8-6-2019 2:00 PM

Advances in Moment-Based Distributional Methodologies

Yishan Zang, *The University of Western Ontario*

Supervisor: Provost, Serge B., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Statistics and Actuarial Sciences

Abstract

This thesis includes various results that rely on the moments of a distribution or the sample moments associated with a set of observations. Since a sample of size n is uniquely specified by its first n moments, it is pertinent to make use of sample moments for modeling, classification or statistical inference purposes. Three density mixtures are approximated by adjusting in various ways an initial density approximation referred to a base density by means of certain moment-based functions, and the accuracy of the resulting density approximants are compared. A similar study is carried out in the context of density estimation. Moreover, it is explained that methodologies that are based on moments are, in fact, ideally suited to model massive data sets. Various types of quasi-Monte Carlo deterministic samples are then compared to randomly generated samples with respect to their distributional representativeness. As well, a novel methodology depending on an arctangent transformation is introduced for classifying the tail behaviour of probability laws. Finally, certain approximations to the distributions of quadratic forms in gamma, inverse Gaussian, binomial and Poisson random variables, which rely on a symbolic expansion of their moments, are proposed.

Keywords: Density approximation, data modeling, quasi-Monte Carlo samples, classification of tail behaviour, quadratic forms.

Summary

Various statistical results of interest that are based on the moments of a distribution are presented in this thesis. In fact, the moments associated with a sample of observations contain all the distributional information therein available. Several types of adjustments and samples are investigated in order to determine which ones will provide the most accurate representations of a given distribution. As well, a simple new criterion is proposed to categorize the tail behaviour of probability laws. Finally, an efficient approach is proposed for approximating the distribution of quadratic forms in several types of random variables, which are utilized in connection with contingency tables and generalized linear models.

The journey of a thousand miles starts with

a single step.

Acknowledgements

First of all, I am very grateful to my supervisor, Professor Serge B. Provost, for his dedicated and inspiring guidance, which greatly improved my research and writing skills. Thanks are also due to the thesis examiners, Neil Klar, Wenqing He and Jiandong Ren. I would also like to express my sincere thanks to the Department of Statistical and Actuarial Sciences and the School of Graduate and Postdoctoral Studies for their financial support. Finally, I am indebted to my parents for their love, encouragement and support.

Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction

Several distribution approximation techniques that rely on the moments or the cumulants of a random variable have been proposed in the statistical literature. For example, approximants of this type can be obtained by making use of Pearson or Johnson curves, see Solomon and Stephens (1978), Elderton and Johnson (1969) and Rose and Smith (2002), or saddlepoint approximations as discussed in Reid (1988). These methodologies can provide adequate approximations in a variety of applications involving unimodal distributions. However, they may prove difficult to implement. The approximants proposed in this thesis have relatively simple functional forms that lend themselves to algebraic manipulations and apply to a very wide array of distributions. Moreover, their accuracy can be improved by making use of additional moments. Interestingly, another technique called the inverse Mellin transform, which is based on the complex moments of certain distributions, provides representations of their exact density functions in terms of generalized hypergeometric functions; for theoretical considerations as well as various applications, the reader is referred to Mathai and Saxena (1978) and Provost and Rudiuk (1995).

There exist several types of density estimates and approximants. However, many of these techniques will fail to provide adequate approximations, especially when the target density is not a smooth unimodal function. Silverman (1986) provides a survey of the various available methodologies. Efromovich (1999) presents a unified account of nonparametric approaches to density estimation. Other types of nonparametric density estimates that are based on the L_1 norm are presented in Devroye (1985) while both parametric and nonparametric approaches are discussed in Eggermont (2001). The multivariate case is extensively treated in Scott (2015).

Two of the main approaches advocated in Chapter 2 are based on Results 1 and 2. First, the exact density function associated with a distribution whose first n moments are known can be approximated by means of the product of a base density function, whose parameters are determined by matching moments, and a polynomial of degree n, whose coefficients are obtained by making use of the method of moments. This general semiparametric approach to density approximation, which appeared in Provost (2005), is formally stated in the following result.

Result 1 *Let $f_Y(y)$ be the density function of a continuous random variable Y defined in the interval (a,b), $E(Y^j) \equiv \mu_Y(j)$, $X = (Y-u)/s$ be an affine transformation (oftentimes, $u = E(Y)$ and $s = \sqrt{Var(Y)}$), where $u \in \mathbb{R}$ and $s \in \mathbb{R}^+$, $a_0 = (a-u)/s$, $b_0 = (b-u)/s$, $f_X(x) = s\, f_Y(u+s\,x)$ denote the density function of X whose support is the interval (a_0, b_0), $E(X^j) = E[((Y-u)/s)^j] \equiv \mu_X(j)$, and let the base density function $\psi_X(x) \equiv c_T\, w(x)$, where c_T is a positive normalizing constant, be an initial density approximation to $f_X(x)$ with $\int_{a_0}^{b_0} x^j\, \psi_X(x)\, \mathrm{d}x \equiv m_X(j)$. Assuming that the sequence $\mu_X(i)$, $i = 0, 1, 2, \ldots$, uniquely defines the distribution of X, that $m_X(j)$ exists for $j = 0, 1, \ldots, 2\,n$, and that whenever $\psi_X(x)$ is nontrivial function of x, its tail behavior is congruent to that of $f_X(x)$, the density function of X can be approximated by*

$$f_{X_n}(x) = \psi_X(x) \sum_{\ell=0}^{n} \xi_\ell\, x^\ell \qquad (1.1)$$

with $(\xi_0, \ldots, \xi_n)' = M^{-1}(\mu_X(0), \ldots, \mu_X(n))'$, where M is an $(n+1)\times(n+1)$ matrix whose $(h+1)^{th}$ row is $m_X(h), \ldots, m_X(h+n)$, $h = 0, 1, \ldots, n$. When $\psi_X(x)$ depends on r parameters, these are determined by equating $m_X(j)$ to $\mu_X(j)$, $j = 1, \ldots, r$. The corresponding density approximant for Y is then

$$f_{Y_n}(y) = \psi_X\left(\frac{y-u}{s}\right)\sum_{\ell=0}^{n}\frac{\xi_\ell}{s}\left(\frac{y-u}{s}\right)^{\ell}. \tag{1.2}$$

Note that the base density is selected by applying some goodness-of-fit tests, such as the Kolmogorov-Smirnov and chi-square tests to certain density functions suggested by a histogram of the data or the exact density when it is known.

The other primary technique which is summarized in Result 2 yields what is referred to as differentiated logdensity approximants (DLA's):

Result 2 *Let*

$$f_{\nu,\delta}(x) = k\, e^{\int_\alpha^x p_{\nu,\delta}(y)\,\mathrm{d}y} \tag{1.3}$$

be the approximation of a density function defined on the interval (α, β) *where*

$$p_{\nu,\delta}(x) = \frac{\sum_{i=0}^{\nu} a_i\, x^i}{\sum_{j=0}^{\delta} c_j\, x^j} \equiv \frac{N(x)}{D(x)}, \tag{1.4}$$

with $c_\delta = 1$*, k being a positive constant such that* $f_{\nu,\delta}(\cdot)$ *integrates to one over the interval* (α, β)*. The function* $p_{\nu,\delta}(x)$ *is a rational function of orders* ν *and* δ *and the resulting approximation, as a differentiated logdensity approximant or DLA.*

As shown below, the coefficients a_i, $i = 0, 1, \ldots, \nu$, and c_j, $j = 0, 1, \ldots, \delta - 1$, can be determined by solving a system of linear equations. On differentiating $\ln(f_{\nu,\delta}(x))$, one obtains

$$\frac{\mathrm{d}}{\mathrm{d}x}\ln(f_{\nu,\delta}(x)) = \frac{f'_{\nu,\delta}(x)}{f_{\nu,\delta}(x)} = p_{\nu,\delta}(x), \tag{1.5}$$

which yields the relationship,

$$f'_{\nu,\delta}(x)\sum_{j=0}^{\delta} c_j\, x^j = f_{\nu,\delta}(x)\sum_{i=0}^{\nu} a_i\, x^i. \tag{1.6}$$

Then, on multiplying both sides of this equality by x^h and integrating from α to β, one has

$$\int_{\alpha}^{\beta} f'_{\nu,\delta}(x) \sum_{j=0}^{\delta} c_j\, x^{j+h}\, \mathrm{d}x = \int_{\alpha}^{\beta} f_{\nu,\delta}(x) \sum_{i=0}^{\nu} a_i\, x^{i+h}\, \mathrm{d}x,\ h = 0, 1, \ldots, \nu + \delta, \tag{1.7}$$

which, on integrating the left-hand side by parts, yields

$$\begin{aligned} &f_{\nu,\delta}(x) \sum_{j=0}^{\delta} c_j\, x^{j+h}\Big|_{\alpha}^{\beta} - \sum_{j=0}^{\delta} c_j\,(j+h) \int_{\alpha}^{\beta} x^{j+h-1} f_{\nu,\delta}(x)\, \mathrm{d}x \\ &= \sum_{i=0}^{\nu} a_i \int_{\alpha}^{\beta} x^{i+h} f_{\nu,\delta}(x)\, \mathrm{d}x, \quad h = 0, 1, \ldots, \nu + \delta, \end{aligned} \tag{1.8}$$

where

$$f_{\nu,\delta}(x) \sum_{j=0}^{\delta} c_j\, x^{j+h}\Big|_{\alpha}^{\beta} = \sum_{j=0}^{\delta} c_j\,(f_{\nu,\delta}(\beta)\,\beta^{j+h} - f_{\nu,\delta}(\alpha)\,\alpha^{j+h}). \tag{1.9}$$

Thus, letting μ_h, $h = 0, 1, \ldots, \nu + \delta$, denote the h^{th} moment of the approximate distribution specified by $f_{\nu,\delta}(x)$, one has

$$\sum_{j=0}^{\delta} c_j\,(f_{\nu,\delta}(\beta)\,\beta^{j+h} - f_{\nu,\delta}(\alpha)\,\alpha^{j+h} - (j+h)\,\mu_{j+h-1}) = \sum_{i=0}^{\nu} a_i\,\mu_{i+h}, \tag{1.10}$$

for $h = 0, 1, \ldots, \delta + \nu$, with $\mu_0 = 1$. Now, on replacing $f_{\nu,\delta}(\alpha)$, $f_{\nu,\delta}(\beta)$ and μ_h by $f(\alpha)$, $f(\beta)$ and $\mu_X(h)$, $h = 0, 1, \ldots, \delta + \nu$, respectively, one obtains the recursive relationship,

$$\sum_{j=0}^{\delta} c_j\,(f(\beta)\,\beta^{j+h} - f(\alpha)\,\alpha^{j+h} - (j+h)\,\mu_X(j+h-1)) = \sum_{i=0}^{\nu} a_i\,\mu_X(i+h), \tag{1.11}$$

for $h = 0, 1, \ldots, \delta + \nu$.

The coefficients c_i, $i = 0, 1, \ldots, \delta - 1$, and a_i, $i = 0, 1, \ldots, \nu$, can be determined by solving these $\nu + \delta + 1$ linear equations. Note that when j and h are both equal to zero, the value of $\mu_X(-1)$ is not required since its coefficient happens to be zero.

The number of moments that are required for given values of ν and δ can be determined from the above linear system. Whenever $\nu \geq \delta$, as is often the case, $2\nu+\delta$ moments are needed, whereas $2\delta + \nu + 1$ moments are required when $\delta > \nu$.

Once the coefficients have been determined, the differential equation,

$$f'_{\nu,\delta}(x) = p_{\nu,\delta}(x) f_{\nu,\delta}(x), \tag{1.12}$$

can be solved by making use of symbolic computational packages such as *Mathematica* or Maple. The resulting density approximant is denoted DLA(ν, δ).

This thesis is mainly concerned with methodologies that rely on moments. This is justified by Result 3 relating a sample to its moments, which was established in Zareamoghaddam (2018). There are instances in multivariate statistical analysis where the exact distribution of a test statistic is unknown whereas its exact moments can be determined. Thus the proposed methodologies are not only useful for modeling purpose but also for making statistical inference.

Result 3 *A set of n observations is uniquely determined by the first n associated sample moments.*

Proof Let $S = \{x_1, x_2, \ldots, x_n\}$, $M = \{m_1, m_2, \ldots, m_n\}$ and $m_h = \sum_{i=1}^{n} x_i^h / n$. According to the fundamental theorem of algebra, $p(z) = a_0 + a_1\, z + \cdots + a_{n-1}\, z^{n-1} + z^n$ is uniquely specified by its n roots x_i's for $i = 1, 2, \ldots, n$.

Moreover, given S, the coefficients of $p(x)$ can be expressed in terms of the sequence of moments M via the Newton-Girard identity. Accordingly, a given polynomial of degree n, say $p(x)$, can be represented as follows:

$$\prod_{i=1}^{n} (x - x_i) = \sum_{k=0}^{n} (-1)^{n-k} e_{n-k}\, x^k, \tag{1.13}$$

where $e_0 = 1$ and

$$e_\ell = \frac{n}{\ell} \sum_{j=1}^{\ell} (-1)^{j-1} e_{\ell-j}\, m_j, \quad \ell = 1, \ldots, n. \tag{1.14}$$

Thus, given the first n sample moments associated with S, a sample of size n, one can determine the right hand side of Equation 1.13, whose roots are precisely $x_1, x_2, ..., x_n$. This establishes that S is uniquely specified by M.

We note that moment-based density estimation techniques are ideally suited for modeling massive data sets: Once the moments have been evaluated, which is easily achieved even for extremely large data sets, the determination of the estimated density function does not depend on the sample size. Moreover, once a new set of observations, $X_{n_1+1}, \ldots, X_n$, becomes available in addition to an initial data set, $X_1, \ldots, X_{n_1}$, there is no need to make use of each of the n_1 original data points since the h^{th} updated moment will then be

$$(n_1 m_h + \sum_{i=n_1+1}^{n} x_i^h)/n$$

where m_h denotes the h^{th} sample moment evaluated from the initial data set.

Chapter 2 and 3 propose several types of moment-based density approximants and estimates, respectively. In Chapter 4, Monte Carlo samples are compared to quasi-Monte Carlo samples with respect to their distributional representativeness. An easy-to-apply methodology is proposed in Chapter 5 for classifying the tail behavior of probability laws. An approximation to the distribution of quadratic forms in various types of discrete and continuous random variables, which is based on a symbolic expansion of their associated moments, is provided in Chapter 6. Some concluding remarks and an outline of further possible developments in the area are included in the last chapter. As much effort has been expended on creating the code for implementing the results presented in this thesis, it is included in the Appendix for the benefit of the readers.

Bibliography

[1] Devroye, L. and Gyorfi, L. (1985). *Nonparametric Density Estimation: The L_1 View*. John Wiley & Sons, New York.

[2] Efromovich, S. (1999). *Nonparametric Curve Estimation: Methods, Theory and Applications*. Springer Science & Business Media, New York.

[3] Eggermont, P. P. B., and LaRiccia, V. N. (2001). *Maximum Penalized Likelihood Estimation: Volume I: Density Estimation.* Springer Science & Business Media, New York.

[4] Elderton, W. P. and Johnson, N. L. (1969). *Systems of Frequency Curves*. Cambridge University Press, Oxford.

[5] Mathai, A. M. and Saxena, R. K. (1978). *The H-function with Applications in Statistics and Other Disciplines*. John Wiley & Sons, New York.

[6] Provost, S. B. and Rudiuk, E. M. (1995). Moments and densities of test statistics for covariance structures. *International Journal of Mathematical and Statistical Sciences* **4**(1), 85–104.

[7] Provost, S. B. (2005). Moment-based density approximants. *The Mathematica Journal* **9**, 727-756.

[8] Reid, N. (1988). Saddlepoint methods and statistical inference. *Statistical Science* **3**(2), 213–238.

[9] Rose, C. and Smith, M. D. (2002), *Mathematical Statistics with Mathematica*. Springer-Verlag, New York.

[10] Scott, D. W. (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization, 2nd Edition*. John Wiley & Sons, New York.

[11] Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis Series: Monographs on Statistics and Applied Probability*. Chapman and Hall, London.

[12] Solomon, H. and Stephens, M. (1978). Approximations to density functions using Pearson curves. *Journal of the American Statistical Association* **73**, 153–160.

[13] Zareamoghaddam, H. (2018). Advances in semi-nonparametric density estimation and shrinkage regression. *Electronic Thesis and Dissertation Repository. 5234.* https://ir.lib.uwo.ca/etd/5234

Chapter 2

On Comparing Various Types of Moment-Based Density Approximants

2.1 Introduction

It is often the case that the exact moments of a statistic of the continuous type can be explicitly determined, while its density function either does not lend itself to numerical evaluation or proves to be mathematically intractable. This is the case for instance for quadratic form in random variables. Several approaches are discussed in this chapter with a view to determining density approximants that are based on the exact moments of the distributions at hand. The first, second and fifth ones are known and compared to four related adjustments as to their accuracy.

The approximations discussed in this chapter are expressed in terms of a base density that provides an initial approximation and an adjustment consisting of a polynomial or a function thereof. The polynomials coefficients are determined by equating the first n moments of the exact distribution to those of the approximant. For comparison purposes, all the distributions are mapped onto the interval $(0, 1)$.

For distributions whose support is the interval $(0, 1)$, we can apply the approaches directly.

For other distributions, we perform an appropriate transformation of variables before implementing these methodologies. After determining the approximant of the transformed distribution, we apply the inverse transformation in order to obtain an approximant for the original distribution.

If the random variable Y follows a distribution on a bounded interval (a, b), $-\infty < a < b < \infty$, let $X = (Y - a)/(b - a)$. This random variable transformation maps $Y \in (a, b)$ to $X \in (0, 1)$. After determining the approximant to the probability density function of X, we apply the inverse transformation $Y = (b - a)X + a$. If the random variable Y follows a distribution on a left-bounded interval (a, ∞), $a > -\infty$, we can apply the transformation $X = 1-1/(Y-a+1)$ and the inverse transformation $Y = 1/(1-X)+a-1$. If random variable Y follows a distribution on a right-bounded interval $(-\infty, b)$, $b < \infty$, we can apply the transformation $X = 1-1/(b-Y+1)$ and the inverse transformation $Y = b + 1 - 1/(1 - X)$. If the random variable Y has the real line $(-\infty, \infty)$ as its support, we apply the transformation $X = \dfrac{\arctan Y}{\pi} + \dfrac{1}{2}$, the inverse transformation being $Y = \tan[\pi(X - 1/2)]$.

2.2 Seven types of moment-based density approximants

2.2.1 Approximants expressed as the product of a base density and a polynomial (Type 1)

Let the density function and the integer moment of order h of a random variable X defined on the interval $(0, 1)$ be respectively denoted by $f(x)$ and $\mu_X(h) = E(X^h)$. Let the approximant be

$$f_1(x) = b(x)\, p_1(x), \tag{2.2.1}$$

where $b(x)$ is the base density function and $p_1(x)$ is the polynomial function that adjusts the base density function.

We can utilize the pdf of a beta(α, β) distribution as $b(x)$, where

$$\alpha = \mu_X(1)\frac{\mu_X(1) - \mu_X(2)}{\mu_X(2) - \mu_X(1)^2} \tag{2.2.2}$$

and

$$\beta = \frac{(1 - \mu_X(1))\,\alpha}{\mu_X(1)}. \tag{2.2.3}$$

This beta distribution is such that its first two moments are identical to those of X.

Let the integer moment of order h of this beta distribution be denoted by $\mu_b(h)$, and let

$$p_1(x) = \sum_{i=0}^{n} a_i\, x^i. \tag{2.2.4}$$

Then, on setting the first n moments of $f(x)$ and $f_1(x)$ to be equal, one has

$$\begin{aligned} \mu_X(h) &= \int_0^1 x^h b(x) p_1(x)\,\mathrm{d}x \\ &= \int_0^1 x^h b(x) \sum_{i=0}^{n} a_i x^i\,\mathrm{d}x \\ &= \sum_{i=0}^{n} a_i \int_0^1 x^{i+h} b(x)\,\mathrm{d}x \\ &= \sum_{i=0}^{n} a_i\,\mu_b(i+h), \quad h = 0, 1, \ldots, n, \end{aligned} \tag{2.2.5}$$

or in matrix form,

$$\begin{bmatrix} \mu_b(0) & \mu_b(1) & \cdots & \mu_b(n) \\ \mu_b(1) & \mu_b(2) & \cdots & \mu_b(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ \mu_b(n) & \mu_b(n+1) & \cdots & \mu_b(2n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \mu_X(0) \\ \mu_X(1) \\ \vdots \\ \mu_X(n) \end{bmatrix}. \tag{2.2.6}$$

The coefficients a_i are obtained by solving this system. This approach which was introduced in Provost (2005) is stated in more generality in the Introduction as Result 1.

2.2.2 Approximants expressed as polynomials (Type 2)

Let the approximant be

$$f_2(x) = p_2(x), \tag{2.2.7}$$

where

$$p_2(x) = \sum_{i=0}^{n} a_i\, x^i. \tag{2.2.8}$$

Such approximation can be viewed as a special case of the type 1 approximants wherein $b(x) = 1$ (the pdf of a Uniform(0,1) distribution).

Then, on setting the first n moments of $f(x)$ and $f_2(x)$ to be equal, one has

$$\begin{aligned} \mu_X(h) &= \sum_{i=0}^{n} a_i \int_0^1 x^{i+h}\,\mathrm{d}x \\ &= \sum_{i=0}^{n} a_i \frac{1}{i+h+1}, \quad h = 0, 1, \ldots, n, \end{aligned} \tag{2.2.9}$$

or in matrix form,

$$\begin{bmatrix} 1 & 1/2 & \cdots & 1/n \\ 1/2 & 1/3 & \cdots & 1/(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/(n+1) & \cdots & 1/(2n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \mu_X(0) \\ \mu_X(1) \\ \vdots \\ \mu_X(n) \end{bmatrix}. \tag{2.2.10}$$

The coefficients a_i are obtained by solving this system.

As mentioned at the beginning of this chapter, when the support of the random variable Y is a bounded interval (a, b), we can use the transformation $X = (Y - a)/(b - a)$ before applying this approach. Once the approximant of the transformed distribution is obtained, we apply the inverse transformation $Y = (b - a)X + a$. Note that, in this case, we can also use this approach directly without resorting to a transformation.

Let the density function and integer moment of order h of the random variable Y defined on the interval (a, b) be respectively denoted by $g(y)$ and $\mu_Y(h) = E(Y^h)$. Let the approximant be

$$g_2(y) = p_2(y). \tag{2.2.11}$$

Then, setting the first n moments of $g(y)$ and $g_2(y)$ to be equal yields

$$\begin{aligned}\mu_Y(h) &= \sum_{i=0}^{n} a_i \int_a^b y^{i+h}\,\mathrm{d}y \\ &= \sum_{i=0}^{n} a_i \frac{b^{i+h+1} - a^{i+h+1}}{i+h+1}, \quad h = 0, 1, \ldots, n,\end{aligned} \tag{2.2.12}$$

or in matrix form,

$$\begin{bmatrix} b-a & \frac{b^2-a^2}{2} & \cdots & \frac{b^{n+1}-a^{n+1}}{n+1} \\ \vdots & & \ddots & \vdots \\ \frac{b^{n+1}-a^{n+1}}{n+1} & \frac{b^{n+2}-a^{n+2}}{n+2} & \cdots & \frac{b^{2n+1}-a^{2n+1}}{2n+1} \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \mu_X(0) \\ \vdots \\ \mu_X(n) \end{bmatrix}. \tag{2.2.13}$$

The coefficients a_i are obtained by solving this system. Clearly, type 2 approximants require a finite support.

2.2.3 Approximants expressed as the sum of a base density and a polynomial (Type 3)

Let the approximant be of the form,

$$f_3(x) = b(x) + p_3(x), \tag{2.2.14}$$

where $b(x)$ is the base density function and $p_3(x)$ is a polynomial adjustment that integrates to zero on the interval $(0, 1)$.

We can utilize the pdf of a beta(α, β) distribution as $b(x)$, where

$$\alpha = \mu_X(1)\frac{\mu_X(1) - \mu_X(2)}{\mu_X(2) - \mu_X(1)^2} \tag{2.2.15}$$

and

$$\beta = \frac{(1 - \mu_X(1))\,\alpha}{\mu_X(1)}. \tag{2.2.16}$$

This beta distribution is such that its first two moments are identical to those of X.

Let

$$p_3(x) = \sum_{i=0}^{n} a_i\, x^i. \tag{2.2.17}$$

Then $f_3(x) - b(x) = p_3(x)$, and $f(x) - b(x)$ can be treated as a function to which type 2 approximants apply.

On setting the first n 'moments' of $f(x) - b(x)$ and $p_3(x)$ to be equal, one has

$$\mu_X(h) - \mu_b(h) = \sum_{i=0}^{n} a_i\,\frac{1}{i+h+1}, \quad h = 0, 1, \ldots, n, \tag{2.2.18}$$

or in matrix form,

$$\begin{bmatrix} 1 & 1/2 & \cdots & 1/n \\ 1/2 & 1/3 & \cdots & 1/(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/(n+1) & \cdots & 1/(2n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \mu_X(0) - \mu_b(0) \\ \mu_X(1) - \mu_b(1) \\ \vdots \\ \mu_X(n) - \mu_b(n) \end{bmatrix}. \tag{2.2.19}$$

The coefficients a_i are obtained by solving this system.

2.2.4 Approximants expressed as the product of a base density and an exponentiated polynomial (Type 4)

Let the approximant be of the form

$$f_4(x) = b(x)\, e^{p_4(x)}, \tag{2.2.20}$$

where

$$p_4(x) = \sum_{i=0}^{n} a_i\, x^i. \tag{2.2.21}$$

Then $\log f_4(x) - \log b(x) = p_4(x)$, and $\log f(x) - \log b(x)$ can be treated as a function to which type 2 approximation apply.

Let

$$\mu_4(h) = \int_0^1 x^h\, (\log f(x) - \log b(x))\, \mathrm{d}x. \tag{2.2.22}$$

Note that this approach requires that $f(x)$ be known or that a preliminary estimate thereof is available. On setting the first n 'moments' of $\log f(x) - \log b(x)$ and $p_4(x)$ to be equal, one has

$$\mu_4(h) = \sum_{i=0}^{n} a_i\, \frac{1}{i+h+1}, \quad h = 0, 1, \ldots, n, \tag{2.2.23}$$

or in matrix form,

$$\begin{bmatrix} 1 & 1/2 & \cdots & 1/n \\ 1/2 & 1/3 & \cdots & 1/(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/(n+1) & \cdots & 1/(2n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \mu_4(0) \\ \mu_4(1) \\ \vdots \\ \mu_4(n) \end{bmatrix}. \tag{2.2.24}$$

The coefficients a_i's are obtained by solving this system.

2.2.5 Differentiated logdensity approximants (DLA) (Type 5)

Let

$$f_5(x) = k\, e^{\int_\alpha^x p_{\nu,\delta}(y)\,\mathrm{d}y}, \tag{2.2.25}$$

where

$$p_{\nu,\delta}(x) = \frac{\sum_{i=0}^{\nu} a_i\, x^i}{\sum_{j=0}^{\delta} c_j\, x^j} \equiv \frac{N(x)}{D(x)}, \tag{2.2.26}$$

with $c_\delta = 1$, k being a positive constant such that $f_5(\cdot)$ integrates to one over the interval delimited by the end points of the support, (α, β). The function $p_{\nu,\delta}(x)$ will be referred to as a rational function of orders ν and δ, and the resulting approximation, as a differentiated logdensity approximant or DLA, whose solution is provided in the Introduction.

Referring to Result 2, when $f_5(\alpha) = f_5(\beta) = 0$, as is often the case, one has

$$\sum_{j=0}^{\delta} c_j\,(-(j+h)\,\mu_X(j+h-1)) = \sum_{i=0}^{\nu} a_i\,\mu_X(i+h). \tag{2.2.27}$$

If δ equals to 0,

$$\sum_{i=0}^{\nu} a_i\,\mu_X(i+h) = -h\,\mu_X(h-1), \tag{2.2.28}$$

which is equivalent to

$$\begin{bmatrix} \mu_X(0) & \cdots & \mu_X(\nu) \\ \mu_X(1) & \cdots & \mu_X(\nu+1) \\ \vdots & \ddots & \vdots \\ \mu_X(\nu) & \cdots & \mu_X(2\nu) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_\nu \end{bmatrix} = \begin{bmatrix} 0 \\ -\mu_X(0) \\ \vdots \\ -\nu\,\mu_X(\nu-1) \end{bmatrix} \tag{2.2.29}$$

in matrix form.

The coefficients a_i's are obtained by solving this system. When the differentiated logdensities being considered are expressed as polynomials rather than rational functions, they will be referred to as polynomial differentiated logdensity approximants or PDLA. The system has then a unique solution, and theoretically, more accurate density approximants can be obtained by increasing the value of ν.

2.2.6 Approximants expressed in terms of a base density and a DLA (Type 6)

Let the approximant be of the form

$$f_6(x) = b(x)\, e^{\int_\alpha^x p_{\nu,\delta}(y)\,\mathrm{d}y}, \tag{2.2.30}$$

where

$$p_{\nu,\delta}(x) = \frac{\sum_{i=0}^{\nu} a_i\, x^i}{\sum_{j=0}^{\delta} c_j\, x^j} \equiv \frac{N(x)}{D(x)}. \tag{2.2.31}$$

Let

$$f_6^*(x) = \frac{f(x)}{b(x)} = e^{\int_\alpha^x p_{\nu,\delta}(y)\,\mathrm{d}y} \tag{2.2.32}$$

and treat $f_6^*(x)$ as our original function. Then, on making use of a type 5 approximant, one has

$$\sum_{j=0}^{\delta} c_j\,(f(\beta)\,\beta^{j+h} - f(\alpha)\,\alpha^{j+h} - (j+h)\,\mu_6(j+h-1)) = \sum_{i=0}^{\nu} a_i\,\mu_6(i+h), \tag{2.2.33}$$

for $h = 0, 1, \ldots, \delta+\nu$, with

$$\mu_6(h) = \int_0^1 x^h\, \frac{f(x)}{b(x)}\,\mathrm{d}x. \tag{2.2.34}$$

Note that this approach requires that $f(x)$ be known or that a preliminary estimate thereof such as a kernel density estimate be available.

When $f(0) = f(1) = 0$, as is often the case, one has

$$\sum_{j=0}^{\delta} c_j\,(-(j+h)\,\mu_6(j+h-1)) = \sum_{i=0}^{\nu} a_i\,\mu_6(i+h). \tag{2.2.35}$$

If δ equals to 0,

$$\sum_{i=0}^{\nu} a_i\, \mu_6(i+h) = -h\, \mu_6(h-1), \quad h = 0, 1, \ldots, \nu. \tag{2.2.36}$$

The coefficients a_i's are obtained by solving this system.

2.2.7 Approximants expressed as a mixture of a base density and a DLA (Type 7)

Let the approximant be the following mixture:

$$f_7(x) = w\, b(x) + (1-w)\, e^{\int_\alpha^x p_{\nu,\delta}(y)\,\mathrm{d}y}, \tag{2.2.37}$$

where $0 < w < 1$ and

$$p_{\nu,\delta}(x) = \frac{\sum_{i=0}^{\nu} a_i\, x^i}{\sum_{j=0}^{\delta} c_j\, x^j} \equiv \frac{N(x)}{D(x)}. \tag{2.2.38}$$

Let

$$f_7^*(x) = \frac{f_7(x) - w\, b(x)}{1-w} = e^{\int_\alpha^x p_{\nu,\delta}(y)\,\mathrm{d}y}. \tag{2.2.39}$$

On treating $f_7^*(x)$ as our original function and making use of a type 5 approximant, one has

$$\sum_{j=0}^{\delta} c_j\, (f(\beta)\, \beta^{j+h} - f(\alpha)\, \alpha^{j+h} - (j+h)\, \mu_7(j+h-1)) = \sum_{i=0}^{\nu} a_i\, \mu_7(i+h), \tag{2.2.40}$$

for $h = 0, 1, \ldots, \delta + \nu$, with

$$\mu_7(h) = \frac{1}{1-w}\, (\mu_X(h) - w\, \mu_b(h)). \tag{2.2.41}$$

When $f(0) = f(1) = 0$, as is often the case, one has

$$\sum_{j=0}^{\delta} c_j\, (-(j+h)\, \mu_7(j+h-1)) = \sum_{i=0}^{\nu} a_i\, \mu_7(i+h). \tag{2.2.42}$$

and if δ equals to 0,

$$\sum_{i=0}^{\nu} a_i\, \mu_7(i+h) = -h\, \mu_7(h-1), \quad h = 0, 1, \ldots, \nu. \tag{2.2.43}$$

The coefficients a_i's are obtained by solving this system.

