{aligned}
 & \text{(i) } p = 0.50, q = 0.50, & \mathbf{P}(X_t = r) \rightarrow \pi(r) = \frac{q}{p+q} = \frac{0.50}{0.50 + 0.50} = 0.5000, \\
 & \text{(ii) } p = 0.85, q = 0.35, & \mathbf{P}(X_t = r) \rightarrow \pi(r) = \frac{q}{p+q} = \frac{0.35}{0.85 + 0.35} = 0.2917, \\
 & \text{(iii) } p = 0.15, q = 0.95, & \mathbf{P}(X_t = r) \rightarrow \pi(r) = \frac{q}{p+q} = \frac{0.95}{0.15 + 0.95} = 0.8636.
 \end{aligned}$$

Now let us generalise the lessons learned from Example 119.

Proposition 55 For a finite Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with state space $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$, initial distribution

$$\mu_0 := (\mu_0(s_1), \mu_0(s_2), \dots, \mu_0(s_k)),$$

where $\mu_0(s_i) = \mathbf{P}(X_0 = s_i)$, and transition matrix

$$P := (P(s_i, s_j))_{(s_i, s_j) \in \mathbb{X}^2},$$

we have for any $t \in \mathbb{Z}_+$ that the distribution at time t given by:

$$\mu_t := (\mu_t(s_1), \mu_t(s_2), \dots, \mu_t(s_k)),$$

where $\mu_t(s_i) = \mathbf{P}(X_t = s_i)$, satisfies:

$$\mu_t = \mu_0 P^t. \quad (9.7)$$

Proof: We will prove this by induction on $\mathbb{Z}_+ := \{0, 1, 2, \dots\}$. First consider the case when $t = 0$. Since P^0 is the identity matrix I , we get the desired equality:

$$\mu_0 P^0 = \mu_0 I = \mu_0.$$

Next consider the case when $t = 1$. We get for each $j \in \{1, 2, \dots, k\}$, that

$$\begin{aligned} \mu_1(s_j) &= \mathbf{P}(X_1 = s_j) = \sum_{i=1}^k \mathbf{P}(X_1 = s_j, X_0 = s_i) \\ &= \sum_{i=1}^k \mathbf{P}(X_1 = s_j | X_0 = s_i) \mathbf{P}(X_0 = s_i) \\ &= \sum_{i=1}^k P(s_i, s_j) \mu_0(s_i) \\ &= (\mu_0 P)(s_j), \quad \text{the } j\text{-th entry of the row vector } (\mu_0 P). \end{aligned}$$

Hence, $\mu_1 = \mu_0 P$. Now, we will fix m and suppose that (9.7) holds for $t = m$ and prove that (9.7) also holds for $t = m + 1$. For each $j \in \{1, 2, \dots, k\}$, we get

$$\begin{aligned} \mu_{m+1}(s_j) &= \mathbf{P}(X_{m+1} = s_j) = \sum_{i=1}^k \mathbf{P}(X_{m+1} = s_j, X_m = s_i) \\ &= \sum_{i=1}^k \mathbf{P}(X_{m+1} = s_j | X_m = s_i) \mathbf{P}(X_m = s_i) \\ &= \sum_{i=1}^k P(s_i, s_j) \mu_m(s_i) \\ &= (\mu_m P)(s_j), \quad \text{the } j\text{-th entry of the row vector } (\mu_m P). \end{aligned}$$

Hence, $\mu_{m+1} = \mu_m P$. But $\mu_m = \mu_0 P^m$ by the induction hypothesis, and therefore:

$$\mu_{m+1} = \mu_m P = \mu_0 P^m P = \mu_0 P^{m+1}.$$

Thus by the principle of mathematical induction we have proved the proposition.

Thus, multiplying a row vector μ_0 by P^t on the right takes you from current distribution over the state space to the distribution in t steps of the chain.

Since we will be interested in Markov chains on $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$ with the same transition matrix P but different initial distributions, we introduce \mathbf{P}_μ and \mathbf{E}_μ for probabilities and expectations given that the initial distribution is μ , respectively. When the initial distribution is concentrated at a single initial state x given by:

$$\mathbf{1}_{\{x\}}(y) := \begin{cases} 1 & \text{if } y = x \\ 0 & \text{if } y \neq x \end{cases}$$

we represent it by e_x , the $1 \times k$ ortho-normal basis row vector with a 1 in the x -th entry and a 0 elsewhere. We simply write \mathbf{P}_x for $\mathbf{P}_{\mathbf{1}_{\{x\}}}$ or \mathbf{P}_{e_x} and \mathbf{E}_x for $\mathbf{E}_{\mathbf{1}_{\{x\}}}$ or \mathbf{E}_{e_x} . Thus, Proposition 55 along with our new notations means that:

$$\mathbf{P}_x(X_t = y) = (e_x P^t)(y) = P^t(x, y) .$$

In words, the probability of going to y from x in t steps is given by the (x, y) -th entry of P^t , the **t -step transition matrix**. We refer to the x -th row and the x -th column of P by $P(x, \cdot)$ and $P(\cdot, x)$, respectively.

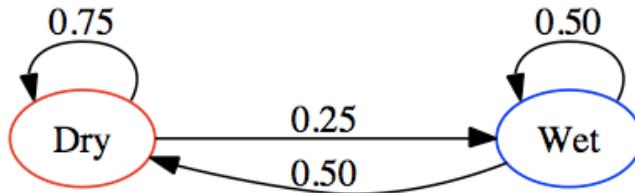
Let the function $f(x) : \mathbb{X} \rightarrow \mathbb{R}$ be represented by the column vector $f := (f(s_1), f(s_2), \dots, f(s_k)) \in \mathbb{R}^{k \times 1}$. Then the x -th entry of $P^t f$ is:

$$P^t f(x) = \sum_y P^t(x, y) f(y) = \sum_y f(y) \mathbf{P}_x(X_t = y) = \mathbf{E}_x(f(X_t)) .$$

This is the expected value of f under the distribution of states in t steps given that we start at state x . Thus multiplying a column vector f by P^t from the left takes you from a function on the state space to its expected value in t steps of the chain.

Example 120 (Dry-Wet Christchurch Weather) Consider a toy weather model for dry or wet days in Christchurch using a Markov chain with state space $\{d, w\}$. Let the transition diagram in Figure 9.3 give the transition matrix P for our dry-wet Markov chain. Using (9.7) we can find

Figure 9.3: Transition Diagram of Dry and Wet Days in Christchurch.



that the probability of being dry on the day after tomorrow is 0.625 given that it is wet today as follows:

```

>> P=[0.75 0.25; 0.5 0.5] % Transition Probability Matrix
P =
    0.7500    0.2500
    0.5000    0.5000
>> mu0=[0 1] % it is wet today gives the initial distribution
mu0 =
    0    1
>> mu0 * P^2 % the distribution in 2 days from today
ans =
    0.6250    0.3750
  
```

Suppose you sell \$100 of lemonade at a road-side stand on a hot day but only \$50 on a cold day. Then we can compute your expected sales tomorrow if today is dry as follows:

```
>> P=[0.75 0.25; 0.5 0.5] % Transition Probability Matrix
P =
    0.7500    0.2500
    0.5000    0.5000
>> f = [100; 50] % sales of lemonade in dollars on a dry and wet day
f =
    100
    50
>> P*f % expected sales tomorrow
ans =
    87.5000
    75.0000
>> mu0 = [1 0] % today is dry
mu0 =
    1     0
>> mu0*P*f % expected sales tomorrow if today is dry
ans =    87.5000
```

Exercise 121 (Freddy discovers a gold coin) Flippant Freddy of Example 119 found a gold coin at the bottom of the pond. Since this discovery he jumps around differently in the enchanted pond. He can be found now in one of three states: flipopia, rollophia and hydropia (when he dives into the pond). His state space is $\mathbb{X} = \{r, f, h\}$ now and his transition mechanism is as follows: If he rolls an odd number with his fair die in rollophia he will jump to flipopia but if he rolls an even number then he will stay in rollophia only if the outcome is 2 otherwise he will dive into hydropia. If the fair gold coin toss at the bottom of hydropia is Heads then Freddy will swim to flipopia otherwise he will remain in hydropia. Finally, if he is in flipopia he will remain there if the silver coin lands Heads otherwise he will jump to rollophia.

Make a Markov chain model of the new jumping mechanism adopted by Freddy. Draw the transition diagram, produce the transition matrix P and compute using MATLAB the probability that Freddy will be in hydropia after one, two, three, four and five jumps given that he starts in hydropia.

Exercise 122 Let $(X_t)_{t \in \mathbb{Z}_+}$ be a Markov chain with state space $\{a, b, c\}$, initial distribution $\mu_0 = (1/3, 1/3, 1/3)$ and transition matrix

$$P = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{matrix}.$$

For each t , define $Y_t = \mathbf{1}_{\{b,c\}}(X_t)$. Show that $(Y_t)_{t \in \mathbb{Z}_+}$ is not a Markov chain.

Exercise 123 Let $(X_t)_{t \in \mathbb{Z}_+}$ be a (homogeneous) Markov chain on $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$ with transition matrix P and initial distribution μ_0 . For a given $m \in \mathbb{N}$, let $(Y_t)_{t \in \mathbb{Z}_+}$ be a stochastic sequence with $Y_t = X_{mt}$. Show that $(Y_t)_{t \in \mathbb{Z}_+}$ is a Markov chain with transition matrix P^m . This establishes that Markov chains that are sampled at regular time steps are also Markov chains.

Until now our Markov chains have been **homogeneous** in time according to Definition 54, i.e., the transition matrix P does not change with time. We define inhomogeneous Markov chains that allow their transition matrices to possibly change with time. Such Markov chains are more realistic as models in some situations and more flexible as algorithms in the sequel.

Definition 56 (Inhomogeneous finite Markov chain) Let P_1, P_2, \dots be a sequence of $k \times k$ stochastic matrices satisfying the conditions in Equation 9.2. Then, the stochastic sequence $(X_t)_{t \in \mathbb{Z}_+} := (X_0, X_1, \dots)$ with finite state space $\mathbb{X} := \{s_1, s_2, \dots, s_k\}$ is called an inhomogeneous Markov chain with transition matrices P_1, P_2, \dots , if for all pairs of states $(x, y) \in \mathbb{X} \times \mathbb{X}$, all integers $t \geq 1$, and all probable historical events $H_{t-1} := \bigcap_{n=0}^{t-1} \{X_n = x_n\}$ with $\mathbf{P}(H_{t-1} \cap \{X_t = x\}) > 0$, the following **Markov property** is satisfied:

$$\mathbf{P}(X_{t+1} = y | H_{t-1} \cap \{X_t = x\}) = \mathbf{P}(X_{t+1} = y | X_t = x) =: P_{t+1}(x, y) . \quad (9.8)$$

Proposition 57 For a finite inhomogeneous Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with state space $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$, initial distribution

$$\mu_0 := (\mu_0(s_1), \mu_0(s_2), \dots, \mu_0(s_k)) ,$$

where $\mu_0(s_i) = \mathbf{P}(X_0 = s_i)$, and transition matrices

$$(P_1, P_2, \dots) , \quad P_t := (P_t(s_i, s_j))_{(s_i, s_j) \in \mathbb{X} \times \mathbb{X}} , \quad t \in \{1, 2, \dots\}$$

we have for any $t \in \mathbb{Z}_+$ that the distribution at time t given by:

$$\mu_t := (\mu_t(s_1), \mu_t(s_2), \dots, \mu_t(s_k)) ,$$

where $\mu_t(s_i) = \mathbf{P}(X_t = s_i)$, satisfies:

$$\mu_t = \mu_0 P_1 P_2 \cdots P_t . \quad (9.9)$$

Proof: Left as Exercise 124.

Exercise 124 Prove Proposition 57 using induction as done for Proposition 55.

Example 125 (a more sophisticated dry-wet chain) Let us make a more sophisticated version of the dry-wet chain of Example 120 with state space $\{d, w\}$. In order to take some seasonality into account in our weather model for dry and wet days in Christchurch, let us have two transition matrices for hot and cold days:

$$P_{\text{hot}} = \begin{matrix} d & w \\ \begin{pmatrix} 0.95 & 0.05 \\ 0.75 & 0.25 \end{pmatrix} \end{matrix}, \quad P_{\text{cold}} = \begin{matrix} d & w \\ \begin{pmatrix} 0.65 & 0.35 \\ 0.45 & 0.55 \end{pmatrix} \end{matrix} .$$

We say that a day is hot if its maximum temperature is more than 20° Celsius, otherwise it is cold. We use the transition matrix for today to obtain the state probabilities for tomorrow. If today is dry and hot and tomorrow is supposed to be cold then what is the probability that the day after tomorrow will be wet? We can use (9.9) to obtain the answer as 0.36:

```
>> Phot = [0.95 0.05; 0.75 0.25] % Transition Probability Matrix for hot day
Phot =
    0.9500    0.0500
    0.7500    0.2500
>> Pcold = [0.65 0.35; 0.45 0.55] % Transition Probability Matrix for cold day
Pcold =
    0.6500    0.3500
    0.4500    0.5500
>> mu0 = [1 0] % today is dry
mu0 =      1      0
```

```

>> mu1 = mu0 * Phot % distribution for tomorrow since today is hot
mu1 =
    0.9500    0.0500
>> mu2 = mu1 * Pcold % distribution for day after tomorrow since tomorrow is supposed to be cold
mu2 =
    0.6400    0.3600
>> mu2 = mu0 * Phot * Pcold % we can also get the distribution for day after tomorrow directly
mu2 =
    0.6400    0.3600

```

Exercise 126 For the Markov chain in Example 125 compute the probability that the day after tomorrow is wet if today is dry and hot but tomorrow is supposed to be cold.

9.2 Random Mapping Representation and Simulation

In order to simulate (x_0, x_1, \dots, x_n) , a sequential realisation or sequence of states visited by a Markov chain, say the sequence of lily pads that Flippant Freddy visits on his jumps, we need a random mapping representation of a Markov chain and its computer implementation.

Definition 58 (Random mapping representation (RMR)) A **random mapping representation (RMR)** of a transition matrix $P := (P(x, y))_{(x,y) \in \mathbb{X}^2}$ is a function

$$\rho(x, w) : \mathbb{X} \times \mathbb{W} \rightarrow \mathbb{X} , \quad (9.10)$$

along with the auxiliary \mathbb{W} -valued random variable W , satisfying

$$\mathbf{P}(\{\rho(x, W) = y\}) = P(x, y), \quad \text{for each } (x, y) \in \mathbb{X}^2 . \quad (9.11)$$

Proposition 59 (Markov chain from RMR) If $W_1, W_2, \dots \stackrel{IID}{\sim} W$, the auxiliary RV in a RMR of a transition matrix $P := (P(x, y))_{(x,y) \in \mathbb{X}^2}$, and $X_0 \sim \mu_0$, then $(X_t)_{t \in \mathbb{Z}_+}$ defined by

$$X_t = \rho(X_{t-1}, W_t) , \quad \text{for all } t \geq 1$$

is a Markov chain with transition matrix P and initial distribution μ_0 on state space \mathbb{X} .

Proof: Left as Exercise 127.

Exercise 127 Do the proof of Proposition 59 by using the necessary Definitions.

Example 128 (An RMR for Flippant Freddy) Reconsider the Markov chain of Flippant Freddy with fair dice and fair coin on state space $\mathbb{X} = \{r, f\}$ with transition matrix

$$P = \begin{matrix} & r & f \\ r & \left(\begin{matrix} 1/2 & 1/2 \end{matrix} \right) \\ f & \left(\begin{matrix} 1/2 & 1/2 \end{matrix} \right) \end{matrix} .$$

Let the auxiliary RV W have sample space $\mathbb{W} = \{0, 1\}$. Then an RMR $\rho : \mathbb{X} \times \mathbb{W} \rightarrow \mathbb{X}$ for this P is given by

$$\rho(x, w) : \{r, f\} \times \{0, 1\} \rightarrow \{r, f\}, \quad \rho(r, 0) = r, \quad \rho(r, 1) = f, \quad \rho(f, 0) = f, \quad \rho(f, 1) = r,$$

with $\mathbf{P}(W = 0) = \mathbf{P}(W = 1) = 1/2$. Now let us check that our ρ and W satisfy Equation 9.11:

$$\begin{aligned} r & \quad f \\ r \begin{pmatrix} \mathbf{P}(\{\rho(r, W) = r\}) & \mathbf{P}(\{\rho(r, W) = f\}) \\ \mathbf{P}(\{\rho(f, W) = r\}) & \mathbf{P}(\{\rho(f, W) = f\}) \end{pmatrix} &= r \begin{pmatrix} \mathbf{P}(W = 0) & \mathbf{P}(W = 1) \\ \mathbf{P}(W = 1) & \mathbf{P}(W = 0) \end{pmatrix} \\ &= r \begin{pmatrix} r & f \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} = P . \end{aligned}$$

Thus, by Proposition 59 we can obtain Freddy's Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ by initialising $X_0 \sim \mu_0 = (1, 0)$, i.e., setting $X_0 = r$ since Freddy starts at r , and defining

$$X_t = \rho(X_{t-1}, W_t), \quad \text{for all } t \geq 1, \text{ where, } W_1, W_2, \dots \stackrel{IID}{\sim} \text{Bernoulli}(1/2) \text{ RV} .$$

In other words, we can simulate a sequence of states or lily pads visited by Freddy by merely doing independent Bernoulli(1/2) trials and use the mapping ρ . A MATLAB implementation of this RMR ρ as a MATLAB function is:

```
RMR10fFairFreddy.m
function y = RMR10fFairFreddy(x,w)
% Random Mapping Representation Number 1 of P=[1/2 1/2; 1/2 12/]
% input: character x as 'r' or 'f' and w as 0 or 1
% output: character y as 'r' or 'f'
if (x =='r')
    if (w==0)
        y = 'r';
    elseif (w==1)
        y = 'f';
    else
        y = Nan;
        print "when x = 'r' w is neither 0 nor 1!";
    end
elseif (x =='f')
    if (w==0)
        y = 'f';
    elseif (w==1)
        y = 'r';
    else
        y = Nan;
        print "when x='f' w is neither 0 nor 1!";
    end
else
    y = Nan;
    print "x is neither 'r' nor 'f'";
end
```

We can simulate one realisation of the first two states (x_0, x_1) visited by (X_0, X_1) as follows:

```
>> % set PRNG to be twister with seed 19731511
>> RandStream.setDefaultStream(RandStream('mt19937ar','seed',19731511));
>> x0 = 'r' % set x_0 = 'r'
x0 = r
>> w1 = floor( rand + 0.5 ) % a Bernoulli(0.5) trial
w1 =
0
>> x1 = RMR10fFairFreddy(x0,w1) % x_1 = rho(x_0,w1) is the state at time t=1
x1 = r
```

We can simulate one realisation of the first 10 states (x_0, x_1, \dots, x_9) visited by (X_0, X_1, \dots, X_9) using a for loop as follows:

```
>> % set PRNG to be twister with seed 19731511
>> RandStream.setDefaultStream(RandStream('mt19937ar','seed',19731511));
>> x0 = 'r' % set x_0 = 'r'
x0 =
>> xt = x0; % current state x_t is x_0
>> Visited = x0; % initialise the variable Visited to hold the visited states
>> for t = 1:9 % start a for loop for t = 1,2,...,9
xt = RMR10fFairFreddy(xt, floor(rand+0.5) ); % update the current state at t
Visited = strcat(Visited,',',xt); % store the visited state in string Visited
end
>> Visited % disclose the string of visited state separated by commas
Visited = r,r,f,f,r,r,f,f,f,r
```

If we change the seed to some other number and repeat the code above, we will get another realisation of visits (x_0, x_1, \dots, x_9) of (X_0, X_1, \dots, X_9) . However, there are many distinct RMRs of the same transition matrix P . For example, we can define a new RMR ρ' from our first RMR ρ for P by $\rho'(x, w) = \rho(x, 1 - w)$. The reader should check that ρ' also satisfies Equation 9.11 with $W \sim \text{Bernoulli}(1/2)$. But note that even for the same seed and the same PRNG the sequence of states (x_0, x_1, \dots, x_9) visited by (X_0, X_1, \dots, X_9) under the new RMR ρ' is different from that of the original RMR ρ :

```
>> % set PRNG to be twister with seed 19731511
>> RandStream.setDefaultStream(RandStream('mt19937ar','seed',19731511));
>> x0 = 'r' % set x_0 = 'r'
x0 =
>> xt = x0; % current state x_t is x_0
>> Visited = x0; % initialise the variable Visited to hold the visited states
>> for t = 1:9 % start a for loop for t = 1,2,...,9
xt = RMR10fFairFreddy(xt, 1-floor(rand+0.5) ); % update the current state at t with new RMR rho'
Visited = strcat(Visited,',',xt); % store the visited state in string Visited
end
>> Visited % disclose the string of visited state separated by commas under new RMR rho'
Visited = r,f,f,r,r,f,f,r,f,f
```

Proposition 60 (Existence and non-uniqueness of RMR) Every transition matrix P on a finite state space \mathbb{X} has a random mapping representation (RMR) that is not necessarily unique.

Proof: Let $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$ be sequentially accessible by $\psi(i) = s_i : \{1, 2, \dots, k\} \rightarrow \mathbb{X}$. We will prove the proposition constructively via the inversion sampler for \mathbb{X} -valued family of ψ -transformed de Moivre RVs. Let the auxiliary RV W be Uniform(0, 1) with $\mathbb{W} = [0, 1]$ and let $\rho(x, w) : \mathbb{X} \times \mathbb{W} \rightarrow \mathbb{X}$ be given by $F^{[-1]}(u; \theta_1, \theta_2, \dots, \theta_k)$ of Equation 6.16, the inverse DF of the de Moivre($\theta_1, \theta_2, \dots, \theta_k$) RV, as follows:

$$\rho(x, w) = \psi \left(F^{[-1]}(w; P(x, s_1), P(x, s_2), \dots, P(x, s_k)) \right), \quad \text{for each } x \in \mathbb{X} .$$

Then, by construction, this ρ is indeed an RMR of P since

$$\mathbf{P}(\{\rho(x, W) = y\}) = P(x, y) \quad \text{for each } (x, y) \in \mathbb{X}^2 .$$

Non-uniqueness is established by constructing another RMR for P as $\rho'(x, w) = \rho(x, 1 - w)$.

Labwork 129 (Markov chain from $\{\text{de Moivre}(P(x,.))\}_{x \in \mathbb{X}}$ RVs) Let us implement a function that will take a transition matrix P as input and produce a sequence of n states $(x_0, x_1, \dots, x_{n-1})$ visited by the corresponding Markov chain (X_0, X_1, \dots, X_n) using the function in the following M-file.

```
MCSimBydeMoivre.m
function VisitedStateIdxs = MCSimBydeMoivre(idx0, P, n)
% input: idx0 = index of initial state x_0, psi(idx0) = x_0
%         P = transition probability matrix (has to be stochastic matrix)
%         n = number of time steps to simulate, n >= 0
% output: VisitedStateIdxs = idx0, idx1, ..., idxn
VisitedStateIdxs = zeros(1,n);
VisitedStateIdxs(1, 0+1) = idx0; % initial state index is the input idx0
for i=1:n-1
    CurrentState = VisitedStateIdxs(1, i); % current state
    Thetas = P(CurrentState,:);
    VisitedStateIdxs(1, i+1) = SimdeMoivreOnce(rand,Thetas); % next state
end
end
```

Simulation 130 (Another simulation of Freddy's jumps) Let us simulate a sequence of 10 jumps of Flippant Freddy with fair dice and coin by using the function `MCSimBydeMoivre` defined in Labwork 129 as follows:

```
>> % set PRNG to be twister with seed 19731511
>> RandStream.setDefaultStream(RandStream('mt19937ar','seed',19731511));
>> MCSimBydeMoivre(1,[0.5 0.5; 0.5 0.5], 10)
ans =
     1     1     2     1     2     1     2     1     1     2
```

Here we need to further transform the output by $\psi : \{1, 2\} \rightarrow \{r, f\}$ with $\psi(1) = r$ and $\psi(2) = f$.

Labwork 131 (Markov chain from $\{\text{de Moivre}(P(x,.))\}_{x \in \mathbb{X}}$ RVs by Recursion) Let us implement a recursive function that will take a transition matrix P as input and produce a sequence of n states $(x_0, x_1, \dots, x_{n-1})$ visited by the corresponding Markov chain (X_0, X_1, \dots, X_n) using the function in the following M-file.

```
MCSimBydeMoivreRecurse.m
function VisitedStateIdxs = MCSimBydeMoivreRecurse(VisitedStateIdxs, P, n)
% input: VisitedStateIdxs = array of indexes of states visited so far
%         P = transition probability matrix (has to be stochastic matrix)
%         n = number of time steps to simulate, n >= 0
% output: VisitedStateIdxs = idx0, idx1, ..., idxn
i = length(VisitedStateIdxs);
if i < n
    CurrentState = VisitedStateIdxs(1, i); % current state
    Thetas = P(CurrentState,:);
    % recursion
    VisitedStateIdxs= MCSimBydeMoivreRecurse([VisitedStateIdxs SimdeMoivreOnce(rand,Thetas)],P,n); % next state
end
end
```

Now, let us compare this recursive function to the function `MCSimBydeMoivre` defined in Labwork 129 as follows:

```
CompareMCSimBydeMoivreMethods.m
format compact
P=[1/3 2/3;1/4 4/5]
```

```

initial = 2
visited = [initial];
n = 12;

s = RandStream('mt19937ar','Seed', 5489);
RandStream.setDefaultStream(s) % reset the PRNG to default state Mersenne Twister with seed=5489

VisitByMethod1 = MCSimBydeMoivre(initial, P, n)

s = RandStream('mt19937ar','Seed', 5489);
RandStream.setDefaultStream(s) % reset the PRNG to default state Mersenne Twister with seed=5489

VisitByMethod2 = MCSimBydeMoivreRecurse(visited, P, n)

```

```

>> CompareMCSimBydeMoivreMethods
P =
    0.3333    0.6667
    0.2500    0.8000
initial =
    2
VisitByMethod1 =
    2    2    2    1    2    2    1    1    2    2    2    1
VisitByMethod2 =
    2    2    2    1    2    2    1    1    2    2    2    1

```

Therefore, both methods produce the same output. The recursive version of the function is more versatile and useful in the sequel.

Simulation 132 Using the function `MCSimBydeMoivre` of Labwork 129 simulate twenty states visited by the Markov chain in Exercise 121.

Simulation 133 (Drunkard's walk around the block) Consider the Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ on $\mathbb{X} = \{0, 1, 2, 3\}$ with initial distribution $\mathbf{1}_{\{3\}}(x)$ and transition matrix

$$P = \begin{pmatrix} & 0 & 1 & 2 & 3 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 1 & 1/2 & 0 & 1/2 & 0 \\ 2 & 0 & 1/2 & 0 & 1/2 \\ 3 & 1/2 & 0 & 1/2 & 0 \end{pmatrix}.$$

Draw the transition diagram for this Markov chain. Do you see why this chain can be called the “drunkard's walk around the block”? Using the function `MCSimBydeMoivre` of Labwork 129 simulate a sequence of ten states visited by the drunkard (don't forget to subtract 1 from the output of `MCSimBydeMoivre` since $\psi(i) = i - 1$ here).

There are many distinct and interesting RMRs of any given transition matrix P beyond that constructed in the proof above. Good RMRs will typically simplify the simulation of a Markov chain. Let us consider examples of Markov chains that can be simulated by simpler methods.

Example 134 (Jukes & Cantor Model of DNA mutation) The “blueprint” of organisms on earth are typically given by a long sequence of deoxyribonucleic acid or DNA. A DNA sequence of length n can be thought of as a string made up of n alphabets from the set of four nucleotides $\{a, c, g, t\}$. For example a DNA sequence of length 3 is *agg* and another is *act*. When an organism

goes through time to “stay alive” it has to copy its DNA. This copying process is not perfect and mistakes or mutations are made. We can look at a particular position of a DNA sequence and keep track of its mutations using a simple Markov chain due to Jukes and Cantor [Jukes TH and Cantor CR (1969) Evolution of protein molecules. In Munro HN, editor, Mammalian Protein Metabolism, pp. 21-132, Academic Press, New York.] with the following transition probability matrix:

$$P = \begin{pmatrix} & a & c & g & t \\ a & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ c & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ g & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ t & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix}.$$

Suppose you initially observe the particular position of a DNA sequence at state c and want to simulate a sequence of states visited due to mutation under this Markov chain model. We can achieve this by improvising the inversion sampler for the equi-probable de Moivre($1/3, 1/3, 1/3$) RV (Algorithm 5) in the following RMR:

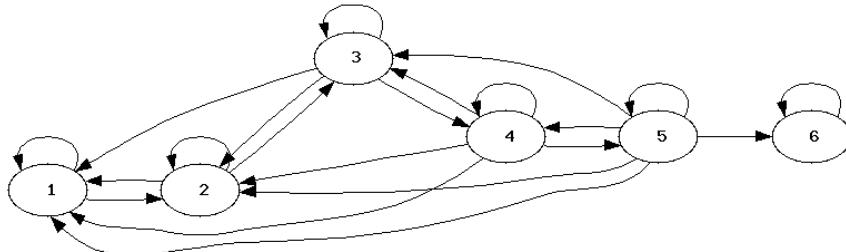
$$\rho(x, U) : \{a, c, g, t\} \times [0, 1] \rightarrow \{a, c, g, t\}, \quad \rho(x, U) = \psi_x(\lceil 3U \rceil), \quad U \sim \text{Uniform}(0, 1),$$

with any fixed bijection $\psi_x(i) : \{1, 2, 3\} \rightarrow \{a, c, g, t\} \setminus \{x\}$ for each $x \in \{a, c, g, t\}$. Then we can produce a sequence of visited states as follows:

$$X_0 \leftarrow c, \quad X_i \leftarrow \rho(X_{i-1}, U_i), \quad i = 1, 2, \dots.$$

Example 135 (Six Lounges) Suppose there are six lounges with doors that allow you to go only in one direction. These lounges are labelled by 1, 2, 3, 4, 5 and 6 and form our state space \mathbb{X} with one-way-doors as shown in Figure 9.4. Every hour an alarm rings and it can be heard in all six

Figure 9.4: Transition diagram over six lounges (without edge probabilities).



lounges. In each lounge $i \in \{1, 2, 3, 4, 5\}$ there is a fair i -sided polyhedral cylinder whose i faces are marked with lounge numbers $1, 2, \dots, i$ but in lounge 6 there is a hexagonal cylinder with all six faces marked by 6. Suppose you start from lounge number 1. When the hourly alarm rings you toss the polyhedral cylinder in the current lounge over the floor. When the cylinder comes to rest, you note the number on the face that touches the floor and go to the lounge labelled by this number. This scheme of lounge hopping can be formalised as a Markov chain starting at lounge

number 1 and evolving according to the transition matrix P :

$$P = \begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 2 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 3 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 4 & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & 0 \\ 5 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ 6 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The inversion samplers for the family of equi-probable $\{\text{de Moivre}(1/i, 1/i, \dots, 1/i)\}_{i \in \{1, 2, \dots, 5\}}$ RVs (Algorithm 5) and the Point Mass(6) RV (Simulation 76) can be combined in the random mapping representation:

$$\rho(i, U) : \mathbb{X} \times [0, 1] \rightarrow \mathbb{X}, \quad \rho(i, U) = \lceil iU \rceil \mathbf{1}_{\{1, 2, 3, 4, 5\}}(i) + 6\mathbf{1}_{\{6\}}(i), \quad U \sim \text{Uniform}(0, 1),$$

in order to simulate a sequence of states from this markov chain as follows:

$$X_0 \leftarrow 1, \quad X_i \leftarrow \rho(X_{i-1}, U_i), \quad i = 1, 2, \dots. \quad (9.12)$$

Simulation 136 (Trapped in lounge 6) Implement the Algorithm described in Equation 9.12 in a MATLAB program to simulate the first ten states visited by the Markov chain in Example 135. Recall the “Hotel California” character of lounge 6 – *you can check out anytime you like, but you can never leave!* Repeat this simulation 1000 times and find the fraction of times your are not trapped in lounge 6 by the tenth time step.

Exercise 137 (Drunkard’s walk around a polygonal block with k corners) Can you think of another way to simulate the “drunkard’s walk around a polygonal block with k corners” labelled by $0, 1, \dots, k - 1$ that is more efficient than using the `MCSimBydeMoivre` function which relies on the `SimdeMoivreOnce` function that implements Algorithm 6 with an average-case efficiency that is linear in k ?

Hint: think of the drunkard tossing a fair coin to make his decision of where to go next from each corner and arithmetic mod k .

9.3 Irreducibility and Aperiodicity

The utility of our mathematical constructions with Markov chains depends on a delicate balance between generality and specificity. We introduce two specific conditions called irreducibility and aperiodicity that make Markov chains more useful to model real-word phenomena.

Definition 61 (Communication between states) Let $(X_t)_{t \in \mathbb{Z}_+}$ be a homogeneous Markov chain with transition matrix P on state space $\mathbb{X} := \{s_1, s_2, \dots, s_k\}$. We say that a state s_i **communicates** with a state s_j and write $s_i \rightarrow s_j$ or $s_j \leftarrow s_i$ if there exists an $\eta(s_i, s_j) \in \mathbb{N}$ such that:

$$\mathbf{P}\left(X_{t+\eta(s_i, s_j)} = s_j | X_t = s_i\right) = P^{\eta(s_i, s_j)}(s_i, s_j) > 0.$$

In words, s_i communicates with s_j if you can eventually reach s_j from s_i . If $P^\eta(s_i, s_j) = 0$ for every $\eta \in \mathbb{N}$ then we say that s_i **does not communicate** with s_j and write $s_i \not\rightarrow s_j$ or $s_j \not\leftarrow s_i$.

We say that two states s_i and s_j **intercommunicate** and write $s_i \leftrightarrow s_j$ if $s_i \rightarrow s_j$ and $s_j \rightarrow s_i$. In words, two states intercommunicate if you can eventually reach one from another and vice versa. When s_i and s_j do not intercommunicate we write $s_i \not\leftrightarrow s_j$.

Definition 62 (Irreducible) A homogeneous Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with transition matrix P on state space $\mathbb{X} := \{s_1, s_2, \dots, s_k\}$ is said to be **irreducible** if $s_i \leftrightarrow s_j$ for each $(s_i, s_j) \in \mathbb{X}^2$. Otherwise the chain is said to be **reducible**.

We have already seen examples of reducible and irreducible Markov chains. For example, Flippant Freddy's family of Markov chains with the (p, q) -parametric family of transition matrices, $\{P_{(p,q)} : (p, q) \in [0, 1]^2\}$, where each $P_{(p,q)}$ is given by Equation 9.3. If $(p, q) \in (0, 1)^2$, then the corresponding Markov chain is irreducible because we can go from rollovia to flippopia or vice versa in just one step with a positive probability. Thus, the Markov chains with transition matrices in $\{P_{(p,q)} : (p, q) \in (0, 1)^2\}$ are irreducible. But if p or q take probability values at the boundary of $[0, 1]$, i.e., $p \in \{0, 1\}$ or $q \in \{0, 1\}$ then we have to be more careful because we may never get from at least one state to the other and the corresponding Markov chains may be reducible. For instance, if $p = 0$ or $q = 0$ then we will be stuck in either rollovia or flippopia, respectively. However, if $p = 1$ and $q \neq 0$ or $q = 1$ and $p \neq 0$ then we can get from each state to the other. Therefore, only the transition matrices in $\{P_{(p,q)} : p \in \{0\} \text{ or } q \in \{0\}\}$ are reducible.

The simplest way to verify whether a Markov chain is irreducible is by looking at its transition diagram (without the positive edge probabilities) and checking that from each state there is a sequence of arrows leading to any other state. For instance, from the transition diagram in Figure 9.4 of the lounge-hopping Markov chain of Example 135, it is clear that if you start at state 6 you cannot find any arrow going to any other state. Therefore, the chain is reducible since $6 \not\rightarrow i$ for any $i \in \{1, 2, 3, 4, 5\}$.

Exercise 138 Revisit all the Markov chains we have considered up to now and determine whether they are reducible or irreducible by checking that from each state there is a sequence of arrows leading to any other state in their transition graphs.

Definition 63 (Return times and period) Let $\mathbb{T}(x) := \{t \in \mathbb{N} : P^t(x, x) > 0\}$ be the set of **possible return times** to the starting state x . The **period** of state x is defined to be $\gcd(\mathbb{T}(x))$, the greatest common divisor of $\mathbb{T}(x)$. When the period of a state x is 1, i.e., $\gcd(\mathbb{T}(x)) = 1$, then x is said to be an **aperiodic state**.

Proposition 64 If the Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with transition matrix P on state space \mathbb{X} is irreducible then $\gcd(\mathbb{T}(x)) = \gcd(\mathbb{T}(y))$ for any $(x, y) \in \mathbb{X}^2$.

Proof: Fix any pair of states $(x, y) \in \mathbb{X}^2$. Since, P is irreducible, $x \leftrightarrow y$ and therefore there exists natural numbers $\eta(x, y)$ and $\eta(y, x)$ such that $P^{\eta(x,y)}(x, y) > 0$ and $P^{\eta(y,x)}(y, x) > 0$. Let $\eta' = \eta(x, y) + \eta(y, x)$ and observe that $\eta' \in \mathbb{T}(x) \cap \mathbb{T}(y)$, $\mathbb{T}(x) \subset \mathbb{T}(y) - \eta' := \{t - \eta' : t \in \mathbb{T}(y)\}$ and $\gcd(\mathbb{T}(y))$ divides all elements in $\mathbb{T}(x)$. Thus, $\gcd(\mathbb{T}(y)) \leq \gcd(\mathbb{T}(x))$. By a similar argument we can also conclude that $\gcd(\mathbb{T}(x)) \leq \gcd(\mathbb{T}(y))$. Therefore $\gcd(\mathbb{T}(x)) = \gcd(\mathbb{T}(y))$.

Definition 65 (Aperiodic) A Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with transition matrix P on state space \mathbb{X} is said to be aperiodic if all of its states are aperiodic, i.e., $\gcd(\mathbb{T}(x)) = 1$ for every $x \in \mathbb{X}$. If a chain is not aperiodic, we call it **periodic**.

We have already seen example of irreducible Markov chains that were either periodic or aperiodic. For instance, Freddy's Markov chain with $(p, q) \in (0, 1)^2$ is aperiodic since the period of either of its two states is given by $\gcd(\{1, 2, 3, \dots\}) = 1$. However, the Markov chain model for a drunkard's walk around a block over the state space $\{0, 1, 2, 3\}$ (Simulation 133) is periodic because you can only return to the starting state in an even number of time steps and

$$\gcd(\mathbb{T}(0)) = \gcd(\mathbb{T}(1)) = \gcd(\mathbb{T}(2)) = \gcd(\mathbb{T}(3)) = \gcd(\{2, 4, 6, \dots\}) = 2 \neq 1 .$$

Exercise 139 Show that the Markov chain corresponding to a drunkard's walk around a polygonal block with k corners is irreducible for any integer $k > 1$. Show that it is aperiodic only when k is odd and has period 2 when k is even.

Proposition 66 Let $A = \{a_1, a_2, \dots\} \subset \mathbb{N}$ that satisfies the following two conditions:

1. A is a **nonlattice**, meaning that $\gcd(A) = 1$ and
2. A is closed under addition, meaning that if $(a, a') \in A^2$ then $a + a' \in A$.

Then there exists a positive integer $\eta < \infty$ such that $n \in A$ for all $n \geq \eta$.

Proof: See Proofs of Lemma 1.1, Lemma 1.2 and Theorem 1.1 in Appendix of *Pierre Brémaud, Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues, Springer, 1999*.

Proposition 67 If the Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with transition matrix P on state space \mathbb{X} is irreducible and aperiodic then there is an integer τ such that $P^t(x, x) > 0$ for all $t \geq \tau$ and all $x \in \mathbb{X}$.

Proof: TBD

Proposition 68 If the Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with transition matrix P on state space \mathbb{X} is irreducible and aperiodic then there is an integer τ such that $P^t(x, y) > 0$ for all $t \geq \tau$ and all $(x, y) \in \mathbb{X}^2$.

Proof: TBD

Exercise 140 (King's random walk on a chessboard) Consider the squares in the chessboard as the state space $\mathbb{X} = \{0, 1, 2, \dots, 7\}^2$ with a randomly walking black king, i.e., for each move from current state $(u, v) \in \mathbb{X}$ the king chooses one of his $k(u, v)$ possible moves uniformly at random. Is the Markov chain corresponding to the randomly walking black king on the chessboard irreducible and/or aperiodic?

Exercise 141 (King's random walk on a chesstorus) We can obtain a chesstorus from a pliable chessboard by identifying the eastern edge with the western edge (roll the chessboard into a cylinder) and then identifying the northern edge with the southern edge (gluing the top and bottom end of the cylinder together by turning into a doughnut or torus). Consider the squares in the chesstorus as the state space $\mathbb{X} = \{0, 1, 2, \dots, 7\}^2$ with a randomly walking black king, i.e., for each move from current state $(x, y) \in \mathbb{X}$ the king chooses one of his 8 possible moves uniformly at random according to the scheme: $X_t \leftarrow X_{t-1} + W_t$, where W_t is independent and identically distributed as follows:

$$\mathbf{P}(W_t = w) = \begin{cases} \frac{1}{8} & \text{if } w \in \{(1, 1), (1, 0), (1, -1), (0, -1), (-1, -1), (-1, 0), (-1, 1), (0, 1)\}, \\ 0 & \text{otherwise.} \end{cases}$$

Is the Markov chain corresponding to the randomly walking black king on the chesstorus irreducible and/or aperiodic? Write a MATLAB script to simulate a sequence of n states visited by the king if he started from $(0, 0)$ on the chesstorus.

9.4 Stationarity

We are interested in statements about a Markov chain that has been running for a long time. For any nontrivial Markov chain (X_0, X_1, \dots) the value of X_t will keep fluctuating in the state space \mathbb{X} as $t \rightarrow \infty$ and we cannot hope for convergence to a fixed point state $x^* \in \mathbb{X}$ or to a k -cycle of states $\{x_1, x_2, \dots, x_k\} \subset \mathbb{X}$. However, we can look one level up into the space of probability distributions over \mathbb{X} that give the probability of the Markov chain visiting each state $x \in \mathbb{X}$ at time t , and hope that the distribution of X_t over \mathbb{X} settles down as $t \rightarrow \infty$. The Markov chain convergence theorem indeed states that the distribution of X_t over \mathbb{X} settles down as $t \rightarrow \infty$, provided the Markov chain is irreducible and aperiodic.

Definition 69 (Stationary distribution) Let $(X_t)_{t \in \mathbb{Z}_+}$ be a Markov chain with state space $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$ and transition matrix $P = (P(x, y))_{(x,y) \in \mathbb{X}^2}$. A row vector

$$\pi = (\pi(s_1), \pi(s_2), \dots, \pi(s_k)) \in \mathbb{R}^{1 \times k}$$

is said to be a **stationary distribution** for the Markov chain, if it satisfies the conditions of being:

1. *a probability distribution:* $\pi(x) \geq 0$ for each $x \in \mathbb{X}$ and $\sum_{x \in \mathbb{X}} \pi(x) = 1$, and
2. *a fixed point:* $\pi P = \pi$, i.e., $\sum_{x \in \mathbb{X}} \pi(x)P(x, y) = \pi(y)$ for each $y \in \mathbb{X}$.

Definition 70 (Hitting times) If a Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with state space $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$ and transition matrix $P = (P(x, y))_{(x,y) \in \mathbb{X}^2}$ starts at state x , then we can define the **hitting time**

$$T(x, y) = \min\{t \geq 1 : X_t = y\} .$$

and let $T(x, y) = \min\{\} = \infty$ if the Markov chain never visits y after having started from x . Let the **mean hitting time**

$$\tau(x, y) := \mathbf{E}(T(x, y)),$$

be the expected time taken to reach y after having started at x . Note that $\tau(x, x)$ is the **mean return time** to state x .

Proposition 71 (Hitting times of irreducible aperiodic Markov chains) If $(X_t)_{t \in \mathbb{Z}_+}$ is an irreducible aperiodic Markov chain with state space $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$, transition matrix $P = (P(x, y))_{(x,y) \in \mathbb{X}^2}$ then for any pair of states $(x, y) \in \mathbb{X}^2$,

$$\mathbf{P}(T(x, y) < \infty) = 1 ,$$

and the mean hitting time is finite, i.e.,

$$\tau(x, y) < \infty .$$

Proposition 72 (Existence of Stationary distribution) For any irreducible and aperiodic Markov chain there exists at least one stationary distribution.

Proof: TBD

Definition 73 (Total variation distance) If $\nu_1 := (\nu_1(x))_{x \in \mathbb{X}}$ and $\nu_2 := (\nu_2(x))_{x \in \mathbb{X}}$ are elements of $\mathcal{P}(\mathbb{X})$, the set of all probability distributions on $\mathbb{X} := \{s_1, s_2, \dots, s_k\}$, then we define the **total variation distance** between ν_1 and ν_2 as

$$d_{TV}(\nu_1, \nu_2) := \frac{1}{2} \sum_{x \in \mathbb{X}} \text{abs}(\nu_1(x) - \nu_2(x)), \quad d_{TV} : \mathcal{P}(\mathbb{X}) \times \mathcal{P}(\mathbb{X}) \rightarrow [0, 1] . \quad (9.13)$$

If ν_1, ν_2, \dots and ν are probability distributions on \mathbb{X} , then we say that ν_t **converges in total variation** to ν as $n \rightarrow \infty$ and write $\nu_t \xrightarrow{TV} \nu$, if

$$\lim_{t \rightarrow \infty} d_{TV}(\nu_t, \nu) = 0 .$$

Observe that if $d_{TV}(\nu_1, \nu_2) = 0$ then $\nu_1 = \nu_2$. The constant $1/2$ in Equation 9.13 ensures that the range of d_{TV} is in $[0, 1]$. If $d_{TV}(\nu_1, \nu_2) = 1$ then ν_1 and ν_2 have disjoint supports, i.e., we can partition \mathbb{X} into \mathbb{X}_1 and \mathbb{X}_2 , i.e., $\mathbb{X} = \mathbb{X}_1 \cup \mathbb{X}_2$ and $\mathbb{X}_1 \cap \mathbb{X}_2 = \emptyset$, such that $\sum_{x \in \mathbb{X}_1} \nu_1(x) = 1$ and $\sum_{x \in \mathbb{X}_2} \nu_2(x) = 1$. The total variation distance gets its name from the following natural interpretation:

$$d_{TV}(\nu_1, \nu_2) = \max_{A \subset \mathbb{X}} \text{abs}(\nu_1(A) - \nu_2(A)) .$$

This interpretation means that the total variation distance between ν_1 and ν_2 is the maximal difference in probabilities that the two distributions assign to any one event $A \in \sigma(\mathbb{X}) = 2^{\mathbb{X}}$.

In words, Proposition 74 says that if you run the chain for a sufficiently long enough time t , then, regardless of the initial distribution μ_0 , the distribution at time t will be close to the stationary distribution π . This is referred to as the Markov chain **approaching equilibrium** or **stationarity** as $t \rightarrow \infty$.

Proposition 74 (Markov chain convergence theorem) Let $(X_t)_{t \in \mathbb{Z}_+}$ be an irreducible aperiodic Markov chain with state space $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$, transition matrix $P = (P(x, y))_{(x, y) \in \mathbb{X}^2}$ and initial distribution μ_0 . Then for any distribution π which is stationary for the transition matrix P , we have

$$\mu_t \xrightarrow{TV} \pi . \quad (9.14)$$

Proof: TBD

Proposition 75 (Uniqueness of stationary distribution) Any irreducible aperiodic Markov chain has a unique stationary distribution.

Proof: TBD

Exercise 142 Consider the Markov chain on $\{1, 2, 3, 4, 5, 6\}$ with the following transition matrix:

$$P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 2 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 3 & 0 & 0 & \frac{1}{4} & \frac{3}{4} & 0 \\ 4 & 0 & 0 & \frac{3}{4} & \frac{1}{4} & 0 \\ 5 & 0 & 0 & 0 & 0 & \frac{3}{4} \\ 6 & 0 & 0 & 0 & 0 & \frac{1}{4} \end{pmatrix} .$$

Show that this chain is reducible and it has three stationary distributions:

$$(1/2, 1/2, 0, 0, 0, 0), \quad (0, 0, 1/2, 1/2, 0, 0), \quad (0, 0, 0, 0, 1/2, 1/2) .$$

Exercise 143 If there are two stationary distributions π and π' then show that there is a infinite family of stationary distributions $\{\pi_p : p \in [0, 1]\}$, called the convex combinations of π and π' .

Exercise 144 Show that for a drunkard's walk chain started at state 0 around a polygonal block with k corners labelled $\{0, 1, 2, \dots, k - 1\}$, the state probability vector at time step t

$$\mu_t \xrightarrow{\text{TV}} \pi$$

if and only if k is odd. Explain what happens to μ_t when k is even.

9.5 Reversibility

We introduce another specific property called reversibility. This property will assist in conjuring Markov chains with a desired stationary distribution.

Definition 76 (Reversible) A probability distribution π on $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$ is said to be a **reversible distribution** for a Markov chain $(X_t)_{t \in \mathbb{Z}}$ on \mathbb{X} with transition matrix P if for every pair of states $(x, y) \in \mathbb{X}^2$:

$$\pi(x)P(x, y) = \pi(y)P(y, x) . \quad (9.15)$$

A Markov chain that has a reversible distribution is said to be a reversible Markov chain.

In words, $\pi(x)P(x, y) = \pi(y)P(y, x)$ says that if you start the chain at the reversible distribution π , i.e., $\mu_0 = \pi$, then the probability of going from x to y is the same as that of going from y to x .

Proposition 77 (A reversible π is a stationary π) Let $(X_t)_{t \in \mathbb{Z}_+}$ be a Markov chain on $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$ with transition matrix P . If π is a reversible distribution for $(X_t)_{t \in \mathbb{Z}_+}$ then π is a stationary distribution for $(X_t)_{t \in \mathbb{Z}_+}$.

Proof: Suppose π is a reversible distribution for $(X_t)_{t \in \mathbb{Z}_+}$ then π is a probability distribution on \mathbb{X} and $\pi(x)P(x, y) = \pi(y)P(y, x)$ for each $(x, y) \in \mathbb{X}^2$. We need to show that for any $y \in \mathbb{X}$ we have

$$\pi(y) = \sum_{x \in \mathbb{X}} \pi(y)P(y, x) .$$

Fix a $y \in \mathbb{X}$,

$$\begin{aligned} LHS &= \pi(y) = \pi(y) \cdot 1 = \pi(y) \sum_{x \in \mathbb{X}} P(y, x), \text{ since } P \text{ is a stochastic matrix} \\ &= \sum_{x \in \mathbb{X}} \pi(y)P(y, x) = \sum_{x \in \mathbb{X}} \pi(x)P(x, y), \text{ by reversibility} \\ &= RHS . \end{aligned}$$

Definition 78 (Graph) A **Graph** $\mathbb{G} := (\mathbb{V}, \mathbb{E})$ consists of a **vertex set** $\mathbb{V} := \{v_1, v_2, \dots, v_k\}$ together with an **edge set** $\mathbb{E} := \{e_1, e_2, \dots, e_l\}$. Each edge connects two of the vertices in \mathbb{V} . An edge e_h connecting vertices v_i and v_j is denoted by $\langle v_i, v_j \rangle$. Two vertices are **neighbours** if they share an edge. The **neighbourhood** of a vertex v_i denoted by $\text{nbhd}(v_i) := \{v_j : \langle v_i, v_j \rangle \in \mathbb{E}\}$ is the set of neighbouring vertices of v_i . The number of neighbours of a vertex v_i in an undirected graph is called its **degree** and is denoted by $\deg(v_i)$. Note that $\deg(v_i) = \#\text{nbhd}(v_i)$. In a graph

we only allow one edge per pair of vertices but in a **multigraph** we allow more than one edge per pair of vertices. An edge can be **directed** to preserve the order of the pair of vertices they connect or they can be **undirected**. An edge can be **weighted** by being associated with a real number called its weight. We can represent a directed graph by its **adjacency matrix** given by:

$$A := (A(v_i, v_j))_{(v_i, v_j) \in \mathbb{V} \times \mathbb{V}}, \quad A(v_i, v_j) = \begin{cases} 1 & \text{if } \langle v_i, v_j \rangle \in \mathbb{E} \\ 0 & \text{otherwise.} \end{cases}$$

Thus the adjacency matrix of an undirected graph is symmetric. In a directed graph, each vertex v_i has **in-edges** that come into it and **out-edges** that go out of it. The number of in-edges and out-edges of v_i is denoted by $\text{ideg}(v_i)$ and $\text{odeg}(v_i)$ respectively. Note that a transition diagram of a Markov chain is a weighted directed graph and is represented by the transition probability matrix.

Model 26 (Random Walk on an Undirected Graph) A random walk on an undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ is a Markov chain with state space $\mathbb{V} := \{v_1, v_2, \dots, v_k\}$ and the following transition rules: if the chain is at vertex v_i at time t then it moves uniformly at random to one of the neighbours of v_i at time $t + 1$. If $\deg(v_i)$ is the degree of v_i then the transition probabilities of this Markov chain is

$$P(v_i, v_j) = \begin{cases} \frac{1}{\deg(v_i)} & \text{if } \langle v_i, v_j \rangle \in \mathbb{E} \\ 0 & \text{otherwise,} \end{cases}$$

Proposition 79 The random walk on an undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, with vertex set $\mathbb{V} := \{v_1, v_2, \dots, v_k\}$ and degree sum $d = \sum_{i=1}^k \deg(v_i)$ is a reversible Markov chain with the reversible distribution π given by:

$$\pi = \left(\frac{\deg(v_1)}{d}, \frac{\deg(v_2)}{d}, \dots, \frac{\deg(v_k)}{d} \right).$$

Proof: First note that π is a probability distribution provided that $d > 0$. To show that π is reversible we need to verify Equation 9.15 for each $(v_i, v_j) \in \mathbb{V}^2$. Fix a pair of states $(v_i, v_j) \in \mathbb{V}^2$, then

$$\pi(v_i)P(v_i, v_j) = \begin{cases} \frac{\deg(v_i)}{d} \frac{1}{\deg(v_i)} = \frac{1}{d} = \frac{\deg(v_j)}{d} \frac{1}{\deg(v_j)} = \pi(v_j)P(v_j, v_i) & \text{if } \langle v_i, v_j \rangle \in \mathbb{E} \\ 0 = \pi(v_j)P(v_j, v_i) & \text{otherwise.} \end{cases}$$

By Proposition 77 π is also the stationary distribution.

Exercise 145 Prove Proposition 79 by directly showing that $\pi P = \pi$, i.e., for each $v_i \in \mathbb{V}$, $\sum_{i=1}^k \pi(v_i)P(v_i, v_j) = \pi(v_j)$.

Example 146 (Random Walk on a regular graph) A graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ is called regular if every vertex in $\mathbb{V} = \{v_1, v_2, \dots, v_k\}$ has the same degree δ , i.e., $\deg(v_i) = \delta$ for every $v_i \in \mathbb{V}$. Consider the random walk on a regular graph with symmetric transition matrix

$$Q(v_i, v_j) = \begin{cases} \frac{1}{\delta} & \text{if } \langle v_i, v_j \rangle \in \mathbb{E} \\ 0 & \text{otherwise} \end{cases}.$$

By Proposition 79, the stationary distribution of the random walk on \mathbb{G} is the uniform distribution on \mathbb{V} given by

$$\pi = \left(\frac{\delta}{\delta \#\mathbb{V}}, \dots, \frac{\delta}{\delta \#\mathbb{V}} \right) = \left(\frac{1}{\#\mathbb{V}}, \dots, \frac{1}{\#\mathbb{V}} \right).$$

Example 147 (Triangulated Quadrangle) The random walk on the undirected graph

$$\mathbb{G} = (\{1, 2, 3, 4\}, \{\langle 1, 2 \rangle, \langle 3, 1 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle, \langle 4, 3 \rangle\})$$

depicted below with adjacency matrix A is a Markov chain on $\{1, 2, 3, 4\}$ with transition matrix P :

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{matrix}, \quad P = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} \end{matrix}, \quad \text{Diagram: } \begin{array}{c} \text{1} \\ \text{2} \\ \text{3} \\ \text{4} \end{array} \quad \begin{array}{ccccc} & & & & \\ & \text{1} & \text{2} & \text{3} & \text{4} \\ & & \diagup & \diagdown & \\ & & \text{2} & \text{3} & \text{4} \\ & & \diagdown & \diagup & \\ & & \text{1} & & \end{array} .$$

By Proposition 79, the stationary distribution of the random walk on \mathbb{G} is

$$\pi = \left(\frac{\deg(v_1)}{d}, \frac{\deg(v_2)}{d}, \frac{\deg(v_3)}{d}, \frac{\deg(v_4)}{d} \right) = \left(\frac{2}{10}, \frac{3}{10}, \frac{3}{10}, \frac{2}{10} \right) .$$

Exercise 148 Show that the Drunkard's walk around the block from Simulation 133 is a random walk on the undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ with $\mathbb{V} = \{0, 1, 2, 3\}$ and $\mathbb{E} = \{\langle 0, 1 \rangle, \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 0, 3 \rangle\}$. What is its reversible distribution?

Example 149 (Drunkard's biased walk around the block) Consider the Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ on $\mathbb{X} = \{0, 1, 2, 3\}$ with initial distribution $\mathbf{1}_{\{3\}}(x)$ and transition matrix

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 0 & 1/3 & 0 & 2/3 \\ 1/3 & 0 & 2/3 & 0 \\ 0 & 1/3 & 0 & 2/3 \\ 1/3 & 0 & 2/3 & 0 \end{pmatrix} \end{matrix} .$$

Draw the transition diagram for this Markov chain that corresponds to a drunkard who flips a biased coin to make his next move at each corner. The stationary distribution is $\pi = (1/4, 1/4, 1/4, 1/4)$ (verify $\pi P = \pi$).

We will show that $(X_t)_{t \in \mathbb{Z}_+}$ is not a reversible Markov chain. Since $(X_t)_{t \in \mathbb{Z}_+}$ is irreducible (aperiodicity is not necessary for uniqueness of π) π is the unique stationary distribution. Due to Proposition 77, π has to be a reversible distribution in order for $(X_t)_{t \in \mathbb{Z}_+}$ to be a reversible Markov chain. But reversibility fails for π since,

$$\pi(0)P(0, 1) = \frac{1}{4} \times \frac{1}{3} = \frac{1}{12} < \frac{1}{6} = \frac{1}{4} \times \frac{2}{3} = \pi(1)P(1, 0) .$$

Exercise 150 Find the stationary distribution of the Markov chain in Exercise 141.

Model 27 (Random Walk on a Directed Graph) A random walk on a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ is a Markov chain with state space $\mathbb{V} := \{v_1, v_2, \dots, v_k\}$ and transition matrix given by:

$$P(v_i, v_j) = \begin{cases} \frac{1}{\text{odeg}(v_i)} & \text{if } \langle v_i, v_j \rangle \in \mathbb{E} \\ 0 & \text{otherwise,} \end{cases}$$

Example 151 (Directed Triangulated Quadrangle) The random walk on the directed graph

$$\mathbb{G} = (\{1, 2, 3, 4\}, \{\langle 1, 2 \rangle, \langle 3, 1 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle, \langle 4, 3 \rangle\})$$

depicted below with adjacency matrix A is a Markov chain on $\{1, 2, 3, 4\}$ with transition matrix P :

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}, \quad P = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}, \quad \text{Diagram: } \begin{array}{c} \text{1} \xrightarrow{\quad} \text{2} \xrightarrow{\quad} \text{4} \xrightarrow{\quad} \text{3} \\ \text{1} \xleftarrow{\quad} \text{2} \end{array}.$$

Exercise 152 Show that there is no reversible distribution for the Markov chain in Example 151.

Example 153 (Random surf on the word wide web) Consider the huge graph with vertices as webpages and hyper-links as undirected edges. Then Model 26 gives a random walk on this graph. However if a page has no links to other pages, it becomes a sink and therefore terminates the random walk. Let us modify this random walk into a **random surf** to avoid getting stuck. If the random surfer arrives at a sink page, she picks another page at random and continues surfing at random again. Google's PageRank formula uses a random surfer model who gets bored after several clicks and switches to a random page. The PageRank value of a page reflects the chance that the random surfer will land on that page by clicking on a link. The stationary distribution of the random surfer on the world wide web is a very successful model for ranking pages.

Model 28 (Lazy Random Walk) You can convert a random walk on an undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ into a **lazy random walk** on \mathbb{G} by the following steps:

- Add loops to each vertex in $\mathbb{V} = \{v_1, v_2, \dots, v_k\}$ to obtain a new set of edges $\mathbb{E}' = \mathbb{E} \cup \{\langle v_1, v_1 \rangle, \langle v_2, v_2 \rangle, \dots, \langle v_k, v_k \rangle\}$.
- Construct the lazy graph $\mathbb{G}' = (\mathbb{V}, \mathbb{E}')$.
- Do a random walk on the undirected graph \mathbb{G}' .

The lazy random walk allows us to introduce aperiodicity quite easily.

Exercise 154 (Lazy Random Walk on the Triangulated Quadrangle) Consider the random walk of Example 147 on the undirected graph

$$\mathbb{G} = (\{1, 2, 3, 4\}, \{\langle 1, 2 \rangle, \langle 3, 1 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle, \langle 4, 3 \rangle\}).$$

Construct the lazy random walk on \mathbb{G} , obtain its transition probability matrix and state transition diagram. Show that the stationary distribution of this lazy random walk on \mathbb{G} is

$$\pi = \left(\frac{3}{14}, \frac{4}{14}, \frac{4}{14}, \frac{3}{14} \right).$$

Model 29 (Random Walks on Groups) Under 

Model 30 (Birth-Death chains) Under 

9.6 Metropolis-Hastings Markov chain

Definition 80 (Metropolis-Hastings Markov chain) If we are given an irreducible Markov chain $(Y_t)_{t \in \mathbb{Z}_+}$ called the **base chain** or the **proposal chain** on a finite state space $\mathbb{X} = \{s_1, s_2, \dots, s_k\}$ with transition probability matrix $Q = (Q(x, y))_{(x,y) \in \mathbb{X}^2}$ and some probability distribution π on \mathbb{X} of interest that may only be known up to a normalizing constant as $\tilde{\pi}$, i.e., $\pi(x) = (\sum_{z \in \mathbb{X}} \tilde{\pi}(z))^{-1} \tilde{\pi}(x)$ for each $x \in \mathbb{X}$, then we can construct a new Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ called the **Metropolis-Hastings chain** on \mathbb{X} with the following transition probabilities:

$$P(x, y) = \begin{cases} Q(x, y)a(x, y) & \text{if } x \neq y \\ 1 - \sum_{z \in \{z \in \mathbb{X}: z \neq x\}} Q(x, z)a(x, z) & \text{if } x = y \end{cases}, \quad (9.16)$$

where the acceptance probability is

$$a(x, y) := \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \frac{Q(y, x)}{Q(x, y)} \right\}. \quad (9.17)$$

Note that we only need to know π up to ratios. Thus, $\pi(y)/\pi(x)$ in $a(x, y)$ can be replaced by $\tilde{\pi}(y)/\tilde{\pi}(x)$ since

$$\frac{\pi(y)}{\pi(x)} = \frac{(\sum_{z \in \mathbb{X}} \tilde{\pi}(z))^{-1} \tilde{\pi}(y)}{(\sum_{z \in \mathbb{X}} \tilde{\pi}(z))^{-1} \tilde{\pi}(x)} = \frac{\tilde{\pi}(y)}{\tilde{\pi}(x)}.$$

Algorithm 11 describes how to simulate samples from a Metropolis-Hastings Markov chain.

Proposition 81 (Stationarity of the Metropolis-Hastings chain) The Metropolis-Hastings chain constructed according to Definition 80 has π as its stationary distribution.

Proof: It suffices to show that π is the reversible distribution for $(X_t)_{t \in \mathbb{Z}_+}$, i.e., for each $(x, y) \in \mathbb{X}^2$, $\pi(x)P(x, y) = \pi(y)P(y, x)$. Fix a pair $(x, y) \in \mathbb{X}^2$ and suppose $x \neq y$. Then,

$$\begin{aligned} \pi(x)P(x, y) &= \pi(x)Q(x, y)a(x, y) \\ &= \pi(x)Q(x, y) \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \frac{Q(y, x)}{Q(x, y)} \right\} \\ &= \min \left\{ \pi(x)Q(x, y), \pi(x)Q(x, y) \frac{\pi(y)}{\pi(x)} \frac{Q(y, x)}{Q(x, y)} \right\} \\ &= \min \{ \pi(x)Q(x, y), \pi(y)Q(y, x) \} \\ &= \min \{ \pi(y)Q(y, x), \pi(x)Q(x, y) \} \\ &= \min \left\{ \pi(y)Q(y, x), \pi(y)Q(y, x) \frac{\pi(x)}{\pi(y)} \frac{Q(x, y)}{Q(y, x)} \right\} \\ &= \pi(y)Q(y, x) \min \left\{ 1, \frac{\pi(x)}{\pi(y)} \frac{Q(x, y)}{Q(y, x)} \right\} \\ &= \pi(y)P(y, x). \end{aligned}$$

When $x = y$, reversibility is trivially satisfied since $\pi(x)P(x, y) = \pi(y)P(y, x) = \pi(x)P(x, x)$.

Definition 82 If the base chain $(Y_t)_{t \in \mathbb{Z}_+}$ in the Metropolis-Hastings Markov chain of Definition 80 has a symmetric transition matrix Q with $Q(x, y) = Q(y, x)$ for each $(x, y) \in \mathbb{X}^2$ then the acceptance probability in Equation 9.17 simplifies to

$$a(x, y) = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \right\},$$

and the corresponding Metropolis-Hastings chain $(X_t)_{t \in \mathbb{Z}_+}$ is called the **Metropolis chain**.

Algorithm 11 Metropolis-Hastings Markov chain1: *input:*(1) shape of a target density $\tilde{\pi}(x) = (\sum_{x \in \mathbb{X}} \tilde{\pi}(x)) \pi(x)$,(2) sampler for the base chain that can produce samples $y \sim Q(x, \cdot)$.2: *output:* a sequence of samples x_0, x_1, \dots, x_n from the Metropolis-Hastings Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ with stationary distribution π 3: Choose initial state $x_0 \in \mathbb{X}$ according to μ_0 4: **repeat**5: At iteration t ,6: Generate $y \sim Q(x_{t-1}, \cdot)$ and $u \sim \text{Uniform}(0, 1)$,7: Compute *acceptance probability*

$$a(x_{t-1}, y) = \min \left\{ 1, \frac{\tilde{\pi}(y)}{\tilde{\pi}(x_{t-1})} \frac{Q(y, x_{t-1})}{Q(x_{t-1}, y)} \right\},$$

8: **If** $u \leq a(x_{t-1}, y)$ **then** $x_t \leftarrow y$, **else** $x_t \leftarrow x_{t-1}$ 9: **until** desired number of samples n are obtained from $(X_t)_{t \in \mathbb{Z}_+}$

Suppose you know neither the vertex set \mathbb{V} nor the edge set \mathbb{E} entirely for an undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ but you are capable of walking locally on \mathbb{G} . In other words, if you are currently at vertex x you are able to make a move to one of the neighbouring vertices of x . However, you do not know every single vertex in \mathbb{V} or the entire set of edges \mathbb{E} as an adjacency matrix for instance. Several real-world problems fall in this class. Some examples include the random surfer on www to rank web pages (Example 153), social network analyses in facebook or twitter, exact tests for contingency tables, etc.

Model 31 (Metropolis-Hastings Random Walk on Graph) Let $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ be an undirected graph and let $(Y_t)_{t \in \mathbb{Z}_+}$ with transition matrix Q be an irreducible random walk on \mathbb{G} and let π be a probability distribution on $\mathbb{V} = \{v_1, v_2, \dots, v_k\}$ that is known upto a normalizing constant as $\tilde{\pi}$. The **Metropolis-Hastings random walk** on \mathbb{G} is the Metropolis-Hastings Markov chain $(X_t)_{t \in \mathbb{Z}_+}$ on \mathbb{V} with base chain $(Y_t)_{t \in \mathbb{Z}_+}$ and the following transition probabilities:

$$P(x, y) = \begin{cases} \frac{1}{\deg(v_i)} \min \left\{ 1, \frac{\tilde{\pi}(v_j)}{\tilde{\pi}(v_i)} \frac{\deg(v_i)}{\deg(v_j)} \right\} & \text{if } \langle v_i, v_j \rangle \in \mathbb{E} \\ 1 - \sum_{v_l \in \text{nbhd}(v_i)} \left(\frac{1}{\deg(v_i)} \min \left\{ 1, \frac{\tilde{\pi}(v_l)}{\tilde{\pi}(v_i)} \frac{\deg(v_i)}{\deg(v_l)} \right\} \right) & \text{if } v_i = v_j \\ 0 & \text{otherwise} \end{cases}.$$

By Proposition 81, $(X_t)_{t \in \mathbb{Z}_+}$ has π as its stationary distribution. This Markov chain can be simulated as follows:

- Suppose $x_t = v_i$ at time t
- Propose v_j uniformly at random from $\text{nbhd}(v_i)$
- Sample u from $\text{Uniform}(0, 1)$
- If $u < \min\{1, \pi(v_j) \deg(v_i) / \pi(v_i) \deg(v_j)\}$ then $x_{t+1} = v_j$ else $x_{t+1} = x_t$

Model 32 (Metropolis chain on a regular graph) Consider the random walk $(Y_t)_{t \in \mathbb{Z}_+}$ on a regular graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ with $\deg(v_i) = \delta$ for every vertex $v_i \in \mathbb{V} = \{v_1, v_2, \dots, v_k\}$ and the symmetric transition matrix

$$Q(v_i, v_j) = \begin{cases} \frac{1}{\delta} & \text{if } \langle v_i, v_j \rangle \in \mathbb{E} \\ 0 & \text{otherwise} \end{cases}.$$

You can sample from a given distribution π on \mathbb{V} by constructing the Metropolis chain with stationary distribution π from the base chain given by $(Y_t)_{t \in \mathbb{Z}_+}$.

Model 33 (sampling from a uniform distribution over an irregular graph) A graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ that is not regular is said to be irregular. Clearly, the stationary distribution of a random walk on \mathbb{G} is not uniform. Suppose you want to sample uniformly from \mathbb{V} according to $\pi(v_i) = (\#\mathbb{V})^{-1}$ for each $v_i \in \mathbb{V}$. We can accomplish this by constructing a Metropolis-Hastings Markov chain with the random walk on \mathbb{G} as the base chain and the following transition probabilities:

$$P(v_i, v_j) = \begin{cases} \frac{1}{\deg(v_i)} \min \left\{ 1, \frac{\deg(v_i)}{\deg(v_j)} \right\} & \text{if } \langle v_i, v_j \rangle \in \mathbb{E} \\ 1 - \sum_{v_l \in \text{nbhd}(v_i)} \left(\frac{1}{\deg(v_i)} \min \left\{ 1, \frac{\deg(v_i)}{\deg(v_l)} \right\} \right) & \text{if } v_i = v_j \\ 0 & \text{otherwise} \end{cases}.$$

Thus the Metropolis-Hastings walk on \mathbb{G} is biased against visiting higher degree vertices and thereby samples uniformly from \mathbb{V} at stationarity.

Example 155 (Stochastic Optimization) Let $f : \mathbb{V} \rightarrow \mathbb{R}$ and $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ be an undirected graph. Let the global maximum be

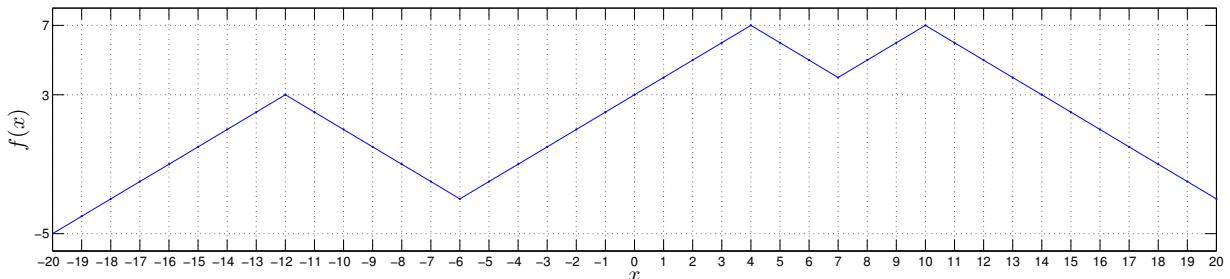
$$f^* := \max_{y \in \mathbb{V}} f(y),$$

and the set of maximizers of f be

$$\mathbb{V}^* := \underset{x \in \mathbb{V}}{\operatorname{argmax}} f(x) = \{x \in \mathbb{V} : f(x) = f^*\}.$$

In many problems such as maximum likelihood estimation, minimizing a cost function by maximizing its negative, etc, one is interested in $\mathbb{V}^* \subset \mathbb{V}$. This global maximization problem is difficult when $\#\mathbb{V}$ is huge. A deterministic hill-climbing or gradient ascent algorithm that iteratively moves from the current state v_i to a neighbouring state v_j if $f(v_j) > f(v_i)$ can easily get trapped in a local peak of f and thereby miss the global peak attained by elements in \mathbb{V}^* .

Figure 9.5: Stochastic Optimization with Metropolis chain.



For example consider the global maximization problem shown in Figure 9.5 with

$$f^* = 7 \text{ and } \mathbb{V}^* = \{4, 10\} \subset \mathbb{V} = \{-20, -19, \dots, 19, 20\} .$$

The deterministic hill-climbing algorithm will clearly miss \mathbb{V}^* and terminate at the local maximum of 3 at -12 if initialised at any element in $\{-20, -19, \dots, -8, -7\}$. Also, this algorithm will not find both elements in \mathbb{V}^* even when initialised more appropriately.

We will construct a Markov chain to solve this global maximization problem. For a fixed parameter $\lambda \in \mathbb{R}_{>0}$, let

$$\pi_\lambda(x) = \frac{\lambda^{f(x)}}{\sum_{z \in \mathbb{V}} \lambda^{f(z)}} .$$

Since $\pi_\lambda(x)$ is increasing in $f(x)$, $\pi_\lambda(x)$ favours vertices with large $f(x)$. First form a graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ by adding edges between the vertices in \mathbb{V} so that you can get from any vertex to any other vertex in \mathbb{V} by following a sequence of edges in \mathbb{E} . Now using the random walk on \mathbb{G} as the base chain let us construct a Metropolis-Hastings chain $(X_t)_{t \in \mathbb{Z}_+}$ on \mathbb{G} with π_λ on \mathbb{V} as its stationary distribution.