Not that the cdf obtained from $f_i(x)$ will be denoted $F_i(x)$, $i = 1, 2, \ldots, 7$.

2.3 Obtaining a bona fide density function

The density approximants should be bona fide, that is, they should be non-negative and integrate to one. Another desirable property is that they be smooth (that is, differentiable everywhere) functions. We can always normalize a function so that it integrates to one.

When a polynomial adjustment is applied, the resulting function can occasionally be negative on subranges of the support. When this occurs, we can then define the negative part(s) to be zero, normalize the resulting function and then denote it by $f^*(x)$. As a final step, we may apply the DLA $(\nu, 0)$ approximation methodology in conjunction with the exact moments of $f^*(x)$. The resulting density function is smooth and bona fide on the support.

Alternatively, we could apply the correction algorithms proposed by Gajek (1986) or Glad *et al.* (2003) in order to obtain legitimate pdf's.

2.4 Application of the methodologies

2.4.1 Mixture of beta pdf's

Consider an equally weighted mixture of two beta density functions with parameters (8, 12) and (3, 15), to which the seven proposed types of approximants are applied. Mixtures of densities are utilized in a related context in Lindsay *et al.* (2000). Our objective is to determine which methodology provides the best approximation. As well, we wish to assess which one is more accurate based on a given number of moments. If two methodologies can provide approximations that are comparable in terms of accuracy, we select the one requiring fewer moments according to the principle of parsimony.

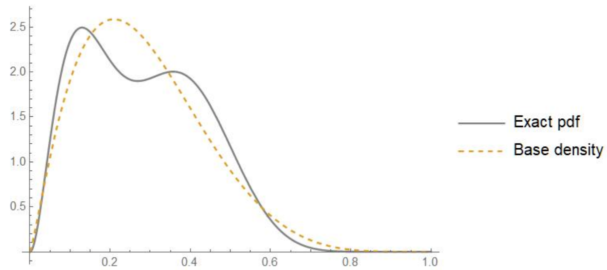

Figure 2.1: Exact PDF and base density

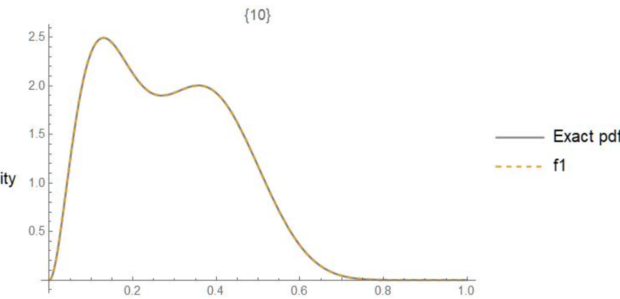

Figure 2.2: $f1$, n=10

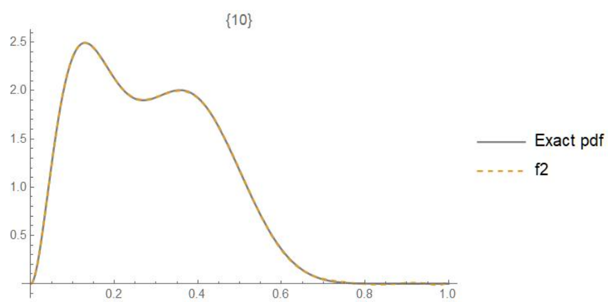

Figure 2.3: $f2$, n=10

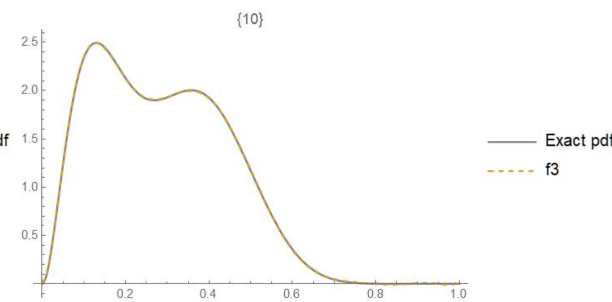

Figure 2.4: $f3$, n=10

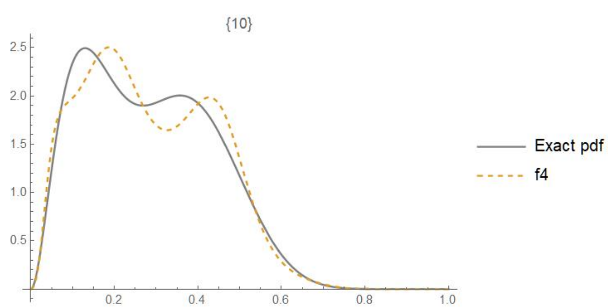

Figure 2.5: $f4$, n=10

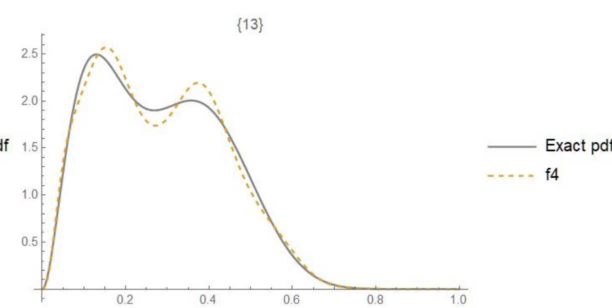

Figure 2.6: $f4$, n=13, best approximation

It follows from the figures and Table 2.1 which includes the integrated squared differences (ISD) between the exact and approximated pdf's and the number of moments required to obtain the approximate pdf's, that types 1, 2 and 3 approximants can provide quite accurate approximants with only 10 moments. The type 5 approximants can also yield reasonably close

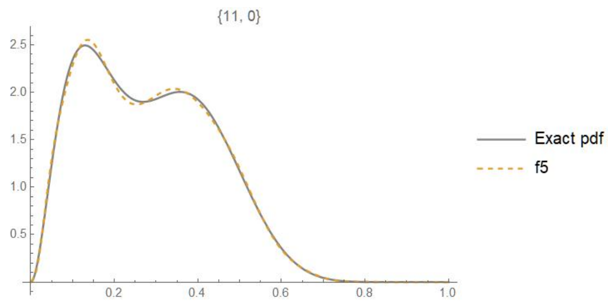

Figure 2.7: $f5$, v=11, δ=0

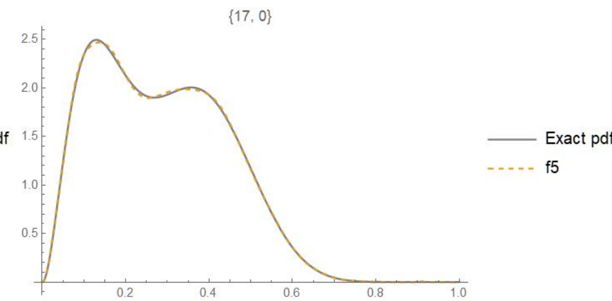

Figure 2.8: $f5$, v=17, δ=0, best approximation

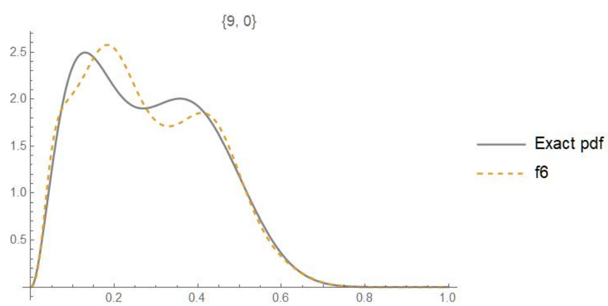

Figure 2.9: $f6$, v=9

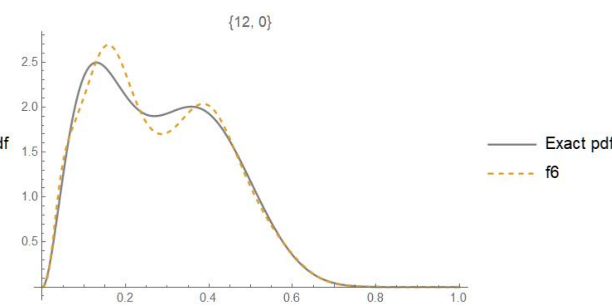

Figure 2.10: $f6$, v=12, δ=0, best approximation

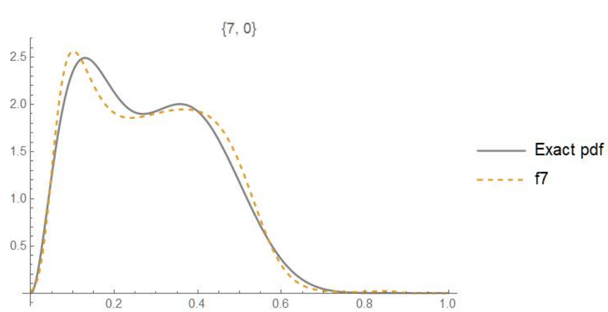

Figure 2.11: $f7$, w=0.3, v=7, $\delta = 0$

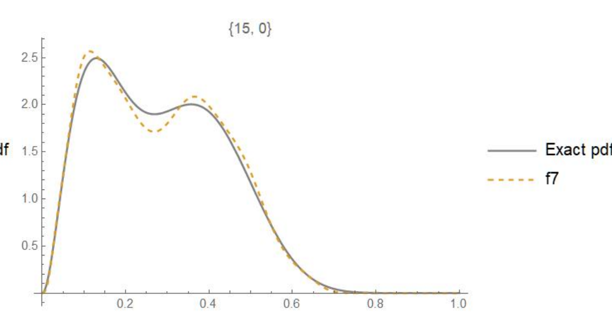

Figure 2.12: $f7$, w=0.2, v=15, $\delta = 0$

Table 2.1: Comparison of the ISD's for different types and degrees

Type	Degree	# of moments	ISD	Degree	# of moments	ISD
1	10	10	**0.0000023**[1]			
2	10	10	0.0000281			
3	10	10	0.0000336			
4	10	10	0.0295788	13	13	0.0081179
5	11, 0	22	0.0008106	17, 0	34	0.0001887
6	9, 0	18	0.0215733	12, 0	24	0.0011875
7	5, 0	10	0.0133151	15, 0	30	0.0048238

approximations but require more moments than types 1,2 and 3. As for types 4, 6 and 7, the resulting approximations are not as accurate.

2.4.2 Mixture of gamma pdf's

Consider an equally weighted mixture of two gamma density functions with parameters (2, 2) and (9, 1) whose support is the positive half-line. We rescale the distribution using the transformation

$$Y = \frac{X}{\sigma},$$

where σ is the standard deviation of the mixture. On applying the transformation

$$Z = \frac{Y}{Y+1},$$

we obtain a random variable whose support is (0,1). We then apply the proposed methodologies and compare the accuracy of the approximants.

It follows from the figures and Table 2.2 which includes the integrated squared differences between the exact and approximated pdf's, that in this case, the type 5 approximants provide the most accurate approximations. Types 1, 2, 3 and 7 can also yield reasonably close approximations. However, type 7 approximants require more moments than types 2 and 3 and the resulting approximations are not as accurate.

[1]The bold-face numbers appearing in the tables correspond to the smallest integrated squared difference and thus the most accurate approximant.

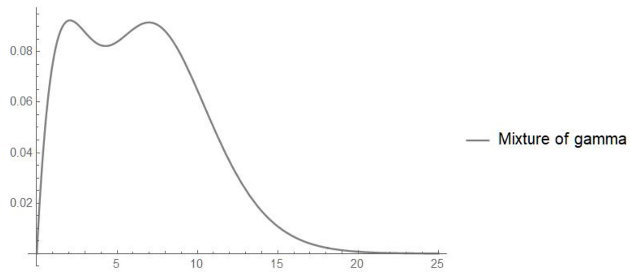

Figure 2.13: Exact density

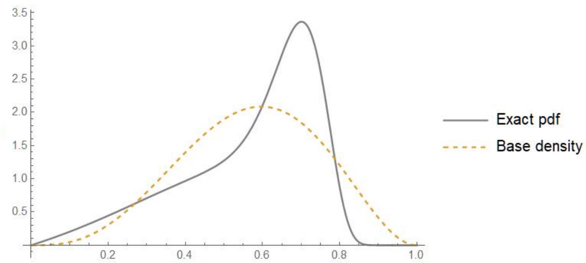

Figure 2.14: Transformed PDF's

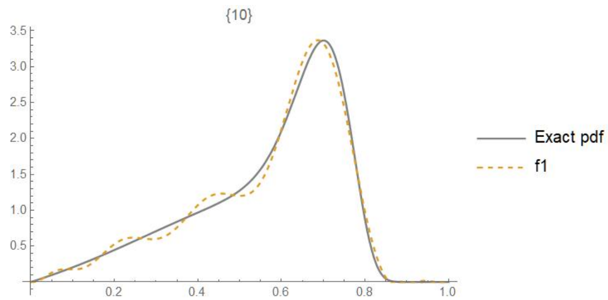

Figure 2.15: $f1$, n=10

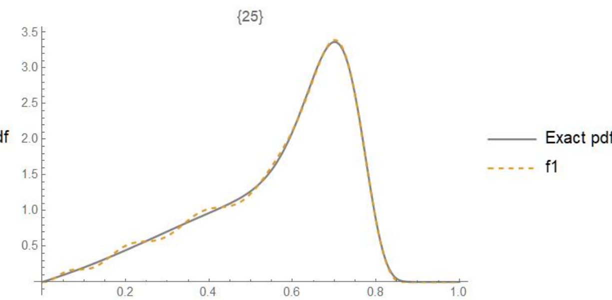

Figure 2.16: $f1$, n=25, best approximation

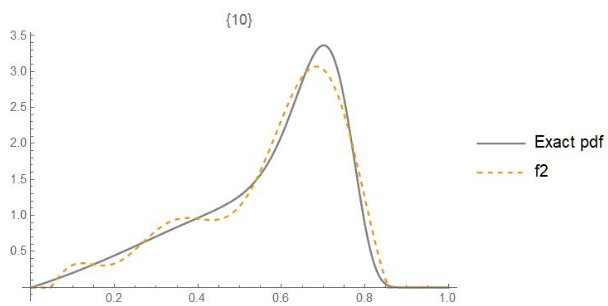

Figure 2.17: $f2$, n=10

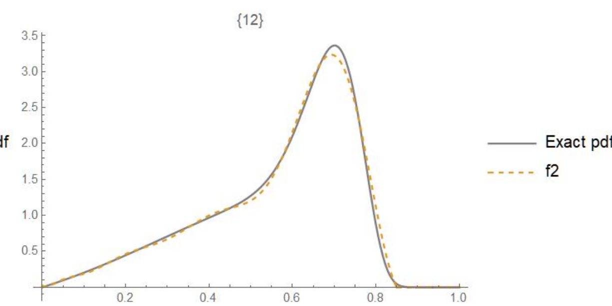

Figure 2.18: $f2$, n=12, best approximation

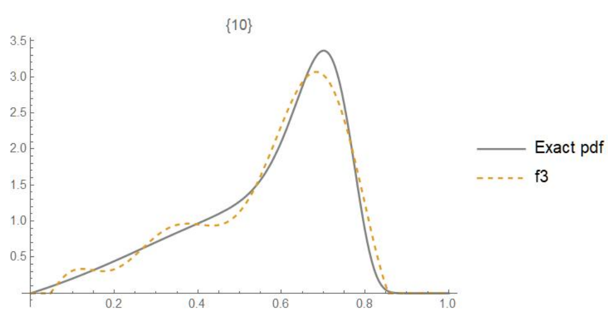

Figure 2.19: $f3$, n=10

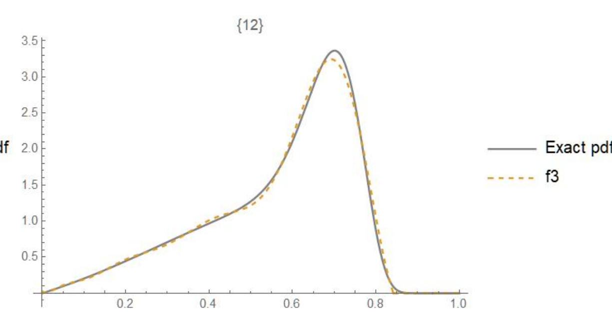

Figure 2.20: $f3$, n=12, best approximation

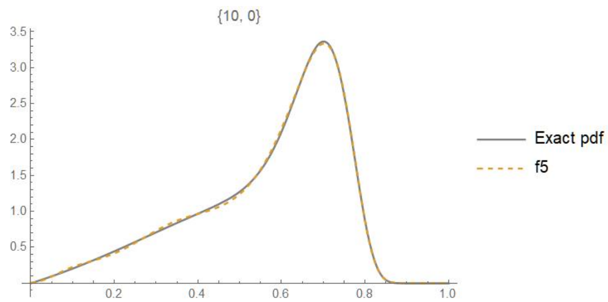

Figure 2.21: $f5$, ν=10, $\delta = 0$

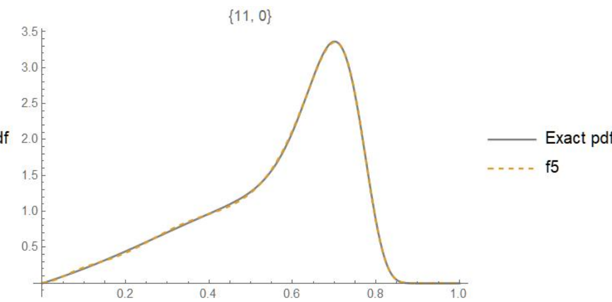

Figure 2.22: $f5$, ν=11, $\delta = 0$, best approximation

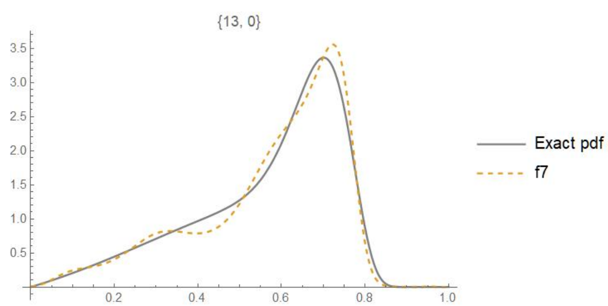

Figure 2.23: $f7$, $w = 0.1$, ν=13, $\delta = 0$

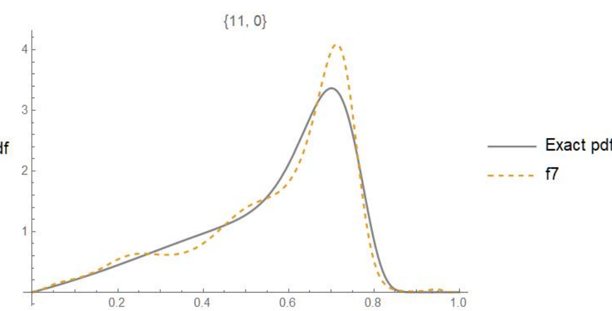

Figure 2.24: $f7$, $w = 0.2$, ν=11, $\delta = 0$

Table 2.2: Comparison of the ISD's for different types and degrees

Type	Degree	# of moments	ISD	Degree	# of moments	ISD
1	10	10	0.0093	25	25	0.0011
2	10	10	0.0286	12	12	0.0047
3	10	10	0.0223	12	12	0.0036
5	10, 0	20	0.0004	11, 0	22	**0.0002**
7	11, 0	22	0.0451	13, 0	26	0.0204

We can also obtain density approximants by truncating the distribution. Since $F_X(25) = 0.999937$, which is nearly equal to one, we may focus on the interval $(0, 25)$. On applying a linear transformation that maps the random variable X from $(0, 25)$ to $(0, 1)$ as well as the proposed methodologies, we obtained the results that follow.

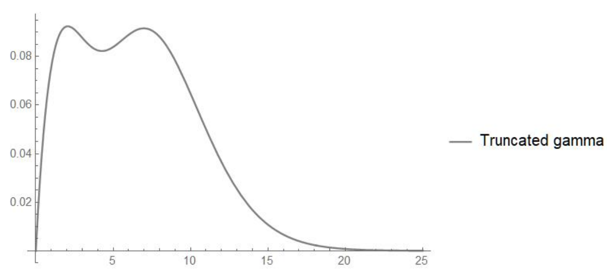

Figure 2.25: Truncated distribution

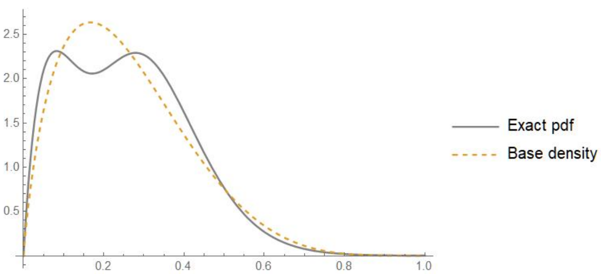

Figure 2.26: Exact PDF and base density

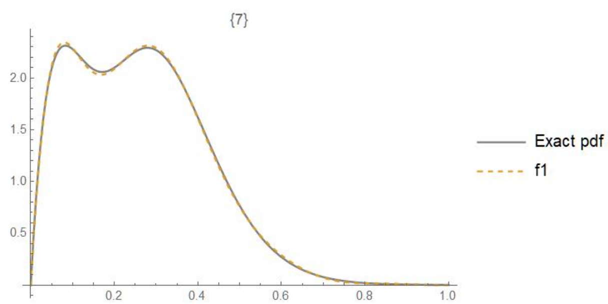

Figure 2.27: $f1$, n=7

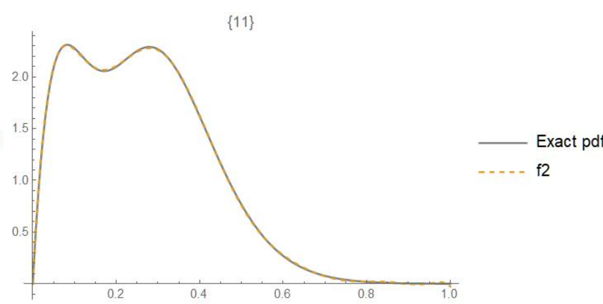

Figure 2.28: $f2$, n=11

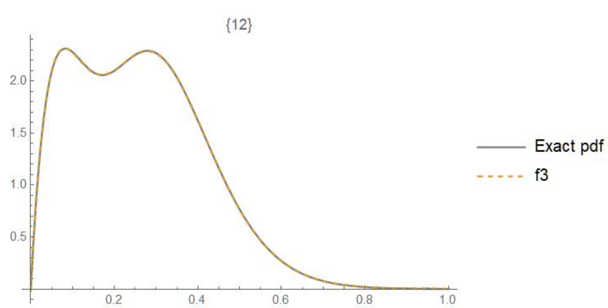

Figure 2.29: $f3$, n=12

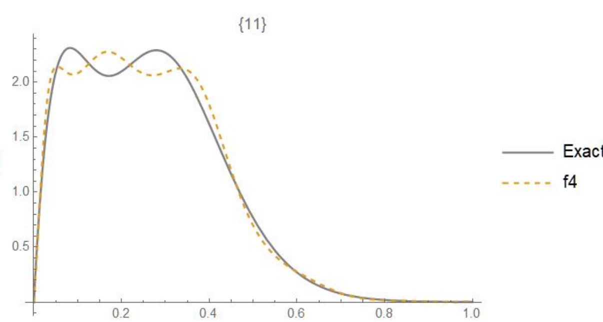

Figure 2.30: $f4$, n=11

It follows from the figures and Table 2.3 which includes the integrated squared differences between the exact and approximated pdf's, that the type 3 density approximant provides the most accurate approximation. Types 1, 2 can also yield reasonably close approximations. As for Types 4, 5 and 7, the resulting approximations are not as accurate in this case.

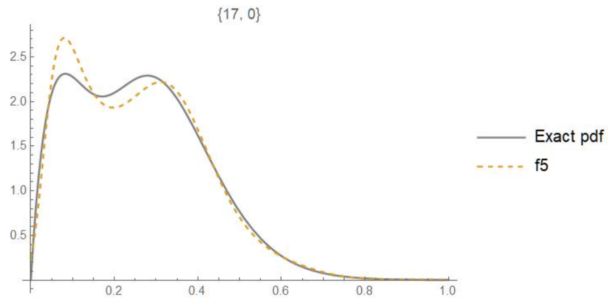

Figure 2.31: $f5$, ν=17, $\delta = 0$

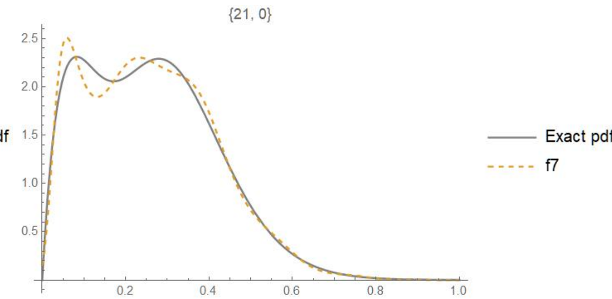

Figure 2.32: $f7$, ν=21, $\delta = 0$

Table 2.3: Comparison of the ISD's for different types and degrees

Type	Degree	# of moments	ISD
1	7	7	0.0002147
2	11	11	0.0000513
3	12	12	**0.0000086**
4	11	11	0.0105211
5	17, 0	34	0.0138683
7	21, 0	42	0.0093896

2.4.3 Mixture of normal pdf's

Consider an equally weighted mixture of two normal distribution density functions with parameters (0, 1) and (5, 3). We rescale the mixture using the transformation

$$Y = \frac{X}{\sigma},$$

where σ is its standard deviation. The support of Y is still $(-\infty, \infty)$. On applying the transformation

$$Z = \frac{\arctan Y}{\pi} + \frac{1}{2},$$

we obtain a random variable whose support is (0, 1). We then implement the methodologies and compare the results.

It follows from the figures and Table 2.4 that type 5 density approximants provide the most accurate approximations. Types 1 and 2 can also yield reasonably close approximation. As for types 3 and 7, the resulting approximations are not as accurate.

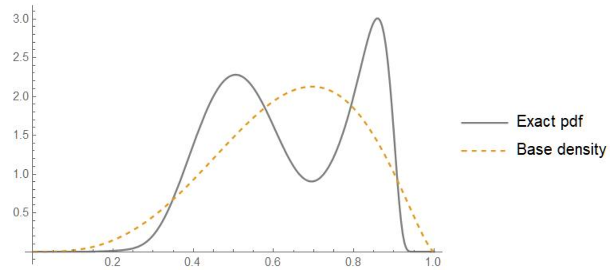

Figure 2.33: Transformed PDF's

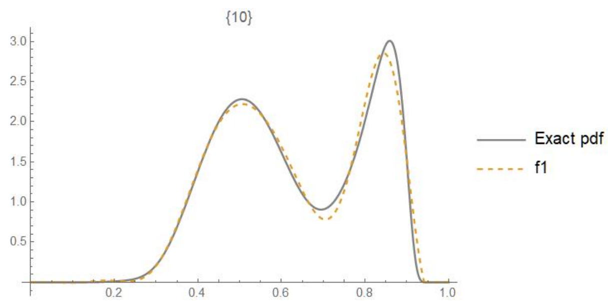

Figure 2.34: $f1$, n=10

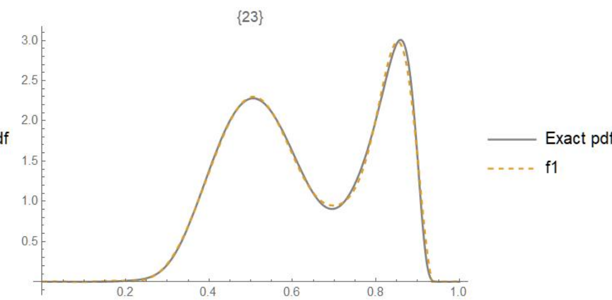

Figure 2.35: $f1$, n=23, best approximation

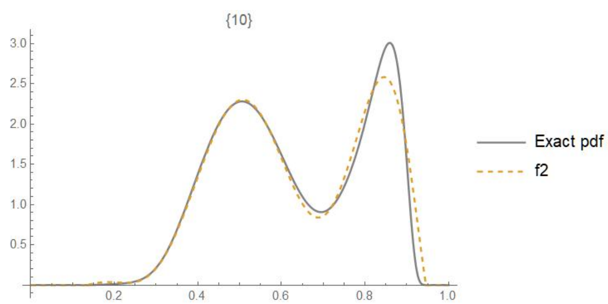

Figure 2.36: $f2$, n=10

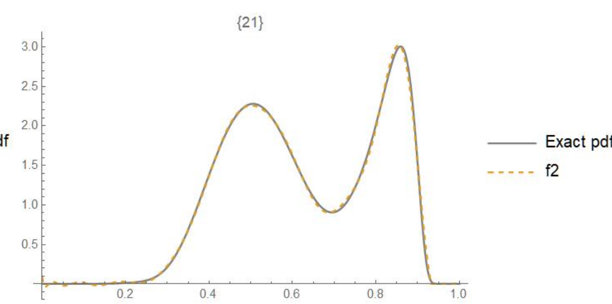

Figure 2.37: $f2$, n=21, best approximation

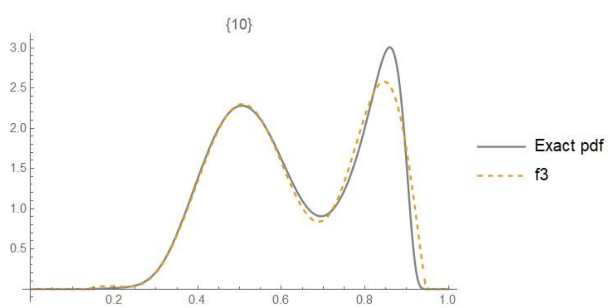

Figure 2.38: $f3$, n=10

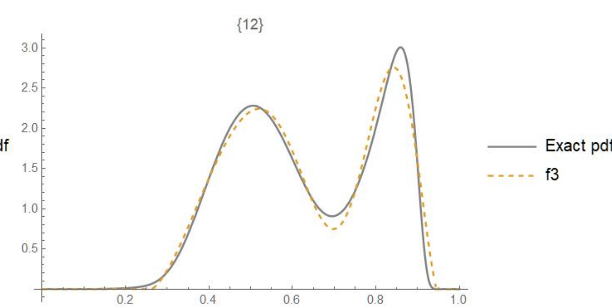

Figure 2.39: $f3$, n=12, best approximation

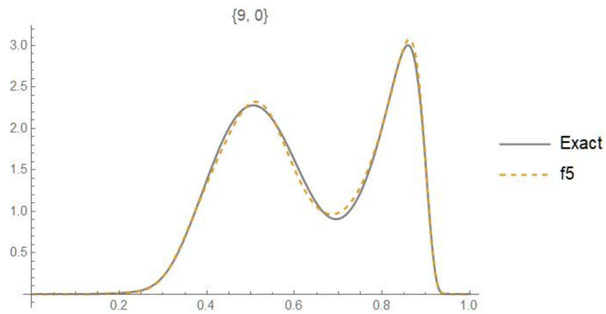

Figure 2.40: $f5$, ν=9, $\delta = 0$

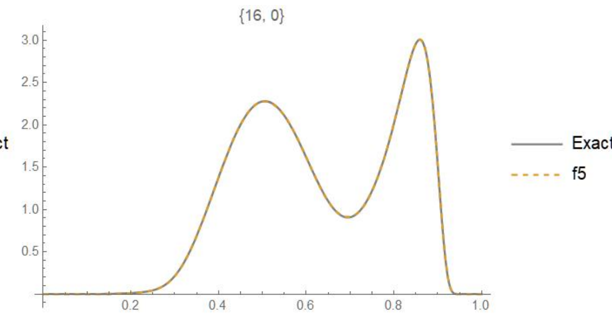

Figure 2.41: $f5$, n=16, $\delta = 0$, best approximation

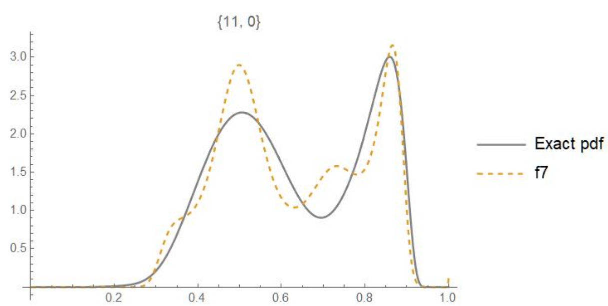

Figure 2.42: $f7$, $w = 0.1$, ν=11, $\delta = 0$

Table 2.4: Comparison of the ISD's for different types and degrees

Type	Degree	# of moments	ISD	Degree	# of moments	ISD
1	10	10	0.0113249	23	23	0.0023321
2	10	10	0.0302238	21	21	0.0014080
3	10	10	0.0304753	12	12	0.0195965
5	9, 0	18	0.0020256	16, 0	32	**0.0000073**
7	11, 0	22	0.0774153			

Alternatively, we can determine the approximants without standardizing the distribution, select a normal density as base density and apply a type 1 approximation.

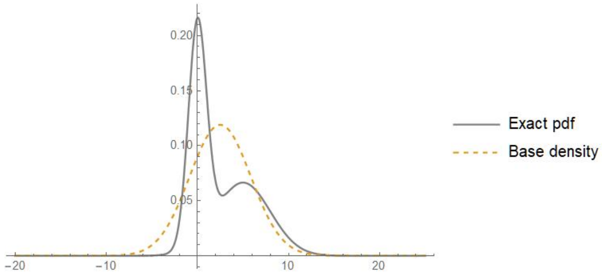

Figure 2.43: Mixture distribution PDF and base PDF

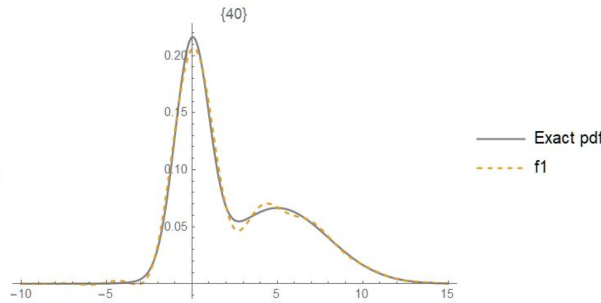

Figure 2.44: $f1$, n=40

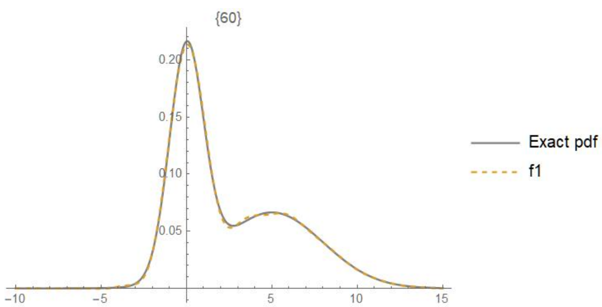

Figure 2.45: $f1$, n=60

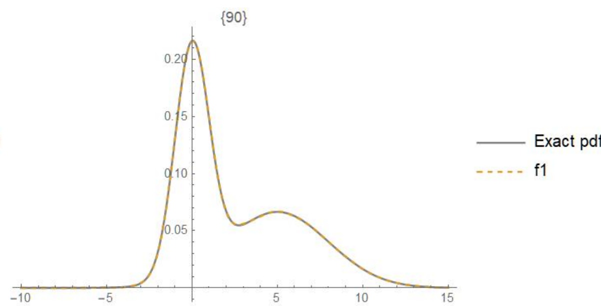

Figure 2.46: $f1$, n=90

Table 2.5: Comparison of the ISD's for different degrees (Type 1 approximant)

Degree	ISD
40	0.000302
60	0.000034
90	0.000002

It follows from the figures and Table 2.5 that in this case very accurate type 1 approximations can be obtained using a large number of moments.

In conclusion, depending of the type of distributions being approximated, types 1, 2, 3 or 5 approximants outperform types 4, 6 or 7 in terms of accuracy.

Bibliography

[1] Gajek, L. (1986). On improving density estimators which are not bona fide functions. *Annals of Statistics* **14**(4), 1612–1618.

[2] Glad, I. K., Hjort, N. L., and Ushakov, N. G. (2003). Correction of density estimators that are not densities. *Scandinavian Journal of Statistics* **30**(2), 415–427.

[3] Lindsay, B. G., Pilla, R. S. and Basak, P. (2000). Moment–based approximations of distribution using mixtures: theory and applications. *Annals of the Institute of Statistical Mathematics* **52**(2), 215–230.

[4] Provost, S. B. (2005). Moment–based density approximants. *The Mathematica Journal* **9**, 727–756.

Chapter 3

On Comparing Various Types of Moment-Based Density Estimates

3.1 Introduction

Density approximants are obtained by making use of the exact moments of the target distribution. When it comes to estimating a density function on the basis of a sample of observations, some of the previously introduced approaches can still be implemented by replacing the population moments by sample moments. The type 4 and type 6 approximants are not applicable in this case since the moments μ_4 and μ_6 cannot be evaluated unless the exact distribution is known.

These methodologies were applied to distributions whose support is the interval $(0, 1)$, which, in some cases, required a transformation of variable. This chapter pertains to the estimation of a density function from the sample moments associated with a set of observations. In the case of samples, one can assign endpoints a and b to the underlying distribution of Y, which can then be regarded as having a bounded support.

For example, we can let a equals to the minimum of the samples minus a constant times the sample standard deviation and set the right endpoint b similarly by adding a multiple of the standard deviation to the maximum of the sample. Then, we let $X = (Y - a)/(b - a)$, which maps $Y \in (a, b)$ onto $X \in (0, 1)$. After obtaining an estimate of the density function of the X, we apply the inverse transformation $Y = (b - a)X + a$ to obtain a density estimate for Y. Since this is a linear transformation, the density estimates obtained before and after carrying out the transformation have the same shape.

3.2 Simulated data

Before applying the methodologies to actual observations, we can assess their viability by generating data sets from known distributions.

First, we select a certain mixture of density functions as our underlying distribution. Then we fix a sample size and generate sample points from this distribution by making use of the Monte-Carlo simulation technique. We then apply several of the methodologies introduced in Chapter 2 by replacing the exact moments of the distribution with the sample moments determined from the generated data. Our aim in this subsection is to verify the applicability of the methodologies and determine their accuracy.

In this case, the sum of the squared differences (SSD's) between the empirical cdf and the estimated cdf is used as criterion to assess the accuracy of an estimate at the simulated points.

3.2.1 A mixture of beta pdf's

Consider an equally weighted mixture of two beta density function with parameters (6,2) and (3,9). We generated 500 sample points from this mixture. The base density is taken to be a beta distribution that has the same first two moments as the sample moments. For comparison purposes, plots of the kernel pdf and cdf estimates are included in certain graphs. The kernel density estimates were determined from Silverman's methodology.

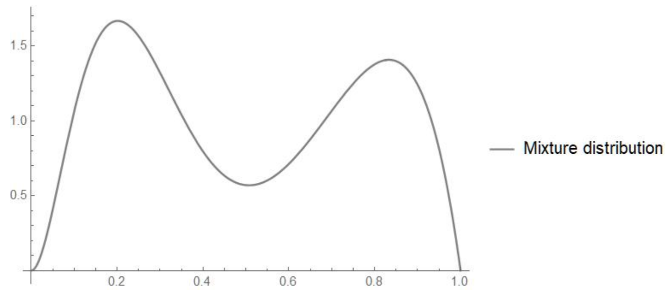

Figure 3.1: Mixture of beta PDF's

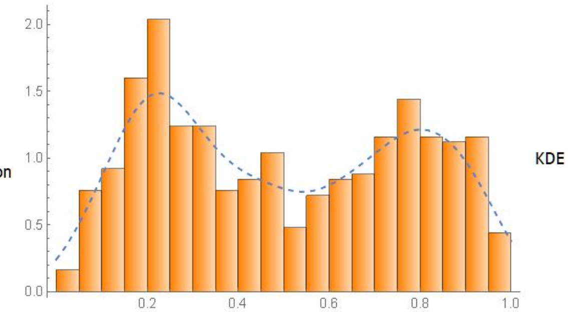

Figure 3.2: Histogram and kernel density

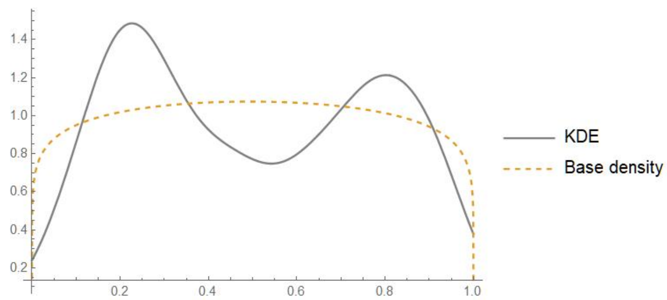

Figure 3.3: Kernel density and base density

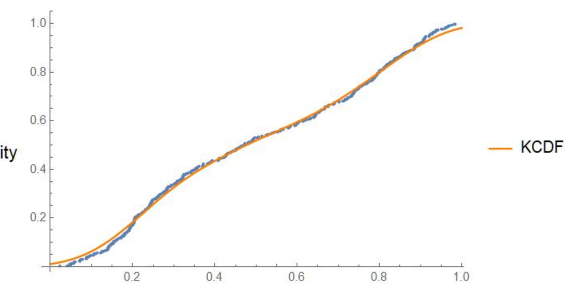

Figure 3.4: Empirical CDF and kernel CDF, (SSD=0.0832852)

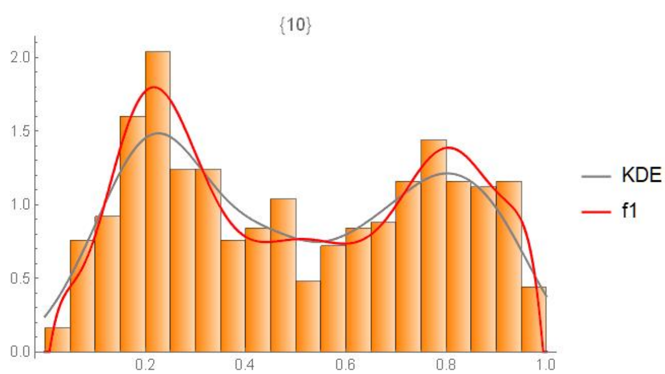

Figure 3.5: Histogram, kernel density and $f1$

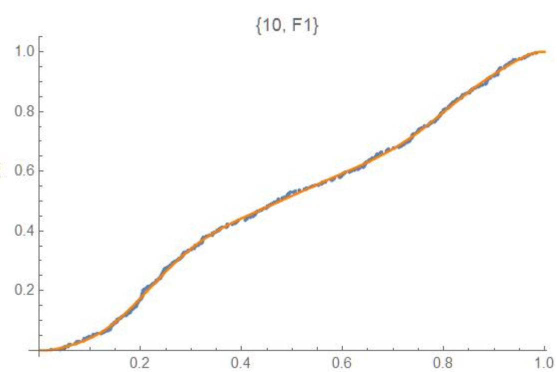

Figure 3.6: Empirical CDF and $F1$

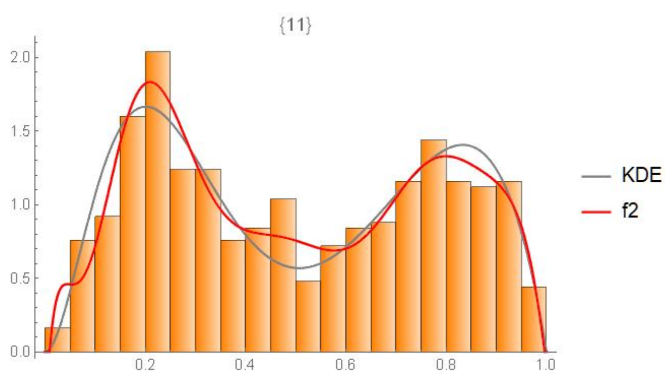

Figure 3.7: Histogram, kernel density and $f2$

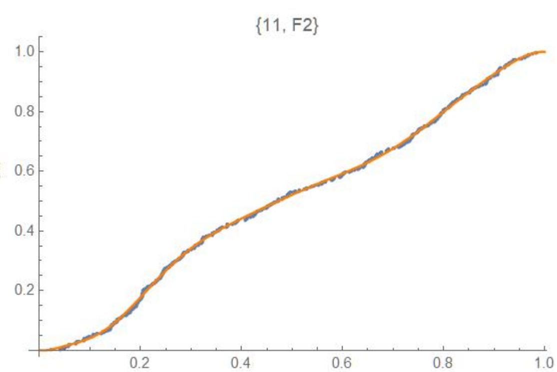

Figure 3.8: Empirical CDF and $F2$

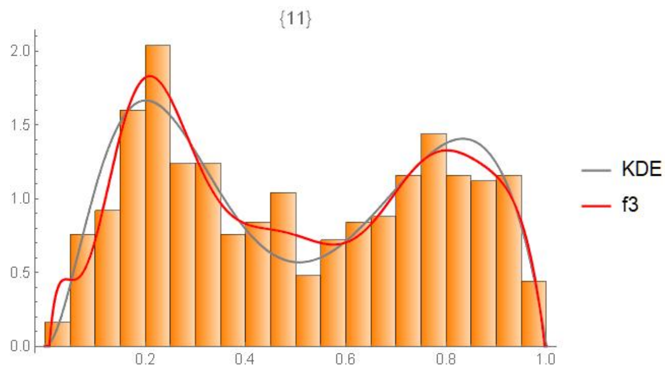

Figure 3.9: Histogram, kernel density and $f3$

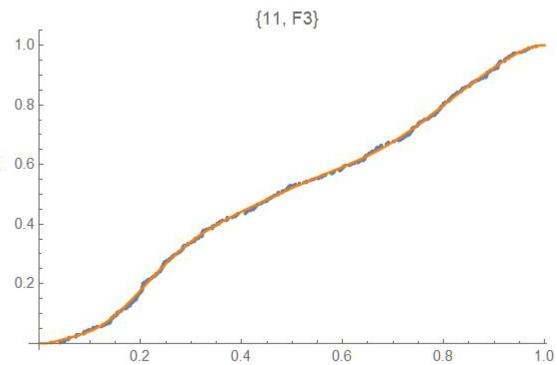

Figure 3.10: Empirical CDF and $F3$

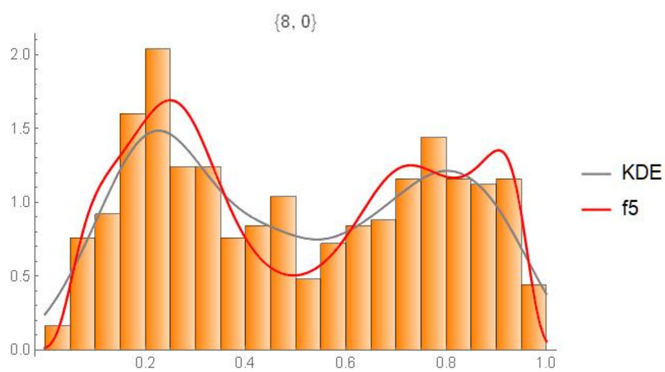

Figure 3.11: Histogram, kernel density and $f5$

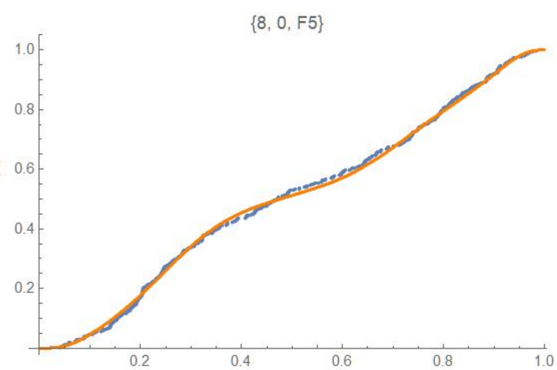

Figure 3.12: Empirical CDF and $F5$

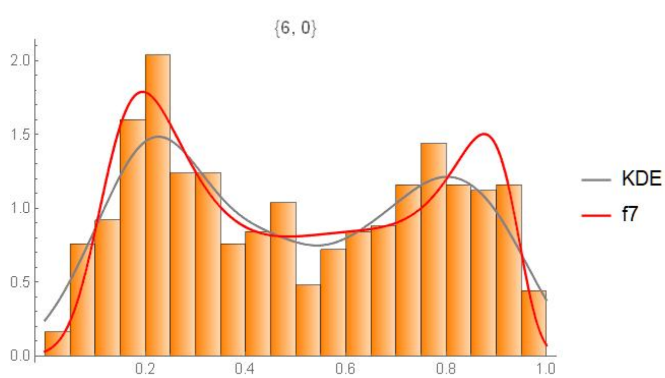

Figure 3.13: Histogram, kernel density and $f7$, $w = 0.1$

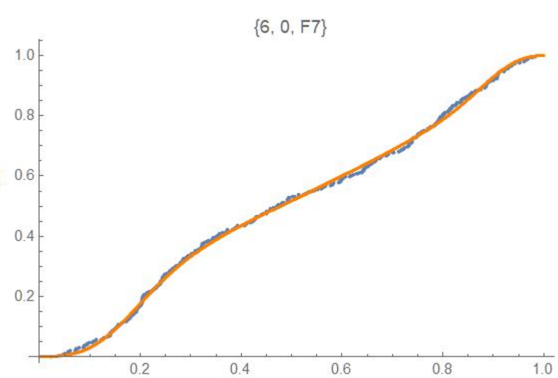

Figure 3.14: Empirical CDF and $F7$, $w = 0.1$

Table 3.1: Optimal density estimates of various types and SSD's

Type	Degree	Number of moments	SSD
1	10	10	0.0151
2	11	11	**0.0138**
3	11	11	0.0139
5	(8,0)	16	0.0545
7	(6,0)	12	0.0405

The degree(s) associated with the adjustments which are specified in Table 3.1, are the minimizers of the sum of squared differences between the associated cdf estimates and the empirical cdf (ECDF). The bold-face number corresponds to the smallest SSD and thus the best density estimate. It is seen that in this case, a type 2 estimate (single polynomial) is the most accurate. Types 1 and 3 can also provide excellent estimates. As for types 5 and 7, the resulting density estimates are not quite as accurate; moreover, they require more moments.

3.2.2 A mixture of gamma pdf's

Consider an equally weighted mixture of two beta density function with parameters (2, 2) and (15, 1). We generated 500 sample points from this mixture. For this sample, min=0166412, max=27.1662 and sd=6.53497.

After generating the sample points, we can obtain a density estimate for a distribution whose support is for instance (0, max+sd). We then map the distribution onto the interval (0, 1) and take as a base density a beta distribution whose first two moments coincide with those of the sample.

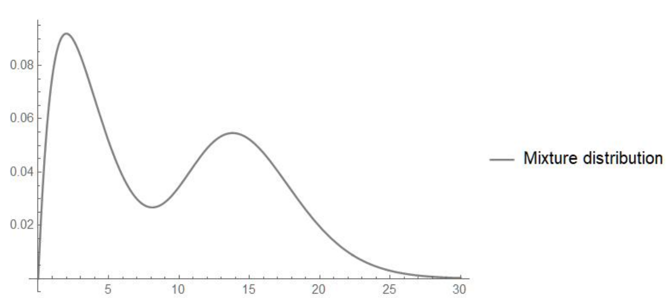

Figure 3.15: Mixture of gamma PDF's

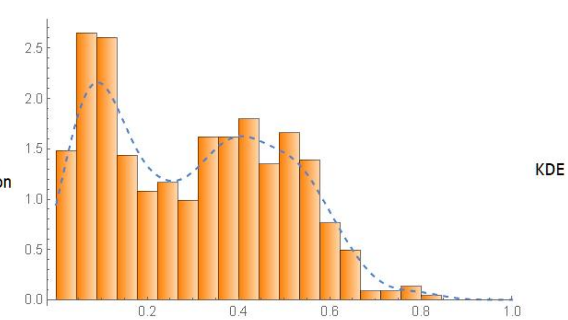

Figure 3.16: Histogram and kernel density

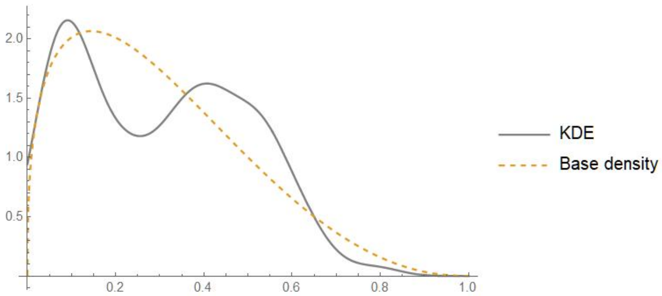

Figure 3.17: Kernel density and base density

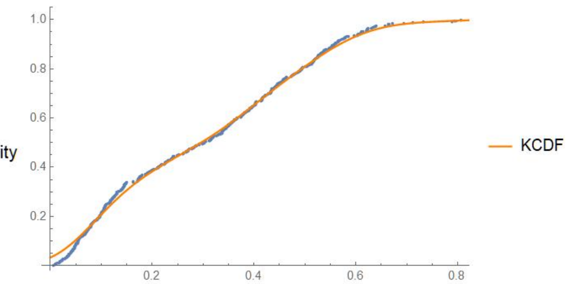

Figure 3.18: Empirical CDF and kernel CDF, (SSD=0.0944037)

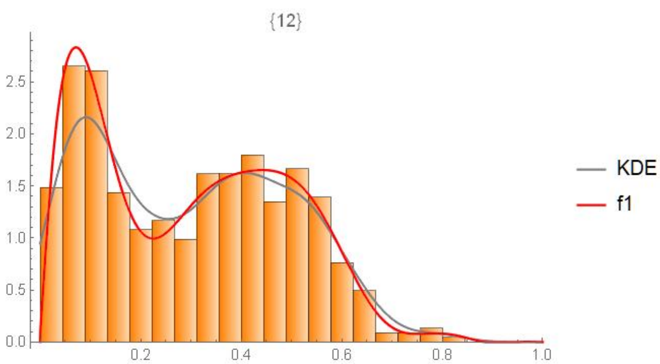

Figure 3.19: Histogram, kernel density and $f1$

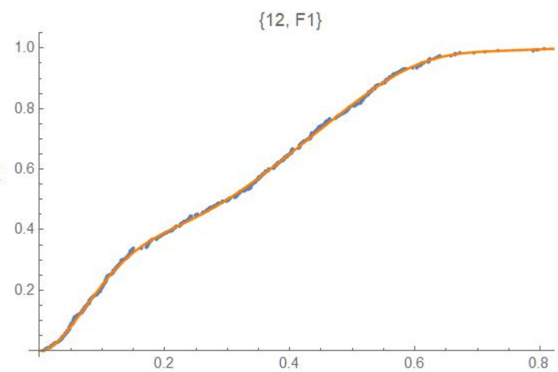

Figure 3.20: Empirical CDF and $F1$

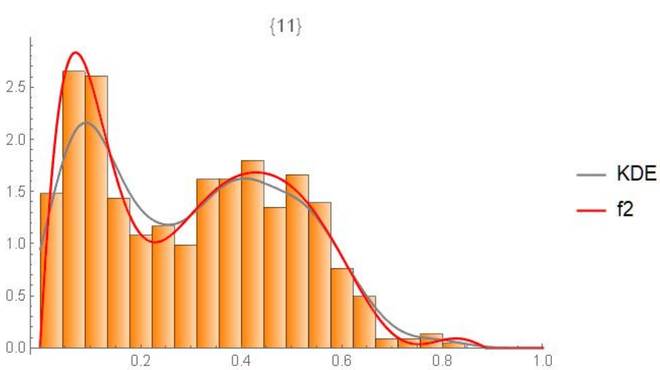

Figure 3.21: Histogram, kernel density and $f2$

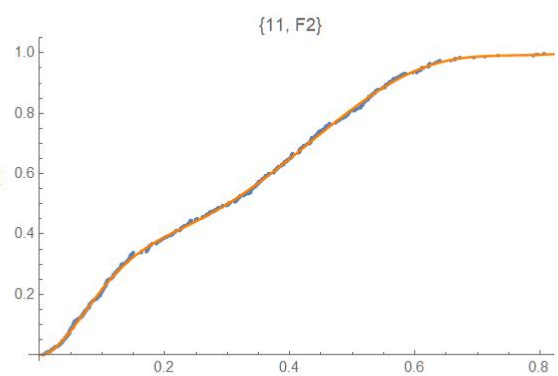

Figure 3.22: Empirical CDF and $F2$

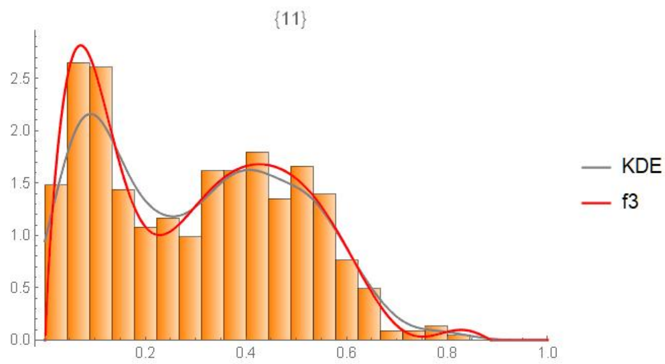

Figure 3.23: Histogram, kernel density and $f3$

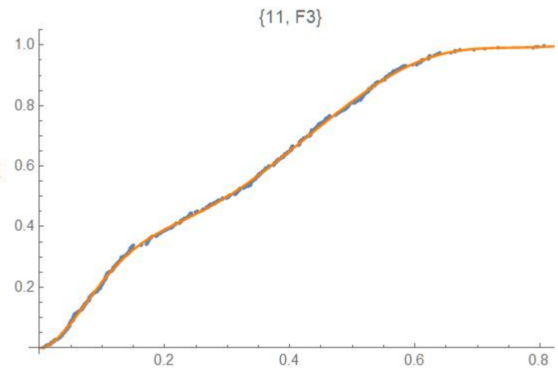

Figure 3.24: Empirical CDF and $F3$

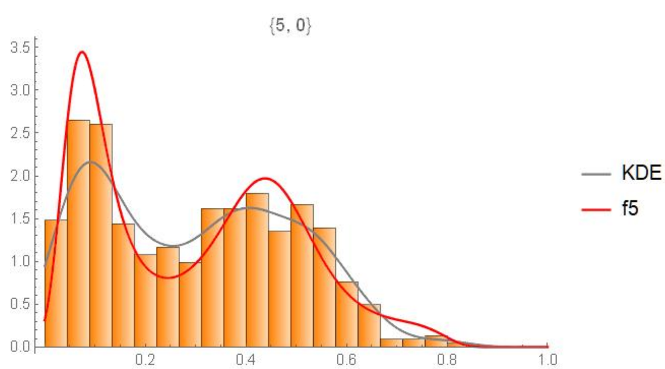

Figure 3.25: Histogram, kernel density and $f5$

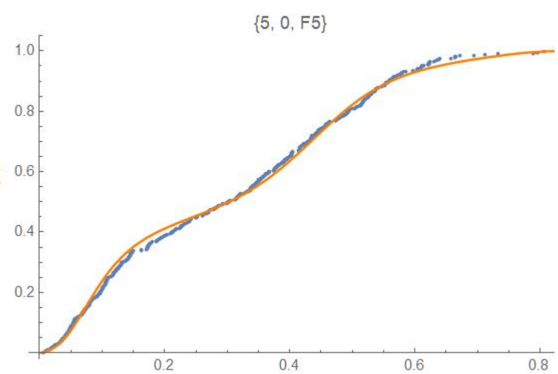

Figure 3.26: Empirical CDF and $F5$

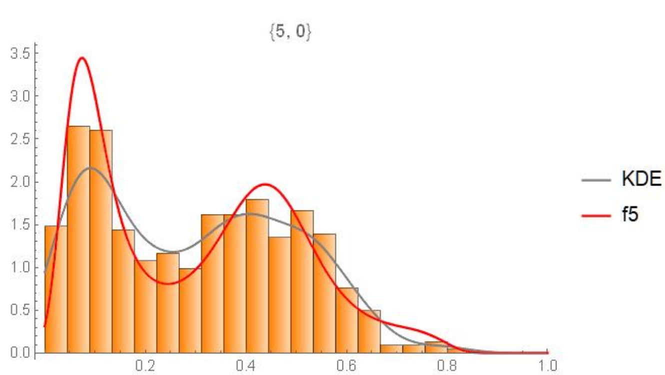

Figure 3.27: Histogram, kernel density and $f7$, $w = 0.1$

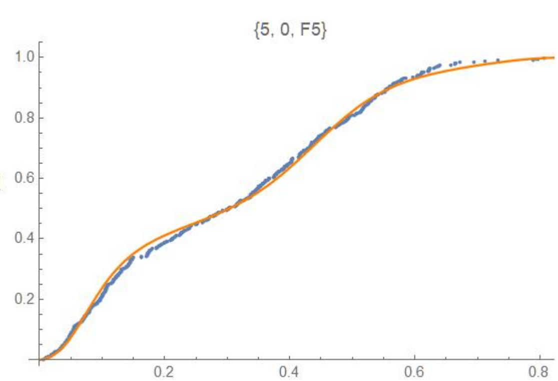

Figure 3.28: Empirical CDF and $F7$, $w = 0.1$

Table 3.2: Optimal density estimates of various types and SSD's

Type	Degree	# of moments	SSD
1	12	12	**0.0136**
2	11	11	0.0144
3	11	11	0.0142
5	5, 0	10	0.1164
7	9, 0	18	0.0440

It is seen from Table 3.2 that in this case, types 1, 2 and 3 density estimates perform fairly similarly in terms of accuracy, with type 5 being the least accurate. The type 1 estimate advocated in Provost (2005) appears to be the most accurate.

3.3 Examples

We now model actual data sets whose underlying distribution is unknown.

3.3.1 The Buffalo snowfall data

This data set, which is available from the R library, contains 63 data points corresponding to annual snowfall accumulation in inches as observed in Buffalo from 1910 to 1972; (min=25.0, max=126.4, sd=23.7). Let a = min − sd = 1.03 and b = max + sd = 150.12.

If the density estimates are negative on certain subranges of the support, the algorithms proposed by Gajek (1982) or Glad *et al.* (2003) can be applied. Alternatively, the negative parts can be removed, the resulting function, normalize and a DLA approximation, applied.