For simplicity, let us suppose that \mathbb{G} is a regular graph with a symmetric transition matrix Q for the base chain and thereby making $(X_t)_{t \in \mathbb{Z}_+}$ a Metropolis chain. For instance, in the Example from Figure 9.5 with $\mathbb{V} = \{-20, -19, \dots, 19, 20\}$, we can obtain a Metropolis chain on \mathbb{V} with stationary distribution π_λ by taking \mathbb{E} in $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ to be

$$\mathbb{E} = \{\langle -20, -19 \rangle, \langle -19, -18 \rangle, \langle -18, -17 \rangle, \dots, \langle 17, 18 \rangle, \langle 18, 19 \rangle, \langle 19, 20 \rangle\} .$$

Then, if $f(y) < f(x)$, the Metropolis chain accepts a transition from x to y with probability

$$\frac{\pi_\lambda(y)}{\pi_\lambda(x)} = \frac{\lambda^{f(y)}}{\lambda^{f(x)}} = \lambda^{f(y)-f(x)} = \lambda^{-(f(x)-f(y))} .$$

As $\lambda \rightarrow \infty$, the Metropolis chain approaches the deterministic hill-climbing algorithm and yields a uniform distribution over \mathbb{V}^* as follows:

$$\lim_{\lambda \rightarrow \infty} \pi_\lambda(x) = \lim_{\lambda \rightarrow \infty} \frac{\lambda^{f(x)}/\lambda^{f^*}}{\#\mathbb{V}^* + \sum_{z \in \mathbb{V} \setminus \mathbb{V}^*} \lambda^{f(z)}/\lambda^{f^*}} = \frac{\mathbb{1}_{\mathbb{V}^*}(x)}{\#\mathbb{V}^*} .$$

9.7 Glauber Dynamics

Let \mathbb{S} be a finite set of states. Let \mathbb{V} be a set of vertices. Typically, \mathbb{S} contains characters or colours that can be taken by each site or vertex in \mathbb{V} . Let $x \in \mathbb{S}^\mathbb{V}$ be a configuration, i.e., a function from \mathbb{V} to \mathbb{S} . A configuration can be thought of as a labelling of vertices in \mathbb{V} with elements in \mathbb{S} .

Definition 83 (Glauber dynamics for π) Let \mathbb{V} and \mathbb{S} be finite sets and let $\mathbb{X} \subset \mathbb{S}^\mathbb{V}$ which forms the support of the probability distribution π on $\mathbb{S}^\mathbb{V}$, i.e.,

$$\mathbb{X} = \{x \in \mathbb{S}^\mathbb{V} : \pi(x) > 0\} .$$

The **Glauber dynamics** or **Gibbs sampler** for π is a reversible Markov chain on \mathbb{X} with stationary distribution π under the following transition mechanism. Let the current state at time t be x . To obtain the state at time $t+1$ first choose a vertex v uniformly at random from \mathbb{V} and then choose

a new state according to π conditioned on the set of states equal to x at all vertices other than v . We give the details of this transition mechanism next.

For $x \in \mathbb{X}$ and $v \in \mathbb{V}$, define the set of states identical to x everywhere except possibly at v as

$$\mathbb{X}(x, v) := \{y \in \mathbb{X} : y(w) = x(w) \text{ for all } w \neq v\} .$$

Now let

$$\pi^{x,v}(y) := \pi(y|\mathbb{X}(x, v)) = \begin{cases} \left(\sum_{z \in \mathbb{X}(x, v)} \pi(z)\right)^{-1} \pi(y) & \text{if } y \in \mathbb{X}(x, v) \\ 0 & \text{if } y \notin \mathbb{X}(x, v) \end{cases}$$

be the distribution π conditioned on $\mathbb{X}(x, v)$. Therefore the rule for updating the current state x is:

- pick a vertex v uniformly at random from \mathbb{V} ,
- choose a new configuration by sampling from $\pi^{x,v}$.

Proposition 84 (Stationarity of Glauber dynamics) The Glauber dynamics for π on $\mathbb{X} \subset \mathbb{S}^{\mathbb{V}}$ has π as its reversible and stationary distribution.

Proof: Exercise.

Model 34 (Hard-core model) Let $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ be an undirected graph. An assignment of elements of $\mathbb{S} = \{0, 1\}$ to vertices in \mathbb{V} is called a configuration. Thus, the configuration x is a function $x : \mathbb{V} \rightarrow \mathbb{S}$ and $x \in \mathbb{S}^{\mathbb{V}}$. The vertices v of a configuration x with $x(v) = 1$ are said to be occupied and those with $x(v) = 0$ are said to be vacant. Thus a configuration models a placement of particles on the vertices of \mathbb{V} . A hard-core configuration is a configuration in which no two neighbouring vertices are occupied. More formally, a configuration x is called hard-core if $\sum_{(v_i, v_j) \in \mathbb{E}} x(v_i)x(v_j) = 0$. Let the set of hard-core configurations be \mathbb{X} and let π be the uniform distribution on \mathbb{X} , given by

$$\pi(x) = \begin{cases} \frac{1}{\#\mathbb{X}} & \text{if } x \in \mathbb{X} \\ 0 & \text{otherwise} \end{cases} .$$

The Glauber dynamics $(X_t)_{t \in \mathbb{Z}_+}$ for the uniform distribution π on hard-core configurations can be simulated as follows:

- initialize with vacant vertices, i.e., $X_0(w) = 0$ for each $w \in \mathbb{V}$,
- let the current hard-core configuration be $x_t : \mathbb{V} \rightarrow \{0, 1\}$ at time t ,
- choose a vertex v uniformly at random from \mathbb{V} ,
- if any neighbour of v is occupied then v is left vacant, i.e., $x_{t+1}(v) = 0$
- if every neighbour of v is vacant then v is occupied with probability $1/2$, i.e., $x_{t+1}(v) = 1$,
- leave the values at all other vertices unchanged, i.e., $x_{t+1}(w) = x_t(w)$ for each $w \neq v$,
- the possibly modified configuration x_{t+1} is the updated hard-core configuration at time $t + 1$.

Proposition 85 The Glauber dynamics of Model 34 does indeed have π as its stationary distribution.

Proof: First we need to verify that $(X_t)_{t \in \mathbb{Z}_+}$, the Markov chain given by the Glauber dynamics for π in Model 34, is irreducible and aperiodic. Clearly $(X_t)_{t \in \mathbb{Z}_+}$ is aperiodic since we can get from any hard-core configuration $x \in \mathbb{X}$ to itself in one time step by choosing a vertex with at least one occupied neighbour and leaving the chosen vertex unchanged or by choosing a vertex with no occupied neighbours and leaving the chosen vertex unchanged with probability $1/2$. Next we need to establish irreducibility, i.e., we need to show that we can get from any hardcore configuration x to any other hardcore configuration x' in finitely many steps. Let the vacant configuration be \tilde{x} , i.e., $\tilde{x}(v) = 0$ for every vertex $v \in \mathbb{V}$. In finitely many steps, we can go from any x to \tilde{x} and from \tilde{x} to x' . If x has $s(x) := \sum_{v \in \mathbb{V}} x(v)$ occupied sites or vertices then we can go to the vacant configuration \tilde{x} with $s(\tilde{x}) = 0$ in $s(x)$ time steps by picking one of the currently occupied sites and making it vacant as follows:

$$\mathbf{P}(X_{t+s(x)} = \tilde{x} | X_t = x) = \prod_{i=0}^{s(x)-1} \frac{(s(x) - i)}{\#\mathbb{V}} \frac{1}{2} > 0 .$$

Similarly, we can go from \tilde{x} to any other configuration x' with $s(x')$ many occupied sites in $s(x')$ time steps with the following positive probability:

$$\mathbf{P}(X_{t+s(x')} = x' | X_t = \tilde{x}) = \prod_{i=0}^{s(x')-1} \frac{(s(x') - i)}{\#\mathbb{V}} \frac{1}{2} > 0 .$$

Note that this is not the shortest possible number of steps to go from x to x' but just a finite number of steps. Thus we have established that $x \leftrightarrow x'$ for every $(x, x') \in \mathbb{X}$ and thereby established irreducibility of the chain $(X_t)_{t \in \mathbb{Z}_+}$.

If we now show that π is reversible for $(X_t)_{t \in \mathbb{Z}_+}$ then by Proposition 77 π is also stationary for $(X_t)_{t \in \mathbb{Z}_+}$ and finally π is the unique stationary distribution due to irreducibility and aperiodicity. Let $P(x, y)$ be the probability of going from x to y in one time step of $(X_t)_{t \in \mathbb{Z}_+}$. We need to show that for any pair of hardcore configurations $(x, y) \in \mathbb{X}^2$ the following equality holds:

$$\pi(x)P(x, y) = \pi(y)P(y, x), \quad \pi(x) = \frac{1}{\#\mathbb{X}} .$$

Let the number of vertices at which x and y differ be $d(x, y) := \sum_{v \in \mathbb{V}} \text{abs}(x(v) - y(v))$. Let us consider three cases of $(x, y) \in \mathbb{X}^2$.

Case i: When $d(x, y) = 0$ the two configurations are identical, i.e., $x = y$, and therefore we have the trivial equality:

$$\pi(x)P(x, y) = \pi(x)P(x, x) = \pi(y)P(y, x) .$$

Case ii: When $d(x, y) > 1$ the two configurations differ at more than one vertex and therefore $P(x, y) = 0$ and we have the trivial equality:

$$\pi(x)P(x, y) = \pi(x)0 = 0 = \pi(y)P(y, x) .$$

Case iii: When $d(x, y) = 1$ the two configurations differ at exactly one vertex v and therefore all neighbouring vertices of v must be vacant, i.e., take the value 0, in both x and y with $P(x, y) = P(y, x) = \frac{1}{\#\mathbb{V}} \frac{1}{2}$. Thus,

$$\pi(x)P(x, y) = \frac{1}{\#\mathbb{X}} \left(\frac{1}{\#\mathbb{V}} \frac{1}{2} \right) = \pi(y)P(y, x) .$$

We have established that $\pi(x) = 1/\#\mathbb{X}$ for each $x \in \mathbb{X}$ is the reversible distribution and thereby also the unique stationarity distribution for $(X_t)_{t \in \mathbb{Z}_+}$, the Markov chain given by the Glauber dynamics for π in Model 34.

Exercise 156 (1-D hardcore model) Let \mathbb{X}_n be the set of hardcore configurations on a path graph with n vertices. Recall that a path graph $\mathbb{G}_n = (\mathbb{V}_n, \mathbb{E}_n)$ has n vertices and $n - 1$ edges, as follows:

$$\mathbb{V}_n = \{v_1, v_2, \dots, v_n\}, \quad \mathbb{E}_n = \{\langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \dots, \langle v_{n-1}, v_n \rangle\} .$$

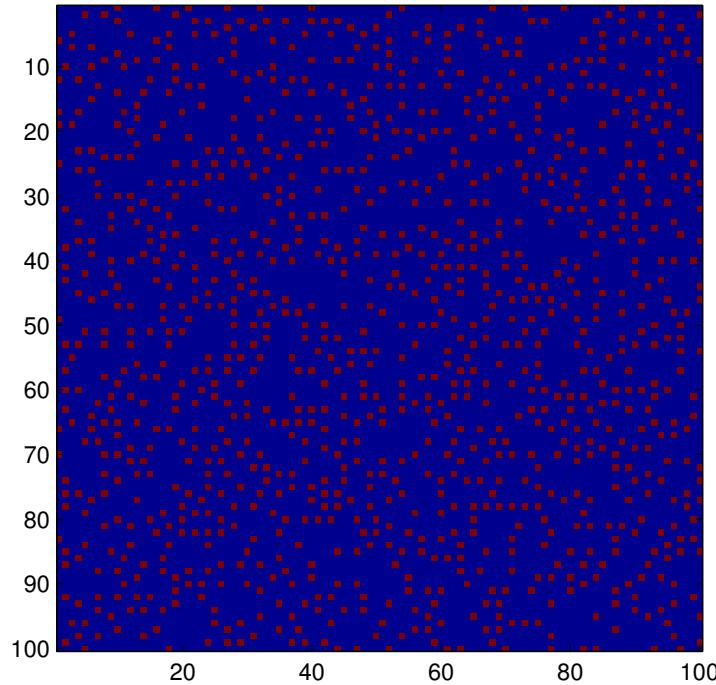
Draw all five hardcore configurations in \mathbb{X}_3 . Show that for any positive integer n ,

$$\#\mathbb{X}_n = \text{fib}(n + 1) ,$$

the $(n + 1)$ -th Fibonacci number, that is defined recursively as follows:

$$\text{fib}(0) := \text{fib}(1) := 1, \quad \text{fib}(n) = \text{fib}(n - 1) + \text{fib}(n - 2), \quad n \geq 1 .$$

Figure 9.6: The sample at time step 10^6 from the Glauber dynamics for the hardcore model on 100×100 regular torus grid. A red site is occupied while a blue site is vacant.



Simulation 157 (Glauber dynamics for the hardcore model on a 2D regular torus) Let us implement a program in MATLAB that will simulate Glauber dynamics to sample uniformly from the hardcore configurations on the undirected regular torus graph. We can report the sample mean of the fraction of occupied sites on this graph from the simulated sequence and make a movie of the simulations (last frame is shown in Figure 9.6).

```
>> Hardcore2D
Avg1s = 0.1128
```

The simulation was implemented in the following M-file:

```
HardCore2D.m
```

```
% simulation of Glauber dynamics for the hardcore model on
% 2D regular torus grid
clf; %clear; clc; % clear current settings
Seed=347632321; rand('twister',Seed); % set seed for PRNG
MaxSteps=1000000; % number of time steps to simulate
DisplayStepSize=10000; % display interval
Steps=0; % initialize time-step to 0
StepsM=1; % index for movie frame
Rows=100; % number of rows
Cols=100; % number of columns
CC = zeros(Rows,Cols,'int8'); %initialize all sites to be vacant
Delta=[-1,0,+1]; % neighbourhood of indices along one coordinate
Avg1s=0.0;%initialise the Average Fraction of occupied sites
while(Steps <= MaxSteps)
    % find a random site with 0 for possible swap
    I=ceil(Rows*rand); J=ceil(Cols*rand);
    % Get the Nbhd of CC(I,J)
    RowNbhd = mod((I-1)+Delta,Rows)+1;
    ColNbhd = mod((J-1)+Delta,Cols)+1;
    Nbhd=CC(RowNbhd, ColNbhd);
    To1Is=find(Nbhd); % find the 1s in Nbhd of CC(I,J)
    Num1s=length(To1Is); % total number of 1s in Nbhd
    if(Num1s > 0)
        CC(I,J)=0; % set site to be vacant
    elseif(rand < 0.5)
        CC(I,J)=1; % set site to be occupied
    else
        CC(I,J)=0; % set site to be vacant
    end
    Steps=Steps+1; % increment time step
    Frac1s=sum(sum(CC))/(Rows*Cols); % fraction of occupied sites
    Avg1s = Avg1s + (Frac1s - Avg1s)/Steps; % online sample mean
    if(mod(Steps,DisplayStepSize)==0)
        A(StepsM)=getframe; % get the frame into A
        imagesc(CC)
        axis square
        StepsM=StepsM+1;
    end
end
Avg1s % print the sample mean of fraction of occupied sites
movie(A,5) % make a movie
```

Model 35 (Ising model) Let $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ be an undirected graph. The Ising model is a probability distribution on $\mathbb{X} = \{-1, +1\}^{\mathbb{V}}$, i.e., a way of randomly assigning elements from the set $\{-1, +1\}$ to vertices of \mathbb{G} . The physical interpretation of the model is that each vertex is the position of an atom in a ferromagnetic material and $+1$'s or -1 's denote the two possible spin orientations of the atoms. There is a parameter β in the model called inverse temperature and $\beta \in [0, \infty)$. Associated with each spin configuration $x \in \mathbb{X}$ is its energy

$$H(x) = - \sum_{\langle u,v \rangle \in \mathbb{E}} x(u)x(v)$$

where $x(u)$ and $x(v)$ give the spin orientations of the atoms at vertices u and v , respectively. So, each edge $\langle u, v \rangle$ adds 1 to the energy $H(x)$ if its neighbouring vertices have opposite spins and subtracts 1 from $H(x)$ otherwise. Thus, lower energy is equivalent to a higher agreement in spins between neighbouring vertices.

The Ising model on \mathbb{G} at inverse temperature β means a random spin configuration X with

$$\mathbf{P}(X = x) = \pi_{\mathbb{G}, \beta}(x) = \frac{1}{Z_{\mathbb{G}, \beta}} \exp(-\beta H(x)) = \frac{1}{Z_{\mathbb{G}, \beta}} \exp\left(\beta \sum_{\langle u, v \rangle \in \mathbb{E}} x(u)x(v)\right),$$

where $Z_{\mathbb{G}, \beta} = \sum_{x \in \mathbb{X}} \exp(-\beta H(x))$ is the normalising constant.

Labwork 158 (Glauber dynamics for the Ising model on a 2D regular torus) Implement a program in MATLAB to simulate from the Ising model on the undirected regular torus graph.

Let us explore the physical interpretation of the Ising model further. If the inverse temperature $\beta = 0$ then we are at infinite temperature and therefore every configuration in \mathbb{X} is equally likely, i.e., $\pi_{\mathbb{G}, 0} = 1/\#\mathbb{X}$. At the other extreme, if $\beta \rightarrow \infty$ then we are approaching zero temperature and the probability over \mathbb{X} under $\pi_{\mathbb{G}, \infty}$ is equally split between “all +1” configuration and “all -1” configuration. However, if $\beta > 0$, then we are at some temperature $1/\beta$ that is neither absolutely hot or absolutely cold and therefore the model will favour configurations with lower energy as opposed to higher energy. Such favourable low energy configurations tend to have neighbouring clumps of identical spins. We say that there is a phase transition in β since the Ising model’s qualitative behaviour depends on whether β is above or below a critical threshold β_c .

Model 36 (Proper q -colourings) A proper q -colouring of an undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ is an assignment of q colours labelled $\{1, 2, \dots, q\}$ to vertices in \mathbb{V} , subject to the constraint that neighbouring vertices do not receive the same colour. Let \mathbb{X} denote the set of all proper q -colourings of \mathbb{G} . If \mathbb{V} is large then \mathbb{X} can be a large and complicated subset of $\{1, 2, \dots, q\}^{\mathbb{V}}$. Note that proper q colourings are a natural generalisation of the hardcore model.

9.7.1 Random Walks on \mathbb{Z} and the reflection principle



9.8 Coupling from the past

MCMC algorithms make it easy to implement a Markov chain that has a given distribution as its stationary distribution. When used on their own, however, MCMC algorithms can only provide sample values that approximate a desired distribution. To obtain sample values that have a desired distribution *exactly* or *perfectly*, MCMC algorithms must be used in conjunction with ideas that make clever use of coupling.

MCMC convergence diagnostics based on *multiple* independent or *coupled* Markov chains running *forward* in time have been suggested, but are not completely reliable. The chains are coupled if the same sequence of random numbers is used to propagate all of them. By adopting a different perspective - running multiple coupled chains from the past or *backward coupling* - Propp & Wilson (1996) developed the *coupling from the past (CFTP)* algorithm, which allowed exact sample values to be obtained from the stationary distribution of an ergodic Markov chain with *finite* state space.

Let us first appreciate the trouble with MCMC algorithms such as Metropolis-Hastings chain, Metropolis chain and Glauber dynamics. Firstly, no matter how large we make time t to be we

cannot avoid the discrepancy between the t -step distribution μ_t and the stationary distribution π . Consider the following transition probability matrix:

$$P = \begin{matrix} & s_1 & s_2 \\ s_1 & \left(\begin{array}{cc} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{array} \right) \\ s_2 & & \end{matrix}$$

We can prove by induction that

$$\mu_t = \left(\frac{1}{2} (1 + 2^{-t}), \frac{1}{2} (1 - 2^{-t}) \right)$$

for every $t \in \mathbb{Z}_+$. The stationary distribution is $\pi = (1/2, 1/2)$. So, as t approaches infinity $\mu_t \xrightarrow{\text{TV}} \pi$, however for any t the total variation distance between $d_{\text{TV}}(\mu_t, \pi) = 2^{-t}$ is strictly positive. Even in this simple example μ_t may never equal π for any finite t , however large. Thus, we have to settle for an approximation to π with some acceptable error ϵ . Secondly, to make the approximation error measured by $d_{\text{TV}}(\mu_t, \pi)$ smaller than ϵ we have to find the ϵ -burnin time τ_ϵ by which $d_{\text{TV}}(\mu_{\tau_\epsilon}, \pi) < \epsilon$. Determining τ_ϵ is nontrivial except in special cases and constitutes an active field of research.

The following material is under .

Demonstration 159 (Applet – Perfect sampling.) The CFTP algorithm starts multiple Markov chains, one for each possible state, at some time $t_0 < 0$ in the past, and uses coupled transitions to propagate them to time 0. If all the chains *coalesce*, (i.e. end up having the same state, at or before time 0), then they will have “forgotten” their starting values and will evolve as a single chain from that point onwards. The common state at time zero ($X^{(0)}$) is an exact sample value from the stationary distribution. Intuitively, if coalescence occurs at some finite time, $t^* < 0$, then if the chains had been started in the infinite past, coupling with the same sequence of random numbers will ensure that they coalesce at t^* , and the common chain at time 0 must be stationary because it had been running for an infinitely long time. Thus, the existence of a finite coalescence time can give a stationary sample value in finite time. The use of coupling is essential to induce coalescence in a finite length of time.

Consider a Markov chain with finite state space, $S = 1, 2, \dots, K$. The CFTP algorithm starts K Markov chains, one from each state in S , at some time $t_0 < 0$ in the past. A sequence of t_0 random vectors, $R^{t+1}, R^{t+2}, \dots, R^0$, is generated and used to propagate all K Markov chains to time 0. Let $X^{t,k(t_0)}$ represent the state of the Markov chain at time t , starting from state $k \in S$ at time $t_0 < t$, and let φ be the update function of the Markov chain, such that:

$$X^{(t+1,k(t_0))} = \varphi(X^{(t,k(t_0))}, R^{(t+1)}) \quad (9.18)$$

9.8.1 Algorithm – Coupling from the past.

Set $t_0 = 0$.

Repeat

 Set $t_0 = t_0 - 1$, (take 1 time-step back)

 Generate $R^{(t_0+1)}$,

 For $k = 1, 2, \dots, K$, (for each state)

Set $X^{(t_0, k(t_0))} = k$, (start chain in that state)
 For $t = t_0, t_0 + 1, \dots, -1$, (propagate chain to time 0)
 Set $X^{(t+1, k(t_0))} = \varphi(X^{(t, k(t_0))}, R^{(t+1)})$.
 Until $X^{(0, 1(t_0))} = X^{(0, 2(t_0))} = \Lambda = X^{(0, K(t_0))}$. (check for coalescence at time 0)
 Return $X^{(0)}$.

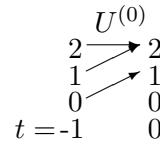
Example 160 Suppose that the Markov chain has the state space, $S = 0, 1, 2$, and a transition matrix:

$$Q = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.4 & 0.4 & 0.2 \\ 0.3 & 0.4 & 0.3 \end{pmatrix}$$

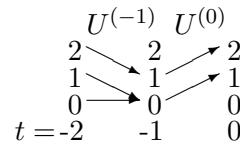
where the (i, j) -element is the conditional probability, $P(X^{(t+1)} = j | X^{(t)} = i)$. The matrix of conditional cumulative probabilities is

$$C = \begin{pmatrix} 0.6 & 0.9 & 1 \\ 0.4 & 0.8 & 1 \\ 0.3 & 0.7 & 1 \end{pmatrix}$$

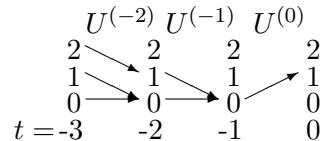
where the (i, j) -element is the probability, $P(X^{(t+1)} = j | X^{(t)} = i)$. Beginning at $t_0 = -1$, three chains are started at 0, 1 and 2. A uniform $(0, 1)$ random number, $U^{(0)}$, is generated (in this example, $R^{(0)} = U^{(0)}$) and used to propagate all three chains to time 0. Suppose that $U^{(0)} \in (0.8, 0.9)$. Then the three chains are updated as shown:



The chains have not coalesced at $t = 0$, so we need to move one time-step back to $t_0 = -2$, start three chains at 0, 1 and 2, generate a second uniform $(0, 1)$ random number, $U^{(-1)}$ and use it along with the previous $U^{(0)}$ to propagate the chains to time 0. Suppose that $U^{(-1)} \in (0.3, 0.4)$. The three chains then evolve as shown:



The chains have still not coalesced at $t = 0$, so we must move another time-step back to $t_0 = -3$ and start again, generating a third uniform $(0, 1)$ random number, $U^{(-2)}$. Suppose that $U^{(-2)} \in (0.3, 0.4)$. This is used with $U^{(-1)}$ and $U^{(0)}$ from before, giving the following transitions:



All three chains have now coalesced at $t = 0$ and so $X^{(0)} = 1$ is accepted as a sample value from the stationary distribution. The whole process is repeated to get another independent sample value. It is important to note that even though the chains have coalesced at $t = 1$, with the common value $X^{(-1)} = 0$; this value at the time of coalescence is not accepted as being from the stationary distribution. This is because the time of coalescence is a random time that depends only on the sequence of random numbers, $U^{(0)}, U^{(-1)}, \dots$; while the time at which a coalesced state has the required stationary distribution must be a fixed time. In the CFTP algorithm, this *fixed* time has been arbitrarily specified to be $t = 0$.

Example 161 To see that the state at the time of coalescence does not have the stationary distribution, suppose that the state space is $S = 1, 2$ and the transition matrix is:

$$Q = \begin{pmatrix} 0.5 & 0.5 \\ 1 & 0 \end{pmatrix}.$$

Since $Q(2, 1) = 1$, the two coupled chains must be in state 1 at the time of coalescence. However, the stationary distribution of this Markov chain is $f(1) = 2/3$ and $f(2) = 1/3$, and so the state at the time of coalescence cannot be from the stationary distribution.

Instead of taking a single step back when the two bounding chains fail to coalesce, any decreasing sequence of time-steps may be used. The “double-until-overshoot” choice of $t_0 = -2^0, -2^1, -2^2, \dots$ is optimal in the sense that it minimises the worst-case number of steps and almost minimises the expected number of steps for coalescence.

Exercise 162 Implement the CFTP algorithm for the Markov chain in Example 2.5.3 and use it to generate 1000 sample points from the stationary distribution of the chain. The stationary distribution can be shown to be:

x	0	1	2
$f(x)$	0.4789	0.3521	0.1690

Compare the relative frequencies of the generated sample with the true stationary probabilities.

Exercise 163 2.6.19 Consider a Markov chain with a state space $S = 0, 1, 2, 3$ and the transition matrix:

$$Q = \begin{pmatrix} 0.6 & 0.4 & 0 & 0 \\ 0.4 & 0.2 & 0.4 & 0 \\ 0.2 & 0.4 & 0 & 0.4 \\ 0 & 0.2 & 0.4 & 0.4 \end{pmatrix}.$$

Let $f = (f_0, f_1, f_2, f_3)$ be a row vector containing the stationary probabilities of the chain.

- (a) By solving $fQ = f$ and $f_0 + f_1 + f_2 + f_3 = 1$ simultaneously, show that the stationary distribution of the chain is $f = (14/35, 11/35, 6/35, 4/35)$.
- (b) Implement the “double-until-overshoot” version of the CFTP algorithm to generate from the stationary distribution, and use it to obtain 1000 sample points. Compare the relative frequencies of the generated sample with the true stationary probabilities.

Chapter 10

General Markov Chains

In this chapter we will study Markov chains on a general state space, i.e., state spaces that are not necessarily finite or countable.

10.1 Markov Chain Monte Carlo

This Section is under .

The methods described so far are generally unsuitable for complex multivariate or multi-dimensional distributions. One way to generate from such distributions is based on the simple idea that if a Markov chain is designed with a desired distribution as its stationary distribution, the states of the stationary chain will provide the required sample values. This approach is known as *Markov Chain Monte Carlo (MCMC)*.

A distribution with density/mass function f is a *stationary distribution* of a Markov chain if $X^{(t)} \sim f$ implies $X^{(t+1)} \sim f$.

To generate from a desired distribution using MCMC, the Markov chain must have the following properties:

- (a) The state space of the Markov chain must coincide with the support of the desired distribution.
- (b) *Irreducible*: The Markov chain must be free to move over the entire state space.
- (c) *Harris recurrent*: The Markov chain must not get stuck in any subset of the state space.
- (d) *Positive*: The Markov chain must converge to a unique stationary distribution regardless of the starting state.
- (e) *Aperiodic*: The Markov chain must not exhibit any deterministic pattern of movement.

Definition 86 (Ergodic Markov Chain) An *ergodic* Markov chain is one that is irreducible, positive, Harris recurrent and *aperiodic*.

MCMC is based on the observation that the state of an *ergodic* Markov chain will eventually converge to a stationary distribution, no matter which state the chain starts in. Thus, to obtain a sample from a desired distribution f , an ergodic Markov chain with f as its stationary distribution can be constructed and then run till it is stationary. The required sample values are given by the states of the stationary chain. Let $X^{(0)}, X^{(1)}, \dots$ represent a sequence of states for an ergodic Markov chain and suppose that it reaches its stationary distribution f after transition T . Then a

sample with distribution f is given by $\{X^{(t)} : t > T\}$. Note that unlike sample points given by the previous methods, which are independent, the sample points given by MCMC are *dependent* because they are states from a Markov chain.

Generic algorithms that allow an ergodic Markov chain with a specified stationary distribution to be constructed easily are available. Many of these algorithms can be regarded as variants of the *Metropolis-Hastings algorithm*.

Algorithm 12 Metropolis-Hastings Sampler

1: *input:*

- (1) shape of a target density $\tilde{f}(x) = \left(\int \tilde{f}(x)dx\right) f(x)$,
- (2) a *transition kernel*, $q(y|x)$.

2: *output:* a sequence of samples x_0, \dots from the Markov chain $\{X\}_{i \in \mathbb{Z}_+}$ with stationary distribution f

3: Choose initial state $X^{(0)}$ and *proposal distribution* g .

4: **repeat**

- 5: At iteration t ,
- 6: Generate $X \sim g(x|X^{(t-1)})$ and $U \sim U(0, 1)$,
- 7: Compute *acceptance probability*

$$\alpha = \min \left\{ 1, \frac{f(\tilde{X})g(X^{(t-1)}|\tilde{X})}{f(X^{(t-1)})g(\tilde{X})|X^{(t-1)}} \right\}, \quad (10.1)$$

8: **If** $U \leq \alpha$ **then** $X^{(t)} \leftarrow \tilde{X}$, **else** $X^{(t)} \leftarrow X^{(t-1)}$

9: **until** desired number of samples are obtained from $\{X\}_{i \in \mathbb{Z}_+}$

Definition 87 (Transition Kernel) The transitions of a Markov chain are governed by a conditional density/mass function known as the *transition kernel*, $q(y|x)$. For a discrete state space:

$$q(y|x) = P(X^{(t+1)} = y | X^{(t)} = x), \quad (10.2)$$

while for a continuous state space:

$$\int_A q(y|x)dy = P(X^{(t+1)} \in A | X^{(t)} = x), \quad (10.3)$$

for any subset A of the state space.

Proposition 88 If a Markov chain with transition kernel $q(y|x)$ satisfies the *detailed balance condition*:

$$q(x|y)f(y) = q(y|x)f(x), \quad (10.4)$$

where f is a density/mass function, then f is a stationary distribution of the chain.

Proof: Let S be the state space of the Markov chain and let $A \subset S$. Suppose that $X^{(t)} \sim f$. Then:

$$\begin{aligned}
P(X^{(t+1)} \in A) &= \int_S P(X^{(t+1)} \in A, X^{(t)} = y) dy \\
&= \int_S P(X^{(t+1)} \in A | X^{(t)} = y) f(y) dy \\
&= \int_S \int_A q(x|y) f(y) dx dy \\
&= \int_S \int_A q(y|x) f(x) dx dy \\
&= \int_A f(x) dx.
\end{aligned}$$

Therefore, $X^{(t+1)} \sim f$ and so f is a stationary distribution.

Proposition 89 In the Metropolis-Hastings algorithm, if the support of the proposal distribution is at least as large as the support of f , then the algorithm produces a Markov chain that has a stationary distribution f .

Proof: Let

$$\alpha(x, y) = \min \left\{ 1, \frac{f(y)g(x|y)}{f(x)g(y|x)} \right\}.$$

The transition kernel of the Metropolis-Hastings chain is:

$$q(y|x) = \alpha(x, y)g(y|x) + [1 - \beta(x)]\delta_x(y), \quad (10.5)$$

where:

$$\beta(x) = \int_S \alpha(x, y)g(y|x)dy, \quad (10.6)$$

and $\delta_x(\cdot)$ is the Dirac delta function at x .

It is enough to show that the Metropolis-Hastings chain satisfies the detailed balance condition with f , i.e. that $q(y|x)f(x) = q(x|y)f(y)$. This follows from:

$$\alpha(x, y)g(y|x)f(x) = \min\{f(x)g(u|x), f(y)g(x|y)\} = \alpha(y, x)g(x|y)f(y),$$

and:

$$[1 - \beta(x)]\delta_x(y)f(x) = [1 - \beta(y)]\delta_y(x)f(y).$$

Since f is used only to compute the acceptance probability, and appears both in the numerator and denominator, the algorithm is applicable even if f is not known completely but only up to a multiplicative constant. This is frequently the case in practice, where f is available as an un-normalised distribution. Some common variants of the Metropolis-Hastings algorithm include the *Metropolis sampler*, *independent Metropolis-Hastings sampler*, *single-component Metropolis-Hastings sampler* and *Gibbs sampler*.

The *Metropolis sampler* is obtained when a symmetric proposal distribution is used, i.e. $g(\tilde{X}|X^{(t-1)}) = g(X^{(t-1)}|\tilde{x})$. In this case, the acceptance probability simplifies to:

$$\alpha = \min \left\{ 1, \frac{f(\tilde{X})}{f(X^{(t-1)})} \right\} \quad (10.7)$$

A special case where $g(\tilde{X}|X^{(t-1)}) = g(|\tilde{X} - X^{(t-1)}|)$ is known as the *random walk Metropolis-Hastings (RWMH) sampler*.

Example 164 (rwmh_vonMises_uniform) The von Mises density with the location parameter $a \in [-\pi, \pi]$ and a scale parameter $b > 0$ is given by:

$$f(x) = \frac{e^{b \cos(x-a)}}{2\pi I_0(b)}, \quad (10.8)$$

for $x \in [-\pi, \pi]$, and where I_0 is the modified Bessel function of the first kind and order zero. Implement the RWMH sampler for generating from the von Mises density with $a = 0$ and $b = 3$ by using the $U(-1, 1)$ density to generate steps in the random walk, i.e. $g(\cdot|x) = U(x - 1, x + 1)$.

Matlab code: For $m = 1000$ iterations of the IMH sampler,

```

b = 3;
m = 1000;
x = ones(1,m); % allocate storage and initialise to 1
for k = 2:m
    y = x(k-1) + unifrnd(-1,1); % unifrnd(a,b) is the Matlab function for generating
    % U(a,b) random variables
    alpha = min(1,exp(b * (cos(y) - cos(x(k-1)))));
    if rand < alpha
        x(k) = y;
    else
        x(k) = x(k-1);
    end % if
end % for

```

When the proposal distribution is independent of $X^{(t-1)}$, the *independent Metropolis-Hastings (IMH) sampler* is obtained, with the acceptance probability given by:

$$\alpha = \min \left\{ 1, \frac{f(\tilde{X}g(X^{(t-1)}))}{f(X^{(t-1)})g(\tilde{X})} \right\} \quad (10.9)$$

This algorithm usually works well if g is close to f and has heavier tails than f .

Example 165 Consider the log-normal distribution whose density is:

$$f(x) = \frac{1}{x\sqrt{2\pi}} \exp \left\{ -\frac{(\log x)^2}{2} \right\}, \quad (10.10)$$

for $x \geq 0$. Use the IMH sampler with a gamma distribution proposal to generate from the log-normal distribution.

The gamma density with shape parameter $a > 0$ and scale parameter $b > 0$ is given by:

$$g(x) = \frac{1}{b^a \Gamma(a)} x^{a-1} e^{-x/b}, \quad (10.11)$$

for $x \geq 0$. Note that the IMH acceptance probability can be written as:

$$\alpha = \min \left\{ 1, \frac{f(\tilde{X})/g(\tilde{X})}{f(X^{(t-1)})/g(X^{(t-1)})} \right\},$$

which involves the ratio f/g in both the numerator and denominator, thus making multiplicative constants in f and g irrelevant and able to be discarded. In other words, it is enough to use:

$$\frac{\tilde{f}(x)}{\tilde{g}(x)} = \frac{\exp[-(\log x)^2/2]}{x^a e^{-x/b}} = \frac{1}{x^a} \exp\left[\frac{x}{b} - \frac{(\log x)^2}{2}\right]$$

to compute α .

Matlab code: For $m = 1000$ iterations of the IMH sampler using $gamma(1.5, 2.5)$ as proposal distribution:

```

m = 1000;
x = 0.5 * ones(1,m); % allocate storage and initialise to 0.5
for k = 2:m
    y = gamrnd(1.5,2.5); % gamrnd is the Matlab function for generating gamma
    % random variables
    alpha = (x(k-1) / y)^1.5 * \exp((y - x(k-1)) / 2.5 + (log(x(k-1))^2 - log(y)^2) / 2) ;
    alpha = min(1,alpha);
    if rand < alpha
        x(k) = y;
    else
        x(k) = x(k-1);
    end % if
end % for

```

In general, X may be multivariate. The idea behind the *single-component Metropolis-Hastings sampler* is to update X using a series of steps at each iteration, rather than a single step. To do this, X is partitioned into d parts: $X = (X_{[1]}, X_{[2]}, \dots, X_{[d]})$. Let $X_{[-j]}$ denote X with the j^{th} part omitted, and suppose that the conditional distributions, $f(x_{[j]}|x_{[-j]})$, are known.

10.1.1 Algorithm – Single-component Metropolis-Hastings sampler.

Choose initial state $X^{(0)}$ and proposal distribution g .

At iteration t ,

For $j = 1, 2, \dots, d$,

Generate $\tilde{X}_{[j]} \sim g(x_{[j]}|X_{[1]}^{(t)}, \dots, X_{[j-1]}^{(t)}, X_{[j]}^{(t-1)}, \dots, X_{[d]}^{(t-1)},)$ and $U_j \sim U(0, 1)$,
Compute

$$\alpha_j = \min\left\{1, \frac{f(\tilde{X}_{[j]}|X_{[1]}^{(t)}, \dots, X_{[j-1]}^{(t)}, X_{[j+1]}^{(t-1)}, \dots, X_{[d]}^{(t-1)},)}{f(X_{[j]}^{(t-1)}|X_{[1]}^{(t)}, \dots, X_{[j-1]}^{(t)}, X_{[j+1]}^{(t-1)}, \dots, X_{[d]}^{(t-1)},)} \cdot \frac{g(X_{[j]}^{(t-1)}|X_{[1]}^{(t)}, \dots, X_{[j-1]}^{(t)}, \tilde{X}_{[j]}, X_{[j+1]}^{(t-1)}, \dots, X_{[d]}^{(t-1)},)}{g(\tilde{X}_{[j]}|X_{[1]}^{(t)}, \dots, X_{[j-1]}^{(t)}, X_{[j]}^{(t-1)}, X_{[j+1]}^{(t-1)}, \dots, X_{[d]}^{(t-1)},)}\right\} \quad (10.12)$$

If $U_j \leq \alpha_j$

Set $X_{[j]}^{(t)} = \tilde{X}_{[j]}$.

(accept $\tilde{X}_{[j]}$ with probability α_j)

Else

Set $X_{[j]}^{(t)} = X_{[j]}^{(t-1)}$.

If it is possible to generate from the conditional distributions, $f(x_{[j]}|x - [-j])$, then by choosing them as the proposal distribution in the single-component Metropolis-Hastings sampler, the acceptance probabilities will always be one and the proposals will always be accepted. The resulting algorithm is known as the *Gibbs sampler*, which effectively generates from the conditional distributions.

10.1.2 Algorithm –Gibbs sampler.

Choose initial state $X^{(0)}$.

At iteration t ,

For $j = 1, 2, \dots, d$,
Generate $X_{[j]}^{(t)} \sim f(x_{[j]}|X_{[1]}^{(t)}, \dots, X_{[j-1]}^{(t)}, X_{[j+1]}^{(t-1)}, \dots, X_{[d]}^{(t-1)})$.

Example 166 (gibbs_example.m.) Consider the joint density:

$$f(x, y, z) \propto x^4 y^3 z^2 (1 - x - y - z)$$

where $x, y, z > 0$ and $x + y + z < 1$. Let $B(a, b)$ represent a beta distribution with parameters a and b . The conditional distributions for x, y and z are given by:

$$\begin{aligned} x|y, z &\sim (1 - y - z)q, q \sim B(5, 2), \\ y|x, z &\sim (1 - x - z)r, r \sim B(4, 2), \\ z|x, y &\sim (1 - x - y)s, s \sim B(3, 2). \end{aligned}$$

In other words, the conditional distribution of x , given y and z , is the same as the distribution of $(1 - y - z)q$ where q has a $B(5, 2)$ distribution, and so on. Implement a Gibbs sampler to generate samples from the joint density.

MATLAB code: For $m = 1000$ iterations of the Gibbs sampler:

```

m = 1000;
x = 0.3 * ones(1,m); % allocate storage and initialise to 0.3
y = x;
z = y;
for k = 2:m
    x(k) = (1 - y(k-1) - z(k-1)) * betarnd(5,2); % betarnd is the Matlab function for
    % generating beta random variables
    y(k) = (1 - x(k) - z(k-1)) * betarnd(4,2);
    z(k) = (1 - x(k) - y(k)) * betarnd(3,2);
end

```

Hybrid combinations of single-component Metropolis-Hastings and Gibbs sampling are possible, with some parts of X updated using Gibbs updates, and others (which cannot be generated from their conditional distributions) using Metropolis-Hastings updates.

In practice, a MCMC sampler is used to generate a long sequence of Markov chain states. After an initial *burn-in period*, the states are assumed to have the required stationary distribution, at least approximately. The difficulty in using MCMC is deciding how long the burn-in period should be.

10.2 Exercises

Exercise 167 Implement the IMH sampler in Example 2.4.9. Perform 10,000 iterations and plot the outputs sequentially. Comment on the appearance of the plot with regard to convergence to the target density. Plot the density histogram for the last 5000 iterations, and superimpose the target and proposal densities onto it.

Exercise 168 Implement the Gibbs sampler in Example 2.4.12. Perform 10,000 Gibbs iterations, and plot the sequential outputs for x , y and z . Comment on the appearance of the plots with regard to convergence to the target density. Obtain a three-dimensional scatter plot of the last 5000 sample points (use the `plot3` function).

Exercise 169 (a) Generate a sample of size 20 from the $N(0.06, 1)$ distribution.

(b) Treat the sample from Part (a) as observations from an $N(\theta, 1)$ distribution. Pretend that you do not know θ and wish to infer its value using the Bayesian approach. Denoting the sample by $z = z_1, \dots, z_{20}$, the posterior density of θ , given z , is given by Bayes' theorem as:

$$f(\theta|z) \propto f(z|\theta)f(\theta),$$

where $f(z|\theta)$ is the likelihood function, i.e.:

$$f(z|\theta) \propto \exp\left[-\frac{1}{2} \sum_{i=1}^n (z_i - \theta)^2\right],$$

and $f(\theta)$ is a prior density for θ . Choosing the Cauchy(0, 1) density as the prior density, the posterior density is therefore:

$$f(\theta|z) \propto \exp\left[-\frac{1}{2} \sum_{i=1}^n (z_i - \theta)^2\right] \frac{1}{1 + \theta^2}.$$

Implement the IMH sampler for generating from the posterior density, using the Cauchy(0, 1) as the proposal density.

(c) Use your IMH sampler to generate 1000 values from the posterior distribution. Use the generated values to estimate the mean of the posterior distribution and to obtain an approximate 95% probability interval for θ .

Exercise 170 Suppose x and y have conditional distributions that are exponential distributions restricted to the interval (0, 5). Then:

$$f(x|y) \propto ye^{-xy} \text{ and } f(y|x) \propto xe^{-xy}.$$

- (a) Implement the Gibbs sampler for generating sample points from the joint distribution $f(x, y)$.
- (b) Use your Gibbs sampler to generate 5000 sample points from $f(x, y)$. Use appropriate plots of the Markov chain outputs to assess convergence to the target distribution.
- (c) Obtain a two-dimensional scatter plot of your generated sample points.

Chapter 11

Fundamentals of Estimation

11.1 Introduction

Now that we have been introduced to two notions of convergence for RV sequences, we can begin to appreciate the basic limit theorems used in statistical inference. The problem of estimation is of fundamental importance in statistical inference and learning. We will formalise the general estimation problem here. There are two basic types of estimation. In point estimation we are interested in estimating a particular point of interest that is supposed to belong to a set of points. In (confidence) set estimation, we are interested in estimating a set with a particular form that has a specified probability of “trapping” the particular point of interest from a set of points. Here, a point should be interpreted as an element of a collection of elements from some space.

11.2 Point Estimation

Point estimation is any statistical methodology that provides one with a “**single best guess**” of some specific quantity of interest. Traditionally, we denote this **quantity of interest as θ^*** and **its point estimate as $\hat{\theta}$ or $\hat{\theta}_n$** . The subscript n in the point estimate $\hat{\theta}_n$ emphasises that our estimate is based on n observations or data points from a given statistical experiment to estimate θ^* . This quantity of interest, which is usually unknown, can be:

- an **integral** $\vartheta^* := \int_A h(x) dx \in \Theta$. If ϑ^* is finite, then $\Theta = \mathbb{R}$, or
- a **parameter** θ^* which is an element of the **parameter space** Θ , denoted $\theta^* \in \Theta$,
- a **distribution function (DF)** $F^* \in \mathbb{F} :=$ the set of all DFs
- a **density function (pdf)** $f \in \{$ “not too wiggly Sobolev functions” $\}$, or
- a **regression function** $g^* \in \mathbb{G}$, where \mathbb{G} is a class of regression functions in a regression experiment with model: $Y = g^*(X) + \epsilon$, such that $\mathbf{E}(\epsilon) = 0$, from pairs of observations $\{(X_i, Y_i)\}_{i=1}^n$, or
- a **classifier** $g^* \in \mathbb{G}$, i.e. a regression experiment with discrete $Y = g^*(X) + \epsilon$, or
- a **prediction** in a regression experiment, i.e. when you want to estimate Y_i given X_i .

Recall that a statistic is an RV $T(X)$ that maps every data point x in the data space \mathbb{X} with $T(x) = t$ in its range \mathbb{T} , i.e. $T(x) : \mathbb{X} \rightarrow \mathbb{T}$ (Definition 30). Next, we look at a specific class of statistics whose range is the parameter space Θ .

Definition 90 (Point Estimator) A **point estimator** $\hat{\Theta}$ of some **fixed and possibly unknown** $\theta^* \in \Theta$ is a statistic that associates each data point $x \in \mathbb{X}$ with an estimate $\hat{\Theta}(x) = \hat{\theta} \in \Theta$,

$$\boxed{\hat{\Theta} := \hat{\Theta}(x) = \hat{\theta} : \mathbb{X} \rightarrow \Theta} .$$

If our data point $x := (x_1, x_2, \dots, x_n)$ is an n -vector or a point in the n -dimensional real space, i.e. $x := (x_1, x_2, \dots, x_n) \in \mathbb{X}_n \subset \mathbb{R}^n$, then we emphasise the dimension n in our point estimator $\hat{\Theta}_n$ of $\theta^* \in \Theta$.

$$\boxed{\hat{\Theta}_n := \hat{\Theta}_n(x := (x_1, x_2, \dots, x_n)) = \hat{\theta}_n : \mathbb{X}_n \rightarrow \Theta, \quad \mathbb{X}_n \subset \mathbb{R}^n} .$$

The typical situation for us involves point estimation of $\theta^* \in \Theta$ on the basis of one realisation $x \in \mathbb{X}_n \subset \mathbb{R}^n$ of an independent and identically distributed (IID) random vector $X = (X_1, X_2, \dots, X_n)$, such that $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} X_1$ and the DF of X_1 is $F(x_1; \theta^*)$, i.e. the distribution of the IID RVs, X_1, X_2, \dots, X_n , is parameterised by $\theta^* \in \Theta$.

Example 171 (Coin Tossing Experiment) ($X_1, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta^*)$) I tossed a coin that has an unknown probability θ^* of landing Heads independently and identically 10 times in a row. Four of my outcomes were Heads and the remaining six were Tails, with the actual sequence of Bernoulli outcomes (Heads $\rightarrow 1$ and Tails $\rightarrow 0$) being $(1, 0, 0, 0, 1, 1, 0, 0, 1, 0)$. I would like to estimate the probability $\theta^* \in \Theta = [0, 1]$ of observing Heads using the natural estimator $\hat{\Theta}_n((X_1, X_2, \dots, X_n))$ of θ^* :

$$\hat{\Theta}_n((X_1, X_2, \dots, X_n)) := \hat{\Theta}_n = \frac{1}{n} \sum_{i=1}^n X_i =: \bar{X}_n$$

For the coin tossing experiment I just performed ($n = 10$ times), the point estimate of the unknown θ^* is:

$$\begin{aligned} \hat{\theta}_{10} &= \hat{\Theta}_{10}((x_1, x_2, \dots, x_{10})) = \hat{\Theta}_{10}((1, 0, 0, 0, 1, 1, 0, 0, 1, 0)) \\ &= \frac{1 + 0 + 0 + 0 + 1 + 1 + 0 + 0 + 1 + 0}{10} = \frac{4}{10} = 0.40 . \end{aligned}$$

Labwork 172 (Bernoulli(38/75) Computer Experiment) Simulate one thousand IID samples from a $\text{Bernoulli}(\theta^* = 38/75)$ RV and store this data in an array called **Samples**. Use your student ID to initialise the fundamental sampler. Now, pretend that you don't know the true θ^* and estimate θ^* using our estimator $\hat{\Theta}_n = \bar{X}_n$ from the data array **Samples** for each sample size $n = 1, 2, \dots, 1000$. Plot the one thousand estimates $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_{1000}$ as a function of the corresponding sample size. Report your observations regarding the behaviour of our estimator as the sample size increases.

11.3 Some Properties of Point Estimators

Given that an estimator is merely a function from the data space to the parameter space, we need choose only the best estimators available. Recall that a point estimator $\hat{\Theta}_n$, being a statistic or an RV of the data has a probability distribution over its range Θ . This distribution over Θ is called the **sampling distribution** of $\hat{\Theta}_n$. Note that the sampling distribution not only depends on the statistic $\hat{\Theta}_n := \hat{\Theta}_n(X_1, X_2, \dots, X_n)$ but also on θ^* which in turn determines the distribution of the IID data vector (X_1, X_2, \dots, X_n) . The following definitions are useful for selecting better estimators from some lot of them.

Definition 91 (Bias of a Point Estimator) The bias_n of an estimator $\widehat{\Theta}_n$ of $\theta^* \in \Theta$ is:

$$\text{bias}_n = \text{bias}_n(\widehat{\Theta}_n) := \mathbf{E}_{\theta^*}(\widehat{\Theta}_n) - \theta^* = \int_{\mathbb{X}_n} \widehat{\Theta}_n(x) dF(x; \theta^*) - \theta^*. \quad (11.1)$$

We say that the estimator $\widehat{\Theta}_n$ is **unbiased** if $\text{bias}_n(\widehat{\Theta}_n) = 0$ or if $\mathbf{E}_{\theta^*}(\widehat{\Theta}_n) = \theta^*$ for every n . If $\lim_{n \rightarrow \infty} \text{bias}_n(\widehat{\Theta}_n) = 0$, we say that the estimator is **asymptotically unbiased**.

Since the expectation of the sampling distribution of the point estimator $\widehat{\Theta}_n$ depends on the unknown θ^* , we emphasise the θ^* -dependence by $\mathbf{E}_{\theta^*}(\widehat{\Theta}_n)$.

Example 173 (Bias of our Estimator of θ^*) Consider the sample mean estimator $\widehat{\Theta}_n := \overline{X}_n$ of θ^* , from $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta^*)$. That is, we take the sample mean of the n IID Bernoulli(θ^*) trials to be our point estimator of $\theta^* \in [0, 1]$. Then, **this estimator is unbiased** since:

$$\mathbf{E}_{\theta^*}(\widehat{\Theta}_n) = \mathbf{E}_{\theta^*} \left(n^{-1} \sum_{i=1}^n X_i \right) = n^{-1} \mathbf{E}_{\theta^*} \left(\sum_{i=1}^n X_i \right) = n^{-1} \sum_{i=1}^n \mathbf{E}_{\theta^*}(X_i) = n^{-1} n \theta^* = \theta^*.$$

Definition 92 (Standard Error of a Point Estimator) The standard deviation of the point estimator $\widehat{\Theta}_n$ of $\theta^* \in \Theta$ is called the **standard error**:

$$\text{se}_n = \text{se}_n(\widehat{\Theta}_n) = \sqrt{\mathbf{V}_{\theta^*}(\widehat{\Theta}_n)} = \sqrt{\int_{\mathbb{X}_n} (\widehat{\Theta}_n(x) - \mathbf{E}_{\theta^*}(\widehat{\Theta}_n))^2 dF(x; \theta^*)}. \quad (11.2)$$

Since the variance of the sampling distribution of the point estimator $\widehat{\Theta}_n$ depends on the fixed and possibly unknown θ^* , as emphasised by \mathbf{V}_{θ^*} in (11.2), the se_n is also a possibly unknown quantity and may itself be estimated from the data. The estimated standard error, denoted by $\widehat{\text{se}}_n$, is calculated by replacing $\mathbf{V}_{\theta^*}(\widehat{\Theta}_n)$ in (11.2) with its appropriate estimate.

Example 174 (Standard Error of our Estimator of θ^*) Consider the sample mean estimator $\widehat{\Theta}_n := \overline{X}_n$ of θ^* , from $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta^*)$. Observe that the statistic:

$$T_n((X_1, X_2, \dots, X_n)) := n \widehat{\Theta}_n((X_1, X_2, \dots, X_n)) = \sum_{i=1}^n X_i$$

is the Binomial(n, θ^*) RV. The standard error se_n of this estimator is:

$$\text{se}_n = \sqrt{\mathbf{V}_{\theta^*}(\widehat{\Theta}_n)} = \sqrt{\mathbf{V}_{\theta^*} \left(\sum_{i=1}^n \frac{X_i}{n} \right)} = \sqrt{\left(\sum_{i=1}^n \frac{1}{n^2} \mathbf{V}_{\theta^*}(X_i) \right)} = \sqrt{\frac{n}{n^2} \mathbf{V}_{\theta^*}(X_i)} = \sqrt{\theta^*(1 - \theta^*)/n}.$$

Another reasonable property of an estimator is that it converge to the “true” parameter θ^* – here “true” means the supposedly fixed and possibly unknown θ^* , as we gather more and more IID data from a θ^* -specified DF $F(x; \theta^*)$. This property is stated precisely next.

Definition 93 (Asymptotic Consistency of a Point Estimator) A point estimator $\widehat{\Theta}_n$ of $\theta^* \in \Theta$ is said to be **asymptotically consistent** if:

$$\widehat{\Theta}_n \xrightarrow{P} \theta^* \quad \text{i.e., for any real } \epsilon > 0, \quad \lim_{n \rightarrow \infty} \mathbf{P}(|\widehat{\Theta}_n - \theta^*| > \epsilon) = 0.$$

Definition 94 (Mean Squared Error (MSE) of a Point Estimator) Often, the quality of a point estimator $\hat{\Theta}_n$ of $\theta^* \in \Theta$ is assessed by the **mean squared error** or MSE_n defined by:

$$\boxed{\text{MSE}_n = \text{MSE}_n(\hat{\Theta}_n) := \mathbf{E}_{\theta^*} ((\hat{\Theta}_n - \theta^*)^2) = \int_{\mathbb{X}} (\hat{\Theta}_n(x) - \theta^*)^2 dF(x; \theta^*)} . \quad (11.3)$$

The following proposition shows a simple relationship between the mean square error, bias and variance of an estimator $\hat{\Theta}_n$ of θ^* .

Proposition 95 (The $\sqrt{\text{MSE}_n}$: se_n – Sided Right Triangle of an Estimator) Let $\hat{\Theta}_n$ be an estimator of $\theta^* \in \Theta$. Then:

$$\boxed{\text{MSE}_n(\hat{\Theta}_n) = (\text{se}_n(\hat{\Theta}_n))^2 + (\text{bias}_n(\hat{\Theta}_n))^2} . \quad (11.4)$$

Proof:

$$\begin{aligned} & LHS \\ &= \text{MSE}_n(\hat{\Theta}_n) \\ &:= \mathbf{E}_{\theta^*} ((\hat{\Theta}_n - \theta^*)^2), \quad \text{by definition of } \text{MSE}_n \text{ (11.3)} \\ &= \mathbf{E}_{\theta^*} \left(\left(\underbrace{\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n)}_A \right)^2 + \underbrace{(\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*)}_B \right)^2, \quad \text{by subtracting and adding the constant } \mathbf{E}_{\theta^*}(\hat{\Theta}_n) \\ &= \mathbf{E}_{\theta^*} \left(\underbrace{(\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n))^2}_{A^2} + \underbrace{2(\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n))(\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*)}_{2AB} + \underbrace{(\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*)^2}_{B^2} \right), \quad \because (A+B)^2 = A^2 + 2AB + B^2 \\ &= \mathbf{E}_{\theta^*} \left((\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n))^2 \right) + \mathbf{E}_{\theta^*} \left(2(\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n))(\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*) \right) + \mathbf{E}_{\theta^*} \left((\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*)^2 \right), \\ &= \mathbf{E}_{\theta^*} \left((\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n))^2 \right) + \underbrace{2(\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*)}_{C} \underbrace{\mathbf{E}_{\theta^*}((\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n)))}_{D} + \mathbf{E}_{\theta^*} \left((\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*)^2 \right), \quad \because C \text{ is constant} \\ &= \mathbf{E}_{\theta^*} \left((\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n))^2 \right) + 0 + \mathbf{E}_{\theta^*} \left((\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*)^2 \right), \quad \because D := \mathbf{E}_{\theta^*}((\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n))) = \mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \mathbf{E}_{\theta^*}(\hat{\Theta}_n) = 0 \\ &= \mathbf{V}_{\theta^*}(\hat{\Theta}_n) + \mathbf{E}_{\theta^*} \left((\mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*)^2 \right), \quad \because \mathbf{V}_{\theta^*}(\hat{\Theta}_n) := \mathbf{E}_{\theta^*} \left((\hat{\Theta}_n - \mathbf{E}_{\theta^*}(\hat{\Theta}_n))^2 \right), \text{ by definition of variance} \\ &= \left(\sqrt{\mathbf{V}_{\theta^*}(\hat{\Theta}_n)} \right)^2 + \mathbf{E}_{\theta^*} \left((\text{bias}_n(\hat{\Theta}_n))^2 \right), \quad \because \text{bias}_n(\hat{\Theta}_n) = \mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^*, \text{ by definition of } \text{bias}_n \text{ of an estimator } \hat{\Theta}_n \\ &= (\text{se}_n(\hat{\Theta}_n))^2 + \mathbf{E}_{\theta^*} \left((\text{bias}_n(\hat{\Theta}_n))^2 \right), \quad \because \text{se}_n(\hat{\Theta}_n) := \sqrt{\mathbf{V}_{\theta^*}(\hat{\Theta}_n)}, \text{ by definition (11.2)} \\ &= (\text{se}_n(\hat{\Theta}_n))^2 + (\text{bias}_n(\hat{\Theta}_n))^2, \quad \because \text{bias}_n(\hat{\Theta}_n) = \mathbf{E}_{\theta^*}(\hat{\Theta}_n) - \theta^* \text{ and } (\text{bias}_n(\hat{\Theta}_n))^2 \text{ are constants.} \\ &= RHS \end{aligned}$$

Proposition 96 (Asymptotic consistency of a point estimator) Let $\hat{\Theta}_n$ be an estimator of $\theta^* \in \Theta$. Then, if $\text{bias}_n(\hat{\Theta}_n) \rightarrow 0$ and $\text{se}_n(\hat{\Theta}_n) \rightarrow 0$ as $n \rightarrow \infty$, the estimator $\hat{\Theta}_n$ is asymptotically consistent:

$$\hat{\Theta}_n \xrightarrow{P} \theta^* .$$

Proof: If $\text{bias}_n(\hat{\Theta}_n) \rightarrow 0$ and $\text{se}_n(\hat{\Theta}_n) \rightarrow 0$, then by (11.4), $\text{MSE}_n(\hat{\Theta}_n) \rightarrow 0$, i.e. that $\mathbf{E}_{\theta^*}((\hat{\Theta}_n - \theta^*)^2) \rightarrow 0$. This type of convergence of the RV $\hat{\Theta}_n$ to the *Point Mass(θ^*)* RV as $n \rightarrow \infty$ is called convergence in **quadratic mean** or **convergence in BL_2** and denoted by $\hat{\Theta}_n \xrightarrow{qm} \theta^*$. Convergence in quadratic mean is a stronger notion of convergence than convergence in probability, in the sense that

$$\mathbf{E}_{\theta^*}((\hat{\Theta}_n - \theta^*)^2) \rightarrow 0 \quad \text{or} \quad \hat{\Theta}_n \xrightarrow{qm} \theta^* \implies \hat{\Theta}_n \xrightarrow{P} \theta^* .$$

Thus, if we prove the above implication we are done with the proof of our proposition. To show that convergence in quadratic mean implies convergence in probability for general sequence of RVs X_n converging to an RV X , we first assume that $X_n \xrightarrow{qm} X$. Now, fix any $\epsilon > 0$. Then by Markov's inequality (8.1),

$$\mathbf{P}(|X_n - X| > \epsilon) = \mathbf{P}(|X_n - X|^2 > \epsilon^2) \leq \frac{\mathbf{E}(|X_n - X|^2)}{\epsilon^2} \rightarrow 0,$$

and we have shown that the definition of convergence in probability holds provided convergence in quadratic mean holds.

We want our estimator to be unbiased with small standard errors as the sample size n gets large. The **point estimator** $\hat{\Theta}_n$ will then produce a **point estimate** $\hat{\theta}_n$:

$$\hat{\Theta}_n((x_1, x_2, \dots, x_n)) = \hat{\theta}_n \in \Theta,$$

on the basis of the **observed data** (x_1, x_2, \dots, x_n) , that is close to the **true parameter** $\theta^* \in \Theta$.

Example 175 (Asymptotic consistency of our Estimator of θ^*) Consider the sample mean estimator $\hat{\Theta}_n := \bar{X}_n$ of θ^* , from $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta^*)$. Since $\text{bias}_n(\hat{\Theta}_n) = 0$ for any n and $\text{se}_n = \sqrt{\theta^*(1-\theta^*)/n} \rightarrow 0$, as $n \rightarrow \infty$, by Proposition 96, $\hat{\Theta}_n \xrightarrow{P} \theta^*$. That is $\hat{\Theta}_n$ is an **asymptotically consistent estimator** of θ^* .

11.4 Confidence Set Estimation

As we saw in Section 11.2, the point estimate $\hat{\theta}_n$ is a “single best guess” of the fixed and possibly unknown parameter $\theta^* \in \Theta$. However, if we wanted to make a statement about our confidence in an estimation procedure, then one possibility is to produce subsets from the parameter space Θ called **confidence sets** that “engulf” θ^* with a probability of at least $1 - \alpha$.

Formally, an **$B1 - \alpha$ confidence interval** for the parameter $\theta^* \in \Theta \subset \mathbb{R}$, based on n observations or data points X_1, X_2, \dots, X_n , is an interval C_n that is a function of the data:

$$C_n := [\underline{C}_n, \bar{C}_n] = [\underline{C}_n(X_1, X_2, \dots, X_n), \bar{C}_n(X_1, X_2, \dots, X_n)],$$

such that:

$$\mathbf{P}_{\theta^*}(\theta^* \in C_n := [\underline{C}_n, \bar{C}_n]) \geq 1 - \alpha.$$

Note that the confidence interval $C_n := [\underline{C}_n, \bar{C}_n]$ is a two-dimensional RV or a random vector in \mathbb{R}^2 that depends on the two statistics $\underline{C}_n(X_1, X_2, \dots, X_n)$ and $\bar{C}_n(X_1, X_2, \dots, X_n)$, as well as θ^* , which in turn determines the distribution of the data (X_1, X_2, \dots, X_n) . In words, C_n engulfs the true parameter $\theta^* \in \Theta$ with a probability of at least $1 - \alpha$. We call $1 - \alpha$ as the **coverage** of the confidence interval C_n .

Formally, a $1 - \alpha$ **confidence set** C_n for a vector-valued $\theta^* \in \Theta \subset \mathbb{R}^k$ is any subset of Θ such that $\mathbf{P}_{\theta^*}(\theta^* \in C_n) \geq 1 - \alpha$. The typical forms taken by C_n are k -dimensional boxes or hyper-cuboids, hyper-ellipsoids and subsets defined by inequalities involving level sets of some estimator of θ^* .

Typically, we take $\alpha = 0.05$ because we are interested in the $1 - \alpha = 0.95$ or 95% confidence interval/set $C_n \subset \Theta$ of $\theta^* \in \Theta$ from an estimator $\hat{\Theta}_n$ of θ^* .

The following property of an estimator makes it easy to obtain confidence intervals.

Definition 97 (Asymptotic Normality of Estimators) An estimator $\hat{\Theta}_n$ of a fixed and possibly unknown parameter $\theta^* \in \Theta$ is **asymptotically normal** if:

$$\frac{\hat{\Theta}_n - \theta^*}{\text{se}_n} \rightsquigarrow \text{Normal}(0, 1). \quad (11.5)$$

That is, $\hat{\Theta}_n \rightsquigarrow \text{Normal}(\theta^*, \text{se}_n^2)$. By a further estimation of $\text{se}_n := \sqrt{\mathbf{V}_{\theta^*}(\hat{\Theta}_n)}$ by $\widehat{\text{se}}_n$, we can see that $\hat{\Theta}_n \rightsquigarrow \text{Normal}(\theta^*, \widehat{\text{se}}_n^2)$ on the basis of (8.12).

Proposition 98 (Normal-based Asymptotic Confidence Interval) Suppose an estimator $\widehat{\Theta}_n$ of parameter $\theta^* \in \Theta \subset \mathbb{R}$ is asymptotically normal:

$$\widehat{\Theta}_n \rightsquigarrow \text{Normal}(\theta^*, \widehat{s}\widehat{e}_n^2) .$$

Let the RV $Z \sim \text{Normal}(0, 1)$ have DF Φ and inverse DF Φ^{-1} . Let:

$$z_{\alpha/2} = \Phi^{-1}(1 - (\alpha/2)), \quad \text{that is, } \mathbf{P}(Z > z_{\alpha/2}) = \alpha/2 \text{ and } \mathbf{P}(-z_{\alpha/2} < Z < z_{\alpha/2}) = 1 - \alpha .$$

Then:

$$\mathbf{P}_{\theta^*}(\theta^* \in C_n) = \mathbf{P}\left(\theta^* \in [\widehat{\Theta}_n - z_{\alpha/2}\widehat{s}\widehat{e}_n, \widehat{\Theta}_n + z_{\alpha/2}\widehat{s}\widehat{e}_n]\right) \rightarrow 1 - \alpha .$$

Therefore:

$$C_n := [\underline{C}_n, \overline{C}_n] = [\widehat{\Theta}_n - z_{\alpha/2}\widehat{s}\widehat{e}_n, \widehat{\Theta}_n + z_{\alpha/2}\widehat{s}\widehat{e}_n]$$

is the $1 - \alpha$ Normal-based asymptotic confidence interval that relies on the asymptotic normality of the estimator $\widehat{\Theta}_n$ of $\theta^* \in \Theta \subset \mathbb{R}$.

Proof: Define the centralised and scaled estimator as $Z_n := (\widehat{\Theta}_n - \theta^*)/\widehat{s}\widehat{e}_n$. By assumption, $Z_n \rightsquigarrow Z \sim \text{Normal}(0, 1)$. Therefore,

$$\begin{aligned} \mathbf{P}_{\theta^*}(\theta^* \in C_n) &= \mathbf{P}_{\theta^*}\left(\theta^* \in [\widehat{\Theta}_n - z_{\alpha/2}\widehat{s}\widehat{e}_n, \widehat{\Theta}_n + z_{\alpha/2}\widehat{s}\widehat{e}_n]\right) \\ &= \mathbf{P}_{\theta^*}\left(\widehat{\Theta}_n - z_{\alpha/2}\widehat{s}\widehat{e}_n \leq \theta^* \leq \widehat{\Theta}_n + z_{\alpha/2}\widehat{s}\widehat{e}_n\right) \\ &= \mathbf{P}_{\theta^*}\left(-z_{\alpha/2}\widehat{s}\widehat{e}_n \leq \widehat{\Theta}_n - \theta^* \leq z_{\alpha/2}\widehat{s}\widehat{e}_n\right) \\ &= \mathbf{P}_{\theta^*}\left(-z_{\alpha/2} \leq \frac{\widehat{\Theta}_n - \theta^*}{\widehat{s}\widehat{e}_n} \leq z_{\alpha/2}\right) \\ &\rightarrow \mathbf{P}_{\theta^*}(-z_{\alpha/2} \leq Z \leq z_{\alpha/2}) \\ &= 1 - \alpha \end{aligned}$$

Figure 11.1: Density and Confidence Interval of the Asymptotically Normal Point Estimator

For 95% confidence intervals, $\alpha = 0.05$ and $z_{\alpha/2} = z_{0.025} = 1.96 \approx 2$. This leads to the **approximate B95% confidence interval** of $\widehat{\theta}_n \pm 2\widehat{s}\widehat{e}_n$, where $\widehat{\theta}_n = \widehat{\Theta}_n(x_1, x_2, \dots, x_n)$ and x_1, x_2, \dots, x_n are the data or observations of the RVs X_1, X_2, \dots, X_n .

Example 176 (Confidence interval for θ^* from n Bernoulli(θ^*) trials) Let $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta^*)$ for some fixed but unknown parameter $\theta^* \in \Theta = [0, 1]$. Consider the following point estimator of θ^* :

$$\widehat{\Theta}_n((X_1, X_2, \dots, X_n)) = n^{-1} \sum_{i=1}^n X_i .$$

That is, we take the sample mean of the n IID Bernoulli(θ^*) trials to be our point estimator of $\theta^* \in [0, 1]$. Then, we already saw that **this estimator is unbiased**

We already saw that the standard error se_n of this estimator is:

$$\text{se}_n = \sqrt{\theta^*(1 - \theta^*)/n} .$$

Since θ^* is unknown, we obtain the estimated standard error $\widehat{\text{se}}_n$ from the point estimate $\widehat{\theta}_n$ of θ^* on the basis of n observed data points $x = (x_1, x_2, \dots, x_n)$ of the experiment:

$$\widehat{\text{se}}_n = \sqrt{\widehat{\theta}_n(1 - \widehat{\theta}_n)/n}, \quad \text{where, } \widehat{\theta}_n = \widehat{\Theta}_n((x_1, x_2, \dots, x_n)) = n^{-1} \sum_{i=1}^n x_i .$$

By the central limit theorem, $\widehat{\Theta}_n \rightsquigarrow \text{Normal}(\theta^*, \widehat{\text{se}}_n)$, i.e. $\widehat{\Theta}_n$ is asymptotically normal. Therefore, an asymptotically (for large sample size n) approximate $1 - \alpha$ normal-based confidence interval is:

$$\widehat{\theta}_n \pm z_{\alpha/2} \widehat{\text{se}}_n = \widehat{\theta}_n \pm z_{\alpha/2} \sqrt{\frac{\widehat{\theta}_n(1 - \widehat{\theta}_n)}{n}} := \left[\widehat{\theta}_n - z_{\alpha/2} \sqrt{\frac{\widehat{\theta}_n(1 - \widehat{\theta}_n)}{n}}, \widehat{\theta}_n + z_{\alpha/2} \sqrt{\frac{\widehat{\theta}_n(1 - \widehat{\theta}_n)}{n}} \right]$$

We also saw that $\widehat{\Theta}_n$ is an **asymptotically consistent estimator** of θ^* due to Proposition 96.

The confidence Interval for the coin tossing experiment in Example 171 with the observed sequence of Bernoulli outcomes (Heads $\rightarrow 1$ and Tails $\rightarrow 0$) being $(1, 0, 0, 0, 1, 1, 0, 0, 1, 0)$. We estimated the probability θ^* of observing Heads with the **unbiased, asymptotically consistent estimator** $\widehat{\Theta}_n((X_1, X_2, \dots, X_n)) = n^{-1} \sum_{i=1}^n X_i$ of θ^* . The point estimate of θ^* was:

$$\begin{aligned} \widehat{\theta}_{10} = \widehat{\Theta}_{10}((x_1, x_2, \dots, x_{10})) &= \widehat{\Theta}_{10}((1, 0, 0, 0, 1, 1, 0, 0, 1, 0)) \\ &= \frac{1 + 0 + 0 + 0 + 1 + 1 + 0 + 0 + 1 + 0}{10} = \frac{4}{10} = 0.40 . \end{aligned}$$

The normal-based confidence interval for θ^* may not be a valid approximation here with just $n = 10$ samples. Nevertheless, we will compute a 95% normal-based confidence interval:

$$C_{10} = 0.40 \pm 1.96 \sqrt{\frac{0.40(1 - 0.40)}{10}} = 0.40 \pm 0.3036 = [0.0964, 0.7036]$$

with a width of 0.6072. When I increased the sample size n of the experiment from 10 to 100 by tossing the same coin another 90 times, I discovered that a total of 57 trials landed as Heads. Thus my point estimate and confidence interval for θ^* are:

$$\widehat{\theta}_{100} = \frac{57}{100} = 0.57 \quad \text{and} \quad C_{100} = 0.57 \pm 1.96 \sqrt{\frac{0.57(1 - 0.57)}{100}} = 0.57 \pm 0.0495 = [0.5205, 0.6195]$$

with a much smaller width of 0.0990. Thus our confidence interval shrank considerably from a width of 0.6072 after an additional 90 Bernoulli trials. Thus, we can make the width of the confidence interval as small as we want by making the number of observations or sample size n as large as we can.