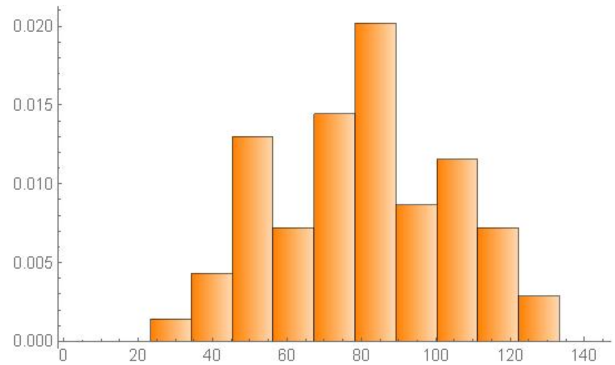

Figure 3.29: Histogram of the Buffalo snowfall data

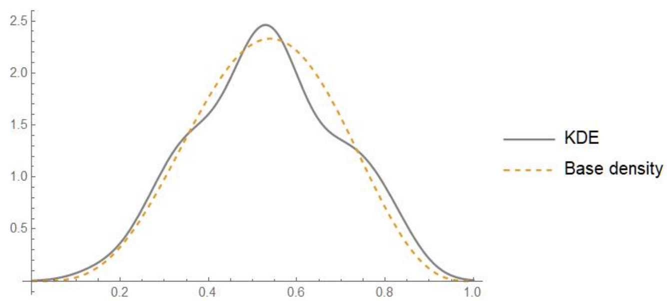

Figure 3.30: kde (grey line) and base density

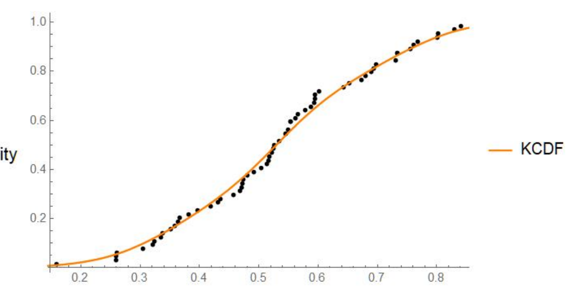

Figure 3.31: Empirical CDF and kernel CDF, (SSD=0.0286895)

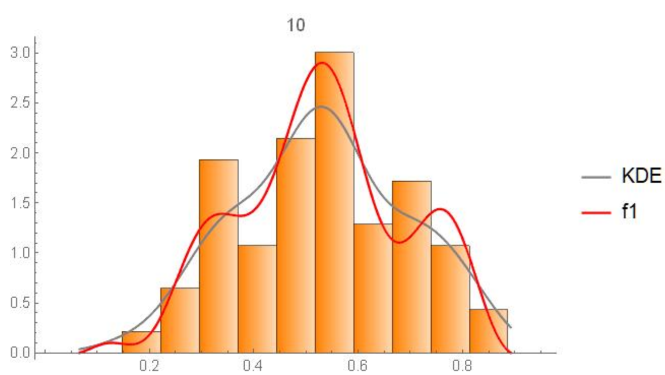

Figure 3.32: kde (grey line) and $f1$, n=10

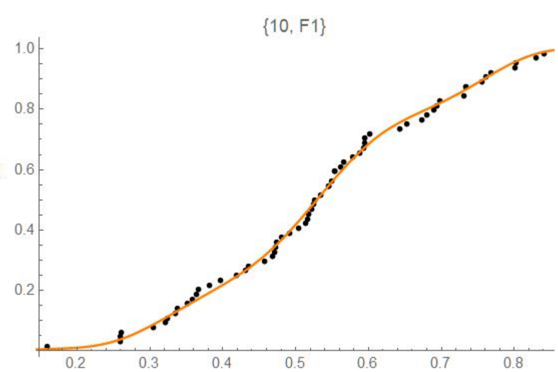

Figure 3.33: ECDF and $F1$, n=10

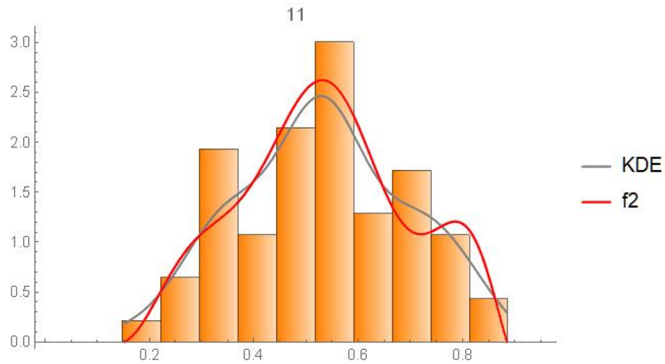

Figure 3.34: kde (grey line) and $f2$, n=11

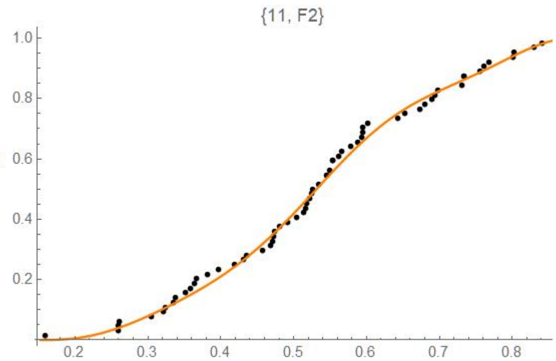

Figure 3.35: ECDF and $F2$, n=11

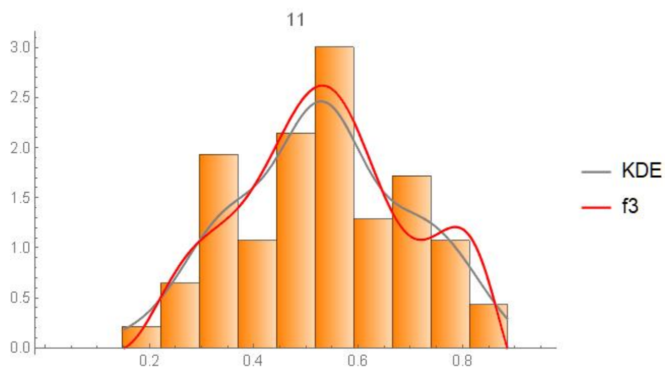

Figure 3.36: kde (grey line) and $f3$, n=11

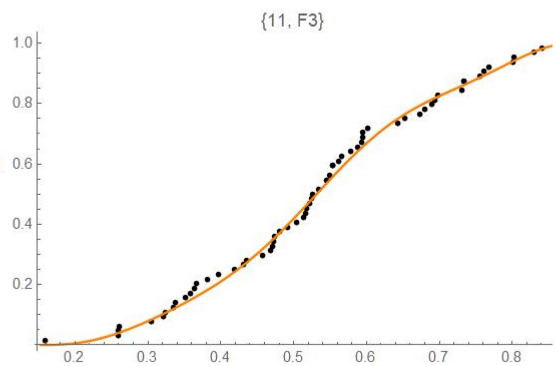

Figure 3.37: ECDF and $F3$, n=11

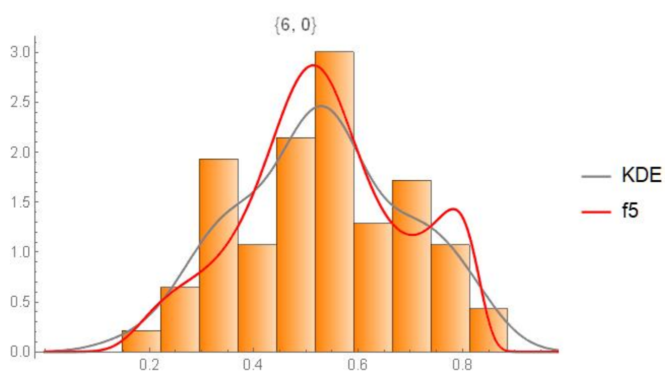

Figure 3.38: kde (grey line) and $f5$, ν=6, $\delta = 0$

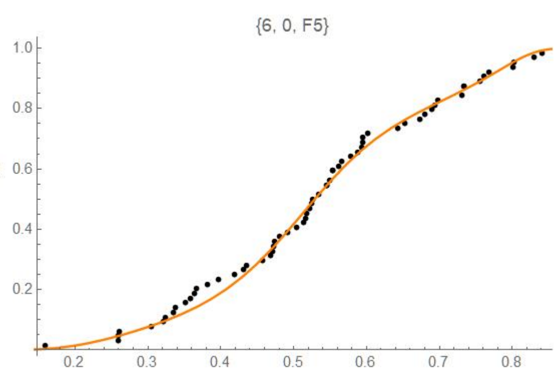

Figure 3.39: ECDF and $F5$, ν=6, $\delta = 0$

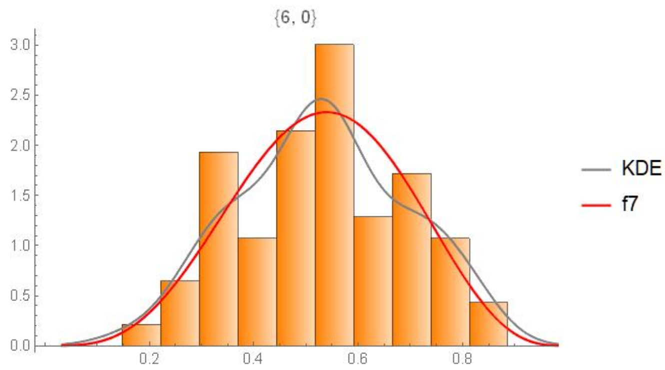

Figure 3.40: kde (grey line) and $f7$, ν=6, δ = 0

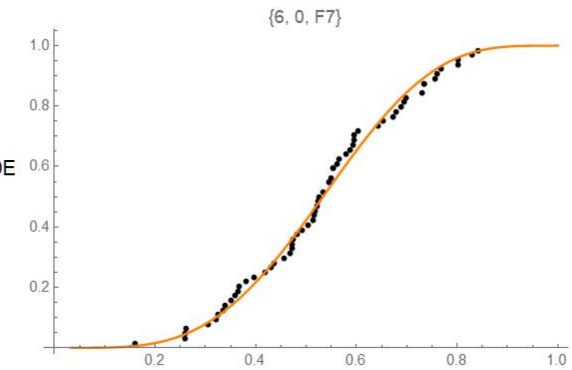

Figure 3.41: ECDF and $F7$, ν=6, δ = 0

Table 3.3: Optimal density estimates of various types and SSD's

Type	Degree	# of moments	SSD
1	10	10	**0.0150**
2	11	11	0.0260
3	11	11	0.0262
5	6, 0	12	0.0310
7	6, 0	12	0.0451

According to Table 3.3, the most accurate density estimate is of type 1 in this case. Moreover, this estimate requires fewer moments than the other ones.

3.3.2 The Old Faithful geyser data

This data set, which is available from the R library, contains 272 observations consisting of durations between consecutive eruptions in minutes; (min=1.60, max=5.10, sd=1.14). Let a = min − sd = 0.46 and b = max + sd = 6.24.

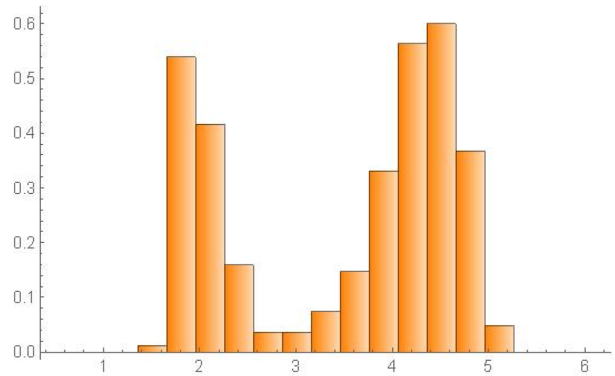

Figure 3.42: Histogram of the Old Faithful data

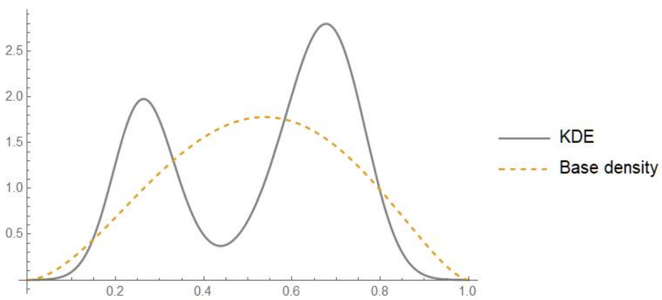

Figure 3.43: kde (grey line) and base density

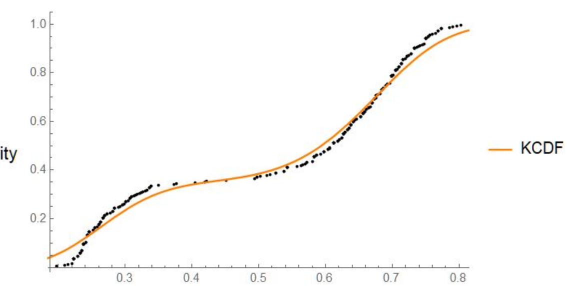

Figure 3.44: Empirical CDF and kernel CDF, (SSD=0.257145)

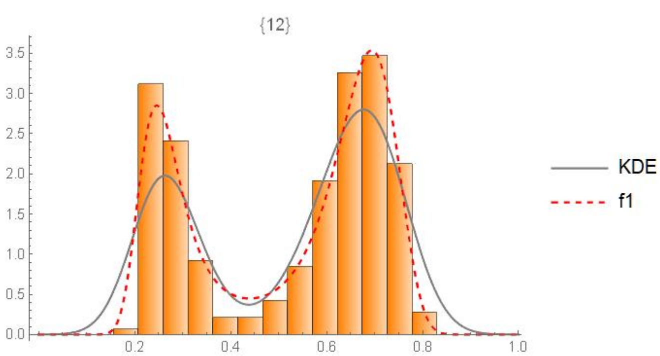

Figure 3.45: kde (grey line) and $f1$, n=12

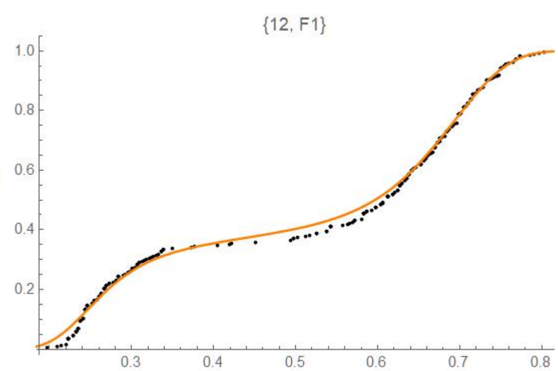

Figure 3.46: ECDF and $F1$, n=12

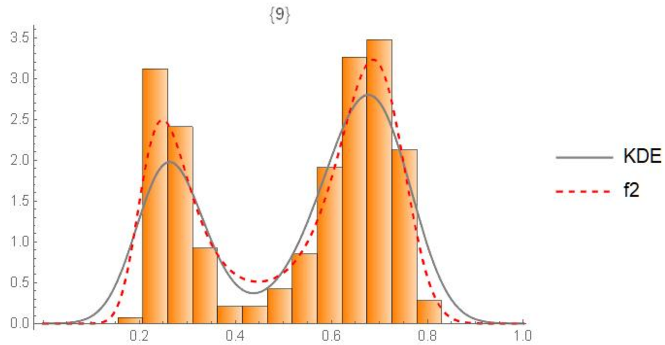

Figure 3.47: kde (grey line) and $f2$, n=9

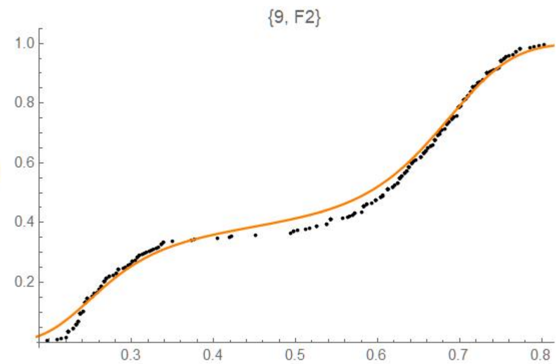

Figure 3.48: ECDF and $F2$, n=9

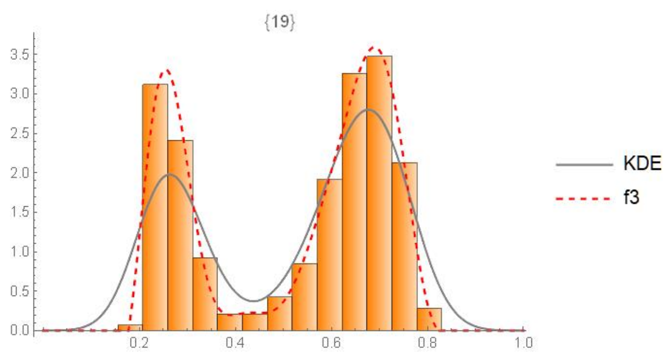

Figure 3.49: kde (grey line) and $f3$, n=10

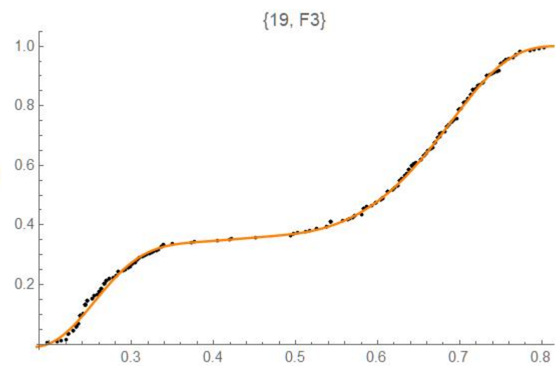

Figure 3.50: ECDF and $F3$, n=10

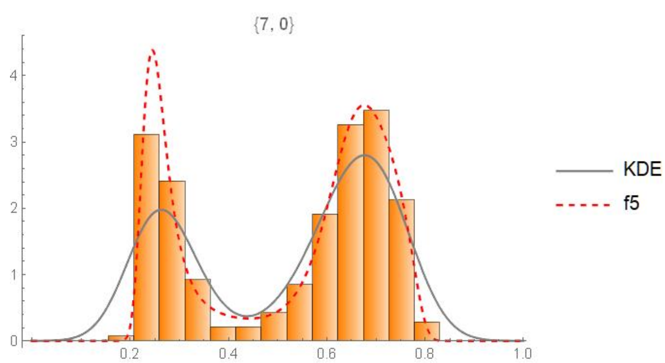

Figure 3.51: kde (grey line) and $f5$, n=7, δ=0

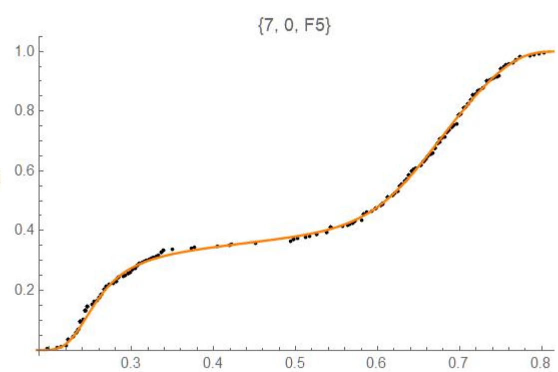

Figure 3.52: ECDF and $F5$, ν=7, δ=0

Table 3.4: Optimal density estimates of various types and SSD's

Type	Degree	# of moments	SSD
1	12	12	0.0732
2	9	9	0.1614
3	19	19	0.0276
5	(7,0)	14	**0.0230**

Table 3.4 indicates that the type 3 and 5 density estimates which require more moments than the other types are the most accurate for this bimodal data set.

3.3.3 A large data set: US household income in 2016

This data set, which represents the mean US household income per city/village (excluding zero values) is available from the website `www.kaggle.com/datasets`. It contains 32211 observations (min=5000, max=242857, sd=29873.1) that based on the US household incomes in year 2016. Let $a = 0$ and $b = \max + \text{sd} = 272730.1$. As previously explained, this data was mapped onto the interval (0, 1) before applying the various density estimation methodologies.

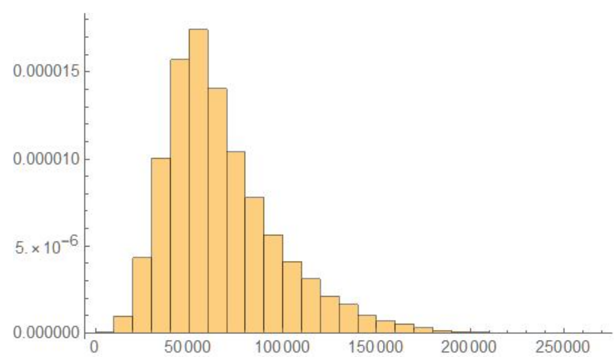

Figure 3.53: Histogram of the US household income data

It is seen graphically that in this case, the type 2 estimate provides the closest fit to the empirical cumulative distribution of the data.

We conclude from this initial study that types 1, 2, 3 and 5 density estimates can, in most of instances, model the data more accurately than kernel density estimates. However, more moments are usually required in the case of the fifth type.

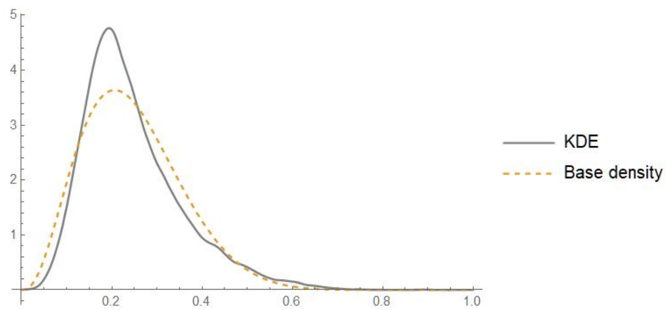

Figure 3.54: kde (grey line) and base density

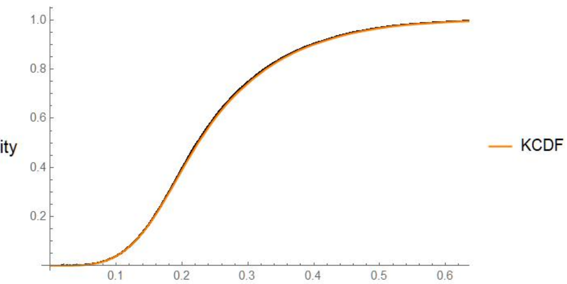

Figure 3.55: Empirical CDF and kernel CDF, (SSD=0.0923982)

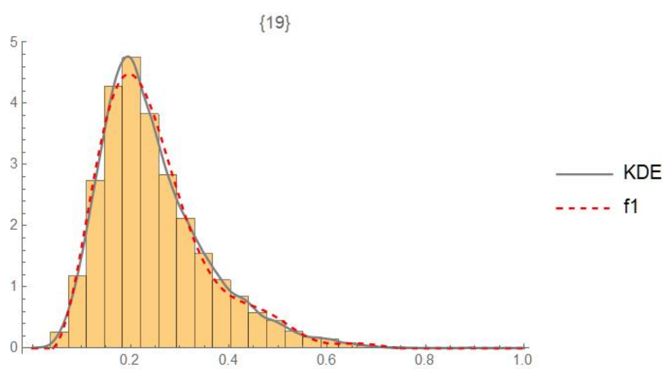

Figure 3.56: kde (grey line) and $f1$, n=19

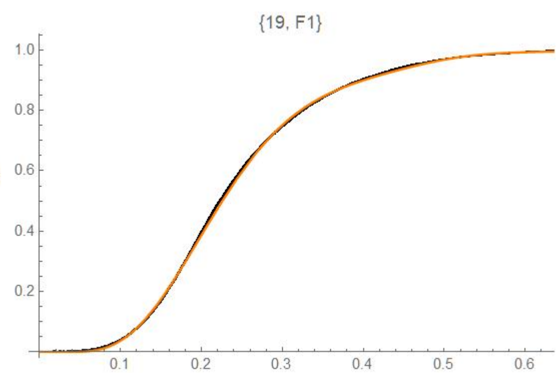

Figure 3.57: ECDF and $F1$, n=19

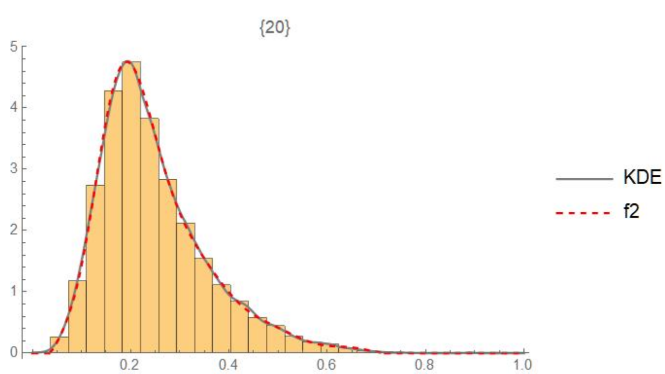

Figure 3.58: kde (grey line) and $f2$, n=20

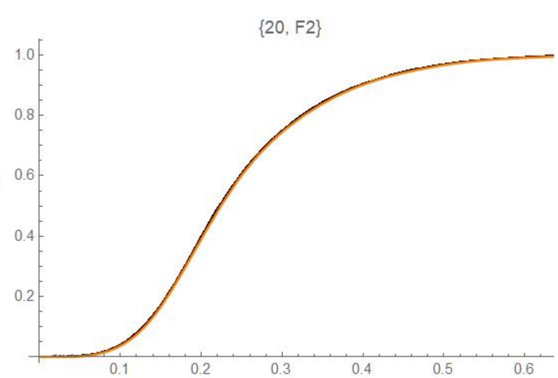

Figure 3.59: ECDF and $F2$, n=20

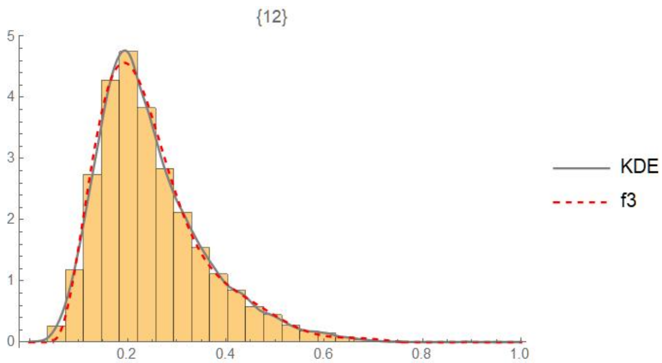

Figure 3.60: kde (grey line) and $f3$, n=12

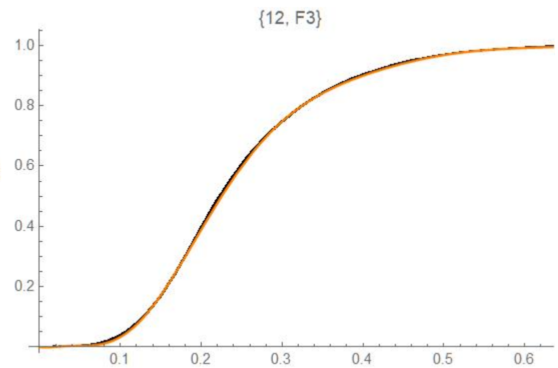

Figure 3.61: ECDF and $F3$, n=12

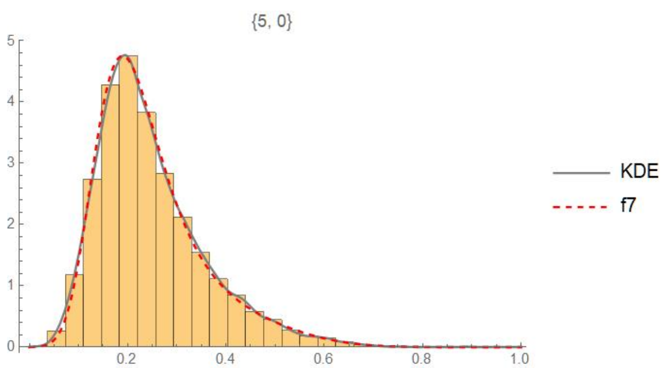

Figure 3.62: kde (grey line) and $f5$, ν=5, $\delta = 0$

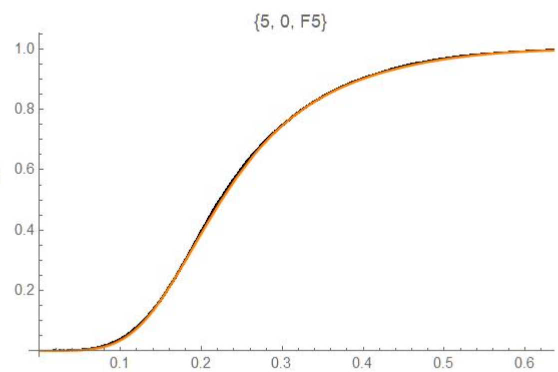

Figure 3.63: ECDF and $F5$, ν=5, $\delta = 0$

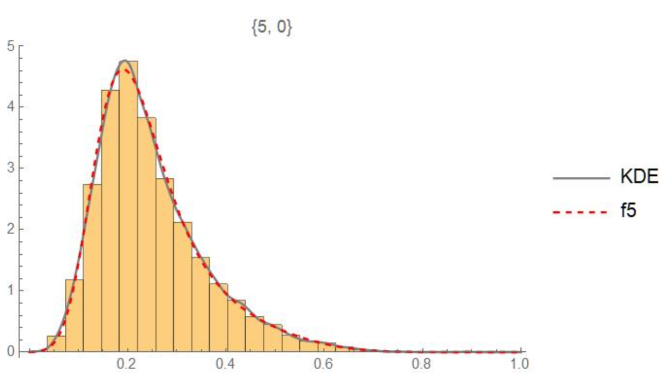

Figure 3.64: kde (grey line) and $f7$, w=0.1, ν=5, $\delta = 0$

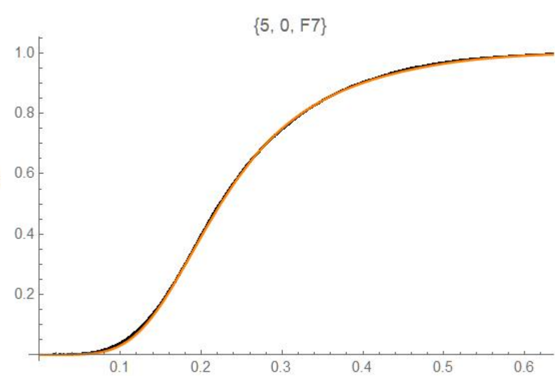

Figure 3.65: ECDF and $F7$, $w = 0.1$, ν=5, $\delta = 0$

Bibliography

[1] Gajek, L. (1986). On improving density estimators which are not bona fide functions. *Annals of Statistics* **14**(4), 1612–1618.

[2] Glad, I. K., Hjort, N. L., and Ushakov, N. G. (2003). Correction of density estimators that are not densities. *Scandinavian Journal of Statistics* **30**(2), 415–427.

[3] Provost, S. B.(2005). Moment–based density approximants. *The Mathematica Journal* **9**, 727–756.

Chapter 4

Certain Types of Samples and their Distributional Representativeness

4.1 Introduction

This chapter compares the stochastic simulation of samples from a given distribution via the Monte Carlo technique with four deterministic (quasi-Monte Carlo) approaches, which consist of making use of the percentiles resulting from a mapping with uniformly distributed discrete points or the percentiles obtained from the mean, mode and median of the order statistics associate with the target distribution. Those four alternative types of samples which require that the exact distribution be known, exhibit a more even coverage of the distributions. Such data generation techniques could find applications in finance, actuarial science, engineering and biology among other fields of scientific investigation. Some relevant references on Monte Carlo and quasi-Monte Carlo methodologies include Hammersley and Handscomb (1975), Niederreiter (1992), Caflish (1998), Robert and Casella (2004), Kalos and Whitlock (2008) and Metodi *et al.* (2018).

4.2 Types of samples and criteria for determining their distributional representativeness

Our objective is to determine a distribution's most representative sample of size n. A group of n representative sample points should be such that their sample moments are similar to those of the exact distribution up to a given order. Moreover, the empirical cumulative distribution function based on the n sample points should be relatively close to the exact cumulative density function.

Suppose X that follows a given distribution with probability density function $f_X(x)$ and cumulative distribution function $F_X(x)$, and let $X_1, X_2, \ldots, X_n$ be a random sample from this distribution. Let the order statistics be denoted as $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$, the probability density function of $X_{(k)}$ being

$$f_{X_{(k)}}(x) = \frac{n!}{(k-1)!(n-k)!} F_X(x)^{k-1}(1 - F[x])^{n-k} f(x) \tag{4.2.1}$$

on the support of the distribution.

Let $y_1(k)$ be the expected value of $X_{(k)}$, i.e.,

$$y_1(k) = E[X_{(k)}];$$

$y_2(k)$ be the mode of $X_{(k)}$, i.e.,

$$y_2(k) = \text{argmax}[f_{X_{(k)}}(x)];$$

$y_3(k)$ be the median of $X_{(k)}$, i.e., the 50^{th} percentile of the distribution of $X_{(k)}$; and

$y_4(k)$ be the $(2k-1)/(2n) \times 100^{\text{th}}$ quantile of $f_X(x)$.

Then, let

$s_1 = (y_1(1), y_1(2), \ldots, y_1(n));$

$s_2 = (y_2(1), y_2(2), \ldots, y_2(n))$;

$s_3 = (y_3(1), y_3(2), \ldots, y_3(n))$;

$s_4 = (y_4(1), y_4(2), \ldots, y_4(n))$;

and $s_5 = (x_1, x_2, \ldots, x_n)$ be a given simple random sample of size n.

These samples will be respectively referred to as types 1, 2, 3, 4 and 5 samples.

In this preliminary investigation, we make use of two distributions to determine which one of the five types of samples is the most representative.

Four criteria are considered to determine the most representative sample of size n generated from a given distribution.

Criterion 1: The first n sample moments, which according to Result 3 as stated in the Introduction contain all the information included in the sample, are compared to the first n exact moments by evaluating the sum of squared differences.

Criterion 2: The integrated squared difference between the resulting kernel density estimate and the exact density is evaluated.

Criterion 3: The integrated squared difference between the empirical CDF (ECDF) and the exact CDF is evaluated.

Criterion 4: The sum of the squared differences between the empirical CDF values corrected with the factor $n/(n+1)$ and the exact CDF values at the sample points. (Such points are for instance plotted in Figure 4.5)

4.3 Assessment of distributional representativeness

4.3.1 Sample generated from a beta(2,5) distribution

First, we consider a sample of size 5 from a beta(2,5) distribution.

It is seen from Table 4.1 and Table 4.2 that the type 1 and 4 samples present some advantages when compared to the other types.

Type 1 samples are generated using the expected value of the order statistics. The first

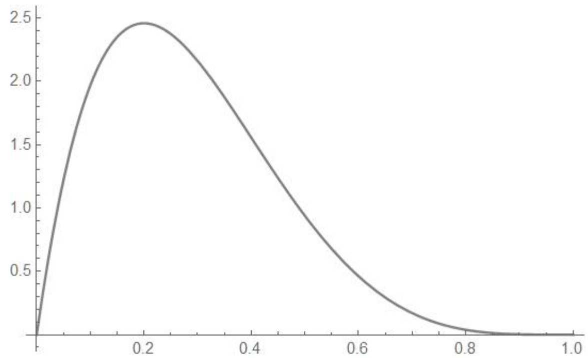

Figure 4.1: Beta(2,5) PDF

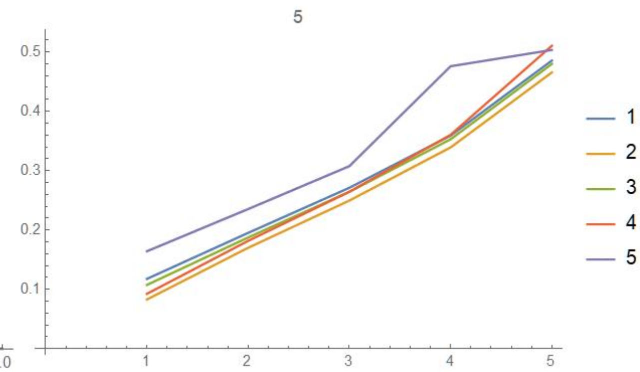

Figure 4.2: Samples of 5 types

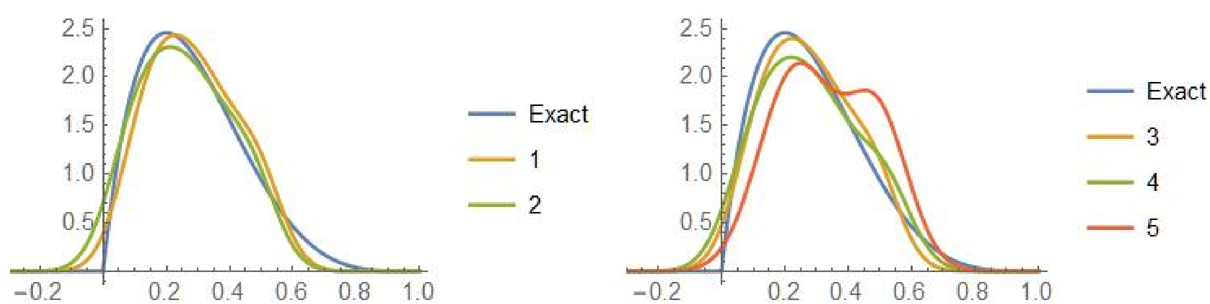

Figure 4.3: kde's for types 1 & 2 samples (left panel) and types 3, 4 & 5 samples (right panel)

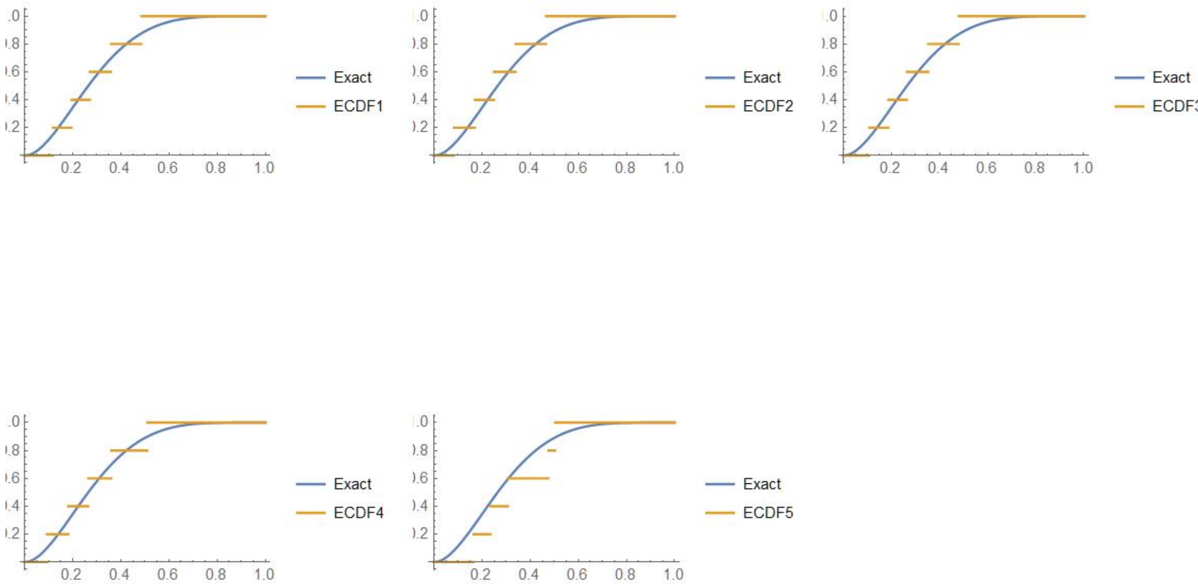

Figure 4.4: Exact and empirical CDF's for the 5 types of samples

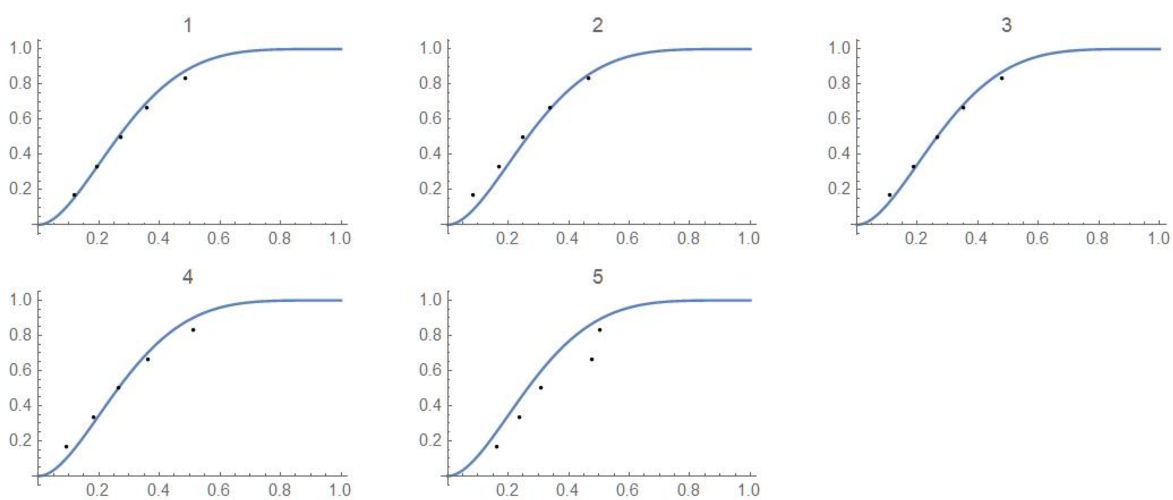

Figure 4.5: Corrected ECDF's for the 5 types of samples

sample moment of the type 1 sample happens to be the same as the exact one. Criterion 4 yields the smallest of the SSD's for this type of sample. Type 4 samples are generated using percentiles corresponding to equidistant cdf values. According to criteria 1 and 3, the type 4 sample outperforms the other samples.

In Table 4.3, the entries of Table 4.2 are divided by average of the values obtained for the 4 deterministic samples for each criterion and the row averages are evaluated for each type of samples. A bold-face number appearing in a table corresponds to the most representative type of sample in a given column.

According to Table 4.3, types 1 and 3 samples produce more accurate density estimates since the corresponding averages are smaller.

Table 4.1: Relative errors between sample and exact moments

Number of moments	1	2	3	4	5
Type 1 Sample	**0**%	– 8.53%	– 20.34%	– 33.48%	– 46.57%
Type 2 Sample	– 8.48%	– 19.76%	– 32.41%	– 45.45%	– 52.72%
Type 3 Sample	– 2.55%	– 12.01%	– 24.10%	– 37.20%	– 50.04%
Type 4 Sample	– 1.33%	**– 6.32**%	**– 13.90**%	– 23.78%	– 35.03%
Type 5 Sample	18.06%	22.61%	18.39%	**7.71**%	**– 7.12**%

Table 4.2: Rankings of the criteria for determine the most representative samples

Criterion	1	2	3	4	Average rank
Type 1 Sample	0.00028 (2)	0.0311 (4)	0.0026 (3)	**0.0033 (1)**	2.50
Type 2 Sample	0.00145 (4)	**0.0201 (1)**	0.0029 (4)	0.0126 (4)	3.25
Type 3 Sample	0.00047 (3)	0.0219 (2)	0.0024 (2)	0.0035 (2)	2.25
Type 4 Sample	**0.00016 (1)**	0.0293 (3)	**0.0021 (1)**	0.0111 (3)	**2.00**
Type 5 Sample	0.00333 (5)	0.1968 (5)	0.0107 (5)	0.0706 (5)	5.00

Table 4.3: Average of the criteria values

Criterion	1	2	3	4	Average
Type 1 Sample	0.47	1.22	1.04	**0.43**	0.79
Type 2 Sample	2.46	**0.78**	1.15	1.65	1.51
Type 3 Sample	0.80	0.86	0.97	0.46	**0.77**
Type 4 Sample	**0.27**	1.14	**0.85**	1.46	0.93

4.3.2 Samples generated from a mixture of beta pdf's

We now consider a sample of size 5 from an equally weighted mixture of two beta distributions with parameters (8, 12) and (3, 15). In order to determine the most representative sample of the distribution, the same four criteria are considered.

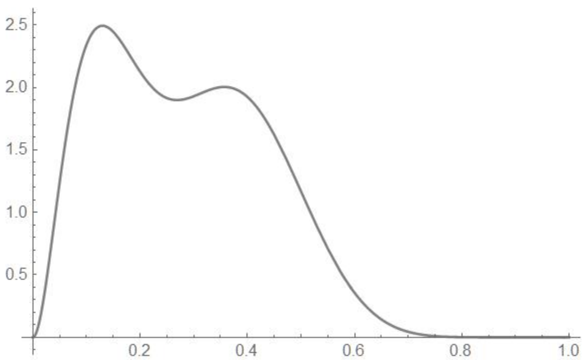

Figure 4.6: PDF of the mixture

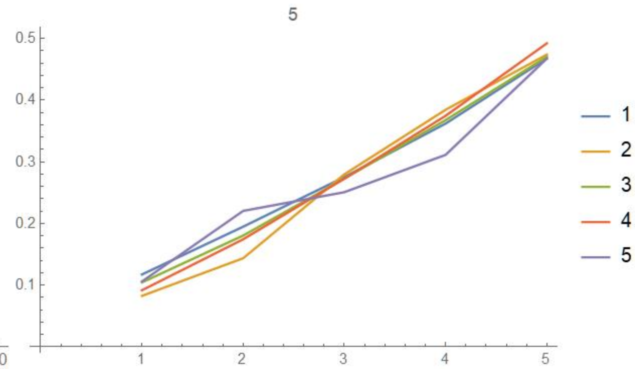

Figure 4.7: Samples of 5 types

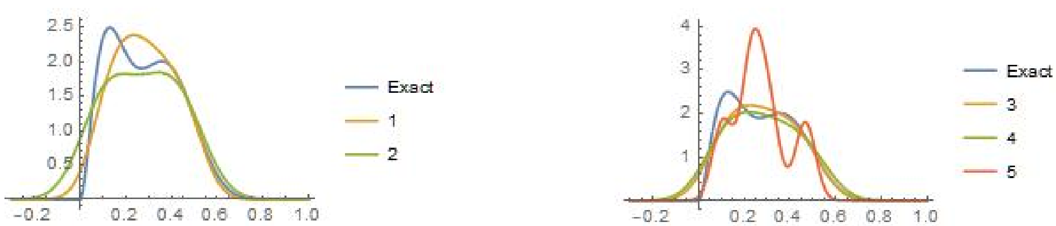

Figure 4.8: kde's for types 1 & 2 samples (left panel) and types 3, 4 & 5 samples (right panel)

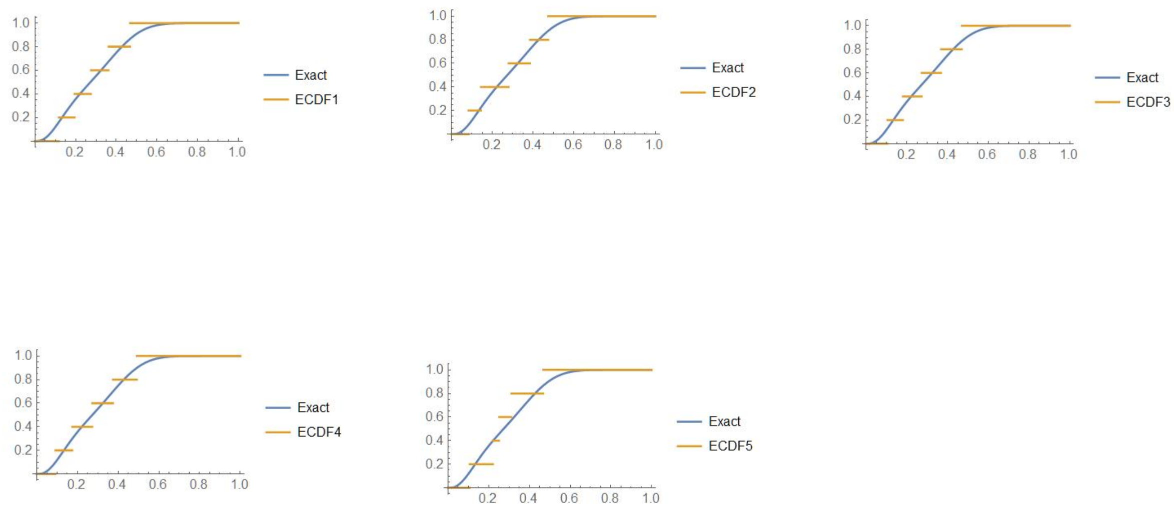

Figure 4.9: Exact and empirical CDF's for the 5 types of samples

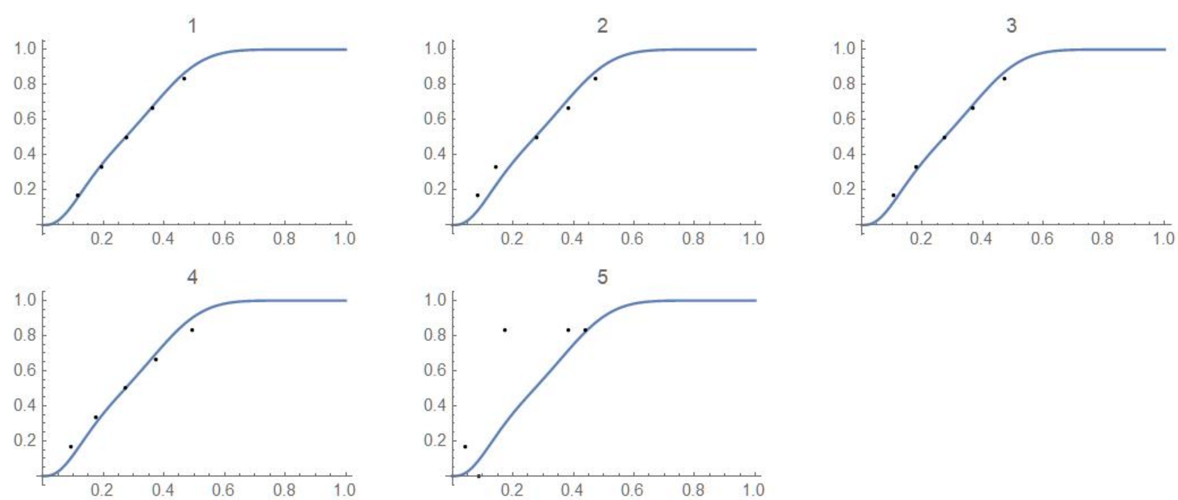

Figure 4.10: Corrected ECDF's for the 5 types of samples

Table 4.4: Relative errors between sample and exact moments

Order of moments	1	2	3	4	5
Type 1 Sample	**0**%	– 7.61%	– 17.19%	– 27.28%	– 37.35%
Type 2 Sample	– 3.72%	– 7.43%	– 12.98%	– 20.70%	– 29.87%
Type 3 Sample	– 1.52%	– 8.19%	– 16.52%	– 25.73%	– 35.34%
Type 4 Sample	– 0.81%	– 4.02%	**– 8.64**%	**– 14.69**%	**– 22.00**%
Type 5 Sample	– 15.34%	**3.15**%	– 81.35%	– 78.81%	– 37.53%

Table 4.5: Rankings of criteria for determining the most representative samples

Criterion	1	2	3	4	Average rank
Type 1 Sample	0.00016 (2)	0.068 (3)	0.0026 (4)	**0.0014 (1)**	2.50
Type 2 Sample	0.00023 (4)	0.078 (4)	0.0025 (3)	0.0237 (4)	3.75
Type 3 Sample	0.00018 (3)	**0.051 (1)**	0.0021 (2)	0.0035 (2)	2.00
Type 4 Sample	**0.00005 (1)**	0.062 (2)	**0.0019 (1)**	0.0111 (3)	**1.75**
Type 5 Sample	0.00334 (5)	0.887 (5)	0.1867 (5)	0.2874 (5)	5.00

Table 4.6: Average of the criteria values

Criterion	1	2	3	4	Average
Type 1 Sample	1.04	1.09	1.14	**0.14**	0.85
Type 2 Sample	1.47	1.20	1.10	2.39	1.54
Type 3 Sample	1.16	**0.79**	0.93	0.36	**0.81**
Type 4 Sample	**0.32**	0.96	**0.83**	1.12	**0.81**

It is seen from Tables 4.4, 4.5 and 4.6 that the types 1, 3 and 4 samples seem to outperform the other types.

The first sample moment of the type 1 sample happens to be the same as the exact one. According to Criteria 1 and 3, the type 4 sample provides the best density estimates. The entries included in Table 4.6 are obtained similarly to those appearing in Table 4.3. As confirmed by the results appearing in Table 4.6, according to the 4$^{\text{th}}$ criterion, the most accurate density estimate is obtained from the type 1 sample. However, on average the types 3 and 4 samples perform slightly better.

In light of this initial study, we conclude that types 1, 3 and 4 are viable candidates for providing the most representative samples and that the deterministic samples outperform Monte

Carlo random samples in that respect, which is to be expected since in the latter case, the distribution being simulated is generally unknown. Among the quasi-Monte Carlo samples, type 2, which is based on the mode of the distribution of the order statistics, appears to be the least representative. Further investigations involving other distributions and larger samples would be required to reach more definite conclusions.

Bibliography

[1] Caflisch, R. E. (1998). Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica* **7**, 1-49.

[2] Hammersley, J. M. and Handscomb, D. C. (1975). *Monte Carlo Methods*. Methuen: London.

[3] Kalos, M. H. and Whitlock, P. A. (2008). *Monte Carlo Methods*. Wiley-Blackwell: Weinheim.

[4] Leobacher, G. and Pillichshammer, F. (2014). *Introduction to quasi-Monte Carlo Integration and Applications*. Springer International Publishing: Switzerland.

[5] Metodi, M., Dobriyan, B. and Nikolai, V. (2018). *The Monte Carlo Method: Engineering Applications*. ACMO Academic Press: Mississauga.

[6] Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics: Philadelphia, Pa.

[7] Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods (2nd ed.)*. Springer: New York.

Chapter 5

A New Methodology for Characterizing Distributional Tail Behaviour

5.1 Introduction

First we present a review of various approaches used for classifying the tail behavior of a distribution. Then a new methodology that is easy to implement is proposed.

Klugman *et al.* (2012) provided several classification categories between light- and heavy-tailed distribution which are based, for instance, on moments, the hazard rate function and the mean excess loss function. Parzen (1979) examined the limiting behavior of density quantile functions which can be expressed as

$$f(Q(u)) \sim \begin{cases} (1-u)^{\alpha} & \text{for } \alpha > 0 \text{ and } \alpha \neq 1 \\ (1-u)\left(\log \frac{1}{1-u}\right)^{1-\beta} & \text{for } \alpha = 1 \text{ and } 0 \leq \beta \leq 1 \end{cases} \tag{5.1.1}$$

where f and Q represent the density and quantile function, respectively, and $f_1(u) \sim f_2(u)$ denotes that the ratio $f_1(u)/f_2(u)$ converges to a positive finite constant as $u \to 1$. The parameter α determines three types of tail behavior: short tails, medium tails and long tails which correspond to $\alpha < 1$, $\alpha = 1$ and $\alpha > 1$, respectively.

In order to refine the tail classification advocated by Parzen (1979), Schuster (1984) relied on two quantities, namely,

$$\alpha = \lim_{u \to 1^-} -\frac{1-u}{f(Q(u))} \frac{\partial \log [f(Q(u))]}{\partial u} \tag{5.1.2}$$

and

$$c = \lim_{u \to 1^-} (1-u)/f(Q(u)) = \lim_{u \to 1^-} 1/h(Q(u)), \tag{5.1.3}$$

where f, Q, and h represent the density, quantile and hazard function, respectively, to obtain five categories of tail behavior:

Short	$0 < \alpha < 1$	
Medium-Short	$\alpha = 1$	$c = 0$
Medium-Medium	$\alpha = 1$	$0 < c < \infty$
Medium-Long	$\alpha = 1$	$c = \infty$
Long	$\alpha > 1$.	

The latter criterion has a theoretical connection with the limiting size of extreme spacings.

The reader may also refer to Rojo (1996), whose classification, which involves more categories, is based on the residual lifetime distributions. Rojo (2010) reviewed the aforementioned classifications of tail behavior, carried out simulation studies and assigned categories to samples of observations. Table 5.1 provides a classification for some commonly used distributions as specified in Parzen (1979), Schuster (1984) and Rojo (1996). Heavy-tailed distributions belong to the medium-long and long tail categories in Schuster's classification.

5.2 Classifying the right tail of a distribution by means of the arctan transformation

We propose to make use of the percentiles of a transformed distribution as a criterion for characterizing the tail behaviour of the original distribution. Suppose that the random variable X follows a distribution with finite mean μ, finite variance σ^2 and density function $f_X(x)$. Let

$$Y = \frac{X - \mu}{\sigma} \tag{5.2.1}$$

denote the corresponding standardized random variable. On mapping Y from $(-\infty, \infty)$ or a subset thereof to $(-1, 1)$ or a subset thereof using the transformation

$$Z = \frac{2}{\pi} \arctan Y, \tag{5.2.2}$$

one has

$$Z = \frac{2}{\pi} \arctan\left(\frac{X - \mu}{\sigma}\right) \tag{5.2.3}$$

Table 5.1: Tail behavior classification for certain distributions

Distribution	Parzen	Schuster	Rojo
Uniform	Short	Short	Super-Short
Beta	Short	Short	Super-Short
Normal	Medium	Medium-Short	Weakly-Short
Extreme value	Medium	Medium-Short	Moderately-Short
Logistic	Medium	Medium-Medium	Medium
Exponential	Medium	Medium-Medium	Medium
Weibull ($k > 1$)	Medium	Medium-Short	Weakly-Short
Weibull ($k = 1$)	Medium	Medium-Medium	Medium
Weibull ($k < 1$)	Medium	Medium-Long	Weakly-Long
Lognormal	Medium	Medium-Long	Weakly-Long
Cauchy	Long	Long	Weakly-Long
Pareto ($k > 1$)	Long	Long	Weakly-Long
Pareto ($k = 1$)	Long	Long	Moderately-Long
Pareto ($k < 1$)	Long	Long	Super-Long

so that

$$X = \sigma \tan(\pi z/2) + \mu, \tag{5.2.4}$$

the density function of Z being

$$f_Z(z) = \frac{\pi\sigma}{2} \sec^2(\pi z/2)\, f_X(\sigma \tan(\pi z/2) + \mu). \tag{5.2.5}$$

The transformed density functions of certain distributions are plotted in Figure 5.1–5.8. Let q_α denotes the 100αth percentile of the distribution of Z. We make use of the difference between the 99.9999th and the 90th percentiles as the proposed criterion for classifying the right tail behavior of X and denote the criterion value by p.

Distributions whose mean is not finite are classified as having a super long tail. Distributions whose mean is finite but whose variance is infinite are classified as having a moderately long tail. Generally, the fewer the number of finite moment a distribution possesses, the heavier its associated distributional tail is.

The proposed classification criteria which are based on

$$p = q_{.999999} - q_{.90}$$

are summarized in Table 5.2.

Table 5.2: Classification of tail behaviour

Category	Criterion
Super-Short	$p < 0.1$
Moderately-Short	$0.1 \le p < 0.2$
Weakly-Short	$0.2 \le p < 0.3$
Medium	$0.3 \le p < 0.4$
Extended-Medium	$0.4 \le p < 0.5$
Weakly-Long	$p \ge 0.5$
Moderately-Long	indefinite second moment
Super-Long	indefinite first moment

5.3 Comparison with other criteria

Based on the criterion described in Section 5.2, we can construct a table containing various different distributions and compare our classification with other ones. The results are summarized in Table 5.3. The proposed criterion and the other 3 criteria yield similar classification results.

Note that for the beta, Weibull and lognormal distributions, the shapes of the standardized densities and thus the tail behavior and the associated value of p vary with the parameters.

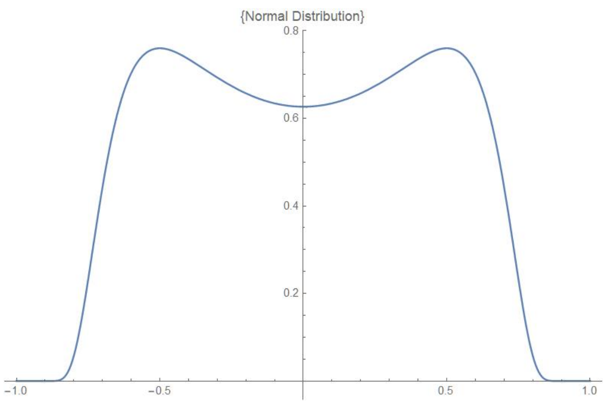

Figure 5.1: Normal distribution, $f_z(z)$

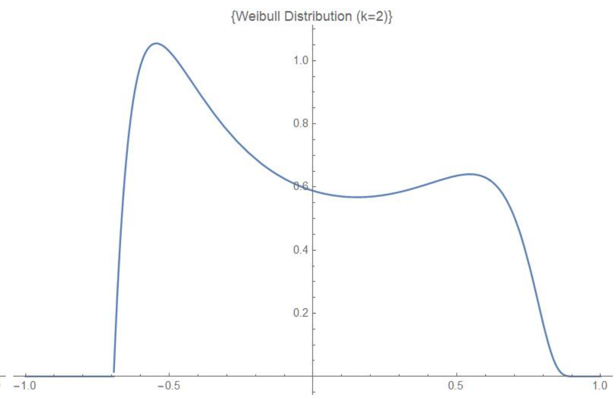

Figure 5.2: Weibull distribution (k=2), $f_z(z)$

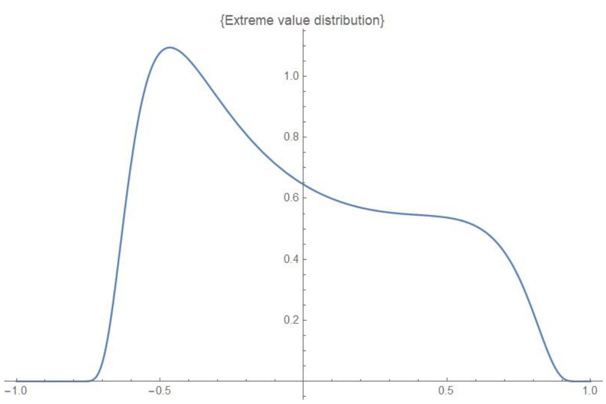

Figure 5.3: Extreme value distribution, $f_z(z)$

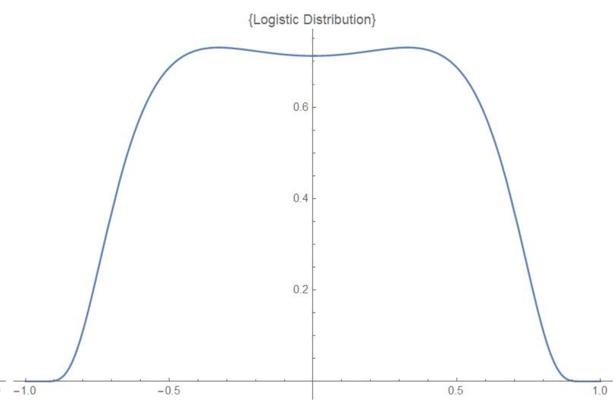

Figure 5.4: Logistic distribution, $f_z(z)$

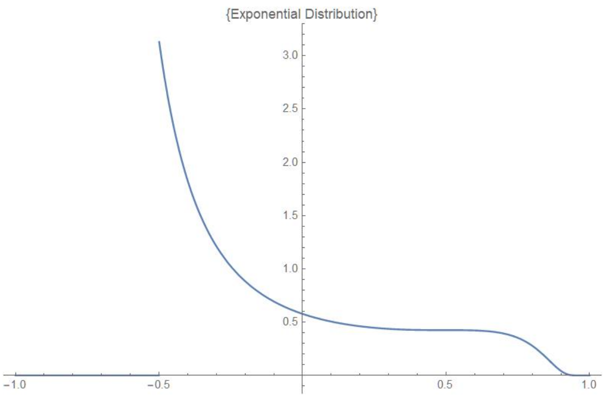

Figure 5.5: Exponential distribution, $f_z(z)$

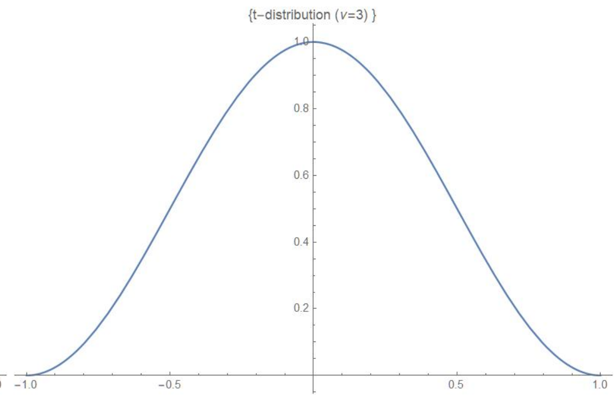

Figure 5.6: t-distribution, $\nu = 3$ $f_z(z)$

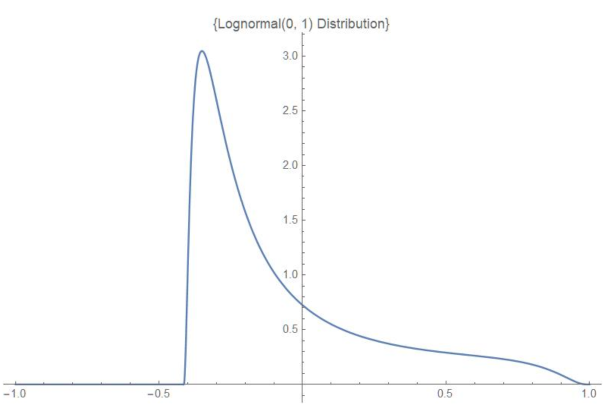

Figure 5.7: Lognormal(0, 1) distribution, $f_z(z)$

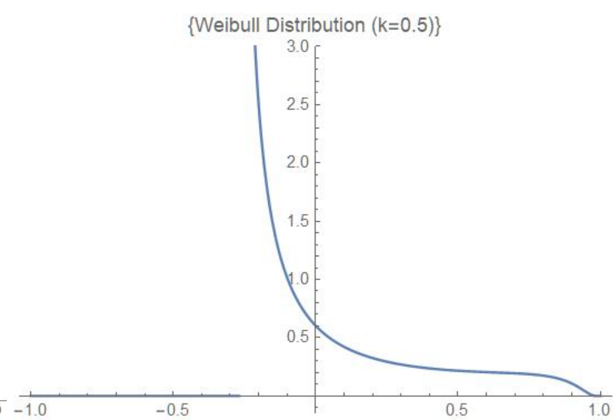

Figure 5.8: Weibull distribution (k=0.5), $f_z(z)$

Table 5.3: Comparative classification of tail behavior for certain distributions

Distribution	Parzen	Schuster	Rojo	Our category	p
Uniform	Short	Short	Super-Short	Super-Short	0.0646
Beta(5, 2)	Short	Short	Super-Short	Moderaterly-Short	0.1152
Normal	Medium	Medium-Short	Weakly-Short	Weakly-Short	0.2898
Weibull ($k = 2$)	Medium	Medium-Short	Weakly-Short	Weakly-Short	0.2998
Extreme value	Medium	Medium-Short	Moderately-Short	Medium	0.3549
Logistic	Medium	Medium-Medium	Medium	Medium	0.3562
Exponential	Medium	Medium-Medium	Medium	Medium	0.3672
Weibull ($k = 1$)	Medium	Medium-Medium	Medium	Medium	0.3672
Lognormal(0, 1)	Medium	Medium-Long	Weakly-Long	Weakly-Long	0.5201
Weibull ($k = 0.5$)	Medium	Medium-Long	Weakly-Long	Weakly-Long	0.5800
Pareto ($k > 1$)	Long	Long	Weakly-Long	Moderately-Long	Table 5.2
Pareto ($k = 1$)	Long	Long	Moderately-Long	Super-Long	Table 5.2
Pareto ($k < 1$)	Long	Long	Super-Long	Super-Long	Table 5.2
Cauchy	Long	Long	Weakly-Long	Super-Long	Table 5.2

We also used the criterion to determine the tail behavior of some other distributions. The results are included in Table 5.4. The "*" as the second parameter of the gamma distribution indicates that the scale parameter may be any value, the resulting value of p remaining unchanged. beta2(α, β) denotes the beta distribution of the second kind whose pdf is

$$f(x) = \frac{x^{\alpha-1}(1+x)^{-\alpha-\beta}}{B(\alpha,\beta)},$$

where $B(\cdot)$ is the Beta function.