11.5 Likelihood

We take a look at one of the most fundamental concepts in Statistics.

Definition 99 (Likelihood Function) Suppose X_1, X_2, \dots, X_n have joint density $f(x_1, x_2, \dots, x_n; \theta)$ specified by parameter $\theta \in \Theta$. Let the observed data be x_1, x_2, \dots, x_n .

The **likelihood** function given by $L_n(\theta)$ is proportional to $f(x_1, x_2, \dots, x_n; \theta)$, the joint probability of the data, with the exception that we see it as a function of the parameter:

$$L_n(\theta) := L_n(x_1, x_2, \dots, x_n; \theta) = f(x_1, x_2, \dots, x_n; \theta) . \quad (11.6)$$

The likelihood function has a simple product structure when the observations are independently and identically distributed:

$$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} f(x; \theta) \implies \boxed{L_n(\theta) := L_n(x_1, x_2, \dots, x_n; \theta) = f(x_1, x_2, \dots, x_n; \theta) := \prod_{i=1}^n f(x_i; \theta)} . \quad (11.7)$$

The **log-likelihood** function is defined by:

$$\boxed{\ell_n(\theta) := \log(L_n(\theta))} \quad (11.8)$$

Example 177 (Likelihood of the IID Bernoulli(θ^*) experiment) Consider our IID Bernoulli experiment:

$$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Bernoulli}(\theta^*), \text{ with PDF } f(x; \theta) = \theta^x(1 - \theta)^{1-x}\mathbf{1}_{\{0,1\}}(x) .$$

Let us understand the likelihood function for one observation first. There are two possibilities for the first observation.

If we only have one observation and it happens to be $x_1 = 1$, then our likelihood function is:

$$L_1(\theta) = L_1(x_1; \theta) = f(x_1; \theta) = \theta^1(1 - \theta)^{1-1}\mathbf{1}_{\{0,1\}}(1) = \theta(1 - \theta)^01 = \theta$$

If we only have one observation and it happens to be $x_1 = 0$, then our likelihood function is:

$$L_1(\theta) = L_1(x_1; \theta) = f(x_1; \theta) = \theta^0(1 - \theta)^{1-0}\mathbf{1}_{\{0,1\}}(0) = 1(1 - \theta)^11 = 1 - \theta$$

If we have n observations (x_1, x_2, \dots, x_n) , i.e. a vertex point of the unit hyper-cube $\{0, 1\}^n$, then our likelihood function is obtained by multiplying the densities:

$$\begin{aligned} L_n(\theta) &:= L_n(x_1, x_2, \dots, x_n; \theta) = f(x_1, x_2, \dots, x_n; \theta) \\ &= f(x_1; \theta)f(x_2; \theta) \cdots f(x_n; \theta) := \prod_{i=1}^n f(x_i; \theta) \\ &= \theta^{\sum_{i=1}^n x_i}(1 - \theta)^{n - \sum_{i=1}^n x_i} := \theta^{t_n}(1 - \theta)^{n - t_n} \end{aligned}$$

In the last step, we have formally defined the following statistic of the data:

$$T_n(X_1, X_2, \dots, X_n) = \sum_{i=1}^n X_i : \mathbb{X}_n \rightarrow \mathbb{T}_n$$

with the corresponding realisation $t_n := T_n(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i \in \mathbb{T}_n$.

Figure 11.2: Data Spaces $\mathbb{X}_1 = \{0, 1\}$, $\mathbb{X}_2 = \{0, 1\}^2$ and $\mathbb{X}_3 = \{0, 1\}^3$ for one, two and three IID Bernoulli trials, respectively and the corresponding likelihood functions.

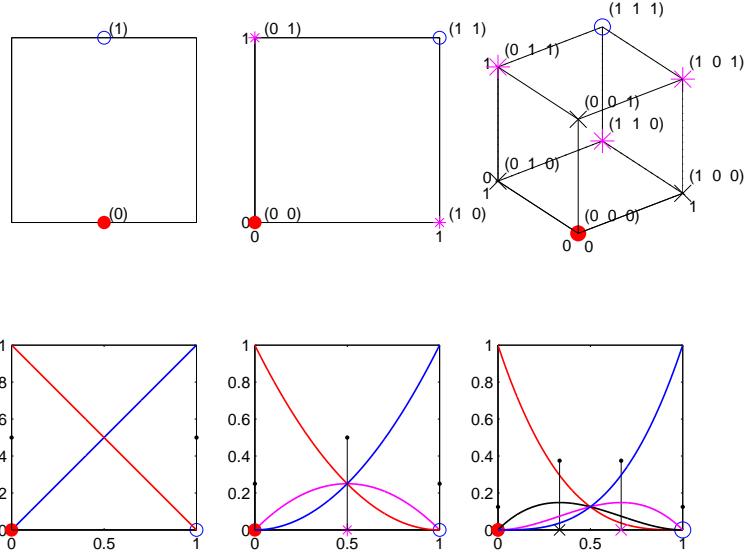
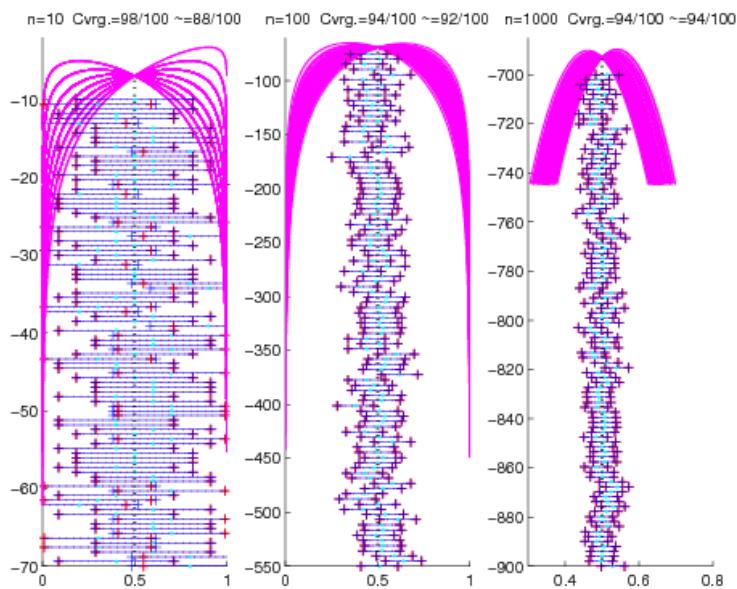


Figure 11.3: 100 realisations of $C_{10}, C_{100}, C_{1000}$ based on samples of size $n = 10, 100$ and 1000 drawn from the Bernoulli($\theta^* = 0.5$) RV as per Labwork 254. The MLE $\hat{\theta}_n$ (cyan dot) and the log-likelihood function (magenta curve) for each of the 100 replications of the experiment for each sample size n are depicted. The approximate normal-based 95% confidence intervals with blue boundaries are based on the exact $\text{se}_n = \sqrt{\theta^*(1 - \theta^*)/n} = \sqrt{1/4}$, while those with red boundaries are based on the estimated $\widehat{\text{se}}_n = \sqrt{\hat{\theta}_n(1 - \hat{\theta}_n)/n}$. The fraction of times the true parameter $\theta^* = 0.5$ was engulfed by the exact and approximate confidence interval (empirical coverage) over the 100 replications of the experiment for each of the three sample sizes are given by the numbers after Cvrg. = and \sim , above each sub-plot, respectively.



Chapter 12

Maximum Likelihood Estimator

Next we look at a specific point estimator called the maximum likelihood estimator (MLE) of a possibly unknown but fixed parameter θ^* in a parametric experiment, i.e. $\theta^* \in \Theta \subset \mathbb{R}^k$ with $k < \infty$. Other point estimators in such a setting include the moment estimator (MME).

Recall that the likelihood function (See Definition 99) for an IID experiment with observations x_1, x_2, \dots, x_n is simply the product of the densities:

$$L_n(\theta) = \prod_{i=1}^n f(x_i; \theta) : \Theta \rightarrow (0, \infty) ,$$

and its logarithm or log-likelihood function is:

$$\ell_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^n \log(f(x_i)) : \Theta \rightarrow (-\infty, \infty) .$$

12.1 Introduction to Maximum Likelihood Estimation

Definition 100 (Maximum Likelihood Estimator (MLE)) Let $X_1, \dots, X_n \sim f(x_1, \dots, x_n; \theta^*)$. The maximum likelihood estimator (MLE) $\widehat{\Theta}_n$ of the fixed and possibly unknown parameter $\theta^* \in \Theta$ is the value of θ that maximises the likelihood function:

$$\widehat{\Theta}_n := \widehat{\Theta}_n(X_1, X_2, \dots, X_n) := \operatorname{argmax}_{\theta \in \Theta} L_n(\theta) ,$$

Equivalently, MLE is the value of θ that maximises the log-likelihood function:

$$\widehat{\Theta}_n := \operatorname{argmax}_{\theta \in \Theta} \ell_n(\theta) ,$$

since the maximum of the likelihood coincides with that of the log-likelihood. It is analytically and numerically convenient to work with the log-likelihood instead of the likelihood. Optimisation algorithms can be used to find the MLE numerically. Such algorithms by convention tend to find the minimum and the value that minimises a function. So, the MLE is also the the value of θ that minimises the negative likelihood or negative log-likelihood functions:

$$\widehat{\Theta}_n := \operatorname{argmin}_{\theta \in \Theta} -L_n(\theta), \quad \widehat{\Theta}_n := \operatorname{argmin}_{\theta \in \Theta} -\ell_n(\theta) .$$

Once again, the realisation of the MLE, namely $\widehat{\theta}_n = \widehat{\Theta}_n(x_1, \dots, x_n)$ based on the observation is the maximum likelihood estimate (MLE) of the θ^* .

Example 178 (Coin Tossing Experiment) ($X_1, \dots, X_{10} \stackrel{IID}{\sim} \text{Bernoulli}(\theta^*)$) I tossed a coin that has an unknown probability θ^* of landing Heads independently and identically 10 times in a row. Four of my outcomes were Heads and the remaining six were Tails, with the actual sequence of Bernoulli outcomes (Heads $\rightarrow 1$ and Tails $\rightarrow 0$) being $(1, 0, 0, 0, 1, 1, 0, 0, 1, 0)$. I would like to estimate the probability $\theta^* \in \Theta = [0, 1]$ of observing Heads using the maximum likelihood estimator or MLE $\widehat{\Theta}_n((X_1, X_2, \dots, X_n))$ of θ . We derive the MLE next.

First, the likelihood function is:

$$L_n(\theta) := L_n(x_1, x_2, \dots, x_n; \theta) = \prod_{i=1}^n f(x_i | \theta) = \theta^{\sum_{i=1}^n x_i} (1 - \theta)^{n - \sum_{i=1}^n x_i} := \theta^{t_n} (1 - \theta)^{n - t_n}$$

In the last step, we have formally defined the following statistic of the data:

$$T_n(X_1, X_2, \dots, X_n) = \sum_{i=1}^n X_i : \mathbb{X}_n \rightarrow \mathbb{T}_n$$

with the corresponding realisation $t_n := T_n(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i \in \mathbb{T}_n$. Let us now take the natural logarithm of both sides:

$$\log(L_n(\theta)) := \log(L(x_1, x_2, \dots, x_n; \theta)) = \log(\theta^{t_n} (1 - \theta)^{n - t_n}) = t_n \log(\theta) + (n - t_n) \log(1 - \theta)$$

Next, we take the derivative with respect to the parameter θ :

$$\begin{aligned} \frac{\partial}{\partial \theta} \log(L_n(\theta)) &= \frac{\partial}{\partial \theta} t_n \log(\theta) + \frac{\partial}{\partial \theta} (n - t_n) \log(1 - \theta) \\ &= \frac{t_n}{\theta} - \frac{n - t_n}{1 - \theta} \end{aligned}$$

Now, set $\frac{\partial}{\partial \theta} \log(L_n(\theta)) = 0$ and solve for θ to obtain the maximum likelihood estimate $\widehat{\theta}_n$:

$$\frac{\partial}{\partial \theta} \log(L(\theta)) = 0 \iff \frac{t_n}{\theta} = \frac{n - t_n}{1 - \theta} \iff \frac{1 - \theta}{\theta} = \frac{n - t_n}{t_n} \iff \frac{1}{\theta} - 1 = \frac{n}{t_n} - 1 \iff \widehat{\theta}_n = \frac{t_n}{n}$$

Therefore the MLE is:

$$\widehat{\Theta}_n(X_1, X_2, \dots, X_n) = \frac{1}{n} T_n(X_1, X_2, \dots, X_n) = \frac{1}{n} \sum_{i=1}^n X_i = \bar{X}_n$$

For the coin tossing experiment I just performed ($n = 10$ times), the point estimate of θ is:

$$\begin{aligned} \widehat{\theta}_{10} = \widehat{\Theta}_{10}((x_1, x_2, \dots, x_{10})) &= \widehat{\Theta}_{10}((1, 0, 0, 0, 1, 1, 0, 0, 1, 0)) \\ &= \frac{1 + 0 + 0 + 0 + 1 + 1 + 0 + 0 + 1 + 0}{10} = \frac{4}{10} = 0.40. \end{aligned}$$

12.2 Practical Excursion in One-dimensional Optimisation

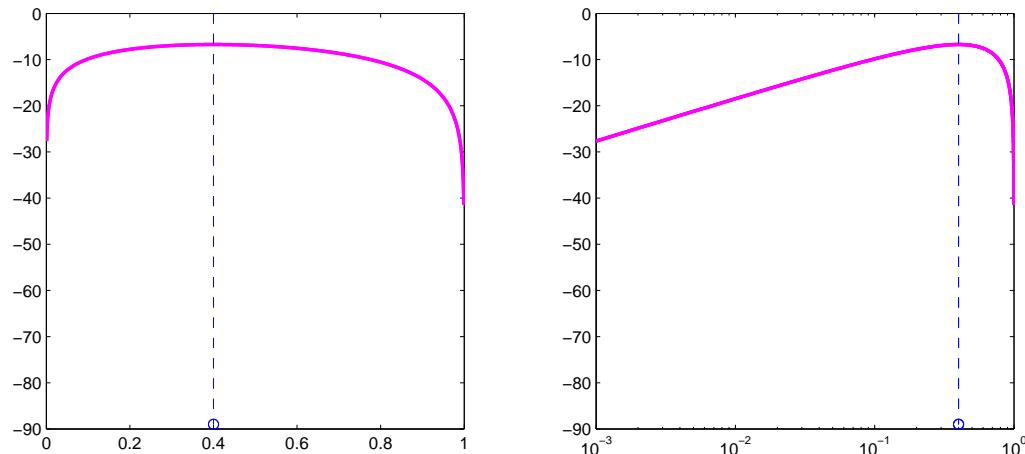
Numerically maximising a log-likelihood function of one parameter is a useful technique. This can be used for models with no analytically known MLE. A fairly large field of maths, called optimisation, exists for this sole purpose. Conventionally, in optimisation, one is interested in minimisation. Therefore, the basic algorithms are cast in the “find the minimiser and the minimum” of a target function $f : \mathbb{R} \rightarrow \mathbb{R}$. Since we are interested in maximising our target, which is the likelihood

or log-likelihood function, say $\log(L(x_1, \dots, x_n; \theta)) : \Theta \rightarrow \mathbb{R}$, we will simply apply the standard optimisation algorithms directly to $-\log(L(x_1, \dots, x_n; \theta)) : \Theta \rightarrow \mathbb{R}$.

The algorithm implemented in `fminbnd` is based on the golden section search and an inverse parabolic interpolation, and attempts to find the minimum of a function of one variable within a given fixed interval. Briefly, the golden section search proceeds by successively **bracketing** the minimum of the target function within an acceptably small interval inside the given starting interval [see Section 8.2 of Forsythe, G. E., M. A. Malcolm, and C. B. Moler, 1977, *Computer Methods for Mathematical Computations*, Prentice-Hall]. MATLAB's `fminbnd` also relies on Brent's inverse parabolic interpolation [see Chapter 5 of Brent, Richard. P., 1973, *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, New Jersey]. Briefly, additional smoothness conditions are assumed for the target function to aid in a faster bracketing strategy through polynomial interpolations of past function evaluations. MATLAB's `fminbnd` has several limitations, including:

- The likelihood function must be continuous.
- Only local MLE solutions, i.e. those inside the starting interval, are given.
- One needs to know or carefully guess the starting interval that contains the MLE.
- MATLAB's `fminbnd` exhibits slow convergence when the solution is on a boundary of the starting interval.

Figure 12.1: Plot of $\log(L(1, 0, 0, 0, 1, 1, 0, 0, 1, 0; \theta))$ as a function of the parameter θ over the parameter space $\Theta = [0, 1]$ and the MLE $\hat{\theta}_{10}$ of 0.4 for the coin-tossing experiment.



Labwork 179 (Coin-tossing experiment) The following script was used to study the coin-tossing experiment in MATLAB. The plot of the log-likelihood function and the numerical optimisation of MLE are carried out using MATLAB's built-in function `fminbnd` (See Figure 12.1).

BernoulliMLE.m

```
% To simulate n coin tosses, set theta=probability of heads and n
% Then draw n IID samples from Bernoulli(theta) RV
% theta=0.5; n=20; x=floor(rand(1,n) + theta);
% enter data from a real coin tossing experiment
x=[1 0 0 0 1 1 0 0 1 0]; n=length(x);
t = sum(x); % statistic t is the sum of the x_i values
```

```
% display the outcomes and their sum
display(x)
display(t)

% Analytically MLE is t/n
MLE=t/n

% l is the log-likelihood of data x as a function of parameter theta
l=@(theta)log(theta ^ t * (1-theta)^(n-t));
ThetaS=[0:0.001:1]; % sample some values for theta

% plot the log-likelihood function and MLE in two scales
subplot(1,2,1);
plot(ThetaS,arrayfun(l,ThetaS),'m','LineWidth',2);
hold on; stem([MLE],[-89],'b--'); % plot MLE as a stem plot
subplot(1,2,2);
semilogx(ThetaS,arrayfun(l,ThetaS),'m','LineWidth',2);
hold on; stem([MLE],[-89],'b--'); % plot MLE as a stem plot

% Now we will find the MLE by finding the minimiser or argmin of -l
% negative log-likelihood function of parameter theta
negl=@(theta)-(log(theta ^ t * (1-theta)^(n-t)));
% read help fminbnd
% you need to supply the function to be minimised and its search interval
% NumericalMLE = fminbnd(negl,0,1)
% to see the iteration in the numerical minimisation
NumericalMLE = fminbnd(negl,0,1,optimset('Display','iter'))
```

```
>> BernoulliMLE
x =     1     0     0     0     1     1     0     0     1     0
t =      4
MLE =    0.4000
Func-count      x          f(x)        Procedure
      1    0.381966    6.73697    initial
      2    0.618034    7.69939    golden
      3    0.236068    7.3902    golden
      4    0.408979    6.73179    parabolic
      5    0.399339    6.73013    parabolic
      6    0.400045    6.73012    parabolic
      7    0.400001    6.73012    parabolic
      8    0.399968    6.73012    parabolic
Optimisation terminated:
the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04
NumericalMLE =    0.4000
```

Example 180 (MLE of an IID Exponential(λ^*) experiment) Let us derive the MLE $\hat{\Lambda}_n$ of the fixed and possibly unknown λ^* for the IID experiment:

$$X_1, \dots, X_n \stackrel{IID}{\sim} \text{Exponential}(\lambda^*), \quad \lambda^* \in \mathbf{A} = (0, \infty) .$$

Note that \mathbf{A} is the parameter space.

We first obtain the log-likelihood function of λ for the data $x_1, x_2, \dots, x_n \stackrel{IID}{\sim} \text{Exponential}(\lambda)$.

$$\begin{aligned} \ell(\lambda) &:= \log(L(x_1, x_2, \dots, x_n; \lambda)) = \log \left(\prod_{i=1}^n f(x_i; \lambda) \right) = \log \left(\prod_{i=1}^n \lambda e^{-\lambda x_i} \right) \\ &= \log \left(\lambda e^{-\lambda x_1} \cdot \lambda e^{-\lambda x_2} \cdots \lambda e^{-\lambda x_n} \right) = \log \left(\lambda^n e^{-\lambda \sum_{i=1}^n x_i} \right) \\ &= \log(\lambda^n) + \log \left(e^{-\lambda \sum_{i=1}^n x_i} \right) = \log(\lambda^n) - \lambda \sum_{i=1}^n x_i \end{aligned}$$

Now, let us take the derivative with respect to λ ,

$$\begin{aligned}\frac{\partial}{\partial \lambda}(\ell(\lambda)) &:= \frac{\partial}{\partial \lambda} \left(\log(\lambda^n) - \lambda \sum_{i=1}^n x_i \right) = \frac{\partial}{\partial \lambda} (\log(\lambda^n)) - \frac{\partial}{\partial \lambda} \left(\lambda \sum_{i=1}^n x_i \right) \\ &= \frac{1}{\lambda^n} \frac{\partial}{\partial \lambda} (\lambda^n) - \sum_{i=1}^n x_i = \frac{1}{\lambda^n} n \lambda^{n-1} - \sum_{i=1}^n x_i = \frac{n}{\lambda} - \sum_{i=1}^n x_i .\end{aligned}$$

Next, we set the derivative to 0, solve for λ , and set the solution equal to the ML estimate $\hat{\lambda}_n$.

$$0 = \frac{\partial}{\partial \lambda}(\ell(\lambda)) \iff 0 = \frac{n}{\lambda} - \sum_{i=1}^n x_i \iff \sum_{i=1}^n x_i = \frac{n}{\lambda} \iff \lambda = \frac{n}{\sum_{i=1}^n x_i} \iff \boxed{\hat{\lambda}_n = \frac{1}{\bar{x}_n}} .$$

Therefore, the ML estimate $\hat{\lambda}_n$ of the unknown rate parameter $\lambda^* \in \mathbb{A}$ on the basis of n IID observations $x_1, x_2, \dots, x_n \stackrel{IID}{\sim} \text{Exponential}(\lambda^*)$ is $1/\bar{x}_n$ and the ML estimator $\hat{\Lambda}_n = 1/\bar{X}_n$. Let us apply this ML estimator of the rate parameter for the supposedly exponentially distributed waiting times at the on-campus Orbiter bus-stop.

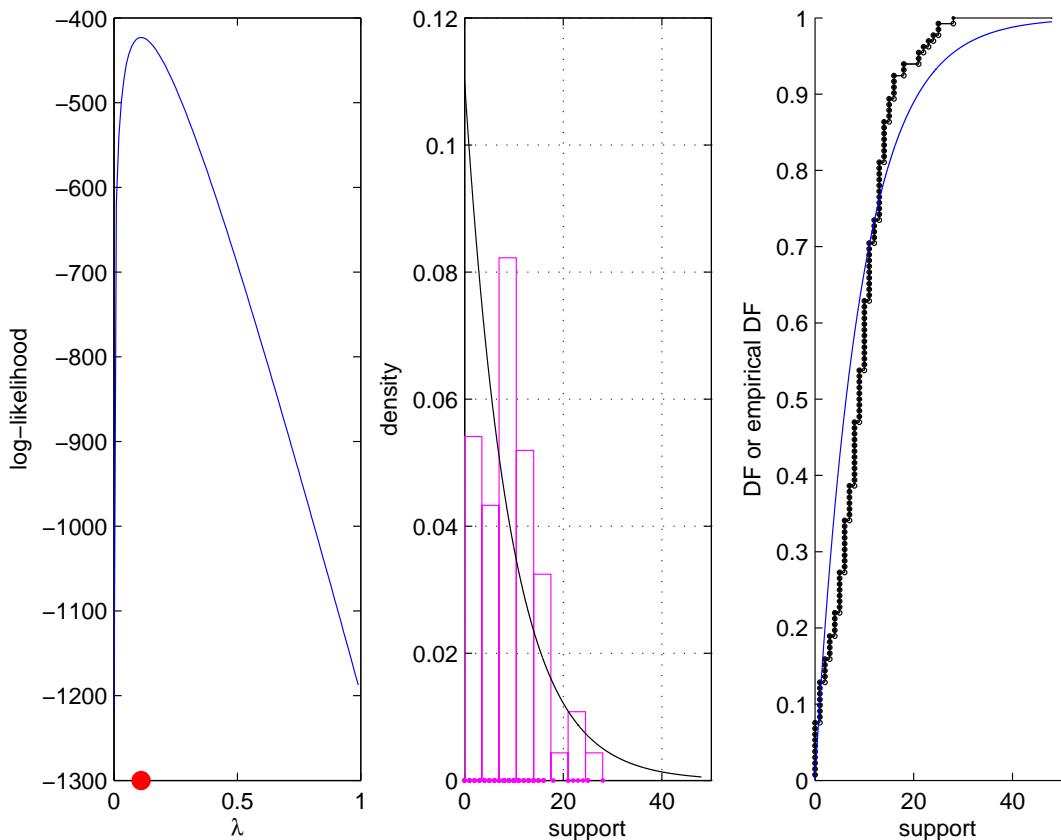
Labwork 181 (Numerical MLE of λ from n IID Exponential(λ) RVs) Joshua Fenemore and Yiran Wang collected data on waiting times between buses at an Orbiter bus-stop close to campus and modelled the waiting times as IID Exponential(λ^*) RVs (<http://www.math.canterbury.ac.nz/~r.sainudiin/courses/STAT218/projects/Stat218StudentProjects2007.pdf>). We can use their data `sampleTimes` to find the MLE of λ^* under the assumption that the waiting times X_1, \dots, X_{132} are IID Exponential(λ^*). We find the ML estimate $\hat{\lambda}_{132} = 0.1102$ and thus the estimated mean waiting time is $1/\hat{\lambda}_{132} = 9.0763$ minutes. The estimated mean waiting time for a bus to arrive is well within the 10 minutes promised by the Orbiter bus company. The following script was used to generate the Figure 12.2:

```
%> ExponentialMLEOrbiter.m
% Joshu Fenemore's Data from 2007 on Waiting Times at Orbiter Bust Stop
%The raw data -- the waiting times in minutes for each direction
antiTimes=[8 3 7 18 18 3 7 9 9 25 0 0 25 6 10 0 10 8 16 9 1 5 16 6 4 1 3 21 0 28 3 8 ...
6 6 11 8 10 15 0 8 7 11 10 9 12 13 8 10 11 8 7 11 5 9 11 14 13 5 8 9 12 10 13 6 11 13];
clockTimes=[0 0 11 1 9 5 14 16 2 10 21 1 14 2 10 24 6 1 14 14 0 14 4 11 15 0 10 2 13 2 22 ...
10 5 6 13 1 13 10 11 4 7 9 12 8 16 15 14 5 10 12 9 8 0 5 13 13 6 8 4 13 15 7 11 6 23 1];
sampleTimes=[antiTimes clockTimes];% pool all times into 1 array
% L = Log Likelihood of data x as a function of parameter lambda
L=@(lambda)sum(log(lambda*exp(-lambda * sampleTimes)));
LAMBDA=[0.0001:0.01:1]; % sample some values for lambda
clf;
subplot(1,3,1);
plot(LAMBDA,arrayfun(L,LAMBDA)); % plot the Log Likelihood function
% Now we will find the Maximum Likelihood Estimator by finding the minimizer of -L
MLE = fminbnd(@(lambda)-sum(log(lambda*exp(-lambda * sampleTimes))),0.0001,1)
MeanEstimate=1/MLE
hold on; % plot the MLE
plot([MLE],[-1300],'.','MarkerSize',25); ylabel('log-likelihood'); xlabel('\lambda');
subplot(1,3,2); % plot a histogram estimate
histogram(sampleTimes,1,[min(sampleTimes),max(sampleTimes)],'m',2);
hold on; TIMES=[0.0001:0.01:max(sampleTimes)+20]; % points on support
plot(TIMES,MLE*exp(-MLE * TIMES ),'k-') % plot PDF at MLE to compare with histogram
% compare the empirical DF to the best fitted DF
subplot(1,3,3)
ECDF(sampleTimes,5,0.0,20); hold on
plot(TIMES,ExponentialCdf(TIMES,MLE),'b-')
ylabel('DF or empirical DF'); xlabel('support');
```

The script output the following in addition to the plot:

```
>> ExponentialMLEOrbiter
MLE =      0.1102
MeanEstimate =   9.0763
```

Figure 12.2: Plot of $\log(L(\lambda))$ as a function of the parameter λ and the MLE $\hat{\lambda}_{132}$ of 0.1102 for Fenemore-Wang Orbiter Waiting Times Experiment from STAT 218 S2 2007. The density or PDF and the DF at the MLE of 0.1102 are compared with a histogram and the empirical DF.

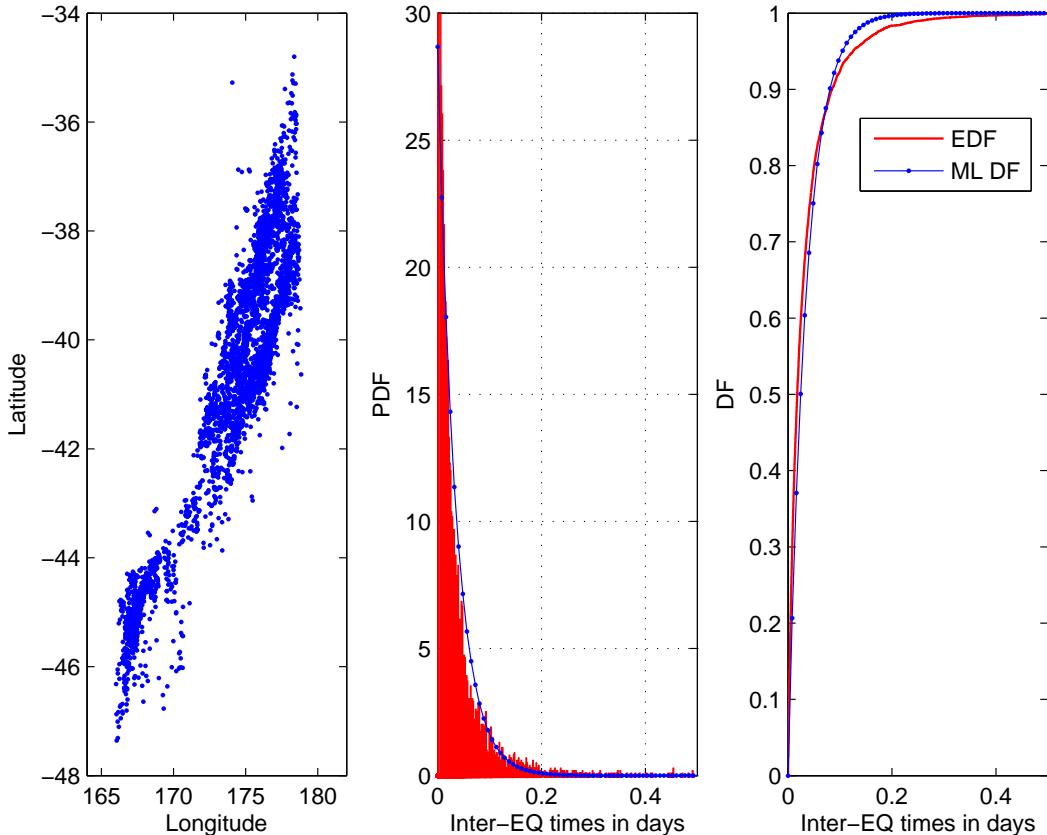


Notice how poorly the exponential PDF $f(x; \hat{\lambda}_{132} = 0.1102)$ and the DF $F(x; \hat{\lambda}_{132} = 0.1102)$ based on the MLE fits with the histogram and the empirical DF, respectively. This is an indication of the inadequacy of our parametric model. Partly this discrepancy is due to the resolution of the measurements being confined to whole minutes. We can overcome this problem by fitting a minute-discretized PMF from the $\text{Exponential}(\lambda)$ PDF. In the next Labwork, we simulate data from an $\text{Exponential}(\lambda^* = 0.1)$ RV to conduct point estimation in the theoretically ideal setting.

Labwork 182 (MLE of the rate parameter for waiting times at my bus stop) Recall Labwork 65 where you modeled the arrival of buses at a bus stop using the IID $\text{Exponential}(\lambda^* = 0.1)$ distributed inter-arrival times with a mean of $1/\lambda^* = 10$ minutes. Once again, seed the fundamental sampler by your Student ID (e.g. if your ID is 11424620 then type `rand('twister', 11424620);`), just before simulating the inter-arrival times of the next seven buses. Hand in the following six items:

1. Waiting times x_1, x_2, \dots, x_7 between arrivals of the next seven buses at your ID-seeded bus stop;
2. A plot of the empirical DF \hat{F}_n from your (simulated) data x_1, x_2, \dots, x_7 . [You may use the MATLAB function ECDF of Labwork 247)];
3. The first, second and third sample quartiles as well as the 0.20th sample quantile for your data x_1, x_2, \dots, x_7 . [You may use the MATLAB function qthSampleQuantile of Labwork 248];
4. Pretending that you did not know the true parameter ($\lambda^* = 0.1$) used in the simulation, produce the maximum likelihood estimate (ML estimate) $\hat{\lambda}_7$ from your seven observations x_1, x_2, \dots, x_7 ;
5. Plot the log-likelihood function for your data x_1, x_2, \dots, x_7 as a function of the parameter λ ; and
6. Show that you have verified that the numerical optimisation routine fminbnd returns the correct ML estimate $\hat{\lambda}_7$.

Figure 12.3: Comparing the Exponential($\hat{\lambda}_{6128} = 28.6694$) PDF and DF with a histogram and empirical DF of the times (in units of days) between earth quakes in NZ. The epicentres of 6128 earth quakes are shown in left panel.



Labwork 183 (Time between Earth Quakes in NZ) We model the time between 6128 earth-quakes in NZ from 18-Jan-2008 02:23:44 to 18-Aug-2008 19:29:29 as:

$$X_1, X_2, \dots, X_{6128} \stackrel{IID}{\sim} \text{Exponential}(\lambda^*) .$$

Then, the ML estimate of $\lambda^* = 1/\bar{x}_{6128} = 1/0.0349 = 28.6694$ as computed in the following script:

```
NZSIEarthQuakesExponentialMLE.m
%% The columns in earthquakes.csv file have the following headings
%% CUSP_ID,LAT,LONG,NZMGE,NZMGN,ORI_YEAR,ORI_MONTH,ORI_DAY,ORI_HOUR,ORI_MINUTE,ORI_SECOND,MAG,DEPTH
EQ=dlmread('earthquakes.csv',','); % load the data in the matrix EQ
size(EQ) % report thr size of the matrix EQ

% Read help datenum -- converts time stamps into numbers in units of days
MaxD=max(datenum(EQ(:,6:11)));% maximum datenum
MinD=min(datenum(EQ(:,6:11)));% minimum datenum
disp('Earth Quakes in NZ between')
disp(strcat(datestr(MinD), ' and ', datestr(MaxD)))% print MaxD and MinD as a date string

% get an array of sorted time stamps of EQ events starting at 0
Times=sort(datenum(EQ(:,6:11))-MinD);
TimeDiff=diff(Times); % inter-EQ times = times between successive EQ events
clf % clear any current figures
%figure
%plot(TimeDiff) % plot the inter-EQ times
subplot(1,3,1)
plot(EQ(:,3),EQ(:,2),'.')
axis([164 182 -48 -34])
xlabel('Longitude'); ylabel('Latitude');

subplot(1,3,2) % construct a histogram estimate of inter-EQ times
histogram(TimeDiff',1,[min(TimeDiff),max(TimeDiff)],'r',2);
SampleMean=mean(TimeDiff) % find the sample mean
% the MLE of LambdaStar if inter-EQ times are IID Exponential(LambdaStar)
MLELambdaHat=1/SampleMean
hold on;
TIMEs=linspace(0,max(TimeDiff),100);
plot(TIMEs,MLELambdaHat*exp(-MLELambdaHat*TIMEs),'b.-')
axis([0 0.5 0 30])
xlabel('Inter-EQ times in days'); ylabel('PDF');

subplot(1,3,3)
[x y]=ECDF(TimeDiff,0,0,0); % get the coordinates for empirical DF
stairs(x,y,'r','LineWidth',1) % draw the empirical DF
hold on; plot(TIMEs,ExponentialCdf(TIMEs,MLELambdaHat),'b.-');% plot the DF at MLE
axis([0 0.5 0 1])
xlabel('Inter-EQ times in days'); ylabel('DF'); legend('EDF','ML DF')
```

We first load the data in the text file `earthquakes.csv` into a matrix `EQ`. Using the `datenum` function in MATLAB we transform the time stamps into a number starting at zero. These transformed time stamps are in units of days. Then we find the times between consecutive events and estimate a histogram. We finally compute the ML estimate of λ^* and super-impose the PDF of the Exponential($\hat{\lambda}_{6128} = 28.6694$) upon the histogram.

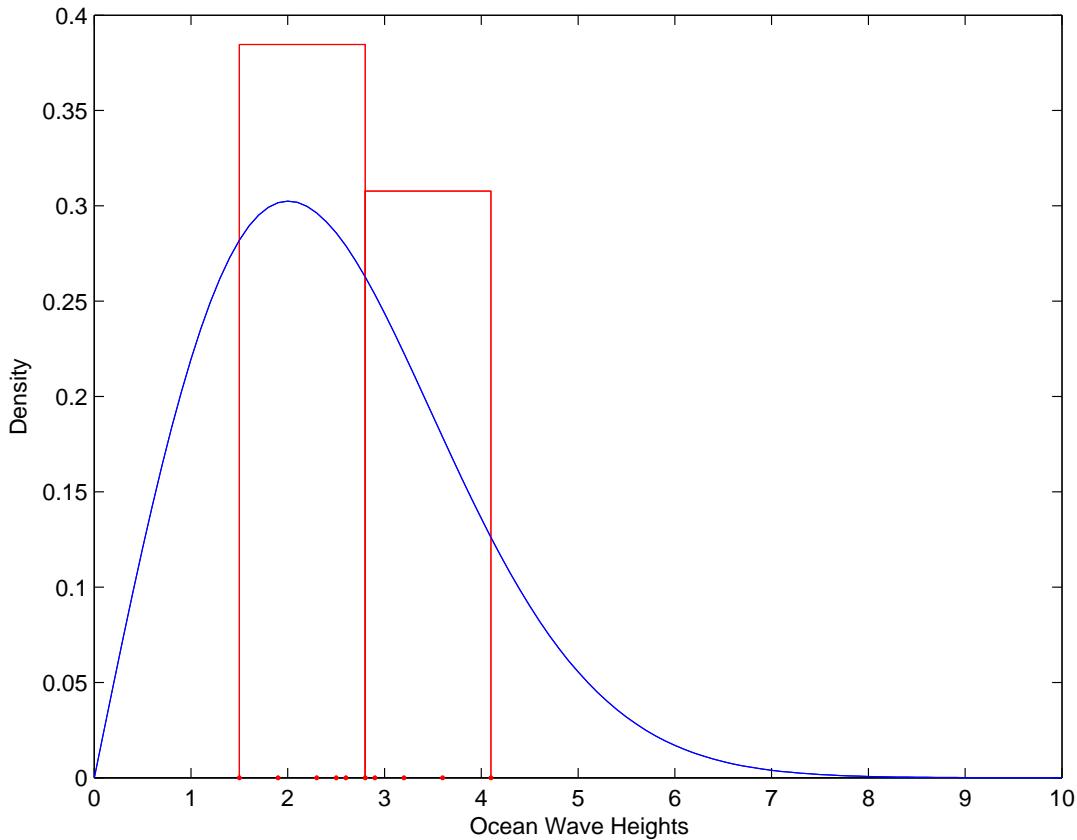
```
>> NZSIEarthQuakesExponentialMLE
ans =      6128      13

Earth Quakes in NZ between
18-Jan-2008 02:23:44 and 18-Aug-2008 19:29:29

SampleMean =    0.0349
MLELambdaHat =  28.6694
```

Thus, the average time between earth quakes is $0.0349 * 24 * 60 = 50.2560$ minutes.

Figure 12.4: The ML fitted Rayleigh($\hat{\alpha}_{10} = 2$) PDF and a histogram of the ocean wave heights.



Labwork 184 (6.7, p. 275 of Ang & Tang) The distribution of ocean wave heights, H , may be modeled with the Rayleigh(α) RV with parameter α and probability density function,

$$f(h; \alpha) = \frac{h}{\alpha^2} \exp\left(-\frac{1}{2}(h/\alpha)^2\right), \quad h \in \mathbb{H} := [0, \infty) .$$

The parameter space for α is $\mathbb{A} = (0, \infty)$. Suppose that the following measurements h_1, h_2, \dots, h_{10} of wave heights in meters were observed to be

$$1.50, 2.80, 2.50, 3.20, 1.90, 4.10, 3.60, 2.60, 2.90, 2.30 ,$$

respectively. Under the assumption that the 10 samples are IID realisations from a Rayleigh(α^*) RV with a fixed and unknown parameter α^* , find the ML estimate $\hat{\alpha}_{10}$ of α^* .

We first obtain the log-likelihood function of α for the data $h_1, h_2, \dots, h_n \stackrel{IID}{\sim} \text{Rayleigh}(\alpha)$.

$$\begin{aligned}\ell(\alpha) &:= \log(L(h_1, h_2, \dots, h_n; \alpha)) = \log \left(\prod_{i=1}^n f(h_i; \alpha) \right) = \sum_{i=1}^n \log(f(h_i; \alpha)) \\ &= \sum_{i=1}^n \log \left(\frac{h_i}{\alpha^2} e^{-\frac{1}{2}(h_i/\alpha)^2} \right) = \sum_{i=1}^n \left(\log(h_i) - 2 \log(\alpha) - \frac{1}{2}(h_i/\alpha)^2 \right) \\ &= \sum_{i=1}^n (\log(h_i)) - 2n \log(\alpha) - \sum_{i=1}^n \left(\frac{1}{2} h_i^2 \alpha^{-2} \right)\end{aligned}$$

Now, let us take the derivative with respect to α ,

$$\begin{aligned}\frac{\partial}{\partial \alpha} (\ell(\alpha)) &:= \frac{\partial}{\partial \alpha} \left(\sum_{i=1}^n (\log(h_i)) - 2n \log(\alpha) - \sum_{i=1}^n \left(\frac{1}{2} h_i^2 \alpha^{-2} \right) \right) \\ &= \frac{\partial}{\partial \alpha} \left(\sum_{i=1}^n (\log(h_i)) \right) - \frac{\partial}{\partial \alpha} (2n \log(\alpha)) - \frac{\partial}{\partial \alpha} \left(\sum_{i=1}^n \left(\frac{1}{2} h_i^2 \alpha^{-2} \right) \right) \\ &= 0 - 2n \frac{1}{\alpha} - \sum_{i=1}^n \left(\frac{1}{2} h_i^2 (-2\alpha^{-3}) \right) = -2n\alpha^{-1} + \alpha^{-3} \sum_{i=1}^n (h_i^2)\end{aligned}$$

Next, we set the derivative to 0, solve for α , and set the solution equal to the ML estimate $\hat{\alpha}_n$.

$$\begin{aligned}0 = \frac{\partial}{\partial \alpha} (\ell(\alpha)) &\iff 0 = -2n\alpha^{-1} + \alpha^{-3} \sum_{i=1}^n h_i^2 \iff 2n\alpha^{-1} = \alpha^{-3} \sum_{i=1}^n h_i^2 \\ &\iff 2n\alpha^{-1}\alpha^3 = \sum_{i=1}^n h_i^2 \iff \alpha^2 = \frac{1}{2n} \sum_{i=1}^n h_i^2 \iff \hat{\alpha}_n = \sqrt{\frac{1}{2n} \sum_{i=1}^n h_i^2}\end{aligned}$$

Therefore, the ML estimate of the unknown $\alpha^* \in \mathbb{A}$ on the basis of our 10 observations h_1, h_2, \dots, h_{10} of wave heights is

$$\begin{aligned}\hat{\alpha}_{10} &= \sqrt{\frac{1}{2 * 10} \sum_{i=1}^{10} h_i^2} \\ &= \sqrt{\frac{1}{20} (1.50^2 + 2.80^2 + 2.50^2 + 3.20^2 + 1.90^2 + 4.10^2 + 3.60^2 + 2.60^2 + 2.90^2 + 2.30^2)} \approx 2\end{aligned}$$

We use the following script file to compute the MLE $\hat{\alpha}_{10}$ and plot the PDF at $\hat{\alpha}_{10}$ in Figure 12.4.

```
RayleighOceanHeightsMLE.m
OceanHeights=[1.50, 2.80, 2.50, 3.20, 1.90, 4.10, 3.60, 2.60, 2.90, 2.30];% data
histogram(OceanHeights,1,[min(OceanHeights),max(OceanHeights)],'r',2); % make a histogram
Heights=0:0.1:10; % get some heights for plotting
AlphaHat=sqrt(sum(OceanHeights .^ 2)/(2*length(OceanHeights))) % find the MLE
hold on; % superimpose the PDF at the MLE
plot(Heights,(Heights/AlphaHat.^2) .* exp(-((Heights/AlphaHat).^2)/2))
xlabel('Ocean Wave Heights'); ylabel('Density');
```

```
>> RayleighOceanHeightsMLE
AlphaHat = 2.0052
```

12.3 Properties of the Maximum Likelihood Estimator

Next, we list some nice properties of the ML Estimator $\hat{\Theta}_n$ for the fixed and possibly unknown $\theta^* \in \Theta$.

1. The ML Estimator is asymptotically consistent, i.e. $\hat{\Theta}_n \xrightarrow{P} \theta^*$.
2. The ML Estimator is asymptotically normal, i.e. $(\hat{\Theta}_n - \theta^*)/\hat{s}\hat{e}_n \rightsquigarrow \text{Normal}(0, 1)$.
3. The estimated standard error of the ML Estimator, $\hat{s}\hat{e}_n$, can usually be computed analytically using the **Fisher Information**.
4. Because of the previous two properties, the $1 - \alpha$ confidence interval can also be computed analytically as $\hat{\Theta}_n \pm z_{\alpha/2}\hat{s}\hat{e}_n$.
5. The ML Estimator is **equivariant**, i.e. $\hat{\psi}_n = g(\hat{\theta}_n)$ is the ML Estimate of $\psi^* = g(\theta^*)$, for some smooth function $g(\theta) = \psi : \Theta \rightarrow \Psi$.
6. We can also obtain the estimated standard error of the estimator $\hat{\Psi}_n$ of $\psi^* \in \Psi$ via the **Delta Method**.
7. The ML Estimator is **asymptotically optimal** or **efficient**. This means that the MLE has the smallest variance among the well-behaved class of estimators as the sample size gets larger.
8. ML Estimator is close to the Bayes estimator (obtained in the Bayesian inferential paradigm).

12.4 Fisher Information

Let $X_1, X_2, \dots, X_n \stackrel{IID}{\sim} f(X_1; \theta)$. Here, $f(X_1; \theta)$ is the probability density function (pdf) or the probability mass function (pmf) of the RV X_1 . Since all RVs are identically distributed, we simply focus on X_1 without loss of generality.

Definition 101 (Fisher Information) The **score function** of an RV X for which the density is parameterised by θ is defined as:

$$\mathcal{S}(X; \theta) := \frac{\partial \log f(X; \theta)}{\partial \theta}, \quad \text{and} \quad \mathbf{E}_\theta(\mathcal{S}(X; \theta)) = 0 .$$

The **Fisher Information** is

$$I_n := \mathbf{V}_\theta \left(\sum_{i=1}^n \mathcal{S}(X_i; \theta) \right) = \sum_{i=1}^n \mathbf{V}_\theta(\mathcal{S}(X_i; \theta)) = nI_1(\theta), \quad (12.1)$$

where I_1 is the Fisher Information of just one of the RVs X_i , e.g. X :

$$\begin{aligned} I_1(\theta) &:= \mathbf{V}_\theta(\mathcal{S}(X; \theta)) = \mathbf{E}_\theta(\mathcal{S}^2(X; \theta)) \\ &= -\mathbf{E}_\theta \left(\frac{\partial^2 \log f(X; \theta)}{\partial^2 \theta} \right) = \begin{cases} -\sum_{x \in \mathbb{X}} \left(\frac{\partial^2 \log f(x; \theta)}{\partial^2 \theta} \right) f(x; \theta) & \text{for discrete } X \\ -\int_{x \in \mathbb{X}} \left(\frac{\partial^2 \log f(x; \theta)}{\partial^2 \theta} \right) f(x; \theta) dx & \text{for continuous } X \end{cases} \end{aligned} \quad (12.2)$$

Next, we give a **general method** for obtaining:

1. The standard error $\text{se}_n(\hat{\Theta}_n)$ of **any** maximum likelihood estimator $\hat{\Theta}_n$ of the possibly unknown and fixed parameter of interest $\theta^* \in \Theta$, and
2. The $1 - \alpha$ confidence interval for θ^* .

Proposition 102 (Asymptotic Normality of the ML Estimator & Confidence Intervals)

Let $\hat{\Theta}_n$ be the maximum likelihood estimator of $\theta^* \in \Theta$ with standard error $\text{se}_n := \sqrt{\mathbf{V}_{\theta^*}(\hat{\Theta}_n)}$. Under appropriate regularity conditions, the following propositions are true:

1. The standard error se_n can be approximated by the side of a square whose area is the inverse Fisher Information at θ^* , and the distribution of $\hat{\Theta}_n$ approaches that of the $\text{Normal}(\theta^*, \text{se}_n^2)$ distribution as the samples size n gets larger. In other terms:

$$\text{se}_n \approx \sqrt{1/I_n(\theta^*)} \quad \text{and} \quad \frac{\hat{\Theta}_n - \theta^*}{\text{se}_n} \rightsquigarrow \text{Normal}(0, 1)$$

2. The approximation holds even if we substitute the ML Estimate $\hat{\theta}_n$ for θ^* and use the estimated standard error $\hat{\text{se}}_n$ instead of se_n . Let $\hat{\text{se}}_n = \sqrt{1/I_n(\hat{\theta}_n)}$. Then:

$$\frac{\hat{\Theta}_n - \theta^*}{\hat{\text{se}}_n} \rightsquigarrow \text{Normal}(0, 1)$$

3. Using the fact that $\hat{\Theta}_n \rightsquigarrow \text{Normal}(\theta^*, \hat{\text{se}}_n^2)$, we can construct the estimate of an approximate Normal-based $1 - \alpha$ confidence interval as:

$$C_n = [\underline{C}_n, \bar{C}_n] = [\hat{\theta}_n - z_{\alpha/2} \hat{\text{se}}_n, \hat{\theta}_n + z_{\alpha/2} \hat{\text{se}}_n] = \hat{\theta}_n \pm z_{\alpha/2} \hat{\text{se}}_n$$

Now, let us do an example.

Example 185 (MLE and Confidence Interval for the IID Poisson(λ) experiment) Suppose the fixed parameter $\lambda^* \in \Lambda = (0, \infty)$ is unknown. Let $X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Poisson}(\lambda^*)$. We want to find the ML Estimate $\hat{\lambda}_n$ of λ^* and produce a $1 - \alpha$ confidence interval for λ^* .

The MLE can be obtained as follows:

The likelihood function is:

$$L(\lambda) := L(x_1, x_2, \dots, x_n; \lambda) = \prod_{i=1}^n f(x_i; \lambda) = \prod_{i=1}^n e^{-\lambda} \frac{\lambda^{x_i}}{x_i!}$$

Hence, the log-likelihood function is:

$$\begin{aligned} \ell(\theta) := \log(L(\lambda)) &= \log \left(\prod_{i=1}^n e^{-\lambda} \frac{\lambda^{x_i}}{x_i!} \right) = \sum_{i=1}^n \log \left(e^{-\lambda} \frac{\lambda^{x_i}}{x_i!} \right) = \sum_{i=1}^n (\log(e^{-\lambda}) + \log(\lambda^{x_i}) - \log(x_i!)) \\ &= \sum_{i=1}^n (-\lambda + x_i \log(\lambda) - \log(x_i!)) = \sum_{i=1}^n -\lambda + \sum_{i=1}^n x_i \log(\lambda) - \sum_{i=1}^n \log(x_i!) \\ &= n(-\lambda) + \log(\lambda) \left(\sum_{i=1}^n x_i \right) - \sum_{i=1}^n \log(x_i!) \end{aligned}$$

Next, take the derivative of $\ell(\lambda)$:

$$\frac{\partial}{\partial \lambda} \ell(\lambda) = \frac{\partial}{\partial \lambda} \left(n(-\lambda) + \log(\lambda) \left(\sum_{i=1}^n x_i \right) - \sum_{i=1}^n \log(x_i!) \right) = n(-1) + \frac{1}{\lambda} \left(\sum_{i=1}^n x_i \right) + 0$$

and set it equal to 0 to solve for λ , as follows:

$$0 = n(-1) + \frac{1}{\lambda} \left(\sum_{i=1}^n x_i \right) + 0 \iff n = \frac{1}{\lambda} \left(\sum_{i=1}^n x_i \right) \iff \lambda = \frac{1}{n} \left(\sum_{i=1}^n x_i \right) = \bar{x}_n$$

Finally, the ML Estimator of λ^* is $\hat{\Lambda}_n = \bar{X}_n$ and the ML estimate is $\hat{\lambda}_n = \bar{x}_n$.

Now, we want an $1 - \alpha$ confidence interval for λ^* using the $\hat{s}\text{e}_n \approx \sqrt{1/I_n(\hat{\lambda}_n)}$ that is based on the Fisher Information $I_n(\lambda) = nI_1(\lambda)$ given in (12.1). We need I_1 given in (12.2). Since $X_1, X_2, \dots, X_n \sim \text{Poisson}(\lambda)$, we have discrete RVs:

$$I_1 = - \sum_{x \in \mathbb{X}} \left(\frac{\partial^2 \log(f(x; \lambda))}{\partial^2 \lambda} \right) f(x; \lambda) = - \sum_{x=0}^{\infty} \left(\frac{\partial^2 \log(f(x; \lambda))}{\partial^2 \lambda} \right) f(x; \lambda)$$

First find

$$\begin{aligned} \frac{\partial^2 \log(f(x; \lambda))}{\partial^2 \lambda} &= \frac{\partial}{\partial \lambda} \left(\frac{\partial}{\partial \lambda} \log(f(x; \lambda)) \right) = \frac{\partial}{\partial \lambda} \left(\frac{\partial}{\partial \lambda} \log \left(e^{-\lambda} \frac{\lambda^x}{x!} \right) \right) \\ &= \frac{\partial}{\partial \lambda} \left(\frac{\partial}{\partial \lambda} (-\lambda + x \log(\lambda) - \log(x!)) \right) = \frac{\partial}{\partial \lambda} \left(-1 + \frac{x}{\lambda} - 0 \right) = -\frac{x}{\lambda^2} \end{aligned}$$

Now, substitute the above expression into the right-hand side of I_1 to obtain:

$$I_1 = - \sum_{x=0}^{\infty} \left(-\frac{x}{\lambda^2} \right) f(x; \lambda) = \frac{1}{\lambda^2} \sum_{x=0}^{\infty} (x) f(x; \lambda) = \frac{1}{\lambda^2} \sum_{x=0}^{\infty} (x) e^{-\lambda} \frac{\lambda^x}{x!} = \frac{1}{\lambda^2} \mathbf{E}_{\lambda}(X) = \frac{1}{\lambda^2} \lambda = \frac{1}{\lambda}$$

In the third-to-last step above, we recognise the sum as the expectation of the $\text{Poisson}(\lambda)$ RV X , namely $\mathbf{E}_{\lambda}(X) = \lambda$. Therefore, the estimated standard error is:

$$\hat{s}\text{e}_n \approx \sqrt{1/I_n(\hat{\lambda}_n)} = \sqrt{1/(nI_1(\hat{\lambda}_n))} = \sqrt{1/(n(1/\hat{\lambda}_n))} = \sqrt{\hat{\lambda}_n/n}$$

and the approximate $1 - \alpha$ confidence interval is

$$\hat{\lambda}_n \pm z_{\alpha/2} \hat{s}\text{e}_n = \hat{\lambda}_n \pm z_{\alpha/2} \sqrt{\hat{\lambda}_n/n}$$

Thus, using the MLE and the estimated standard error via the Fisher Information, we can carry out point estimation and confidence interval construction in **most** parametric families of RVs encountered in typical engineering applications.

Example 186 (Fisher Information of the Bernoulli Experiment) Suppose $X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Bernoulli}(\theta^*)$. Also, suppose that $\theta^* \in \Theta = [0, 1]$ is unknown. We have already shown in Example 178 that the ML estimator of θ^* is $\hat{\theta}_n = \bar{X}_n$. Using the identity:

$$\hat{s}\text{e}_n = \frac{1}{\sqrt{I_n(\hat{\theta}_n)}}$$

(1) we can compute $\widehat{\text{se}}_n(\widehat{\theta}_n)$, the estimated standard error of the unknown parameter θ^* as follows:

$$\widehat{\text{se}}_n(\widehat{\theta}_n) = \frac{1}{\sqrt{I_n(\widehat{\theta}_n)}} = \frac{1}{\sqrt{nI_1(\widehat{\theta}_n)}} .$$

So, we need to first compute $I_1(\theta)$, the Fisher Information of one sample. Due to (12.2) and the fact that the Bernoulli(θ^*) distributed RV X is discrete with probability mass function $f(x; \theta) = \theta^x(1 - \theta)^{1-x}$, for $x \in \mathbb{X} := \{0, 1\}$, we have,

$$I_1(\theta) = -\mathbf{E}_\theta \left(\frac{\partial^2 \log f(X; \theta)}{\partial^2 \theta} \right) = - \sum_{x \in \mathbb{X}=\{0,1\}} \left(\frac{\partial^2 \log (\theta^x(1 - \theta)^{1-x})}{\partial^2 \theta} \right) \theta^x(1 - \theta)^{1-x}$$

Next, let us compute,

$$\begin{aligned} \frac{\partial^2 \log (\theta^x(1 - \theta)^{1-x})}{\partial^2 \theta} &:= \frac{\partial}{\partial \theta} \left(\frac{\partial}{\partial \theta} (\log (\theta^x(1 - \theta)^{1-x})) \right) = \frac{\partial}{\partial \theta} \left(\frac{\partial}{\partial \theta} (x \log(\theta) + (1 - x) \log(1 - \theta)) \right) \\ &= \frac{\partial}{\partial \theta} (x\theta^{-1} + (1 - x)(1 - \theta)^{-1}(-1)) = \frac{\partial}{\partial \theta} (x\theta^{-1} - (1 - x)(1 - \theta)^{-1}) \\ &= x(-1)\theta^{-1-1} - (1 - x)(-1)(1 - \theta)^{-1-1}(-1) = -x\theta^{-2} - (1 - x)(1 - \theta)^{-2} \end{aligned}$$

Now, we compute the expectation I_1 , i.e. the sum over the two possible values of $x \in \{0, 1\}$,

$$\begin{aligned} I_1(\theta) &= - \sum_{x \in \mathbb{X}=\{0,1\}} \left(\frac{\partial^2 \log (\theta^x(1 - \theta)^{1-x})}{\partial^2 \theta} \right) \theta^x(1 - \theta)^{1-x} \\ &= - ((-0\theta^{-2} - (1 - 0)(1 - \theta)^{-2})\theta^0(1 - \theta)^{1-0} + (-1\theta^{-2} - (1 - 1)(1 - \theta)^{-2})\theta^1(1 - \theta)^{1-1}) \\ &= - ((0 - 1(1 - \theta)^{-2})1(1 - \theta)^1 + (-\theta^{-2} - 0)\theta^11) = (1 - \theta)^{-2}(1 - \theta)^1 + \theta^{-2}\theta^1 \\ &= (1 - \theta)^{-1} + \theta^{-1} = \frac{1}{1 - \theta} + \frac{1}{\theta} = \frac{\theta}{\theta(1 - \theta)} + \frac{1 - \theta}{\theta(1 - \theta)} = \frac{\theta + (1 - \theta)}{\theta(1 - \theta)} = \frac{1}{\theta(1 - \theta)} \end{aligned}$$

Therefore, the desired estimated standard error of our estimator, can be obtained by substituting the ML estimate $\widehat{\theta}_n = \bar{x}_n := n^{-1} \sum_{i=1}^n x_i$ of the unknown θ^* as follows:

$$\widehat{\text{se}}_n(\widehat{\theta}_n) = \frac{1}{\sqrt{I_n(\widehat{\theta}_n)}} = \frac{1}{\sqrt{nI_1(\widehat{\theta}_n)}} = \sqrt{\frac{1}{n \frac{1}{\widehat{\theta}_n(1 - \widehat{\theta}_n)}}} = \sqrt{\frac{\widehat{\theta}_n(1 - \widehat{\theta}_n)}{n}} = \sqrt{\frac{\bar{x}_n(1 - \bar{x}_n)}{n}} .$$

(2) Using $\widehat{\text{se}}_n(\widehat{\theta}_n)$ we can construct an approximate 95% confidence interval C_n for θ^* , due to the asymptotic normality of the ML estimator of θ^* , as follows:

$$C_n = \widehat{\theta}_n \pm 1.96 \sqrt{\frac{\widehat{\theta}_n(1 - \widehat{\theta}_n)}{n}} = \bar{x}_n \pm 1.96 \sqrt{\frac{\bar{x}_n(1 - \bar{x}_n)}{n}}$$

Recall that C_n is the realisation of a random set based on your observed samples or data x_1, x_2, \dots, x_n . Furthermore, C_n 's construction procedure ensures the engulfing of the unknown θ^* with probability approaching 0.95 as the sample size n gets large.

Example 187 ([Fisher Information of the Exponential Experiment]) Let us get our hands dirty with a continuous RV next. Let $X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Exponential}(\lambda^*)$. We saw that the ML

estimator of $\lambda^* \in \Lambda = (0, \infty)$ is $\widehat{\Lambda}_n = 1/\bar{X}_n$ and its ML estimate is $\widehat{\lambda}_n = 1/\bar{x}_n$, where x_1, x_2, \dots, x_n are our observed data.

(1) Let us obtain the Fisher Information I_n for this experiment to find the standard error:

$$\widehat{\text{se}}_n(\widehat{\Lambda}_n) = \frac{1}{\sqrt{I_n(\widehat{\lambda}_n)}} = \frac{1}{\sqrt{nI_1(\widehat{\lambda}_n)}}$$

and construct an approximate 95% confidence interval for λ^* using the asymptotic normality of its ML estimator $\widehat{\Lambda}_n$.

So, we need to first compute $I_1(\theta)$, the Fisher Information of one sample. Due to (12.2) and the fact that the Exponential(λ^*) distributed RV X is continuous with probability density function $f(x; \lambda) = \lambda e^{-\lambda x}$, for $x \in \mathbb{X} := [0, \infty)$, we have,

$$I_1(\theta) = -\mathbf{E}_\theta \left(\frac{\partial^2 \log f(X; \theta)}{\partial^2 \theta} \right) = - \int_{x \in \mathbb{X} = [0, \infty)} \left(\frac{\partial^2 \log (\lambda e^{-\lambda x})}{\partial^2 \lambda} \right) \lambda e^{-\lambda x} dx$$

Let us compute the above integrand next.

$$\begin{aligned} \frac{\partial^2 \log (\lambda e^{-\lambda x})}{\partial^2 \lambda} &:= \frac{\partial}{\partial \lambda} \left(\frac{\partial}{\partial \lambda} (\log (\lambda e^{-\lambda x})) \right) = \frac{\partial}{\partial \lambda} \left(\frac{\partial}{\partial \lambda} (\log(\lambda) + \log(e^{-\lambda x})) \right) \\ &= \frac{\partial}{\partial \lambda} \left(\frac{\partial}{\partial \lambda} (\log(\lambda) - \lambda x) \right) = \frac{\partial}{\partial \lambda} (\lambda^{-1} - x) = -\lambda^{-2} - 0 = -\frac{1}{\lambda^2} \end{aligned}$$

Now, let us evaluate the integral by recalling that the expectation of the constant 1 is 1 for any RV X governed by some parameter, say θ . For instance when X is a continuous RV, $\mathbf{E}_\theta(1) = \int_{x \in \mathbb{X}} 1 f(x; \theta) = \int_{x \in \mathbb{X}} f(x; \theta) = 1$. Therefore, the Fisher Information of one sample is

$$\begin{aligned} I_1(\theta) &= - \int_{x \in \mathbb{X} = [0, \infty)} \left(\frac{\partial^2 \log (\lambda e^{-\lambda x})}{\partial^2 \lambda} \right) \lambda e^{-\lambda x} dx = - \int_0^\infty \left(-\frac{1}{\lambda^2} \right) \lambda e^{-\lambda x} dx \\ &= - \left(-\frac{1}{\lambda^2} \right) \int_0^\infty \lambda e^{-\lambda x} dx = \frac{1}{\lambda^2} 1 = \frac{1}{\lambda^2} \end{aligned}$$

Now, we can compute the desired estimated standard error, by substituting in the ML estimate $\widehat{\lambda}_n = 1/(\bar{x}_n) := 1/(\sum_{i=1}^n x_i)$ of λ^* , as follows:

$$\widehat{\text{se}}_n(\widehat{\Lambda}_n) = \frac{1}{\sqrt{I_n(\widehat{\lambda}_n)}} = \frac{1}{\sqrt{nI_1(\widehat{\lambda}_n)}} = \frac{1}{\sqrt{n \frac{1}{\widehat{\lambda}_n^2}}} = \frac{\widehat{\lambda}_n}{\sqrt{n}} = \frac{1}{\sqrt{n} \bar{x}_n}$$

Using $\widehat{\text{se}}_n(\widehat{\lambda}_n)$ we can construct an approximate 95% confidence interval C_n for λ^* , due to the asymptotic normality of the ML estimator of λ^* , as follows:

$$C_n = \widehat{\lambda}_n \pm 1.96 \frac{\widehat{\lambda}_n}{\sqrt{n}} = \frac{1}{\bar{x}_n} \pm 1.96 \frac{1}{\sqrt{n} \bar{x}_n} .$$

Let us compute the ML estimate and the 95% confidence interval for the rate parameter for the waiting times at the Orbiter bus-stop (see labwork 181). The sample mean $\bar{x}_{132} = 9.0758$ and the ML estimate is:

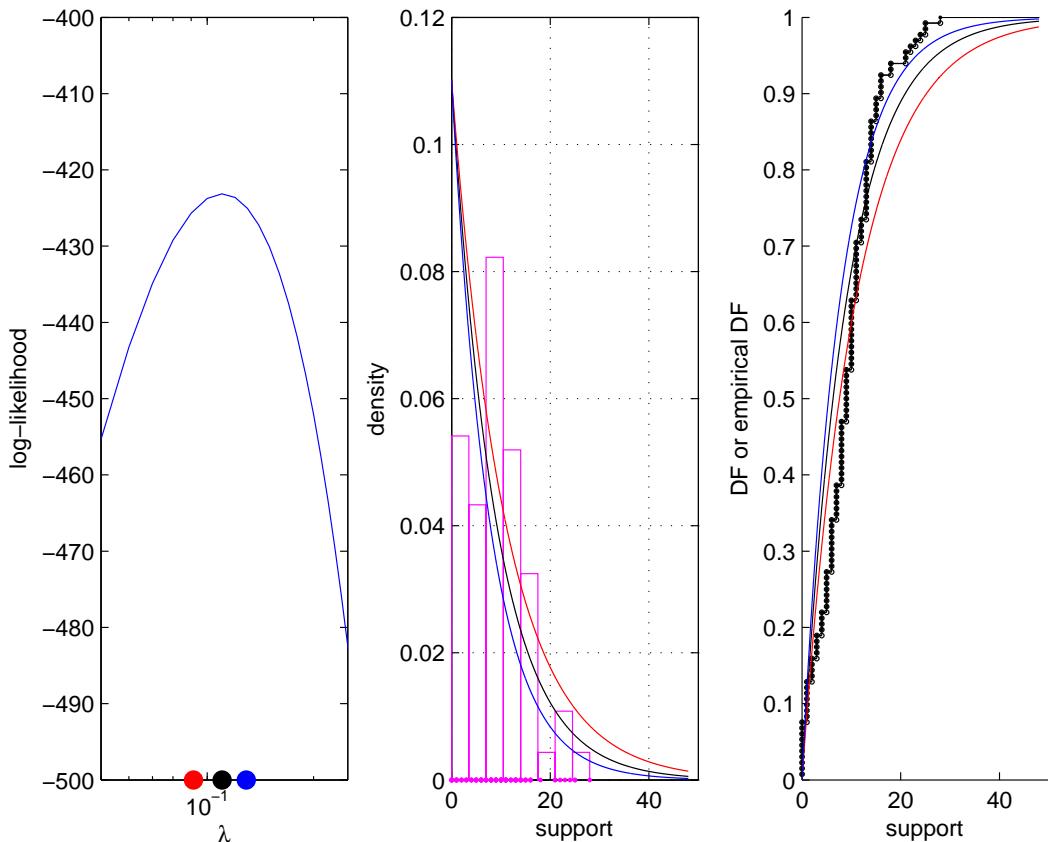
$$\widehat{\lambda}_{132} = 1/\bar{x}_{132} = 1/9.0758 = 0.1102 ,$$

and the 95% confidence interval is:

$$C_n = \hat{\lambda}_{132} \pm 1.96 \frac{\hat{\lambda}_{132}}{\sqrt{132}} = \frac{1}{\bar{x}_{132}} \pm 1.96 \frac{1}{\sqrt{132} \bar{x}_{132}} = 0.1102 \pm 1.96 \cdot 0.0096 = [0.0914, 0.1290] .$$

Notice how poorly the exponential PDF $f(x; \hat{\lambda}_{132} = 0.1102)$ and the DF $F(x; \hat{\lambda}_{132} = 0.1102)$ based on the MLE fits with the histogram and the empirical DF, respectively, in Figure 12.5, despite taking the the confidence interval into account. This is a further indication of the inadequacy of our parametric model.

Figure 12.5: Plot of $\log(L(\lambda))$ as a function of the parameter λ , the MLE $\hat{\lambda}_{132} = 0.1102$ and 95% confidence interval $C_n = [0.0914, 0.1290]$ for Fenemore-Wang Orbiter Waiting Times Experiment from STAT 218 S2 2007. The PDF and the DF at (1) the MLE 0.1102 (black), (2) lower 95% confidence bound 0.0914 (red) and (3) upper 95% confidence bound 0.1290 (blue) are compared with a histogram and the empirical DF.



Labwork 188 (Maximum likelihood estimation for Orbiter bus-stop) The above analysis was undertaken with the following M-file:

```
ExponentialMLECIOrbiter.m
OrbiterData; % load the Orbiter Data sampleTimes
% L = Log Likelihood of data x as a function of parameter lambda
L=@(lambda)sum(log(lambda*exp(-lambda * sampleTimes)));
LAMBDA=[0.01:0.01:1]; % sample some values for lambda
```

```

clf;
subplot(1,3,1);
semilogx(LAMBDA, arrayfun(L,LAMBDA)); % plot the Log Likelihood function
axis([0.05 0.25 -500 -400])
SampleMean = mean(sampleTimes) % sample mean
MLE = 1/SampleMean % ML estimate is 1/mean
n=length(sampleTimes) % sample size
StdErr=1/(sqrt(n)*SampleMean) % Standard Error
MLE95CI=[MLE-(1.96*StdErr), MLE+(1.96*StdErr)] % 95 % CI
hold on; % plot the MLE
plot([MLE], [-500], 'k.', 'MarkerSize',25);
plot([MLE95CI(1)], [-500], 'r.', 'MarkerSize',25);
plot([MLE95CI(2)], [-500], 'b.', 'MarkerSize',25);
ylabel('log-likelihood'); xlabel('\lambda');
subplot(1,3,2); % plot a histogram estimate
histogram(sampleTimes,1,[min(sampleTimes),max(sampleTimes)],'m',2);
hold on; TIMES=[0.00001:0.01:max(sampleTimes)+20]; % points on support
% plot PDF at MLE and 95% CI to compare with histogram
plot(TIMES,MLE*exp(-MLE*TIMES), 'k-')
plot(TIMES,MLE*exp(-MLE95CI(1)*TIMES), 'r-'); plot(TIMES,MLE*exp(-MLE95CI(2)*TIMES), 'b-')
% compare the empirical DF to the best fitted DF at MLE and 95% CI
subplot(1,3,3)
ECDF(sampleTimes,5,0.0,20); hold on; plot(TIMES,ExponentialCdf(TIMES,MLE), 'k-');
plot(TIMES,ExponentialCdf(TIMES,MLE95CI(1)), 'r-'); plot(TIMES,ExponentialCdf(TIMES,MLE95CI(2)), 'b-')
ylabel('DF or empirical DF'); xlabel('support');

```

A call to the script generates Figure 12.5 and the following output of the sample mean, MLE, sample size, standard error and the 95% confidence interval.

```

>> ExponentialMLECIOrbiter
SampleMean =    9.0758
MLE =      0.1102
n =      132
StdErr =    0.0096
MLE95CI =    0.0914    0.1290

```

Labwork 189 (Maximum likelihood estimation for your bus-stop) Recall labwork 65 where you modeled the arrival of buses using $\text{Exponential}(\lambda^* = 0.1)$ distributed inter-arrival time with a mean of $1/\lambda^* = 10$ minutes. Using the data of these seven inter-arrival times at your ID-seeded bus stop and pretending that you do not know the true λ^* , report (1) the ML estimate of λ^* , (2) 95% confidence interval for it and (3) whether the true value $\lambda^* = 1/10$ is engulfed by your confidence interval.