The proposed criterion which is generally consistent with those previously introduced in the literature, has the advantage of being simpler to apply.

Table 5.4: Classification of the tail behavior of some other distributions

Distribution	Our category	p
Rayleigh	Weakly-Short	0.2998
gamma(50, *)	Medium	0.3070
gamma(20, *)	Medium	0.3148
t(20)	Medium	0.3269
gamma(5, *)	Medium	0.3316
beta2(50, 30)	Medium	0.3444
gamma(2, *)	Medium	0.3479
gamma(1, *)	Medium	0.3672
t(5)	Extended-Medium	0.4244
beta2(2, 5)	Extended-Medium	0.4591
t(3)	Weakly-Long	0.5071
beta2(5, 3)	Weakly-Long	0.5609

Bibliography

[1] Klugman, S. A. and Panjer, H. H. and Willmot, G. E. (2012). *Loss models: from data to decisions, 4th Ed.* John Wiley & Sons, New York.

[2] Parzen, E. (1979). Nonparametric statistical data modeling. *Journal of the American Statistical Association* **74**, 105–121.

[3] Rojo, J. (1996). On tail categorization of probability laws. *Journal of the American Statistical Association* **91**, 378–384.

[4] Rojo, J. and Ott, R. C. (2010). Testing for tail behavior using extreme spacings. *arXiv preprint arXiv:1011.6458*

[5] Schuster, E. F. (1984). Classification of probability laws by tail behavior. *Journal of the American Statistical Association* **79**, 936–939.

Chapter 6

Quadratic Forms in Various Types of Random Variables

6.1 Introduction

In this chapter, we approximate the distribution of quadratic forms in certain random variables, namely the gamma, inverse Gaussian, binomial and Poisson. Some distributional limit theorems such as those that are discussed in Del Barrio *et al.* (2005) in connection with a certain empirical quantile process, involve quadratic forms in exponential random variables (a particular case of the gamma random variables considered in this chapter). Moreover, three test statistics that can be expressed as quadratic forms in exponential random variables, are described in Donald and Paarsch (2002). Whittle (1960) considered the case of random variables in Poisson variables in connection with contingency tables.

In generalized linear models (GLM's), the conditional distribution of the response variable, Y_i, is often taken to be a member of an exponential family, such as the Gaussian (normal), binomial, Poisson, gamma, or inverse-Gaussian families of distributions, and the determination of the distribution of quadratic forms in such random variable, including those of the form $\epsilon'\hat{\Sigma}^{-1}\epsilon$, could contribute to further developments in connection with GLM's. The case of quadratic forms in Gaussian variables is already discussed at length in Mathai and Provost (1992) and numerous of the references included therein. Thus, we will focus on approximating the distribution of quadratic forms in the other types of random variables previously mentioned. This will be achieved by applying a technique introduced in Provost (2005), that is, by adjusting an appropriate base density by a polynomial whose coefficients are determined from the moments of the quadratic forms and those of the base density. The resulting approximate pdf and cdf will be respectively denoted by $f1$ and $F1$. As shown in Provost *et al.* (2009), this approach can also be utilized to approximate the probability mass functions of discrete distributions.

Let $X_1, X_2, \ldots, X_n$ be random variables and A be a $n \times n$ symmetric matrix, where

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix}.$$

The quadratic form,

$$Y = \mathbf{X}' A \mathbf{X},$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_n)'$ can be expanded as follows:

$$\begin{aligned} Y &= a_{11}X_1^2 + a_{22}X_2^2 + \cdots + a_{nn}X_n^2 + 2a_{12}X_1X_2 + 2a_{13}X_1X_3 + \cdots + 2a_{n-1,n}\, X_{n-1}\, X_n \\ &= \sum_{i=1}^{n} a_{ii}\, X_{ii}^2 + \sum_{1\le i<j\le n} 2a_{ij}\, X_i X_j. \end{aligned}$$

6.2 On evaluating the exact moments of quadratic forms via the symbolic approach

Suppose that the random vector $\mathbf{X}$ follows a distribution with density function $f_X(x_1, x_2, \ldots, x_n)$. We can make use of the moments associated with $f_X(\cdot)$ to determine the h^{th} moment of the random variable Y. Let $\mu_{\mathbf{X}}(h_1, h_2, \ldots, h_n)$ be the joint moment of order $(h_1, h_2, \ldots, h_n)$ of $X_1, X_2, \ldots, X_n$. The h^{th} moment of Y is then

$$\begin{aligned} E[Y^h] &= E\left[\left(\sum_{i=1}^{n} a_{ii}\, X_{ii}^2 + \sum_{1\le i<j\le n} 2\, a_{ij}\, X_i\, X_j\right)^h\right] \\ &\equiv E\left(\sum_i c_i\, X_1^{h_{i1}}\, X_2^{h_{i2}} \cdots X_n^{h_{in}}\right), \end{aligned} \tag{6.2.1}$$

where the c_i's and h_{ij}'s can be determined by expanding symbolically the expression within the square brackets. Moreover, in light of the linearity property of mathematical expectations, we have

$$\begin{aligned} E[Y^h] &= \sum_i E[c_i\, X_1^{h_{i1}} X_2^{h_{i2}} \cdots X_n^{h_{in}}] \\ &= \sum_i c_i\, \mu_{\mathbf{X}}(h_{i1}, h_{i2}, \ldots, h_{in}). \end{aligned} \tag{6.2.2}$$

Thus, we can evaluate the moments of Y from the joint moments of $\mathbf{X}$ without having to determine the exact distribution of Y.

For example, letting $\mathbf{X} = (X_1, X_2)$ follow a bivariate normal distribution with mean $(-1, 2)'$ and covariance matrix

$$\begin{pmatrix} 2 & -1 \\ -1 & 3 \end{pmatrix}$$

and

$$Y = \begin{pmatrix} X_1 & X_2 \end{pmatrix} \begin{pmatrix} 1 & 4 \\ 4 & -2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix},$$

the second moment of Y is obtained as follows:

$$\begin{aligned} E[Y^2] &= E\left[\left(\begin{pmatrix} X_1 & X_2 \end{pmatrix}\begin{pmatrix} 1 & 4 \\ 4 & -2 \end{pmatrix}\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}\right)^2\right] \\ &= E[X_1^4 + 16X_1^3X_2 + 60X_1^2X_2^2 - 32X_1X_2^3 + 4X_2^4] \\ &= E[X_1^4] + 16E[X_1^3X_2] + 60E[X_1^2X_2^2] - 32E[X_1X_2^3] + 4E[X_2^2] \\ &= \mu_{\mathbf{X}}(4,0) + 16\mu_{\mathbf{X}}(3,1) + 60\mu_{\mathbf{X}}(2,2) - 32\mu_{\mathbf{X}}(1,3) + 4\mu_{\mathbf{X}}(0,2) \\ &= 3481. \end{aligned}$$

Alternatively, one can also evaluate the second moment of Y by integration:

$$E[Y^2] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\begin{pmatrix} x_1 & x_2 \end{pmatrix}\begin{pmatrix} 1 & 4 \\ 4 & -2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right)^2 f_X(x_1, x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2. \tag{6.2.3}$$

However the proposed approach is more efficient, especially when n is large.

6.3 Applications

6.3.1 Quadratic forms in gamma random variables

The pdf of a gamma distribution with shape parameter α and scale parameter β is

$$f(x) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} x^{\alpha-1} e^{-x/\beta}.$$

Consider two independently distributed gamma random variables X_1 and X_2 with parameters (2,2) and (9,1), respectively.

Let

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$$

and $Y = (X_1, X_2)A(X_1, X_2)'$. While it is challenging to find the exact probability density function of Y, we can use the methodology described in Result 1 included in the Introduction namely, multiplying a base density by a polynomial, to obtain an approximation thereof. This type of density approximant will be utilized throughout this chapter. Since we do not know the exact probability density and cumulative distribution functions of Y, we can resort to histograms of simulated quadratic forms (n=10,000) and the associated empirical cumulative distributions for assessing the accuracy of the approximants. We use the sum of squared differences (SSD) between the simulated cumulative distribution function and the approximated cumulative distribution function as our criterion for determining the degree of the polynomial adjustment. The smaller the SSD, the more accurate the approximation. We select the degree beyond which the SSD does not decrease significantly or appears to increase. This criterion will be used for the quadratic forms in continuous r.v.'s considered in this chapter.

We make use of a gamma density as base density and determine its parameters by setting the first two moments of the gamma distribution equal to those of the quadratic form which are obtained from the symbolic approach. The distributional results are illustrated graphically in Figures 6.1 to 6.5.

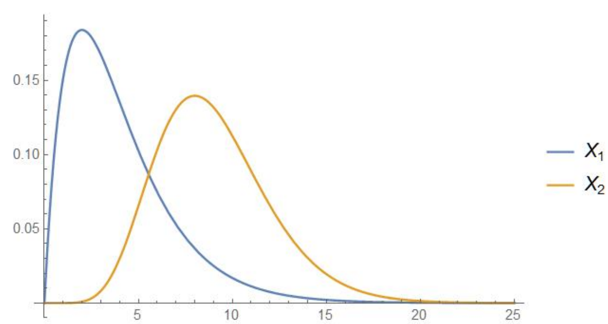

Figure 6.1: PDF's of X_1 and X_2

In this case as well as in all the other cases presented in this chapter, it is seen that the empirical and approximated cdf's are indeed in very close agreement.

Now consider the case where

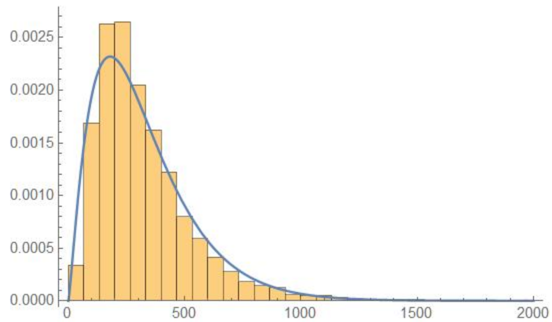

Figure 6.2: Histogram and the base density

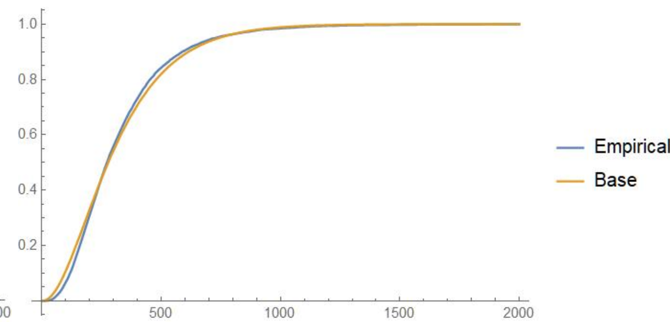

Figure 6.3: Empirical CDF and the base CDF

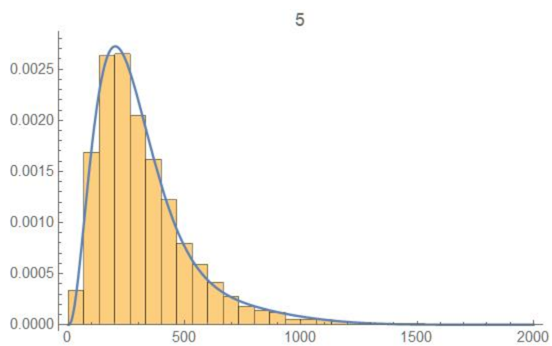

Figure 6.4: Histogram and $f1$

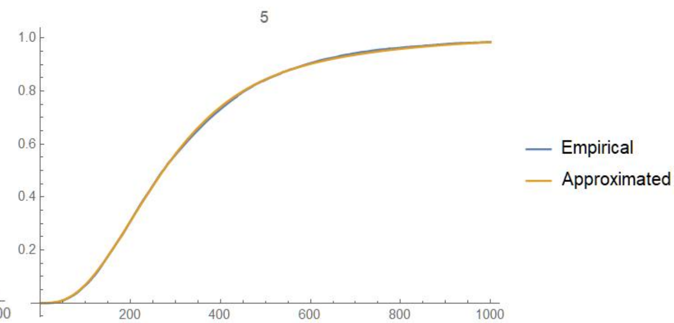

Figure 6.5: Empirical CDF and F1

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

and X_1, X_2, X_3, X_4 are independent gamma distributed random variables with respective parameters (2,2), (9,1), (2,1), (12,1).

We determined an approximant to the the distribution of $Y = (X_1, X_2, X_3, X_4)A(X_1, X_2, X_3, X_4)'$ as previously explained. Since A is a positive definite matrix and the histogram is positively skewed, we chose a gamma density as base density. The distributional results are illustrated graphically in Figures 6.6 to 6.10.

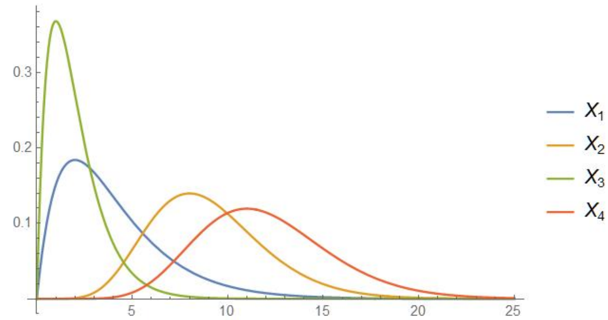

Figure 6.6: PDF's of X_i's

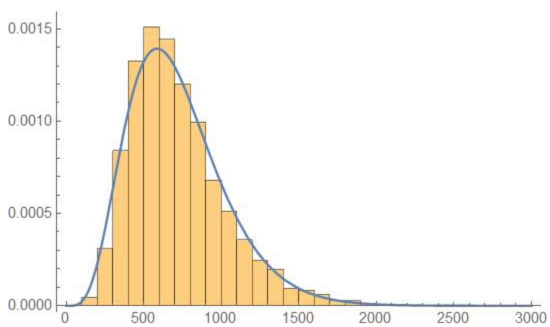

Figure 6.7: Histogram and the base density of Y

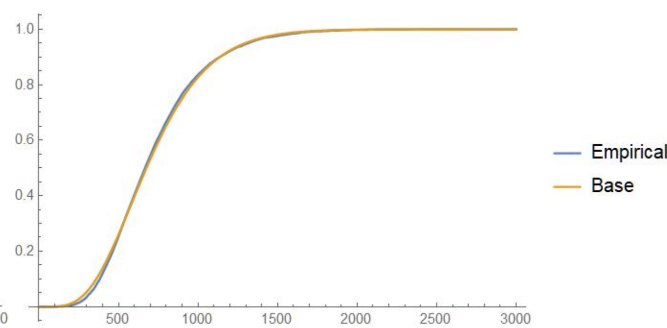

Figure 6.8: Empirical CDF and the base CDF

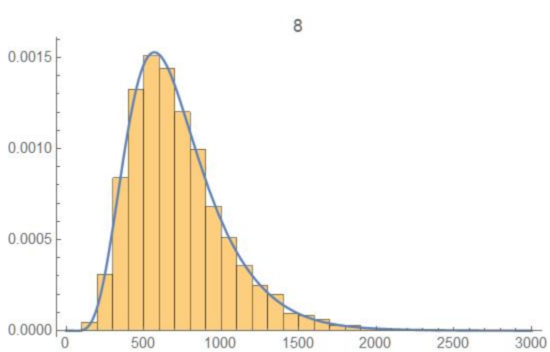

Figure 6.9: Histogram and $f1$

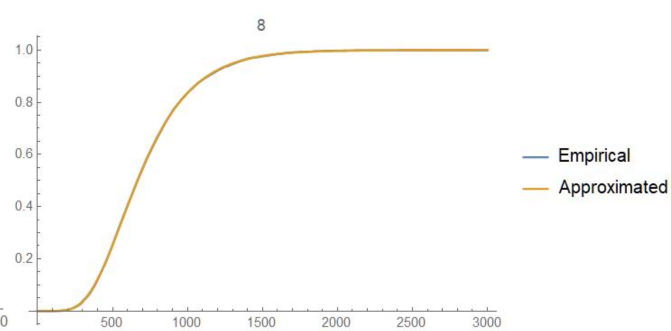

Figure 6.10: Empirical CDF and $F1$

6.3.2 Quadratic forms in inverse Gaussian random variables

The pdf of inverse Gaussian distribution with parameters (λ, μ) is

$$f(x) = \left(\frac{\lambda}{2\,\pi x^3}\right)^{1/2} e^{-\lambda\,(x-\mu)^2/(2\,\mu^2 x)}$$

Consider inverse Gaussian distributed random variables X_1 and X_2 with parameters (2,5) and (3,6). Let

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix},$$

and $Y = (X_1, X_2)A(X_1, X_2)'$.

In this case, we use an inverse Gaussian density as the base density for our approximation. The parameters are determined by setting the first two moments of the inverse Gaussian distribution equal to those of the quadratic form, cf Figures 6.11 to 6.15.

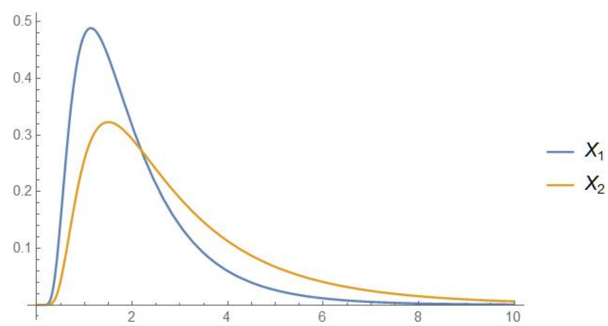

Figure 6.11: PDF's of X_1 and X_2

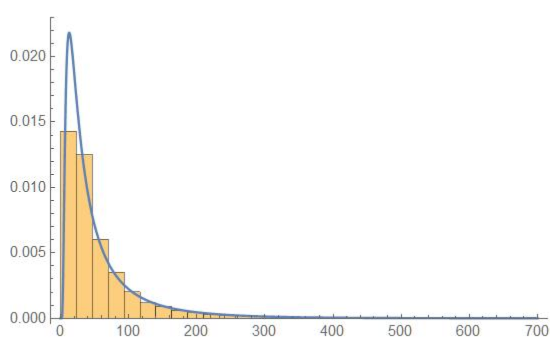

Figure 6.12: Histogram and the base density of Y

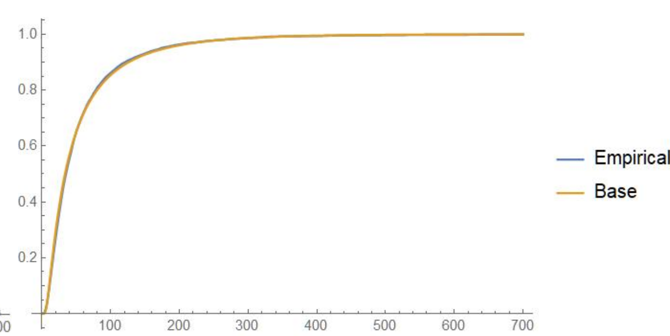

Figure 6.13: Empirical CDF and the base CDF

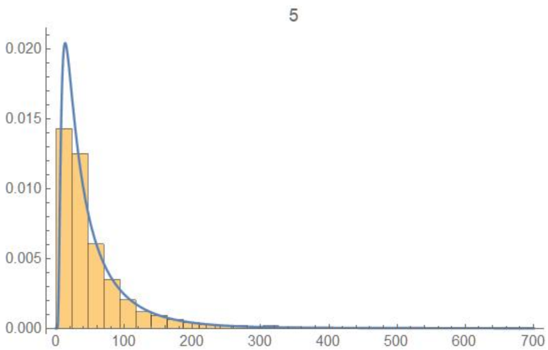

Figure 6.14: Histogram and $f1$

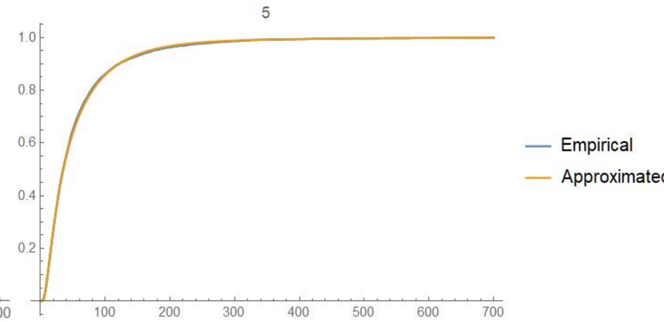

Figure 6.15: Empirical CDF and $F1$

Now consider the case where

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

and X_1, X_2, X_3, X_4 are independent gamma distributed random variables with respective parameters (2,5), (3,6), (2,2), (3,4).

We determined an approximant to the distribution of $Y = (X_1, X_2, X_3, X_4)A(X_1, X_2, X_3, X_4)'$. A is a positive definite matrix and the histogram suggests that an inverse Gaussian distribution would be suitable as base density. The distributional results are illustrated graphically in Figures 6.16 to 6.20.

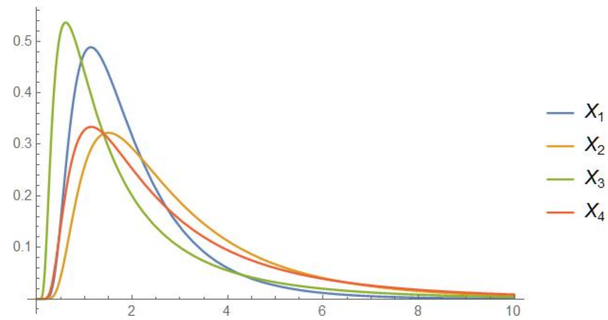

Figure 6.16: PDF's of X_i's

As the following subsections illustrate, the methodology also applies in the case of discrete

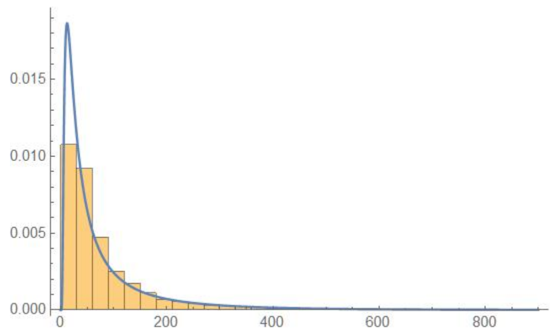

Figure 6.17: Histogram and the base density of Y

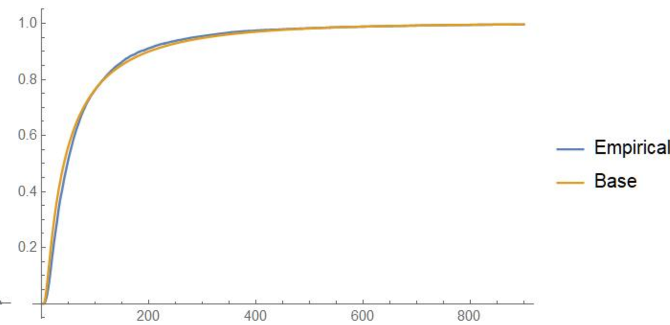

Figure 6.18: Empirical CDF and the base CDF

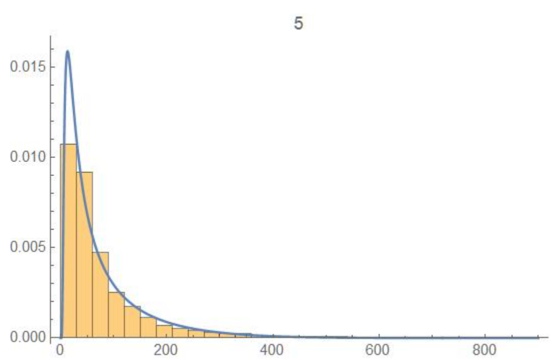

Figure 6.19: Histogram and $f1$

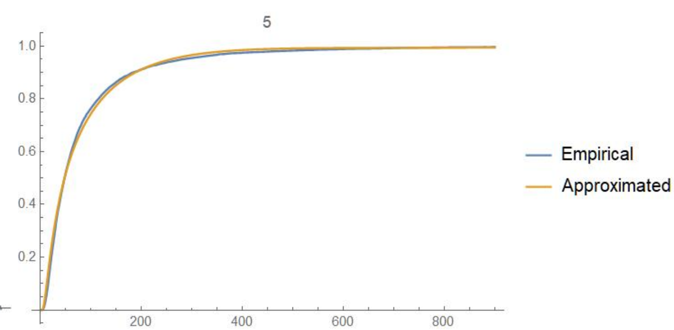

Figure 6.20: Empirical CDF and $F1$

random variables.

6.3.3 Quadratic forms in binomial random variables

The pmf of a binomial distribution with parameters (n, p) is

$$p(k) = \binom{n}{k} p^k (1-p)^{n-k}.$$

Consider two independently distributed binomial random variables X_1 and X_2 with parameters (20,1/4) and (30,1/2). Let

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$$

and $Y = (X_1, X_2)A(X_1, X_2)'$. It may prove difficult to obtain the exact probability density function of Y; however, an accurate approximation can be determined. We use the sum squared difference (SSD) between the empirical cumulative distribution function and the approximated cumulative distribution function as our criterion to determine the degree of the polynomial adjustment in the case of quadratic forms in discrete r.v.'s.

We utilize a gamma density as the base density of our approximation and determine its parameters by setting the first two moments of the gamma distribution and those of the quadratic form to be equal. The distributional results are illustrated graphically in Figures 6.21 to 6.25.

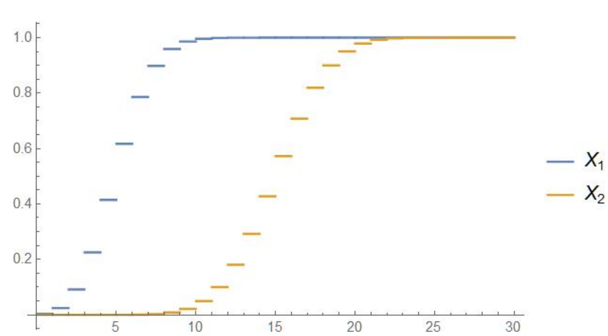

Figure 6.21: CDF's of X_1 and X_2

Now consider the case where

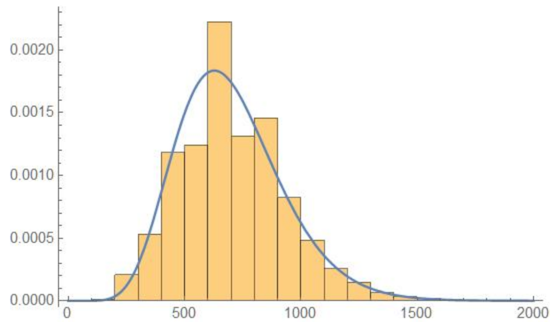

Figure 6.22: Histogram and the base density of Y

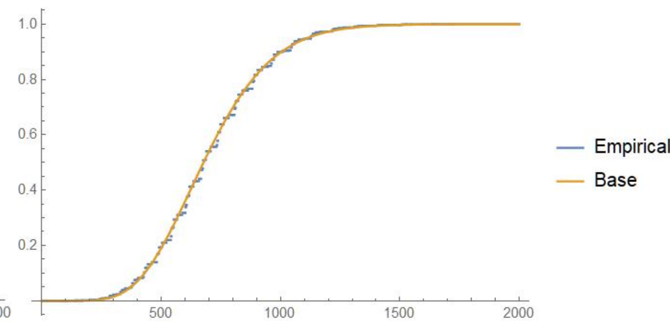

Figure 6.23: Empirical CDF and the base CDF

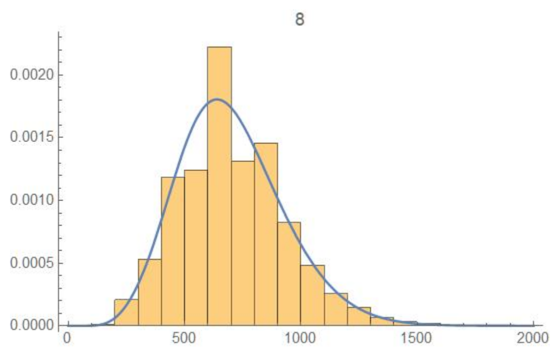

Figure 6.24: Histogram and $f1$

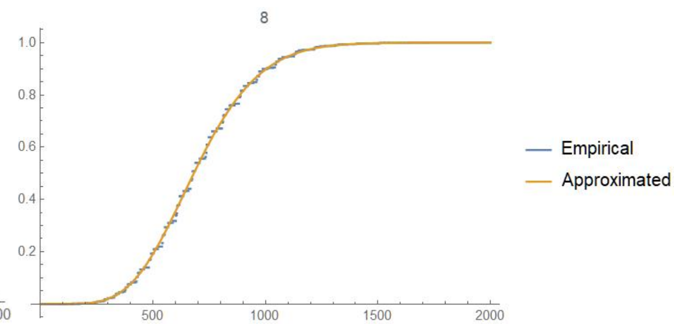

Figure 6.25: Empirical CDF and $F1$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

and X_1, X_2, X_3, X_4 are independent binomial distributed random variables with respective parameters (20,1/4), (30,1/2), (20,1/2), (30,1/3).

We determined an approximant to the distribution of $Y = (X_1, X_2, X_3, X_4)A(X_1, X_2, X_3, X_4)'$ as previously explained. A is a positive definite matrix and in this case, a gamma density function is appropriate as base density, cf Figures 6.26 to 6.30.

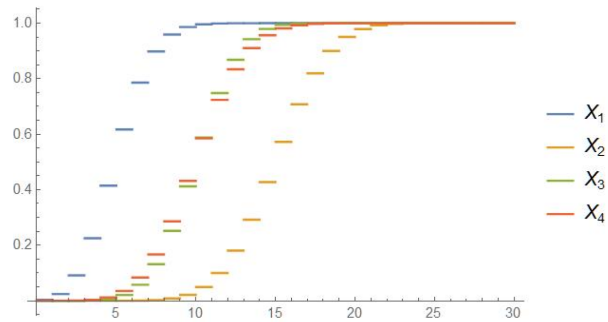

Figure 6.26: CDF's of X_i's

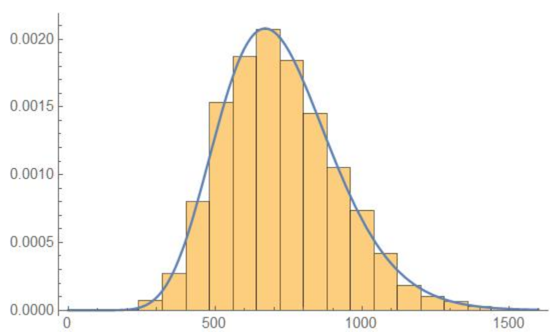

Figure 6.27: Histogram and the base density of Y

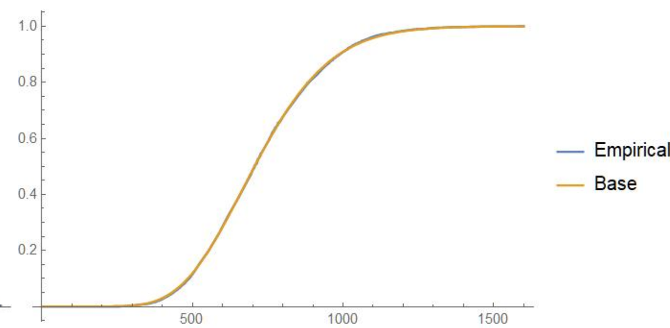

Figure 6.28: Empirical CDF and the base CDF

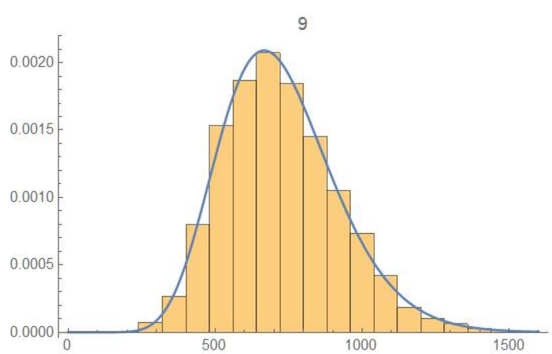

Figure 6.29: Histogram and $f1$

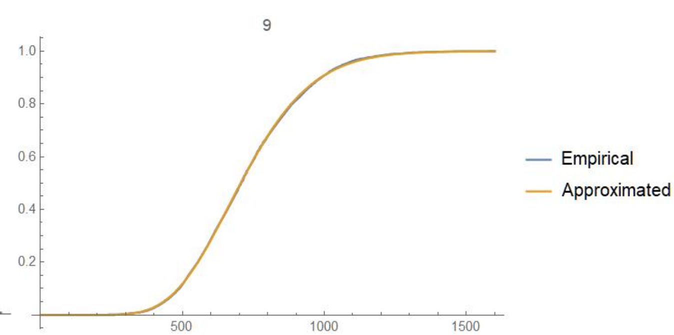

Figure 6.30: Empirical CDF and $F1$

6.3.4 Quadratic forms in Poisson random variables

The pmf of Poisson distribution with parameter λ is

$$p(k) = \frac{\lambda^k e^{-\lambda}}{k!}.$$

Consider binomial distributed random variables X_1 and X_2 with parameters 3 and 5. Let

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix},$$

and $Y = (X_1, X_2)A(X_1, X_2)^{'}$.

We use a gamma density as the base density of our approximation and determine its parameters by setting the first two moments of the gamma distribution equal to those of the quadratic form, cf Figures 6.31 to 6.35.

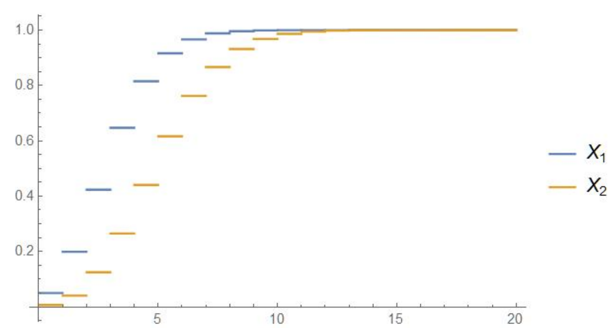

Figure 6.31: CDF's of X_1 and X_2

Now consider the case where

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

and X_1, X_2, X_3, X_4 are independent Poisson distributed random variables with parameters 3, 4,

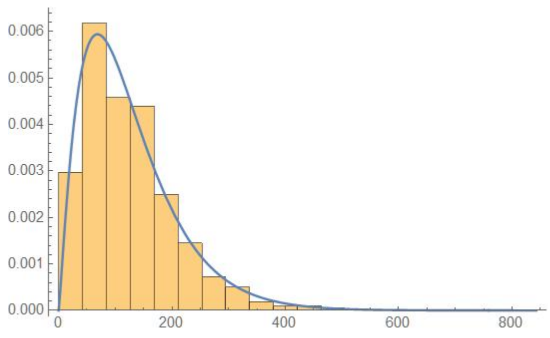

Figure 6.32: Histogram and the base density of Y

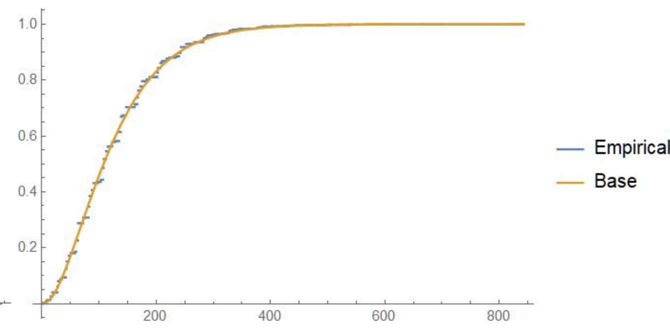

Figure 6.33: Empirical CDF and the base CDF

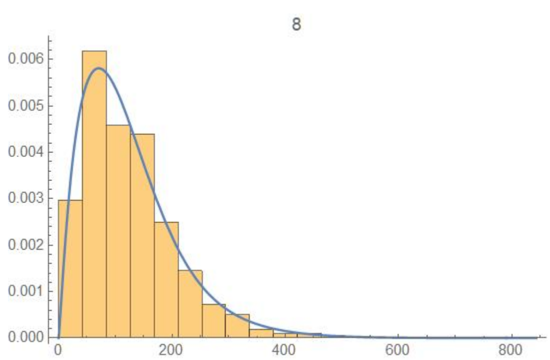

Figure 6.34: Histogram and $f1$

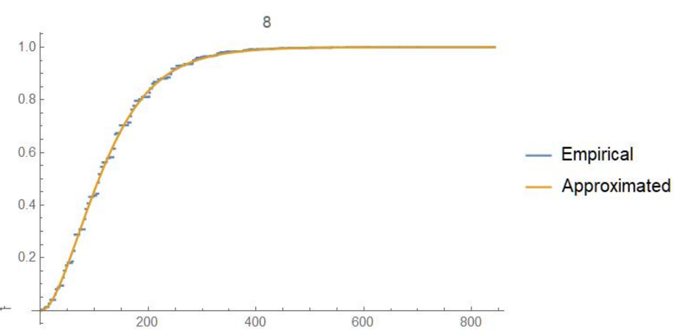

Figure 6.35: Empirical CDF and $F1$

5 and 6.

We determined an approximat to the distribution of $Y = (X_1, X_2, X_3, X_4)A(X_1, X_2, X_3, X_4)'$ as previously explained. A is a positive definite matrix and in this case, a gamma density function is appropriate as base density. The distributional results are illustrated graphically in Figures 6.36 to 6.40.

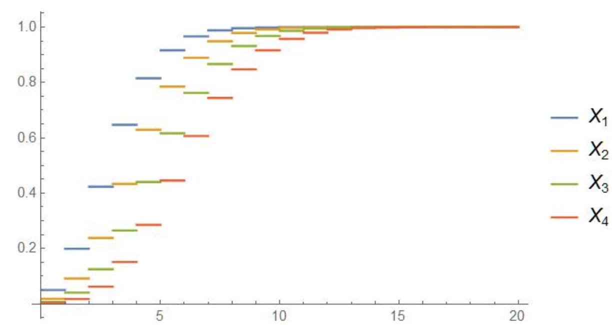

Figure 6.36: CDF's of X_i's

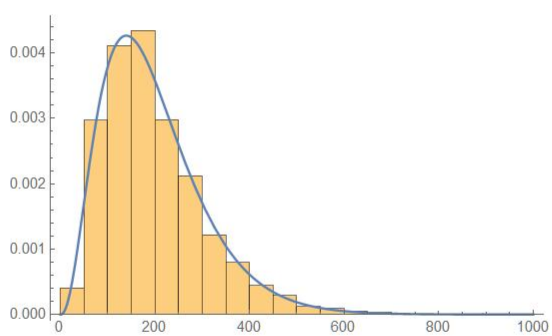

Figure 6.37: Histogram and the base density of Y

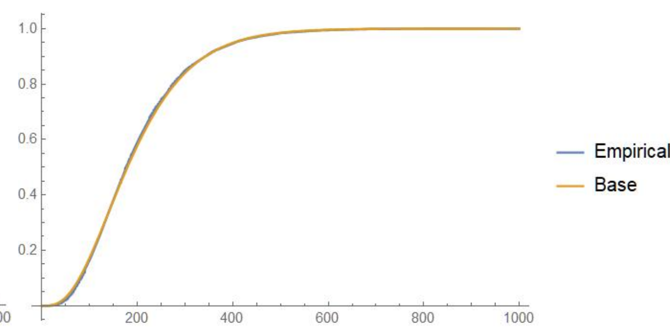

Figure 6.38: Empirical CDF and the base CDF

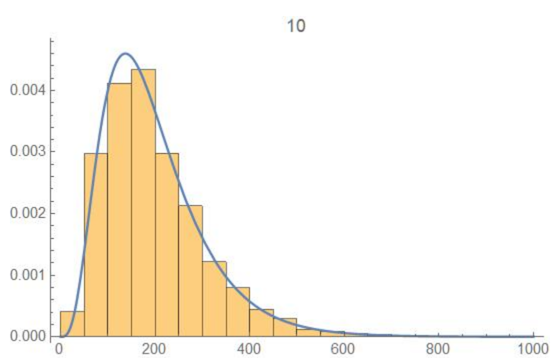

Figure 6.39: Histogram and $f1$

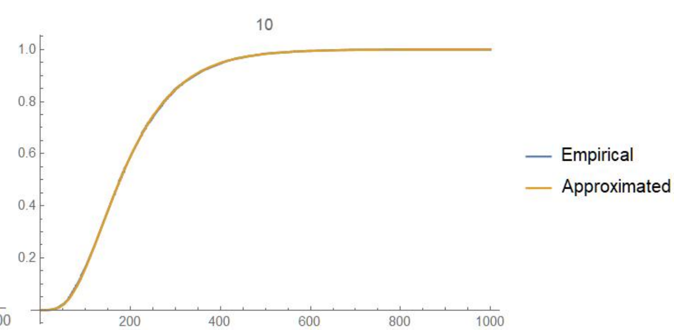

Figure 6.40: Empirical CDF and $F1$

In general, the distribution of quadratic forms in various types of random variables can be efficiently approximated by applying the methodology introduced in Provost (2005) in conjunction with the symbolic evaluation of their moments.

Bibliography

[1] Barrio, E. D., Gine, E. and Utzet, F. (2005). Asymptotics for L_2 functionals of the empirical quantile process, with applications to tests of fit based on weighted Wasserstein distances. *Bernoulli* **11**, 131–189.

[2] Donald, S. G. and Paarsch, H. J. (2002). Superconsistent estimation and inference in structural econometric models using extreme order statistics. *Journal of Econometrics* **109**, 305–340.

[3] Mathai, A. M. and Provost, S. B. (1992). *Quadratic Forms in Random Variables*. Marcel Dekker, New York.

[4] Provost, S. B. (2005). Moment-based density approximants. *The Mathematica Journal* **9**, 727–756.

[5] Provost, S. B., Jiang, M., and Ha, H.-T. (2009). Moment-based approximations of probability mass functions with applications involving rank-sum test statistics. *Communications in Statistics - Theory and Methods* **38**, 1969–1981.

[6] Whittle, P. (1960). Quadratic forms in Poisson and multi-nomial variables. *Journal of the Australian Mathematical Society* **1**, 233–240.

Chapter 7

Concluding Remarks and Future Work

A variety of problems of interest have been tackled in this thesis. All the methodologies rely on the population moments of a distributions or the sample moments associated with a sample of observations. Since the sample moments associated with a data set contain all the distributional information available, it is appropriate to make use of them to address statistical problems. The population and sample moments are respectively employed to obtain density approximants and estimates. It was explained that moment-based methodologies are well suited to model and analyse 'big data' since once the sample moments have been evaluated, the techniques require the same amount of computing resources irrespective of the sample size. Incidentally, the symbolic computation software package Mathematica was utilized to execute the calculations and produce the graphs.

Several types of density estimates that rely on moments were compared with respect to their accuracy when approximating or estimating certain density functions. It was determined that the product of a base density function and a polynomial adjustment often produces the best results. Monte Carlo and quasi Monte Carlo samples of fixed sizes were compared as to how they depict certain underlying distributions in terms of several criteria. We found that samples obtained from the percentiles corresponding to equidistant cumulative distribution function values frequently turn out to be most representative. A novel methodology that is based on the difference of certain distributional percentiles after applying an arctangent transformation was introduced to classify the tail behaviour of probability laws. Our categories generally agree with those published in the literature but the proposed criterion proves much simpler to apply. Additionally, the distribution of quadratic forms in Poisson, inverse Gaussian, binomial and gamma random variables were approximated from their moments which were evaluated via symbolic computations. Various applications were pointed out including those related to contingency tables, certain tests statistics and generalized linear models.

This thesis can be viewed as a preliminary study on the various topics that were tackled therein. Further work is planned at the doctoral level to obtain more definite results by for example considering additional distributions and samples in connection with sample representativeness and the various types of moment-based density estimates or approximants that have been considered. The idea to approximate a density estimate by a differentiated logdensity approximant in order to obtain a smooth bona fide density estimate will be pursued as well. We are also planning to assess the relative amount of information contained in a given sample moment by making use of the one-to-one correspondence between a set of n observations and its first n sample moments. The determination of the tail behaviour associated with samples of observations will be investigated, this aspect being informed by our work on the tail classification of theoretical distributions. The results on quadratic forms will be extended to those involving vectors of other types of random variables and to ratios thereof. Then, the application of these results to specific inferential problems arising in Statistics will be considered and numerical studies will be carried out.

Appendix A

Mathematica Code

The Mathematica code utilized for implementing the main numerical examples presented in this dissertation is included in this appendix. The evaluation of the moments which are utilized for approximating or estimating density functions is carried out with rational numbers so as to prevent any loss of precision.

Chapter 2

2.9.1 Mixture of beta pdf’s

```
α1 = 8; β1 = 12; α2 = 3; β2 = 15;
ClearAll[f, F]
f[y_] :=
 f[y] = (PDF[BetaDistribution[α1, β1], y] + PDF[BetaDistribution[α2, β2], y]) / 2


ClearAll[μ, b]
μ[h_] :=
 μ[h] = (Moment[BetaDistribution[α1, β1], h] + Moment[BetaDistribution[α2, β2], h]) / 2
μ[1]; μ[2];
α = μ[1] (μ[1] - μ[2]) / (μ[2] - μ[1]^2);
β = (1 - μ[1]) α / μ[1];
{α, β} // N
b[y_] := b[y] = PDF[BetaDistribution[α, β], y]
Plot[{f[y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "Base density"}, PlotRange → All]
```

f1[*x*] = *b*[*x*] p1[*x*], n=10

```
n = 10;
ClearAll[p1, f1]
M1 = Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := b[y] p1[y]
ISD = NumberForm[NIntegrate[(f[y] - f1[y])^2, {y, 0, 1}], 6]
Plot[{f[y], f1[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f1"}, PlotLabel → {n}]
```

f2[*x*] = p2[*x*], n=10

```
n = 10;
ClearAll[p2, f2]
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Table[μ[i], {i, 0, n}];
coe2 = Inverse[M2].μ2;
p2[y_] := Sum[coe2[[i]] y^(i-1), {i, 1, n+1}]
k2 = Rationalize[NIntegrate[p2[y], {y, 0, 1}], 10^-10];
f2[y_] := p2[y] / k2
ISD = NumberForm[NIntegrate[(f[y] - f2[y])^2, {y, 0, 1}], 6]
Plot[{f[y], f2[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f2"}, PlotLabel → {n}]
```

f3[*x*] = *b*[*x*] + p3[*x*], n=10

```
n = 10;
ClearAll[f31, p3, f3]
M3 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ3 = Table[μ[h] - Moment[BetaDistribution[α, β], h], {h, 0, n}];
coe3 = Inverse[M3].μ3;
p3[x_] := Sum[coe3[[i]] x^(i-1), {i, 1, n+1}];
f31[y_] := b[y] + p3[y];
k3 = Rationalize[NIntegrate[f31[y], {y, 0, 1}], 10^-10];
f3[y_] := f31[y] / k3
ISD = NumberForm[NIntegrate[(f[y] - f3[y])^2, {y, 0, 1}], 6]
Plot[{f[y], f3[y]}, {y, 0, 1}, PlotLegends → {"Exact pdf", "f3"},
 PlotStyle → {Gray, Dashed}, PlotLabel → {n}]
```

f4[*x*] = *b*[*x*] $e^{p4[x]}$, n=13

```
n = 13;
ClearAll[f40, μ40, p4, f41, f4];
f40[y_] := Log[f[y]] - Log[b[y]];
M4 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ40[h_] := Rationalize[NIntegrate[y^h f40[y], {y, 0, 1}], 10^-10]
μ4 = Table[μ40[i], {i, 0, n}];
coe4 = Inverse[M4].μ4;
p4[x_] := Sum[coe4[[i]] x^(i-1), {i, 1, n+1}];
f41[y_] := b[y] Exp[p4[y]];
k4 = Rationalize[NIntegrate[f41[y], {y, 0, 1}], 10^-10];
f4[y_] := f4[y] = f41[y] / k4;

ISD = NumberForm[NIntegrate[(f[y] - f4[y])^2, {y, 0, 1}], 6]
Plot[{f[y], f4[y]}, {y, 0, 1}, PlotRange → All,
 PlotStyle → {Gray, Dashed}, PlotLegends → {"Exact pdf", "f4"}, PlotLabel → {n}]
```

f5[x] = k$e^{\int_\alpha^x r[y]\,dy}$, *v*=17

```
ν = 17;
δ = 0;
ClearAll[r, IR, fν, f5];
M5 = Rationalize[Table[μ[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := Exp[IR[x]];
k5 = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f5[x_] := fν[x] / k5;
ISD = NumberForm[NIntegrate[(f[y] - f5[y])^2, {y, 0, 1}], 6]
Plot[{f[y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
 PlotStyle → {Gray, Dashed}, PlotLegends → {"Exact pdf", "f5"}, PlotLabel → {ν, δ}]
```

$f6[x] = c x^{\alpha-1}(1-x)^{\beta-1} e^{\int_a^x p6[y]\,dy}$, $\nu = 12$

```
ν = 12;
ClearAll[f60, μ60, p6, IP6, fν6, f6];
f60[y_] := f[y]/b[y];
δ = 0;

μ60[h_] := Rationalize[NIntegrate[y^h f60[y], {y, 0, 1}], 10^-100];
M6 = Table[μ60[i + j], {i, 0, ν}, {j, 0, ν}];
μ6 = Prepend[Table[-h * μ60[h - 1], {h, 1, ν}], 0];
coe6 = Inverse[M6].μ6;
p6[x_] := Sum[coe6[[i]] x^(i-1), {i, 1, ν+1}];
IP6[x_] := Integrate[p6[y], {y, 0, x}, Assumptions → 0 < x < 1];
fν6[x_] := Exp[IP6[x]] b[x];
k6 = Rationalize[NIntegrate[fν6[x], {x, 0, 1}], 10^-10];
f6[x_] := fν6[x] / k6;
ISD = NumberForm[NIntegrate[(f[y] - f6[y])^2, {y, 0, 1}], 6]
Plot[{f[y], Evaluate[f6[y]]}, {y, 0, 1}, PlotRange → All,
 PlotStyle → {Gray, Dashed}, PlotLegends → {"Exact pdf", "f6"}, PlotLabel → {ν, δ}]
```

$f7[x] = w\, b(x) + (1-w)\, e^{\int_a^x p7[y]\,dy}$, w=2/10, $\nu = 15$

```
ν = 15;
w = 2 / 10;
δ = 0;
ClearAll[f70, μ70, r, IR, fν, f7];
f70[y_] := 1/(1 - w) (f[y] - w * b[y]);
μ70[h_] := 1/(1 - w) (μ[h] - w * Moment[BetaDistribution[α, β], h]);
M7 = Table[μ70[i + j], {i, 0, ν}, {j, 0, ν}];
μ7 = Prepend[Table[-h * μ70[h - 1], {h, 1, ν}], 0];
coe7 = Inverse[M7].μ7;
r[x_] := Sum[coe7[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := Rationalize[Integrate[r[y], {y, 0, x}], 10^-10]
fν[x_] := w b[x] + (1 - w) Exp[IR[x]]
k7 = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f7[x_] := fν[x] / k7
ISD = NumberForm[NIntegrate[(f[y] - f7[y])^2, {y, 0, 1}], 6]
Plot[{f[y], Evaluate[f7[y]]}, {y, 0, 1}, PlotRange → All,
 PlotStyle → {Gray, Dashed}, PlotLegends → {"Exact pdf", "f7"}, PlotLabel → {ν, δ}]
```

2.9.2 Mixture of gamma pdf's (using the transformation $Z = \frac{Y}{Y+1}$)

```
α1 = 2; β1 = 2; α2 = 9; β2 = 1;
ClearAll[f00]
f00[y_] :=
 f00[y] = (PDF[GammaDistribution[α1, β1], y] + PDF[GammaDistribution[α2, β2], y]) / 2
Plot[f00[y], {y, 0, 25}, PlotRange → All,
 PlotLegends → {"Mixture of gamma"}, PlotStyle → Gray]
𝒟 = ProbabilityDistribution[f00[y], {y, 0, ∞}];
ClearAll[f0]
σ = StandardDeviation[𝒟];
ClearAll[f0]
f0[y_] := f0[y] = σ * f00[σ * y]
ClearAll[f]
f[y_] := f[y] = 1/(1 - y)^2 f0[1/(1 - y) - 1]
ClearAll[μ, b]
μ[h_] := μ[h] = NIntegrate[y^h f[y], {y, 0, 1}]
α = μ[1] (μ[1] - μ[2])/(μ[2] - μ[1]^2);
β = (1 - μ[1]) α/μ[1];
b[y_] := b[y] = PDF[BetaDistribution[α, β], y]
Plot[{f[y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "Base density"}, PlotRange → All]
```

f1[x] = b[x] p_1[x], n=25

```
n = 25;
ClearAll[p1, f1]
M1 =
  Rationalize[Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}], 10^-100];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := p1[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = b[y] p1[y]
ClearAll[bf1, bf2]
bf1[y_] := bf1[y] = 1/2 (f1[y] + Abs[f1[y]])
kb1 = NIntegrate[bf1[y], {y, 0, 1}];
bf2[y_] := bf2[y] = bf1[y] / kb1
Plot[{f[y], bf2[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f1"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - bf2[y])^2, {y, 0, 1}], 6]
```

$f2[x] = p_1[x]$, n=12

```
n = 12;
ClearAll[p2, f2]
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Table[μ[i], {i, 0, n}];
coe2 = Inverse[M2].μ2;
p2[y_] := Sum[coe2[[i]] y^(i-1), {i, 1, n+1}]
k2 = Rationalize[NIntegrate[p2[y], {y, 0, 1}], 10^-10];
f2[y_] := p2[y] / k2
a1 = FindRoot[f2[y] == 0, {y, 0.005}][[1, 2]];
b1 = FindRoot[f2[y] == 0, {y, 0.8}][[1, 2]];
ClearAll[bf1, bf2]
bf1[y_] := bf1[y] = If[a1 < y < b1, f2[y], 0]
kb1 = NIntegrate[f21[y], {y, a1, b1}];
bf2[y_] := bf2[y] = bf1[y] / kb1
Plot[{f[y], bf2[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f2"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - bf2[y])^2, {y, 0, 1},
   WorkingPrecision → 100, AccuracyGoal → 20, PrecisionGoal → 10], 6]
```

$f3[x] = b[x] + p_2[x]$, n=12

```
n = 12;
ClearAll[μ30, p3, f31, f3]
M3 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ30[h_] := μ30[h] = μ[h] - Moment[BetaDistribution[α, β], h]
μ3 = Table[μ30[i], {i, 0, n}];
coe3 = Inverse[M3].μ3;
p3[x_] := p3[x] = Sum[coe3[[i]] x^(i-1), {i, 1, n+1}]
f31[y_] := f31[y] = b[y] + p3[y]
k3 = Rationalize[NIntegrate[f31[y], {y, 0, 1}], 10^-10];
f3[y_] := f3[y] = f31[y] / k3
a1 = FindRoot[f3[y] == 0, {y, 0.05}][[1, 2]];
b1 = FindRoot[f3[y] == 0, {y, 0.8}][[1, 2]];
ClearAll[bf1, bf2]
bf1[y_] := bf1[y] = If[a1 < y < b1, f3[y], 0]
kb1 = NIntegrate[bf1[y], {y, a1, b1}];
bf2[y_] := bf2[y] = bf1[y] / kb1
Plot[{f[y], bf2[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f3"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - bf2[y])^2, {y, 0, 1}], 6]
```

$f5[x] = ke^{\int_{\alpha}^{x} r[y]\, dy}$, v=11

```
ν = 11;
δ = 0;
ClearAll[r, IR, fν, f5];
M5 = Table[μ[i + j], {i, 0, ν}, {j, 0, ν}];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := fν[x] = Exp[IR[x]];
k = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f5[x_] := f5[x] = fν[x] / k;
Plot[{f[y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
 PlotStyle → {Gray, Dashed}, PlotLegends → {"Exact pdf", "f5"}, PlotLabel → {ν, δ}]
ISD = NumberForm[NIntegrate[(f[y] - f5[y])^2, {y, 0, 1}], 6]
```

$f7[x] = w\, b(x) + (1 - w)\, e^{\int_{\alpha}^{x} p7[y]\, dy}$, w=2/10, v=13

```
ν = 13;
δ = 0;
w = 1 / 10;
ClearAll[μ70, r, IR, fν, f7];
μ70[h_] := μ70[h] = 1/(1 - w) (μ[h] - w Moment[BetaDistribution[α, β], h]);
M7 = Rationalize[Table[μ70[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ7 = Prepend[Table[-h * μ70[h - 1], {h, 1, ν}], 0];
coe7 = Inverse[M7].μ7;
r[x_] := r[x] = Sum[coe7[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := w b[x] + (1 - w) Exp[IR[x]];
k = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f7[x_] := f7[x] = fν[x] / k;
Plot[{f[x], f7[x]}, {x, 0, 1}, PlotLegends → {"Exact pdf", "f7"},
 PlotLabel → {ν, δ}, PlotStyle → {Gray, Dashed}, PlotRange → All]
ISD = NumberForm[NIntegrate[(f[y] - f7[y])^2, {y, 0, 1}], 6]
```

2.9.2 Mixture of gamma pdf’s (truncated distribution)

```
α1 = 2; β1 = 2; α2 = 9; β2 = 1;
ClearAll[f0]
f0[y_] :=
 f0[y] = (PDF[GammaDistribution[α1, β1], y] + PDF[GammaDistribution[α2, β2], y]) / 2
Plot[f0[y], {y, 0, 25}, PlotRange → All,
 PlotLegends → {"Truncated gamma"}, PlotStyle → Gray]
F0[y_] := (CDF[GammaDistribution[α1, β1], y] + CDF[GammaDistribution[α2, β2], y]) / 2
a0 = 25;
F0[a0] // N
ClearAll[f]
f[y_] := f[y] = a0 * f0[a0 * y]
k0 = NIntegrate[f[y], {y, 0, 1}];
ClearAll[μ, b]
μ[h_] := μ[h] = NIntegrate[y^h f[y], {y, 0, 1}]
α = μ[1] (μ[1] - μ[2]) / (μ[2] - μ[1]^2);
β = (1 - μ[1]) α / μ[1];
b[y_] := b[y] = PDF[BetaDistribution[α, β], y]
Plot[{f[y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "Base density"}, PlotRange → All]
```

f1[*x*] = *b*[*x*] p1[*x*], n=7

```
n = 7;
ClearAll[p1, f10, f1]
M1 =
  Rationalize[Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}], 10^-100];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f10[y_] := b[y] p1[y]
k1 = NIntegrate[f10[y], {y, 0, 1}];
f1[y_] := f10[y] * k0 / k1
Plot[{f[y], f1[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f1"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - f1[y])^2, {y, 0, 1}], 6]
```

f2[x] = p2[x], n=11

```
n = 11;
ClearAll[p2, f2]
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Table[μ[i], {i, 0, n}];
coe2 = Inverse[M2].μ2;
p2[y_] := Sum[coe2[[i]] y^(i-1), {i, 1, n+1}]

k2 = Rationalize[NIntegrate[p2[y], {y, 0, 1}], 10^-10];
f2[y_] := p2[y] * k0/k2
Plot[{f[y], f2[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f2"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - f2[y])^2, {y, 0, 1}], 6]
```

f3[x] = b[x] + p3[x], n=12

```
n = 12;
ClearAll[f30, μ30, p3, f31, f3]
f30[y_] := f[y] - b[y]
M3 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ30[h_] := μ[h] - Moment[BetaDistribution[α, β], h]
μ3 = Table[μ30[i], {i, 0, n}];
coe3 = Inverse[M3].μ3;
p3[x_] := Sum[coe3[[i]] x^(i-1), {i, 1, n+1}]
f31[y_] := b[y] + p3[y]
k3 = Rationalize[NIntegrate[f31[y], {y, 0, 1}], 10^-10];
f3[y_] := f31[y] * k0/k3
Plot[{f[y], f3[y]}, {y, 0, 1}, PlotLegends → {"Exact pdf", "f3"},
 PlotStyle → {Gray, Dashed}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - f3[y])^2, {y, 0, 1}], 6]
```

f4[x] = b[x] $e^{p4[x]}$, n=11

```
n = 11;
ClearAll[f40, μ40, p4, f41, f4]
f40[y_] := f40[y] = Log[f[y]] - Log[b[y]];
M4 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ40[h_] := μ40[h] = Rationalize[NIntegrate[y^h f40[y], {y, 0, 1}], 10^-10]
μ4 = Table[μ40[i], {i, 0, n}];
coe4 = Inverse[M4].μ4;
p4[x_] := p4[x] = Sum[coe4[[i]] x^(i-1), {i, 1, n+1}]
f41[y_] := f41[y] = b[y] Exp[p4[y]]
k4 = Rationalize[NIntegrate[f41[y], {y, 0, 1}], 10^-10];
f4[y_] := f4[y] = f41[y] * k0/k4
Plot[{f[y], f4[y]}, {y, 0, 1}, PlotRange → All,
 PlotStyle → {Gray, Dashed}, PlotLegends → {"Exact", "f4"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - f4[y])^2, {y, 0, 1}], 6]
```

f5[x] = k$e^{\int_\alpha^x r[y]\,dy}$, ν=17

```
ν = 17;
δ = 0;
ClearAll[r, IR, fν, f5]
M5 = Rationalize[Table[μ[i + j], {i, 0, ν}, {j, 0, ν}], 10^-10];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10]
fν[x_] := fν[x] = Exp[IR[x]]
k5 = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f5[x_] := f5[x] = fν[x] * k0/k5
Plot[{f[y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
 PlotStyle → {Gray, Dashed}, PlotLegends → {"Exact pdf", "f5"}, PlotLabel → {ν, δ}]
ISD = NumberForm[NIntegrate[(f[y] - f5[y])^2, {y, 0, 1}], 6]
```

f7[x] $= w\,b(x) + (1-w)\,e^{\int_a^x \mathrm{p7}[y]\,dy}$, w=2/10, ν =21

```
ν = 21;
δ = 0;
w = 1 / 10;
ClearAll[f70, μ70, r, IR, fν, f7];
f70[y_] := f70[y] = 1/(1 - w) (f[y] - w * b[y]);
μ70[h_] := μ70[h] = 1/(1 - w) (μ[h] - w Moment[BetaDistribution[α, β], h]);
M7 = Rationalize[Table[μ70[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ7 = Prepend[Table[-h * μ70[h - 1], {h, 1, ν}], 0];
coe7 = Inverse[M7].μ7;
r[x_] := r[x] = Sum[coe7[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := w b[x] + (1 - w) Exp[IR[x]];
k = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f7[x_] := f7[x] = fν[x] / k;
Plot[{f[x], f7[x]}, {x, 0, 1}, PlotLegends → {"Exact pdf", "f7"},
 PlotLabel → {ν, δ}, PlotStyle → {Gray, Dashed}, PlotRange → All]
ISD = NumberForm[NIntegrate[(f[y] - f7[y])^2, {y, 0, 1}], 6]
```