12.5 Delta Method

A more general estimation problem of interest concerns some function of the parameter $\theta \in \Theta$, say $g(\theta) = \psi : \Theta \rightarrow \Psi$. So, $g(\theta) = \psi$ is a function from the parameter space Θ to Ψ . Thus, we are not only interested in estimating the fixed and possibly unknown $\theta^* \in \Theta$ using the ML estimator $\hat{\Theta}_n$ and its ML estimate $\hat{\theta}_n$, but also in estimating $\psi^* = g(\theta^*) \in \Psi$ via an estimator $\hat{\Psi}_n$ and its estimate $\hat{\psi}_n$. We exploit the equivariance property of the ML estimator $\hat{\Theta}_n$ of θ^* and use the Delta method to find the following analytically:

1. The ML estimator of $\psi^* = g(\theta^*) \in \Psi$ is

$$\hat{\Psi}_n = g(\hat{\Theta}_n)$$

and its point estimate is

$$\widehat{\psi}_n = g(\widehat{\theta}_n)$$

2. Suppose $g(\theta) = \psi : \Theta \rightarrow \Psi$ is **any** smooth function of θ , i.e. g is differentiable, and $g'(\theta) := \frac{\partial}{\partial \theta} g(\theta) \neq 0$. Then, the distribution of the ML estimator $\widehat{\Psi}_n$ is asymptotically $\text{Normal}(\psi^*, \widehat{s}\text{e}_n(\widehat{\Psi}_n)^2)$, i.e.:

$$\frac{\widehat{\Psi}_n - \psi^*}{\widehat{s}\text{e}_n(\widehat{\Psi}_n)} \rightsquigarrow \text{Normal}(0, 1)$$

where the standard error $\widehat{s}\text{e}_n(\widehat{\Psi}_n)$ of the ML estimator $\widehat{\Psi}_n$ of the unknown quantity $\psi^* \in \Psi$ can be obtained from the standard error $\widehat{s}\text{e}_n(\widehat{\Theta}_n)$ of the ML estimator $\widehat{\Theta}_n$ of the parameter $\theta^* \in \Theta$, as follows:

$$\widehat{s}\text{e}_n(\widehat{\Psi}_n) = |g'(\widehat{\theta}_n)| \widehat{s}\text{e}_n(\widehat{\Theta}_n)$$

3. Using $\text{Normal}(\psi^*, \widehat{s}\text{e}_n(\widehat{\Psi}_n)^2)$, we can construct the estimate of an approximate Normal-based $1 - \alpha$ confidence interval for $\psi^* \in \Psi$:

$$C_n = [\underline{C}_n, \bar{C}_n] = \widehat{\psi}_n \pm z_{\alpha/2} \widehat{s}\text{e}_n(\widehat{\psi}_n)$$

Let us do an example next.

Example 190 Let $X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Bernoulli}(\theta^*)$. Let $\psi = g(\theta) = \log(\theta/(1-\theta))$. Suppose we are interested in producing a point estimate and confidence interval for $\psi^* = g(\theta^*)$. We can use the Delta method as follows:

First, the estimated standard error of the ML estimator of θ^* , as shown in Example 186, is

$$\widehat{s}\text{e}_n(\widehat{\Theta}_n) = \sqrt{\frac{\widehat{\theta}_n(1-\widehat{\theta}_n)}{n}}.$$

The ML estimator of ψ^* is:

$$\widehat{\Psi}_n = \log(\widehat{\Theta}_n/(1-\widehat{\Theta}_n))$$

and the ML estimate of ψ^* is:

$$\widehat{\psi}_n = \log(\widehat{\theta}_n/(1-\widehat{\theta}_n)).$$

Since, $g'(\theta) = 1/(\theta(1-\theta))$, by the Delta method, the estimated standard error of the ML estimator of ψ^* is:

$$\widehat{s}\text{e}_n(\widehat{\Psi}_n) = |g'(\widehat{\theta}_n)| (\widehat{s}\text{e}_n(\widehat{\Theta}_n)) = \frac{1}{\widehat{\theta}_n(1-\widehat{\theta}_n)} \sqrt{\frac{\widehat{\theta}_n(1-\widehat{\theta}_n)}{n}} = \frac{1}{\sqrt{n\widehat{\theta}_n(1-\widehat{\theta}_n)}} = \frac{1}{\sqrt{n\bar{x}_n(1-\bar{x}_n)}}.$$

An approximate 95% confidence interval for $\psi^* = \log(\theta^*/(1-\theta^*))$ is:

$$\widehat{\psi}_n \pm \frac{1.96}{\sqrt{n\widehat{\theta}_n(1-\widehat{\theta}_n)}} = \log(\widehat{\theta}_n/(1-\widehat{\theta}_n)) \pm \frac{1.96}{\sqrt{n\widehat{\theta}_n(1-\widehat{\theta}_n)}} = \log(\bar{x}_n/(1-\bar{x}_n)) \pm \frac{1.96}{\sqrt{n\bar{x}_n(1-\bar{x}_n)}}.$$

Example 191 (Delta Method for a Normal Experiment) Let us try the Delta method on a continuous RV. Let $X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Normal}(\mu^*, \sigma^{*2})$. Suppose that μ^* is known and σ^* is unknown. Let us derive the ML estimate $\hat{\psi}_n$ of $\psi^* = \log(\sigma^*)$ and a 95% confidence interval for it in 6 steps.

(1) First let us find the log-likelihood function $\ell(\sigma)$

$$\begin{aligned}
\ell(\sigma) := \log(L(\sigma)) &:= \log(L(x_1, x_2, \dots, x_n; \sigma)) = \log \left(\prod_{i=1}^n f(x_i; \sigma) \right) = \sum_{i=1}^n \log(f(x_i; \sigma)) \\
&= \sum_{i=1}^n \log \left(\frac{1}{\sigma \sqrt{2\pi}} \exp \left(-\frac{1}{2\sigma^2}(x_i - \mu)^2 \right) \right) \quad \because f(x_i; \sigma) \text{ in (6.11) is pdf of } \text{Normal}(\mu, \sigma^2) \text{ RV with known } \mu \\
&= \sum_{i=1}^n \left(\log \left(\frac{1}{\sigma \sqrt{2\pi}} \right) + \log \left(\exp \left(-\frac{1}{2\sigma^2}(x_i - \mu)^2 \right) \right) \right) \\
&= \sum_{i=1}^n \log \left(\frac{1}{\sigma \sqrt{2\pi}} \right) + \sum_{i=1}^n \left(-\frac{1}{2\sigma^2}(x_i - \mu)^2 \right) = n \log \left(\frac{1}{\sigma \sqrt{2\pi}} \right) + \left(-\frac{1}{2\sigma^2} \right) \sum_{i=1}^n (x_i - \mu)^2 \\
&= n \left(\log \left(\frac{1}{\sqrt{2\pi}} \right) + \log \left(\frac{1}{\sigma} \right) \right) - \left(\frac{1}{2\sigma^2} \right) \sum_{i=1}^n (x_i - \mu)^2 \\
&= n \log \left(\sqrt{2\pi}^{-1} \right) + n \log(\sigma^{-1}) - \left(\frac{1}{2\sigma^2} \right) \sum_{i=1}^n (x_i - \mu)^2 \\
&= -n \log \left(\sqrt{2\pi} \right) - n \log(\sigma) - \left(\frac{1}{2\sigma^2} \right) \sum_{i=1}^n (x_i - \mu)^2
\end{aligned}$$

(2) Let us find its derivative with respect to the unknown parameter σ next.

$$\begin{aligned}
\frac{\partial}{\partial \sigma} \ell(\sigma) &:= \frac{\partial}{\partial \sigma} \left(-n \log \left(\sqrt{2\pi} \right) - n \log(\sigma) - \left(\frac{1}{2\sigma^2} \right) \sum_{i=1}^n (x_i - \mu)^2 \right) \\
&= \frac{\partial}{\partial \sigma} \left(-n \log \left(\sqrt{2\pi} \right) \right) - \frac{\partial}{\partial \sigma} (n \log(\sigma)) - \frac{\partial}{\partial \sigma} \left(\left(\frac{1}{2\sigma^2} \right) \sum_{i=1}^n (x_i - \mu)^2 \right) \\
&= 0 - n \frac{\partial}{\partial \sigma} (\log(\sigma)) - \left(\frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2 \right) \frac{\partial}{\partial \sigma} (\sigma^{-2}) \\
&= -n\sigma^{-1} - \left(\frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2 \right) (-2\sigma^{-3}) = -n\sigma^{-1} + \sigma^{-3} \sum_{i=1}^n (x_i - \mu)^2
\end{aligned}$$

(3) Now, let us set the derivative equal to 0 and solve for σ .

$$\begin{aligned}
0 = -n\sigma^{-1} + \sigma^{-3} \sum_{i=1}^n (x_i - \mu)^2 &\iff n\sigma^{-1} = \sigma^{-3} \sum_{i=1}^n (x_i - \mu)^2 \iff n\sigma^{-1}\sigma^{+3} = \sum_{i=1}^n (x_i - \mu)^2 \\
&\iff n\sigma^{-1+3} = \sum_{i=1}^n (x_i - \mu)^2 \iff n\sigma^2 = \sum_{i=1}^n (x_i - \mu)^2 \\
&\iff \sigma^2 = \left(\sum_{i=1}^n (x_i - \mu)^2 \right) / n \iff \sigma = \sqrt{\sum_{i=1}^n (x_i - \mu)^2 / n}
\end{aligned}$$

Finally, we set the solution, i.e. the maximiser of the concave-down log-likelihood function of σ with a known and fixed μ^* as our ML estimate $\hat{\sigma}_n = \sqrt{\sum_{i=1}^n (x_i - \mu^*)^2 / n}$. Analogously, the ML estimator

of σ^* is $\widehat{\Sigma}_n = \sqrt{\sum_{i=1}^n (X_i - \mu^*)^2/n}$. Don't confuse Σ , the upper-case sigma, with $\sum_{i=1}^n \bigcirc_i$, the summation over some \bigcirc_i 's. This is usually clear from the context.

(4) Next, let us get the estimated standard error $\widehat{s}\mathbf{e}_n$ for the estimator of σ^* via Fisher Information. The Log-likelihood function of σ , based on one sample from the $\text{Normal}(\mu, \sigma^2)$ RV with known μ is,

$$\log f(x; \sigma) = \log \left(\frac{1}{\sigma \sqrt{2\pi}} \exp \left(-\frac{1}{2\sigma^2} (x - \mu)^2 \right) \right) = -\log(\sqrt{2\pi}) - \log(\sigma) - \left(\frac{1}{2\sigma^2} \right) (x - \mu)^2$$

Therefore, in much the same way as in part (2) earlier,

$$\begin{aligned} \frac{\partial^2 \log f(x; \sigma)}{\partial^2 \sigma} &:= \frac{\partial}{\partial \sigma} \left(\frac{\partial}{\partial \sigma} \left(-\log(\sqrt{2\pi}) - \log(\sigma) - \left(\frac{1}{2\sigma^2} \right) (x - \mu)^2 \right) \right) \\ &= \frac{\partial}{\partial \sigma} (-\sigma^{-1} + \sigma^{-3}(x - \mu)^2) = \sigma^{-2} - 3\sigma^{-4}(x - \mu)^2 \end{aligned}$$

Now, we compute the Fisher Information of one sample as an expectation of the continuous RV X over $\mathbb{X} = (-\infty, \infty)$ with density $f(x; \sigma)$,

$$\begin{aligned} I_1(\sigma) &= - \int_{x \in \mathbb{X} = (-\infty, \infty)} \left(\frac{\partial^2 \log f(x; \sigma)}{\partial^2 \lambda} \right) f(x; \sigma) dx = - \int_{-\infty}^{\infty} (\sigma^{-2} - 3\sigma^{-4}(x - \mu)^2) f(x; \sigma) dx \\ &= \int_{-\infty}^{\infty} -\sigma^{-2} f(x; \sigma) dx + \int_{-\infty}^{\infty} 3\sigma^{-4}(x - \mu)^2 f(x; \sigma) dx \\ &= -\sigma^{-2} \int_{-\infty}^{\infty} f(x; \sigma) dx + 3\sigma^{-4} \int_{-\infty}^{\infty} (x - \mu)^2 f(x; \sigma) dx \\ &= -\sigma^{-2} + 3\sigma^{-4}\sigma^2 \quad \because \sigma^2 = \mathbf{V}(X) = \mathbf{E}(X - \mathbf{E}(X))^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x; \sigma) dx \\ &= -\sigma^{-2} + 3\sigma^{-4+2} = -\sigma^{-2} + 3\sigma^{-2} = 2\sigma^{-2} \end{aligned}$$

Therefore, the estimated standard error of the estimator of the unknown σ^* is

$$\widehat{s}\mathbf{e}_n(\widehat{\Sigma}_n) = \frac{1}{\sqrt{I_n(\widehat{\sigma}_n)}} = \frac{1}{\sqrt{nI_1(\widehat{\sigma}_n)}} = \frac{1}{\sqrt{n2\sigma^{-2}}} = \frac{\sigma}{\sqrt{2n}} .$$

(5) Given that $\psi = g(\sigma) = \log(\sigma)$, we derive the estimated standard error of $\psi^* = \log(\sigma^*)$ via the Delta method as follows:

$$\widehat{s}\mathbf{e}_n(\widehat{\Psi}_n) = |g'(\sigma)| \widehat{s}\mathbf{e}_n(\widehat{\Sigma}_n) = \left| \frac{\partial}{\partial \sigma} \log(\sigma) \right| \frac{\sigma}{\sqrt{2n}} = \frac{1}{\sigma} \frac{\sigma}{\sqrt{2n}} = \frac{1}{\sqrt{2n}} .$$

(6) Finally, the 95% confidence interval for ψ^* is $\widehat{\psi}_n \pm 1.96 \widehat{s}\mathbf{e}_n(\widehat{\Psi}_n) = \log(\widehat{\sigma}_n) \pm 1.96 \frac{1}{\sqrt{2n}}$.

Chapter 13

Maximum Likelihood Estimation for Multiparameter Models

13.1 Introduction

When two or more parameters index a statistical experiment we want to estimate the vector-valued parameter $\theta^* := (\theta_1^*, \dots, \theta_k^*)$. Here we will find the maximum likelihood estimates of vector-valued parameters.

The maximum likelihood estimator (MLE) of a possibly unknown but fixed parameter $\theta^* := (\theta_1^*, \dots, \theta_k^*)$ in a multi-parametric experiment, i.e. $\theta^* \in \Theta \subset \mathbb{R}^k$ with $1 < k < \infty$ is defined analogously to Definition 100 with the exception that we allow the parameter to be a vector. We take an excursion in multi-dimensional optimisation before finding the MLE of a parametric experiment involving two parameters.

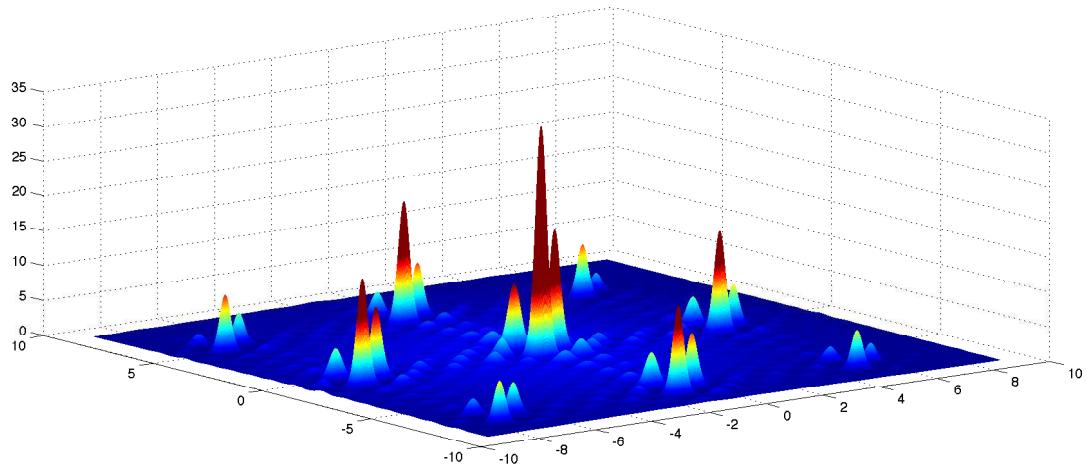
13.2 Practical Excursion in Multi-dimensional Optimisation

The basic idea involves multi-dimensional iterations that attempt to converge on a local maximum close to the starting vector $\theta^{(0)} \in \Theta$ (our initial guess). We can employ MATLAB's built-in function `fminsearch` to find the MLE of vector-valued parameters such as in the Lognormal model with two parameters, i.e. $\theta = (\lambda, \zeta) \in \Theta \subset \mathbb{R}^2$. The function `fminsearch` is similar to `fminbnd` except that it handles a given function of many variables, and the user specifies a starting vector $\theta^{(0)}$ rather than a starting interval. Thus, `fminsearch` tries to return a vector $\theta^{(*)}$ that is a local minimiser of, $-\log(L(x_1, x_2, \dots, x_n; \theta))$, the negative log-likelihood function of the vector-valued parameter θ , near this starting vector $\theta^{(0)}$. We illustrate the use of `fminsearch` on a more challenging target called the Levy density:

$$f(x, y) = \exp \left(-\frac{1}{50} \left(\left(\sum_{i=1}^5 i \cos((i-1)x + i) \right) \left(\sum_{j=1}^5 j \cos((j+1)y + j) \right) + (x + 1.42513)^2 + (y + 0.80032)^2 \right) \right) \quad (13.1)$$

`fminsearch` uses the simplex search method [Nelder, J.A., and Mead, R. 1965, Computer Journal, vol. 7, p. 308-313]. For an animation of the method and more details, please visit http://en.wikipedia.org/wiki/Nelder-Mead_method. An advantage of the method is that it does not use numerical (finite differencing) or analytical (closed-form expressions) gradients but relies on a direct

Figure 13.1: Plot of Levy density as a function of the parameter $(x, y) \in [-10, 10]^2$ scripted in Labwork 251.



search method. Briefly, the simplex algorithm tries to “tumble and shrink” a simplex towards the local valley of the function to be minimised. If k is the dimension of the parameter space or domain of the function to be optimised, a k -dimensional simplex is specified by its $k + 1$ distinct vertices each of dimension k . Thus, a simplex is a triangle in a two-dimensional space and a pyramid in a three-dimensional space. At each iteration of the algorithm:

1. A new point inside or nearby the current simplex is proposed.
2. The function’s value at the newly proposed point is compared with its values at the vertices of the simplex.
3. One of the vertices is typically replaced by the proposed point, giving rise to a new simplex.
4. The first three steps are repeated until the diameter of the simplex is less than the specified tolerance.

A major limitation of `fminsearch`, as demonstrated with the Levy target (encoded in Labwork 252) is that it can only give local solutions. The **global maximiser** of the Levy function $f(x, y)$ is $(-1.3069, -1.4249)$ and the **global maximum** is $f(-1.3069, -1.4249) = 33.8775$. For instance, if we start the search close to, say $(x^{(0)}, y^{(0)}) = (-1.3, -1.4)$, as shown below, then the simplex algorithm converges as desired to the solution $(-1.3068, -1.4249)$.

```
>> [params, fvalue, exitflag, output] = fminsearch('NegLevyDensity', [-1.3 -1.4], options)
params =   -1.3068   -1.4249
fvalue =  -33.8775
exitflag =      1
output =
    iterations: 24
    funcCount: 46
    algorithm: 'Nelder-Mead simplex direct search'
    message: [1x194 char]
```

However, if we start the search further away, say $(x^{(0)}, y^{(0)}) = (1.3, 1.4)$, as shown below, then the algorithm converges to the **local maximiser** $(1.1627, 1.3093)$ with a **local maximum** value of $f(1.1627, 1.3093) = 0.9632$, which is clearly smaller than the global maximum of 33.8775.

```
>> [params, fvalue, exitflag, output] = fminsearch('NegLevyDensity',[1.3 1.4],options)
params = 1.1627    1.3093
fvalue = -0.9632
exitflag = 1
output =
    iterations: 29
    funcCount: 57
    algorithm: 'Nelder-Mead simplex direct search'
    message: [1x194 char]
```

Therefore, we have to be extremely careful when using point-valued, iterative, local optimisation algorithms, implemented in floating-point arithmetic to find the global maximum. Other examples of such algorithms include:

- **Conjugate Gradient Method:**
http://en.wikipedia.org/wiki/Conjugate_gradient_method
- **Broyden-Fletcher-Goldfarb-Shanno (BFGS) method:**
http://en.wikipedia.org/wiki/BFGS_method
- **Simulated Annealing:**
http://en.wikipedia.org/wiki/Simulated_annealing

In general, we have no guarantee that the output of such local optimisation routines will indeed be the global optimum. In practice, you can start the search at several distinct starting points and choose the best local maximum from the lot.

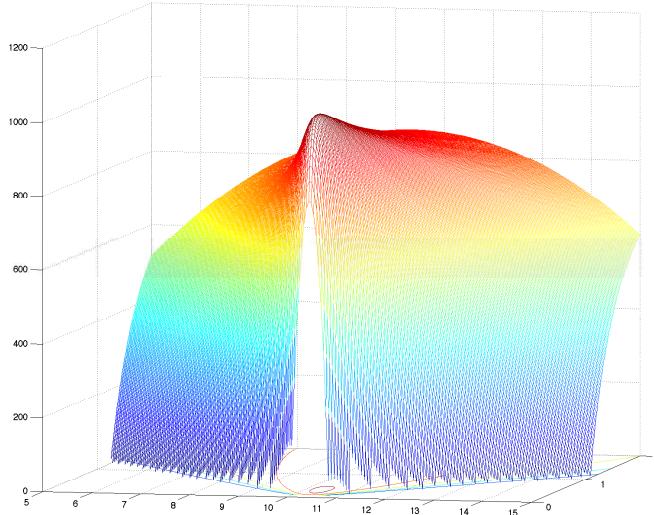
When the target function is “well-behaved,” i.e. uni-modal or single-peaked and not too spiky, the optimisation routine can be expected to perform well. Log-likelihood functions are often well-behaved. Let us generate 100 samples from an RV $C \sim \text{Lognormal}(\lambda^* = 10.36, \zeta^* = 0.26)$ by exponentiating the samples from the $\text{Normal}(10.36, 0.26^2)$ RV, and then compute the corresponding MMEs and MLEs for parameters (λ, ζ) using the formulae in Table 13.1.

```
>> rand('twister',001); % set the fundamental sampler
>> % draw 100 samples from the Lognormal(10.36,0.26) RV
>> Cs = exp(arrayfun(@(u)(Sample1NormalByNewRap(u,10.36,0.26^2)),rand(1,100)));
>> MLElambdahat = mean(log(Cs)) % maximum likelihood estimate of lambda
MLElambdahat = 10.3397
>> MLEzetahat = sqrt(mean( (log(Cs)-MLElambdahat) .^ 2)) % max. lkl. estimate of zeta
MLEzetahat = 0.2744
>> MMEzetaahat = sqrt(log(var(Cs)/(mean(Cs)^2) + 1)) % moment estimate of zeta
MMEzetaahat = 0.2624
>> MMElambdahat = log(mean(Cs))-(0.5*MMEzetaahat^2) % moment estimate of lambda
MMElambdahat = 10.3417
```

Let us try to apply the simplex algorithm to find the MLE numerically. We first encode the negative log-likelihood function of the parameters $(\lambda, \zeta) \in (0, \infty)^2$ for the given data x , as follows:

```
function l = NegLogNormalLogLkl(x,params)
% Returns the -log likelihood of [lambda zeta]=exp(params)
% for observed data vector x=(x_1,...,x_n) ~ IID LogNormal(lambda, zeta).
% We define lambda and zeta as exp(params) to allow for unconstrained
```

Figure 13.2: Plot of the “well-behaved” (uni-modal and non-spiky) $\log(L((x_1, x_2, \dots, x_{100}); \lambda, \zeta))$, based on 100 samples $(x_1, x_2, \dots, x_{100})$ drawn from the Lognormal($\lambda^* = 10.36, \zeta^* = 0.26$) as per Labwork 253.



```
% minimisation by fminsearch and respect the positive domain constraints
% for Lambda and zeta. So in the end we re-transform, i.e. [lambda zeta]=exp(params)
% lambda=params(1); zeta=params(1);
lambda=exp(params(1)); zeta=exp(params(2));
% minus Log-likelihood function
l = -sum(log((1 ./ (sqrt(2*pi)*zeta) .* x) .* exp((-1/(2*zeta^2))*(log(x)-lambda).^2)));
```

Here is how we can call `fminsearch` and find the MLE after the re-transformation.

```
>> [params, fvalue, exitflag, output] = ...
fminsearch(@(params)(NegLogNormalLogLkl(Cs,params)),[log(5), log(1)])
params =    2.3360   -1.2931
fvalue =  -1.0214e+03
exitflag =      1
output =
    iterations: 74
    funcCount: 131
    algorithm: 'Nelder-Mead simplex direct search'
    message: [1x194 char]
>> % But we want exp(params) since we defined lambda and zeta as exp(params)
exp(params)
ans =    10.3397    0.2744
```

Note that the MLEs $(\hat{\lambda}_{100}, \hat{\zeta}_{100}) = (10.3397, 0.2744)$ from 74 iterations or “tumbles” of the ‘Nelder-Mead simplex (triangle)’ and the MLEs agree well with the direct evaluations `MLElambdahat` and `MLEzetahat` based on the formulae in Table 13.1.

Summarizing Table of Point Estimators

Using the sample mean \bar{X}_n and sample standard deviation S_n defined in (5.1) and (5.5), respectively, we summarise the two point estimators of the parameters of some common distributions below. For

some cases, the MLE is the same as the MME (method of moments) and can be solved analytically.

Table 13.1: Summary of the Method of Moment Estimator (MME) and the Maximum Likelihood Estimator (MLE) for some IID Experiments.

Statistical Experiment	MLE	MME
$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Bernoulli}(\theta)$	$\hat{\theta} = \bar{X}_n$	same as MLE
$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Exponential}(\lambda)$	$\hat{\lambda} = 1/\bar{X}_n$	same as MLE
$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Normal}(\mu, \sigma^2)$	$\hat{\mu} = \bar{X}_n, \hat{\sigma} = \sqrt{\frac{n-1}{n} S_n^2}$	$\hat{\mu} = \bar{X}_n, \hat{\sigma} = S_n$
$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Lognormal}(\lambda, \zeta)$	$\hat{\lambda} = \frac{1}{n} \sum_{i=1}^n \log(X_i)$ $\hat{\zeta} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(X_i) - \hat{\lambda})^2}$	$\hat{\lambda} = \log(\bar{X}_n) - \frac{1}{2} \hat{\zeta}^2$ $\hat{\zeta} = \sqrt{\log(S_n^2/\bar{X}_n^2 + 1)}$

13.3 Confidence Sets for Multiparameter Models

We will extend the Fisher Information and Delta method to models with more than one parameter:

$$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} f(x; \theta^*), \quad \theta^* := (\theta_1^*, \theta_2^*, \dots, \theta_k^*) \in \Theta \subset \mathbb{R}^k .$$

Let, the ML estimator of the fixed and possibly unknown vector-valued parameter θ^* be:

$$\widehat{\Theta}_n := \left(\widehat{\Theta}_{1,n}, \widehat{\Theta}_{2,n}, \dots, \widehat{\Theta}_{k,n} \right), \quad \widehat{\Theta}_n := \widehat{\Theta}_n(X_1, X_2, \dots, X_n) : \mathbb{X}_n \rightarrow \Theta$$

and the ML estimate based on n observations x_1, x_2, \dots, x_n be:

$$\widehat{\theta}_n := \left(\widehat{\theta}_{1,n}, \widehat{\theta}_{2,n}, \dots, \widehat{\theta}_{k,n} \right), \quad \widehat{\theta}_n := \widehat{\theta}_n(x_1, x_2, \dots, x_n) \in \Theta .$$

Let the log-likelihood function and its Hessian matrix $H = (H_{i,j})_{i,j=1,2,\dots,k}$ of partial derivatives be:

$$\ell_n(\theta) := \ell_n(\theta_1, \theta_2, \dots, \theta_k) := \sum_{i=1}^n \log(f(x_i; (\theta_1, \theta_2, \dots, \theta_k))), \quad H_{i,j} := \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} \ell_n(\theta_1, \theta_2, \dots, \theta_k) ,$$

respectively, provided the log-likelihood function is sufficiently smooth.

Definition 103 (Fisher Information Matrix) The Fisher Information matrix is:

$$I_n(\theta) := I_n(\theta_1, \theta_2, \dots, \theta_k) = - \begin{bmatrix} \mathbf{E}_\theta(H_{1,1}) & \mathbf{E}_\theta(H_{1,2}) & \cdots & \mathbf{E}_\theta(H_{1,k}) \\ \mathbf{E}_\theta(H_{2,1}) & \mathbf{E}_\theta(H_{2,2}) & \cdots & \mathbf{E}_\theta(H_{2,k}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{E}_\theta(H_{k,1}) & \mathbf{E}_\theta(H_{k,2}) & \cdots & \mathbf{E}_\theta(H_{k,k}) \end{bmatrix} \quad (13.2)$$

and its matrix inverse is denoted by $I_n^{-1}(\theta)$.

Proposition 104 (Asymptotic Normality of MLE in Multiparameter Models) Let

$$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} f(x_1; \theta_1^*, \theta_2^*, \dots, \theta_k^*), \quad \theta^* = (\theta_1^*, \theta_2^*, \dots, \theta_k^*) \in \Theta \subset \mathbb{R}^k,$$

for some fixed and possibly unknown $\theta^* \in \Theta \subset \mathbb{R}^k$. Then, under appropriate regularity conditions:

$$\widehat{\Theta}_n := (\widehat{\Theta}_{1,n}, \widehat{\Theta}_{2,n}, \dots, \widehat{\Theta}_{k,n}) \rightsquigarrow \text{Normal}(\theta^*, I_n^{-1})$$

In other words, the vector-valued estimator $\widehat{\Theta}_n$ converges in distribution to the multivariate Normal distribution centred at the unknown parameter θ^* with the variance-covariance matrix given by inverse Fisher Information matrix I_n^{-1} . Furthermore, let $I_n^{-1}(j,j)$ denote the j^{th} diagonal entry of I_n^{-1} . In this case:

$$\frac{\widehat{\Theta}_{j,n} - \theta_j^*}{\sqrt{I_n^{-1}(j,j)}} \rightsquigarrow \text{Normal}(0, 1)$$

and the approximate covariance of $\widehat{\Theta}_{i,n}$ and $\widehat{\Theta}_{j,n}$ is:

$$\text{Cov}(\Theta_{i,n}, \Theta_{j,n}) \approx I_n^{-1}(i,j).$$

Now, let us look at a way of obtaining ML estimates and confidence sets for functions of θ . Suppose the real-valued function $g(\theta) = \psi : \Theta \rightarrow \Psi$ maps points in the k -dimensional parameter space $\Theta \subset \mathbb{R}^k$ to points in $\Psi \subset \mathbb{R}$. Let the gradient of g be

$$\nabla g(\theta) := \nabla g(\theta_1, \theta_2, \dots, \theta_k) = \begin{pmatrix} \frac{\partial}{\partial \theta_1} g(\theta_1, \theta_2, \dots, \theta_k) \\ \frac{\partial}{\partial \theta_2} g(\theta_1, \theta_2, \dots, \theta_k) \\ \vdots \\ \frac{\partial}{\partial \theta_k} g(\theta_1, \theta_2, \dots, \theta_k) \end{pmatrix}.$$

Proposition 105 (Multiparameter Delta Method) Suppose:

1. $X_1, X_2, \dots, X_n \stackrel{IID}{\sim} f(x_1; \theta_1^*, \theta_2^*, \dots, \theta_k^*), \quad \theta^* = (\theta_1^*, \theta_2^*, \dots, \theta_k^*) \in \Theta \subset \mathbb{R}^k,$
2. Let $\widehat{\Theta}_n$ be a ML estimator of $\theta^* \in \Theta$ and let $\widehat{\theta}_n$ be its ML estimate, and
3. Let $g(\theta) = \psi : \Theta \rightarrow \Psi \subset \mathbb{R}$ be a smooth function such that $\nabla g(\widehat{\theta}_n) \neq 0$.

Then:

1. $\widehat{\Psi}_n = g(\widehat{\Theta}_n)$ is the ML estimator and $\widehat{\psi}_n = g(\widehat{\theta}_n)$ is the ML estimate of $\psi^* = g(\theta^*) \in \Psi$,
2. The standard error of the ML estimator of ψ^* is:

$$\widehat{\text{se}}_n(\widehat{\Psi}_n) = \sqrt{\left(\nabla g(\widehat{\theta}_n) \right)^T I_n^{-1}(\widehat{\theta}_n) \left(\nabla g(\widehat{\theta}_n) \right)},$$

3. The ML estimator of ψ^* is asymptotically normal, i.e.:

$$\frac{\widehat{\Psi}_n - \psi^*}{\widehat{\text{se}}_n(\widehat{\Psi}_n)} \rightsquigarrow \text{Normal}(0, 1),$$

4. And a $1 - \alpha$ confidence interval for ψ^* is:

$$\hat{\psi}_n \pm z_{\alpha/2} \widehat{se}_n(\hat{\Psi}_n)$$

Let us put the theory to practice in the problem of estimating the coefficient of variation from samples of size n from an RV.

Example 192 (Estimating the Coefficient of Variation of a $\text{Normal}(\mu^*, \sigma^{*2})$ RV) Let

$$\psi^* = g(\mu^*, \sigma^*) = \sigma^*/\mu^*, \quad X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Normal}(\mu^*, \sigma^{*2}).$$

We do not know the fixed parameters (μ^*, σ^*) and are interested in estimating the coefficient of variation ψ^* based on n IID samples x_1, x_2, \dots, x_n . We have already seen that the ML estimates of μ^* and σ^* are:

$$\hat{\mu}_n = \bar{x}_n := \frac{1}{n} \sum_{i=1}^n x_i, \quad \hat{\sigma}_n = s_n := \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_n)^2}.$$

Thus, the ML estimate of $\psi^* = \sigma^*/\mu^*$ is:

$$\hat{\psi}_n = \frac{\hat{\sigma}_n}{\hat{\mu}_n} = \frac{s_n}{\bar{x}_n}$$

We can now derive the standard error of the ML estimator $\hat{\Psi}_n$ by first computing $I_n(\mu, \sigma)$, $I_n^{-1}(\mu, \sigma)$, and $\nabla g(\mu, \sigma)$. A careful computation shows that:

$$I_n(\mu, \sigma) = \begin{bmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{2n}{\sigma^2} \end{bmatrix}, \quad I_n^{-1}(\mu, \sigma) = \frac{1}{n} \begin{bmatrix} \sigma^2 & 0 \\ 0 & \frac{\sigma^2}{2} \end{bmatrix}, \quad \nabla g(\mu, \sigma) = \begin{pmatrix} -\frac{\sigma}{\mu^2} \\ \frac{1}{\mu} \end{pmatrix}.$$

Therefore, the standard error of interest is:

$$\widehat{se}_n(\hat{\Psi}_n) = \sqrt{\left(\nabla g(\hat{\theta}_n) \right)^T I_n^{-1}(\hat{\theta}_n) \left(\nabla g(\hat{\theta}_n) \right)} = \frac{1}{\sqrt{n}} \sqrt{\frac{1}{\hat{\mu}_n^4} + \frac{\hat{\sigma}_n^2}{2\hat{\mu}_n^2}}$$

and the 95% confidence interval for the unknown coefficient of variation ψ^* is:

$$\hat{\psi}_n \pm z_{\alpha/2} \widehat{se}_n(\hat{\Psi}_n) = \frac{s_n}{\bar{x}_n} \pm z_{\alpha/2} \left(\frac{1}{\sqrt{n}} \sqrt{\frac{1}{\hat{\mu}_n^4} + \frac{\hat{\sigma}_n^2}{2\hat{\mu}_n^2}} \right)$$

Let us get our hands dirty in the machine with Labwork 193 next.

Labwork 193 (Computing the coefficient of variation of a $\text{Normal}(\mu^*, \sigma^{*2})$ RV) Let us apply these results to $n = 100$ simulated samples from $\text{Normal}(100, 10^2)$ as follows.

```

n=100; % sample size
Mustar=100; % true mean
Sigmastar=10; % true standard deviation
rand('twister',67345); Us=rand(1,100); % draw some Uniform(0,1) samples
x=arrayfun(@(u)(Sample1NormalByNewRap(u,Mustar,Sigmastar^2)),Us); % get normal samples
Muhat=mean(x) % sample mean is MLE of Mustar
Sigmahat=std(x) % sample standard deviation is MLE for Sigmastar
Psihat=Sigmahat/Muhat % MLE of coefficient of variation std/mean
Sehat = sqrt((1/Muhat^4)+(Sigmahat^2/(2*Muhat^2)))/sqrt(n) % standar error estimate
ConfInt95=[Psihat-1.96*Sehat, Psihat+1.96*Sehat] % 1.96 since 1-alpha=0.95

```

```
>> CoeffOfVarNormal  
Muhat = 100.3117  
Sigmahat = 10.9800  
Psihat = 0.1095  
Sehat = 0.0077  
ConfInt95 = 0.0943    0.1246
```

Chapter 14

Non-parametric DF Estimation

So far, we have been interested in some estimation problems involved in parametric experiments. In parametric experiments, the parameter space Θ can have many dimensions, but these are finite. For example, in the n IID Bernoulli(θ^*) and the n IID Exponential(λ^*) experiments:

$$\begin{aligned} X_1, \dots, X_n &\stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta^*), & \theta^* \in \Theta = [0, 1] \subset \mathbb{R}^1, \\ X_1, \dots, X_n &\stackrel{\text{IID}}{\sim} \text{Exponential}(\lambda^*), & \lambda^* \in \Lambda = (0, \infty) \subset \mathbb{R}^1, \end{aligned}$$

the parameter spaces Θ and Λ are of dimension 1. Similarly, in the n IID Normal(μ, σ^2) and the n IID Lognormal(λ, ζ), experiments:

$$\begin{aligned} X_1, \dots, X_n &\stackrel{\text{IID}}{\sim} \text{Normal}(\mu, \sigma^2), & (\mu, \sigma^2) \in \Theta = (-\infty, +\infty) \times (0, +\infty) \subset \mathbb{R}^2 \\ X_1, \dots, X_n &\stackrel{\text{IID}}{\sim} \text{Lognormal}(\lambda, \zeta), & (\lambda, \zeta) \in \Theta = (0, +\infty) \times (0, +\infty) \subset \mathbb{R}^2 \end{aligned}$$

the parameter space is of dimension 2.

An experiment with an infinite dimensional parameter space Θ is said to be **non-parametric**. Next we consider a non-parametric experiment in which n IID samples are drawn according to some fixed and possibly unknown DF F^* from the space of **All Distribution Functions**:

$X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F^*, \quad F^* \in \Theta = \{\text{All DFs}\} := \{F(x; F) : F \text{ is a DF}\}$

where the DF $F(x; F)$ is indexed or parameterised by itself. Thus, the parameter space $\Theta = \{\text{All DFs}\}$ is the **infinite dimensional** space of **All DFs**. In this section, we look at estimation problems in non-parametric experiments with an infinite dimensional parameter space. That is, we want to estimate the DF F^* from which our IID data are drawn.

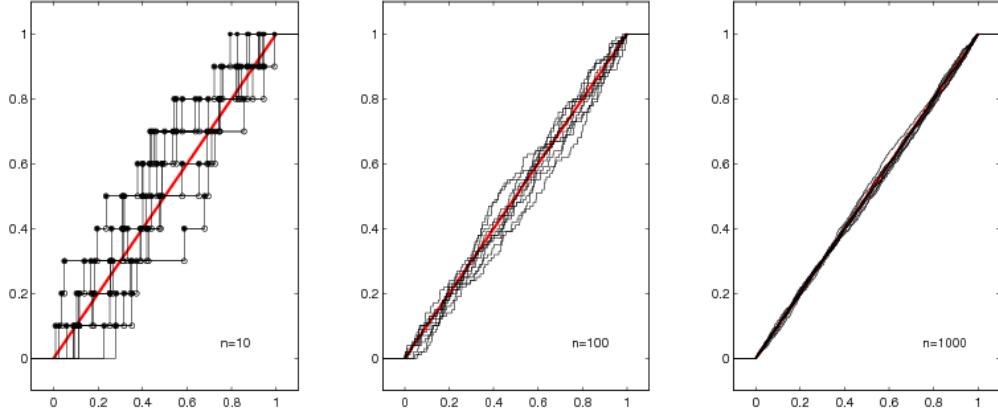
The next proposition is often referred to as the **fundamental theorem of statistics** and is at the heart of non-parametric inference, empirical processes, and computationally intensive bootstrap techniques. Recall Definition 35 of the n -sample empirical distribution function (EDF or ECDF) \hat{F}_n that assigns a probability mass of $1/n$ at each data point x_i :

$$\hat{F}_n(x) = \frac{\sum_{i=1}^n \mathbf{1}(X_i \leq x)}{n}, \quad \text{where} \quad \mathbf{1}(X_i \leq x) := \begin{cases} 1 & \text{if } x_i \leq x \\ 0 & \text{if } x_i > x \end{cases}$$

Proposition 106 (Gilvenko-Cantelli Theorem) Let $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F^*$. Then:

$$\sup_x |\hat{F}_n(x) - F^*(x)| \xrightarrow{P} 0.$$

Figure 14.1: Plots of ten distinct ECDFs \hat{F}_n based on 10 sets of n IID samples from Uniform(0, 1) RV X , as n increases from 10 to 100 to 1000. The DF $F(x) = x$ over $[0, 1]$ is shown in red. The script of Labwork 255 was used to generate this plot.



Heuristic Interpretation of the Gilvenko-Cantelli Theorem: As the sample size n increases, the empirical distribution function \hat{F}_n converges to the true DF F^* in probability, as shown in Figure 14.1.

Proposition 107 (The Dvoretzky-Kiefer-Wolfowitz (DKW) Inequality) Let $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F^*$. Then, for any $\epsilon > 0$:

$$P \left(\sup_x |\hat{F}_n(x) - F^*(x)| > \epsilon \right) \leq 2 \exp(-2n\epsilon^2) \quad (14.1)$$

Recall that $\sup(A)$ or supremum of a set $A \subset \mathbb{R}$ is the least upper bound of every element in A .

14.1 Estimating DF

Let $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F^*$, where F^* is some particular DF in the space of all possible DFs, i.e. the experiment is non-parametric. Then, based on the data sequence X_1, X_2, \dots, X_n we want to estimate F^* .

For any fixed value of x , the expectation and variance of the empirical DF (5.8) are:

$$\mathbf{E}(\hat{F}_n(x)) = F^*(x) \implies \text{bias}_n(\hat{F}_n(x)) = 0 \quad (14.2)$$

$$\mathbf{V}(\hat{F}_n(x)) = \frac{F^*(x)(1 - F^*(x))}{n} \implies \lim_{n \rightarrow \infty} \text{se}_n(\hat{F}_n(x)) = 0 \quad (14.3)$$

Therefore, by Proposition 96, the empirical DF evaluated at x , i.e. $\hat{F}_n(x)$ is an asymptotically consistent estimator of the DF evaluated at x , i.e. $F^*(x)$. More formally, (14.2) and (14.3), by Proposition 96, imply that for any fixed value of x :

$$\hat{F}_n(x) \xrightarrow{P} F^*(x).$$

We are interested in a point estimate of the entire DF F^* , i.e. $F^*(x)$ over all x . A point estimator $T_n = T_n(X_1, X_2, \dots, X_n)$ of a fixed and possibly unknown $F \in \{\text{All DFs}\}$ is the empirical DF \hat{F}_n .

This estimator has an asymptotically desirable property:

$$\sup_x |\hat{F}_n(x) - F^*(x)| \xrightarrow{P} 0$$

because of the Gilvenko-Cantelli theorem in Proposition 106. Thus, we can simply use \hat{F}_n , based on the realized data (x_1, x_2, \dots, x_n) , as a point estimate of F^* .

On the basis of the DKW inequality (14.1), we can obtain a $1 - \alpha$ confidence set or **confidence band** $C_n(x) := [\underline{C}_n(x), \bar{C}_n(x)]$ about our point estimate of F^* :

$$\begin{aligned}\underline{C}_n(x) &= \max\{\hat{F}_n(x) - \epsilon_n, 0\}, \\ \bar{C}_n(x) &= \min\{\hat{F}_n(x) + \epsilon_n, 1\}, \\ \epsilon_n &= \sqrt{\frac{1}{2n} \log\left(\frac{2}{\alpha}\right)}.\end{aligned}\tag{14.4}$$

It follows from (14.1) that for any fixed and possibly unknown F^* :

$$P(\underline{C}_n(x) \leq F^*(x) \leq \bar{C}_n(x)) \geq 1 - \alpha.$$

Let us look at a simple example next.

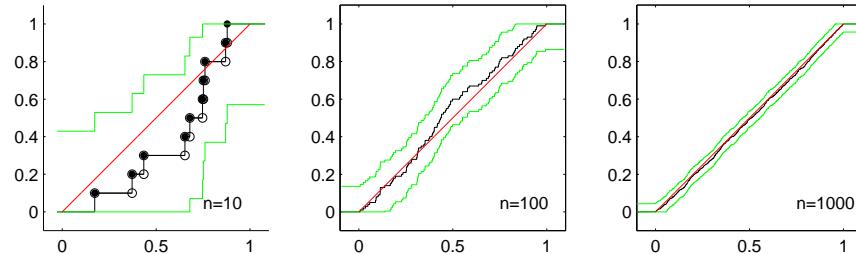
Labwork 194 (Estimating the DF of Uniform(0, 1) RV) Consider the problem of estimating the DF of Uniform(0, 1) RV U on the basis of $n=10$ samples. We use the function ECDF of Lab-work 247 and MATLAB's built-in function stairs to render the plots. Figure 14.2 was generated by PlotUniformECDFsConfBands.m given below.

PlotUniformECDFsConfBands.m

```
% script PlotUniformECDFsConfBands.m to plot the ECDF from 10 and 100 samples
% from Uniform(0,1) RV
rand('twister',76534); % initialize the Uniform(0,1) Sampler
N = 3; % 10^N is the maximum number of samples from Uniform(0,1) RV
u = rand(1,10^N); % generate 1000 samples from Uniform(0,1) RV U

% plot the ECDF from the first 10 samples using the function ECDF
for i=1:N
    SampleSize=10^i;
    subplot(1,N,i)
    % Get the x and y coordinates of SampleSize-based ECDF in x1 and y1 and
    % plot the ECDF using the function ECDF
    if (i==1) [x1 y1] = ECDF(u(1:SampleSize),2,0.2,0.2);
    else
        [x1 y1] = ECDF(u(1:SampleSize),0,0.1,0.1);
        stairs(x1,y1,'k');
    end
    % Note PlotFlag is 1 and the plot range of x-axis is
    % incremented by 0.1 or 0.2 on either side due to last 2 parameters to ECDF
    % being 0.1 or 0.2
    Alpha=0.05; % set alpha to 5% for instance
    Epsn = sqrt((1/(2*SampleSize))*log(2/Alpha)); % epsilon_n for the confidence band
    hold on;
    stairs(x1,max(y1-Epsn,zeros(1,length(y1))), 'g'); % lower band plot
    stairs(x1,min(y1+Epsn,ones(1,length(y1))), 'g'); % upper band plot
    axis([-0.1 1.1 -0.1 1.1]);
    axis square;
    x=[0:0.001:1];
    plot(x,x,'r'); % plot the DF of Uniform(0,1) RV in red
    LabelString=['n=' num2str(SampleSize)];
    text(0.75,0.05,LabelString)
    hold off;
end
```

Figure 14.2: The empirical DFs $\hat{F}_n^{(1)}$ from sample size $n = 10, 100, 1000$ (black), is the point estimate of the fixed and known DF $F(x) = x, x \in [0, 1]$ of Uniform(0, 1) RV (red). The 95% confidence band for each \hat{F}_n are depicted by green lines.



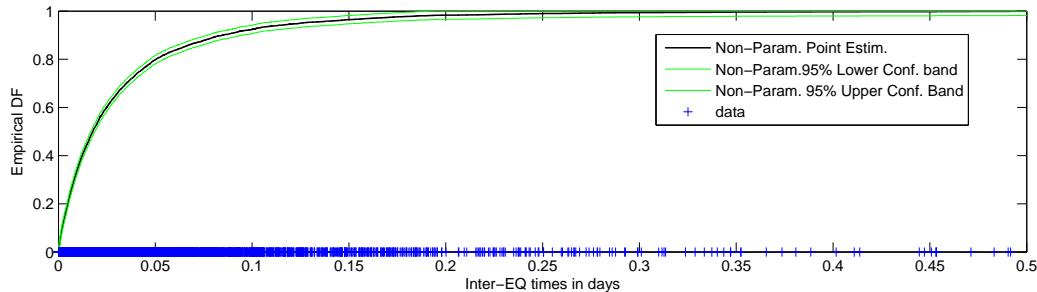
Next we look at a more interesting example involving real-world data.

Labwork 195 (Non-parametric Estimation of the DF of Times Between Earth Quakes)
Suppose that the 6,128 observed times between Earth quakes in NZ between 18-Jan-2008 02:23:44 and 18-Aug-2008 19:29:29 are:

$$X_1, \dots, X_{6128} \stackrel{IID}{\sim} F^*, \quad F^* \in \{\text{all DFs}\} .$$

Then the non-parametric point estimate of the unknown F^* is \hat{F}_{6128} , the ECDF of the inter earth quake times. We plot the non-parametric point estimate as well as the 95% confidence bands for F^* in Figure 14.3.

Figure 14.3: The empirical DF \hat{F}_{6128} for the inter earth quake times and the 95% confidence bands for the non-parametric experiment.



```
%>>> %% The columns in earthquakes.csv file have the following headings
%% CUSP_ID,LAT,LONG,NZMGE,NZMGN,ORI_YEAR,ORI_MONTH,ORI_DAY,ORI_HOUR,ORI_MINUTE,ORI_SECOND,MAG,DEPTH
EQ=dlmread('earthquakes.csv',';');
size(EQ) % report thr size of the matrix EQ
% Read help datenum -- converts time stamps into numbers in units of days
MaxD=max(datenum(EQ(:,6:11))); % maximum datenum
MinD=min(datenum(EQ(:,6:11))); % minimum datenum
% get an array of sorted time stamps of EQ events starting at 0
Times=sort(datenum(EQ(:,6:11))-MinD);
TimeDiff=diff(Times); % inter-EQ times = times between successive EQ events
n=length(TimeDiff); %sample size
clf % clear any current figures
%% Non-parametric Estimation X_1,X_2,...,X_132 ~ IID F
[x y] = ECDF(TimeDiff,0,0,0); % get the coordinates for empirical DF
```

```

stairs(x,y,'k','linewidth',1) % draw the empirical DF
hold on;
% get the 5% non-parametric confidence bands
Alpha=0.05; % set alpha to 5% for instance
Epsn = sqrt((1/(2*n))*log(2/Alpha)); % epsilon_n for the confidence band
stairs(x,max(y-Epsn,zeros(1,length(y))), 'g'); % non-parametric 95% lower confidence band
stairs(x,min(y+Epsn,ones(1,length(y))), 'g'); % non-parametric 95% upper confidence band
plot(TimeDiff,zeros(1,n),'+')
axis([0 0.5 0 1])
xlabel('Inter-EQ times in days'); ylabel('Empirical DF');
legend('Non-Param. Point Estim.', 'Non-Param. 95% Lower Conf. band', 'Non-Param. 95% Upper Conf. Band', 'data')

```

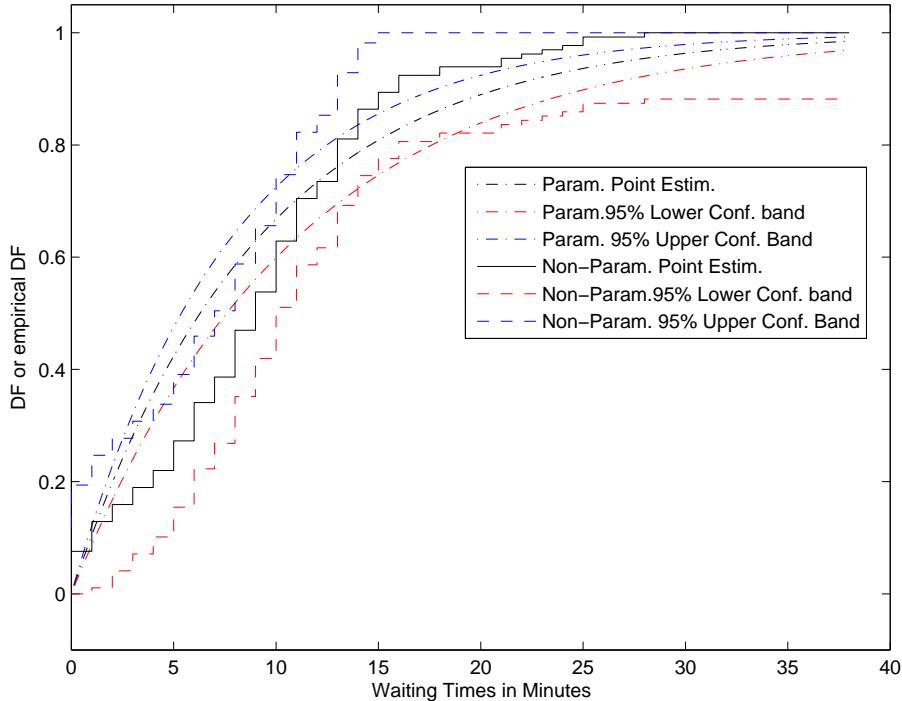
Recall the poor fit of the Exponential PDF at the MLE for the Orbiter waiting time data. We can attribute the poor fit to coarse resolution of the waiting time measurements in minutes and the rigid decaying form of the exponential PDFs. Let us revisit the Orbiter waiting time problem with our non-parametric estimator.

Labwork 196 (Non-parametric Estimation of Orbiter Waiting Times DF) Suppose that the waiting times at the Orbiter bus stop are:

$$X_1, \dots, X_{132} \stackrel{IID}{\sim} F^*, \quad F^* \in \{\text{all DFs}\}.$$

Then the non-parametric point estimate of F^* is \hat{F}_{132} , the ECDF of the 132 Orbiter waiting times. We compute and plot the non-parametric point estimate as well as the 95% confidence bands for

Figure 14.4: The empirical DF \hat{F}_{132} for the Orbiter waiting times and the 95% confidence bands for the non-parametric experiment.



the unknown DF F^* beside the parametric estimate and 95% confidence bands from Labwork 188. Clearly, the non-parametric estimate is preferable to the parametric one for this example. Notice how the non-parametric confidence bands do not contain the parametric estimate of the DF.

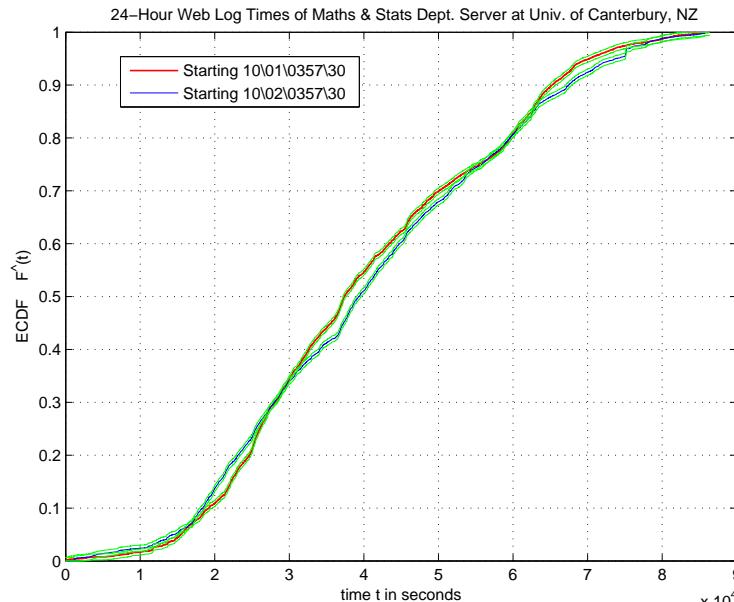
```

OrbiterData; % load the Orbiter Data sampleTimes
clf; % clear any current figures
%% Parametric Estimation X_1,X_2,...,X_132 ~ IID Exponential(lambda)
SampleMean = mean(sampleTimes) % sample mean
MLE = 1/SampleMean % ML estimate is 1/mean
n=length(sampleTimes) % sample size
StdErr=1/(sqrt(n)*SampleMean) % Standard Error
MLE95CI=[MLE-(1.96*StdErr), MLE+(1.96*StdErr)] % 95 % CI
TIMES=[0.00001:0.01:max(sampleTimes)+10]; % points on support
plot(TIMES,ExponentialCdf(TIMES,MLE),'k-.'); hold on; % Parametric Point Estimate
plot(TIMES,ExponentialCdf(TIMES,MLE95CI(1)), 'r--');% Normal-based Parametric 95% lower C.I.
plot(TIMES,ExponentialCdf(TIMES,MLE95CI(2)), 'b--');% Normal-based Parametric 95% upper C.I.
ylabel('DF or empirical DF');
xlabel('Waiting Times in Minutes');
%% Non-parametric Estimation X_1,X_2,...,X_132 ~ IID F
[x1 y1] = ECDF(sampleTimes,0,0.0,10); stairs(x1,y1,'k');% plot the ECDF
% get the 5% non-parametric confidence bands
Alpha=0.05; % set alpha to 5% for instance
Epsn = sqrt((1/(2*n))*log(2/Alpha)); % epsilon_n for the confidence band
stairs(x1,max(y1-Epsn,zeros(1,length(y1))), 'r--'); % non-parametric 95% lower confidence band
stairs(x1,min(y1+Epsn,ones(1,length(y1))), 'b--'); % non-parametric 95% upper confidence band
axis([0 40 -0.1 1.05]);
legend('Param. Point Estim.', 'Param. 95% Lower Conf. band', 'Param. 95% Upper Conf. Band',...
'Non-Param. Point Estim.', 'Non-Param. 95% Lower Conf. band', 'Non-Param. 95% Upper Conf. Band')

```

Example 197 First take a look at Data 256 to understand how the web login times to our Maths & Stats Department's web server (or requests to our WWW server) were generated. Figure 14.5 shows the login times in units of seconds over a 24 hour period starting at 0357 hours and 30 seconds (just before 4:00AM) on October 1st, 2007 (red line) and on October 2nd, 2007 (magenta). If we assume

Figure 14.5: The empirical DFs $\hat{F}_{n_1}^{(1)}$ with $n_1 = 56485$, for the web log times starting October 1, and $\hat{F}_{n_2}^{(2)}$ with $n_2 = 53966$, for the web log times starting October 2nd. Their 95% confidence bands are indicated by the green.



that some fixed and unknown DF $F^{(1)}$ specifies the distribution of login times for October 1st data and another DF $F^{(2)}$ for October 2nd data, then the non-parametric point estimates of $F^{(1)}$ and

$F^{(2)}$ are simply the empirical DFs $\hat{F}_{n_1}^{(1)}$ with $n_1 = 56485$ and $\hat{F}_{n_2}^{(2)}$ with $n_2 = 53966$, respectively, as depicted in Figure 14.5. See the script of `WebLogDataProc.m` in Data 256 to appreciate how the ECDF plots in Figure 14.5 were made.

14.2 Plug-in Estimators of Statistical Functionals

Recall from Chapter 5 that a **statistical functional** is simply any function of the DF F . For example, the median $T(F) = F^{[-1]}(1/2)$ is a statistical functional. Thus, $T(F) : \{\text{All DFs}\} \rightarrow \mathbb{T}$, being a map or function from the space of DFs to its range \mathbb{T} , is a functional. The idea behind the plug-in estimator for a statistical functional is simple: just plug-in the point estimate \hat{F}_n instead of the unknown DF F^* to estimate the statistical functional of interest.

Definition 108 (Plug-in estimator) Suppose, $X_1, \dots, X_n \stackrel{IID}{\sim} F^*$. The plug-in estimator of a statistical functional of interest, namely, $T(F^*)$, is defined by:

$$\hat{T}_n := \hat{T}_n(X_1, \dots, X_n) = T(\hat{F}_n) .$$

Definition 109 (Linear functional) If $T(F) = \int r(x)dF(x)$ for some function $r(x) : \mathbb{X} \rightarrow \mathbb{R}$, then T is called a **linear functional**. Thus, T is linear in its arguments:

$$T(aF + a'F') = aT(F) + a'T(F') .$$

Proposition 110 (Plug-in Estimator of a linear functional) The plug-in estimator for a linear functional $T = \int r(x)dF(x)$ is:

$$T(\hat{F}_n) = \int r(x)d\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n r(X_i) .$$

Some specific examples of statistical linear functionals we have already seen include:

1. The **mean** of RV $X \sim F$ is a function of the DF F :

$$T(F) = \mathbf{E}(X) = \int x dF(x) .$$

2. The **variance** of RV $X \sim F$ is a function of the DF F :

$$T(F) = \mathbf{E}(X - \mathbf{E}(X))^2 = \int (x - \mathbf{E}(X))^2 dF(x) .$$

3. The **value of DF at a given $x \in \mathbb{R}$** of RV $X \sim F$ is also a function of DF F :

$$T(F) = F(x) .$$

4. The q^{th} **quantile** of RV $X \sim F$:

$$T(F) = F^{[-1]}(q) \quad \text{where } q \in [0, 1] .$$

5. The **first quartile** or the 0.25^{th} **quantile** of the RV $X \sim F$:

$$T(F) = F^{[-1]}(0.25) .$$

6. The **median** or the **second quartile** or the 0.50^{th} **quantile** of the RV $X \sim F$:

$$T(F) = F^{[-1]}(0.50) .$$

7. The **third quartile** or the 0.75^{th} **quantile** of the RV $X \sim F$:

$$T(F) = F^{[-1]}(0.75) .$$

Labwork 198 (Plug-in Estimate for Median of Web Login Data) Compute the plug-in estimates for the median for each of the data arrays:

WebLogSeconds20071001035730 and WebLogSeconds20071002035730

that can be loaded into memory by following the commands in the first 13 lines of the script file `WebLogDataProc.m` of Data 256.

Labwork 199 (Plug-in Estimates of Times Between Earth Quakes) Compute the plug-in estimates for the median and mean time in minutes between earth quakes in NZ using the data in `earthquakes.csv`.

```
%>> NZSIEQTimesPlugInEstimates.m
%% The columns in earthquakes.csv file have the following headings
%% CUSP_ID, LAT, LONG, NZMGE, NZMGN, ORI_YEAR, ORI_MONTH, ORI_DAY, ORI_HOUR, ORI_MINUTE, ORI_SECOND, MAG, DEPTH
EQ=dlmread('earthquakes.csv',','); % load the data in the matrix EQ
% Read help datenum -- converts time stamps into numbers in units of days
MaxD=max(datenum(EQ(:,6:11)));% maximum datenum
MinD=min(datenum(EQ(:,6:11)));% minimum datenum
% get an array of sorted time stamps of EQ events starting at 0
Times=sort(datenum(EQ(:,6:11))-MinD);
TimeDiff=diff(Times); % inter-EQ times = times between successive EQ events
n=length(TimeDiff); %sample size
PlugInMedianEstimate=median(TimeDiff) % plug-in estimate of median
PlugInMedianEstimateMinutes=PlugInMedianEstimate*24*60 % median estimate in minutes
PlugInMeanEstimate=mean(TimeDiff) % plug-in estimate of mean
PlugInMeanEstimateMinutes=PlugInMeanEstimate*24*60 % mean estimate in minutes
```

```
>> NZSIEQTimesPlugInEstimates
PlugInMedianEstimate =    0.0177
PlugInMedianEstimateMinutes =   25.5092
PlugInMeanEstimate =    0.0349
PlugInMeanEstimateMinutes =   50.2278
```

Note that any statistical functional can be estimated using the plug-in estimator. However, to produce a $1 - \alpha$ confidence set for the plug-in point estimate, we need bootstrap methods. The subject of next chapter.

Chapter 15

Bootstrap

The **bootstrap** is a statistical method for estimating standard errors and confidence sets of statistics, such as estimators.

15.1 Non-parametric Bootstrap for Confidence Sets

Let $T_n := T_n((X_1, X_2, \dots, X_n))$ be a statistic, i.e. any function of the data $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F^*$. Suppose we want to know its variance $\mathbf{V}_{F^*}(T_n)$, which clearly depends on the fixed and possibly unknown DF F^* .

If our statistic T_n is one with an analytically unknown variance, then we can use the bootstrap to estimate it. The bootstrap idea has the following two basic steps:

Step 1: Estimate $\mathbf{V}_{F^*}(T_n)$ with $\mathbf{V}_{\widehat{F}_n}(T_n)$.

Step 2: Approximate $\mathbf{V}_{\widehat{F}_n}(T_n)$ using simulated data from the “Bootstrap World.”

For example, if $T_n = \bar{X}_n$, in Step 1, $\mathbf{V}_{\widehat{F}_n}(T_n) = s_n^2/n$, where $s_n^2 = n^{-1} \sum_{i=1}^n (x_i - \bar{x}_n)^2$ is the sample variance and \bar{x}_n is the sample mean. In this case, Step 1 is enough. However, when the statistic T_n is more complicated (e.g. $T_n = \tilde{X}_n = F^{[-1]}(0.5)$), the sample median, then we may not be able to find a simple expression for $\mathbf{V}_{\widehat{F}_n}(T_n)$ and may need Step 2 of the bootstrap.

$$\begin{aligned} \text{Real World Data come from } & F^* \implies X_1, X_2, \dots, X_n \implies T_n((X_1, X_2, \dots, X_n)) = t_n \\ \text{Bootstrap World Data come from } & \widehat{F}_n \implies X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet \implies T_n((X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet)) = t_n^\bullet \end{aligned}$$

Observe that drawing an observation from the ECDF \widehat{F}_n is equivalent to drawing one point at random from the original data (think of the indices $[n] := \{1, 2, \dots, n\}$ of the original data X_1, X_2, \dots, X_n being drawn according to the equi-probable de Moivre($1/n, 1/n, \dots, 1/n$) RV on $[n]$). Thus, to simulate $X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet$ from \widehat{F}_n , it is enough to draw n observations with replacement from X_1, X_2, \dots, X_n .

In summary, the algorithm for Bootstrap Variance Estimation is:

Step 1: Draw $X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet \sim \widehat{F}_n$

Step 2: Compute $t_n^\bullet = T_n((X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet))$

Step 3: Repeat Step 1 and Step 2 B times, for some large B , say $B > 1000$, to get $t_{n,1}^{\bullet}, t_{n,2}^{\bullet}, \dots, t_{n,B}^{\bullet}$

Step 4: Several ways of estimating the bootstrap confidence intervals are possible:

- (a) The $1 - \alpha$ Normal-based bootstrap confidence interval is:

$$C_n = [T_n - z_{\alpha/2} \hat{se}_{boot}, T_n + z_{\alpha/2} \hat{se}_{boot}] ,$$

where the bootstrap-based standard error estimate is:

$$\hat{se}_{boot} = \sqrt{v_{boot}} = \sqrt{\frac{1}{B} \sum_{b=1}^B \left(t_{n,b}^{\bullet} - \frac{1}{B} \sum_{r=1}^B t_{n,r}^{\bullet} \right)^2}$$

- (b) The $1 - \alpha$ percentile-based bootstrap confidence interval is:

$$C_n = [\widehat{G}_n^{\bullet-1}(\alpha/2), \widehat{G}_n^{\bullet-1}(1 - \alpha/2)],$$

where \widehat{G}_n^{\bullet} is the empirical DF of the bootstrapped $t_{n,1}^{\bullet}, t_{n,2}^{\bullet}, \dots, t_{n,B}^{\bullet}$ and $\widehat{G}_n^{\bullet-1}(q)$ is the q^{th} sample quantile (5.9) of $t_{n,1}^{\bullet}, t_{n,2}^{\bullet}, \dots, t_{n,B}^{\bullet}$.

Labwork 200 (Confidence Interval for Median Estimate of Inter Earth Quake Times)
 Let us find the 95% Normal-based bootstrap confidence interval as well as the 95% percentile-based bootstrap confidence interval for our plug-in estimate of the median of inter earth quake times from Labwork 199 using the following script:

```
%> NZSIEQTimesMedianBootstrap.m
%% The columns in earthquakes.csv file have the following headings
%% CUSP_ID,LAT,LONG,NZMGE,NZMGN,ORI_YEAR,ORI_MONTH,ORI_DAY,ORI_HOUR,ORI_MINUTE,ORI_SECOND,MAG,DEPTH
EQ=dlmread('earthquakes.csv',','); % load the data in the matrix EQ
% Read help datenum -- converts time stamps into numbers in units of days
MaxD=max(datenum(EQ(:,6:11)));% maximum datenum
MinD=min(datenum(EQ(:,6:11)));% minimum datenum
% get an array of sorted time stamps of EQ events starting at 0
Times=sort(datenum(EQ(:,6:11))-MinD);
TimeDiff=diff(Times); % inter-EQ times = times between successive EQ events
n=length(TimeDiff) %sample size
Medianhat=median(TimeDiff)*24*60 % plug-in estimate of median in minutes
B= 1000 % Number of Bootstrap replications
% REPEAT B times: PROCEDURE of sampling n indices uniformly from 1,...,n with replacement
BootstrappedDataSet = TimeDiff([ceil(n*rand(n,B))]);
size(BootstrappedDataSet) % dimension of the BootstrappedDataSet
BootstrappedMedians=median(BootstrappedDataSet)*24*60; % get the statistic in Bootstrap world
% 95% Normal based Confidence Interval
SehatBoot = std(BootstrappedMedians); % std of BootstrappedMedians
% 95% C.I. for median from Normal approximation
ConfInt95BootNormal = [Medianhat-1.96*SehatBoot, Medianhat+1.96*SehatBoot]
% 95% Percentile based Confidence Interval
ConfInt95BootPercentile = ...
[qthSampleQuantile(0.025,sort(BootstrappedMedians)),...
 qthSampleQuantile(0.975,sort(BootstrappedMedians))]
```

We get the following output when we call the script file.

```
>> NZSIEQTimesMedianBootstrap
n = 6127
Medianhat = 25.5092
B = 1000
ans = 6127 1000
ConfInt95BootNormal = 24.4383 26.5800
ConfInt95BootPercentile = 24.4057 26.4742
```

Labwork 201 (Confidence Interval for Median Estimate of Web Login Data) Find the 95% Normal-based bootstrap confidence interval as well as the 95% percentile-based bootstrap confidence interval for our plug-in estimate of the median for each of the data arrays:

WebLogSeconds20071001035730 and WebLogSeconds20071002035730 .

Once again, the arrays can be loaded into memory by following the commands in the first 13 lines of the script file `WebLogDataProc.m` of Section 256. Produce four intervals (two for each data-set). Do the confidence intervals for the medians for the two days intersect?

```
>> WebLogDataProc % load in the data
>> Medianhat = median(WebLogSeconds20071001035730) % plug-in estimate of median
Medianhat =
      37416
>> % store the length of data array
>> K=length(WebLogSeconds20071001035730)
K =
      56485
>> B= 1000 % Number of Bootstrap replications
B =
      1000
>> BootstrappedDataSet = WebLogSeconds20071001035730([ceil(K*rand(K,B))]);
>> size(BootstrappedDataSet) % dimension of the BootstrappedDataSet
ans =
      56485      1000
>> BootstrappedMedians=median(BootstrappedDataSet); % get the statistic in Bootstrap world
>> % 95% Normal based Confidence Interval
>> SehatBoot = std(BootstrappedMedians); % std of BootstrappedMedians
>> % 95% C.I. for median from Normal approximation
>> ConfInt95BootNormal = [Medianhat-1.96*SehatBoot, Medianhat+1.96*SehatBoot]
ConfInt95BootNormal =
      37242      37590
>> % 95% Percentile based Confidence Interval
ConfInt95BootPercentile = ...
    [qthSampleQuantile(0.025,sort(BootstrappedMedians)),...
     qthSampleQuantile(0.975,sort(BootstrappedMedians))]
ConfInt95BootPercentile =
      37239      37554
```

Labwork 202 (Confidence interval for correlation) Here is a classical data set used by Bradley Efron (the inventor of bootstrap) to illustrate the method. The data are LSAT (Law School Admission Test in the U.S.A.) scores and GPA of fifteen individuals.

Thus, we have bivariate data of the form (Y_i, Z_i) , where $Y_i = \text{LSAT}_i$ and $Z_i = \text{GPA}_i$. For example, the first individual had an LSAT score of $y_1 = 576$ and a GPA of $z_1 = 3.39$ while the fifteenth individual had an LSAT score of $y_{15} = 594$ and a GPA of $z_{15} = 3.96$. We suppose that the bivariate data $(Y_i, Z_i) \stackrel{IID}{\sim} F^*$, such that $F^* \in \{\text{all bivariate DFs}\}$. This is a bivariate non-parametric experiment. The bivariate data are plotted in Figure .

The law school is interested in the correlation between the GPA and LSAT scores:

$$\theta^* = \frac{\int \int (y - \mathbf{E}(Y))(z - \mathbf{E}(Z))dF(y, z)}{\sqrt{\int (y - \mathbf{E}(Y))^2 dF(y) \int (z - \mathbf{E}(Z))^2 dF(z)}}$$

The plug-in estimate of the population correlation θ^* is the sample correlation:

$$\widehat{\Theta}_n = \frac{\sum_{i=1}^n (Y_i - \bar{Y}_n)(Z_i - \bar{Z}_n)}{\sqrt{\sum_{i=1}^n (Y_i - \bar{Y}_n)^2 \sum_{i=1}^n (Z_i - \bar{Z}_n)^2}}$$

LSATGPACorrBootstrap.m

```
%% Data from Bradley Efron's LSAT,GPA correlation estimation
LSAT=[576 635 558 578 666 580 555 661 651 605 653 575 545 572 594]; % LSAT data
GPA=[3.39 3.30 2.81 3.03 3.44 3.07 3.00 3.43 3.36 3.13 3.12 2.74 2.76 2.88 3.96]; % GPA data
```

```

subplot(1,2,1); plot(LSAT,GPA,'o'); xlabel('LSAT'); ylabel('GPA') % make a plot of the data
CC=corrcoef(LSAT,GPA); % use built-in function to compute sample correlation coefficient matrix
SampleCorrelation=CC(1,2) % plug-in estimate of the correlation coefficient
%% Bootstrap
B = 1000; % Number of Bootstrap replications
BootstrappedCCs=zeros(1,B); % initialise a vector of zeros
N = length(LSAT); % sample size
rand('twister',767671); % initialise the fundamental sampler
for b=1:B
    Indices=ceil(N*rand(N,1));% uniformly sample random indices from 1 to 15 with replacement
    BootstrappedLSAT = LSAT([Indices]); % bootstrapped LSAT data
    BootstrappedGPA = GPA([Indices]); % bootstrapped GPA data
    CCB=corrcoef(BootstrappedLSAT,BootstrappedGPA);
    BootstrappedCCs(b)=CCB(1,2); % sample correlation of bootstrapped data
end
%plot the histogram of Bootstrapped Sample Correlations with 15 bins
subplot(1,2,2);hist(BootstrappedCCs,15);xlabel('Bootstrapped Sample Correlations')

% 95% Normal based Confidence Interval
SehatBoot = std(BootstrappedCCs); % std of BootstrappedMedians
% 95% C.I. for median from Normal approximation
ConfInt95BootNormal = [SampleCorrelation-1.96*SehatBoot, SampleCorrelation+1.96*SehatBoot]
% 95% Percentile based Confidence Interval
ConfInt95BootPercentile = ...
[qthSampleQuantile(0.025,sort(BootstrappedCCs)),...
qthSampleQuantile(0.975,sort(BootstrappedCCs))]

```

We get the following output when we call the script file.

```

>> LSATGPACorrBootstrap
SampleCorrelation = 0.5459
ConfInt95BootNormal = 0.1770 0.9148
ConfInt95BootPercentile = 0.2346 0.9296

```

15.2 Parametric Bootstrap for Confidence Sets

The **bootstrap** may also be employed for estimating standard errors and confidence sets of statistics, such as estimators, even in a parametric setting. This is much easier than the the variance calculation based on Fisher Information and/or the Delta method.

The only difference in the **parametric bootstrap** as opposed to the **non-parametric bootstrap** we saw earlier is that our statistic of interest $T_n := T_n((X_1, X_2, \dots, X_n))$ is a function of the data:

$$X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} F(x; \theta^*) .$$

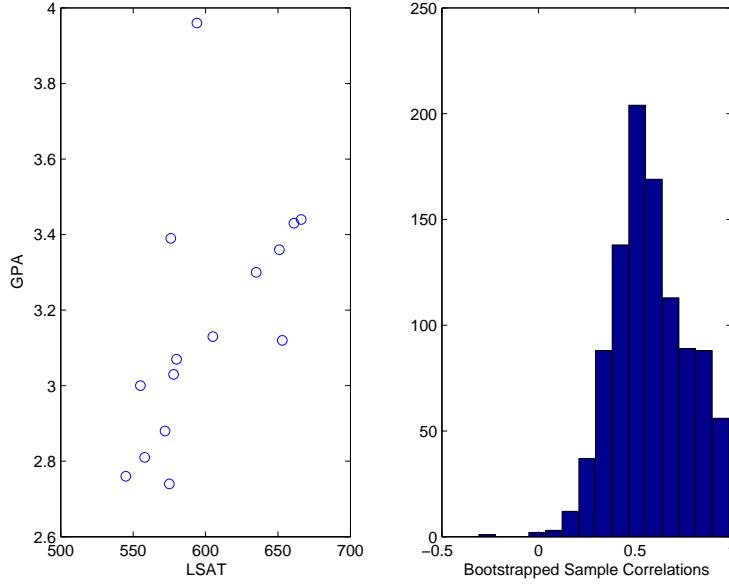
That is, our data come from a parametric distribution $F(x; \theta^*)$ and we want to know the variance of our statistic T_n , i.e. $\mathbf{V}_{\theta^*}(T_n)$.

The parametric bootstrap concept has the following two basic steps:

Step 1: Estimate $\mathbf{V}_{\theta^*}(T_n)$ with $\mathbf{V}_{\widehat{\theta}_n}(T_n)$, where $\widehat{\theta}_n$ is an estimate of θ^* based on maximum likelihood or the method of moments.

Step 2: Approximate $\mathbf{V}_{\widehat{\theta}_n}(T_n)$ using simulated data from the “Bootstrap World.”

Figure 15.1: Data from Bradley Efrons LSAT and GPA scores for fifteen individuals (left). The confidence interval of the sample correlation, the plug-in estimate of the population correlation, is obtained from the sample correlation of one thousand bootstrapped data sets (right).



For example, if $T_n = \bar{X}_n$, the sample mean, then in Step 1, $\mathbf{V}_{\hat{\theta}_n}(T_n) = n^{-1} \sum_{i=1}^n (x_i - \bar{x}_n)^2$ is the sample variance. Thus, in this case, Step 1 is enough. However, when the statistic T_n is more complicated, say $T_n = \tilde{X}_n = F^{[-1]}(0.5)$, the sample median, then we may not be able to write down a simple expression for $\mathbf{V}_{\hat{\theta}_n}(T_n)$ and may need Step 2 of the bootstrap.

$$\begin{aligned} \text{Real World Data come from } & F(\theta^*) \implies X_1, X_2, \dots, X_n \implies T_n((X_1, X_2, \dots, X_n)) = t_n \\ \text{Bootstrap World Data come from } & F(\hat{\theta}_n) \implies X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet \implies T_n((X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet)) = t_n^\bullet \end{aligned}$$

To simulate $X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet$ from $F(\hat{\theta}_n)$, we must have a simulation algorithm that allows us to draw IID samples from $F(\theta)$, for instance the inversion sampler. In summary, the algorithm for Bootstrap Variance Estimation is:

Step 1: Draw $X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet \sim F(\hat{\theta}_n)$

Step 2: Compute $t_n^\bullet = T_n((X_1^\bullet, X_2^\bullet, \dots, X_n^\bullet))$

Step 3: Repeat Step 1 and Step 2 B times, for some large B , say $B \geq 1000$, to get $t_{n,1}^\bullet, t_{n,2}^\bullet, \dots, t_{n,B}^\bullet$

Step 4: We can estimate the bootstrap confidence intervals in several ways:

(a) The $1 - \alpha$ normal-based bootstrap confidence interval is:

$$C_n = [T_n - z_{\alpha/2} \hat{s}e_{boot}, T_n + z_{\alpha/2} \hat{s}e_{boot}] ,$$

where the bootstrap-based standard error estimate is:

$$\hat{s}e_{boot} = \sqrt{v_{boot}} = \sqrt{\frac{1}{B} \sum_{b=1}^B \left(t_{n,b}^\bullet - \frac{1}{B} \sum_{r=1}^B t_{n,r}^\bullet \right)^2}$$

- (b) The $1 - \alpha$ percentile-based bootstrap confidence interval:

$$C_n = [\widehat{G}^{\bullet n}^{-1}(\alpha/2), \widehat{G}^{\bullet n}^{-1}(1 - \alpha/2)],$$

where $\widehat{G}^{\bullet n}$ is the empirical DF of the bootstrapped $t_{n,1}^{\bullet}, t_{n,2}^{\bullet}, \dots, t_{n,B}^{\bullet}$ and $\widehat{G}^{\bullet n}^{-1}(q)$ is the q^{th} sample quantile (5.9) of $t_{n,1}^{\bullet}, t_{n,2}^{\bullet}, \dots, t_{n,B}^{\bullet}$.

Let us apply the bootstrap method to the previous problem of estimating the standard error of the coefficient of variation from $n = 100$ samples from $\text{Normal}(100, 10^2)$ RV. The confidence intervals from bootstrap-based methods are similar to those from the Delta method.

CoeffOfVarNormalBoot.m

```
n=100; Mustar=100; Sigmistar=10; % sample size, true mean and standard deviation
rand('twister',67345);
x=arrayfun(@(u)(Sample1NormalByNewRap(u,Mustar,Sigmistar^2)),rand(n,1)); % normal samples
Muhat=mean(x) Sigmahat=std(x) Psihat=Sigmahat/Muhat % MLE of Mustar, Sigmistar and Psistar
Sehat = sqrt((1/Muhat^4)+(Sigmahat^2/(2*Muhat^2)))/sqrt(n) % standard error estimate
% 95% Confidence interval by Delta Method
ConfInt95DeltaMethod=[Psihat-1.96*Sehat, Psihat+1.96*Sehat] % 1.96 since 1-alpha=0.95
B = 1000; % B is number of bootstrap replications
% Step 1: draw n IID samples in Bootstrap World from Normal(Muhat,Sigmahat^2)
xBoot = arrayfun(@(u)(Sample1NormalByNewRap(u,Muhat,Sigmahat^2)),rand(n,B));
% Step 2: % Compute Bootstrapped Statistic Psihat
PsihatBoot = std(xBoot) ./ mean(xBoot);
% 95% Normal based Confidence Interval
SehatBoot = std(PsihatBoot); % std of PsihatBoot
ConfInt95BootNormal = [Psihat-1.96*SehatBoot, Psihat+1.96*SehatBoot] % 1-alpha=0.95
% 95% Percentile based Confidence Interval
ConfInt95BootPercentile = ...
[qthSampleQuantile(0.025,sort(PsihatBoot)),qthSampleQuantile(0.975,sort(PsihatBoot))]
```

```
>> CoeffOfVarNormal
Muhat = 100.3117
Sigmahat = 10.9800
Psihat = 0.1095
Sehat = 0.0077
ConfInt95DeltaMethod = 0.0943 0.1246
ConfInt95BootNormal = 0.0943 0.1246
ConfInt95BootPercentile = 0.0946 0.1249
```

15.3 Empirical distribution function



Let x_1, \dots, x_n be a random sample of size n . The *empirical distribution function* (EDF) of x_1, \dots, x_n is, for any real number t :

$$\hat{F}_n(t) = \frac{|\{x_i : x_i \leq t\}|}{n} = \frac{1}{n} \sum_{i=1}^n I_{(-\infty, t]}(x_i), \quad (15.1)$$

i.e. the proportion of sample points that are less than or equal to t .

Note that the EDF takes values between 0 and 1. Note also that if the sample comes from a continuous distribution, then the EDF takes a step of $1/n$ at each sample value; if the sample comes from a discrete distribution, the EDF may take steps that are multiples of $1/n$ at distinct sample values.

Let x_1^*, \dots, x_m^* be the distinct points in x_1, \dots, x_n , so that $m \leq n$, and let:

$$c_j = |\{x_i : x_i = x_j^*\}| = \sum_{i=1}^n I_{(x_j^*)}(x_i), \quad (15.2)$$

i.e. c_j is the number of sample values that are equal to x_j^* . The EDF can also be regarded as a discrete distribution that assigns a probability mass of $1/n$ to each of the observations, x_1, \dots, x_n , i.e. with an *empirical mass function* (EMF):

$$\hat{f}_n(x) = \frac{1}{n} \sum_{j=1}^m c_j I_{(x_j^*)}(x) = \begin{cases} c_j/n, & \text{if } x = x_j^*, \\ 0, & \text{otherwise.} \end{cases} \quad (15.3)$$

If the sample values are continuous, then $m = n$ and $c_1 = \dots = c_n = 1$.

Example 203 *Continuous data.* Suppose the observed values of 10 continuous random variables, arranged in increasing order, are:

$$\begin{array}{cccccc} 1.5937 & 1.4410 & 1.3362 & 0.6918 & 0.2944 \\ 0.5711 & 0.7143 & 0.8580 & 1.2540 & 1.6236 \end{array}$$

Example 204 *Discrete data.* Suppose the observed values of 10 discrete random variables, arranged in increasing order, are

$$1 \ 2 \ 2 \ 2 \ 2 \ 2 \ 3 \ 4 \ 5 \ 5$$

The EDF is an estimator for the distribution function and can therefore be used as a model for the distribution function. The EDF is a *nonparametric* model because its parameters are the sample points, x_1, \dots, x_n , and so the number of parameters increases as the sample size increases. The following two results show why the EDF is a good estimator for the distribution function.

Proposition 111 For any real number t :

$$E[\hat{F}_n(t)] = F(t), \quad (15.4)$$

and:

$$Var[\hat{F}_n(t)] = \frac{F(t)[1 - F(t)]}{n}, \quad (15.5)$$

and therefore $\hat{F}_n(t)$ is an unbiased and consistent estimator of $F(t)$.

Proof: Consider a fixed real number t . By definition:

$$\hat{F}_n(t) = \frac{1}{n} \sum_{i=1}^n I_{(-\infty, t]}(x_i).$$

The result follows by noting that $I_{(-\infty, t]}(x_i)$ is a Bernoulli random variable with parameter (“success” probability):

$$P[I_{(-\infty, t]}(x_i) = 1] = P(x_i \leq t) = F(t),$$

since x_i has distribution function F .

Proposition 112 (Glivenko-Cantelli theorem) For any real number t , $\hat{F}_n(t)$ converges almost certainly (i.e. with a probability of 1) and uniformly to $F(t)$, i.e.:

$$P(\lim_{n \rightarrow \infty} \sup_t |\hat{F}_n(t) - F(t)| = 0) = 1. \quad (15.6)$$

The Glivenko-Cantelli theorem says that the largest absolute difference between \hat{F}_x and F converges to 0 as n goes to infinity, with a probability of 1.

The next result allows us to construct a confidence band for the EDF.

Proposition 113 (Dvoretzky-Kiefer-Wolfowitz inequality) For any $\epsilon > 0$:

$$P(\sup_t |\hat{F}_n(t) - F(t)| > \epsilon) \leq 2 \exp(-2n\epsilon^2). \quad (15.7)$$

For $\alpha \in (0, 1)$, a $(1 - \alpha)$ confidence band for F should contain F with probability of at least $(1 - \alpha)$. In other words, the probability of F being outside the band is at most α . Hence, the bound in the Dvoretzky-Kiefer-Wolfowitz inequality should be α :

$$2 \exp(-2n\epsilon^2) = \alpha \Leftrightarrow \epsilon = \sqrt{\frac{1}{2n} \ln\left(\frac{2}{\alpha}\right)}, \quad (15.8)$$

i.e.:

$$P(\sup_t |\hat{F}_n(t) - F(t)| > \sqrt{\frac{1}{2n} \ln\left(\frac{2}{\alpha}\right)}) \leq \alpha,$$

or, for all t :

$$P(\max\{\hat{F}_n(t) - \sqrt{\frac{1}{2n} \ln\left(\frac{2}{\alpha}\right)}, 0\} \leq F(t) \leq \min\{\hat{F}_n(t) + \sqrt{\frac{1}{2n} \ln\left(\frac{2}{\alpha}\right)}, 1\})$$

Therefore, a $(1 - \alpha)$ confidence band for F is:

$$[\max\{\hat{F}_n(t) - \sqrt{\frac{1}{2n} \ln\left(\frac{2}{\alpha}\right)}, 0\}, \min\{\hat{F}_n(t) + \sqrt{\frac{1}{2n} \ln\left(\frac{2}{\alpha}\right)}, 1\}]. \quad (15.9)$$

Example 205 Referring to Example 4.1.3 where $n = 10$, a 0.95 confidence band for F is given by:

$$\begin{aligned} & [\max\{\hat{F}_{10}(t) - \sqrt{\frac{\ln 40}{20}}, 0\}, \min\{\hat{F}_{10}(t) + \sqrt{\frac{\ln 40}{20}}, 1\}] \\ & = [\max\{\hat{F}_{10}(t) - 0.4295, 0\}, \min\{\hat{F}_{10}(t) + 0.4295\}] \end{aligned}$$

The function edfplot, which is available from the course web-page, produced the figure below:

15.4 Nonparametric bootstrap

 Let x_1, \dots, x_n be a random sample and suppose that we wish to estimate an unknown quantity, θ , using an estimator $\hat{\theta}$ that is based on x_1, \dots, x_n . The performance of $\hat{\theta}$ as an estimator for θ can be assessed by its *bias*, *variance* and *mean squared error*.

(a) The *bias* of $\hat{\theta}$ is:

$$\text{Bias}(\hat{\theta}) = E(\hat{\theta}) - \theta. \quad (15.10)$$

(b) The *variance* of $\hat{\theta}$ is:

$$\text{Var}(\hat{\theta}) = E[(\hat{\theta} - E(\hat{\theta}))^2] = E(\hat{\theta}^2) - E(\hat{\theta})^2. \quad (15.11)$$

(c) The *mean squared error (MSE)* of $\hat{\theta}$ is:

$$\text{MSE}(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] = \text{Var}(\hat{\theta}) + \text{Bias}(\hat{\theta})^2. \quad (15.12)$$

When the distribution of x_1, \dots, x_n is known, one way to estimate the MSE of $\hat{\theta}$ is to use Monte Carlo simulation to generate N new random samples, each of size n , from which N new estimates of θ can be obtained:

$$\begin{aligned} \text{original sample : } & \{x_1, \dots, x_n\} \rightarrow \hat{\theta} \\ \text{generated samples : } & \{x_{1,1}, \dots, x_{1,n}\} \rightarrow \hat{\theta}_1 \\ & \vdots \\ & \{x_{N,1}, \dots, x_{N,n}\} \rightarrow \hat{\theta}_N. \end{aligned}$$

Here, $x_{j,i}$ denotes the i^{th} value in the j^{th} sample. An estimate of $\text{MSE}(\hat{\theta})$ is given by:

$$\text{MES}(\hat{\theta}) \approx \frac{1}{N} \sum_{j=1}^N (\hat{\theta}_j - \hat{\theta})^2. \quad (15.13)$$

Furthermore, if N is large, an approximate $(1-\alpha)$ confidence interval for θ is given by $(\hat{\theta}_{(\lceil N\alpha/2 \rceil)}, \hat{\theta}_{(\lceil N(1-\alpha/2) \rceil)})$. For example, if $N = 1000$, then an approximate 0.95 confidence interval is $(\hat{\theta}_{(25)}, \hat{\theta}_{(975)})$.

If the distribution of x_1, \dots, x_n is unknown, the idea behind the *nonparametric bootstrap* is to use the EDF as an estimate of F , and then perform Monte Carlo simulation with \hat{F}_n to estimate the MSE and to get approximate confidence intervals.

15.4.1 Bootstrap estimates of bias, variance and mean squared error

 Recall that the EDF can be regarded as a discrete distribution that assigns a probability mass of $1/n$ to each of the observations, x_1, \dots, x_n . Thus, using \hat{F}_n as an estimate of F , a random sample can be generated from \hat{F}_n by *randomly sampling with replacement* from x_1, \dots, x_n . A random sample of size n obtained in this way is called a *bootstrap sample*. The MSE of an estimator $\hat{\theta}$ can be obtained as follows:

(a) Compute $\hat{\theta}$ using x_1, \dots, x_n .

(b) Obtain N bootstrap samples, each of size n , by randomly sampling with replacement from x_1, \dots, x_n : $\{x_{1,1}, \dots, x_{1,n}\}, \{x_{2,1}, \dots, x_{2,n}\}, \dots, \{x_{N,1}, \dots, x_{N,n}\}$.

- (c) For each bootstrap sample, compute the *bootstrap estimate* of θ :

$$\begin{aligned}\{x_{1,1}, \dots, x_{1,n}\} &\rightarrow \hat{\theta}_1 \\ M \\ \{x_{N,1}, \dots, x_{N,n}\} &\rightarrow \hat{\theta}_N.\end{aligned}$$

- (d) Compute the mean of the bootstrap estimates:

$$\bar{\hat{\theta}} = \frac{1}{N} \sum_{j=1}^N \hat{\theta}_j. \quad (15.14)$$

- (e) Estimate the bias, variance and MSE by:

$$Bias(\hat{\theta}) \approx \bar{\hat{\theta}} - \hat{\theta}, \quad (15.15)$$

$$Var(\hat{\theta}) \approx \frac{1}{N} \sum_{j=1}^N (\hat{\theta}_j - \bar{\hat{\theta}})^2, \quad MSE(\hat{\theta}) \approx \frac{1}{N} \sum_{j=1}^N (\hat{\theta}_j - \bar{\hat{\theta}})^2. \quad (15.16)$$

Labwork 206 The table below contains the number, rounded to the nearest thousand, of open-close cycles of 20 door latches before they fail. Find the bias, variance and MSE of the sample mean.

Sample mean = 38.65										
7	11	15	16	20	22	24	25	29	33	
34	37	41	42	49	57	66	71	84	90	

j	j^{th} bootstrap sample										$\hat{\theta}_j$
1	22	57	42	16	24	11	20	7	41	90	
	90	16	66	25	90	66	25	24	84	66	44.1
2	90	37	15	84	29	57	57	57	11	49	
	41	57	84	71	37	20	29	84	7	15	46.6
3	49	84	29	41	57	11	49	42	90	34	
	71	33	41	84	49	66	20	20	29	15	45.7
M	M										M

```
x = load('latch.txt') % load data from text file and store in x
n = length(x); % determine number of data values
N = 100000; % number of bootstrap samples
nN = n * N;
xmean = mean(x) % mean of original data
xboot = randsample(x,nN,true); % sample with replacement nN values from x
xboot = reshape(xboot,n,N); % organise resampled values into N columns of n
```

Sodium contents (mg) of single servings from 40 packages of a food product.											
i	1	2	3	4	5	6	7	8	9	10	
$x_{(i)}$	72.1	72.8	72.9	73.3	73.3	73.3	73.9	74.0	74.2	74.2	
i	11	12	13	14	15	16	17	18	19	20	
$x_{(i)}$	74.3	74.6	74.7	75.0	75.1	75.1	75.2	75.3	75.3	75.3	
i	21	22	23	24	25	26	27	28	29	30	
$x_{(i)}$	75.4	76.1	76.5	76.5	76.6	76.9	77.1	77.2	77.4	77.4	
i	31	32	33	34	35	36	37	38	39	40	
$x_{(i)}$	77.7	78.0	78.3	78.6	78.8	78.9	79.7	80.3	80.5	81.0	

```
% values each so that each column is a bootstrap
% sample of size n
xbootmean = mean(xboot); % means of bootstrap samples
bmean = mean(xbootmean); % mean of bootstrap means
bias = bmean - xmean
variance = mean((xbootmean - bmean).^2)
mse = variance + bias * bias
```

Results:

```
xmean = 38.6500
bias = 0.0308
variance = 27.3295
mse = 27.3304
```

15.4.2 Percentile interval

 Let $\hat{\theta}_1, \dots, \hat{\theta}_N$ be the bootstrap estimates of θ from N bootstrap samples. An approximate $1-\alpha$ confidence interval for θ , known as a $1-\alpha$ percentile interval, is given by $(\hat{\theta}_{(\lceil N\alpha/2 \rceil)}, \hat{\theta}_{(\lceil N(1-\alpha/2) \rceil)})$.

Labwork 207 The sodium contents of single servings from 40 packages of a food product are measured and given in the table below. Find a 0.95 percentile interval for the median amount of sodium in a single serving of this food product.

MATLAB code:

```
x = load('sodium.txt'); % load data from text file and store in x
n = length(x); % determine number of data values
N = 100000; % number of bootstrap samples
nN = n * N;
alpha = 0.05;
alpha2 = alpha / 2;
alpha21 = 1 - alpha2;
xmed = median(x) % median of original data
xboot = randsample(x,nN,true); % sample with replacement nN values from x
xboot = reshape(xboot,n,N); % organise resampled values into N columns of n
                                % values each so that each column is a bootstrap
                                % sample of size n
xbootmed = median(xboot); % medians of bootstrap samples
```

```
xbootmedsort = sort(xbootmed); % sort medians in increasing order
% (1-alpha) percentile interval:
[xbootmedsort(ceil(N*alpha2)) xbootmedsort(ceil(N*alpha21))]
```

Results:

```
xmed = 75.3500
ans = 75.0500 77.0000
```

Therefore, a 0.95 percentile interval for the median is $(\hat{\theta}_{(2500)}, \hat{\theta}_{(97500)}) = (75.05, 77)$.

15.4.3 Properties of the percentile interval.



(a) *Transformation-invariant.* Let g be a one-to-one transformation and let $\psi = g(\theta)$. Let $(\hat{\theta}_{(\lceil N\alpha/2 \rceil)}, \hat{\theta}_{(\lceil N(1-\alpha/2) \rceil)})$ be a $1-\alpha$ percentile interval for θ . If g is increasing, then $(g(\hat{\theta}_{(\lceil N\alpha/2 \rceil)}), g(\hat{\theta}_{(\lceil N(1-\alpha/2) \rceil)}))$ is a $1-\alpha$ percentile interval for ψ . If g is decreasing, then $(g(\hat{\theta}_{(\lceil N(1-\alpha/2) \rceil)}), g(\hat{\theta}_{(\lceil N\alpha/2 \rceil)}))$ is a $1-\alpha$ percentile interval for ψ .

(b) *Range-preserving.* A percentile interval for θ lies within the range of possible values for θ .

(c) *First-order accurate.* The error in the coverage probability of a percentile interval goes to zero at rate $1/\sqrt{n}$.

15.4.4 Bias-corrected and accelerated (BCA) interval



The BCA interval is an improvement of the percentile interval that corrects for median bias (the difference between $\hat{\theta}$ and the median of $\hat{\theta}_1, \dots, \hat{\theta}_N$) and has a coverage probability that is closer to $1 - \alpha$. Like the percentile interval, the BCA interval's end-points are chosen from the ordered values of $\hat{\theta}_1, \dots, \hat{\theta}_N$, and so the interval has the form $(\hat{\theta}_{(r)}, \hat{\theta}_{(s)})$, where $1r < sN$. The variables r and s are chosen to correct for median bias and improve coverage probability.

Let Φ denote the standard normal distribution function and let z_p be the p -quantile of the standard normal distribution. Then:

$$r = \text{round}[N\Phi(z_{\hat{b}} + \frac{z_{\hat{b}} + z_{\alpha/2}}{1 - \hat{a}(z_{\hat{b}} + z_{\alpha/2})})], \quad (15.17)$$

and:

$$s = \text{round}[N\Phi(z_{\hat{b}} + \frac{z_{\hat{b}} + z_{\alpha/2}}{1 - \hat{a}(z_{\hat{b}} + z_{\alpha/2})})], \quad (15.18)$$

where \hat{a} and \hat{b} are yet to be defined. Before defining them, observe that if $\hat{a} = z_{\hat{b}} = 0$, then the BCA interval reduces to the percentile interval.

Now $z_{\hat{b}}$ is a measure of the median bias of the bootstrap estimates, $\hat{\theta}_1, \dots, \hat{\theta}_N$, and so:

$$\hat{b} = \frac{|\{\hat{\theta}_j : \hat{\theta}_j < \hat{\theta}\}|}{N} = \frac{1}{N} \sum_{j=1}^N I_{(-\infty, \hat{\theta})(\hat{\theta}_j)}. \quad (15.19)$$

If the median of $\hat{\theta}_1, \dots, \hat{\theta}_N$ coincides with $\hat{\theta}$, then $\hat{b} = 0.5$ and $z_{0.5} = 0$, and so there is no median bias. The symbol \hat{a} signifies the acceleration because it measures the rate of change of the standard deviation of $\hat{\theta}$ with respect to θ . If the standard deviation of $\hat{\theta}$ is assumed to be the same for all θ , then \hat{a} is 0; this is often unrealistic, so \hat{a} corrects for this. One way to compute \hat{a} is:

$$\hat{a} = \frac{\sum_{j=1}^N (\bar{\hat{\theta}} - \hat{\theta}_j)^3}{6[\sum_{j=1}^N (\bar{\hat{\theta}} - \hat{\theta}_j)^2]^{3/2}}. \quad (15.20)$$

Labwork 208 Continuing with the previous example, find a 0.95 BCA interval for the median.

```
% (1-alpha) BCA interval:
b = sum(xbootmed < xmed) / N;
xbootmedmean = mean(xbootmed);
a = sum((xbootmedmean - xbootmed).^3) / (6 * (sum((xbootmedmean -
xbootmed).^2)^1.5));
zb = norminv(b,0,1);
zalpha2 = norminv(alpha2,0,1);
zalpha21 = norminv(alpha21,0,1);
r = round(N * normcdf(zb + ((zb + zalpha2) / (1 - a * (zb + zalpha2))),0,1));
s = round(N * normcdf(zb + ((zb + zalpha21) / (1 - a * (zb + zalpha21))),0,1));
[xbootmedsort(r) xbootmedsort(s)]
```

Results: (continued)

```
r = 1188
s = 95149
ans = 74.8500    76.7500
```

Therefore, the 0.95 BCA interval for the median is $(\hat{\theta}_{1188}, \hat{\theta}_{95149}) = (74.85, 76.75)$.

15.4.5 Properties of the BCA interval



- (a) The BCA interval is transformation-invariant and range-preserving.
- (b) *Second-order accurate.* The error in the coverage probability of a BCA interval tends to zero at rate $1/n$.

15.5 Extension to multivariate data and linear regression



The extension of the nonparametric bootstrap to multivariate data is straightforward. Bootstrap samples are obtained by randomly sampling with replacement from the multivariate data points.

Labwork 209 This is an example involving bivariate data. The data in `shoe.txt` are the shoe sizes (column 1) and heights (column 2, in inches) of 24 college-age men. Represent each bivariate data point by (x_i, y_i) , where x_i is the shoe size and y_i is the height of the i^{th} man. The correlation between x and y is:

$$\rho = \frac{E\{[x - E(x)][y - E(y)]\}}{\sqrt{E\{[x - E(x)]^2\}E\{[y - E(y)]^2\}}},$$

which can be estimated by the sample correlation:

$$\hat{\rho} = \frac{\sum_{i=1}^{24} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum_{i=1}^{24} (x_i - \bar{x})^2][\sum_{i=1}^{24} (y_i - \bar{y})^2]}}$$

where \bar{x} and \bar{y} are the sample means of x and y respectively. Find a 0.95 BCA interval for the correlation.

A bootstrap sample is obtained by randomly sampling with replacement from $(x_1, y_1), \dots, (x_{24}, y_{24})$. Denoting the j^{th} bootstrap sample by $(x_{j,1}, y_{j,1}), \dots, (x_{j,24}, y_{j,24})$, the j^{th} bootstrap estimate of the correlation coefficient is:

$$\hat{\rho}_j = \frac{\sum_{i=1}^{24} (x_{j,i} - \bar{x}_j)(y_{j,i} - \bar{y}_j)}{\sqrt{[\sum_{i=1}^{24} (x_{j,i} - \bar{x}_j)^2][\sum_{i=1}^{24} (y_{j,i} - \bar{y}_j)^2]}}.$$

```

data = load('shoe.txt'); % load data from text file
x = data(:,1); % store shoe size in x
y = data(:,2); % store height in y
n = length(x); % determine number of data values
N = 100000; % number of bootstrap samples
nN = n * N;
alpha = 0.05;
alpha2 = alpha / 2;
alpha21 = 1 - alpha2;
bootcxy = zeros(1,N); % storage for correlations of bootstrap samples
cxy = corr(x,y) % sample correlation between x and y
iboot = randsample(n,nN,true); % sample with replacement nN values from the
                                % indices 1,...,n
iboot = reshape(iboot,n,N); % organise resampled values into N columns of n
                                % values each so that each column contains the
                                % indices for a bootstrap sample of size n
for i = 1:N
    bootcxy(i) = corr(x(iboot(:,i)),y(iboot(:,i))); % correlation of bootstrap sample i
end
bootcxySort = sort(bootcxy); % sort correlations in increasing order

% (1-alpha) BCA interval:
b = sum(bootcxy < cxy) / N;
bootcxyMean = mean(bootcxy);
a = sum((bootcxyMean - bootcxy).^3) / (6 * (sum((bootcxyMean -
bootcxy).^2)^1.5));
zb = norminv(b,0,1);
zalpha2 = norminv(alpha2,0,1);
zalpha21 = norminv(alpha21,0,1);
r = round(N * normcdf(zb + ((zb + zalpha2) / (1 - a * (zb + zalpha2))),0,1))
s = round(N * normcdf(zb + ((zb + zalpha21) / (1 - a * (zb + zalpha21))),0,1))
[bootcxySort(r) bootcxySort(s)]

```

Results:

```

cxy = 0.7176
r = 1596
s = 96198
ans = 0.3647    0.8737

```

Therefore, the 0.95 BCA interval for the correlation is $(\hat{\rho}_{(1596)}, \hat{\rho}_{(96198)}) = (0.3647, 0.8737)$. For comparison, the usual asymptotic 0.95 confidence interval is $(0.4422, 0.8693)$.

15.5.1 Confidence intervals for regression coefficients

AFor the shoe data, consider a simple linear regression of height on shoe size:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where the $\epsilon_1, \epsilon_2, \dots$ are assumed to be IID with mean 0 and variance σ^2 . Let:

$$B = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ M \\ y_{24} \end{pmatrix} \text{ and } X = \begin{pmatrix} 1 & x_1 \\ M & M \\ 1 & x_{24} \end{pmatrix}.$$

The least-squares estimates of the regression coefficients are given by:

$$\hat{B} = (X^T X)^{-1} X^T Y,$$

which is equivalent to:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

```
data = load('shoe.txt'); % load data from text file
x = data(:,1); % store shoe size in x
y = data(:,2); % store height in y

n = length(x); % determine number of data values
X = [ones(n,1) x]; % predictor matrix
be = regress(y,X) % linear regression of y on x
```

Results:

```
be =
      59.2285
      1.1988
```

Therefore, the least-squares line is $y = 59.23 + 1.2x$.

To get 0.95 BCA intervals for the regression coefficients, we obtain bootstrap samples of size 24 and recompute the least-squares estimates for each bootstrap sample. As in the previous example, a bootstrap sample is obtained by randomly sampling with replacement from $(x_1, y_1), \dots, (x_{24}, y_{24})$. Denote the j^{th} bootstrap sample by $(x_{j,1}, y_{j,1}), \dots, (x_{j,24}, y_{j,24})$ and let:

$$Y_j = \begin{pmatrix} y_{j,1} \\ M \\ y_{j,24} \end{pmatrix} \text{ and } X_j = \begin{pmatrix} 1 & x_{j,1} \\ M & M \\ 1 & x_{j,24} \end{pmatrix}.$$

Then the j^{th} bootstrap estimates of the regression coefficients are given by:

$$\hat{B}_j = (X_j^T X_j)^{-1} X_j^T Y_j.$$

MATLAB code: (continued)

```

N = 100000; % number of bootstrap samples
nN = n * N;
alpha = 0.05;
alpha2 = alpha / 2;
alpha21 = 1 - alpha2;
bootbe = zeros(2,N);
iboot = randsample(n,nN,true); % sample with replacement nN values from the
                                % indices 1,...,n
iboot = reshape(iboot,n,N); % organise resampled values into N columns of n
                                % values each so that each column contains the
                                % indices for a bootstrap sample of size n
for i = 1:N
    % regression for bootstrap sample i:
    bootbe(:,i) = regress(y(iboot(:,i)),[ones(n,1) x(iboot(:,i))]);
end
bootbesort = sort(bootbe,2); % sort regression coefficients in increasing order

% (1-alpha) BCA interval for be0:
b = sum(bootbe(1,:) < be(1)) / N;
bootbe0mean = mean(bootbe(1,:));
a = sum((bootbe0mean - bootbe(1,:)).^3) / (6 * (sum((bootbe0mean -
bootbe(1,:)).^2)^1.5));
zb = norminv(b,0,1);
zalpha2 = norminv(alpha2,0,1);
zalpha21 = norminv(alpha21,0,1);
r = round(N * normcdf(zb + ((zb + zalpha2) / (1 - a * (zb + zalpha2))),0,1))
s = round(N * normcdf(zb + ((zb + zalpha21) / (1 - a * (zb + zalpha21))),0,1))
[bootbesort(1,r) bootbesort(1,s)]

% (1-alpha) BCA interval for be1:
b = sum(bootbe(2,:) < be(2)) / N;
bootbe1mean = mean(bootbe(2,:));
a = sum((bootbe1mean - bootbe(2,:)).^3) / (6 * (sum((bootbe1mean -
bootbe(2,:)).^2)^1.5));
zb = norminv(b,0,1);
zalpha2 = norminv(alpha2,0,1);
zalpha21 = norminv(alpha21,0,1);
r = round(N * normcdf(zb + ((zb + zalpha2) / (1 - a * (zb + zalpha2))),0,1))
s = round(N * normcdf(zb + ((zb + zalpha21) / (1 - a * (zb + zalpha21))),0,1))
[bootbesort(2,r) bootbesort(2,s)]

```

Results: (continued)

```

r = 2489
s = 97489
ans = 55.3841    63.9200

r = 2407
s = 97404
ans = 0.6913    1.5887

```

Therefore, the 0.95 BCA interval for β_0 is $(\hat{\beta}_{0,(2489)}, \hat{\beta}_{0,(97489)}) = (55.38, 63.92)$ and for β_1 is $(\hat{\beta}_{1,(2407)}, \hat{\beta}_{1,(97404)}) = (0.6913, 1.5887)$.

15.5.2 Alternative bootstrap method for regression

 Consider simple linear regression of response y on predictor x with $(x_1, y_1), \dots, (x_n, y_n)$, where the predictors x_1, \dots, x_n are fixed and deterministic. As in the previous example:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i,$$

where the $\epsilon_1, \epsilon_2, \dots$ are assumed to be IID with mean 0 and variance σ^2 . Using the same notations as before, the least-squares estimates of the regression coefficients are given by:

$$\begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = (X^T X)^{-1} X^T Y.$$

For $i = 1, \dots, n$, compute the residuals:

$$\hat{\epsilon}_i = y_i - \hat{\beta}_0 + \hat{\beta}_1 x_i,$$

and then the centred residuals:

$$\hat{\epsilon}_i^* = \hat{\epsilon}_i - \bar{\hat{\epsilon}},$$

where $\bar{\hat{\epsilon}}$ is the sample mean of $\hat{\epsilon}_1, \dots, \hat{\epsilon}_n$.

The j^{th} bootstrap sample and bootstrap regression coefficients are obtained as follows:

- (a) Randomly sample with replacement from $\hat{\epsilon}_1^*, \dots, \hat{\epsilon}_n^*$ to get $\hat{\epsilon}_{j,1}, \dots, \hat{\epsilon}_{j,n}$.
- (b) For $i = 1, \dots, n$, obtain bootstrap responses by:

$$y_{j,i} = \hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{\epsilon}_{j,i}.$$

The bootstrap sample is $(x_1, y_{j,1}), \dots, (x_n, y_{j,n})$.

- (c) The bootstrap regression coefficients are given by:

$$\begin{pmatrix} \hat{\beta}_{j,0} \\ \hat{\beta}_{j,1} \end{pmatrix} = (X^T X)^{-1} X^T Y_j.$$

Labwork 210 Referring to Example 4.3.2 for the shoe data, obtain 0.95 BCA intervals for the regression coefficients by bootstrapping residuals.

```

data = load('shoe.txt'); % load data from text file
x = data(:,1); % store shoe size in x
y = data(:,2); % store height in y
n = length(x); % determine number of data values
X = [ones(n,1) x]; % predictor matrix

% linear regression of y on x:
[be,beint,res] = regress(y,X); % store residuals in res
res = res - mean(res); % centred residuals

N = 100000; % number of bootstrap samples
nN = n * N;
alpha = 0.05;
alpha2 = alpha / 2;
alpha21 = 1 - alpha2;
bootbe = zeros(2,N);

rboot = randsample(res,nN,true); % sample with replacement nN values from res
rboot = reshape(rboot,n,N); % organise resampled values into N columns of n
                                % values each so that each column contains n
                                % bootstrapped residuals

for i = 1:N
    yboot = X * be + rboot(:,i);
    bootbe(:,i) = regress(yboot,X); % regression for bootstrap sample i
end
bootbesort = sort(bootbe,2); % sort regression coefficients in increasing order

```

```
% (1-alpha) BCA interval for be0:
b = sum(bootbe(1,:) < be(1)) / N;
bootbe0mean = mean(bootbe(1,:));
a = sum((bootbe0mean - bootbe(1,:)).^3) / (6 * (sum((bootbe0mean - bootbe(1,:)).^2)^1.5));
zb = norminv(b,0,1);
zalpha2 = norminv(alpha2,0,1);
zalpha21 = norminv(alpha21,0,1);
r = round(N * normcdf(zb + ((zb + zalpha2) / (1 - a * (zb + zalpha2))),0,1))
s = round(N * normcdf(zb + ((zb + zalpha21) / (1 - a * (zb + zalpha21))),0,1))
[bootbesort(1,r) bootbesort(1,s)]

% (1-alpha) BCA interval for be1:
b = sum(bootbe(2,:) < be(2)) / N;
bootbe1mean = mean(bootbe(2,:));
a = sum((bootbe1mean - bootbe(2,:)).^3) / (6 * (sum((bootbe1mean - bootbe(2,:)).^2)^1.5));
zb = norminv(b,0,1);
zalpha2 = norminv(alpha2,0,1);
zalpha21 = norminv(alpha21,0,1);
r = round(N * normcdf(zb + ((zb + zalpha2) / (1 - a * (zb + zalpha2))),0,1))
s = round(N * normcdf(zb + ((zb + zalpha21) / (1 - a * (zb + zalpha21))),0,1))
[bootbesort(2,r) bootbesort(2,s)]
```

Results:

```
r = 2551
s = 97550
ans = 54.6419    63.9479

r = 2372
s = 97366
ans = 0.7148    1.6548
```

Therefore, the 0.95 BCA interval for β_0 is $(\hat{\beta}_{0,(2551)}, \hat{\beta}_{0,(97550)}) = (54.64, 63.95)$; for β_1 , it is $(\hat{\beta}_{1,(2372)}, \hat{\beta}_{1,(97366)}) = (0.7148, 1.6548)$.

15.6 Extension to dependent data

 Recall that the nonparametric bootstrap requires the data values to be IID. We briefly describe how the nonparametric bootstrap can be applied when the data values are not independent, particularly in the context of time series data. The key idea is to divide the data into blocks that are “approximately independent” and then perform random sampling with replacement on these blocks rather than on the individual data values. Hence, this extension of the nonparametric bootstrap to dependent data is sometimes referred to as the *block bootstrap*.

15.6.1 Block bootstrap

 Let x_1, \dots, x_n be a sequence of time series measurements, with the indices denoting the times at which the measurements are taken, i.e. x_1 is obtained before x_2 and so on. The measurements are not independent but have some form of dependence over time. One of the simplest forms of the block bootstrap is as follows:

- (a) Specify a *block length* b (b must be smaller than n). Let $m = \text{round}(n/b)$.

- (b) Divide x_1, \dots, x_n into blocks as follows:

$$\begin{aligned}B_1 &= \{x_1, \dots, x_b\} \\B_2 &= \{x_2, \dots, x_b + 1\} \\&\vdots \\B_{n-b+1} &= \{x_{n-b+1}, \dots, x_n\}\end{aligned}$$

- (c) Randomly pick m blocks with replacement, calling them B_1^*, \dots, B_m^* .
 (d) Concatenate B_1^*, \dots, B_m^* to get a bootstrap sample of the time series.
 (e) Repeat Steps (c) and (d) to get another bootstrap sample.

The block bootstrap procedure looks simple but the difficulty lies in the choice of the block length. A good block length depends on at least three things: the time series, the statistic of interest and the purpose for bootstrapping the statistic. Unfortunately, further discussion on block length choice involves concepts that are beyond the level of this course and so we shall have to stop here.

15.7 Exercises



Exercise 211 Download the MATLAB function, `edfplot.m`, from the course webpage. Use it to obtain the EDF and 0.95 confidence band for:

- (a) the continuous data in Example 4.1.3;
 (b) the discrete data in Example 4.1.4.

Exercise 212 The sodium data in Example 4.2.5 are given in `sodium.txt`. Use the nonparametric bootstrap with 100,000 bootstrap samples to find the bias, variance and MSE of the sample median.

Exercise 213 The latch data in Example 4.2.3 are given in `latch.txt`. Use the nonparametric bootstrap with 100,000 bootstrap samples to find a 0.95 percentile interval and the 0.95 BCA for the standard deviation. (MATLAB function for standard deviation is `std`.)

Exercise 214 The data in `hemoglobin.txt` are the haemoglobin levels (in g/dl) of 20 Canadian Olympic ice hockey players.

- (A) Use the nonparametric bootstrap with 100000 bootstrap samples to find:

- (i) the bias, variance and MSE of the sample 0.1-quantile;
 (ii) a 0.95 percentile interval and the 0.95 BCA for the 0.1-quantile.

(MATLAB function for quantile is `quantile`.)

(B) Using the BCA interval that you have already found in Part (a), find the 0.95 BCA interval for the square root of the 0.1-quantile.

- (C) Plot a density histogram for the bootstrap estimates of the 0.1-quantile.

Exercise 215 The data in `tar.txt` are the tar contents in a sample of 30 cigars of a particular brand.

- (a) Use the nonparametric bootstrap with 100,000 bootstrap samples to find:

- (i) the bias, variance and MSE of the sample interquartile range;
- (ii) a 0.95 percentile interval and the 0.95 BCA for the interquartile range.

(MATLAB function for interquartile range is `iqr`.)

- (b) Plot a density histogram for the bootstrap estimates of the interquartile range.

Exercise 216 Work through Examples 4.3.1, 4.3.2 and 4.3.4 for the shoe data.

Exercise 217 The data in `stream.txt` contain chloride concentrations (in mg/l) found at the surface of streams (column 1), and road densities (in %) in the vicinities of the streams (column 2).

- (a) Obtain a scatter plot of chloride concentration against road density and find a 0.95 BCA interval (using 100,000 nonparametric bootstrap samples) for the correlation coefficient between road density and chloride concentration. What can you conclude from the scatter plot and the confidence interval?
- (b) Consider a simple linear regression of chloride concentration (y or response) on road density (x or predictor). Obtain 0.95 BCA intervals for the regression coefficients, using 100,000 nonparametric bootstrap samples and by:
 - (i) bootstrapping sample points;
 - (ii) bootstrapping residuals.

Exercise 218 The data in `salmon.txt` contain the number of recruits (column 1) and the number of spawners (column 2) in 40 salmon farms. The units are thousands of fish. Recruits are fish that are big enough to be sold. Spawners are fish that are kept for laying eggs, after which they die.

The Beverton-Holt model for the relationship between recruits and spawners is:

$$R = \frac{1}{\beta_0 + (\beta_1/S)},$$

where R and S are the numbers of recruits and spawners, and $\beta_0, \beta_1 \geq 0$.

- (a) Use the data to estimate β_0 and β_1 for the Beverton-Holt model by using linear regression with the transformed variables $1/R$ and $1/S$.
- (b) Consider the problem of maintaining a sustainable farm. The salmon population stabilises when $R = S$. Show that the stable population size is given by:

$$R = S = \frac{1 - \beta_1}{\beta_0}.$$

Using the estimates from Part (a), estimate the stable population size.

- (c) Use the nonparametric bootstrap with 100,000 bootstrap samples to find 0.95 BCA intervals for the stable population by:
 - (i) bootstrapping sample points;
 - (ii) bootstrapping residuals.

Chapter 16

Monte Carlo Estimation

Various problems can be solved by taking advantage of randomness and random number generators. Such an approach yields a Monte Carlo solution.

16.1 Monte Carlo Integral Estimation

Suppose we want to estimate the integral

$$\vartheta^* = \int_A h(x) dx , \text{ where, } h(x) : \mathbb{R}^k \rightarrow \mathbb{R}, A \subset \mathbb{R}^k .$$

If the integrand function h is simple enough, eg. polynomial, simple transcendental or trigonometric, then we can evaluate ϑ^* **analytically by hand**. When this is not the case, then a closed form expression for ϑ^* may not exist and we may approximate it by **numerical quadrature**, eg. Mid-point rectangles rule (QuadMR), or adaptive Simpson's quadrature rule (quad) or adaptive Lobatto's quadrature rule (quad1). Numerical quadratures are inefficient in higher dimensions since the number of samples or grid-points grow exponentially with the dimension k of the domain A .

Basic Monte Carlo (BMC) integral estimation or **Basic Monte Carlo integration** is a stochastic method to estimate the integral ϑ^* by its point estimate $\hat{\Theta}_n$ based on n samples drawn at random from the domain A . BMC integration is renowned for its **simplicity**, **generality** (class of integrands) and **scalability** (dimension of the integration domain).

Let the domain be a k -dimensional box or hyper-cuboid $A = ([\underline{a}_1, \bar{a}_1], [\underline{a}_2, \bar{a}_2], \dots, [\underline{a}_k, \bar{a}_k])$. We can rewrite the integral as:

$$\vartheta^* = \int_A h(x) dx = \int_A w(x)f(x) dx , \tag{16.1}$$

where the functions $w(x)$ and $f(x)$ are:

$$w(x) = h(x) \prod_{j=1}^k (\bar{a}_j - \underline{a}_j), \quad f(x) = \frac{1}{\prod_{j=1}^k (\bar{a}_j - \underline{a}_j)} .$$

The PDF of the RV $X \sim \text{Uniform}(A) = \text{Uniform}(\underline{a}_1, \bar{a}_1) \times \text{Uniform}(\underline{a}_2, \bar{a}_2) \times \dots \times \text{Uniform}(\underline{a}_k, \bar{a}_k)$ is $f(x)$ over the box A . Therefore:

$$\vartheta^* = \int_A w(x)f(x) dx = \mathbf{E}_f(w(X)) = \mathbf{E}(w(X)) .$$

We subscript the expectation E by f to emphasize the PDF with respect to which we integrate. Thus, if we generate $X_1, X_2, \dots, X_n \stackrel{\text{IID}}{\sim} \text{Uniform}(A)$, then by the WLLN:

$$\widehat{\Theta}_n := \frac{1}{n} \sum_{i=1}^n w(X_i) \xrightarrow{P} \mathbf{E}(w(X_1)) = \vartheta^* .$$

The standard error:

$$\text{se}_n := \sqrt{\mathbf{V}(\widehat{\Theta}_n)} = \sqrt{\mathbf{V}\left(\frac{1}{n} \sum_{i=1}^n w(X_i)\right)} = \sqrt{\frac{1}{n^2} \sum_{i=1}^n \mathbf{V}(w(X_i))} = \sqrt{\frac{1}{n^2} n \mathbf{V}(w(X_1))} = \frac{1}{\sqrt{n}} \sqrt{\mathbf{V}(w(X_1))} .$$

We can get the estimated standard error $\widehat{\text{se}}_n$ of the integral estimate $\widehat{\Theta}_n$ from the sample standard deviation s_n :

$$\widehat{\text{se}}_n = \frac{s_n}{\sqrt{n}}, \quad s_n^2 = S_n^2((y_1, y_2, \dots, y_n)) = \frac{1}{n-1} \sum_{i=1}^n (y_i - \widehat{\Theta}_n)^2 ,$$

where, $Y_i = w(X_i)$. A Normal-based $1-\alpha$ confidence interval (CI) for ϑ^* is $\widehat{\Theta}_n \pm z_{\alpha/2} \widehat{\text{se}}_n$. Therefore, we can take n as large as we want and thereby shrink the width of the confidence interval as small as we want. Our recipe for estimating ϑ^* is simply summarized in Algorithm 13.

Algorithm 13 Basic Monte Carlo Integral Estimation for $\vartheta^* = \int_A h(x)dx$

1: *input:*

1. $n \leftarrow$ the number of samples.
2. $h(x) \leftarrow$ the integrand function over \mathbb{R}
3. $[\underline{a}_j, \bar{a}_j] \leftarrow$ lower and upper bounds of integration for each $j = 1, 2, \dots, k$
4. capability to draw nk IID samples from $\text{Uniform}(0, 1)$ RV

2: *output:* a point estimate $\widehat{\vartheta}_n$ of ϑ^* and the estimated standard error $\widehat{\text{se}}_n$

3: *initialize:* $y \leftarrow (0, 0, \dots, 0)$, initialize y as a zero vector of length n

4: **while** $i \leq n$ **do**

5: 1. *i* $\leftarrow i + 1$,

 2. $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,k})$, with $x_{i,j} \leftarrow u_j$, $u_j \sim \text{Uniform}(\underline{a}_j, \bar{a}_j)$, for $j = 1, 2, \dots, k$,

 3. $y_i \leftarrow w(x_i) = h(x_i) \prod_{j=1}^k (\bar{a}_j - \underline{a}_j)$

6: **end while**

7: *compute:*

1. $\widehat{\vartheta}_n \leftarrow \bar{y}_n$, the sample mean of $y = (y_1, y_2, \dots, y_n)$

2. $\widehat{\text{se}}_n = s_n(y)/\sqrt{n}$, where $s_n(y)$ is the sample standard deviation of y

8: *return:* $\widehat{\vartheta}_n$ and $\widehat{\text{se}}_n$

Labwork 219 (1D integral over an interval) Estimate $\vartheta^* := \int_0^{0.5} x(1-x^2)^{3/2} dx$ using Algorithm 13. We use IID samples x_1, x_2, \dots, x_n from $\text{Uniform}(0, 0.5)$ RV to estimate ϑ^* as follows:

$$\widehat{\vartheta}_n = \frac{1}{n} \sum_{i=1}^n w(x_i) = \frac{1}{n} \sum_{i=1}^n h(x_i)(0.5 - 0), \quad \text{where, } h(x_i) = x_i(1-x_i^2)^{3/2} .$$

The estimation procedure may be implemented as follows:

```
>> rand('twister',189783)% initialise the fundamental sampler
>> N=1000000; % estimate from 1000000 samples
>> Xs=0.5*rand(1,N); % save 1 million samples from Uniform(0,0.5) in array Xs
>> Ws=(Xs .* ((1 - (Xs .^ 2)) .^ 1.5)) * 0.5; % h(Xs)* 1/f, f is density of Uniform(0,0.5)
>> MCEst=mean(Ws(1:N)) % point estimate of the integral
MCEst = 0.1026
>> StdErr=std(Ws(1:N))/sqrt(N) % estimated standard error
StdErr = 5.0026e-05
>> % approximate 95% confidence interval of the estimate
>> [MCEst-2*StdErr MCEst+2*StdErr]
ans =
0.1025    0.1027
```

The estimates and the associated confidence intervals for different sample sizes are:

n	$\hat{\vartheta}_n$	$\approx 95\%$ C.I.
10^2	0.1087	(0.0985, 0.1189)
10^3	0.1055	(0.1023, 0.1086)
10^4	0.1028	(0.1018, 0.1038)
10^5	0.1023	(0.1020, 0.1027)
10^6	0.1026	(0.1025, 0.1027)

The exact answer is:

$$\int_0^{0.5} x(1-x^2)^{3/2} dx = \left(-\frac{(1-x^2)^{5/2}}{5} \right)_0^{0.5} = 0.10257.$$

Labwork 220 (2D integral over a rectangle) Let us estimate the area of a circle $C := \{(x, y) : \sqrt{x^2 + y^2} \leq 1\}$ centred at the origin with unit radius using IID uniform samples from a unit square $[-1, 1]^2 := [-1, 1] \times [-1, 1]$ that contains C . Therefore, the integral of interest:

$$\vartheta^* := \int_{-1}^1 \int_{-1}^1 \mathbf{1}_C(u_1, u_2) du_1 du_2 = \int_{-1}^1 \int_{-1}^1 (4 \mathbf{1}_C(u_1, u_2)) \frac{1}{4} du_1 du_2 ,$$

where $\frac{1}{4} \mathbf{1}_{[-1,1]^2}(u_1, u_2)$ is the joint PDF of $\text{Uniform}([-1, 1]^2)$ RV, and the Monte Carlo estimate of ϑ^* based on n samples from $\text{Uniform}([-1, 1]^2)$ is:

$$\hat{\vartheta}_n = \frac{1}{n} \sum_{i=1}^n 4 \mathbf{1}_C(u_1, u_2)$$

We know that our integral, namely the area of the unit circle, is $\pi(\text{radius})^2 = \pi 1^2 = \pi = 3.1416 \dots$

```
>> rand('twister',188);%Initialise the fundamental sampler
>> N=10000; % sample size of ten thousand
>> Ws=zeros(1,N); % initialise a vector of zeros
>> % produce N pairs of Uniform(0,1) numbers and set to 4 if they fall inside a unit circle
>> Ws(find(sqrt(sum(rand(2,N) .^ 2)) <= 1.0 ))=4;
>> MCEst=mean(Ws(1:N)); % MC estimate
>> StdErr=std(Ws(1:N))/sqrt(N); % estimated standard error
>> disp([MCEst StdErr MCEst-2*StdErr MCEst+2*StdErr])% display estimate, std error, and approx 95% CI
3.1476    0.0164    3.1148    3.1804

>> % with N=10^7 samples we get a better approximation
```

```
>> rand('twister',188); N=10000000; Ws=zeros(1,N); Ws(find(sqrt(sum(rand(2,N).^2)) <= 1.0 ))=4;
>> MCEst=mean(Ws(1:N)); StdErr=std(Ws(1:N))/sqrt(N);
>> disp([MCEst StdErr MCEst-2*StdErr MCEst+2*StdErr])
    3.1420    0.0005    3.1409    3.1430
>> pi % correct value of pi
ans =      3.1416
```

We can also estimate the integral of functions over unbounded domain by first transforming them to one over a bounded domain. For the integral:

$$\vartheta^* := \int_0^\infty h(x)dx ,$$

the substitution $y = 1/(x + 1)$ will change the interval of integration to a bounded one:

$$\vartheta^* = \int_0^\infty h(x)dx = \int_1^0 h\left(\frac{1}{y} - 1\right) \frac{-1}{y^2} dy = \int_0^1 h\left(\frac{1}{y} - 1\right) \frac{1}{y^2} dy.$$

Therefore, with $y_1, \dots, y_n \stackrel{IID}{\sim} \text{Uniform}(0, 1)$, the integral can be estimated by:

$$\hat{\vartheta}_n = \frac{1}{n} \sum_{i=1}^n h\left(\frac{1}{y_i} - 1\right) \frac{1}{y_i^2} .$$

Similarly, we can estimate the integral:

$$\vartheta^* := \int_{-\infty}^\infty h(x)dx = \int_{-\infty}^0 h(x)dx + \int_0^\infty h(x)dx = \int_0^\infty (h(-x) + h(x)) dx ,$$

can be estimated by:

$$\hat{\vartheta}_n = \frac{1}{n} \sum_{i=1}^n \left(h\left(1 - \frac{1}{y_i}\right) + h\left(\frac{1}{y_i} - 1\right) \right) \frac{1}{y_i^2} .$$

Labwork 221 (1D Integral with Unbounded Domain) Estimate $\vartheta^* = \int_{-\infty}^\infty e^{-x^2} dx$ by the Monte Carlo estimate $\hat{\vartheta}_n = \frac{1}{n} \sum_{i=1}^n e^{-x_i}$, $x_i \stackrel{IID}{\sim} \text{Uniform}(0, 1)$ MATLAB code:

```
>> y=rand(1,1000000);
>> Hy = (exp(-((1-(1./y)).^2)) + exp(-((1./y)-1).^2))) ./ (y .* y);
>> mean(Hy)
ans =      1.7722
>> disp([mean(Hy)- 1.96*std(Hy)/sqrt(1000000), mean(Hy)+ 1.96*std(Hy)/sqrt(1000000)])
1.7694    1.7749
```

The Monte Carlo estimates $\hat{\vartheta}_n$ of ϑ^* for different sample sizes n are:

n	$\hat{\vartheta}_n$	$\approx 95\% \text{ C.I.}$
10^2	1.7062	(1.4, 2.0)
10^4	1.7849	(1.75, 1.81)
10^6	1.7717	(1.769, 1.775)

The exact answer is:

$$\int_{-\infty}^\infty e^{-x^2} dx = \sqrt{\pi} = 1.7725 (\text{ 4 decimal places}),$$

which can be obtained by noting that e^{-x^2} is the un-normalised $\text{Normal}(0, 0.5^2)$ density.

16.2 Variance Reduction via Importance Sampling

Consider the problem of estimating the integral $\vartheta^* = \int h(x)f(x)dx$, where $f(x)$ is a PDF. In basic Monte Carlo method, we can simulate from the RV X with PDF f . However, there are situations where we may not be able to draw samples from X . **Importance sampling** overcomes this problem by generalising the basic Monte Carlo method. Suppose $g(x)$ is a PDF from which we can draw samples, then

$$\vartheta^* = \int h(x)f(x)dx = \int \frac{h(x)f(x)}{g(x)}g(x)dx = \mathbf{E}_g(Y),$$

where, $Y = h(X)f(X)/g(X)$ and the expectation $\mathbf{E}_g(Y)$ is taken with respect to the PDF g . Therefore, we can simulate $x_1, x_2, \dots, x_n \stackrel{IID}{\sim} g$ and estimate ϑ^* by:

$$\widehat{\vartheta}_n = \frac{1}{n} \sum_{i=1}^n y_i = \frac{1}{n} \sum_{i=1}^n \frac{h(x_i)f(x_i)}{g(x_i)}.$$

By the law of large numbers, $\widehat{\vartheta}_n \xrightarrow{P} \vartheta^*$, however $\widehat{\vartheta}_n$ may have an infinite standard error if $g(x)$ is chosen poorly. Since the estimator $\widehat{\vartheta}_n$ of ϑ^* is the mean of $w(X) = h(X)f(X)/g(X)$ and the second moment of $W(X)$, given by:

$$\mathbf{E}_g(w^2(X)) = \int (w(x))^2 g(x)dx = \int \left(\frac{h(x)f(x)}{g(x)} \right)^2 g(x)dx = \int \frac{h^2(x)f^2(x)}{g(x)} dx$$

may be infinite if g has thinner tails than f . Moreover, $\mathbf{E}_g(w^2(X))$ may be large if $g(x)$ is small over some set A where $f(x)$ is large, since the ratio f/g over A could become large. Therefore, we want the **importance sampling density** g to have thicker tails than f and also be of similar shape to f to minimise the ratio f/g . In fact, the optimal choice of the importance sampling density g is given by the following proposition.

Proposition 114 (Optimal Importance Sampling Density) The optimal choice for the importance sampling density g that minimises the variance of the importance sampling estimator

$$\widehat{\vartheta}_n = \frac{1}{n} \sum_{i=1}^n \frac{h(X_i)f(X_i)}{g(X_i)}, \quad X_1, X_2, \dots, X_n \stackrel{IID}{\sim} g$$

of the integral:

$$\vartheta^* = \int h(x)f(x)dx,$$

is

$$g^*(x) = \frac{|h(x)|f(x)}{\int |h(t)|f(t)dt}.$$

Proof: Let $w(X) := \frac{f(X)h(X)}{g(X)}$. The variance of $w(X)$ is:

$$\begin{aligned} \mathbf{V}_g(w(X)) &= \mathbf{E}_g(w^2(X)) - (\mathbf{E}_g(w(X)))^2 \\ &= \int w^2(x)g(x)dx - \left(\int w(x)g(x)dx \right)^2 \\ &= \int \frac{h^2(x)f^2(x)}{g^2(x)}g(x)dx - \left(\int \frac{h(x)f(x)}{g(x)}g(x)dx \right)^2 \\ &= \int \frac{h^2(x)f^2(x)}{g^2(x)}g(x)dx - \left(\int h(x)f(x)dx \right)^2 \end{aligned}$$

Note that $(\int h(x)f(x)dx)^2$ does not depend on g , therefore, minimisation of

$$\mathbf{V}_g(\hat{\Theta}_n) = \mathbf{V}_g\left(\frac{1}{n} \sum_{i=1}^n \frac{h(X_i)f(X_i)}{g(X_i)}\right) = \mathbf{V}_g\left(\sum_{i=1}^n \frac{1}{n} w(X_i)\right) = n \frac{1}{n^2} \mathbf{V}_g(w(X_1)) = \frac{1}{n} \mathbf{V}_g(w(X_1))$$

over all possible densities g , is equivalent to minimisation of $\mathbf{E}_g(w^2(X)) = \int \frac{h^2(x)f^2(x)}{g^2(x)} g(x)dx$, where $X \sim g$. Due to Jensen's inequality:

$$\begin{aligned} \mathbf{E}_g(w^2(X)) &\geq (\mathbf{E}_g(|w(X)|))^2 \\ &= \left(\int \frac{|h(x)f(x)|}{|g(x)|} g(x)dx \right)^2 = \left(\int \frac{|h(x)|f(x)}{g(x)} g(x)dx \right)^2 = \left(\int |h(x)|f(x)dx \right)^2 \end{aligned}$$

This establishes a lower bound on $\mathbf{E}_g(w^2(X))$ and thereby on $\mathbf{V}_g(\hat{\Theta}_n)$. This lower bound is achieved when $g(x) = g^*(x) = \frac{|h(x)|f(x)}{\int |h(t)|f(t)dt}$:

$$\begin{aligned} \mathbf{E}_{g^*}(w^2(X)) &= \int \frac{h^2(x)f^2(x)}{g^{*2}(x)} g^*(x)dx = \int \frac{|h(x)|^2 f^2(x)}{\left(\int |h(x)|^2 f^2(x) dx\right)^2} \frac{|h(x)|f(x)}{\int |h(t)|f(t)dt} dx \\ &= \int \left(\int |h(t)|f(t)dt \right)^2 \frac{|h(x)|f(x)}{\int |h(t)|f(t)dt} dx = \left(\int |h(t)|f(t)dt \right) \int |h(x)|f(x)dx \\ &= \left(\int |h(x)|f(x)dx \right)^2 \end{aligned}$$

There is no free lunch. If we can't sample from $f(x)$, then we probably can't sample from the more complicated optimal importance sampling density $g(x)^* = |h(x)|f(x)/\int |h(t)|f(t)dt$ either. In practice, we sample from a thick-tailed density g that is as close as possible to $g^* = |h|f$.

Labwork 222 (Estimating $\mathbf{P}(Z > \pi)$) Compare the estimates and standard errors of the following estimators of the Gaussian tail probability:

$$\vartheta^* = \mathbf{P}(Z > \pi) = \int_{-\infty}^{\infty} \mathbf{1}_{(\pi, \infty)}(x) \varphi(x) dx, \quad \text{where, } Z \sim \text{Normal}(0, 1), \varphi(x) \text{ is PDF of } Z,$$

for different sample sizes $n = \{10^2, 10^4, 10^6\}$, based on four different importance sampling densities:

1. $g^{(1)} = \varphi$, the PDF of $\text{Normal}(0, 1)$ (simulate x_1, x_2, \dots, x_n from $\text{Normal}(0, 1)$ via `randn(1, n)`),
2. $g^{(2)}$, the PDF of $\text{Normal}(\mu = 4.4, \sigma^2 = 1)$ (simulate x_1, x_2, \dots, x_n from $\text{Normal}(\mu, \sigma^2)$ by $x_i \leftarrow \mu + \sigma z_i$, $z_i \sim \text{Normal}(0, 1)$, i.e. via `4.4 + 1.*randn(1, n);`),
3. $g^{(3)} = \exp(\pi - x)$, the PDF of a π -translated Exponential($\lambda = 1$) RV X with support $[\pi, \infty)$ (simulate $x_1, x_2, \dots, x_n \sim X$ by $x_i \leftarrow \pi + -\log(u_i)$, $u_i \sim \text{Uniform}(0, 1)$) and
4. $g^{(4)}$, the PDF of $|X|$, where $X \sim \text{Normal}(\pi, 1)$.

The Monte Carlo estimate based on $g^{(1)} = \varphi \sim \text{Normal}(0, 1)$:

$$\widehat{\vartheta}_n^{(1)} = \frac{1}{n} \sum_{i=1}^n \frac{\mathbf{1}_{(\pi, \infty)}(x_i) \varphi(x_i)}{g^{(1)}(x_i)} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{(\pi, \infty)}(x_i), \quad x_1, \dots, x_n \stackrel{IID}{\sim} g^{(1)} = \varphi \sim \text{Normal}(0, 1).$$

The Monte Carlo estimate based on $g^{(2)} \sim \text{Normal}(4.4, 1)$:

$$\widehat{\vartheta}_n^{(1)} = \frac{1}{n} \sum_{i=1}^n \frac{\mathbf{1}_{(\pi, \infty)}(x_i) \varphi(x_i)}{g^{(2)}(x_i)} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{(\pi, \infty)}(x_i) \frac{\varphi(x_i)}{g^{(2)}(x_i)}, \quad x_1, \dots, x_n \stackrel{IID}{\sim} g^{(2)} \sim \text{Normal}(4.4, 1).$$

The Monte Carlo estimate based on $g^{(3)}$:

$$\widehat{\vartheta}_n^{(1)} = \frac{1}{n} \sum_{i=1}^n \frac{\mathbf{1}_{(\pi, \infty)}(x_i) \varphi(x_i)}{g^{(3)}(x_i)} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{(\pi, \infty)}(x_i) \frac{\varphi(x_i)}{g^{(3)}(x_i)}, \quad x_1, \dots, x_n \stackrel{IID}{\sim} g^{(3)}.$$

The Monte Carlo estimate based on $g^{(4)}$:

$$\widehat{\vartheta}_n^{(1)} = \frac{1}{n} \sum_{i=1}^n \frac{\mathbf{1}_{(\pi, \infty)}(x_i) \varphi(x_i)}{g^{(4)}(x_i)} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{(\pi, \infty)}(x_i) \frac{\varphi(x_i)}{g^{(4)}(x_i)}, \quad x_1, \dots, x_n \stackrel{IID}{\sim} g^{(4)}.$$

16.3 Sequential Monte Carlo Methods



16.3.1 Sequential Importance Sampling

16.3.2 Population MCMC

16.3.3 Genetic Monte Carlo Algorithms

16.4 Monte Carlo Optimisation



Chapter 17

Hypothesis Testing

The subset of **all posable hypotheses** that remain **falsifiable** is the space of **scientific hypotheses**. Roughly, a falsifiable hypothesis is one for which a statistical experiment can be designed to produce data that an experimenter can use to falsify or reject it. In the statistical decision problem of hypothesis testing, we are interested in empirically falsifying a scientific hypothesis, i.e. we attempt to reject an hypothesis on the basis of empirical observations or data. Thus, hypothesis testing has its roots in the philosophy of science and is based on Karl Popper's falsifiability criterion for demarcating scientific hypotheses from the set of all posable hypotheses.

17.1 Introduction

Usually, the hypothesis we attempt to reject or falsify is called the **null hypothesis** or H_0 and its complement is called the **alternative hypothesis** or H_1 . For example, consider the following two hypotheses:

H_0 : The average waiting time at an Orbiter bus stop is less than or equal to 10 minutes.

H_1 : The average waiting time at an Orbiter bus stop is more than 10 minutes.

If the sample mean \bar{x}_n is much larger than 10 minutes then we may be inclined to reject the null hypothesis that the average waiting time is less than or equal to 10 minutes. We will learn to formally test hypotheses in the sequel.

Suppose we are interested in the following hypothesis test for the bus-stop problem:

H_0 : The average waiting time at an Orbiter bus stop is equal to 10 minutes.

H_1 : The average waiting time at an Orbiter bus stop is not 10 minutes.

Once again we can use the sample mean as the test statistic. Our procedure for rejecting this null hypothesis is different and is often called the Wald test.

More generally, suppose $X_1, X_2, \dots, X_n \stackrel{IID}{\sim} F(x_1; \theta^*)$, with an unknown and fixed $\theta^* \in \Theta$. Let us partition the parameter space Θ into Θ_0 , the null parameter space, and Θ_1 , the alternative parameter space, ie,

$$\Theta_0 \cup \Theta_1 = \Theta, \quad \text{and} \quad \Theta_0 \cap \Theta_1 = \emptyset.$$

Then, we can formalise testing the null hypothesis versus the alternative as follows:

$$H_0 : \theta^* \in \Theta_0 \quad \text{versus} \quad H_1 : \theta^* \subset \Theta_1.$$

The basic idea involves finding an appropriate rejection region \mathbb{X}_R within the data space \mathbb{X} and rejecting H_0 if the observed data $x := (x_1, x_2, \dots, x_n)$ falls inside the rejection region \mathbb{X}_R ,

If $x := (x_1, x_2, \dots, x_n) \in \mathbb{X}_R \subset \mathbb{X}$, then reject H_0 , else do not reject H_0 .

Typically, the rejection region \mathbb{X}_R is of the form:

$$\mathbb{X}_R := \{x := (x_1, x_2, \dots, x_n) : T(x) > c\}$$

where, T is the **test statistic** and c is the **critical value**. Thus, the problem of finding \mathbb{X}_R boils down to that of finding T and c that are appropriate. Once the rejection region is defined, the possible outcomes of a hypothesis test are summarised in the following table.

Table 17.1: Outcomes of an hypothesis test.

	Do not Reject H_0	Reject H_0
H_0 is True	OK	Type I Error
H_1 is True	Type II Error	OK

Definition 115 (Power, Size and Level of a Test) The **power function** of a test with rejection region \mathbb{X}_R is

$$\beta(\theta) := \mathbf{P}_\theta(x \in \mathbb{X}_R). \quad (17.1)$$

So $\beta(\theta)$ is the power of the test at the parameter value θ , i.e. the probability that the observed data x , sampled from the distribution specified by θ , falls in \mathbb{X}_R and thereby leads to a rejection of the null hypothesis.

The **size** of a test with rejection region \mathbb{X}_R is the supreme power under the null hypothesis, i.e. the supreme probability of rejecting the null hypothesis when the null hypothesis is true:

$$\text{size} := \sup_{\theta \in \Theta_0} \beta(\theta) := \sup_{\theta \in \Theta_0} \mathbf{P}_\theta(x \in \mathbb{X}_R). \quad (17.2)$$

The size of a test is often denoted by α . A test is said to have **level α** if its **size** is less than or equal to α .

Let us familiarize ourselves with some terminology in hypothesis testing next.

Table 17.2: Some terminology in hypothesis testing.

Θ	Test: H_0 versus H_1	Nomenclature
$\Theta \subset \mathbb{R}^m, m \geq 1$	$H_0 : \theta^* = \theta_0$ versus $H_1 : \theta^* \neq \theta_1$	Simple Hypothesis Test
$\Theta \subset \mathbb{R}^m, m \geq 1$	$H_0 : \theta^* \in \Theta_0$ versus $H_1 : \theta^* \in \Theta_1$	Composite Hypothesis Test
$\Theta \subset \mathbb{R}^1$	$H_0 : \theta^* = \theta_0$ versus $H_1 : \theta^* \neq \theta_0$	Two-sided Hypothesis Test
$\Theta \subset \mathbb{R}^1$	$H_0 : \theta^* \geq \theta_0$ versus $H_1 : \theta^* < \theta_0$	One-sided Hypothesis Test
$\Theta \subset \mathbb{R}^1$	$H_0 : \theta^* \leq \theta_0$ versus $H_1 : \theta^* > \theta_0$	One-sided Hypothesis Test

We introduce some widely used tests next.

17.2 The Wald Test

The Wald test is based on a direct relationship between the $1 - \alpha$ confidence interval and a **size α** test. It can be used for testing simple hypotheses involving a scalar parameter.

Definition 116 (The Wald Test) Let $\widehat{\Theta}_n$ be an asymptotically normal estimator of the fixed and possibly unknown parameter $\theta^* \in \Theta \subset \mathbb{R}$ in the parametric IID experiment:

$$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} F(x_1; \theta^*) .$$

Consider testing:

$$H_0 : \theta^* = \theta_0 \quad \text{versus} \quad H_1 : \theta^* \neq \theta_0 .$$

Suppose that the null hypothesis is true and the estimator $\widehat{\Theta}_n$ of $\theta^* = \theta_0$ is asymptotically normal:

$$\theta^* = \theta_0, \quad \frac{\widehat{\Theta}_n - \theta_0}{\widehat{\text{se}}_n} \rightsquigarrow \text{Normal}(0, 1) .$$

Then, the Wald test based on the test statistic W is:

$$\text{Reject } H_0 \text{ when } |W| > z_{\alpha/2}, \text{ where } W := W((X_1, \dots, X_n)) = \frac{\widehat{\Theta}_n((X_1, \dots, X_n)) - \theta_0}{\widehat{\text{se}}_n} .$$

The rejection region for the Wald test is:

$$\mathbb{X}_R = \{x := (x_1, \dots, x_n) : |W(x_1, \dots, x_n)| > z_{\alpha/2}\} .$$

Proposition 117 (Asymptotic size of a Wald test) As the sample size n approaches infinity, the size of the Wald test approaches α :

$$\text{size} = \mathbf{P}_{\theta_0} (|W| > z_{\alpha/2}) \rightarrow \alpha .$$

Proof: Let $Z \sim \text{Normal}(0, 1)$. The size of the Wald test, i.e. the supreme power under H_0 is:

$$\begin{aligned} \text{size} &:= \sup_{\theta \in \Theta_0} \beta(\theta) := \sup_{\theta \in \{\theta_0\}} \mathbf{P}_{\theta}(x \in \mathbb{X}_R) = \mathbf{P}_{\theta_0}(x \in \mathbb{X}_R) \\ &= \mathbf{P}_{\theta_0} (|W| > z_{\alpha/2}) = \mathbf{P}_{\theta_0} \left(\frac{|\widehat{\Theta}_n - \theta_0|}{\widehat{\text{se}}_n} > z_{\alpha/2} \right) \\ &\rightarrow \mathbf{P} (|Z| > z_{\alpha/2}) \\ &= \alpha . \end{aligned}$$

Next, let us look at the power of the Wald test when the null hypothesis is false.

Proposition 118 (Asymptotic power of a Wald test) Suppose $\theta^* \neq \theta_0$. The power $\beta(\theta^*)$, which is the probability of correctly rejecting the null hypothesis, is approximately equal to:

$$\Phi \left(\frac{\theta_0 - \theta^*}{\widehat{\text{se}}_n} - z_{\alpha/2} \right) + \left(1 - \Phi \left(\frac{\theta_0 - \theta^*}{\widehat{\text{se}}_n} + z_{\alpha/2} \right) \right) ,$$

where, Φ is the DF of $\text{Normal}(0, 1)$ RV. Since $\widehat{\text{se}}_n \rightarrow 0$ as $n \rightarrow \infty$ the power increase with sample size n . Also, the power increases when $|\theta_0 - \theta^*|$ is large.

Now, let us make the connection between the size α Wald test and the $1 - \alpha$ confidence interval explicit.

Proposition 119 (The size Wald test) The size α Wald test rejects:

$$H_0 : \theta^* = \theta_0 \text{ versus } H_1 : \theta^* \neq \theta_0 \text{ if and only if } \theta_0 \notin C_n := (\hat{\theta}_n - \widehat{\text{se}}_{n,z_{\alpha/2}}, \hat{\theta}_n + \widehat{\text{se}}_{n,z_{\alpha/2}}).$$

Therefore, testing the hypothesis is equivalent to verifying whether the null value θ_0 is in the confidence interval.

Example 223 (Wald test for the mean waiting times at our Orbiter bus-stop) Let us use the Wald test to attempt to reject the null hypothesis that the mean waiting time at our Orbiter bus-stop is 10 minutes under an IID Exponential(λ^*) model. Let $\alpha = 0.05$ for this test. We can formulate this test as follows:

$$H_0 : \lambda^* = \lambda_0 = \frac{1}{10} \quad \text{versus} \quad H_1 : \lambda^* \neq \frac{1}{10}, \quad \text{where,} \quad X_1, \dots, X_{132} \stackrel{IID}{\sim} \text{Exponential}(\lambda^*) .$$

Based on Example 187 and Labwork 188 we obtained the 95% confidence interval to be [0.0914, 0.1290]. Since our null value $\lambda_0 = 0.1$ belongs to this confidence interval, we fail to reject the null hypothesis from a size $\alpha = 0.05$ Wald test.

We can use bootstrap-based confidence interval C_n in conjunction with Wald test as shown by the next example.

Example 224 (Wald test of the bootstrapped correlation coefficient) Recall the problem of estimating the confidence interval for the correlation coefficient between the LSAT scores (Y_1, \dots, Y_{15}) and the GPA (Z_1, \dots, Z_{15}) in Labwork 202. We assumed that the bivariate data $(Y_i, Z_i) \stackrel{IID}{\sim} F^*$, such that $F^* \in \{\text{all bivariate DFs}\}$. Suppose we are interested in testing the null hypothesis that the true correlation coefficient θ^* is 0:

$$H_0 : \theta^* = \theta_0 = 0 \quad \text{versus} \quad H_1 : \theta^* \neq 0, \quad \text{where} \quad \theta^* = \frac{\int \int (y - \mathbf{E}(Y))(z - \mathbf{E}(Z)) dF(y, z)}{\sqrt{\int (y - \mathbf{E}(Y))^2 dF(y) \int (z - \mathbf{E}(Z))^2 dF(z)}} .$$

Since the percentile-based 95% bootstrap confidence interval for the plug-in estimate of the correlation coefficient from Labwork 202 was [0.2346, 0.9296] and this interval does not contain 0, we can reject the null hypothesis that the correlation coefficient is 0 using a size $\alpha = 0.05$ Wald test.

17.3 A Composite Hypothesis Test

Often, we are interested in testing a composite hypothesis, i.e. one in which the null hypothesis is not a singleton set. We revisit the Orbiter waiting time problem from this perspective next.

Example 225 (Testing the Mean Waiting Time at an Orbiter Bus-stop) Let us test the following null hypothesis H_0 .

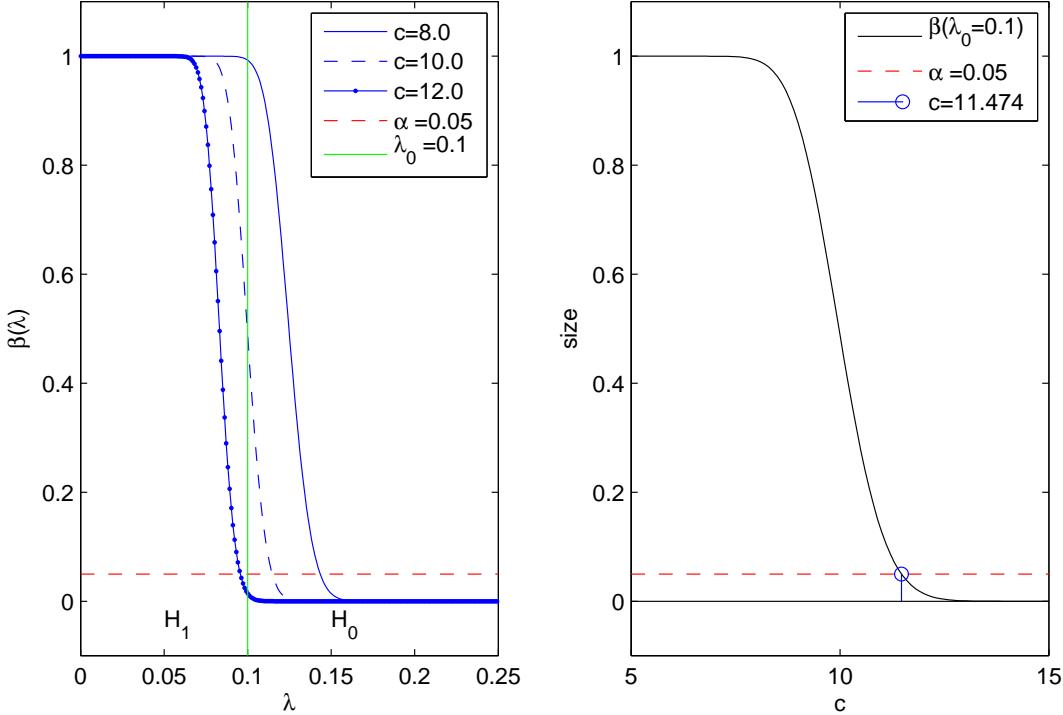
H_0 : The average waiting time at an Orbiter bus stop is less than or equal to 10 minutes.

H_1 : The average waiting time at an Orbiter bus stop is more than 10 minutes.

We have observations of $n = 132$ waiting times x_1, x_2, \dots, x_{132} at the Orbiter bus-stop with $\bar{x}_{132} = 9.0758$. Let us assume a parametric model, say,

$$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{Exponential}(\lambda^*)$$

Figure 17.1: Plot of power function $\beta(\lambda)$ for different values of the critical value c and the size α as function of the critical values.



with an unknown and fixed $\lambda^* \in \Lambda = (0, \infty)$. Since the parameter λ of an $\text{Exponential}(\lambda)$ RV is the reciprocal of the mean waiting time, we can formalise the above hypothesis testing problem of H_0 versus H_1 as follows:

$$H_0 : \lambda^* \in \Lambda_0 = [1/10, \infty) \quad \text{versus} \quad H_1 : \lambda^* \in \Lambda_1 = (0, 1/10)$$

Consider the test:

Reject H_0 if $T > c$.

where the test statistic $T = \bar{X}_n$ and the rejection region is:

$$\mathbb{X}_R = \{(x_1, x_2, \dots, x_n) : T(x_1, x_2, \dots, x_n) > c\} .$$

Since the sum of n IID $\text{Exponential}(\lambda)$ RVs is $\text{Gamma}(\lambda, n)$ distributed, the power function is:

$$\begin{aligned} \beta(\lambda) &= \mathbf{P}_\lambda (\bar{X}_n > c) = \mathbf{P}_\lambda \left(\sum_{i=1}^n X_i > nc \right) = 1 - \mathbf{P}_\lambda \left(\sum_{i=1}^n X_i \leq nc \right) \\ &= 1 - F(nc; \lambda, n) = 1 - \frac{1}{\Gamma(n)} \int_0^{\lambda nc} y^{n-1} \exp(-y) dy \\ &= 1 - \text{gammainc}(\lambda nc, n) \end{aligned}$$

Clearly, $\beta(\lambda)$ is a decreasing function of λ as shown in Figure 17.1. Hence the size of the test as a function of the critical region specified by the critical value c is:

$$\text{size} = \sup_{\lambda \in \Lambda_0} \beta(\lambda) = \sup_{\lambda \geq 1/10} \beta(\lambda) = \beta(1/10) = 1 - \text{gammainc}(132c/10, 132)$$

For a size $\alpha = 0.05$ test we numerically solve for the critical value c that satisfies:

$$0.05 = 1 - \text{gammainc}(132c/10, 132)$$

by trial and error as follows:

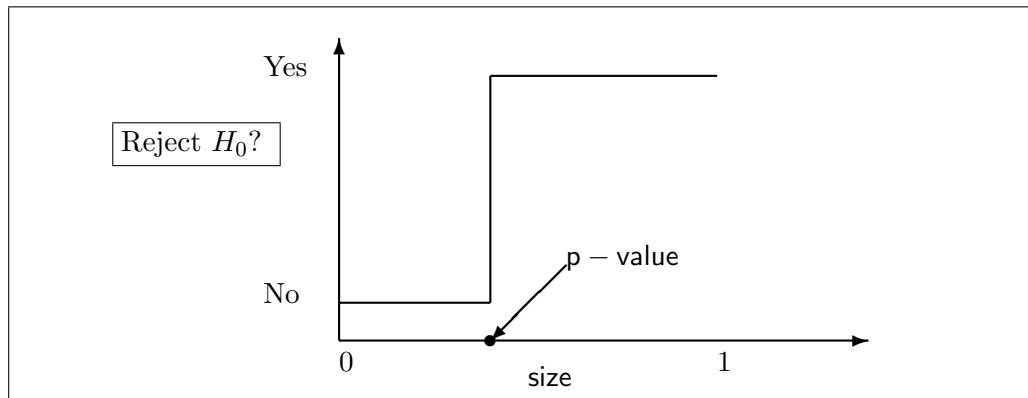
```
>> lambda0=1/10
lambda0 =
0.1000
>> S=@(C)(1-gammainc(lambda0*n*C,n)); % size as a function of c
>> Cs=[10 11 11.474 12 13] % some critical values c
Cs =
10.0000 11.0000 11.4740 12.0000 13.0000
>> Size=arrayfun(S,Cs) % corresponding size
Size =
0.4884 0.1268 0.0499 0.0143 0.0007
```

Thus, we reject H_0 when $\bar{X}_n > 11.4740$ for a level $\alpha = 0.05$ test. Since our observed test statistic $\bar{x}_{132} = 9.0758 < 11.4740$ we fail to reject the null hypothesis that the mean waiting time is less than or equal to 10 minutes. Therefore, there is no evidence that the Orbiter bus company is violating its promise of an average waiting time of no more than 10 minutes.

17.4 p-values

It is desirable to have a more informative decision than simply reporting "reject H_0 " or "fail to reject H_0 ." For instance, we could ask whether the test rejects H_0 for each size $= \alpha$. Typically, if the test rejects at size α it will also reject at a larger size $\alpha' > \alpha$. Therefore, there is a smallest size α at which the test rejects H_0 and we call this α the p – value of the test.

Figure 17.2: The smallest α at which a size α test rejects the null hypothesis H_0 is the p – value.



Definition 120 (p-value) Suppose that for every $\alpha \in (0, 1)$ we have a size α test with rejection region $\mathbb{X}_{R,\alpha}$ and test statistic T . Then,

$$p\text{-value} := \inf\{\alpha : T(X) \in \mathbb{X}_{R,\alpha}\} .$$

That is, the p – value is the smallest α at which a size α test rejects the null hypothesis.

If the evidence against H_0 is strong then the p – value will be small. However, a large p – value is not strong evidence in favour of H_0 . This is because a large p – value can occur for two reasons:

Table 17.3: Evidence scale against the null hypothesis in terms of the range of p-value.

p-value range	Evidence
(0, 0.01]	very strong evidence against H_0
(0.01, 0.05]	strong evidence against H_0
(0.05, 0.1]	weak evidence against H_0
(0.1, 1)	little or no evidence against H_0

1. H_0 is true.
2. H_0 is false but the test has low power.

Finally, it is important to realise that p-value is not the probability that the null hypothesis is true, i.e. $p\text{-value} \neq \mathbf{P}(H_0|x)$, where x is the data. The following tabulation of evidence scale is useful. The next proposition gives a convenient expression for the p-value for certain tests.

Proposition 121 (THe p-value of a hypothesis test) Suppose that the size α test based on the test statistic T and critical value c_α is of the form:

$$\text{Reject } H_0 \text{ if and only if } T := T((X_1, \dots, X_n)) > c_\alpha,$$

then

$$p\text{-value} = \sup_{\theta \in \Theta_0} \mathbf{P}_\theta(T((X_1, \dots, X_n)) \geq t := T((x_1, \dots, x_n))) ,$$

where, (x_1, \dots, x_n) is the observed data and t is the observed value of the test statistic T . In words, the p-value is the supreme probability under H_0 of observing a value of the test statistic the same as or more extreme than what was actually observed.

Let us revisit the Orbiter waiting times example from the p-value perspective.

Example 226 (p-value for the parametric Orbiter experiment) Let the waiting times at our bus-stop be $X_1, X_2, \dots, X_{132} \stackrel{IID}{\sim} \text{Exponential}(\lambda^*)$. Consider the following testing problem:

$$H_0 : \lambda^* = \lambda_0 = \frac{1}{10} \quad \text{versus} \quad H_1 : \lambda^* \neq \lambda_0 .$$

We already saw that the Wald test statistic is:

$$W := W(X_1, \dots, X_n) = \frac{\widehat{\Lambda}_n - \lambda_0}{\widehat{\text{se}}_n(\widehat{\Lambda}_n)} = \frac{\frac{1}{\widehat{X}_n} - \lambda_0}{\frac{1}{\sqrt{n\widehat{X}_n}}} .$$

The observed test statistic is:

$$w = W(x_1, \dots, x_{132}) = \frac{\frac{1}{\widehat{X}_{132}} - \lambda_0}{\frac{1}{\sqrt{132\widehat{X}_{132}}}} = \frac{\frac{1}{9.0758} - \frac{1}{10}}{\frac{1}{\sqrt{132 \times 9.0758}}} = 1.0618 .$$

Since, $W \rightsquigarrow Z \sim \text{Normal}(0, 1)$, the p-value for this Wald test is:

$$\begin{aligned} p\text{-value} &= \sup_{\lambda \in \Lambda_0} \mathbf{P}_\lambda(|W| > |w|) = \sup_{\lambda \in \{\lambda_0\}} \mathbf{P}_\lambda(|W| > |w|) = \mathbf{P}_{\lambda_0}(|W| > |w|) \\ &\rightarrow \mathbf{P}(|Z| > |w|) = 2\Phi(-|w|) = 2\Phi(-|1.0618|) = 2 \times 0.1442 = 0.2884 . \end{aligned}$$

Therefore, there is little or no evidence against H_0 that the mean waiting time under an IID Exponential model of inter-arrival times is exactly ten minutes.

17.5 Permutation Test for the equality of any two DFs

Permutation test is a non-parametric exact method for testing whether two distributions are the same. It is non-parametric because we do not impose any restrictions on the class of DFs that the unknown DF should belong to. It is exact because we do not have any asymptotic approximations involving sample size approaching infinity. So this test works for any sample size.

Formally, we suppose that:

$$X_1, X_2, \dots, X_m \stackrel{IID}{\sim} F^* \quad \text{and} \quad X_{m+1}, X_{m+2}, \dots, X_{m+n} \stackrel{IID}{\sim} G^*,$$

are two sets of independent samples. The possibly unknown DFs $F^*, G^* \in \{\text{all DFs}\}$. Now, consider the following hypothesis test:

$$H_0 : F^* = G^* \quad \text{versus} \quad H_1 : F^* \neq G^*.$$

Let our test statistic $T(X_1, \dots, X_m, X_{m+1}, \dots, X_{m+n})$ be some sensible one – T is large when F^* is too different from G^* , say:

$$T := T(X_1, \dots, X_m, X_{m+1}, \dots, X_{m+n}) = \text{abs} \left(\frac{1}{m} \sum_{i=1}^m X_i - \frac{1}{n} \sum_{i=m+1}^n X_i \right).$$

Then the idea of a permutation test is as follows:

1. Let $N := m + n$ be the pooled sample size and consider all $N!$ permutations of the observed data $x_{\text{obs}} := (x_1, x_2, \dots, x_m, x_{m+1}, x_{m+2}, \dots, x_{m+n})$.
2. For each permutation of the data compute the statistic $T(\text{permuted data } x)$ and denote these $N!$ values of T by $t_1, t_2, \dots, t_{N!}$.
3. Under $H_0 : X_1, \dots, X_m, X_{m+1}, \dots, X_{m+n} \stackrel{IID}{\sim} F^* = G^*$, each of the permutations of $x = (x_1, x_2, \dots, x_m, x_{m+1}, x_{m+2}, \dots, x_{m+n})$ has the same joint probability $\prod_{i=1}^{m+n} f(x_i)$, where $f(x_i) = dF(x_i) = dG(x_i)$. Therefore, the transformation of the data by our statistic T also has the same probability over the values of T , namely $\{t_1, t_2, \dots, t_{N!}\}$. Let \mathbf{P}_0 be this permutation distribution that is discrete and uniform over $\{t_1, t_2, \dots, t_{N!}\}$.
4. Let $t_{\text{obs}} := T(x_{\text{obs}})$ be the observed value of the statistic.
5. Assuming we reject H_0 when T is large, the p-value is:

$$\text{p-value} = \mathbf{P}_0(T \geq t_{\text{obs}}) = \frac{1}{N!} \left(\sum_{j=1}^{N!} \mathbb{1}(t_j \geq t_{\text{obs}}) \right), \quad \mathbb{1}(t_j \geq t_{\text{obs}}) = \begin{cases} 1 & \text{if } t_j \geq t_{\text{obs}} \\ 0 & \text{otherwise} \end{cases}$$

Let us look at a small example involving the diameters of coarse venus shells (*Dosinia anus*) that Guo Yaozong and Chen Shun found on the left and right sides of the New Brighton pier in Spring 2007. We are interested in testing the hypothesis that the distribution of shell diameters for this bivalve species is the same on both sides of the pier.

Example 227 (Guo-Chen Experiment with Venus Shell Diameters) Let us look at the first two samples x_1 and x_2 from the left of pier and the first sample from the right side of pier, namely x_3 . Since the permutation test is exact, we can use this small data set with merely three samples to conduct the following hypothesis test:

$$H_0 : X_1, X_2, X_3 \stackrel{IID}{\sim} F^* = G^*, \quad H_1 : X_1, X_2 \stackrel{IID}{\sim} F^*, X_3 \stackrel{IID}{\sim} G^*, \quad F^* \neq G^*.$$

Let us use the test statistic:

$$T(X_1, X_2, X_3) = \text{abs} \left(\frac{1}{2} \sum_{i=1}^2 X_i - \frac{1}{1} \sum_{i=2+1}^3 X_i \right) = \text{abs} \left(\frac{X_1 + x_2}{2} - \frac{X_3}{1} \right) .$$

The data giving the shell diameters in millimetres and t_{obs} are:

$$(x_1, x_2, x_3) = (52, 54, 58) \quad \text{and} \quad t_{\text{obs}} = \text{abs} \left(\frac{52 + 54}{2} - \frac{58}{1} \right) = \text{abs}(53 - 58) = \text{abs}(-5) = 5 .$$

Let us tabulate the $(2+1)! = 3! = 3 \times 2 \times 1 = 6$ permutations of the data $(x_1, x_2, x_3) = (52, 54, 58)$, the corresponding values of T and their probabilities under the null hypothesis, i.e., the permutation distribution $\mathbf{P}_0(T)$.

Permutation	t	$\mathbf{P}_0(T = t)$
(52, 54, 58)	5	$\frac{1}{6}$
(54, 52, 58)	5	$\frac{1}{6}$
(52, 58, 54)	1	$\frac{1}{6}$
(58, 52, 54)	1	$\frac{1}{6}$
(58, 54, 52)	4	$\frac{1}{6}$
(54, 58, 52)	4	$\frac{1}{6}$

From the table, we get:

$$\text{p-value} = \mathbf{P}_0(T \geq t_{\text{obs}}) = \mathbf{P}_0(T \geq 5) = \frac{1}{6} + \frac{1}{6} = \frac{2}{6} = \frac{1}{3} \approx 0.333 .$$

Therefore, there is little to no evidence against H_0 .

When the pooled sample size $N = m + n$ gets large, $N!$ would be too numerous to tabulate exhaustively. In this situation, we can use a Monte Carlo approximation of the p-value by generating a large number of random permutations of the data according to the following Steps:

Step 1: Compute the observed statistic $t_{\text{obs}} := T(x_{\text{obs}})$ of data $x_{\text{obs}} := (x_1, \dots, x_m, x_{m+1}, \dots, x_{m+n})$.

Step 2: Randomly permute the data and compute the statistic again from the permuted data.

Step 3: Repeat Step 2 B times and let t_1, \dots, t_B denote the resulting values (B is large, say 1000).

Step 4: The (Monte Carlo) approximate p-value is:

$$\frac{1}{B} \sum_{j=1}^B \mathbb{1}(t_j \geq t_{\text{obs}}) .$$

Next we implement the above algorithm on the full data set of Guo and Chen obtained from coarse venus shells sampled from the two sides of the New Brighton pier.

Labwork 228 (Approximate p-value of a permutation test of shell diameters) Test the null hypothesis that the distribution of the diameters of coarse venus shells are the same on both sides of the New Brighton pier.

Shells.m

```
% this data was collected by Guo Yaozong and Chen Shun as part of their STAT 218 project 2007
% coarse venus shell diameters in mm from left side of New Brighton Pier
left=[52 54 60 60 54 47 57 58 61 57 50 60 60 62 44 55 58 55 60 59 65 59 63 51 61 62 61 60 61 65 ...
43 59 58 67 56 64 47 64 60 55 58 41 53 61 60 49 48 47 42 50 58 48 59 55 59 50 47 47 33 51 61 61 ...
52 62 64 64 47 58 58 61 50 55 47 39 59 64 63 63 62 64 61 50 62 61 65 62 66 60 59 58 58 60 59 61 ...
55 55 62 51 61 49 52 59 60 66 50 59 64 64 62 60 65 44 58 63];
% coarse venus shell diameters in mm from right side of New Brighton Pier
right=[58 54 60 55 56 44 60 52 57 58 61 66 56 59 49 48 69 66 49 72 49 50 59 59 59 66 62 ...
44 49 40 59 55 61 51 62 52 63 39 63 52 62 49 48 65 68 45 63 58 55 56 55 57 34 64 66 ...
54 65 61 56 57 59 58 62 58 40 43 62 59 64 64 65 65 59 64 63 65 62 61 47 59 63 44 43 ...
59 67 64 60 62 64 65 59 55 38 57 61 52 61 61 60 34 62 64 58 39 63 47 55 54 48 60 55 ...
60 65 41 61 59 65 50 54 60 48 51 68 52 51 61 57 49 51 62 63 59 62 54 59 46 64 49 61];
Tobs=abs(mean(left)-mean(right));% observed test statistic
nleft=length(left); % sample size of the left-side data
nright=length(right); % sample size of the right-side data
ntotal=nleft+nright; % sample size of the pooled data
total=[left right]; % observed data -- ordered: left-side data followed by right-side data
B=10000; % number of bootstrap replicates
TB=zeros(1,B); % initialise a vector of zeros for the bootstrapped test statistics
ApproxPValue=0; % initialise an accumulator for approximate p-value
for b=1:B % enter the bootstrap replication loop
    % use MATLAB's randperm function to get a random permutation of indices{1,2,...,ntotal}
    PermutatedIndices=randperm(ntotal);
    % use the first nleft of the PermutatedIndices to get the bootstrapped left-side data
    Bleft=total(PermutatedIndices(1:nleft));
    % use the last nright of the PermutatedIndices to get the bootstrapped right-side data
    Bright=total(PermutatedIndices(nleft+1:ntotal));
    TB(b) = abs(mean(Bleft)-mean(Bright)); % compute the test statistic for the bootstrapped data
    if(TB(b)>Tobs) % increment the ApproxPValue accumulator by 1/B if bootstrapped value > Tobs
        ApproxPValue=ApproxPValue+(1/B);
    end
end
ApproxPValue % report the Approximate p-value
```

When we execute the script to perform a permutation test and approximate the p – value, we obtain:

```
>> Shells
ApproxPValue = 0.8576
```

Therefore, there is little or no evidence against the null hypothesis.

17.6 Pearson’s Chi-Square Test for Multinomial Trials

We derive the Chi-square distribution introduced by Karl Pearson in 1900 [*Philosophical Magazine*, Series 5, **50**, 157-175]. This historical work laid the foundations of modern statistics by showing why an experimenter cannot simply plot experimental data and just assert the correctness of his or her hypothesis. This derivation is adapted from Donald E. Knuth’s treatment [*Art of Computer Programming, Vol. II, Seminumerical Algorithms*, 3rd Ed., 1997, pp. 55-56]. We show how de Moivre, Multinomial, Poisson and the Normal random vectors conspire to create the Chi-square random variable.

Part 1: de Moivre trials

Consider n independent and identically distributed de Moivre($\theta_1, \dots, \theta_k$) random vectors (\vec{RV} s):

$$X_1, X_2, \dots, X_n \stackrel{IID}{\sim} \text{de Moivre}(\theta_1, \dots, \theta_k).$$

Recall from Model 11 that $X_1 \sim \text{de Moivre}(\theta_1, \dots, \theta_k)$ means $\mathbf{P}(X_1 = e_i) = \theta_i$ for $i \in \{1, \dots, k\}$, where e_1, \dots, e_k are ortho-normal basis vectors in \mathbb{R}^k . Thus, for each $i \in \{1, 2, \dots, n\}$, the corresponding X_i has k components, i.e. $X_i := (X_{i,1}, X_{i,2}, \dots, X_{i,k})$.

Part 2: Multinomial trial

Suppose we are only interested in the experiment induced by their sum:

$$Y := (Y_1, \dots, Y_k) := \sum_{i=1}^n X_i := \sum_{i=1}^n (X_{i,1}, X_{i,2}, \dots, X_{i,k}) = \left(\sum_{i=1}^n X_{i,1}, \sum_{i=1}^n X_{i,2}, \dots, \sum_{i=1}^n X_{i,k} \right) .$$

The $\vec{RV} Y$, being the sum of n IID $\text{de Moivre}(\theta_1, \dots, \theta_k)$ \vec{RV} s, is the $\text{Multinomial}(n, \theta_1, \dots, \theta_k)$ \vec{RV} of Model 18 and the probability that $Y := (Y_1, \dots, Y_k) = y := (y_1, \dots, y_k)$ is:

$$\frac{n!}{y_1! y_2! \cdots y_k!} \prod_{i=1}^k \theta_i^{y_i} .$$

The support of the $\vec{RV} Y$, i.e. the set of possible realisations of $y := (y_1, \dots, y_k)$ is:

$$\mathbb{Y} := \{(y_1, \dots, y_k) \in \mathbb{Z}_+^k : \sum_{i=1}^k y_i = n\} .$$

Part 3: Conditional sum of Poisson trials

Here we consider an alternative formulation of the $\text{Multinomial}(n, \theta_1, \dots, \theta_k)$ $\vec{RV} Y$. Suppose,

$$Y_1 \sim \text{Poisson}(n\theta_1), Y_2 \sim \text{Poisson}(n\theta_2), \dots, Y_k \sim \text{Poisson}(n\theta_k) ,$$

and that Y_1, \dots, Y_k are independent. Recall from Model 14 that $Y_i \sim \text{Poisson}(n\theta_i)$ means $\mathbf{P}(Y_i = y_i) = e^{-n\theta_i} (n\theta_i)^{y_i} / y_i!$ for $y_i \in \{0, 1, \dots\}$. Then, the joint probability probability of the $\vec{RV} (Y_1, \dots, Y_k)$ is the product of the independent Poisson probabilities:

$$\begin{aligned} \mathbf{P}((Y_1, \dots, Y_k) = (y_1, \dots, y_k)) &:= \mathbf{P}(Y_1 = y_1, \dots, Y_k = y_k) = \prod_{i=1}^k \mathbf{P}(Y_i = y_i) = \prod_{i=1}^k \frac{e^{-n\theta_i} (n\theta_i)^{y_i}}{y_i!} \\ &= \frac{\prod_{i=1}^k e^{-n\theta_i} n^{y_i} \theta_i^{y_i}}{\prod_{i=1}^k y_i!} = \left(e^{-n \sum_{i=1}^k \theta_i} n^{\sum_{i=1}^k y_i} \prod_{i=1}^k \theta_i^{y_i} \right) \frac{1}{\prod_{i=1}^k y_i!} \\ &= \frac{e^{-n} n^n \prod_{i=1}^k \theta_i^{y_i}}{\prod_{i=1}^k y_i!} . \end{aligned}$$

Now, the probability that sum $Y_1 + \dots + Y_k$ will equal n is obtained by summing over the probabilities of all $(y_1, \dots, y_k) \in \mathbb{Y}$:

$$\begin{aligned} \mathbf{P}\left(\sum_{i=1}^k Y_i = n\right) &= \sum_{\substack{(y_1, \dots, y_k) \\ \in \mathbb{Y}}} \mathbf{P}((Y_1, \dots, Y_k) = (y_1, \dots, y_k)) = \sum_{\substack{(y_1, \dots, y_k) \\ \in \mathbb{Y}}} \frac{e^{-n} n^n \prod_{i=1}^k \theta_i^{y_i}}{\prod_{i=1}^k y_i!} \\ &= e^{-n} n^n \underbrace{\frac{1}{n!} \left(\sum_{\substack{(y_1, \dots, y_k) \\ \in \mathbb{Y}}} \frac{n!}{\prod_{i=1}^k y_i!} \prod_{i=1}^k \theta_i^{y_i} \right)}_{=\mathbf{P}(\mathbb{Y})=1} = \frac{e^{-n} n^n}{n!} . \end{aligned}$$

Finally, the conditional probability that $(Y_1, \dots, Y_k) = (y_1, \dots, y_k)$ given $\sum_{i=1}^k Y_i = n$ is:

$$\begin{aligned}\mathbf{P}\left((Y_1, \dots, Y_k) = (y_1, \dots, y_k) \mid \sum_{i=1}^k Y_i = n\right) &= \frac{\mathbf{P}\left((Y_1, \dots, Y_k) = (y_1, \dots, y_k), \sum_{i=1}^k Y_i = n\right)}{\mathbf{P}\left(\sum_{i=1}^k Y_i = n\right)} \\ &= \frac{\mathbf{P}((Y_1, \dots, Y_k) = (y_1, \dots, y_k))}{\mathbf{P}\left(\sum_{i=1}^k Y_i = n\right)} = \frac{e^{-n} n^n \prod_{i=1}^k \theta_i^{y_i}}{\prod_{i=1}^k y_i!} \frac{n!}{e^{-n} n^n} = \frac{n!}{y_1! y_2! \cdots y_k!} \prod_{i=1}^k \theta_i^{y_i}.\end{aligned}$$

Therefore, we may also think of the random vector $Y := (Y_1, \dots, Y_k) \sim \text{Multinomial}(n, \theta_1, \dots, \theta_k)$ as k independent Poisson random variables, $Y_1 \sim \text{Poisson}(n\theta_1), \dots, Y_k \sim \text{Poisson}(n\theta_k)$, that have been conditioned on their sum $\sum_{i=1}^k Y_i$ being n .

Part 4: The Normal approximation of the centred and scaled Poisson

Recall from Model 14 that the expectation and variance of a RV $Y_i \sim \text{Poisson}(n\theta_i)$ are $\mathbf{E}(Y_i) = \mathbf{V}(Y_i) = n\theta_i$. Let Z_i be $\mathbf{E}(Y_i)$ -centred and $\sqrt{\mathbf{V}(Y_i)}$ -scaled Y_i and

$$Z_i := \frac{Y_i - \mathbf{E}(Y_i)}{\sqrt{\mathbf{V}(Y_i)}} = \frac{Y_i - n\theta_i}{\sqrt{n\theta_i}}.$$

The condition that $Y_1 + \dots + Y_k = n$ is equivalent to requiring that $\sqrt{\theta_1}Z_1 + \dots + \sqrt{\theta_k}Z_k = 0$, since:

$$\begin{aligned}\sum_{i=1}^k Y_i = n &\iff \sum_{i=1}^k Y_i - n = 0 \iff \sum_{i=1}^k Y_i - n \sum_{i=1}^k \theta_i = 0 \iff \sum_{i=1}^k Y_i - n\theta_i = 0 \\ &\iff \sum_{i=1}^k \frac{Y_i - n\theta_i}{\sqrt{n}} = 0 \iff \sum_{i=1}^k \sqrt{\theta_i} \frac{Y_i - n\theta_i}{\sqrt{n\theta_i}} = 0 \iff \sum_{i=1}^k \sqrt{\theta_i} Z_i = 0.\end{aligned}$$

Now consider the support of the RV $Z := (Z_1, \dots, Z_k)$ conditioned on $\sum_{i=1}^k \sqrt{\theta_i} Z_i = 0$, i.e. the hyper-plane of $(k-1)$ -dimensional vectors:

$$\mathbb{H} := \{(z_1, \dots, z_k) : \sqrt{\theta_1} z_1 + \dots + \sqrt{\theta_k} z_k = 0\}$$

Each $Z_i \rightsquigarrow \text{Normal}(0, 1)$ by the central limit theorem. Therefore, for large values of n , each Z_i is approximately distributed as the $\text{Normal}(0, 1)$ RV with PDF $f(z_i; 0, 1) = (2\pi)^{-1/2} \exp(-z_i^2/2)$. Since the Z_i s are independent except for the condition that they lie in \mathbb{H} , the point in a differential volume $dz_2 \dots dz_k$ of \mathbb{H} occur with probability approximately proportional to:

$$\exp(-z_1^2/2) \times \dots \times \exp(-z_k^2/2) = \exp(-(z_1^2 + \dots + z_k^2)/2)$$

Part 5: Chi-square distribution as the sum of squared Normals

We are interested in the sum of the area of squares with side-lengths Z_1, \dots, Z_k . Let V be the desired sum of squares:

$$V := \sum_{i=1}^k Z_i^2 = \sum_{i=1}^k \frac{(Y_i - n\theta_i)^2}{n\theta_i}, \quad \text{such that } Z_i \rightsquigarrow \text{Normal}(0, 1), \quad \sum_{i=1}^k \sqrt{\theta_i} Z_i = 0.$$

The probability that $V \leq v$ as $n \rightarrow \infty$ is:

$$\frac{\int_{(z_1, \dots, z_k) \in \mathbb{H} \text{ and } \sum_{i=1}^k z_i^2 \leq v} \exp(-(z_1^2 + \dots + z_k^2)/2) dz_2 \dots dz_k}{\int_{(z_1, \dots, z_k) \in \mathbb{H}} \exp(-(z_1^2 + \dots + z_k^2)/2) dz_2 \dots dz_k}$$

Since the $(k - 1)$ -dimensional hyper-plane \mathbb{H} passes through the origin of \mathbb{R}^k , the domain of integration in the numerator above is the interior of a $(k - 1)$ -dimensional hyper-sphere of radius \sqrt{v} that is centred at the origin. Using a transformation of the above ratio of integrals into generalised polar co-ordinates with radius χ and angles $\alpha_1, \dots, \alpha_{k-2}$, we get:

$$\frac{\int_{\chi^2 \leq v} \exp(-\chi^2/2) \chi^{k-2} g(\alpha_1, \dots, \alpha_{k-2}) d\chi d\alpha_1 \cdots d\alpha_{k-2}}{\int \exp(-\chi^2/2) \chi^{k-2} g(\alpha_1, \dots, \alpha_{k-2}) d\chi d\alpha_1 \cdots d\alpha_{k-2}},$$

for some function g of the angles [See Problem 15 in *Art of Computer Programming, Vol. II, Seminumerical Algorithms*, 3rd Ed., 1997, pp. 59]. The integration over the $(k - 2)$ angles results in the same factor that cancels between the numerator and the denominator. This yields the formula for the probability that $V \leq v$ as $n \rightarrow \infty$:

$$\lim_{n \rightarrow \infty} \mathbf{P}(V \leq v) = \frac{\int_0^{\sqrt{v}} \exp(-\chi^2/2) \chi^{k-2} d\chi}{\int_0^{\infty} \exp(-\chi^2/2) \chi^{k-2} d\chi}.$$

By substituting $t = \chi^2/2$, we can express the integrals in terms of the incomplete Gamma function defined as $\gamma(a, x) := \int_0^a \exp(-t) t^{a-1} dt$ as follows:

$$\mathbf{P}(V \leq v) = \gamma\left(\frac{k-1}{2}, \frac{v}{2}\right) / \Gamma\left(\frac{k-1}{2}\right).$$

This is the DF of the Chi-square distribution with $k - 1$ degrees of freedom.

Model 37 (Chi-square(k) RV) Given a parameter $k \in \mathbb{N}$ called degrees of freedom, we say that V is a Chi-square(k) RV if its PDF is:

$$f(v; k) := \frac{v^{(k/2)-1} e^{-v/2}}{2^{k/2} \Gamma(k/2)} \mathbf{1}_{\{v \in \mathbb{R}: v > 0\}}(v)$$

Also, $\mathbf{E}(V) = k$ and $\mathbf{V}(V) = 2k$.

We can use the test statistic:

$$T := T(Y_1, \dots, Y_k) = \frac{(Y_1 - n\theta_1^*)^2}{n\theta_1^*} + \cdots + \frac{(Y_k - n\theta_k^*)^2}{n\theta_k^*}$$

to test the null hypothesis H_0 that may be formalised in three equivalent ways:

$$\begin{aligned} H_0 : X_1, X_2, \dots, X_n &\stackrel{IID}{\sim} \text{de Moivre}(\theta_1^*, \dots, \theta_k^*) \text{ RV} \\ \iff H_0 : Y := (Y_1, \dots, Y_k) &\sim \sum_{i=1}^n X_i \sim \text{Multinomial}(n, \theta_1^*, \dots, \theta_k^*) \text{ RV} \\ \iff H_0 : Y_1 &\stackrel{IND}{\sim} \text{Poisson}(n\theta_1) \text{ RV}, \dots, Y_k \stackrel{IND}{\sim} \text{Poisson}(n\theta_k) \text{ RV given that } \sum_{i=1}^k Y_i = n \end{aligned}$$

We have seen that under H_0 , the test statistic $T \rightsquigarrow V \sim \text{Chi-square}(k - 1)$. Let t_{obs} be the observed value of the test statistic and let the upper alpha quantile be $\chi_{k-1, \alpha}^2 := F^{[-1]}(1 - \alpha)$, where F is the CDF of $V \sim \text{Chi-square}(k - 1)$. Hence the test:

Reject H_0 if $T > \chi_{k-1, \alpha}^2$ is an asymptotically size α test and the p-value = $\mathbf{P}(V > t_{\text{obs}})$.

Chapter 18

Nonparametric Density Estimation

This chapter is under .

18.1 Histogram

Let x_1, \dots, x_n be independent and identically distributed samples from a univariate continuous RV X with density f . The simplest nonparametric estimator for f is the histogram. It has some serious drawbacks. We have encountered the histogram earlier in 5.2. During that encounter we chose the number of bins in an *ad hoc* fashion, often resorting to the default number of bins of 10 in MATLAB's `hist` function.

A bin width that is too small when the number of bins is too large will give a histogram with many empty bins and many bins containing only a single data point. This is referred as *under-smoothing*. At the other extreme, a bin width that is too large due to a small number of bins results in a histogram that lacks details and results in *over-smoothing*. In both of these situations, the histograms do not represent the underlying density well.

Definition 122 (Density Histogram) Let x_1, \dots, x_n be a random univariate sample with density f and let S_f be the support of f . Let $S_x \subset S_f$ be a connected interval containing x_1, \dots, x_n and let $\{B_1, \dots, B_m\}$ be a finite contiguous partition of S_x into m bins of equal width b . For $x \in S_x$ and $B \in \{B_1, \dots, B_m\}$, let $B(x)$ denote the bin that contains x . The *density histogram* for x_1, \dots, x_n with *bin width* b is:

$$\hat{f}_n(x, b) = \frac{1}{nb} \sum_{i=1}^n I_{B(x)}(x_i). \quad (18.1)$$

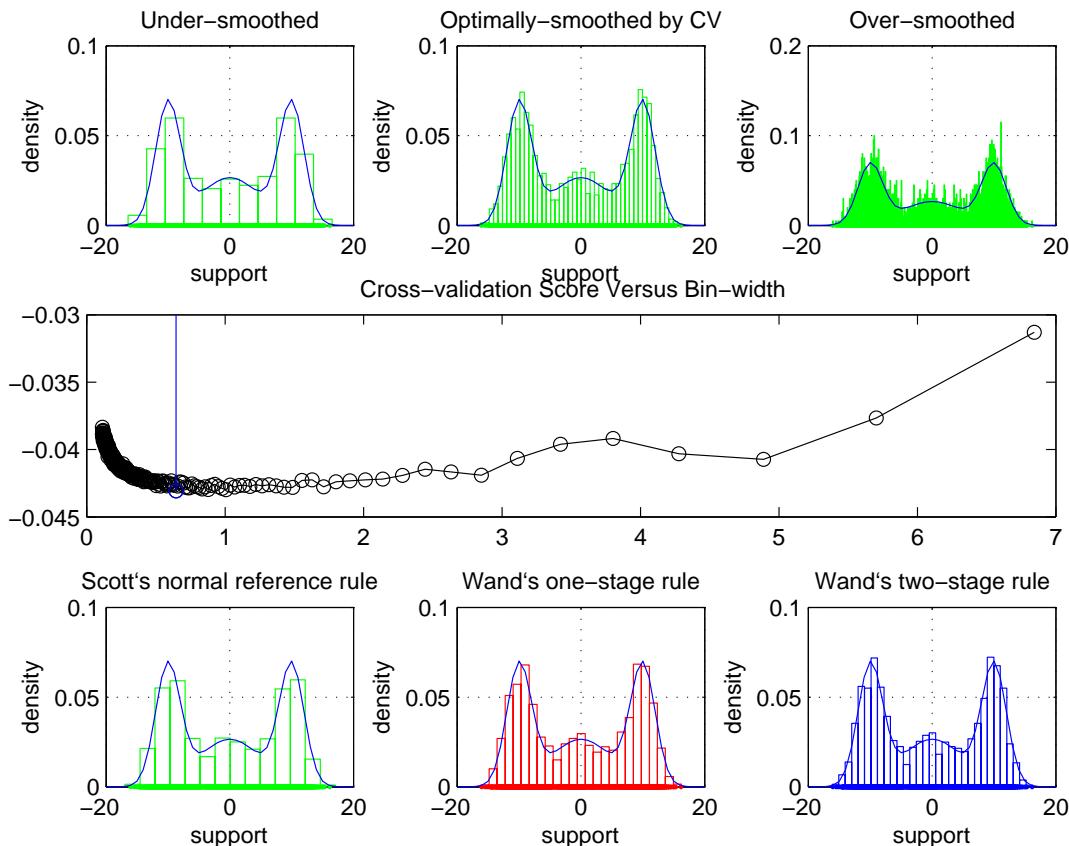
Hence, the density at x is estimated by counting the number of sample points in the bin containing x , and then appropriately normalising this number so that the area covered by the histogram is 1.

Now, let us consider the problem of estimating a histogram from 1500 samples simulated from the equi-weighted mixture of $\text{Normal}(0, 5^2)$, $\text{Normal}(10, 1)$ and $\text{Normal}(-10, 1)$ using the following code:

```
rand('twister',6898962)
randn('state',23121);
A=ceil(3*rand(1,2000));% Mixture Label vector
MuSs=[0 5; 10 2; -10 2];% 5 2; -5 2
x=arrayfun(@(i)(MuSs(i,1)+MuSs(i,2)*randn),A);
```

```
xgrid=-20:1:20;
pdf=NormalPdf(xgrid,MuSs(1,1), (MuSs(1,2))^2)/3 + NormalPdf(xgrid,MuSs(2,1), (MuSs(2,2))^2)/3 ...
+ NormalPdf(xgrid,MuSs(3,1), (MuSs(3,2))^2)/3;
```

Figure 18.1: Histogram estimates for the with nine different bin-widths of $\{2, 4, 6, 8, 11, 14, 17, 20, 35\}$. The fifth histogram with a bin-width of 11 is the optimal one with the right amount of smoothing. The ones with smaller bin-widths are under-smoothed and those with larger bin-widths are over-smoothed.



Drawbacks of the histogram

- (a) While densities of continuous random variables are continuous, the histogram is a step function with discontinuities.
- (b) The histogram does not use data efficiently¹.
- (c) Although only one parameter, the bin width, appears explicitly in the definition of the histogram, the use of the histogram requires the specification of a second parameter. This is the placement of the leftmost bin edge, which can strongly affect the resulting histogram.

Because of these drawbacks, the histogram should only be used as a graphical tool for exploratory data analysis.

¹With an optimally chosen bin width, the mean integrated squared error, $E[\int [\hat{f}_n(x, b) - f(x)]^2 dx]$, converges to 0 at a rate of $n^{-2/3}$.

Selection of histogram bin width

(a) *Sturges' rule*: The number of bins is given by:

$$m = \text{ceil}(1 + \log_2 n). \quad (18.2)$$

In practice, the bin width is usually obtained by:

$$b = \frac{x_{(n)} - x_{(1)}}{m}. \quad (18.3)$$

(b) *Scott's normal reference rule*:

$$b = 3.5\hat{\sigma}n^{-1/3}, \quad (18.4)$$

where $\hat{\sigma}$ is the sample standard deviation.

(c) *Freedman-Diaconis' rule*:

$$b = 2(\hat{q}_{0.75} - \hat{q}_{0.25})n^{-1/3}, \quad (18.5)$$

where \hat{q}_p is the sample p -quantile.

With Scott's bin width or Freedman-Diaconis' bin width, the number of bins can be obtained by:

$$m = \lceil \left(\frac{x_{(n)} - x_{(1)}}{b} \right) \rceil. \quad (18.6)$$

In practice, these methods for determining bin width should be used as starting points for trying several possible bin widths, with the final bin width chosen by visual inspection of the resulting histograms.

A suggestion for the placement of the leftmost bin edge is to use a *centred histogram*. For a histogram with m bins and bin width b , $mb \geq x_{(n)} - x_{(1)}$, and so let:

$$\delta = mb - (x_{(n)} - x_{(1)}). \quad (18.7)$$

To get the centred histogram, place the leftmost bin edge at $x_{(1)} - \delta/2$, so that the histogram extends equally by $\delta/2$ beyond the smallest and largest data values.

MATLAB function for density histogram

```
% histogram.m
% Plots density histogram for data in X.
%
% Usage: histogram(X,plotdata,bounds,colour);
%
% Input: X = row vector of data,
%        plotdata (binary) = plot data points?
%        bounds = [lower bound , upper bound] for possible X values,
%        colour (single-character string) = colour of histogram (default =
%        'y' for yellow),
%        bwmethod (optional, default = 2) = method of computing bin width:
%          0 = Scott's normal reference rule,
%          1 = Wand's one-stage rule,
%          2 = Wand's two-stage rule,
%          3 = manual,
%        bw = manual bin width if bwmethod = 3.
%
% Remark: Bin origin determined by centering the histogram, ie. so that
% left and right bin edges extend beyond min(X) and max(X) respectively
```

```
% by equal amounts.
%
% Reference: Wand M.P. (1997), "Data-based choice of histogram bin width",
% American Statistician 51, 59-64.

function [P,bw] = histogram(X,plotdata,bounds,colour,bwmethod,bw)

n = length(X); Y = zeros(1,n);

% Determine bin width:

n3 = n^(-1/3); n5 = n^(-.2); n7 = n^(-1/7);

Xsort = sort(X); xiq = Xsort(ceil(.75 * n)) - Xsort(ceil(.25 * n));
sdx = std(X);
if xiq == 0, sigma = sdx;
else, sigma = min([sdx (xiq / 1.349)]); end

if nargin == 3, bwmethod = 2; colour = 'y'; end
if nargin == 4, bwmethod = 2; end

if bwmethod == 0, bw = 3.4908 * sigma * n3;
elseif bwmethod == 1
    g11 = 1.3041 * sigma * n5;
    bw = 1.8171 * ((-psi(X,g11,2))^(1/3)) * n3;
elseif bwmethod == 2
    g22 = 1.2407 * sigma * n7;
    g21 = 0.9558 * ((psi(X,g22,4))^(1/2)) * n5;
    bw = 1.8171 * ((-psi(X,g21,2))^(1/3)) * n3;
end

% Determine bin origin:

xmin = min(X); xmax = max(X);
xrange = max(X) - xmin;
nbin = ceil(xrange / bw);
xoffset = (nbin * bw - xrange) / 2;
bbeg = max(bounds(1),xmin - xoffset); bend = min(bounds(2),xmax + xoffset);
bw = (bend - bbeg) / nbin;
BE = bbeg + bw * [0:nbin];

% Count frequencies:

for i = 1:nbin, P(i) = sum(X >= BE(i) & X < BE(i+1)); end
P = P / (n * bw);

% Plot histogram:

YY = [0 1 1 0]' * P; YY = YY(:);
XX = [1 1 0 0]' * BE(1:nbin) + [0 0 1 1]' * BE(2:(nbin+1)); XX = XX(:);

if plotdata
    plot(XX,YY,colour,X,Y,[colour '.']), grid
else
    plot(XX,YY,colour), grid
end
xlabel('support'), ylabel('density')

% Function: psi
% Required by histogram.m, kernel.m, sj91.m
%
% Reference: Wand M.P. (1997), "Data-based choice of histogram bin width",
% American Statistician 51, 59-64: Equations (2.2), (4.1)-(4.4).

function p = psi(X,g,r)
```

```

n = length(X); c = (n^(-2)) * (g^(-r-1));

if n < 1000

In = ones(1,n);

XX = X' * In - In' * X; XX = XX(:) / g; XX2 = XX .* XX;
Phi = gaussian(XX,1);

if r == 2
    p = c * (XX2 - 1)' * Phi;
elseif r == 4
    XX4 = XX2 .* XX2;
    p = c * (XX4 - 6 * XX2 + 3)' * Phi;
elseif r == 6
    XX4 = XX2 .* XX2; XX6 = XX4 .* XX2;
    p = c * (XX6 - 15 * XX4 + 45 * XX2 - 15)' * Phi;
else
    disp('Error: Input r for Function PSI must be 2, 4 or 6.'), return
end

else

xmin = min(X); m = 500; d = (max(X) - xmin) / (m - 1);
Im = ones(1,m); J = [1:m]; X = X - xmin; On = zeros(1,n);

for j = 1:m, C(j) = sum(max([(1 - abs((X / d) - j + 1));On])); end

CC = C' * C;
JJ = J' * Im - Im' * J; JJ = d * JJ(:) / g; JJ2 = JJ .* JJ;
CPhi = CC(:) .* gaussian(JJ,1);

if r == 2
    p = c * (JJ2 - 1)' * CPhi;
elseif r == 4
    JJ4 = JJ2 .* JJ2;
    p = c * (JJ4 - 6 * JJ2 + 3)' * CPhi;
elseif r == 6
    JJ4 = JJ2 .* JJ2; JJ6 = JJ4 .* JJ2;
    p = c * (JJ6 - 15 * JJ4 + 45 * JJ2 - 15)' * CPhi;
else
    disp('Error: Input r for Function PSI must be 2, 4 or 6.'), return
end

end

% Function: gaussian
% Gaussian probability density function.
% Generates normal probability density values corresponding to X.
% Inputs: X = Row vector of support points for which normal density values
%         are required,
%         S = Row vector of standard deviations.
% Output: NPD = Row vector of normal probability density values.

function NPD = gaussian(X,S)

NPD = exp(-(X .* X) ./ (2 * S .* S)) ./ (sqrt(2 * pi) * S);

```

Let x_1, \dots, x_n be a random univariate sample from a continuous distribution with density f . The most common nonparametric estimator for f is the histogram, which has some serious drawbacks.

18.1.1 Definition.

Let x_1, \dots, x_n be a random univariate sample with density f and let S_f be the support of f . Let $S_x \subset S_f$ be a connected interval containing x_1, \dots, x_n and let $\{B_1, \dots, B_m\}$ be a finite contiguous partition of S_x into m bins of equal width b . For $x \in S_x$ and $B \in \{B_1, \dots, B_m\}$, let $B(x)$ denote the bin that contains x . The *density histogram* for x_1, \dots, x_n with *bin width* b is:

$$\hat{f}_H(x, b) = \frac{1}{nb} \sum_{i=1}^n I_{B(x)}(x_i). \quad (18.8)$$

Hence, the density at x is estimated by counting the number of sample points in the bin containing x , and then appropriately normalising this number so that the area of the histogram equals 1.

Example 229 Suppose x_1, \dots, x_{500} is a random sample drawn from the standard normal distribution. A histogram for x_1, \dots, x_{500} is shown in the figure:

18.1.2 Drawbacks of the histogram

- (a) While densities of continuous random variables are continuous, the histogram is a step function with discontinuities.
- (b) The histogram does not use data efficiently².
- (c) Although only one parameter, the bin width, appears explicitly in the definition of the histogram, the use of the histogram requires the specification of a second parameter. This is the placement of the leftmost bin edge, which can strongly affect the resulting histogram.

Because of these drawbacks, the histogram should only be used as a graphical tool for exploratory analysis.

18.1.3 Selection of histogram bin width

- (a) *Sturges' rule*: The number of bins is given by:

$$m = \text{ceil}(1 + \log_2 n). \quad (18.9)$$

In practice, the bin width is usually obtained by:

$$b = \frac{x_{(n)} - x_{(l)}}{m}. \quad (18.10)$$

- (b) *Scott's normal reference rule*:

$$b = 3.5\hat{\sigma}n^{-1/3}, \quad (18.11)$$

where $\hat{\sigma}$ is the sample standard deviation.

- (c) *Freedman-Diaconis' rule*:

$$b = 2(\hat{q}_{0.75} - \hat{q}_{0.25})n^{-1/3}, \quad (18.12)$$

where \hat{q}_p is the sample p -quantile.

With Scott's bin width or Freedman-Diaconis' bin width, the number of bins can be obtained by:

$$m = \text{ceil}\left(\frac{x_{(n)} - x_{(l)}}{b}\right). \quad (18.13)$$

²With an optimally chosen bin width, the mean integrated squared error, $E[\int [\hat{f}_H(x, b) - f(x)]^2 dx]$, converges to 0 at a rate of $n^{-2/3}$.

In practice, these methods for determining bin width should be used as starting points for trying several possible bin widths, with the final bin width chosen by visual inspection of the resulting histograms. A bin width that is too small will give a histogram with many empty bins and many bins containing only a single data point. At the other extreme, a bin width that is too large results in a histogram that lacks details. In both of these situations, the histograms do not represent the underlying density well.

Example 230 (Shuttle data) Joint temperatures of the O-rings for each test firing or actual launch of the space shuttle rocket motor are shown in the table below (from *Presidential Commission on the Space Shuttle Challenger Accident*).

O-ring temperatures ($^{\circ}\text{F}$)										
31	40	45	49	52	53	57	58	58	60	
61	61	63	66	67	67	67	67	68	69	
70	70	70	70	72	73	75	75	76	76	
78	79	80	81	83	84					

A suggestion for the placement of the leftmost bin edge is to use a “centred histogram”. For a histogram with m bins and bin width b , $mb \geq x_{(n)} - x_{(l)}$, and so let:

$$\delta = mb - (x_{(n)} - x_{(l)}). \quad (18.14)$$

To get the centred histogram, place the leftmost bin edge at $x_{(l)} - \delta/2$, so that the histogram extends equally by $\delta/2$ beyond the smallest and largest data values.

18.2 Kernel density estimation

The *kernel density estimator* is a nonparametric estimator for f that is continuous, uses data more efficiently than the histogram, and has only one parameter.

Definition 123 Let x_1, \dots, x_n be a random univariate sample with density f , and let K be a function satisfying:

$$\int K(x)dx = 1. \quad (18.15)$$

The *kernel density estimator* for f with bandwidth $h > 0$ and kernel K is:

$$\hat{f}_K(x, h) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right). \quad (18.16)$$

We shall consider kernel functions that are densities and define:

$$K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right), \quad (18.17)$$

and so the kernel density estimator can be expressed as:

$$\hat{f}_K(x, h) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i), \quad (18.18)$$

which is a proper density.

Name	$K(x)$	Efficiency	σ_K
Epanechnikov	$\frac{3}{4}(1-x^2)I_{[-1,1]}(x)$	1	$\frac{1}{\sqrt{5}}$
Biweight	$\frac{15}{16}(1-x^2)^2I_{[-1,1]}(x)$	0.994	$\frac{1}{\sqrt{7}}$
Triweight	$\frac{35}{32}(1-x^2)^3I_{[-1,1]}(x)$	0.987	$\frac{1}{3}$
Triangle	$(1- x)I_{[-1,1]}(x)$	0.986	$\frac{1}{\sqrt{6}}$
Gaussian	$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$	0.951	1
Uniform	$\frac{1}{2}I_{[-1,1]}(x)$	0.930	$\frac{1}{\sqrt{3}}$

18.2.1 Examples of kernel functions

It turns out that the choice of the kernel has little impact on the performance of the kernel density estimator, and so the convenient Gaussian kernel is a popular choice. The choice of bandwidth, on the other hand, is critical.

Proposition 124 Let σ_K^2 be the variance of kernel function K . The variance of $K_h(x)$ is $h^2\sigma_K^2$.

Proof: Since $K_h(x)$ has mean 0, its variance is:

$$\sigma_K^2 = \int x^2 K_h(x) dx = \frac{1}{h} \int x^2 K(x/h) dx = h^2 \int (x/h)^2 K(x/h) d(x/h) = h^2 \sigma_K^2.$$

Note that for the Gaussian kernel, $\sigma_K = 1$, and therefore $\sigma_K = h$; i.e. the standard deviation of K_h is equal to its bandwidth.

Labwork 231 Using the same standard normal sample from Example 5.1.2, the density estimate obtained from a Gaussian kernel density estimator (with bandwidth 0.3) is shown in the figure, along with the histogram obtained previously.

```
n = 500;
x = randn(1,n);
bw = 0.3;
s = [-3:0.01:3]; % support points to compute density
f = normpdf(s,0,1); % true N(0,1) density
ns = length(s); % number of support points
fker = zeros(1,ns); % storage for kernel density
for i = 1:ns
    fker(i) = mean(normpdf(s(i),x,bw));
end
histogram(x,0,[ -inf inf], 'g'); hold on
plot(s,f,:r',s,fker,'-r')
```

We can think of the kernel as spreading a probability mass of $1/n$ associated with each sample point around its neighbourhood. To see what the Gaussian kernel density estimator does, suppose we have only five sample points, x_1, \dots, x_5 . The estimator puts a normal density with variance h^2 and a probability mass of $1/n$ at each sample point, and estimates the density at support point x by summing the contributions from these normal densities at x . This is illustrated in the figure, where the sample points are marked by ‘x’ on the support axis, the dashed curves are the normal densities whose probability masses are scaled to $1/n$, and the solid curve is the estimated density.

If f is continuous at x and $h \rightarrow 0$ and $nh \rightarrow \infty$ as $n \rightarrow \infty$, then for any $\epsilon > 0$:

$$\lim_{n \rightarrow \infty} P(|\hat{f}_K(x, h) - f(x)| \leq \epsilon) = 1. \quad (18.19)$$

18.2.2 Bandwidth selection

The use of the kernel density estimator requires the specification of the bandwidth, which plays a similar role as the histogram's bin width. A practical and easy-to-use bandwidth choice for the Gaussian kernel is *Scott's bandwidth*:

$$h = \hat{s}n^{-1/5}, \quad (18.20)$$

where $\hat{s} = \min\{\hat{\sigma}, (\hat{q}_{0.75} - \hat{q}_{0.25})/1.348\}$. A slightly different version³ of Scott's bandwidth is given by:

$$h = 1.06\hat{s}n^{-1/5}. \quad (18.21)$$

Since the multiplicative constant of 1.06 is very close to 1, the two bandwidths are very close and we can use either of them.

A more sophisticated bandwidth can be obtained by considering the minimisation of *asymptotic mean integrated squared error*:

$$AMISE = \lim_{n \rightarrow \infty} E[\int [\hat{f}_K(x, h) - f(x)]^2 dx]. \quad (18.22)$$

Using $R(g)$ to denote $\int g(x)^2 dx$, the resulting optimal bandwidth is given by:

$$h_{AMISE} = \left(\frac{R(K)}{n\sigma_K^4 R(f'')} \right)^{1/5}, \quad (18.23)$$

which depends on the unknown density f through $R(f'')$. An approximate bandwidth is obtained by plugging in an appropriate estimate of $R(f'')$ ⁴.

With an optimally chosen bandwidth, the mean integrated squared error of the kernel density estimator converges to 0 at rate $n^{4/5}$.

Let K and L be two kernel functions with standard deviations σ_K and σ_L respectively. Then for the resulting kernel densities to be approximately equal, their bandwidths, h_K and h_L , must satisfy:

$$h_K \sigma_K \approx h_L \sigma_L, \quad (18.24)$$

i.e. the standard deviations of K_h and L_h must be approximately equal.

Recall that the standard deviation of the Gaussian kernel function is 1, so we can use this result to obtain the bandwidth for some other kernel, e.g. kernel function K , from the bandwidth of the Gaussian kernel:

$$h_K \approx \frac{h_{gaussian}}{\sigma_K}. \quad (18.25)$$

For example, with the Epanechnikov kernel, $\sigma_K = 1/\sqrt{5}$, and so Scott's bandwidth for the Epanechnikov kernel is:

$$h_K \approx \sqrt{5}\hat{s}n^{-1/5}.$$

Labwork 232 The data in `geyser.txt` are 107 durations (in minutes) of the eruptions of the Old Faithful geyser. Compare the kernel density estimates for eruption duration using the Gaussian and Epanechnikov kernels with Scott's bandwidth.

³This version is also known as the *normal reference rule*.

⁴Sheather, S. J. and Jones, M. C. (1991), “A reliable data-based bandwidth selection method for kernel density estimation”, *Journal of the Royal Statistical Society, Series B*, 53, 683-690.

```

x = load('geyser.txt');
n = length(x);
n5 = n^(-0.2);
hscottgauss = min(std(x),iqr(x)/1.348) * n5 % Scott's bandwidth for Gaussian
    % kernel
hscottepan = sqrt(5) * hscottgauss % Scott's bandwidth for Epanechnikov kernel

```

Results:

```

hscottgauss = 0.4087
hscottepan = 0.9139

```

18.2.3 Adjustment at boundaries

Suppose we need to estimate a density f whose support S_f is bounded at one or both ends; frequently, $S_f = (0, \infty)$ or $S_f = (a, b)$. The presence of a boundary or boundaries may cause some of the kernels in the kernel density estimator to be truncated. Therefore, the estimator must be adjusted accordingly to ensure that the resulting estimate remains a density:

$$\hat{f}_K(x, h) = \frac{1}{n} \sum_{i=1}^n \frac{K_h(x - x_i)}{p_i}, \quad (18.26)$$

where:

$$p_i = \int_S K_h(x - x_i) dx. \quad (18.27)$$

Thus, $0 < p_i \leq 1$, and $p_i = 1$ if the corresponding kernel is not truncated.

If the Gaussian kernel is used and letting $\Phi(t; \mu, \sigma^2)$ denote the cumulative probability at t of the normal distribution with mean μ and variance σ^2 , then:

$$p_i = 1 - \Phi(0; x_i, h^2), \quad (18.28)$$

when $S_f = (0, \infty)$, and:

$$p_i = \Phi(b; x_i, h^2) - \Phi(a; x_i, h^2), \quad (18.29)$$

when $S_f = (a, b)$.

Labwork 233 Let x_1, \dots, x_{100} be a sample from an exponential distribution with parameter 1. Recall the the support of the exponential distribution is $x \geq 0$.

```

n = 100;
x = exprnd(1,1,n); % n exponential(1) random variables
n5 = n^(-0.2);
h = min(std(x),iqr(x)/1.348) * n5; % Scott's bandwidth for Gaussian kernel
s = [0:0.01:4]; % support points to compute density
f = exppdf(s,1); % true exponential(1) density
ns = length(s); % number of support points
fker1 = zeros(1,ns); % storage for kernel density without boundary correction
fker2 = zeros(1,ns); % storage for kernel density with boundary correction
p = 1 - normcdf(0,x,h); % boundary correction factors
for i = 1:ns
    fker1(i) = mean(normpdf(s(i),x,h)); % kernel density without boundary
        % correction
    fker2(i) = mean(normpdf(s(i),x,h)./p); % kernel density with boundary
        % correction
end
plot(s,f,'-b',s,fker2,'--b',s,fker1,:b')
legend('true density','kernel density with correction','kernel density without correction')

```

18.3 Extension to multivariate data

The extension of the kernel density estimator to multivariate data is straightforward.

Let x_1, \dots, x_n be independent and identically distributed random d -vectors with d -dimensional density f . The kernel density estimator for f is:

$$\hat{f}_K(x, H) = \frac{1}{n|H|} \sum_{i=1}^n K(H^{-1}(x - x_i)), \quad (18.30)$$

where H is a $d \times d$ nonsingular matrix, called the *bandwidth matrix*, and the kernel function K is a d -dimensional density. Let:

$$K_H(x) = \frac{K(H^{-1}x)}{|H|}, \quad (18.31)$$

so that:

$$\hat{f}_K(x, H) = \frac{1}{n} \sum_{i=1}^n K_H(x - x_i). \quad (18.32)$$

The kernel can be simplified by assuming that its components are independent (note that this does not make the components of x independent). This simplifies the kernel matrix to a diagonal matrix with diagonal elements, h_1, \dots, h_d . The kernel can then be expressed as a product of univariate kernels:

$$K_H(x) = K_{h_1, \dots, h_d}(x) = \prod_{j=1}^d K_{h_j}[x(j)], \quad (18.33)$$

where $x(j)$ denotes the j^{th} component of x . This gives the *product kernel density estimator*:

$$\hat{f}_K(x, h_1, \dots, h_d) = \frac{1}{n} \sum_{j=1}^n \left[\prod_{j=1}^d K_{h_j}[x(j) - x_i(j)] \right]. \quad (18.34)$$

18.3.1 Bandwidth selection

For a product kernel density estimator with the Gaussian kernel, *Scott's rule in d dimensions* is:

$$h_j = \hat{s}_j n^{-1/(4+d)}, \quad (18.35)$$

where $\hat{s}_j = \min(\hat{\sigma}_j, (\hat{q}_{0.75j} - \hat{q}_{0.25j})/1.348)$ is the estimate of scale for the j^{th} component that has been computed from $x_1(j), \dots, x_n(j)$. For some other kernel, the required bandwidth may be obtained by dividing h_j by the standard deviation of that kernel function, as in the univariate case.

With an optimally chosen bandwidth, the mean integrated squared error of the multivariate kernel density estimator converges to 0 at rate $n^{-4/(4+d)}$. Thus, its efficiency decreases rapidly with increasing dimension.

Labwork 234 The file `nr1.txt` contains data from 42 rugby league matches. The first column contains the length of game time, in seconds, until the first points are scored by a kick between the posts (penalty, drop goal or conversion); the second column contains the game time (in seconds) until the first try is scored. Denoting the log of the bivariate sample points by $(x_1, y_1), \dots, (x_{42}, y_{42})$, the product Gaussian kernel density estimator for the bivariate density is:

$$\hat{f}_\varphi(x, y, h_x, h_y) = \frac{1}{42} \sum_{i=1}^{42} \varphi(x, x_i, h_x^2) \varphi(y, y_i, h_y^2).$$

Using Scott's rule:

$$h_x = \hat{s}_x \times 42^{-1/6} \text{ and } h_y = \hat{s}_y \times 42^{-1/6}.$$

```

data = load('nrl.txt');
x = log(data(:,1));
y = log(data(:,2));
n = length(x);

% scatter plot:
plot(x,y,'r')
xlabel('x'), ylabel('y')
title('Scatter plot')
axis([3 9 3 9]), axis('square')
drawnow

n6 = n^(-1/6);
hx = min(std(x),iqr(x)/1.348) * n6;
hy = min(std(y),iqr(y)/1.348) * n6;
s = 3:.01:9;
ns = length(s);
phix = zeros(n,ns);
phiy = zeros(n,ns);
for i = 1:n
    phix(i,:) = normpdf(s,x(i),hx);
    phiy(i,:) = normpdf(s,y(i),hy);
end
fker = zeros(ns,ns);
for j = 1:ns
    for i = 1:ns
        fker(j,i) = phiy(:,j)' * phix(:,i) / n;
    end
end

% 3-D surface plot:
figure, mesh(s,s,fker)
xlabel('x'), ylabel('y'), zlabel('density')
colorbar, drawnow

% contour plot:
figure, contourf(s,s,fker,20)
xlabel('x'), ylabel('y')
title('Contour plot')
colorbar, axis('square')

```

18.4 Smoothed bootstrap

The *smoothed bootstrap* is a variation of the bootstrap idea that is used for continuous data. Instead of estimating the distribution function by the EDF and obtaining bootstrap samples by random sampling with replacement from the data values, the smoothed bootstrap estimates the density by a kernel density and generates bootstrap samples from it. After getting the bootstrap samples, the MSE, variance, bias and confidence intervals can be computed for an estimator of interest, in the same way as in the nonparametric bootstrap. If the kernel density is a good estimate of the data density, then the smoothed bootstrap can give a small improvement over the nonparametric bootstrap for continuous data.

18.4.1 Generating from a kernel density

Let:

$$\hat{f}_K(x, h) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i),$$

be the kernel density. To generate a new sample point from the kernel density, randomly pick one of the original data values, x_1, \dots, x_n . Suppose is picked, generate the new sample point from the kernel centred on x_i , i.e. from $K_h(x - x_i)$.

Labwork 235 Generate 1000 random values from the Gaussian kernel density, with Scott's bandwidth, for the geyser data.

```
m = 1000; % number of sample points to generate
x = load('geyser.txt');
n = length(x);
n5 = n^(-0.2);
h = min(std(x), iqr(x)/1.348) * n5; % Scott's bandwidth for Gaussian kernel
xcen = randsample(x,m,true); % randomly resample kernel centres from x
y = normrnd(xcen,h); % generate from Gaussian kernels centred at xcen
histogram(y,0,[0 inf], 'r');
```

18.4.2 Smoothed bootstrap procedure for generating bootstrap samples

Let x_1, \dots, x_n be IID random variables with unknown density $f(x)$.

- (a) Estimate the data density by a kernel density $\hat{f}_K(x, h)$.
- (b) Obtain N bootstrap samples, each of size n , by generating from $\hat{f}_K(x, h)$:

$$\begin{aligned} \{x_{1,1}, \dots, x_{1,n}\} &\sim \hat{f}_K(x, h) \\ M \\ \{x_{N,1}, \dots, x_{N,n}\} &\sim \hat{f}_K(x, h) \end{aligned}$$

Labwork 236 Let us revisit Example 4.2.5 to obtain a 0.95 percentile interval for the median amount of sodium using the smoothed bootstrap with a Gaussian kernel density. Recall that the percentile interval provided by the nonparametric bootstrap was (75.05, 77).

```
x = load('sodium.txt'); % load data from text file and store in x
N = 100000; % number of bootstrap samples
n = length(x); % determine number of data values
nN = n * N;
alpha = 0.05;
alpha2 = alpha / 2;
alpha21 = 1 - alpha2;
n5 = n^(-0.2);
h = min(std(x), iqr(x)/1.348) * n5; % Scott's bandwidth for Gaussian kernel
xcen = randsample(x,nN,true); % randomly resample kernel centres from x
xboot = normrnd(xcen,h); % generate from Gaussian kernels centred at xcen
xboot = reshape(xboot,n,N); % organise resampled values into N columns of n
% values each so that each column is a bootstrap
% sample of size n
xbootmed = median(xboot); % medians of bootstrap samples
xbootmedsort = sort(xbootmed); % sort medians in increasing order
% (1-alpha) percentile interval:
[xbootmedsort(ceil(N*alpha2)) xbootmedsort(N*alpha21)]
```

Results:

```
ans = 74.8616 76.9451
```

Therefore, a 0.95 percentile interval for the median amount of sodium, using the smoothed bootstrap with a Gaussian kernel density, is (74.86, 76.95).

Point-wise confidence band for density using the smoothed bootstrap (see Scott⁵ pp. 259-260; Hardle et al.⁶ section 3.5).

18.5 Exercises

Exercise 237 Consider the two-component normal mixture density given by:

$$f(x) = 0.5\varphi(x, 4, 1) + 0.5\varphi(x, 9, 4),$$

where $\varphi(x, \mu, \sigma^2)$ denotes the normal density with mean μ and variance σ^2 , evaluated at x .

(a) A sample point, x , can be generated from the mixture density by the following algorithm:

Generate $u \sim U[0, 1]$.

If $u \leq 0.5$

 Generate $x \sim N(4, 1)$

Else

 Generate $x \sim N(9, 4)$.

Implement a MATLAB function to generate from the mixture density and use it to get 100 sample points.

(b) Using the 100 sample points from part (a), estimate the underlying density using a histogram with appropriate bin width. Explain clearly how you arrived at your choice of bin width.

(c) Using the 100 sample points from part (a), estimate the underlying density using a Gaussian kernel density estimator. Explore bandwidths ranging from 0.3 to 1.9 to decide on an appropriate one.

Exercise 238 The Bart Simpson density is given by:

$$f(x) = 0.5\varphi(x, 0, 1) + 0.1 \sum_{j=0}^4 \varphi(x, 0.5j - 1, 0.01).$$

To see why it is called the Bart Simpson density, plot the density for a sequence of x values from -3 to 3 (in steps of 0.01, for example).

(a) Write a MATLAB function to generate sample points from the Bart Simpson density and use it to obtain 1000 sample points.

⁵Scott, D.W. (1992), *Multivariate Density Estimation: Theory, Practice and Visualization*, Wiley.

⁶Hardle, W., Muller, M., Sperlich, S. and Werwatz, A. (2004), *Nonparametric and Semiparametric Models*, e-book at www.quantlet.com/mdstat/scripts/spm/html/spmhmltoc.html

- (b) Using the 1000 sample points from part (a), estimate the underlying density using a histogram with appropriate bin width. Explain clearly how you arrived at your choice of bin width.
- (c) Using the 1000 sample points from part (a), estimate the underlying density using a Gaussian kernel density estimator with bandwidths of 0.005, 0.05 and 0.5. Comment on the performance of each of these bandwidths.

Exercise 239 The data in `ceo.txt` contain the ages (column 1) and salaries (column 2, in thousands of dollars) of the CEOs of 59 companies.

- (a) Obtain a Gaussian kernel density estimate for the salaries, using Scott's bandwidth. Compare the transformation method and boundary correction method for handling the boundary at zero.
- (b) Use the nonparametric bootstrap to obtain a point-wise 0.95 BCA confidence band for the density of the salaries.

Exercise 240 The second column of `glass.txt` contains measurements of the refractive index of 214 glass specimens collected in forensic work.

- (a) Obtain a Gaussian kernel density estimate for refractive index, using Scott's bandwidth.
- (b) Use the nonparametric bootstrap to obtain a point-wise 0.95 BCA confidence band for the density of refractive index.

Exercise 241 The data in `whale.txt` are the times of 121 bowhead whale calf sightings during the 2001 spring migration. The time of each sighting is expressed as the number of hours since midnight of April 5, when the first adult whale was sighted.

- (a) Plot a density histogram for the data and superimpose onto it a Gaussian kernel density estimate. Explain your choice of histogram bin width and kernel density bandwidth. For the kernel density estimate, compare the transformation method and the boundary correction method for handling the boundary at zero.
- (b) Obtain a 0.95 BCA interval for the median sighting time, using the nonparametric bootstrap with 10,000 bootstrap samples.
- (c) Repeat Part (b) using the smoothed bootstrap with 10,000 bootstrap samples.

Exercise 242 The data in `density.txt` are 29 measurements of the density of the earth, which were made by Henry Cavendish in 1789.

- (a) Plot a density histogram for the data and superimpose onto it a Gaussian kernel density estimate. Explain your choice of histogram bin width and kernel density bandwidth. For the kernel density estimate, compare the transformation method and the boundary correction method for handling the boundary at zero.
- (b) Obtain a 0.95 BCA interval for the mean density of the earth, using the nonparametric bootstrap with 10,000 bootstrap samples.
- (c) Repeat Part (b) using the smoothed bootstrap with 10000 bootstrap samples.

Exercise 243 Work through Example 5.3.3 for the NRL data.

Exercise 244 The data in `infrared.txt` are measurements of infrared emissions from 628 objects beyond our galaxy. The two columns contain total flux measurements for two different wavelength bands - the first column shows the 12 micrometre band and the second column shows the 100 micrometre band. Estimate the bivariate density for the log of the data using a product Gaussian kernel density estimator with Scott's bandwidth. Plot the estimated density using a three-dimensional surface plot and a two-dimensional contour plot.

Chapter 19

Bayesian Experiments

This chapter is under .

19.1 A Bayesian Quincunx

19.2 Conjugate Families

19.3 Bayesian Model Selection

Chapter 20

Statistical Learning

This chapter is under .

20.1 Supervised Learning

20.2 Unsupervised Learning

20.3 Classification

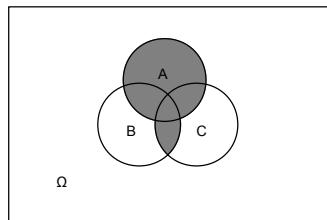
20.4 Regression

Answers to Selected Exercises

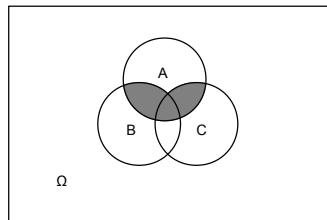
Answer (Ex. 1.1) — By operating with Ω , T , L and S we can obtain the answers as follows:

- | | |
|---|--|
| (a) $T \cap L = \{L_3\}$ | (f) $S \cap L = \emptyset$ |
| (b) $T \cap S = \emptyset$ | (g) $S^c \cap L = \{L_1, L_2, L_3\} = L$ |
| (c) $T \cup L = \{T_1, T_2, T_3, L_3, L_1, L_2\}$ | (h) $T^c = \{L_1, L_2, S_1, S_2, S_3, \dots, S_{50}\}$ |
| (d) $T \cup L \cup S = \Omega$ | (i) $T^c \cap L = \{L_1, L_2\}$ |
| (e) $S^c = \{T_1, T_2, T_3, L_3, L_1, L_2\}$ | (j) $T^c \cap T = \emptyset$ |

Answer (Ex. 1.3) — We can check $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ from the following sketch:



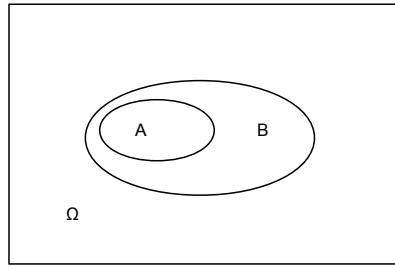
Answer (Ex. 1.3) — We can check $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ from the following sketch:
We can check $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ from the following sketch:



Answer (Ex. 1.4) — To illustrate the idea that $A \subseteq B$ if and only if $A \cup B = B$, we need to illustrate two implications:

- 1.if $A \subseteq B$ then $A \cup B = B$ and
- 2.if $A \cup B = B$ then $A \subseteq B$.

The following Venn diagram illustrates the two implications clearly.



Answer (Ex. 2.1) — (a) $\mathbf{P}(\{Z\}) = 0.1\% = \frac{0.1}{100} = 0.001$

(b) $\mathbf{P}(\text{'picking any letter'}) = \mathbf{P}(\Omega) = 1$

(c) $\mathbf{P}(\{E, Z\}) = \mathbf{P}(\{E\} \cup \{Z\}) = \mathbf{P}(\{E\}) + \mathbf{P}(\{Z\}) = 0.13 + 0.001 = 0.131$, by Axiom (3)

(d) $\mathbf{P}(\text{'picking a vowel'}) = \mathbf{P}(\{A, E, I, O, U\}) = (7.3\% + 13.0\% + 7.4\% + 7.4\% + 2.7\%) = 37.8\%$, by the addition rule for mutually exclusive events, rule (2).

(e) $\mathbf{P}(\text{'picking any letter in the word WAZZZUP'}) = \mathbf{P}(\{W, A, Z, U, P\}) = 14.4\%$, by the addition rule for mutually exclusive events, rule (2).

(f) $\mathbf{P}(\text{'picking any letter in the word WAZZZUP or a vowel'}) =$

$\mathbf{P}(\{W, A, Z, U, P\}) + \mathbf{P}(\{A, E, I, O, U\}) - \mathbf{P}(\{A, U\}) = 14.4\% + 37.8\% - 10\% = 42.2\%$, by the addition rule for two arbitrary events, rule (3).

Answer (Ex. 2.2) — 1. $\{BB, BW, WB, WW\}$

2. $\{RRRR, RRRR, RRLR, RLRR, LRRR, RLRL, RRLL, LLRR, LRLR, LRRL, RLLR, LLLL, LLLR, LLRL, LRLL, RLLL\}$

3. $\{6, 16, 26, 36, 46, 56, 116, 126, 136, 146, 156, 216, 226, 236, 246, 256, \dots\}$

Answer (Ex. 2.3) — 1. The sample space $\Omega = \{W, A, I, M, K, R\}$.

2. Since there are eleven letters in WAIMAKARIRI the probabilities are:

$$\mathbf{P}(\{W\}) = \frac{1}{11}, \mathbf{P}(\{A\}) = \frac{3}{11}, \mathbf{P}(\{I\}) = \frac{3}{11}, \mathbf{P}(\{M\}) = \frac{1}{11}, \mathbf{P}(\{K\}) = \frac{1}{11}, \mathbf{P}(\{R\}) = \frac{2}{11}.$$

3. By the complementation rule, the probability of not choosing the letter R is:

$$1 - \mathbf{P}(\text{choosing the letter R}) = 1 - \frac{2}{11} = \frac{9}{11}.$$

Answer (Ex. 2.4) — 1. First, the sample space is: $\Omega = \{B, I, N, G, O\}$.

2. The probabilities of simple events are:

$$\mathbf{P}(B) = \mathbf{P}(I) = \mathbf{P}(N) = \mathbf{P}(G) = \mathbf{P}(O) = \frac{15}{75} = \frac{1}{5}.$$

3. Using the addition rule for mutually exclusive events,

$$\begin{aligned}
 \mathbf{P}(\Omega) &= \mathbf{P}(\{\mathbf{B}, \mathbf{I}, \mathbf{N}, \mathbf{G}, \mathbf{O}\}) \\
 &= \mathbf{P}(\{\mathbf{B}\} \cup \{\mathbf{I}\} \cup \{\mathbf{N}\} \cup \{\mathbf{G}\} \cup \{\mathbf{O}\}) \\
 &= \mathbf{P}(\mathbf{B}) + \mathbf{P}(\mathbf{I}) + \mathbf{P}(\mathbf{N}) + \mathbf{P}(\mathbf{G}) + \mathbf{P}(\mathbf{O}) \quad \text{simplifying notation} \\
 &= \frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} \\
 &= 1
 \end{aligned}$$

4. Since the events $\{\mathbf{B}\}$ and $\{\mathbf{I}\}$ are disjoint,

$$\mathbf{P}(\{\mathbf{B}\} \cup \{\mathbf{I}\}) = \mathbf{P}(\mathbf{B}) + \mathbf{P}(\mathbf{I}) = \frac{1}{5} + \frac{1}{5} = \frac{2}{5}.$$

5. Using the addition rule for two arbitrary events we get,

$$\begin{aligned}
 \mathbf{P}(C \cup D) &= \mathbf{P}(C) + \mathbf{P}(D) - \mathbf{P}(C \cap D) \\
 &= \mathbf{P}(\{\mathbf{B}, \mathbf{I}, \mathbf{G}\}) + \mathbf{P}(\{\mathbf{G}, \mathbf{I}, \mathbf{N}\}) - \mathbf{P}(\{\mathbf{G}, \mathbf{I}\}) \\
 &= \frac{3}{5} + \frac{3}{5} - \frac{2}{5} \\
 &= \frac{4}{5}.
 \end{aligned}$$

Answer (Ex. 2.5) — We can assume that the first shot is independent of the second shot so we can multiply the probabilities here.

For case A, there is only one shot so the probability of hitting at least once is $\frac{1}{2}$.

For case B, the probability of missing both shots is $\frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9}$, so the probability hitting some target at least once is

$$1 - \mathbf{P}(\text{missing the target both times}) = 1 - \frac{4}{9} = \frac{5}{9}$$

Therefore, case B has the greater probability of hitting the target at least once.

Answer (Ex. 2.6) — 1. The sample space is

$$\begin{aligned}
 &\{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), \\
 &(3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), \\
 &(5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6)\}
 \end{aligned}$$

Note: Order matters here. For example, the outcome “16” refers to a “1” on the first die and a “6” on the second, whereas the outcome “61” refers to a “6” on the first die and a “1” on the second.

2. First tabulate all possible sums as follows:

+	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

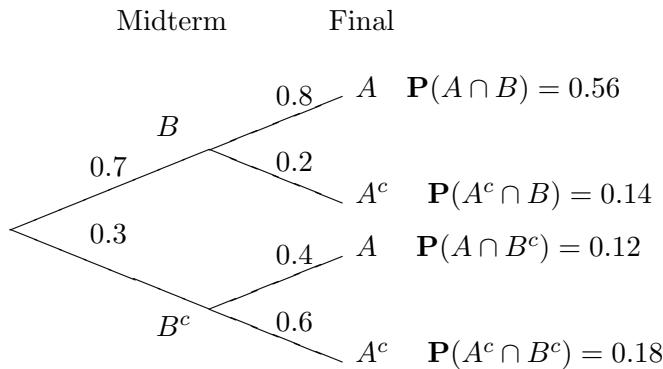
Let A be the event *the sum is 5* and B be the event *the sum is 6*, then A and B are mutually exclusive events with probabilities

$$\mathbf{P}(A) = \frac{4}{36} \quad \text{and} \quad \mathbf{P}(B) = \frac{5}{36}.$$

Therefore,

$$\mathbf{P}(4 < \text{sum} < 7) = \mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B) = \frac{4}{36} + \frac{5}{36} = \frac{1}{4}$$

Answer (Ex. 2.7) — First draw a tree with the first split based on the outcome of the midterm test and the second on the outcome of the final exam. Note that the probabilities involved in this second branch are *conditional* probabilities that depend on the outcome of the midterm test. Let A be the event that the student passes the final exam and let B be the event that the student passes the midterm test.



Then the probability of passing the final exam is:

$$\mathbf{P}(A) = 0.56 + 0.12 = 0.68.$$

To do this with formulae, partitioning according to the midterm test result and using the multiplication rule, we get:

$$\begin{aligned} \mathbf{P}(A) &= \mathbf{P}(A \cap B) + \mathbf{P}(A \cap B^c) \\ &= \mathbf{P}(A|B)\mathbf{P}(B) + \mathbf{P}(A|B^c)\mathbf{P}(B^c) \\ &= (0.8)(0.7) + (0.4)(0.3) = 0.68 \end{aligned}$$

Answer (Ex. 2.8) — Let A be the event that bottles are produced by machine 1; and A^c is the event that bottles are produced by machine 2. R denotes the event that the bottles are rejected; and R^c denotes the event that the bottles are accepted. We know the following probabilities:

$$\mathbf{P}(A) = 0.75 \quad \text{and} \quad \mathbf{P}(A^c) = 0.25$$

$$\mathbf{P}(R|A) = \frac{1}{20} \quad \text{and} \quad \mathbf{P}(R^c|A) = \frac{19}{20}$$

$$\mathbf{P}(R|A^c) = \frac{1}{30} \quad \text{and} \quad \mathbf{P}(R^c|A^c) = \frac{29}{30}$$

We want $\mathbf{P}(A|R^c)$ which is give by

$$\mathbf{P}(A|R^c) = \frac{\mathbf{P}(R^c \cap A)}{\mathbf{P}(R^c)} = \frac{\mathbf{P}(R^c \cap A)}{\mathbf{P}(R^c \cap A) + \mathbf{P}(R^c \cap A^c)}$$

where,

$$\mathbf{P}(R^c \cap A) = \mathbf{P}(R^c|A)\mathbf{P}(A) = \frac{19}{20} \times 0.75$$

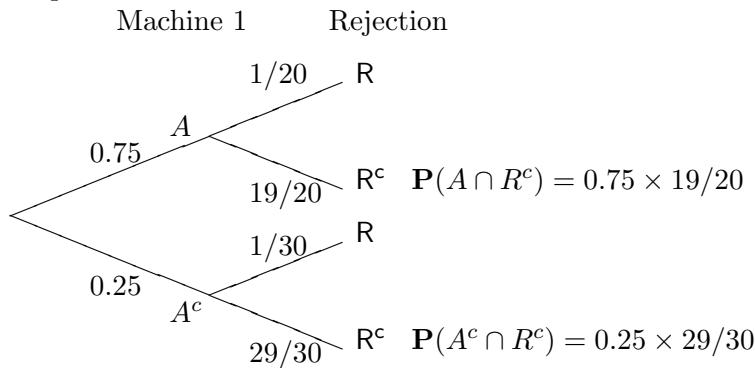
and,

$$\mathbf{P}(R^c \cap A^c) = \mathbf{P}(R^c|A^c)\mathbf{P}(A^c) = \frac{29}{30} \times 0.25$$

Therefore,

$$\mathbf{P}(A|R^c) = \frac{\frac{19}{20} \times 0.75}{\frac{19}{20} \times 0.75 + \frac{29}{30} \times 0.25} \approx 0.747$$

The tree diagram for this problem is:



So the required probability is

$$\mathbf{P}(A|R^c) = \frac{\mathbf{P}(R^c \cap A)}{\mathbf{P}(R^c)} = \frac{\frac{19}{20} \times 0.75}{\frac{19}{20} \times 0.75 + \frac{29}{30} \times 0.25} \approx 0.747$$

Answer (Ex. 2.9) — Let the event that a micro-chip is defective be D , and the event that the test is correct be C . So the probability that the micro-chip is defective is $P(D) = 0.05$, and the probability that it is effective is $P(D^c) = 0.95$.

The probability that the test correctly detects a defective micro-chip is the conditional probability $P(C|D) = 0.8$, and the probability that if a good micro-chip is tested but the test declares it is defective is the conditional probability $P(C^c|D^c) = 0.1$. Therefore, we also have the probabilities $P(C^c|D) = 0.2$, and $P(C|D^c) = 0.9$.

Moreover, the probability that a micro-chip is defective, and has been declared as defective is

$$P(C \cap D) = P(C|D)P(D) = 0.8 \times 0.05 = 0.04.$$

The probability that a micro-chip is effective, and has been declared as effective is

$$P(C \cap D^c) = P(C|D^c)P(D^c) = 0.9 \times 0.95 = 0.855.$$

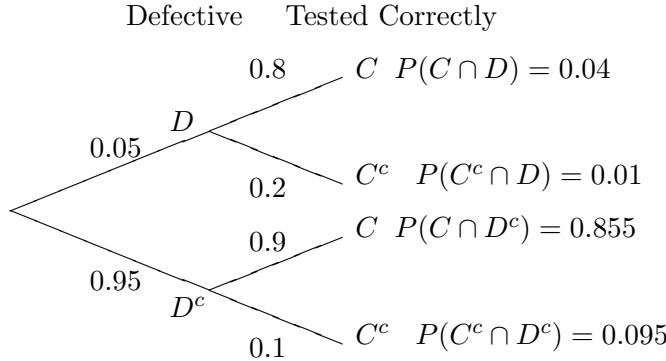
The probability that a micro-chip is defective, and has been declared as effective is

$$P(C^c \cap D) = P(C^c|D)P(D) = 0.2 \times 0.05 = 0.01.$$

The probability that a micro-chip is effective, and has been declared as defective is

$$P(C^c \cap D^c) = P(C^c|D^c)P(D^c) = 0.1 \times 0.95 = 0.095.$$

The tree diagram for these events and probabilities is:



- (a) If a micro-chip is tested to be good, it could be defective but tested incorrectly, or it could be effective and tested correctly. Therefore, the probability that the micro-chip is tested good, but it is actually defective is

$$\frac{P(C^c \cap D)}{P(C^c \cap D) + P(C \cap D^c)} = \frac{0.01}{0.01 + 0.855} \approx 0.012$$

- (b) Similarly, the probability that a micro-chip is tested to be defective, but it was good is

$$\frac{P(C^c \cap D^c)}{P(C \cap D) + P(C^c \cap D^c)} = \frac{0.095}{0.095 + 0.04} \approx 0.704$$

- (c) The probability that both the micro-chips are effective, and have been tested and determined to be good, is

$$\left(\frac{P(C \cap D^c)}{P(C^c \cap D) + P(C \cap D^c)} \right)^2$$

and so the probability that at least one is defective is:

$$1 - \left(\frac{P(C \cap D^c)}{P(C^c \cap D) + P(C \cap D^c)} \right)^2 = 1 - \left(\frac{0.855}{0.01 + 0.855} \right)^2 \approx 0.023$$

Answer (Ex. 2.10) — (a) Let F_1 be the event a gale of force 1 occurs, let F_2 be the event a gale of force 2 occurs and F_3 be the event a gale of force 3 occurs. Now we know that

$$P(F_1) = \frac{2}{3}, \quad P(F_2) = \frac{1}{4}, \quad P(F_3) = \frac{1}{12}.$$

If D is the event that a gale causes damage, then we also know the following conditional probabilities:

$$P(D|F_1) = \frac{1}{4}, \quad P(D|F_2) = \frac{2}{3}, \quad P(D|F_3) = \frac{5}{6}.$$

The probability that a reported gale causes damage is

$$P(D) = P(D \cap F_1) + P(D \cap F_2) + P(D \cap F_3)$$

where

$$P(D \cap F1) = P(D|F1)P(F1) = \frac{1}{4} \times \frac{2}{3} = \frac{1}{6},$$

$$P(D \cap F2) = P(D|F2)P(F2) = \frac{2}{3} \times \frac{1}{4} = \frac{1}{6},$$

and

$$P(D \cap F3) = P(D|F3)P(F3) = \frac{5}{6} \times \frac{1}{12} = \frac{5}{72}.$$

Hence

$$P(D) = \frac{1}{6} + \frac{1}{6} + \frac{5}{72} = \frac{29}{72}$$

(b) Knowing that the gale did cause damage we can calculate the probabilities that it was of the various forces using the probabilities in (a) as follows (Note: $P(D \cap F1) = P(F1 \cap D)$ etc.):

$$P(F1|D) = \frac{P(F1 \cap D)}{P(D)} = \frac{1/6}{29/72} = \frac{12}{29}$$

$$P(F2|D) = \frac{P(F2 \cap D)}{P(D)} = \frac{1/6}{29/72} = \frac{12}{29}$$

$$P(F3|D) = \frac{P(F3 \cap D)}{P(D)} = \frac{5/72}{29/72} = \frac{5}{29}$$

(c) First note that the probability that a reported gale does NOT cause damage is:

$$P(D^c) = 1 - P(D) = 1 - \frac{29}{72} = \frac{43}{72}.$$

Now we need to find probabilities like $P(F1 \cap D^c)$. The best way to do this is to use the partitioning idea of the “Total Probability Theorem”, and write:

$$P(F1) = P(F1 \cap D^c) + P(F1 \cap D),$$

Rearranging this gives

$$P(F1 \cap D^c) = P(F1) - P(F1 \cap D)$$

and so

$$P(F1|D^c) = \frac{P(F1 \cap D^c)}{P(D^c)} = \frac{P(F1) - P(F1 \cap D)}{P(D^c)} = \frac{2/3 - 1/6}{43/72} = \frac{36}{43}.$$

Similarly,

$$P(F2|D^c) = \frac{P(F2 \cap D^c)}{P(D^c)} = \frac{P(F2) - P(F2 \cap D)}{P(D^c)} = \frac{1/4 - 1/6}{43/72} = \frac{6}{43},$$

and

$$P(F3|D^c) = \frac{P(F3 \cap D^c)}{P(D^c)} = \frac{P(F3) - P(F3 \cap D)}{P(D^c)} = \frac{1/12 - 5/72}{43/72} = \frac{1}{43}.$$

Answer (Ex. 6.1) — $\mathbf{P}(X = 3)$ does not satisfy the condition that $0 \leq \mathbf{P}(A) \leq 1$ for any event A . If Ω is the sample space, then $\mathbf{P}(\Omega) = 1$ and so the correct probability is

$$\mathbf{P}(X = 3) = 1 - 0.07 - 0.10 - 0.32 - 0.40 = 0.11.$$

Answer (Ex. 6.2) — 1. Tabulate the values for the probability mass function as follows:

x	1	2	3	4	5
$\mathbf{P}(X = x)$	0.1	0.2	0.2	0.2	0.3

so the distribution function is:

$$F(x) = \mathbf{P}(X \leq x) = \begin{cases} 0 & \text{if } 0 \leq x < 1 \\ 0.1 & \text{if } 1 \leq x < 2 \\ 0.3 & \text{if } 2 \leq x < 3 \\ 0.5 & \text{if } 3 \leq x < 4 \\ 0.7 & \text{if } 4 \leq x < 5 \\ 1 & \text{if } x \geq 5 \end{cases}$$

The graphs of $f(x)$ and $F(x)$ for random variable X are shown below:

2. The probability that the machine needs to be replaced during the first 3 years is:

$$\mathbf{P}(X \leq 3) = \mathbf{P}(X = 1) + \mathbf{P}(X = 2) + \mathbf{P}(X = 3) = 0.1 + 0.2 + 0.2 = 0.5.$$

(This answer is easily seen from the distribution function of X .)

3. The probability that the machine needs no replacement during the first three years is

$$\mathbf{P}(X > 3) = 1 - \mathbf{P}(X \leq 3) = 0.5.$$

Answer (Ex. 6.3) — Assuming that the probability model is being built from the observed relative frequencies, the probability mass function is:

$$f(x) = \begin{cases} \frac{176}{200} & x = 1 \\ \frac{22}{200} & x = 2 \\ \frac{2}{200} & x = 3 \end{cases}$$

Answer (Ex. 6.4) — (a)

x	3	4	5	6	7	8	9	10	11	12	13
$F(x) = \mathbf{P}(X \leq x)$	0.07	0.08	0.17	0.18	0.34	0.59	0.79	0.82	0.84	0.95	1.00

(b) (i) $\mathbf{P}(X \leq 5) = F(5) = 0.17$

(ii) $\mathbf{P}(X < 12) = \mathbf{P}(X \leq 11) = F(11) = 0.84$

$$(iii) \mathbf{P}(X > 9) = 1 - \mathbf{P}(X \leq 9) = 1 - F(9) = 1 - 0.79 = 0.21$$

$$(iv) \mathbf{P}(X \geq 9) = 1 - \mathbf{P}(X < 9) = 1 - \mathbf{P}(X \leq 8) = 1 - 0.59 = 0.41$$

$$(v) \mathbf{P}(4 < X \leq 9) = F(9) - F(4) = 0.79 - 0.08 = 0.71$$

$$(vi) \mathbf{P}(4 < X < 11) = \mathbf{P}(4 < X \leq 10) = F(10) - F(4) = 0.82 - 0.08 = 0.74$$

Answer (Ex. 6.5) — Since we are sampling without replacement,

$$\mathbf{P}(X = 0) = \frac{4}{10} \cdot \frac{3}{9} = \frac{2}{15} \quad (\text{one way of drawing two right screws}),$$

$$\mathbf{P}(X = 1) = \frac{6}{10} \cdot \frac{4}{9} + \frac{4}{10} \cdot \frac{6}{9} = \frac{8}{15} \quad (\text{two ways of drawing one left and one right screw}),$$

$$\mathbf{P}(X = 2) = \frac{6}{10} \cdot \frac{5}{9} = \frac{1}{3} \quad (\text{one way of drawing two left screws}).$$

So the probability mass function of X is:

$$f(x) = \mathbf{P}(X = x) = \begin{cases} \frac{2}{15} & \text{if } x = 0 \\ \frac{8}{15} & \text{if } x = 1 \\ \frac{1}{3} & \text{if } x = 2 \end{cases}$$

The required probabilities are:

1.

$$\mathbf{P}(X \leq 1) = \mathbf{P}(X = 0) + \mathbf{P}(X = 1) = \frac{2}{15} + \frac{8}{15} = \frac{2}{3}$$

2.

$$\mathbf{P}(X \geq 1) = \mathbf{P}(X = 1) + \mathbf{P}(X = 2) = \frac{8}{15} + \frac{1}{3} = \frac{13}{15}$$

3.

$$\mathbf{P}(X > 1) = \mathbf{P}(X = 2) = \frac{1}{3}$$

Answer (Ex. 6.6) — 1. Since f is a probability mass function,

$$\sum_{x=0}^{\infty} \frac{k}{2^x} = 1, \quad \text{that is,} \quad k \sum_{x=0}^{\infty} \frac{1}{2^x} = 1.$$

Now $\sum_{x=0}^{\infty} \frac{1}{2^x}$ is a geometric series with common ratio $r = \frac{1}{2}$ and first term $a = 1$, and so has sum

$$S = \frac{a}{1-r} = \frac{1}{1-\frac{1}{2}} = 2$$

Therefore,

$$2k = 1, \quad \text{that is,} \quad k = \frac{1}{2}.$$

2. From (a), the probability mass function of f is

$$f(x) = \frac{\frac{1}{2}}{2^x} = \frac{1}{2^{x+1}}. \quad (x = 0, 1, 2, \dots)$$

Now

$$\mathbf{P}(X \geq 4) = 1 - \mathbf{P}(X < 4) = 1 - \mathbf{P}(X \leq 3)$$

where

$$\begin{aligned} \mathbf{P}(X \leq 3) &= \sum_{x=0}^3 \frac{1}{2^{x+1}} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \\ &= \frac{8}{16} + \frac{4}{16} + \frac{2}{16} + \frac{1}{16} \\ &= \frac{15}{16}. \end{aligned}$$

That is, $\mathbf{P}(X \geq 4) = \frac{1}{16}$.

Answer (Ex. 6.7) — Note that $\theta = \frac{1}{2}$ here.

1. X has probability mass function

$$f(x) = \begin{cases} \binom{4}{0} \frac{1^0}{2} \frac{1^4}{2} = \frac{1}{16} & x = 0 \\ \binom{4}{1} \frac{1^1}{2} \frac{1^3}{2} = \frac{4}{16} & x = 1 \\ \binom{4}{2} \frac{1^2}{2} \frac{1^2}{2} = \frac{6}{16} & x = 2 \\ \binom{4}{3} \frac{1^3}{2} \frac{1^1}{2} = \frac{4}{16} & x = 3 \\ \binom{4}{4} \frac{1^4}{2} \frac{1^0}{2} = \frac{1}{16} & x = 4 \end{cases}$$

2. The required probabilities are:

$$\mathbf{P}(X = 0) = f(0) = \frac{1}{16}$$

$$\mathbf{P}(X = 1) = f(1) = \frac{4}{16}$$

$$\mathbf{P}(X \geq 1) = 1 - \mathbf{P}(X = 0) = 1 - f(0) = \frac{15}{16}$$

$$\mathbf{P}(X \leq 3) = f(0) + f(1) + f(2) + f(3) = \frac{15}{16}$$

Answer (Ex. 6.8) — 1.If the random variable X denotes the number of type AB blood donors in the sample of 15, then X has a binomial distribution with $n = 15$ and $\theta = 0.05$. Therefore

$$\mathbf{P}(X = 1) = \binom{15}{1} (0.05)^1 (0.95)^{14} = 0.366 \quad (\text{3 sig. fig.}) .$$

2.If the random variable X denotes the number of type B blood donors in the sample of 15, then X has a binomial distribution with $n = 15$ and $\theta = 0.10$. Therefore

$$\begin{aligned} \mathbf{P}(X \geq 3) &= 1 - \mathbf{P}(X = 0) - \mathbf{P}(X = 1) - \mathbf{P}(X = 2) \\ &= 1 - \binom{15}{0} (0.1)^0 (0.9)^{15} - \binom{15}{1} (0.1)^1 (0.9)^{14} - \binom{15}{2} (0.1)^2 (0.9)^{13} \\ &= 1 - 0.2059 - 0.3432 - 0.2669 \\ &= 0.184 \quad (\text{to 3 sig. fig.}) \end{aligned}$$

3.If the random variable X denotes the number of type O or type A blood donors in the sample of 15, then X has a binomial distribution with $n = 15$ and $\theta = 0.85$. Therefore

$$\begin{aligned} \mathbf{P}(X > 10) &= \mathbf{P}(X = 11) + \mathbf{P}(X = 12) + \mathbf{P}(X = 13) + \mathbf{P}(X = 14) + \mathbf{P}(X = 15) \\ &= \binom{15}{11} (0.85)^{11} (0.15)^4 + \binom{15}{12} (0.85)^{12} (0.15)^3 \\ &\quad + \binom{15}{13} (0.85)^{13} (0.15)^2 + \binom{15}{14} (0.85)^{14} (0.15)^1 + \binom{15}{15} (0.85)^{15} (0.15)^0 \\ &= 0.1156 + 0.2184 + 0.2856 + 0.2312 + 0.0874 \\ &= 0.938 \quad (\text{to 3 sig. fig.}) \end{aligned}$$

4.If the random variable X denotes the number of blood donors that are *not* of type A blood donors in the sample of 15, then X has a binomial distribution with $n = 15$ and $\theta = 0.6$. Therefore

$$\begin{aligned} \mathbf{P}(X < 5) &= \mathbf{P}(X = 0) + \mathbf{P}(X = 1) + \mathbf{P}(X = 2) + \mathbf{P}(X = 3) + \mathbf{P}(X = 4) \\ &= \binom{15}{0} (0.6)^0 (0.4)^{15} + \binom{15}{1} (0.6)^1 (0.4)^{14} + \binom{15}{2} (0.6)^2 (0.4)^{13} \\ &\quad + \binom{15}{3} (0.6)^3 (0.4)^{12} + \binom{15}{4} (0.6)^4 (0.4)^{11} \\ &= 0.0000 + 0.0000 + 0.0003 + 0.0016 + 0.0074 \\ &= 0.009 \quad (\text{to 3 DP.}) \end{aligned}$$

Answer (Ex. 6.9) — This is a Binomial experiment with parameters $\theta = 0.1$ and $n = 10$, and so

$$\mathbf{P}(X \geq 1) = 1 - \mathbf{P}(X < 1) = 1 - \mathbf{P}(X = 0) ,$$

where

$$\mathbf{P}(X = 0) = \binom{10}{0} 0.1^0 0.9^{10} \approx 0.3487 .$$

Therefore, the probability that the target will be hit at least once is

$$1 - 0.3487 \approx 0.6513 .$$

Answer (Ex. 6.10) — 1. Since $f(x)$ is a (continuous) probability density function which integrates to one,

$$\int_{-4}^4 kdx = 1 .$$

That is,

$$\begin{aligned} kx \Big|_{-4}^4 &= 1 \\ k(4 - (-4)) &= 1 \\ 8k &= 1 \\ k &= \frac{1}{8} \end{aligned}$$

2. First note that if $x < -4$, then

$$F(x) = \int_{-\infty}^x 0 dv = 0 .$$

If $-4 \leq x \leq 4$, then

$$\begin{aligned} F(x) &= \int_{-\infty}^{-4} 0 dv + \int_{-4}^x \frac{1}{8} dv \\ &= 0 + \left[\frac{1}{8} v \right]_{-4}^x \\ &= \frac{1}{8}(x + 4) \end{aligned}$$

If $x \geq 4$, then

$$\begin{aligned} F(x) &= \int_{-\infty}^{-4} 0 dv + \int_{-4}^4 \frac{1}{8} dv + \int_4^x 0 dv \\ &= 0 + \left[\frac{1}{8} v \right]_{-4}^4 + 0 \\ &= 1 \end{aligned}$$

Hence

$$F(x) = \begin{cases} 0 & x < -4 \\ \frac{1}{8}(x + 4) & -4 \leq x \leq 4 \\ 1 & x \geq 4 \end{cases}$$

3. The graphs of $f(x)$ and $F(x)$ for random variable X are as follows:

Answer (Ex. 6.11) — 1. Since the distribution function is $F(t; \lambda) = 1 - \exp(-\lambda t)$,

$$\mathbf{P}(t > \tau) = 1 - \mathbf{P}(t < \tau) = 1 - F(\tau; \lambda = 0.01) = 1 - (1 - e^{-0.01\tau}) = e^{-0.01\tau} .$$

2. Set

$$\mathbf{P}(t > \tau) = e^{-0.01\tau} = \frac{1}{2}$$

and solve for τ to get then $\tau = -100 \times \log(0.5) = 69.3$ (3 sig. fig.).

Answer (Ex. 6.12) — We are given that 537 flying bombs hit an area A of south London made up of $24 \times 24 = 576$ small equal-sized areas, say A_1, A_2, \dots, A_{576} . Assuming the hits were purely random over A the probability that a particular bomb will hit a given small area, say A_i , is $\frac{1}{576}$. Let X denote the number of hits that a small area A_i receives in this German raid. Since 537 bombs fell over A , we can model X as $\text{Binomial}(n = 537, \theta = \frac{1}{576})$ that is counting the number of ‘successes’ (for German bombers) with probability θ in a sequence of $n = 537$ independent Bernoulli(θ) trials. Finally, we can approximate this $\text{Binomial}(n = 537, \theta = \frac{1}{576})$ random variable by $\text{Poisson}(\lambda)$ random variable with $\lambda = n\theta = \frac{537}{576} \approx 0.933$. Using the probability mass function formula for $\text{Poisson}(\lambda = 0.933)$ random variable X we can obtain the probabilities and compare them with the relative frequencies from the data as follows:

x	observed frequency	observed relative frequency	Prob of x hits
0	229	$229/576 = 0.398$	$f(0; 0.933) = 0.394$
1	211	$211/576 = 0.366$	$f(1; 0.933) = 0.367$
2	93	$93/576 = 0.161$	$f(2; 0.933) = 0.171$
3	35	$35/576 = 0.0608$	$f(3; 0.933) = 0.0532$
4	7	$7/576 = 0.0122$	$f(4; 0.933) = 0.0124$
≥ 5	1	$1/576 = 0.00174$	$1 - \sum_{x=0}^4 f(x; 0.933) = 0.00275$

Answer (Ex. 6.13) — Since 2 defects exist on every 100 meters, we would expect 6 defects on a 300 meter tape. If X is the number of defects on a 300 meter tape, then X is Poisson with $\lambda = 6$ and so the probability of zero defects is

$$\mathbf{P}(X = 0; 6) = \frac{6^0}{0!} e^{-6} = 0.0025 .$$

Answer (Ex. 6.14) — Since X is $\text{Poisson}(\lambda)$ random variable with $\lambda = 0.5$, $\mathbf{P}(X \geq 2)$ is the probability of observing two or more particles during any given second.

$$\mathbf{P}(X \geq 2) = 1 - \mathbf{P}(X < 2) = 1 - \mathbf{P}(X = 1) - \mathbf{P}(X = 0) ,$$

where $\mathbf{P}(X = 1)$ and $\mathbf{P}(X = 0)$ can be carried out by the Poisson probability mass function

$$\mathbf{P}(X = x) = f(x) = \frac{\lambda^x}{x!} e^{-\lambda} .$$

Now

$$\mathbf{P}(X = 0) = \frac{0.5^0}{0!} \times e^{-0.5} = 0.6065$$

and

$$\mathbf{P}(X = 1) = \frac{0.5^1}{1!} \times e^{-0.5} = 0.3033$$

and so

$$\mathbf{P}(X \geq 2) = 1 - 0.9098 = 0.0902 .$$

Answer (Ex. 6.15) — 1.The Probability mass function for Poisson(λ) random variable X is

$$\mathbf{P}(X = x) = f(x; \lambda) = \frac{e^{-\lambda} \lambda^x}{x!}$$

where λ is the mean number of lacunae per specimen and X is the random variable “number of lacunae on a specimen”.

2.If $x = 0$ then $x! = 0! = 1$ and $\lambda^x = \lambda^0 = 1$, and the formula becomes $\mathbf{P}(X = 0) = e^{-\lambda}$.

3.Since $\mathbf{P}(X \geq 1) = 0.1$,

$$\mathbf{P}(X = 0) = 1 - \mathbf{P}(X \geq 1) = 0.9.$$

Using (b) and solving for λ gives:

$$e^{-\lambda} = 0.9 \quad \text{that is, } \lambda = -\ln(0.9) = 0.1 \text{ (approximately.)}$$

Hence

$$\mathbf{P}(X = 2) = \frac{e^{-0.1}(0.1)^2}{2!} = 0.45\% \text{ (approximately.)}$$

4.Occurrence of lacunae may not always be independent. For example, a machine malfunction may cause them to be clumped.

Answer (Ex. 6.16) — The probability that *one* light bulb doesn't need to be replaced in 1200 hours is:

$$\begin{aligned} \mathbf{P}(X > 1.2) &= 1 - \mathbf{P}(X < 1.2) \\ &= 1 - \int_1^{1.2} 6(0.25 - (x - 1.5)^2) dx \\ &= 1 - \int_1^{1.2} 6(0.25 - x^2 + 3x - 2.25) dx \\ &= 1 - \int_1^{1.2} (-6x^2 + 18x - 12) dx \\ &= 1 - [-2x^3 + 9x^2 - 12x]_1^{1.2} \\ &= 1 - 0.1040 \\ &= 0.8960 \end{aligned}$$

Assuming that the three light bulbs function independently of each other, the probability that none of them need to be replaced in the first 1200 hours is

$$\mathbf{P}(\{X_1 > 1.2\} \cap \{X_2 > 1.2\} \cap \{X_3 > 1.2\}) = 0.8960^3 = 0.7193$$

where X_i is the length of time that bulb i lasts.

Answer (Ex. 6.17) — 1.

$$\begin{aligned} \int_0^2 k e^{-x} dx &= 1 \\ [-k e^{-x}]_0^2 &= 1 \\ k(-e^{-2} + 1) &= 1 \\ k = \frac{1}{1 - e^{-2}} &\approx 1.1565 \end{aligned}$$

2.

$$\begin{aligned}\mathbf{P}(X \geq 1) &= 1 - \mathbf{P}(X < 1) \\&= 1 - \int_0^1 k e^{-x} dx \\&= 1 + k (e^{-x}]_0^1 \\&= 1 + \frac{e^{-1} - 1}{1 - e^{-2}} \\&\approx 0.2689\end{aligned}$$

Chapter 21

Appendix

21.1 Code

Labwork 245 (PDF and DF of a Normal(μ, σ^2) RV) Here are the functions to evaluate the PDF and DF of a Normal(μ, σ^2) RV X at a given x .

```
NormalPdf.m
function fx = NormalPdf(x,Mu,SigmaSq)
% Returns the Pdf of Normal(Mu, SigmaSq), at x,
% where Mu=mean and SigmaSq = Variance
%
% Usage: fx = NormalPdf(x,Mu,SigmaSq)
if SigmaSq <= 0
    error('Variance must be > 0')
    return
end

Den = ((x-Mu).^2)/(2*SigmaSq);
Fac = sqrt(2*pi)*sqrt(SigmaSq);

fx = (1/Fac)*exp(-Den);
```

```
NormalCdf.m
function Fx = NormalCdf(x,Mu,SigmaSq)
% Returns the Cdf of Normal(Mu, SigmaSq), at x,
% where Mu=mean and SigmaSq = Variance using
% MATLAB's error function erf
%
% Usage: Fx = NormalCdf(x,Mu,SigmaSq)
if SigmaSq <= 0
    error('Variance must be > 0')
    return
end

Arg2Erf = (x-Mu)/sqrt(SigmaSq*2);
Fx = 0.5*erf(Arg2Erf)+0.5;
```

Plots of the PDF and DF of several Normally distributed RVs depicted in Figure 6.9 were generated using the following script file:

```
PlotPdfCdfNormal.m
% PlotPdfCdfNormal.m script file
% Plot of some pdf's and cdf's of the Normal(mu,SigmaSq) RV X
```

```
%  
x=[-6:0.0001:6]; % points from the subset [-5,5] of the support of X  
subplot(1,2,1) % first plot of a 1 by 2 array of plots  
plot(x,NormalPdf(x,0,1),'r') % pdf of RV Z ~ Normal(0,1)  
hold % to superimpose plots  
plot(x,NormalPdf(x,0,1/10),'b') % pdf of RV X ~ Normal(0,1/10)  
plot(x,NormalPdf(x,0,1/100),'m') % pdf of RV X ~ Normal(0,1/100)  
plot(x,NormalPdf(x,-3,1),'r--') % pdf of RV Z ~ Normal(-3,1)  
plot(x,NormalPdf(x,-3,1/10),'b--') % pdf of RV X ~ Normal(-3,1/10)  
plot(x,NormalPdf(x,-3,1/100),'m--') % pdf of RV X ~ Normal(-3,1/100)  
 xlabel('x')  
 ylabel('f(x; \mu, \sigma^2)')  
 legend('f(x;0,1)', 'f(x;0,10^{-1})', 'f(x;0,10^{-2})', 'f(x;-3,1)', 'f(x;-3,10^{-1})', 'f(x;-3,10^{-2})')  
 subplot(1,2,2) % second plot of a 1 by 2 array of plots  
plot(x,NormalCdf(x,0,1),'r') % DF of RV Z ~ Normal(0,1)  
hold % to superimpose plots  
plot(x,NormalCdf(x,0,1/10),'b') % DF of RV X ~ Normal(0,1/10)  
plot(x,NormalCdf(x,0,1/100),'m') % DF of RV X ~ Normal(0,1/100)  
plot(x,NormalCdf(x,-3,1),'r--') % DF of RV Z ~ Normal(-3,1)  
plot(x,NormalCdf(x,-3,1/10),'b--') % DF of RV X ~ Normal(-3,1/10)  
plot(x,NormalCdf(x,-3,1/100),'m--') % DF of RV X ~ Normal(-3,1/100)  
 xlabel('x')  
 ylabel('F(x; \mu, \sigma^2)')  
 legend('F(x;0,1)', 'F(x;0,10^{-1})', 'F(x;0,10^{-2})', 'F(x;-3,1)', 'F(x;-3,10^{-1})', 'F(x;-3,10^{-2})')
```

Labwork 246 (PDF and DF of an Exponential(λ) RV X) Here are the functions to evaluate the PDF and DF of an Exponential(λ) RV X at a given x (point or a vector).

```
function fx = ExponentialPdf(x,Lambda) ExponentialPdf.m  
% Returns the Pdf of Exponential(Lambda) RV at x,  
% where Lambda = rate parameter  
  
% Usage: fx = ExponentialPdf(x,Lambda)  
if Lambda <= 0  
    error('Rate parameter Lambda must be > 0')  
    return  
end  
  
fx = Lambda * exp(-Lambda * x);
```

```
function Fx = ExponentialCdf(x,Lambda) ExponentialCdf.m  
% Returns the Cdf of Exponential(Lambda) RV at x,  
% where Lambda = rate parameter  
  
% Usage: Fx = ExponentialCdf(x,Lambda)  
if Lambda <= 0  
    error('Rate parameter Lambda must be > 0')  
    return  
end  
  
Fx = 1.0 - exp(-Lambda * x);
```

Plots of the PDF and DF of several Exponentially distributed RVs at four axes scales that are depicted in Figure 6.3 were generated using the following script file:

```
% PlotPdfCdfExponential.m script file  
% Plot of some pdf's and cdf's of the Exponential(Lambda) RV X  
%  
x=[0:0.0001:100]; % points from the subset [0,100] of the support of X
```

```

subplot(2,4,1) % first plot of a 1 by 2 array of plots
plot(x,ExponentialPdf(x,1),'r:','LineWidth',2) % pdf of RV X ~ Exponential(1)
hold on % to superimpose plots
plot(x,ExponentialPdf(x,10),'b--','LineWidth',2) % pdf of RV X ~ Exponential(10)
plot(x,ExponentialPdf(x,1/10),'m','LineWidth',2) % pdf of RV X ~ Exponential(1/10)
xlabel('x')
ylabel('f(x; \lambda)')
legend('f(x;1)', 'f(x;10)', 'f(x;10^{-1})')
axis square
axis([0,2,0,10])
title('Standard Cartesian Scale')
hold off

subplot(2,4,2)
semilogx(x,ExponentialPdf(x,1),'r:','LineWidth',2) % pdf of RV X ~ Exponential(1)
hold on % to superimpose plots
semilogx(x,ExponentialPdf(x,10),'b--','LineWidth',2) % pdf of RV X ~ Exponential(10)
semilogx(x,ExponentialPdf(x,1/10),'m','LineWidth',2) % pdf of RV X ~ Exponential(1/10)
% xlabel('x')
% ylabel('f(x; \lambda)')
% legend('f(x;1)', 'f(x;10)', 'f(x;10^{-1})')
axis square
axis([0,100,0,10])
title('semilog(x) Scale')
hold off

subplot(2,4,3)
semilogy(x,ExponentialPdf(x,1),'r:','LineWidth',2) % pdf of RV X ~ Exponential(1)
hold on % to superimpose plots
semilogy(x,ExponentialPdf(x,10),'b--','LineWidth',2) % pdf of RV X ~ Exponential(10)
semilogy(x,ExponentialPdf(x,1/10),'m','LineWidth',2) % pdf of RV X ~ Exponential(1/10)
% xlabel('x');
% ylabel('f(x; \lambda)');
% legend('f(x;1)', 'f(x;10)', 'f(x;10^{-1})')
axis square
axis([0,100,0,1000000])
title('semilog(y) Scale')
hold off

x=[ 0:0.001:1] [1.001:1:100000]; % points from the subset [0,100] of the support of X
subplot(2,4,4)
loglog(x,ExponentialPdf(x,1),'r:','LineWidth',2) % pdf of RV X ~ Exponential(1)
hold on % to superimpose plots
loglog(x,ExponentialPdf(x,10),'b--','LineWidth',2) % pdf of RV X ~ Exponential(10)
loglog(x,ExponentialPdf(x,1/10),'m','LineWidth',2) % pdf of RV X ~ Exponential(1/10)
% xlabel('x')
% ylabel('f(x; \lambda)')
% legend('f(x;1)', 'f(x;10)', 'f(x;10^{-1})')
axis square
axis([0,100000,0,1000000])
title('loglog Scale')
hold off

x=[0:0.0001:100]; % points from the subset [0,100] of the support of X
subplot(2,4,5) % second plot of a 1 by 2 array of plots
plot(x,ExponentialCdf(x,1),'r:','LineWidth',2) % cdf of RV X ~ Exponential(1)
hold on % to superimpose plots
plot(x,ExponentialCdf(x,10),'b--','LineWidth',2) % cdf of RV X ~ Exponential(10)
plot(x,ExponentialCdf(x,1/10),'m','LineWidth',2) % cdf of RV X ~ Exponential(1/10)
xlabel('x')
ylabel('F(x; \lambda)')
legend('F(x;1)', 'F(x;10)', 'F(x;10^{-1})')
axis square
axis([0,10,0,1])
hold off

```

```

subplot(2,4,6) % second plot of a 1 by 2 array of plots
semilogx(x,ExponentialCdf(x,1),'r:','LineWidth',2) % cdf of RV X ~ Exponential(1)
hold on % to superimpose plots
semilogx(x,ExponentialCdf(x,10),'b--','LineWidth',2) % cdf of RV X ~ Exponential(10)
semilogx(x,ExponentialCdf(x,1/10),'m','LineWidth',2) % cdf of RV X ~ Exponential(1/10)
%xlabel('x')
%ylabel('F(x; \lambda)')
%legend('F(x;1)', 'F(x;10)', 'F(x;10^{-1})')
axis square
axis([0,100,0,1])
%title('semilog(x) Scale')
hold off

subplot(2,4,7)
semilogy(x,ExponentialCdf(x,1),'r:','LineWidth',2) % cdf of RV X ~ Exponential(1)
hold on % to superimpose plots
semilogy(x,ExponentialCdf(x,10),'b--','LineWidth',2) % cdf of RV X ~ Exponential(10)
semilogy(x,ExponentialCdf(x,1/10),'m','LineWidth',2) % cdf of RV X ~ Exponential(1/10)
%xlabel('x');
%ylabel('F(x; \lambda)');
%legend('F(x;1)', 'F(x;10)', 'F(x;10^{-1})')
axis square
axis([0,10,0,1])
%title('semilog(y) Scale')
hold off

x=[ 0:0.001:1] [1.001:1:100000]; % points from the subset of the support of X
subplot(2,4,8)
loglog(x,ExponentialCdf(x,1),'r:','LineWidth',2) % cdf of RV X ~ Exponential(1)
hold on % to superimpose plots
loglog(x,ExponentialCdf(x,10),'b--','LineWidth',2) % cdf of RV X ~ Exponential(10)
loglog(x,ExponentialCdf(x,1/10),'m','LineWidth',2) % cdf of RV X ~ Exponential(1/10)
%xlabel('x')
%ylabel('F(x; \lambda)')
%legend('F(x;1)', 'F(x;10)', 'F(x;10^{-1})')
axis square
axis([0,100000,0,1])
%title('loglog Scale')
hold off

```

Labwork 247 (Plotting the empirical DF) A MATLAB function to plot the empirical DF (5.8) of n user-specified samples efficiently for massive number of samples. Read the following M-file for the algorithm:

ECDF.m

```

function [x1 y1] = ECDF(x, PlotFlag, LoxD, HixD)
% return the x1 and y1 values of empirical CDF
% based on samples in array x of RV X
% plot empirical CDF if PlotFlag is >= 1
%
% Call Syntax: [x1 y1] = ECDF(x, PlotFlag, LoxD,HixD);
% Input      : x = samples from a RV X (a vector),
%               PlotFlag is a number controlling plot (Y/N, marker-size)
%               LoxD is a number by which the x-axis plot range is extended to the left
%               HixD is a number by which the x-axis plot range is extended to the right
% Output     : [x1 y1] & empirical CDF Plot IF PlotFlag >= 1
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R=length(x);           % assume x is a vector and R = Number of samples in x
x1=zeros(1,R+2);
y1=zeros(1,R+2);       % initialize y to null vectors
for i=1:1:R            % loop to append to x and y axis values of plot

```

```

y1(i+1)=i/R; % append equi-increments of 1/R to y
end % end of for loop
x1(2:R+1)=sort(x); % sorting the sample values
x1(1)=x1(2)-LoxD; x1(R+2)=x1(R+1)+HixD; % padding x for emp CDF to start at min(x) and end at max(x)
y1(1)=0; y1(R+2)=1; % padding y so emp CDF start at y=0 and end at y=1

% to make a ECDF plot for large number of points set the PlotFlag<1 and use
% MATLAB's plot function on the x and y values returned by ECDF -- stairs(x,y)
if PlotFlag >= 1 % Plot customized empirical CDF if PlotFlag >= 1
    %newplot;
    MSz=10/PlotFlag; % set Markersize MSz for dots and circles in ECDF plot
    % When PlotFlag is large MSz is small and the
    % Markers effectively disappear in the ecdf plot
    R=length(x1); % update R = Number of samples in x
    hold on % hold plot for superimposing plots

    for i=1:1:R-1
        if(i>1 && i ~= R-1)
            plot([x1(i),x1(i+1)], [y1(i),y1(i)], 'k o -', 'MarkerSize',MSz)
        end
        if (i< R-1)
            plot(x1(i+1),y1(i+1), 'k .', 'MarkerSize', 2.5*MSz)
        end
        plot([x1(i),x1(i+1)], [y1(i),y1(i)], 'k -')
        plot([x1(i+1),x1(i+1)], [y1(i),y1(i+1)], 'k -')
    end

    hold off;
end

```

Ideally, this function needs to be rewritten using primitives such as MATLAB's `line` commands.

Labwork 248 (q-th sample quantile) Let us implement Algorithm 2 as the following MATLAB function:

```

qthSampleQuantile.m
function qthSQ = qthSampleQuantile(q, SortedXs)
%
% return the q-th Sample Quantile from Sorted array of Xs
%
% Call Syntax: qthSQ = qthSampleQuantile(q, SortedXs);
%
% Input      : q = quantile of interest, NOTE: 0 <= q <= 1
%               SortedXs = sorted real data points in ascending order
% Output     : q-th Sample Quantile, ie, inverse ECDF evaluated at q

% store the length of the sorted data array SortedXs in n
N = length(SortedXs);
Nminus1TimesQ = (N-1)*q; % store (N-1)*q in a variable
Index = floor(Nminus1TimesQ); % store its floor in a C-style Index variable
Delta = Nminus1TimesQ - Index;
if Index == N-1
    qthSQ = SortedXs(Index+1);
else
    qthSQ = (1.0-Delta)*SortedXs(Index+1) + Delta*SortedXs(Index+2);
end

```

Labwork 249 (Loading) Let us save all the steps done in Labwork 55 into the following script M-file:

```

%% Load the data from the comma delimited text file 'NZ20110222earthquakes.csv' with
%% the following column IDs
%% CUSP_ID,LAT,LONG,NZMGE,NZMGN,ORI_YEAR,ORI_MONTH,ORI_DAY,ORI_HOUR,ORI_MINUTE,ORI_SECOND,MAG,DEPTH
%% Using MATLAB's dlmread command we can assign the data as a matrix to EQ;
%% note that the option 1,0 to dlmread skips first row of column descriptors
%
% the variable EQall is about to be assigned the data as a matrix
EQall = dlmread('NZ20110222earthquakes.csv', ',', 1, 0);
size(EQall) % report the dimensions or size of the matrix EQall
%ans = 145 14

EQall(any(isnan(EQall),2),:) = []; %Remove any rows containing NaNs from the matrix EQall
% report the size of EQall and see if it is different from before we removed and NaN containing rows
size(EQall)
% output: ans = 145 14
% remove locations outside Chch and assign it to a new variable called EQ
% only keep earthquake hypocenter locations inside Chch
% only keep earthquakes with magnitude >3
EQ = EQall(-43.75<EQall(:,2) & EQall(:,2)<-43.45 & 172.45<EQall(:,3) ...
& EQall(:,3)<172.9 & EQall(:,12)>3, :);
% now report the size of the earthquakes in Christchurch in variable EQ
size(EQ)
% output: ans = 124 14

% assign the four variables of interest
LatData=EQ(:,2); LonData=EQ(:,3); MagData=EQ(:,12); DepData=EQ(:,13);

% finally make a plot matrix of these 124 4-tuples as red points
plotmatrix([LatData,LonData,MagData,DepData], 'r.');

```

Labwork 250 (Importance Resampler Demo) Visualisation of the Importance Resampler of Algorithm 10 by producing approximate samples from Cauchy using samples from $\text{Normal}(0, 1)$.

```

% sir_cauchy_normal.m
ImpResamplerCauchyViaNormal.m

clear all

n = 1; % number of sample points required
m = 10; % number of initial sample points

nseed = 13;
randn('state',nseed), rand('state',nseed)

y = randn(1,m);
y2 = y.*y;
w = exp(.5 * y2) ./ (1 + y2);
w = w / sum(w);
x = randsample(y,n,true,w);

s = [-4:.01:4];
s2 = s .* s;
f = (1 ./ (1 + s2)) / pi;
g = exp(-.5 * s2) / sqrt(2*pi);
plot(s,f,'-r',s,g,'-b')
legend('f=Cauchy(0,1)', 'g=N(0,1)')
title('Sampling/importance resampling: Generating Cauchy(0,1) using N(0,1)')
hold on, pause
plot(x(1),0,'.b',x(1),0,'ob')
plot([x(1)-.02 x(1)-.02],[.005 tpdf(x(1)-.02,1)],'-r'), text(-1.7,.05,'f(y)')
plot([x(1)+.02 x(1)+.02],[.005 normpdf(x(1)+.02,0,1)],'-b'), text(-1.3,.05,'g(y)')
text(-1.2,.03,'w prop. to f(y)/g(y)'), pause, hold off
plot(s,f,'-r',s,g,'-b')
legend('f=Cauchy(0,1)', 'g=N(0,1)')

```

```

title('Sampling/importance resampling: Generating Cauchy(0,1) using N(0,1)')
hold on
plot(y,zeros(1,m),'.b',y,zeros(1,m),'ob'), pause
lplot2(y,w,'v','g'), hold on, pause
plot(x,0,'r',x,0,'or'), pause
plot(x,0,'b',x,0,'ob')
x2 = randsample(y,n,true,w);
plot(x2,0,'r',x2,0,'or'), pause

n = 1000; % number of sample points required
m = 10000; % number of initial sample points

nseed = 23;
randn('state',nseed), rand('state',nseed)

y = randn(1,m);
y2 = y.*y;
w = exp(.5 * y2) ./ (1 + y2);
w = w / sum(w);
x = randsample(y,n,true,w);

hold off
plot(s,f,'-r',s,g,'-b')
legend('f=Cauchy(0,1)', 'g=N(0,1)')
title('Sampling/importance resampling: Generating Cauchy(0,1) using N(0,1)')
hold on
plot(y,zeros(1,m),'.b')
lplot2(y,w,'v','g'), pause
plot(s,f,'-r',s,g,'-b')
legend('f=Cauchy(0,1)', 'g=N(0,1)')
title('Sampling/importance resampling: Generating Cauchy(0,1) using N(0,1)')
hold on
plot(x,zeros(1,n),'r'), pause
histogram(x,1,[ -inf inf ],'r');

```

Labwork 251 (Levy density plot) Figure 13.1 was made with the following script file.

```

LevyDensityPlot.m
x=linspace(-9,9,1500);y=x;
[X, Y]=meshgrid(x,y);
Z1 = (cos((0*X)+1) + 2*cos((1*X)+2) + 3*cos((2*X)+3) + 4*cos((3*X)+4) + 5*cos((4*X)+5));
Z2 = (cos((2*Y)+1) + 2*cos((3*Y)+2) + 3*cos((4*Y)+3) + 4*cos((5*Y)+4) + 5*cos((6*Y)+5));
Temp=50;
Z = exp(-(Z1 .* Z2 + (X + 1.42513) .^2 + (Y + 0.80032) .^ 2)/Temp);
mesh(X,Y,Z)
caxis([0, 10]);
rotate3d on

```

Labwork 252 (Negative of the Levy density) The negative of the Levy density (13.1) is encoded in the following M-file as a function to be passed to MATLAB's `fminsearch`.

```

NegLevyDensity.m
function NegLevyFunVal = NegLevyDensity(parameters);
X=parameters(1); Y=parameters(2); Temp=50.0;
Z1 = (cos((0*X)+1) + 2*cos((1*X)+2) + 3*cos((2*X)+3) + 4*cos((3*X)+4) + 5*cos((4*X)+5));
Z2 = (cos((2*Y)+1) + 2*cos((3*Y)+2) + 3*cos((4*Y)+3) + 4*cos((5*Y)+4) + 5*cos((6*Y)+5));
NegLevyFunVal = -exp(-(Z1 .* Z2 + (X + 1.42513) .^2 + (Y + 0.80032) .^ 2)/Temp);

```

Labwork 253 (Log-likelihood of Lognormal) Figure 13.2 was made with the following script file.

```
----- LogNormalLogLklPlot.m -----
% Plots the log likelihood of LogNormal(lambda, zeta),
% for observed data vector x IIDLogNormal(lambda,zeta),
rand('twister',001);
x=exp(arrayfun(@(u)(Sample1NormalByNewRap(u,10.36,0.26^2)),rand(1,100)));
% log likelihood function
lambda=linspace(5,15.0,200);
zeta=linspace(0.1, 2,200);
[LAMBDA, ZETA]=meshgrid(lambda,zeta);
LAMBDA3=repmat(LAMBDA,[1 1 length(x)]);
ZETA3=repmat(ZETA,[1 1 length(x)]);

xx=zeros([1 1 length(x)]);xx(:)=x;
x3=repmat(xx,[length(lambda) length(zeta) 1]);
%l = -sum(log((1 ./ (sqrt(2*pi)*zeta) .* x) .* exp((-1/(2*zeta.^2))*(log(x)-lambda).^2)));
LOGLKL = sum(log((1 ./ (sqrt(2*pi)*ZETA3) .* x3) .* exp((-1/(2*ZETA3.^2)).*(log(x3)-LAMBDA3).^2)),3);
LOGLKL(LOGLKL<0)=NaN;

caxis([0 0.1]*10^3);colorbar
axis([0 15 0 2 0 0.1*10^3])
clf; meshc(LAMBDA, ZETA, LOGLKL);
rotate3d on;
```

Labwork 254 (Consistency of MLE in Bernoulli experiment) Figure 11.3 was made with the following script file.

```
----- BernoulliMLEConsistency.m -----
clf;%clear any figures
rand('twister',736343); % initialize the Uniform(0,1) Sampler
N = 3; % 10^N is the maximum number of samples from RV
J = 100; % number of Replications for each n
u = rand(J,10^N); % generate 10X10^N samples from Uniform(0,1) RV U
p=0.5; % set p for the Bernoulli(p) trials
PS=[0:0.001:1]; % sample some values for p on [0,1] to plot likelihood
for i=1:N
    if(i==1) Pmin=0.; Pmax=1.0; Ymin=-70; Ymax=-10; Y=linspace(Ymin,Ymax,J); end
    if(i==2) Pmin=0.; Pmax=1.0; Ymin=-550; Ymax=-75; Y=linspace(Ymin,Ymax,J); end
    if(i==3) Pmin=0.3; Pmax=0.8; Ymin=-900; Ymax=-700; Y=linspace(Ymin,Ymax,J); end
    n=10^i;% n= sample size, ie, number of Bernoulli trials
    subplot(1,N,i)
    if(i==1) axis([Pmin Pmax Ymin -2]); end
    if(i==2) axis([Pmin Pmax Ymin -60]); end
    if(i==3) axis([Pmin Pmax Ymin -685]); end
    EmpCovSEhat=0; % track empirical coverage for SEhat
    EmpCovSE=0; % track empirical coverage for exact SE
    for j=1:J
        % transform the Uniform(0,1) samples to n Bernoulli(p) samples
        x=floor(u(j,1:n)+p);
        s = sum(x); % statistic s is the sum of x_i's
        % display the outcomes and their sum
        %display(x)
        %display(s)
        MLE=s/n; % Analytically MLE is s/n
        se = sqrt((1-p)*p/n); % standard error from known p
        sehat = sqrt((1-MLE)*MLE/n); % estimated standard error from MLE p
        ZalphaBy2 = 1.96; % for 95% CI
        if(abs(MLE-p)<=2*sehat) EmpCovSEhat=EmpCovSEhat+1; end
        line([MLE-2*sehat MLE+2*sehat],[Y(j) Y(j)],'Marker','+', 'LineStyle',':', 'LineWidth',1,'Color',[1 .0 .0])
        if(abs(MLE-p)<=2*se) EmpCovSE=EmpCovSE+1; end
        line([MLE-2*se MLE+2*se],[Y(j) Y(j)],'Marker','+', 'LineStyle','-' )
        % l is the Log Likelihood of data x as a function of parameter p
        l=@(p)sum(log(p ^ s * (1-p)^(n-s)));
        hold on;
        % plot the Log Likelihood function and MLE
```

```

semilogx(PS,arrayfun(l,PS),'m','LineWidth',1);
hold on; plot([MLE],[Y(j)],'.','Color','c'); % plot MLE
end
hold on;
line([p p], [Ymin, 1(p)],'LineStyle',':','Marker','none','Color','k','LineWidth',2)
%axis([-0.1 1.1]);
%axis square;
LabelString=['n=' num2str(n) ' ' Cvrg.= ' num2str(EmpCovSE) '/ num2str(J) ...
' ~= ' num2str(EmpCovSEhat) '/ num2str(J)];
%text(0.75,0.05,LabelString)
title(LabelString)
hold off;
end

```

Labwork 255 (Gilvenko-Cantelli Lemma for Uniform(0, 1)) The following script was used to generate the Figure 14.1.

```

----- GilvenkoCantelliUnif01.m -----
% from Uniform(0,1) RV
rand('twister',76534); % initialize the Uniform(0,1) Sampler
N = 3; % 10^N is the maximum number of samples from Uniform(0,1) RV
u = rand(10,10^N); % generate 10 X 10^N samples from Uniform(0,1) RV U
x=[0:0.001:1];
% plot the ECDF from the first 10 samples using the function ECDF
for i=1:N
    SampleSize=10^i;
    subplot(1,N,i)
    plot(x,x,'r','LineWidth',2); % plot the DF of Uniform(0,1) RV in red
    % Get the x and y coordinates of SampleSize-based ECDF in x1 and y1 and
    % plot the ECDF using the function ECDF
    for j=1:10

        hold on;
        if (i==1) [x1 y1] = ECDF(u(j,1:SampleSize),2.5,0.2,0.2);
        else
            [x1 y1] = ECDF(u(j,1:SampleSize),0,0.1,0.1);
            stairs(x1,y1,'k');
        end
    end
    % Note PlotFlag is 1 and the plot range of x-axis is
    % incremented by 0.1 or 0.2 on either side due to last 2 parameters to ECDF
    % being 0.1 or 0.2
    % Alpha=0.05; % set alpha to 5% for instance
    % Epsn = sqrt((1/(2*SampleSize))*log(2/Alpha)); % epsilon_n for the confidence band
    hold on;
    stairs(x1,max(y1-Epsn,zeros(1,length(y1))), 'g'); % lower band plot
    stairs(x1,min(y1+Epsn,ones(1,length(y1))), 'g'); % upper band plot
    axis([-0.1 1.1 -0.1 1.1]);
    %axis square;
    LabelString=['n=' num2str(SampleSize)];
    text(0.75,0.05,LabelString)
    hold off;
end

```

21.2 Data

Here we describe some of the data sets we analyze.

Data 256 (Our Maths & Stats Dept. Web Logs) We assume access to a Unix terminal (Linux, Mac OS X, Sun Solaris, etc). We show how to get your hands dirty with web logs that track

among others, every IP address and its time of login to our department web server over the world-wide-web. The raw text files of web logs may be manipulated but they are typically huge files and need some Unix command-line utilities.

```
rsa64@mathopt03:~> cd October010203WebLogs/
rsa64@mathopt03:~/October010203WebLogs> ls -al
-rw-r--r--+ 1 rsa64 math 7527169 2007-10-04 09:38 access-07_log.2
-rw-r--r--+ 1 rsa64 math 7727745 2007-10-04 09:38 access-07_log.3
```

The files are quite large over 7.5 MB each. So we need to compress it. We use the **gzip** and **gunzip** utility in any Unix environment to compress and decompress these large text files of web logs. After compression the file sizes are more reasonable.

```
rsa64@mathopt03:~/October010203WebLogs> gzip access-07_log.3
rsa64@mathopt03:~/October010203WebLogs> gzip access-07_log.2
rsa64@mathopt03:~/October010203WebLogs> zcat access-07_log.2.gz | grep ' 200 '
| awk '{ print \$4}' | sed -e 's/\\([0-9]\\{2\\}\\)\\(([a-Z]\\{3\\}\\)\\(([0-9]\\{4\\}\\
:\\([0-9]\\{2\\}\\):\\([0-9]\\{2\\}\\):\\([0-9]\\{2\\}\\)/\\3 10 \\1 \\4 \\5 \\6/'
2007 10 02 03 57 48
2007 10 02 03 58 31
.
.
.
2007 10 03 03 56 21
2007 10 03 03 56 52
```

Finally, there are 56485 and 53966 logins for the two 24-hour cycles, starting 01/Oct and 01/Oct, respectively. We can easily get these counts by further piping the previous output into the line counting utility **wc** with the **-l** option. All the Unix command-line tools mentioned earlier can be learned by typing **man** followed by the tool-name, for eg. type **man sed** to learn about the usage of **sed** at a Unix command shell. We further pipe the output of login times for the two 24-hour cycles starting 01/Oct and 02/Oct in format YYYY MM DD HH MM SS to **| sed -e 's/2007 10 //'** > **WebLogTimes20071001035730.dat** and ... > **WebLogTimes20071002035730.dat**, respectively to strip away the redundant information on YYYY MM , namely 2007 10 , and only save the relevant information of DD HH MM SS in files named **WebLogTimes20071001035730.dat** and **WebLogTimes20071002035730.dat**, respectively. These two files have the data of interest to us. Note that the size of these two uncompressed final data files in plain text are smaller than the compressed raw web log files we started out from.

```
rsa64@mathopt03:~/October010203WebLogs> ls -al
-rw-r--r--+ 1 rsa64 math 677820 2007-10-05 15:36 WebLogTimes20071001035730.dat
-rw-r--r--+ 1 rsa64 math 647592 2007-10-05 15:36 WebLogTimes20071002035730.dat
-rw-r--r--+ 1 rsa64 math 657913 2007-10-04 09:38 access-07_log.2.gz
-rw-r--r--+ 1 rsa64 math 700320 2007-10-04 09:38 access-07_log.3.gz
```

Now that we have been familiarized with the data of login times to our web-server over 2 24-hour cycles, let us do some statistics. The log files and basic scripts are courtesy of the Department's computer systems administrators Paul Brouwers and Steve Gourdie. This data processing activity was shared in such detail to show you that statistics is only meaningful when the data and the process that generated it are clear to the experimenter. Let us process the data and visualize the empirical distribution functions using the following script:

WebLogDataProc.m

```

load WebLogTimes20071001035730.dat % read data from first file
% multiply day (October 1) by 24*60*60 seconds, hour by 60*60 seconds,
% minute by 60 seconds and seconds by 1, to rescale time in units of seconds
SecondsScale1 = [24*60*60; 60*60; 60; 1];
StartTime1 = [1 3 57 30] * SecondsScale1; % find start time in seconds scale
%now convert time in Day/Hours/Minutes/Seconds format to seconds scale from
%the start time
WebLogSeconds20071001035730 = WebLogTimes20071001035730 * SecondsScale1 - StartTime1;

% repeat the data entry process above on the second file
load WebLogTimes20071002035730.dat %
SecondsScale1 = [24*60*60; 60*60; 60; 1];
StartTime2 = [2 3 57 30] * SecondsScale1;
WebLogSeconds20071002035730 = WebLogTimes20071002035730 * SecondsScale1 - StartTime2;

% calling a more efficient ECDF function for empirical DF's
[x1 y1]=ECDF(WebLogSeconds20071001035730,0,0,0);
[x2 y2]=ECDF(WebLogSeconds20071002035730,0,0,0);
stairs(x1,y1,'r','linewidth',1) % draw the empirical DF for first dataset
hold on;
stairs(x2,y2,'b') % draw empirical cdf for second dataset

% set plot labels and legends and title
xlabel('time t in seconds')
ylabel('ECDF F^(t)')
grid on
legend('Starting 10\01\0357\30', 'Starting 10\02\0357\30')
title('24-Hour Web Log Times of Maths & Stats Dept. Server at Univ. of Canterbury, NZ')

%To draw the confidence bands
Alpha=0.05; % set alpha
% compute epsilon_n for first dataset of size 56485
Epsn1 = sqrt((1/(2*56485))*log(2/Alpha));
stairs(x1,max(y1-Epsn1,zeros(1,length(y1))), 'g') % lower 1-alpha confidence band
stairs(x1,min(y1+Epsn1,ones(1,length(y1))), 'g') % upper 1-alpha confidence band

% compute epsilon_n for second dataset of size 53966
Epsn2 = sqrt((1/(2*53966))*log(2/Alpha));
stairs(x2,max(y2-Epsn2,zeros(1,length(y2))), 'g') % lower 1-alpha confidence band
stairs(x2,min(y2+Epsn2,ones(1,length(y2))), 'g') % upper 1-alpha confidence band

```

Chapter 22

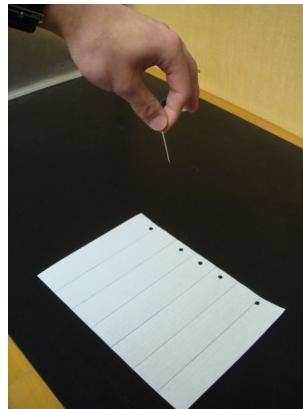
Student Projects

Please visit: <http://www.math.canterbury.ac.nz/~r.sainudiin/courses/STAT218/projects/Stat218StudentProjects2007.pdf>
and also visit: <http://www.math.canterbury.ac.nz/~r.sainudiin/courses/STAT218/projects/Stat218StudentProjects2008.pdf>

to see the term projects completed by students of STAT 218 from 2007 and 2008. Some of the simulation devices and non-perishable data from these projects are archived on the sixth floor of the Erskine Building.

22.1 Testing the Approximation of π by Buffon's Needle Test

Amanda Hughes
and Sung-Lim Suh



Abstract

This project is designed to investigate Buffon's Needle experiment. We replicated the concept of Buffon's Needle and tested the null hypothesis that the outcomes from tossing a needle are directly related to π . The Delta Method and non-parametric methods have been employed in this project.

22.1.1 Introduction & Motivation

The report will firstly cover the background and motivation of this project. Next we will explain the geometric background to why this experiment approximates π and we will look at the statistical methodologies used. Following this we will discuss our results and conclusion. Finally we will discuss potential modifications.

Background - Comte de Buffon

Comte de Buffon was born September 7 1707 in Montbard, France. He studied law in Dijon and medicine in Angers. After his studies, he had a chance to tour around France, Italy, and England to explore his knowledge in science. When he returned to France, Buffon published translations of one of Isaac Newton's works and his interest in science was now clear.

Background - Buffon's Needle

Buffon's Needle Problem was first stated by Comte de Buffon in 1733, the solution was published later in 1777. The problem involves finding the probability that a needle of length l will land on a line, given a floor with equally spaced parallel lines (or floorboards) a distance d apart.

Motivation

The motivation behind this project is to reconstruct Buffon's Needle Experiment. We wanted to see if an approximation of π was found by this somewhat simple experiment over 200 years ago.

22.1.2 Materials & Methods

Materials

We first constructed a board which had several parallel lines. We did this by ruling lines on a piece of A4 paper. The width between the lines was either the same length as the needle or smaller than the length of the needle or larger than the length of the needle. The needle we used was a sewing needle of length 37mm. Instead of changing the needle to a smaller or larger needle for the three trials we ruled up three different sheets of A4 paper, one for each of the situations. The sheet that had the lines the same distance apart as the length of the needle had lines spaced 37mm apart. The sheet that had the lines closer together than the length of the needle had lines spaced 35mm apart. The sheet that had the lines further apart than the length of the needle had lines spaced 42mm apart. We recorded the data by entering it into Excel as the experiment was taking place. Sung-Lim tossed the needle and Amanda recorded the data.

Method

We tossed a needle 200 times for each of the three different distances apart of the lines. Sung-Lim tossed the needle for all trials so that the tossing could be done as identically as possible. Sung-Lim held the needle at the same height and dropped it in exactly the same way for each trial. Sung-Lim called out each toss as either “Yes” for the needle landing on a line or “No” for the needle not landing on a line. Any decisions that had to be made over whether the needle crossed the line or not were made by Sung-Lim. If the needle rolled or bounced off the page we disregarded that trial. A break was taken every 100 trials so that all trials could be done as identically as possible.

Geometric Aspect

We need to look at how this experiment approximates π . Let us begin by looking at the general case, where the needle is the same length as the distance between the floorboards.

Imagine a floor on which there are an infinite number of parallel lines (floorboards) spaced two units apart. You toss a needle that is also two units in length onto the floor. We want to find the probability that the needle crosses a line in the floorboards. l =the length of the needle and d =the distance between the lines.

Take one board to examine. Let x be the shortest distance from the mid point of the needle to the nearest line. Let θ be the angle between x and the needle. Imagine that there is a vertical line down from the end of the needle closest to the line. $\cos(\theta)$ gives the distance from the midpoint of the line to this imaginary line.

As you can imagine, the needle can fall in an infinite amount of ways and positions relative to the parallel lines. Therefore, θ also has an infinite amount of possibilities. We will limit these possibilities in two ways. Firstly only consider the line that is closest to the midpoint of the needle. Secondly we will divide the board into two by drawing an imaginary line down the board halfway between two lines and only consider needle positions on the right side of our halfway line. We will consider the needles whose middle falls to the left of this imaginary line to be a rotation of the right side of the imaginary line. So we now only have two general situations to look at. One of these situations is shown in the above picture, where θ is less than $\pi/2$ radians. The other situation is where θ is larger than $\pi/2$ radians. In this case we will consider the angle θ to actually be the

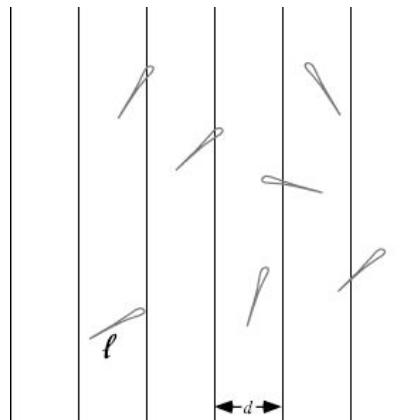


Figure 22.1: Example of Needle Tosses

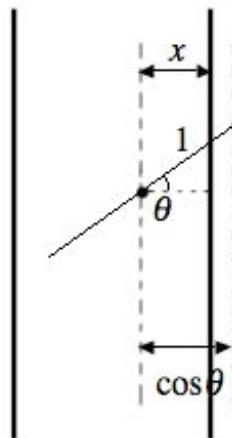


Figure 22.2: Explaining the outcomes of the needle toss

angle below of x and we will think of this angle as negative. Therefore, the support of x is 0 to 1 and the support of θ is $-\pi/2$ to $\pi/2$. This is shown in figure 22.3.

The shaded area in figure 22.3, represents when the needle crosses a line. This happens when $x < \cos(\theta)$. The total outcome space is π and the area under $\cos(\theta)$ from $-\pi/2$ to $\pi/2$ is 2 (integrating $\cos(\theta)$ from $-\pi/2$ to $\pi/2$). Therefore, the chance of a needle falling on a line is $2/\pi$.

For the case where the needle is shorter in length than the distance between the floorboards we need to modify the above slightly. We need to account for the exact length difference between the needle and the floorboards. This is added to the above explanation by multiplying $2/\pi$ by y , which is the ratio of length of needle and distance between the lines, ie. $y = l/d$. Therefore the chance of

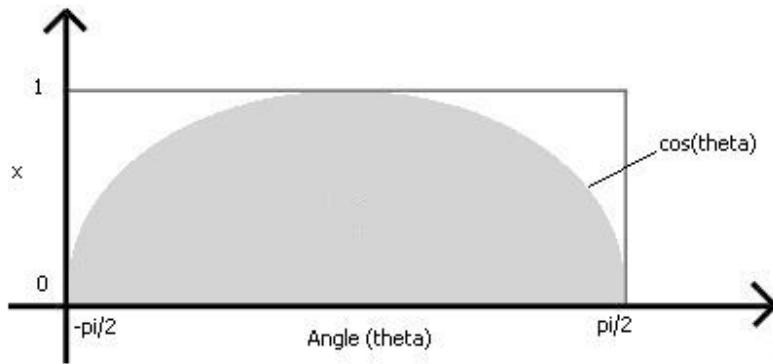


Figure 22.3: Outcome Space of Buffon's Needle Experiment for General Case

a shorter needle landing on a line is $2y/\pi$.

$$\Pr(\text{needle crosses line}) = \int_0^{2\pi} \left(\frac{l|\cos \theta|}{d} \right) \frac{d\theta}{2\pi}$$

$$\Pr(\text{needle crosses line}) = \frac{2l}{d\pi} \int_0^{\frac{\pi}{2}} \cos \theta d\theta$$

$$\Pr(\text{needle crosses line}) = \frac{2l}{d\pi}$$

$$\Pr(\text{needle crosses line}) = \frac{2y}{\pi}$$

For the case where the needle is longer in length than the distance between the floorboards we get a more complex outcome. We again need to account for the exact length difference between the needle and the floorboards. Therefore, the chance of a longer needle landing on a line is:

$$\Pr(\text{needle crosses line}) = \frac{2}{\pi} \left(y - \sqrt{y^2 - 1} + \sec^{-1} y \right)$$

Statistical Methodology

For this experiment we looked at the three different distances apart of the lines. For each of the three circumstances we used the same hypotheses.

H0(null hypothesis): $\varphi^* = \frac{2}{\pi}$, the outcomes of tossing a needle are directly related to π

H1(alternative hypothesis): $\varphi^* \neq \frac{2}{\pi}$, the outcomes of tossing a needle are not directly related to π

For the general case:

$$(\Theta, X) \stackrel{IID}{\sim} \text{Uniform}([\frac{\pi}{2}, \frac{\pi}{2}] \times [0, 1])$$

as shown in figure 1.3. We know that for our simplest situation, where the length of the needle is the same as the distance between the lines:

$$\varphi^* = \Pr((\Theta, X) \in \text{Shaded Region of figure 22.3}) = \frac{2}{\pi}$$

$$N_1, N_2, \dots, N_{200} \stackrel{IID}{\sim} \text{Bernoulli}(\varphi^*)$$

This is the probability of a needle landing on a line. The trials for each of the three different distances apart of the lines are 200 independent and identically distributed Bernoulli trials.

Our maximum likelihood estimator of φ^* is:

$$\hat{\varphi}_{200} = \frac{n_1 + n_2 + \dots + n_{200}}{200}$$

This is the sample mean. But what we really want is a function of φ^* , namely, $\Psi(\Phi) := 2/\Phi$. We now need the Delta Method. The Delta Method gives us the needed correction to transform an estimate and its confidence interval. By the Delta Method:

$$\pi \approx \psi^* = g(\varphi^*) = \frac{2}{\varphi^*}$$

and the maximum likelihood estimate of ψ^* is:

$$\hat{\psi}_{200} = g(\hat{\varphi}_{200}) = \frac{2}{\hat{\varphi}_{200}}$$

Next we can calculate the standard error of our estimator of ψ^* , namely, $\hat{\Psi}_n = g(\hat{\Phi}_n)$, and subsequently confidence intervals:

$$\begin{aligned} se(\hat{\Psi}_n) &= |g'(\varphi)|se(\hat{\Phi}_n) \\ se(\hat{\Psi}_n) &= |\frac{d}{d\varphi}g(\varphi)|se(\hat{\Phi}_n) \\ se(\hat{\Psi}_n) &= |\frac{d}{d\varphi}(2\varphi^{-2})|se(\hat{\Phi}_n) \\ se(\hat{\Psi}_n) &= |-2\varphi^{-2}|se(\hat{\Phi}_n) \\ se(\hat{\Psi}_n) &= (2\varphi^{-2})se(\hat{\Phi}_n) \end{aligned}$$

where the estimated standard error is:

$$se(\hat{\psi}_{200}) = (2\varphi_{200}^{-2})se(\hat{\varphi}_{200}) = (2\varphi_{200}^{-2})\frac{\hat{\varphi}_{200}(1-\hat{\varphi}_{200})}{200}$$

Now we can calculate the 95% confidence interval for $\psi^* \approx \pi$:

$$\hat{\psi}_{200} \pm 1.96se(\hat{\psi}_{200})$$

Now for the needle shorter in length than the distance between the lines, where $\varphi^* = \frac{2y}{\pi}$ and therefore by the Delta Method: $\pi \approx \frac{2y}{\varphi^*}$, $\hat{\varphi}_{200}$ is the sample mean of this set of data.

$$\psi^* = g(\varphi^*) = \frac{2y}{\varphi^*}$$

$$\hat{\Psi}_{200} = g(\hat{\varphi}_{200}) = \frac{2y}{\hat{\varphi}_{200}}$$

From this we can calculate the standard error of the estimator of ψ^* , namely, $\hat{\Psi}_n$, and subsequently confidence intervals as follows:

$$\begin{aligned}
se(\hat{\Psi}_n) &= |g'(\varphi)| * se(\hat{\Phi}_n) \\
se(\hat{\Psi}_n) &= \left| \frac{d}{d\varphi} g(\varphi) \right| * se(\hat{\Phi}_n) \\
se(\hat{\Psi}_n) &= \left| \frac{d}{d\varphi} (2y\varphi^{-1}) \right| * se(\hat{\Phi}_n) \\
se(\hat{\Psi}_n) &= |-2y\varphi^{-2}| * se(\hat{\Phi}_n) \\
se(\hat{\Psi}_n) &= (2y\varphi^{-2}) * se(\hat{\Phi}_n)
\end{aligned}$$

where the estimated standard error is

$$se(\hat{\psi}_n) = (2y\hat{\varphi}_{200}^{-2}) * se(\hat{\Phi}_n) \quad \text{and} \quad se(\hat{\Phi}_n) = \frac{\hat{\varphi}_{200}(1-\hat{\varphi}_{200})}{200}$$

Now we can calculate the 95% confidence interval for $\psi^* \approx \pi$:

$$\hat{\psi}_{200} \pm 1.96 se(\hat{\psi}_{200})$$

Now for the needle longer in length than the distance between the lines, where $\varphi^* = \frac{2}{\pi}(y - \sqrt{y^2 - 1} + \sec^{-1}y)$ and therefore by the Delta Method: $\pi \approx \frac{2}{\varphi^*}(y - \sqrt{y^2 - 1} + \sec^{-1}y)$, $\hat{\varphi}_{200}$ is the sample mean of this set of data.

$$\begin{aligned}
\psi^* &= g(\varphi^*) = \frac{2}{\varphi^*}(y - \sqrt{y^2 - 1} + \sec^{-1}y) \\
\hat{\psi}_{200} &= g(\hat{\varphi}_{200}) = \frac{2}{\hat{\varphi}_{200}}(y - \sqrt{y^2 - 1} + \sec^{-1}y)
\end{aligned}$$

From this we can calculate the standard error of $\hat{\Psi}_{200}$, the estimator of ψ^* , and subsequently its confidence interval:

$$\begin{aligned}
se(\hat{\Psi}_n) &= |g'(\varphi)| * se(\hat{\Phi}_n) \\
se(\hat{\Psi}_n) &= \left| \frac{d}{d\varphi} g(\varphi) \right| * se(\hat{\Phi}_n) \\
se(\hat{\Psi}_n) &= \left| \frac{d}{d\varphi} \frac{2}{\varphi}(y - \sqrt{y^2 - 1} + \sec^{-1}y) \right| * se(\hat{\Phi}_n) \\
se(\hat{\Psi}_n) &= |-2\varphi^{-2}|(y - \sqrt{y^2 - 1} + \sec^{-1}y) * se(\hat{\Phi}_n) \\
se(\hat{\Psi}_n) &= (2\varphi^{-2})(y - \sqrt{y^2 - 1} + \sec^{-1}y) * se(\hat{\Phi}_n)
\end{aligned}$$

where the estimated standard error is:

$$se(\hat{\Psi}_n) = (2\hat{\varphi}_{200}^{-2})(y - \sqrt{y^2 - 1} + \sec^{-1}y) * se(\hat{\Phi}_n) \quad \text{and} \quad se(\hat{\Phi}_n) = \frac{\hat{\varphi}_{200}(1-\hat{\varphi}_{200})}{200}$$

Now we can calculate the 95% confidence interval for $\psi^* \approx \pi$:

$$\hat{\psi}_{200} \pm 1.96 se(\hat{\psi}_{200})$$

22.1.3 Results

For the needle same in length as the distance between the lines, we got an approximation for π of 3.0769. The 95% Confidence Interval (2.7640, 3.3898), contains π .

For the needle shorter in length than the distance between the lines, we got an approximation for π of 2.9122. The 95% Confidence Interval (2.5861, 3.2384), contains π .

For the needle longer in length than the distance between the lines, we got an approximation for π of 2.8042. However, the 95% Confidence Interval (2.5769, 3.0316), does not contain π .

22.1.4 Conclusion

For the first two cases the 95% Confidence intervals both contain π so we cannot reject the null hypothesis that the outcomes from tossing a needle are directly related to π . For the final case the 95% Confidence interval does not contain π so we reject the null hypothesis and conclude that for this case the outcomes from tossing a needle are not directly related to π . The first two outcomes agree with Buffon's Needle while the third one may in fact be false, because our samples were quite small in hindsight.

Potential Modification

There are several ways we could improve this experiment. Firstly we could increase sample size and see if we can get better approximations of π . Secondly we could investigate the chance of a needle landing on a line on a square tiled floor, this is the Buffon-Laplace Needle. We could also extend the idea to not only cover needles and lines but also shapes and more complicated arrangements of tiles or lines.

Author Contributions

Amanda Hughes

Original concept; research; recording of the data collection; MATLAB analysis; writing and construction of the report; LateX writing and compiling; some of the slides for presentation and some of the script for the presentation (mainly sections on the geometric aspect of the problem).

Sung-Lim Suh

Research; collected data; majority of slides for presentation; majority of script for presentation, some of which was used in the report (Background on Comte de Buffon and Background of Buffon's Needle).

Many thanks to our lecturer Dr. Raazesh Sainudiin who spent much of his time discussing the geometric aspect of this report with us. We really appreciated being able to stop by his office almost whenever and have a discussion.

References

Texts:

Stigler, Stephen M., *Statistics on the Table: The History of Statistical Concepts and Methods*, Harvard University Press, USA, 2002

Electronic Texts:

Hyna, Irene; Oetiker, Tobias; Partl, Hubert; Schlegl, Elisabeth., *The Not so Short Introduction to LATEX 2e or LATEX 2e in 90 Minutes*, 2000

Websites:

www.angelfire.com/wa/hurben/buff.html
www.mathworld.wolfram.com/BuffonsNeedleProblem.html
www.mste.uiuc.edu/reese/buffon/buffon.html
www.nndb.com/people/755/000091482/
www.ucmp.berkeley.edu/history/buffon2.html
www.youtube.com/watch?v=Vws1jvMbs64

Appendix

Data and MATLAB Code:

```
Buffon.m
data% this is the data for the experiment where the needle is the same length as the distance between the lines
datalines=[1 1 1 0 0 1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 0 1 0 0 1 0 1 1 ...
1 1 0 0 1 0 1 1 1 0 1 1 1 0 0 0 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 ...
0 1 0 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 1 1 1 0 1 ...
1 0 0 1 1 0 0 0 0 1 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 ...
1 0 0 1 0 1 0 0 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 1 0 1 1 ...
1 0 1 1 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0 1]
MLE_theta_star=mean(datalines)% find the maximum likelihood estimator, the sample mean
piapprox=2*(1/MLE_theta_star)% estimate of pi
StdErr=sqrt((MLE_theta_star*(1-MLE_theta_star))/200)% standard error
CI=[piapprox-(1.96*StdErr*2/(MLE_theta_star)^2),piapprox+(1.96*StdErr*2/(MLE_theta_star)^2)]% 95% Confidence Interval ...
for our approximate of pi

short% this is the data for the experiment where the needle is shorter in length than the distance between the lines
datalines=[1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 0 0 ...
1 1 1 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 0 1 1 1 1 0 1 0 0 1 0 ...
1 1 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 1 ...
1 0 0 0 1 1 1 1 1 0 0 0 1 0 0 1 1 1 1 0 0 1 1 0 1 0 1 1 1 1 ...
0 1 1 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0 1 0 0 0 1 1 1 1 0 0 1 0 0 ...
0 1 1 1 1 0 0 1 1 0 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0 1 0 1 1]
MLE_theta_star=mean(datalines)% find the maximum likelihood estimator, the sample mean
x=37/42%x=length/distance
piapprox=(2*x)*(1/MLE_theta_star)% estimate of pi
StdErr=sqrt((MLE_theta_star*(1-MLE_theta_star))/200)% standard error
CI=[piapprox-(1.96*StdErr*(2*x)/(MLE_theta_star)^2),piapprox+(1.96*StdErr*(2*x)/(MLE_theta_star)^2)]% 95% Confidence ...
Interval for our approximate of pi

long% this is the data for the experiment where the needle is longer in length than the distance between the lines
datalines=[0 0 1 1 0 0 1 1 1 0 1 0 1 1 1 1 1 0 1 1 0 1 0 0 1 1 1 ...
0 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 ...
1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 ...
1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 1 0 0 1 ...
1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 ...
0 1 1 0 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1]
```

22.2 Estimating the Binomial probability p for a Galton's Quincunx

Bry Ashman and Ryan Lawrence

abstract

Galton's Quincunx is a physical device designed to simulate the discrete binomial distribution. we aim to create a physical model of the quincunx that is characterised by the probability of a ball going left is equal to the probability of it going right. From the conceptual model of the quincunx, we derive the binomial probability mass function. In order to evaluate the parameter of interest p , we will derive the maximum likelihood estimator and use this to estimate the actual parameter p of our physical model using 100 samples that are assumed to be independent and identically distributed.



22.2.1 Motivation & Introduction

The binomial distribution is a fundamental discrete probability distribution, being the natural extension of the Bernoulli trial to the sum of Bernoulli trials. The distribution describes the number of successes in a sequence of n binary trials, with a probability p . Each of these trials is a Bernoulli trial parameterised by p . A binomial distribution parameterised by $n = 1$ and p is simply a *Bernoulli*(p) trial.

The quincunx was invented by Sir Francis Galton originally to demonstrate the normal distribution. The quincunx is simply an array of pegs spaced so that when a ball is dropped into a device, it bounces off the pegs with a probability p of going right and a probability of $1-p$ of going left. It bounces off n pegs before being collected in a bin at the bottom of the device.

To this end, we aim to create a quincunx ideally parameterised by $p = 0.5$ with $n = 20$. To verify this, we will use maximum likelihood estimation to test the null hypothesis that $p = 0.5$.

22.2.2 Materials and Methods

Construction of physical model

The physical model that we created consisted of nails arranged in a pattern on a sheet of wood that we hoped would achieve as close to the ideal probability of $p=0.5$.

Materials

- Plywood Sheet (1200 x 600 x 20mm)
- Perspex Sheets (1200 x 600 x 2mm and 550 x 800 x 2mm)
- Timber Strips (1200 x 25mm and 600 x 25mm)
- Nails (30 x 2mm)
- Chrome Balls (20mm)

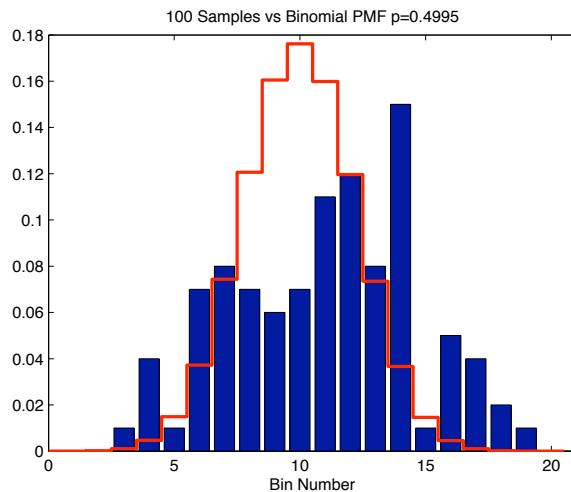
Construction Details

1. Mark 20 horizontal lines with 25mm spacings with the board in a portrait orientation.
2. Mark vertical lines at 25mm spacings from the centre of the board.
3. Place a nail at the top centre marking
4. Continue to place nails on the marked grid such that one marked grid point always separates the nails both vertically and horizontally.
5. Create the bins by attaching perspex strips directly below the nails of the last row.
6. Fit the edges to the main sheet.
7. The perspex sheet can now be attached to the edges of the quincunx.

A desirable feature of the quincunx is a release mechanism at the top to release the balls used to simulate a random variable and a release at the bottom to retrieve the balls after the experiment.

Sample Collection

To collect samples from the quincunx the balls are dropped into the device as identically as possible with sufficient time between each drop to ensure that the balls do not interfere with each other so as to keep the samples as identical as possible. The balls are collected in a series of bins numbered from 0 to 21, 0 representing the leftmost bin that the sample can be in and 21 being the rightmost bin. Since we assume that each sample is identical and independent, we record the cumulative number of balls in each bin after dropping 100 balls. The data is shown in the blue bars in the next figure.



22.2.3 Statistical Methodology

Deriving the binomial distribution

The binomial distribution can be thought of as a random walk in one dimension. The parameters map to this model as p being the probability of taking a step right and $(1 - p)$ the probability of taking a step left, and n being the total number of steps taken. From this, it follows that, for a given number of n steps, x of which are to the right and $n - x$ to the left, to find the probability that a combination of those n steps that will get you to the same point, you have to multiply the probability of the path by how many unique ways you can combine those steps. The number of ways of ordering the x right steps in a set of n steps is given by $\binom{n}{x}$. Therefore, the probability of ending up at a particular endpoint is as follows:

$$P(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}$$

A note about the endpoint: I have used the convention that the leftmost bucket is 0. The endpoint numbers also tell you how many right steps you have in the quincunx.

Parametric Estimation

In order to estimate the parameter p for our physical model, we will use a maximum likelihood estimator (MLE) since it is often regarded as asymptotically optimal. However, for the binomial distribution, the MLE is equivalent to the Method of Moments.

Deriving the maximum likelihood estimator

$$\begin{aligned} L(p) &= \prod_{i=1}^n \binom{N}{x_i} p^{x_i} (1 - p)^{N-x_i} \\ L(p) &= \prod_{i=1}^n \binom{N}{x_i} p^{\sum x_i} (1 - p)^{nN - \sum x_i} \\ \ln L(p) &= \sum_{i=1}^n \ln \binom{N}{x_i} \sum x_i \ln(p) (nN - \sum x_i) \ln(1 - p) \end{aligned}$$

$$\frac{d}{dp} \ln L(p) = \frac{\sum x_i}{p} - \frac{nN - \sum x_i}{1-p}$$

We can now set $\frac{d}{dp} \ln L(p) = 0$ to find the maximum:

$$0 = \frac{\sum x_i}{p} - \frac{nN - \sum x_i}{1-p}$$

$$p = \frac{1}{nN} \sum_{i=1}^n x_i$$

Which is equivalent to:

$$p = \frac{1}{N} E(X)$$

22.2.4 Results & Conclusion

Maximum Likelihood Estimation

The MLE of the parameter p of the quincunx is 0.4995, with a 95% normal based confidence interval of [0.4639,0.5351] calculated as derived above.

Conclusion

Maximum Likelihood Estimation from the 100 samples from the model of the quincunx has estimated the parameter for the binomial distribution to be in the range [0.4639,0.5351]. This would seem to verify that, in fact, even though the quincunx is a non-linear physical device that, overall, it is remarkably fair with $p=0.5$ within the 95% normal based confidence interval.

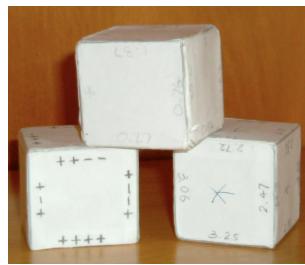
The estimated cumulative distribution function also suggests that the distribution will converge binomially. Thus, we can conclude as $n \rightarrow \infty$, it will converge on the standard normal distribution as a consequence of the central limit theorem.

22.3 Investigation of a Statistical Simulation from the 19th Century

Brett Versteegh and Zhu Sha

Abstract

This project is designed to investigate Sir Francis Galton's statistical dice experiment. We constructed Galton's dice according to his prescriptions and tested the null hypothesis that the outcomes from these dice do indeed follow a discrete approximation to the normal distribution with median error one. The inverse distribution function sampler and Chi Squared test are the statistical methodologies employed in this project.



22.3.1 Introduction and Motivation

The report will firstly cover the background and motivation of this project. Secondly, the methodologies used will be explained before outlining the results and subsequent conclusion found by undertaking this experiment. Finally, a potential modification to Galton's method will be examined as a means of sampling from a standard normal distribution.

Francis Galton

Born in 1822, Francis Galton was considered by many, at an early stage, to be a child prodigy. By the age of two, he could read; at five, he already knew some Greek, Latin and long division.

After his cousin, Charles Darwin, published *The Origin of Species* in 1859, Galton became fascinated by it and thus devoted much of his life to exploring and researching aspects of human variation. Galton's studies of heredity lead him to introduce the statistical concepts of regression and correlation. In addition to his statistical research, Galton also pioneered new concepts and ideologies in the fields of meteorology, psychology and genetics.

Statistical Dice

This experiment came about from Galton's need, as a statistician, to draw a series of values at random to suit various statistical purposes. Dice were chosen as he viewed them to be superior to any other randomisation device. Cards and marked balls were too tedious to be continually shuffled or mixed following each draw, especially if the required sample size was large.

The dice he created made use of every edge of each face which allowed for 24 equal possibilities as opposed to the six of a normal die.

For further details on Galton's experiment, please refer to his article "Dice for Statistical Experiments"; *Nature* (1890) No 1070, Vol 42 (This article is available free for download. Please refer to the references section for the website.)

Motivation

The motivation behind this project is to reconstruct Galton's dice using the methods outlined in his 1890 *Nature* article "Dice for Statistical Experiments" and then harness the power of modern computers to determine how effective this technique was for simulating random numbers from the following distribution.

Galton outlines that the samples were taken from a normal distribution with mean zero and median error one. We shall call this distribution Galton's Normal distribution or GN. However, for the experiment to work, we must use a discrete approximation of the normal distribution, which we will define as Galton's Discrete Normal or GDN. Both will be formally explained in the Methodology section.

To determine the success of this experiment, we formulate the following question as a statistical hypothesis test: "Are our sampled values taken independently and identically from an appropriate discrete distribution which approximates Galton's normal distribution?"

Materials and Methods

Experiment Process

In order to recreate Galton's Dice Experiment, we have chosen to replicate the design he explains in his *Nature* article.

Creating the Dice

We chose to use rimu as it was readily available and inexpensive, unlike the mahogany that Galton had access to. As per his specifications, the wood was cut into six cubes of 1.25 inches (3.2 cm) wide, high and deep, before being covered in a paper template that was designed to fit tightly around the wood. The paper was adhered using general PVA glue.

The only change to Galton's original specification was that we chose to write the values to two decimal places on the faces, as opposed to one decimal place. This was to ensure a higher level of precision when plotting the results.

Collecting the Data

The experiment was carried out by shaking all of the first three dice (dice 1) at once and rolling them across the flat surface of a table top. We interpreted Galton's terminology of the values that "front the eye" to be the results that one can see by looking directly down on top of the dice. The three dice were then lined up into a row and the values called out and entered onto a Notepad document. We used the following formula to calculate the optimal number of trials needed for our investigation: $f(x)_{min} * sample\ size \approx 5$, where $f(x)_{min}$ is the smallest probability for the discrete distribution.

The same rolling process was then performed for dice 2 (two dice at once) and 3 (only one die) with the single exception that we did not need to roll these dice as many times as dice 1.

22.3.2 Statistical Methodology

Firstly, we will define Galton's Normal distribution. As derived from an article published in *Statistical Science*¹, Galton's Normal Distribution has a mean of zero but the variance is not one. Instead,

¹Stochastic Simulation in the Nineteenth Century. *Statistical Science* (1991) Vol 6, No 1, pg 94.

Galton's sample is taken from a half-normal distribution with a "probable error" (median error) of one. This implies that the probability between zero and one is a quarter, allowing us to solve the following equation to determine the variance:

$$\begin{aligned}\varphi(x) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \\ \frac{1}{4} &= \int_0^1 \varphi(x) dx \\ \frac{1}{4} &= \int_0^1 \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \\ \sigma &= 1.4826\end{aligned}$$

$$\therefore \text{GN} \sim N(0, 1.4826^2)$$

Secondly, we must determine how Galton calculated the values² to use on his dice. It was our assumption that he used the midpoints of a set of intervals that partition $[0, 1]$ and we undertook the following processes to confirm this.

We divided the interval $[0, 1]$ equally into 24, with the last 3 intervals further divided into 24 subintervals. In total, this gave us 21 + 24 intervals to allocate along the y-axis. The midpoint of each interval was taken in order to compute its corresponding x value under the inverse CDF map.

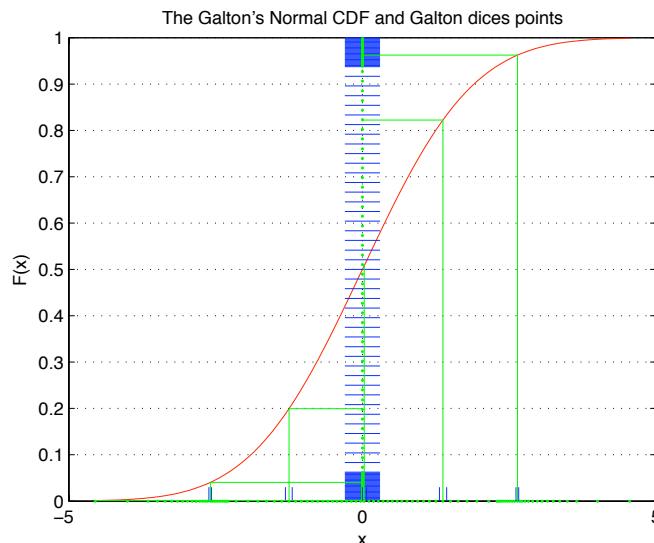


Figure 22.4: Plot showing the midpoints mapping back to specific values on the x axis.

The easiest way to do this would have been to evaluate the inverse CDF function at the midpoints. However, a closed form expression for the inverse CDF does not exist for a Normal distribution. Thus, we applied numerical methods to solve for x (Newton's method).

We believe the midpoint assumption was correct, as the mapped values are very close to Galton's actual figures and the differences can be attributed to an imprecise value for the standard deviation.

Thirdly, we can now determine Galton's discrete approximation to the Normal. This is necessary as the values drawn from throwing Galton's dice come from a discrete distribution, not the continuous Galton Normal. In doing this, we are also able to define our null hypothesis formally: $H_0 : x_1, x_2, \dots, x_n \text{ IID } \sim GDN$ Galton's Discrete Normal (GDN) is an approximation to Galton Normal (GN).

Fourthly, as the distribution is now discrete, we can apply the Chi Squared Test to evaluate our null hypothesis. The test used had the following parameters: Degrees of Freedom: $90 - 1 = 89$ $\alpha = 0.05$; Critical Value = 112.

²See Appendix B.

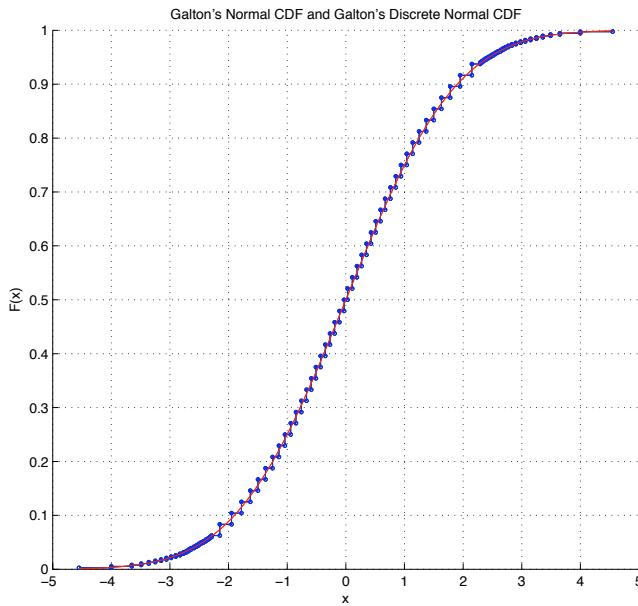


Figure 22.5: Plot showing both the GN and GDN CDFs. They are very similar.

22.3.3 Results

Once the experiment was complete and the results collated, they were run through a methodological tester to ensure all values were correct. Testing the data involved running all our sampled values through a Matlab function which checked each number against Galton's 45 possible values. Any values that did not match were outputted as ones and the erroneous data were removed before a graph was plotted to measure how well our experiment sampled from GDN.

Chi Squared Test

A Chi Squared test was then performed on the data and the results¹ are summarised below.

	Data Values	Observed Count	Expected Count	$(O - E)^2/E$
	-4.55	5	5.046875	0.000435372
	-4	7	5.046875	0.755853328
	-3.65	5	5.046875	0.000435372
	3.49	6	5.046875	0.180001935

	3.65	8	5.046875	1.727989551
	4	11	5.046875	7.022107198
	4.55	5	5.046875	0.000435372
Total		1938	1938	
Chi Test Result				83.54798762

$$T = \sum_{i=1}^{90} \frac{(Observeverd - Expected)^2}{Expected} = 83.548$$

¹For the full table, please see Appendix A.

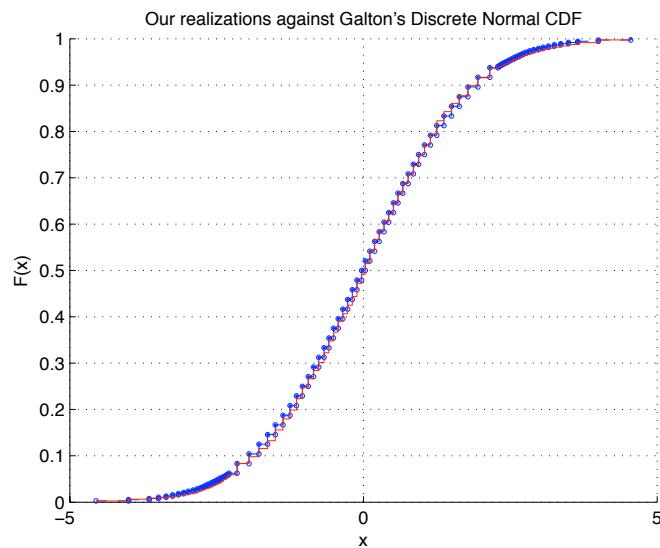
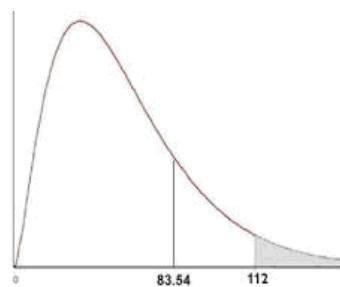


Figure 22.6: Plot showing the empirical DF of our results against GDN. Our values take on a staircase appearance and are very close to GDN. The main deviations occur mostly in the tails.



22.3.4 Conclusion

We cannot reject H_0 at $\alpha = 0.05$ because the observed test statistic is outside the rejection region. In relation to our statistical question, this means that there is insufficient evidence to suggest that our sample is not from GDN.

Potential Modification

Since the standard normal distribution is more common in all areas, we wanted to convert Galton's Dice into a new set which can be used for simulating the standard normal distribution.

In his experiment, Galton took the mid-point of each probability interval, and then found the corresponding x -values. Instead of applying a tedious calculation to find the x -values, we took a z -value table, and found the corresponding z -values to the upper bound of those intervals. This enables the creation of two new dice²:

Dice (1)	0.05	0.10	0.15	0.21	0.27	0.32
	0.37	0.43	0.49	0.55	0.61	0.67
	0.74	0.81	0.89	0.97	1.05	1.15
	1.26	1.38	1.53	*	*	*
Dice (2)	1.56	1.58	1.60	1.62	1.65	1.68
	1.70	1.73	1.76	1.79	1.83	1.86
	1.90	1.94	1.99	2.04	2.09	2.15
	2.23	2.31	2.42	2.56	2.80	4.00

Through Matlab, we were able to map the data gathered during our original experiment into the values shown in previous table, corresponding to the standard Normal, and develop the following plot:

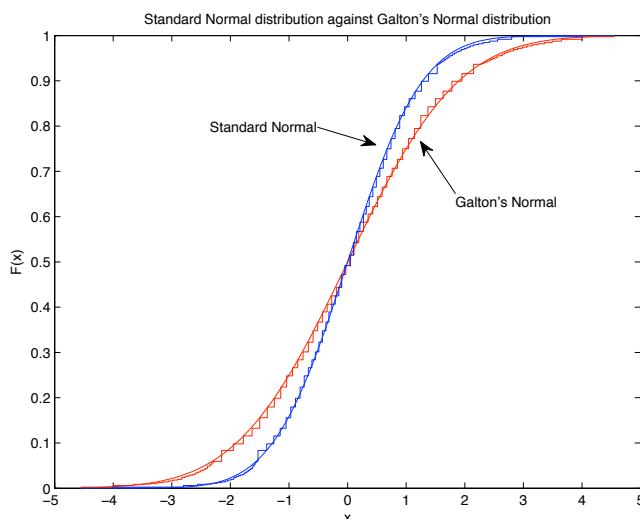


Figure 22.7: Plot showing the Standard Normal Distribution against Galton's Normal Distribution.

²Tables showing the new values for dice 1 & 2. The third dice can remain the same as Galton's.

Author Contributions

Brett - Constructed dice, gathered majority of the data results, constructed report, conducted spell/grammar check.

Joe - Wrote up Matlab code to analyse and plot data, entered in data results, constructed presentation and discovered a modification to Galton's experiment.

References

Dice for Statistical Experiments. *Nature* (1890) Vol 42, No 1070

Stochastic Simulation in the Nineteenth Century. *StatisticalScience* (1991) Vol 6, No 1

<http://www.galton.org>

<http://www.anu.edu.au/nceph/surfstat/surfstat-home/tables/normal.php>

Appendix A

Data	Count	Expected Count	$(O - E)^2/E$	Data	Count	Expected Count	$(O - E)^2/E$
-4.55	5	5.046875	0.000435372	0.11	53	40.375	3.947755418
-4	7	5.046875	0.755853328	0.19	48	40.375	1.44001548
-3.65	5	5.046875	0.000435372	0.27	40	40.375	0.003482972
-3.49	1	5.046875	3.245017415	0.35	35	40.375	0.715557276
-3.36	3	5.046875	0.830156734	0.43	32	40.375	1.737229102
-3.25	3	5.046875	0.830156734	0.51	42	40.375	0.065402477
-3.15	3	5.046875	0.830156734	0.59	41	40.375	0.009674923
-3.06	4	5.046875	0.217153638	0.67	46	40.375	0.783668731
-2.98	4	5.046875	0.217153638	0.76	35	40.375	0.715557276
-2.9	3	5.046875	0.830156734	0.85	38	40.375	0.139705882
-2.83	8	5.046875	1.727989551	0.94	45	40.375	0.529798762
-2.77	5	5.046875	0.000435372	1.04	44	40.375	0.325464396
-2.72	3	5.046875	0.830156734	1.14	43	40.375	0.170665635
-2.68	3	5.046875	0.830156734	1.25	55	40.375	5.297600619
-2.64	3	5.046875	0.830156734	1.37	38	40.375	0.139705882
-2.59	6	5.046875	0.180001935	1.5	35	40.375	0.715557276
-2.55	4	5.046875	0.217153638	1.63	32	40.375	1.737229102
-2.51	5	5.046875	0.000435372	1.78	42	40.375	0.065402477
-2.47	4	5.046875	0.217153638	1.95	33	40.375	1.347136223
-2.43	6	5.046875	0.180001935	2.15	40	40.375	0.003482972
-2.39	10	5.046875	4.861116486	2.29	3	5.046875	0.830156734
-2.35	6	5.046875	0.180001935	2.32	4	5.046875	0.217153638
-2.32	8	5.046875	1.727989551	2.35	3	5.046875	0.830156734
-2.29	6	5.046875	0.180001935	2.39	4	5.046875	0.217153638
-2.15	47	40.375	1.087074303	2.43	6	5.046875	0.180001935
-1.95	28	40.375	3.792956656	2.47	5	5.046875	0.000435372
-1.78	34	40.375	1.006578947	2.51	3	5.046875	0.830156734
-1.63	33	40.375	1.347136223	2.55	8	5.046875	1.727989551
-1.5	45	40.375	0.529798762	2.59	1	5.046875	3.245017415
-1.37	46	40.375	0.783668731	2.64	7	5.046875	0.755853328
-1.25	37	40.375	0.282120743	2.68	5	5.046875	0.000435372
-1.14	48	40.375	1.44001548	2.72	4	5.046875	0.217153638
-1.04	48	40.375	1.44001548	2.77	4	5.046875	0.217153638
-0.94	35	40.375	0.715557276	2.83	6	5.046875	0.180001935
-0.85	34	40.375	1.006578947	2.9	5	5.046875	0.000435372
-0.76	34	40.375	1.006578947	2.98	5	5.046875	0.000435372
-0.67	41	40.375	0.096749293	3.06	6	5.046875	0.180001935
-0.59	49	40.375	1.84249226	3.15	5	5.046875	0.000435372
-0.51	37	40.375	0.282120743	3.25	4	5.046875	0.217153638
-0.43	44	40.375	0.325464396	3.36	5	5.046875	0.000435372
-0.35	33	40.375	1.347136223	3.49	6	5.046875	0.180001935
-0.27	36	40.375	0.474071207	3.65	8	5.046875	1.727989551
-0.19	36	40.375	0.474071207	4	11	5.046875	7.022107198
-0.11	55	40.375	5.297600619	4.55	5	5.046875	0.000435372
-0.03	38	40.375	0.139705882	Total	1938	1938	
0.03	45	40.375	0.529798762	Chi ² Result			83.54798762

Appendix B

Table 1				Table 2				Table 3			
0.03	0.51	1.04	1.78	2.29	2.51	2.77	3.25	++++	+++	-++	+-+
0.11	0.59	1.14	1.95	2.32	2.55	2.83	3.36	+++-	+--	-+-	+-
0.19	0.67	1.25	2.15	2.35	2.59	2.90	3.49	++-+	-+++	--+	-++
0.27	0.76	1.37	*	2.59	2.64	2.98	3.65	++-	-++-	---	-+-
0.35	0.85	1.50	*	2.43	2.68	3.06	4.00	+---	-++-	+++	-+
0.43	0.94	1.63	*	2.47	2.72	3.15	4.55	+---	-+-	++-	---

22.4 Testing the average waiting time for the Orbiter Bus Service

J Fenemore and Y Wang

Abstract

The Metro-owned and operated Orbiter bus service in Christchurch city is a very popular service that links up some of Christchurch's main suburbs, places and attractions. The timetable provided by the Metro bus company claims that on weekdays between 6 a.m. and 7 p.m., a service will arrive at any given stop every ten minutes, regardless of whether that service travels clockwise or anticlockwise. I hypothesise that this is not the case and that arrivals are influenced by many other factors including current traffic volume, traffic accidents, pedestrian volume, traffic light stoppages and passenger boarding times. We tested this hypothesis by sitting at the UCSA bus stops and recording arrival times.



22.4.1 Motivation

The Orbiter is a highly used bus service and I myself often use this service. Many times while waiting for the service, I have noticed that more often than not, two Orbiter buses arrive at the stop at the same time or within a very short time of each other. Because of logistical reasons, I believe the Metro bus company would not run more buses than needed, meaning that if two buses arrived 'back to back' then there would be a twenty minute wait for the next bus (as the waiting time should be only ten minutes, so for two buses, the time is doubled.) This type of scenario significantly affects the times specified by Metro. For this reason, I believe that in reality, the average waiting time/arrival time is not ten minutes. It is important to note that the timetables distributed by Metro give specific times when buses arrive. These times are all ten minutes apart, which I feel can only be interpreted as meaning that a bus will arrive at a stop every ten minutes and the maximum waiting time for a passenger is also ten minutes. So for the two buses arriving in the 'back to back' situation, while the average time of arrival is presented as every ten minutes on paper, in reality, the buses do not arrive specifically ten minutes apart as claimed. This circumstance also gives a variation of ten minutes and decreases the probability of actually waiting only ten minutes. I wish to address this issue of the average waiting times of the buses in relation to the timetables provided and the variation in actual arrivals. Therefore, by examining the arrival times of the buses and recording waiting times, it can be examined just how accurate the timetables given are and whether they are based on average times or specific times. These issues affect Metro's reliability and credibility.

22.4.2 Method

The experiment we carried out is relatively simple. We sat at the bus stop outside the UCSA building on Ilam road and recorded the arrival times of each Orbiter bus and then calculated the waiting times between each bus. This was done for both clockwise and anticlockwise directions. The waiting time for the first bus in both directions was taken from the time of our arrival to the stop. After that, the waiting time was calculated as the times between bus arrivals.

A range of times were recorded, which covered an entire working day - 8 a.m. to 5 p.m.. These times were recorded on different days to assess not only the time of day but also different days, so we could see how these differences affect the times. The different times give a fairer assessment of the waiting times. It was assumed that for each day of the week, the waiting times for specific times of the day are relatively the same. A sample taken any day at a specific time would represent all days in the week at that time. The experiment was conducted in this manner because of availability and time restrictions.

While we realise that taking more samples would increase accuracy and reliability while also giving a better description of actual events, we felt it impractical to sit at the stop and record times for an entire day for each day of the week.

Statistical Methodology

For the experiment, we modelled the distribution of the inter-arrival times or waiting times of the Orbiter bus service using the exponential distribution. The probability distribution function is as shown below. The distribution is continuous.

$$f(x; \lambda) = \lambda * \exp(-\lambda * x)$$

Where: x is the waiting time, and λ is the rate parameter or $1/\text{mean}(x)$.

The mean of this distribution is $1/\lambda$ and has variance $1/\lambda^2$.

The exponential distribution was chosen because of its important memory-less property. Each new waiting time for the next bus is completely independent of the past waiting times. Each bus's arrival is assumed to be independent of the last.

For this experiment, I will be testing whether the average waiting time is ten minutes. More formally:

H_0 (null hypothesis): $\mu = 10$ minutes

H_A (Alternative hypothesis): $\mu \neq 10$ minutes

To test this hypothesis, we used non-parametric bootstrap methods to estimate λ and obtain a 95% confidence interval for this value. These values will be formed by sampling the data observed with replacement, at equal probabilities, 132 times, of which an average will be taken. The whole process was then repeated 1000 times. An overall average calculated λ will be then transformed into an average waiting time using the formula:

$$\mu = 1/\lambda$$

where μ is the average.

This will then be compared and contrasted against the average waiting time found by generating 132 realisations of waiting times then calculating the average of these, then repeating this process 1000 times. This is a parametric bootstrap based technique. For this, $\lambda = 1/10$ (where $\mu = 10$ minutes and using the formula above.) An overall average will be found along with a 95% confidence

interval for this value. By comparing these intervals and mean values, an accurate decision will be made as to whether buses do arrive on average every ten minutes or not.

Probabilities of certain arrival times around ten minutes will be evaluated to show the accuracy of the service.

The **Matlab** code for this process is given in Appendix III.

22.4.3 Results

The raw data is given in Appendix IV.

The average waiting time for the anticlockwise direction = 9.19 mins.

The average waiting time for the clockwise direction = 8.95 mins.

The total average = 9.07 mins.

The minimum waiting time = 0 mins.

The maximum waiting time = 28 mins.

There are 66 waiting time samples for each direction.

Notes for the data:

- Some buses waited at the stop for random amounts of time in order to space the buses apart (this was never for long: 1 or 2 minutes). This was not taken into account when recording arrival times.
- School rush traffic (heavier volumes) was present from 3 p.m. to 3.30 p.m. approx.
- Evening commuter rush was present from approx 4.30 p.m. onwards.
- Morning commuter rush was from 8 a.m. to 9.30 p.m. approx.

Observations on the data: The anticlockwise direction tends to be much more consistent, having a closer average to ten minutes and more observed times close to ten minutes.

22.4.4 Discussion

During less busy hours, buses arrive much more regularly.

The results of the code in (Appendix III) are as follows:

Calculated sample $\lambda = 0.1105$. The calculated 95% confidence interval for this value is [0.1078,0.1129]. The claimed lambda is $\lambda = 0.1$. As the calculated sample is within this interval and the claimed λ is below, we can see that the bus arrival is slightly less than claimed (using $\mu = 1/\lambda$).

Using the claimed λ , the randomly obtained mean value for waiting time is $\mu = 9.9842$. The calculated 95% confidence interval for the mean waiting time is [9.2882,10.6237].

I found from the samples that the anticlockwise, clockwise and total mean waiting times are $\mu = 9.19$, 8.95 and 9.07, respectively. None of these values is within the interval previously stated.

When the calculated sample λ of 0.1105 was used to produce the mean waiting time and its 95% confidence interval, the following was produced: $\mu=9.0350$ and [8.4957, 9.6137].

It is important to note that it is seen in the graph, from the empirical CDF, that the probability of having short waiting times is high - the probability of waiting 10 minutes or less according to our observed data is 6288. This value is quite high but the probability of waiting 15 minutes or more is 0.1288 which, in reality, is relatively high also. Practically, this means 1 in every 10 times you wait for an Orbiter to arrive, it will take 15 minutes or more to come. This value may be acceptable by Metro and indeed is good, considering so many unknown factors in traffic, but it would surely frustrate passengers being 5 minutes behind time.

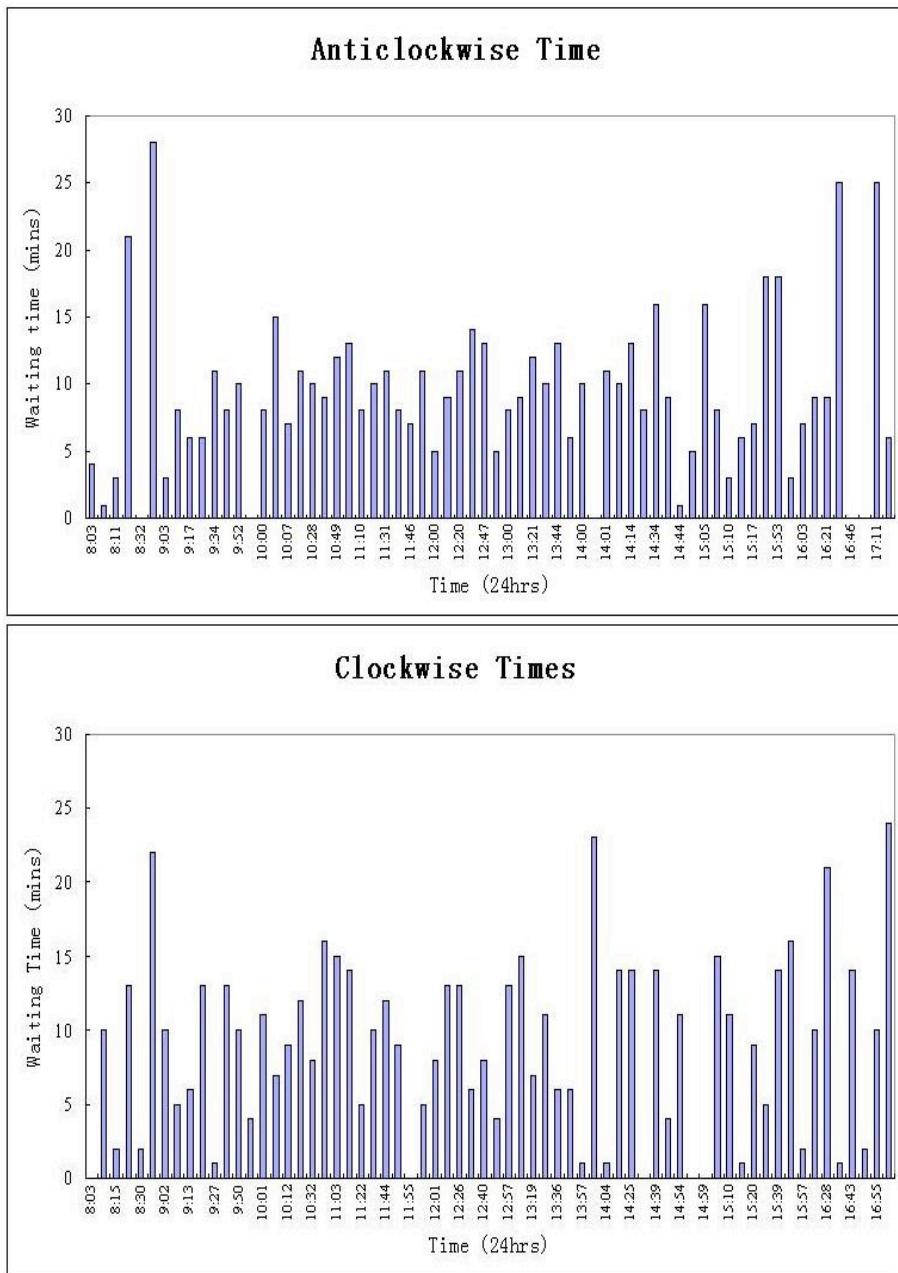


Figure 22.8: From the graphs above, we can see that often, a short wait is followed by a long wait, in both directions. Also, the anticlockwise times are generally much closer to 10 minutes waiting time. It is also seen that around rush hour times (8:30, 15:00, 16:45), a pattern emerged where several buses in quick succession were followed by a long wait for the next bus to arrive. This could be because of the time taken for more passengers than usual to aboard and depart, and areas where traffic volume is greater at these times.

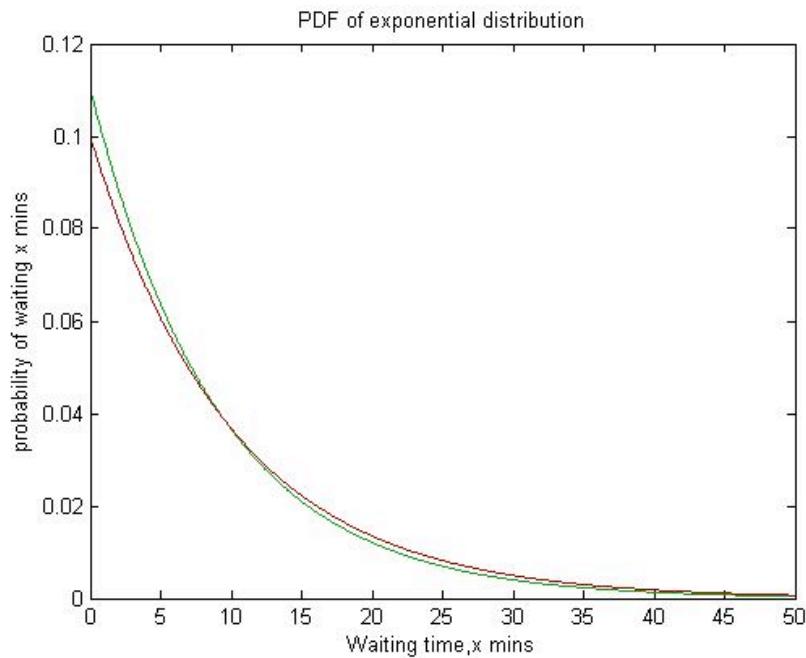


Figure 22.9: This graph shows the probability distribution function for the exponential function with the green line indicating a λ value of 0.1, the claimed λ . The red line indicates the value of λ estimated, 0.1105. From this graph, you can see the probability of getting a short waiting time is high - approximately 0.06, while the probability of a long waiting time is much much lower - approximately 0.01. The **Matlab** code for this graph is shown in Appendix I.

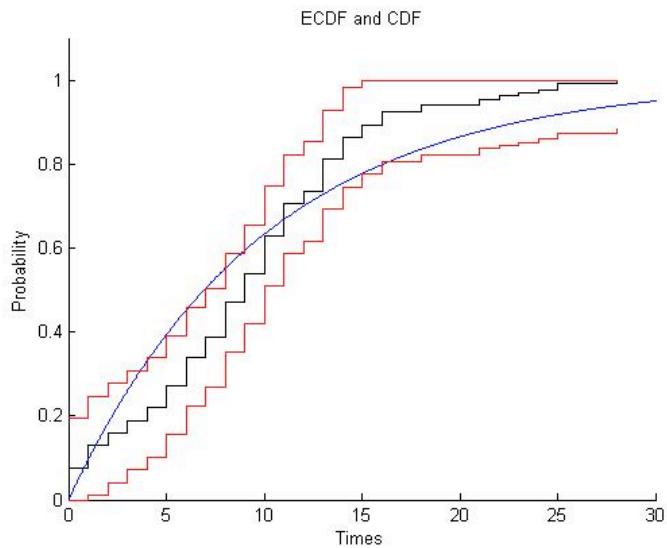


Figure 22.10: This plot is the Empirical CDF plot(black), with a 95% confidence interval (red) and the actual CDF based on claimed $\lambda = 0.1$ (blue). The **Matlab** code for this graph is given in Appendix II. This graph shows the accuracy of the empirical distribution and hence the accuracy of the data we collected. There are some inconsistencies caused by the randomness of inter-arrival times but our empirical CDF is generally good as the actual CDF lies mostly within the interval lines. With more data points, our accuracy would greatly improve.

22.4.5 Conclusion

The calculated value of λ is 0.1105. When this value is used to estimate the mean waiting time, and including the observed waiting times, we can conclude that Metro delivers a service better than claimed. From this λ , we see a mean waiting time of 9.04 minutes - 58 seconds less than the 10 minutes wait claimed.

Furthermore, the mean waiting time estimate calculated and its 95% confidence interval (not including the average waiting times observed) cause us to not accept the null hypothesis, H_0 of $\mu = 10$ minutes at the 95% confidence level.

From all of this, we can confirm that Metro is quite right in claiming an arrival of an Orbiter bus every ten minutes at any stop. In fact, it appears that they do better than this by a whole minute. However, it is all very well to claim this on paper but it is crucial to note that waiting times of 28 minutes do happen, rarely. This illustrates a very important difference between practical and statistical significance. In this case, it has no major effects, as the observed waiting time is less than claimed.

Author Contributions

Josh's Contributions: The original concept; data recordings for Wednesday, Thursday and Friday (5hrs); data organisation; analysis; methodology and implementation and the preliminary report, final report and presentation notes. *Yirang's Contributions:* Monday's and Tuesday's data recordings (4hrs).

Appendices

I

```
x=linspace(0,50,1000);%Array of x points to evaluate
lambda1=0.1105%Estimated lambda
f1=lambda1*exp(-lambda1.*x);%Calculated probabilities
plot(x,f1,'color',[0 0.6 0])%Plot coloured red
hold
lambda2=0.1%Claimed lambda
f2=lambda2*exp(-lambda2.*x);%Calculated probabilities
plot(x,f2,'color',[0.6 0 0])%Plot coloured green
xlabel('Waiting time,x mins')%Graph titles
ylabel('probability of waiting x mins')
title('PDF of exponential distribution')
```

II

```
antiTimes=[8 3 7 18 18 3 7 9 9 25 0 0 25 6 ...
10 0 10 8 16 9 1 5 16 6 4 1 3 21 0 28 3 8 ...
6 6 11 8 10 15 0 8 7 11 10 9 12 13 8 10 11 8 ...
7 11 5 9 11 14 13 5 8 9 12 10 13 6 11 13];
clockTimes=[0 0 11 1 9 5 14 16 2 10 21 1 14 2 10 ...
24 6 1 14 14 0 14 4 11 15 0 10 2 13 2 22 ...
10 5 6 13 1 13 10 11 4 7 9 12 8 16 15 14 5 ...
10 12 9 8 0 5 13 13 6 8 4 13 15 7 11 6 23 1];
%The raw data-the waiting times for each direction
```

```

sampleTimes=[antiTimes clockTimes];%dd all times into 1 array

x=linspace(0,30,1000);%Create array
lambda1=0.1;%Set claimed lambda
f=1-exp(-lambda1*x);%Create cdf realisations based on claimed lambda

[x1 y1]=ECDF2(sampleTimes,7,0,0);
%Call to class distributed ECDF fuction, save output values in arrays x1
%and y1
hold on%Hold plots for superimposition
plot(x,f)

Alpha=0.05;%set alpha to 5%
SampleSize=132;

Epsn=sqrt((1/(2*SampleSize))*log(2/Alpha));%epsilon_n for the confidence band

stairs(x1,max(y1-Epsn,zeros(1,length(y1))),’r’);%lower band plot
stairs(x1,min(y1+Epsn,ones(1,length(y1))),’r’);%upper band plot
hold off
axis([0,30,0,1])
title(’ECDF and CDF’)
xlabel(’Times’)
ylabel(’Probability’)

```

III

```

clear
antiTimes=[8 3 7 18 18 3 7 9 9 25 0 0 25 6 ...
10 0 10 8 16 9 1 5 16 6 4 1 3 21 0 28 3 8 ...
6 6 11 8 10 15 0 8 7 11 10 9 12 13 8 10 11 8 ...
7 11 5 9 11 14 13 5 8 9 12 10 13 6 11 13];
clockTimes=[0 0 11 1 9 5 14 16 2 10 21 1 14 2 10 ...
24 6 1 14 14 0 14 4 11 15 0 10 2 13 2 22 ...
10 5 6 13 1 13 10 11 4 7 9 12 8 16 15 14 5 ...
10 12 9 8 0 5 13 13 6 8 4 13 15 7 11 6 23 1];
%The raw data - the waiting times for each direction

rand(’twister’,489110);%set the seed for rand so results can be reproduced
sampleTimes=[antiTimes clockTimes];%All the sample times collected
lambdaTotal=zeros(1000,1);%An empty array

lambdaObs=1/mean(sampleTimes)%The lambda value for observed samples
lambdaClaimed=1/10 %The lambda claimed by Metro

%This is a non-parametric bootstrap
for j=1:1000%Loop to create 1000 lambdas
    for i=1:132 %A loop to sample with replacement 132 times at equal
        %probability
        u1=rand;%Generate a random number
        x1=deMoivreEqui(u1,132);%Select a random number between 1 and 132
        b(i)=sampleTimes(x1);%Array of random sample times, taken from
        %all samples, using random number generated
    end
    lambdaTotal(j)=1/mean(b);%lambda value for each array of random samples
end

sampleLambda=mean(lambdaTotal)
%The mean lambda for all the lambdas calculted
sortedLambdaTotal=sort(lambdaTotal);%Sort lambdas generated
lowerBound=lambdaTotal(25)%Calculate a 95% confidence interval for lambda
upperBound=lambdaTotal(975)

realisationsClaimed=zeros(1000,1);%An empty array
meanClaimed=zeros(1000,1);%An empty array
%This is parametric bootstrap
for x=1:1000%Loop to create 1000 mean waiting times based on claimed lambda

```

```

for z=1:132 %Loop to generate 1000 waiting times based on claimed lambda
    u2=rand;%Generate a random number
    realisationsClaimed(z)=-(1/lambdaClaimed)*log(u2);
    %Create realisation of x, random number u and lambda claimed
end
meanClaimed(x)=mean(realisationsClaimed);
%Find mean of each array of realisations
end
meanOfClaim=mean(meanClaimed)%Overall mean of realisations created
meanClaimed=sort(meanClaimed);%Sort array
lowerBound=meanClaimed(25)
%Create a 95% confidence interval for the mean found
upperBound=meanClaimed(975)

```

The above code was written 15/10/07 by J Fenemore and makes use of the following function:

```

function x = deMoivreEqui(u,k);
%
% return samples from deMoivre(1/k,1/k,...,1/k) RV X
%
% File Dates : Created 08/06/07 Modified 08/06/07
% Author(s) : Raaz
%
% Call Syntax: x = deMoivreEqui(u,k);
%               deMoivreEqui(u,k);
%
% Input      : u = array of uniform random numbers e.g. rand
%               k = number of equi-probabble outcomes of X
% Output     : x = samples from X
%
x = ceil(k * u) ; % ceil(y) is the smallest integer larger than y
% floor is useful when the outcomes are {0,1,...,k-1}
%x = floor(k * u);
%%%%%%%%%%%%%%%

```

IV. Raw data

Anti-clockwise Route		Clockwise Route	
Bus Arrival Times:	Mins waiting:	Bus Arrival Times:	Mins waiting:
15 : 07	8	14 : 59	0
15 : 10	3	14 : 59	0
15 : 17	7	15 : 10	11
15 : 35	18	15 : 11	1
15 : 53	18	15 : 20	9
15 : 56	3	15 : 25	5
16 : 03	7	15 : 39	14
16 : 12	9	15 : 55	16
16 : 21	9	15 : 57	2
16 : 46	25	16 : 07	10
16 : 46	0	16 : 28	21
16 : 46	0	16 : 29	1
17 : 11	25	16 : 43	14
17 : 17	6	16 : 45	2
		16 : 55	10
		17 : 19	24
14 : 00	10	13 : 56	6
14 : 00	0	13 : 57	1
14 : 10	10	14 : 11	14
14 : 18	8	14 : 25	14
14 : 34	16	14 : 25	0
14 : 43	9	14 : 39	14
14 : 44	1	14 : 43	4
14 : 49	5	14 : 54	11
15 : 05	16	15 : 09	15
15 : 11	6		
8 : 03	4	8 : 03	0
8 : 08	1	8 : 13	10
8 : 11	3	8 : 15	2
8 : 32	21	8 : 28	13
8 : 32	0	8 : 30	2
9 : 00	28	8 : 52	22
9 : 03	3	9 : 02	10
9 : 11	8	9 : 07	5

Anti-clockwise Route		Clockwise Route	
Bus Arrival Times:	Mins waiting:	Bus Arrival Times:	Mins waiting:
9 : 17	6	9 : 13	6
9 : 23	6	9 : 26	13
9 : 34	11	9 : 27	1
9 : 42	8	9 : 40	13
9 : 52	10	9 : 50	10
10 : 07	15	10 : 01	11
9 : 52	0	9 : 56	4
10 : 00	8	10 : 03	7
10 : 07	7	10 : 12	9
10 : 18	11	10 : 24	12
10 : 28	10	10 : 32	8
10 : 37	9	10 : 48	16
10 : 49	12	11 : 03	15
11 : 02	13	11 : 17	14
11 : 10	8	11 : 22	5
11 : 20	10	11 : 32	10
11 : 31	11	11 : 44	12
11 : 39	8	11 : 53	9
11 : 46	7	12 : 01	8
11 : 57	11		
12 : 00	5	11 : 55	0
12 : 09	9	12 : 00	5
12 : 20	11	12 : 13	13
12 : 34	14	12 : 26	13
12 : 47	13	12 : 32	6
12 : 52	5	12 : 40	8
13 : 00	8	12 : 44	4
13 : 09	9	12 : 57	13
13 : 21	12	13 : 12	15
13 : 31	10	13 : 19	7
13 : 44	13	13 : 30	11
13 : 50	6	13 : 36	6
14 : 01	11	13 : 59	23
14 : 14	13	14 : 04	1

22.5 Diameter of *Dosinia* Shells

Guo Yaozong and Shen Chun

22.5.1 Introduction and Objective

We collected some shells from New Brighton Pier that are commonly called *Dosinia anus* (Coarse Venus Shell). This species is a member of the class Bivalvia. Bivalvia lack a radula, and feed by filtering out fine particles of organic matter either from seawater (suspension feeders) or from surface mud (deposit feeders). In each case, food enters the mantle cavity in a current of water produced by cilia on the gills. Gills have a large surface area in relation to the size of the animal, and secrete copious amounts of slime-like mucus that not only traps the food particles but also acts as a lubricant for the passage of food to the mouth. In addition to having this feeding role, gills are the respiratory structures and are richly supplied with blood dorsally. Sexes are separate, although there is no external dimorphism. Gametes are shed into the seawater, where fertilisation occurs.



Unlike Venus shells from other parts of the world, this species has a flat disc-like shell. Found just below the low-tide mark along Brighton beach, it burrows just below the sand surface and feeds using two short, separate siphons. (*Life in The Estuary*, Malcolm B. Jones & Islay D. Marsden, Canterbury University Press, 2005).

Our objective was to test whether the diameters of *Dosinia anus* shells on the north side of New Brighton Pier are identically distributed to those found on the south side of the pier.

22.5.2 Materials and Methods

We collected shells along the New Brighton beach to the left (north) and right (south) of the pier. We walked and picked up all the shells we could see, except broken ones. In about two and a half hours, we collected about two buckets of shells from each side of the pier (i.e. two from the left and two from the right).

After washing, drying and classifying the shells, we found that 254 of them were *Dosinia anus*, 115 collected from north of the pier and 139 from the southern. Then we used mechanical pencils to sketch the outline of each shell onto graph paper and measured the diameter of each in units of millimetres. The way we measured them was from top to bottom (as shown below). After that, we entered the data into a computer, and estimated the empirical CDF as well as confidence bands.

Statistical Methodology

In order to test the null hypothesis that the shell diameters of our species are identically distributed on both sides of the pier, we applied the non-parametric permutation test.

By using the permutation test, we tested whether the absolute difference between the two sample means were significantly different from each other.

Step 1: Observe value: $T = X(\text{left}) - X(\text{right})$

Step 2: Combine [L1 L2 L115 R1 R139] '254 Data'

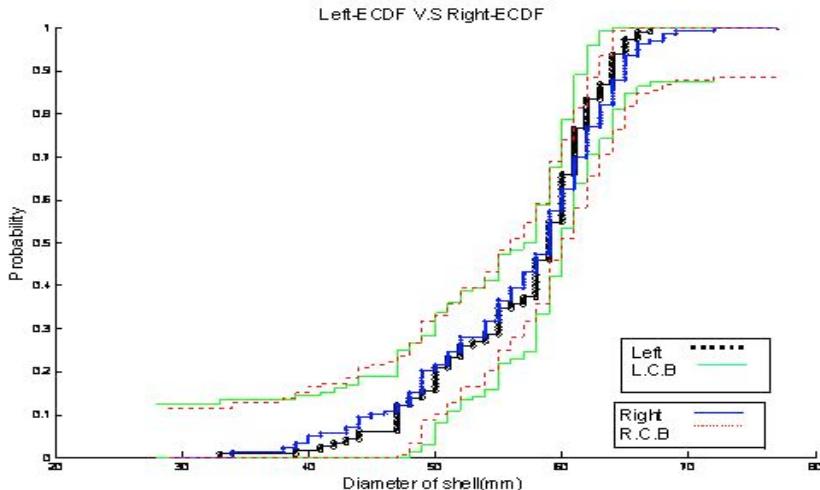
Step 3: Rearrange (MATLAB function: 'randperm'):

$$\begin{array}{c} [\text{R110}, \text{L45}, \text{L78}, \text{R2} \dots | \dots \dots \dots \text{L20}] \\ \downarrow \quad \quad \quad \downarrow \\ |\text{Mean}(1-115) \quad \quad \quad \text{--} \quad \text{Mean}(116-254)| = D_i(\text{recorded}) \\ (i = 1, 2, 3 \dots 10000) \end{array}$$

Step 4: Repeat Step 3 10000 times

Step 5: Find out how often ' D_i ' is greater than ' T ', then divided this value by **10,000**. This is our **P-value**.

22.5.3 Results



Hypothesis testing

$\text{abs}(\text{'mean for north'} - \text{'mean for south'}) = 0.1711$ (observed value) H_0 : No difference can be observed between north and south H_a : A difference can be observed Alpha = 0.05

In the test, we found 8470 numbers were greater than 0.1711, so P-value = $8470 / 10000 = 0.847$

Conclusion

Since p-value is large, we do not reject the null hypothesis, as we do not have enough evidence to say that there is a difference in the distribution of *Dosinia anus* diameters between the north and south sides of the pier in New Brighton Pier.

Author contributions

Shen Chun and Yaozong Guo did all the work together.