2.9.3 Mixture of normal pdf's (using the transformation $Z = \frac{\text{acrtan}\,Y}{\pi} + \frac{1}{2}$)

```
μ1 = 0; μ2 = 5; σ1 = 1; σ2 = 3;
ClearAll[f00]
f00[y_] :=
 f00[y] = (PDF[NormalDistribution[μ1, σ1], y] + PDF[NormalDistribution[μ2, σ2], y]) / 2
𝒟 = ProbabilityDistribution[f00[y], {y, -∞, ∞}];
σ = StandardDeviation[𝒟];
ClearAll[f0]
f0[y_] := f0[y] = σ * f00[σ * y]
ClearAll[f]
f[y_] := f[y] = f0[Tan[π (y - 1 / 2)]] π Sec[π (-1/2 + y)]^2
ClearAll[μ, b]
μ[h_] := μ[h] = NIntegrate[y^h f[y], {y, 0, 1}, WorkingPrecision → 100]
α = μ[1] (μ[1] - μ[2])/(μ[2] - μ[1]^2) // N;
β = (1 - μ[1]) α/μ[1] // N;
b[y_] := b[y] = PDF[BetaDistribution[α, β], y]
Plot[{f[y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "Base density"}, PlotRange → All]
```

f1[x] = b[x] p_1[x], n=23

```
n = 23;
ClearAll[p1, f1]
M1 =
  Rationalize[Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}], 10^-100];
μ1 = Table[μ[h], {h, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := p1[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = b[y] p1[y]
b1 = FindRoot[f1[x] == 0, {x, 0.9}][[1, 2]];
k = NIntegrate[f1[x], {x, 0, b1}, WorkingPrecision → 100];
ClearAll[bf1, bf2]
bf1[x_] := bf1[x] = If[0 < x < b1, f1[x], 0]
bf2[x_] := bf2[x] = bf1[x] / k
Plot[{f[y], bf2[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f1"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - bf2[y])^2, {y, 0, 1}, WorkingPrecision → 100], 6]
```

f2[x] = p_1[x], n=21

```
n = 21;
ClearAll[p2, f2];
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Table[μ[i], {i, 0, n}];
coe2 = Inverse[M2].μ2;
p2[y_] := p2[y] = Sum[coe2[[i]] y^(i-1), {i, 1, n+1}];

k2 = Rationalize[NIntegrate[p2[y], {y, 0, 1}], 10^-10];
f2[y_] := f2[y] = p2[y] / k2;
a1 = FindRoot[f2[x] == 0, {x, 0.2}][[1, 2]];
b1 = FindRoot[f2[x] == 0, {x, 0.9}][[1, 2]];
k = NIntegrate[f2[x], {x, a1, b1}, WorkingPrecision → 100];
ClearAll[bf1, bf2]
bf1[x_] := bf1[x] = If[a1 < x < b1, f2[x], 0]
bf2[x_] := bf2[x] = bf1[x] / k
Plot[{f[y], bf2[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f2"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - bf2[y])^2, {y, 0, 1}, WorkingPrecision → 100], 6]
```

f3[*x*] = *b*[*x*] + p_2[*x*], n=12

```
n = 12;
ClearAll[μ30, p3, f31, f3];
M3 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ30[h_] := μ30[h] = μ[h] - Moment[BetaDistribution[α, β], h];
μ3 = Table[μ30[i], {i, 0, n}];
coe3 = Inverse[M3].μ3;
p3[x_] := p3[x] = Sum[coe3[[i]] x^(i-1), {i, 1, n+1}];
f31[y_] := f31[y] = b[y] + p3[y];
k3 = Rationalize[NIntegrate[f31[y], {y, 0, 1}], 10^-10];
f3[y_] := f3[y] = f31[y] / k3;
a1 = FindRoot[f3[x] == 0, {x, 0.3}][[1, 2]];
b1 = FindRoot[f3[x] == 0, {x, 0.9}][[1, 2]];
k = NIntegrate[f3[x], {x, a1, b1}, WorkingPrecision → 100];
ClearAll[bf1, bf2]
bf1[x_] := bf1[x] = If[a1 < x < b1, f3[x], 0]
bf2[x_] := bf2[x] = bf1[x] / k
Plot[{f[y], bf2[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"Exact pdf", "f3"}, PlotLabel → {n}]
ISD = NumberForm[NIntegrate[(f[y] - bf2[y])^2, {y, 0, 1}, WorkingPrecision → 100], 6]
```

f5[x] = k$e^{\int_\alpha^x r[y] dy}$, *v* =16

```
ν = 16;
δ = 0;
ClearAll[r, IR, fν, f5];
M5 = Rationalize[Table[μ[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := fν[x] = Exp[IR[x]];
k5 = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f5[x_] := f5[x] = fν[x] / k5;
Plot[{f[y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
 PlotStyle → {Gray, Dashed}, PlotLegends → {"Exact", "f5"}, PlotLabel → {ν, δ}]
ISD = NumberForm[NIntegrate[(f[y] - f5[y])^2, {y, 0, 1}, WorkingPrecision → 100], 6]
```

f7[x] = $w\,b(x) + (1-w)\,e^{\int_a^x p7[y]\,dy}$, w=1/10, ν=11

```
ν = 11;
δ = 0;
w = 1 / 10;
ClearAll[f70, μ70, r, IR, fν, f7];
f70[y_] := f70[y] = 1/(1 - w) (f[y] - w * b[y]);
μ70[h_] := μ70[h] = 1/(1 - w) (μ[h] - w Moment[BetaDistribution[α, β], h]);
M7 = Rationalize[Table[μ70[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ7 = Prepend[Table[-h * μ70[h - 1], {h, 1, ν}], 0];
coe7 = Inverse[M7].μ7;
r[x_] := r[x] = Sum[coe7[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := w b[x] + (1 - w) Exp[IR[x]];
k = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f7[x_] := f7[x] = fν[x] / k;
Plot[{f[x], f7[x]}, {x, 0, 1}, PlotLegends → {"Exact pdf", "f7"},
 PlotLabel → {ν, δ}, PlotStyle → {Gray, Dashed}, PlotRange → All]
ISD = NumberForm[NIntegrate[(f[y] - f7[y])^2, {y, 0, 1}], 6]
```

2.9.3 Mixture of normal pdf's (without standardizing)

```
α1 = 0; β1 = 1; α2 = 5; β2 = 3;
ClearAll[f, μ, b]
f[y_] :=
 f[y] = (PDF[NormalDistribution[α1, β1], y] + PDF[NormalDistribution[α2, β2], y]) / 2
μ[h_] := μ[h] =
  (Moment[NormalDistribution[α1, β1], h] + Moment[NormalDistribution[α2, β2], h]) / 2
α = μ[1];
β = (μ[2] - μ[1]^2)^(1/2);
b[y_] := b[y] = PDF[NormalDistribution[α, β], y]
Plot[{f[y], b[y]}, {y, -20, 25}, PlotRange → All,
 PlotLegends → {"Exact pdf", "Base density"}]
```

f1[x] = $b[x]\,p_1[x]$, n=90

```
n = 90;
ClearAll[p1, f1]
M1 = Table[Moment[NormalDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := p1[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = b[y] p1[y]
Plot[{f[y], f1[y]}, {y, -10, 15}, PlotRange → All]
ISD = NIntegrate[(f[y] - f1[y])^2, {y, -∞, ∞}]
```

Chapter 3

3.1.1 A mixture of beta pdf's

```
α1 = 6; β1 = 2; α2 = 3; β2 = 9;
ClearAll[f]
f[y_] :=
 f[y] = (PDF[BetaDistribution[α1, β1], y] + PDF[BetaDistribution[α2, β2], y]) / 2
Plot[f[y], {y, 0, 1}, PlotRange → All, AxesOrigin → {0, 0},
 PlotLegends → {"Mixture distribution"}, PlotStyle → Gray]
SeedRandom[1]
𝒟 = ProbabilityDistribution[f[y], {y, 0, 1}];
size = 500;
data = Rationalize[RandomVariate[𝒟, size], 10^-10];
H1 = Histogram[data, {0, 1, 0.05}, "PDF",
   ChartElementFunction → "FadingRectangle", ChartStyle → Orange];
KD = SmoothKernelDistribution[data];
KDE = Plot[PDF[KD, x], {x, 0, 1}, PlotStyle -> Dashed, PlotLegends → "KDE"];
KCDF = Plot[CDF[KD, x], {x, 0, 1}, PlotStyle → Orange, PlotLegends → {"KCDF"}];
edis1 = EmpiricalDistribution[data];
EmCDF1 = Table[size * CDF[edis1, data[[j]]] / (size + 1), {j, 1, size}];
SSD = Sum[(EmCDF1[[j]] - CDF[KD, data[[j]]])^2, {j, 1, size}]
Lp =
  ListPlot[Table[{data[[j]], size CDF[edis1, data[[j]]] / (size + 1)}, {j, 1, size}]];
Show[H1, KDE]
Show[Lp, KCDF]
ClearAll[μ, b]
μ[h_] := Mean[data^h];
α = μ[1] (μ[1] - μ[2]) / (μ[2] - μ[1]^2) // N;
β = (1 - μ[1]) α / μ[1] // N;
b[y_] := PDF[BetaDistribution[α, β], y]
Plot[{PDF[KD, y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"KDE", "Base density"}, PlotRange → All]
```

f1[*x*] = *b*[*x*] p1[*x*]

```
n = 10;
ClearAll[p1, f10, f1, F1]
M1 =
  Rationalize[Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}], 10^-100];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := p1[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f10[y_] := f10[y] = If[b[y] p1[y] > 0, b[y] p1[y], 0]
k = NIntegrate[f10[y], {y, 0, 1}];
f1[y_] := f1[y] = f10[y] / k
Show[H1, Plot[{PDF[KD, y], f1[y]}, {y, 0, 1},
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f1"}], PlotLabel → {n}]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
EstimCDF1 = Parallelize[Table[F1[data[[i]]], {i, 1, size}]];
SSD = NumberForm[Parallelize[Sum[(EmCDF1[[j]] - EstimCDF1[[j]])^2, {j, 1, size}]], 6]
ECDF = Plot[F1[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {{n}, "F1"}, PlotLegends → {"F1"}]
```

f2[*x*] = p1[*x*], n=11

```
n = 11;
ClearAll[p2, f2, F2]
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Table[μ[i], {i, 0, n}];
coe2 = Inverse[M2].μ2;
p2[y_] := p2[y] = Sum[coe2[[i]] y^(i-1), {i, 1, n+1}];
k2 = NIntegrate[p2[x], {x, 0, 1}];
f2[y_] := f2[y] = p2[y] / k2;
a1 = FindRoot[f2[x] == 0, {x, 0.05}][[1, 2]];
b1 = FindRoot[f2[x] == 0, {x, 0.95}][[1, 2]];
k = NIntegrate[f2[x], {x, a1, b1}];
ClearAll[bf1, bf2]
bf1[x_] := bf1[x] = If[a1 < x < b1, f2[x], 0]
bf2[x_] := bf2[x] = bf1[x] / k
Show[H1, Plot[{PDF[𝒟, y], bf2[y]}, {y, 0, 1},
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f2"}], PlotLabel → {n}]
ClearAll[F2]
F2[y_] := F2[y] = NIntegrate[bf2[x], {x, 0, y}];
ECDF = Plot[F2[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F2"}]
EstimCDF2 = Table[F2[data[[i]]], {i, 1, size}];
SSD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF2[[j]])^2, {j, 1, size}], 6]
```

f3[x] = b[x] + p3[x], n=11

```
n = 11;
ClearAll[f31, p3, f3, F3];
M3 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ3 = Table[μ[h] - Moment[BetaDistribution[α, β], h], {h, 0, n}];
coe3 = Inverse[M3].μ3;
p3[x_] := p3[x] = Sum[coe3[[i]] x^(i-1), {i, 1, n+1}];
f31[y_] := f31[y] = b[y] + p3[y];

k3 = NIntegrate[f31[x], {x, 0, 1}];
f3[y_] := f3[y] = f31[y] / k3;
k = NIntegrate[f3[x], {x, 0.0085, 0.995}];
ClearAll[bf1, bf2]
bf1[x_] := bf1[x] = If[0.0085 < x < 0.996, f3[x], 0]
bf2[x_] := bf2[x] = bf1[x] / k
Show[H1, Plot[{PDF[𝒟, y], bf2[y]}, {y, 0, 1},
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f3"}], PlotLabel → {n}]
ClearAll[F3]
F3[y_] := F3[y] = NIntegrate[bf2[x], {x, 0, y}];
ECDF = Plot[F3[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F3"}]
EstimCDF2 = Table[F3[data[[i]]], {i, 1, size}];
SSD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF2[[j]])^2, {j, 1, size}], 6]
```

$f5[x] = k e^{\int_\alpha^x r[y]\, dy}$, $v = 8$

```
ν = 8;
ClearAll[r, IR, fν, f5, F5]
δ = 0;
M5 = Rationalize[Table[μ[i + j], {i, 0, ν}, {j, 0, ν}], 10^-10];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10]
fν[x_] := fν[x] = Exp[IR[x]]
k5 = NIntegrate[fν[x], {x, 0, 1}];
f5[x_] := f5[x] = fν[x] / k5
Show[H1, Plot[{PDF[KD, y], Evaluate[f5[y]]},
  {y, 0, 1}, PlotRange → All, PlotStyle → {Gray, Red},
  PlotLegends → {"KDE", "f5"}, AxesOrigin → {0, 0}], PlotLabel → {ν, δ}]
F5[y_] := NIntegrate[f5[x], {x, 0, y}]
ECDF = ListPlot[Table[{i, F5[i]}, {i, 0, 1, 0.001}], PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {ν, δ, "F5"}]
EstimCDF5 = Table[F5[data[[i]]], {i, 1, size}];
SSD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF5[[j]])^2, {j, 1, size}], 6]
```

$f7[x] = w\,b[x] + (1-w)\,e^{\int_{\alpha}^{x} r[y]\,dy}$, ν=6

```
ν = 6;
δ = 0;
w = 1/10;
ClearAll[μ70, p7, IP7, fν7, f7, F7];
μ70[h_] := 1/(1 - w) (μ[h] - w Moment[BetaDistribution[α, β], h]);
M7 = Rationalize[Table[μ70[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ7 = Prepend[Table[-h * μ70[h - 1], {h, 1, ν}], 0];
coe7 = Inverse[M7].μ7;
p7[x_] := Sum[coe7[[i]] x^(i-1), {i, 1, ν+1}];
IP7[x_] := Integrate[p7[y], {y, 0, x}, Assumptions → 0 < x < 1];
fν7[x_] := (1 - w) Exp[IP7[x]] + w b[x];
k7 = NIntegrate[fν7[x], {x, 0, 1}];
f7[x_] := fν7[x] / k7;
Show[H1, Plot[{PDF[KD, y], Evaluate[f7[y]]}, {y, 0, 1}, PlotRange → All,
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f7"}], PlotLabel → {ν, δ}]
F7[y_] := NIntegrate[f7[x], {x, 0, y}];
ECDF =
  ListPlot[Table[{i, F7[i]}, {i, 0, 1, 0.001}], PlotStyle → Orange, PlotRange → All];
Show[Lp, ECDF, PlotRange → Automatic, PlotLabel → {ν, δ, "F7"}]
EstimCDF7 = Table[F7[data[[i]]], {i, 1, size}];
SSD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF7[[j]])^2, {j, 1, size}], 6]
```

3.1.2 A mixture of gamma pdf's

```
α1 = 2; β1 = 2; α2 = 15; β2 = 1;
Plot[PDF[MixtureDistribution[{1/2, 1/2},
    {GammaDistribution[α1, β1], GammaDistribution[α2, β2]}], y],
  {y, 0, 30}, PlotStyle → Gray, PlotLegends → {"Mixture distribution"}]
SeedRandom[1234]
X = RandomVariate[MixtureDistribution[{1/2, 1/2},
     {GammaDistribution[α1, β1], GammaDistribution[α2, β2]}], 500];
size = Length[X];
sd = StandardDeviation[X];
a1 = 0;
a2 = Max[X] + sd;
{size, sd};
{Min[X], Max[X]};
{a1, a2};
𝒟0 = SmoothKernelDistribution[X];
H0 = Histogram[X, {a1, a2, 1.5}, "PDF",
    ChartElementFunction → "FadingRectangle", ChartStyle → Orange];
edis  =  EmpiricalDistribution[X];
Lp0 = ListPlot[
    Table[{X[[j]], n CDF[edis, X[[j]]] / (n + 1)}, {j, 1, size}], PlotStyle → Black];
X1 = (X - a1)/(a2 - a1);
H1 = Histogram[X1, {0, 1, 1.5/(a2 - a1)}, "PDF",
    ChartElementFunction → "FadingRectangle", ChartStyle → Orange];
𝒟 = SmoothKernelDistribution[X1];
KDE = Plot[PDF[𝒟, x], {x, 0, 1}, PlotStyle -> Dashed, PlotLegends → "KDE"];
KCDF = Plot[CDF[𝒟, x], {x, 0, 1}, PlotLegends → {"KCDF"}, PlotStyle → Orange];
edis1  =  EmpiricalDistribution[X1];
EmCDF1  =  Table[size * CDF[edis1, X1[[j]]] / (size + 1), {j, 1, size}];
SSD = Sum[(EmCDF1[[j]] - CDF[𝒟, X1[[j]]])^2, {j, 1, size}];
Lp = ListPlot[Table[{X1[[j]], size CDF[edis1, X1[[j]]] / (size + 1)}, {j, 1, size}]];
Show[H1, KDE]
Show[Lp, KCDF]

ClearAll[μ, b]
μ[h_] := μ[h] = Rationalize[Mean[X1^h], 10^-100];
μ[1]; μ[2];
α = μ[1] (μ[1] - μ[2])/(μ[2] - μ[1]^2);
β = (1 - μ[1]) α/μ[1];
b[y_] := b[y] = PDF[BetaDistribution[α, β], y]
Plot[{PDF[𝒟, y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
  PlotLegends → {"KDE", "Base density"}, PlotRange → All]
```

f1[*x*] = *b*[*x*] p1[*x*], n=12

```
n = 12;
ClearAll[p1, f1, F1];
M1 = Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := p1[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}];
f1[y_] := f1[y] = b[y] p1[y];
ClearAll[f11, f12]
f11[x_] := f11[x] = 1/2 (f1[x] + Abs[f1[x]])
k = NIntegrate[f11[x], {x, 0, 1}];
f12[x_] := f12[x] = f11[x] / k
Show[H1, Plot[{PDF[𝒟, y], f12[y]}, {y, 0, 1},
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f1"}], PlotLabel → {n}]
ClearAll[F1]
F1[y_] := F1[y] = NIntegrate[f12[x], {x, 0, y}];
ECDF = Plot[F1[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F1"}]
EstimCDF1 = Table[F1[X1[[i]]], {i, 1, size}];
ISD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF1[[j]])^2, {j, 1, size}], 6]
```

f2[*x*] = p2[*x*], n=11

```
n = 11;
ClearAll[p2, f2, F2];
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Table[μ[i], {i, 0, n}];
coe2 = Inverse[M2].μ2;
p2[y_] := p2[y] = Sum[coe2[[i]] y^(i-1), {i, 1, n+1}];
k2 = Rationalize[NIntegrate[p2[y], {y, 0, 1}], 10^-10];
f2[y_] := f2[y] = p2[y] / k2;
ClearAll[f11, f12]
b1 = FindRoot[f2[x] == 0, {x, 0.85}][[1, 2]];
f11[x_] := f11[x] = If[0 < x < b1, f2[x], 0]
k = NIntegrate[f11[x], {x, 0, 1}];
f12[x_] := f12[x] = f11[x] / k
Show[H1, Plot[{PDF[𝒟, y], f12[y]}, {y, 0, 1},
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f2"}], PlotLabel → {n}]
ClearAll[F2]
F2[y_] := F2[y] = NIntegrate[f12[x], {x, 0, y}];
ECDF = Plot[F2[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F2"}]
EstimCDF1 = Table[F2[X1[[i]]], {i, 1, size}];
ISD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF1[[j]])^2, {j, 1, size}], 6]
```

f3[x] = b[x] + p3[x], n=11

```
n = 11;
ClearAll[f31, p3, f3, F3];
M3 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ3 = Table[μ[h] - Moment[BetaDistribution[α, β], h], {h, 0, n}];
coe3 = Rationalize[Inverse[M3].μ3, 10^-100];
p3[x_] := p3[x] = Sum[coe3[[i]] x^(i-1), {i, 1, n+1}];
f31[y_] := f31[y] = b[y] + p3[y];
k3 = Rationalize[NIntegrate[f31[y], {y, 0, 1}], 10^-10];
f3[y_] := f3[y] = f31[y] / k3;
ClearAll[f11, f12]
a1 = FindRoot[f3[x] == 0, {x, 0.01}][[1, 2]];
b1 = FindRoot[f3[x] == 0, {x, 0.85}][[1, 2]];
{a1, b1};
f11[x_] := f11[x] = If[a1 < x < b1, f3[x], 0]
k = NIntegrate[f11[x], {x, a1, b1}];
f12[x_] := f12[x] = f11[x] / k
Show[H1, Plot[{PDF[𝒟, y], f12[y]}, {y, 0, 1},
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f3"}], PlotLabel → {n}]
ClearAll[F2]
F2[y_] := F2[y] = NIntegrate[f12[x], {x, 0, y}];
ECDF = Plot[F2[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F3"}]
EstimCDF1 = Table[F2[X1[[i]]], {i, 1, size}];
ISD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF1[[j]])^2, {j, 1, size}], 6]
```

$f5[x] = ke^{\int_\alpha^x r[y]\,dy}$, v=5

```
ν = 5;
ClearAll[r, IR, fν, f5, F5]
δ = 0;
M5 = Rationalize[Table[μ[i + j], {i, 0, ν}, {j, 0, ν}], 10^-10];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10]
fν[x_] := Exp[IR[x]]
k5 = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f5[x_] := f5[x] = fν[x] / k5
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
  PlotStyle → {Gray, Red}, PlotLegends → {"KPDF", "f5"}], PlotLabel → {ν, δ}]
F5[y_] := F5[y] = NIntegrate[f5[x], {x, 0, y}]
ECDF = Plot[F5[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {ν, δ, "F5"}]
EstimCDF5 = Table[F5[X1[[i]]], {i, 1, size}];
SSD = Sum[(EmCDF1[[j]] - EstimCDF5[[j]])^2, {j, 1, size}]
```

$f7[x] = w\, b[x] + (1 - w)\, e^{\int_{\alpha}^{x} r[y]\, dy}$, $\nu = 9$

```
ν = 9;
δ = 0;
w = 1 / 10;
ClearAll[μ70, r, IR, fν, f7];
μ70[h_] := μ70[h] = 1/(1 - w) (μ[h] - w * Moment[BetaDistribution[α, β], h]);
M7 = Rationalize[Table[μ70[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ7 = Prepend[Table[-h * μ70[h - 1], {h, 1, ν}], 0];
coe7 = Inverse[M7].μ7;
r[x_] := r[x] = Sum[coe7[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := w b[x] + (1 - w) Exp[IR[x]];
k = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f7[x_] := f7[x] = fν[x] / k;
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f7[y]]}, {y, 0, 1}, PlotRange → All,
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f7"}], PlotLabel → {ν, δ}]
ClearAll[F7]
F7[y_] := F7[y] = NIntegrate[f7[x], {x, 0, y}]
ECDF = Plot[F7[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {ν, δ, "F7"}]
EstimCDF7 = Table[F7[X1[[i]]], {i, 1, size}];
SSD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF7[[j]])^2, {j, 1, size}], 6]
```

3.2.1 The Buffalo snowfall data

```
X = {126.4, 82.4, 78.1, 51.1, 90.9, 76.2, 104.5, 87.4, 110.5, 25.0, 69.3, 53.5, 39.8,
   63.6, 46.7, 72.9, 79.6, 83.6, 80.7, 60.3, 79.0, 74.4, 49.6, 54.7, 71.8, 49.1,
   103.9, 51.6, 82.4, 83.6, 77.8, 79.3, 89.6, 85.5, 58.0, 120.7, 110.5, 65.4, 39.9,
   40.1, 88.7, 71.4, 83.0, 55.9, 89.9, 84.8, 105.2, 113.7, 124.7, 114.5, 115.6,
   102.4, 101.4, 89.8, 71.5, 70.9, 98.3, 55.5, 66.1, 78.4, 120.5, 97.0, 110.0};
size = Length[X];
sd = StandardDeviation[X];
a1 = Min[X] - sd;
a2 = Max[X] + sd;
{size, sd};
{Min[X], Max[X]};
{a1, a2};
𝒟0 = SmoothKernelDistribution[X];
H0 = Histogram[X, {a1, a2, 11}, "PDF",
  ChartElementFunction → "FadingRectangle", ChartStyle → Orange]
edis = EmpiricalDistribution[X];
Lp0 = ListPlot[Table[{X[[j]], size CDF[edis, X[[j]]] / (size + 1)}, {j, 1, size}],
   PlotStyle → Black];
X1 = Sort[(X - a1)/(a2 - a1)];
H1 = Histogram[X1, {0, 1, 11/(a2 - a1)}, "PDF",
   ChartElementFunction → "FadingRectangle", ChartStyle → Orange];
𝒟 = SmoothKernelDistribution[X1];

KDE = Plot[PDF[𝒟, x], {x, 0, 1}, PlotStyle -> Dashed, PlotLegends → "KDE"];
KCDF = Plot[CDF[𝒟, x], {x, 0, 1}, PlotStyle → Orange, PlotLegends → {"KCDF"}];
edis1 = EmpiricalDistribution[X1];
EmCDF1 = Table[size * CDF[edis1, X1[[j]]] / (size + 1), {j, 1, size}];
Sum[(EmCDF1[[j]] - CDF[𝒟, X1[[j]]])^2, {j, 1, size}];
Lp = ListPlot[Table[{X1[[j]], size CDF[edis1, X1[[j]]] / (size + 1)}, {j, 1, size}],
   PlotStyle → Black];
Show[H1, KDE]
Show[Lp, KCDF]

ClearAll[μ, b]
μ[h_] := μ[h] = Mean[X1^h];
α = μ[1] (μ[1] - μ[2])/(μ[2] - μ[1]^2);
β = (1 - μ[1]) α/μ[1];
b[y_] := b[y] = PDF[BetaDistribution[α, β], y]
Plot[{PDF[𝒟, y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"KDE", "Base density"}, PlotRange → All]
```

f1[*x*] = *b*[*x*] p1[*x*], n=10

```
n = 10;
ClearAll[p1, f10, F1]
M1 =
  Rationalize[Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}], 10^-100];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := p1[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f10[y_] := f10[y] = b[y] p1[y]
a11 = If[0 < FindRoot[f10[x] == 0, {x, Min[X1]}][[1, 2]] < 1,
   FindRoot[f10[x] == 0, {x, Min[X1]}][[1, 2]], FindRoot[f10[x] == 0, {x, 0.1}][[1, 2]]];
a22 = If[0 < FindRoot[f10[x] == 0, {x, Max[X1]}][[1, 2]] < 1,
   FindRoot[f10[x] == 0, {x, Max[X1]}][[1, 2]], FindRoot[f10[x] == 0, {x, 0.9}][[1, 2]]];
k = NIntegrate[If[a11 < x < a22, f10[x], 0], {x, 0, 1}];
f1[y_] := f1[y] = If[a11 < y < a22, f10[y], 0] / k
Show[H1, Plot[{PDF[𝒟, y], f1[y]}, {y, a11, a22},
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f1"}], PlotLabel -> n]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, a11, y}]
ECDF = Plot[F1[y], {y, a11, a22}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F1"}]
EstimCDF1 = Table[F1[X1[[i]]], {i, 1, size}];
SSD = Sum[(EmCDF1[[j]] - EstimCDF1[[j]])^2, {j, 1, size}]
```

f2[*x*] = p2[*x*], n=11

```
n = 11;
ClearAll[p2, f2, F2]
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Table[μ[i], {i, 0, n}];
coe2 = Inverse[M2].μ2;
p2[y_] := p2[y] = Sum[coe2[[i]] y^(i-1), {i, 1, n+1}]
a11 = If[0 < FindRoot[p2[x] == 0, {x, Min[X1]}][[1, 2]] < 1,
   FindRoot[p2[x] == 0, {x, Min[X1]}][[1, 2]], FindRoot[p2[x] == 0, {x, 0.1}][[1, 2]]];
a22 = If[0 < FindRoot[p2[x] == 0, {x, Max[X1]}][[1, 2]] < 1,
   FindRoot[p2[x] == 0, {x, Max[X1]}][[1, 2]], FindRoot[p2[x] == 0, {x, 0.9}][[1, 2]]];
{a11, a22};
k2 = NIntegrate[If[a11 < x < a22, p2[x], 0], {x, 0, 1}];
f2[y_] := f2[y] = p2[y] / k2
Show[H1, Plot[{PDF[𝒟, y], f2[y]}, {y, a11, a22},
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f2"}], PlotLabel → n]
F2[y_] := F2[y] = NIntegrate[f2[x], {x, a11, y}]
ECDF = Plot[F2[y], {y, a11, a22}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F2"}]
EstimCDF2 = Table[F2[X1[[i]]], {i, 1, size}];
Sum[(EmCDF1[[j]] - EstimCDF2[[j]])^2, {j, 1, size}]
```

f3[x] = b[x] + p3[x], n=11

```
n = 11;
ClearAll[f31, p3, f3, F3]
M3 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ3 = Table[μ[h] - Moment[BetaDistribution[α, β], h], {h, 0, n}];
coe3 = Inverse[M3].μ3;
p3[x_] := p3[x] = Sum[coe3[[i]] x^(i-1), {i, 1, n+1}]
f31[y_] := f31[y] = b[y] + p3[y]
a11 = If[0 < FindRoot[f31[x] == 0, {x, Min[X1]}][[1, 2]] < 1,
   FindRoot[f31[x] == 0, {x, Min[X1]}][[1, 2]], FindRoot[f31[x] == 0, {x, 0.1}][[1, 2]]];
a22 = If[0 < FindRoot[f31[x] == 0, {x, Max[X1]}][[1, 2]] < 1,
   FindRoot[f31[x] == 0, {x, Max[X1]}][[1, 2]], FindRoot[f31[x] == 0, {x, 0.9}][[1, 2]]];
{a11, a22};
k3 = NIntegrate[If[a11 < x < a22, f31[x], 0], {x, 0, 1}];
f3[y_] := f3[y] = If[a11 < y < a22, f31[y], 0] / k3
Show[H1, Plot[{PDF[𝒟, y], f3[y]}, {y, a11, a22},
  PlotLegends → {"KDE", "f3"}, PlotStyle → {Gray, Red}], PlotLabel → n]
F3[y_] := F3[y] = NIntegrate[f3[x], {x, a11, y}]
ECDF = Plot[F3[y], {y, a11, a22}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F3"}]
EstimCDF3 = Table[F3[X1[[i]]], {i, 1, size}];
SSD = Sum[(EmCDF1[[j]] - EstimCDF3[[j]])^2, {j, 1, size}]
```

$f5[x] = k e^{\int_\alpha^x r[y]\,dy}$, $v=6$

```
ν = 6;
ClearAll[r, IR, fν, f5, F5]
δ = 0;
M5 = Rationalize[Table[μ[i + j], {i, 0, ν}, {j, 0, ν}], 10^-10];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10]
fν[x_] := fν[x] = Exp[IR[x]]
k5 = NIntegrate[fν[x], {x, 0, 1}];
a11 = 0; a22 = 1;
f5[x_] := fν[x] / k5
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f5[y]]}, {y, a11, a22}, PlotRange → All,
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f5"}], PlotLabel → {ν, δ}]
F5[y_] := F5[y] = NIntegrate[f5[x], {x, a11, y}]
ECDF = Plot[F5[y], {y, a11, a22}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {ν, δ, "F5"}]
EstimCDF5 = Table[F5[X1[[i]]], {i, 1, size}];
SSD = Sum[(EmCDF1[[j]] - EstimCDF5[[j]])^2, {j, 1, size}]
```

$f7[x] = w\,b[x] + (1-w)\, e^{\int_\alpha^x r[y]\,dy}$, $\nu = 6$

```
ν = 6;
δ = 0;
w = 1/2;
ClearAll[μ70, p7, IP7, fν7, f7, F7]
μ70[h_] := μ70[h] = 1/(1 - w) (μ[h] - w Moment[BetaDistribution[α, β], h])
M7 = Rationalize[Table[μ70[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ7 = Prepend[Table[-h * μ70[h - 1], {h, 1, ν}], 0];
coe7 = Inverse[M7].μ7;
p7[x_] := p7[x] = Sum[coe7[[i]] x^(i-1), {i, 1, ν+1}];
IP7[x_] := IP7[x] = Integrate[p7[y], {y, 0, x}, Assumptions → 0 < x < 1]
fν7[x_] := fν7[x] = (1 - w) Exp[IP7[x]] + w b[x]
a11 = ArgMin[{fν7[x], 0 < x < 0.1}, x]; a22 = 1;
k7 = NIntegrate[If[a11 < x < a22, fν7[x], 0], {x, 0, 1}];
f7[x_] := f7[x] = fν7[x] / k7;
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f7[y]]}, {y, a11, a22}, PlotRange → All,
  PlotStyle → {Gray, Red}, PlotLegends → {"KDE", "f7"}], PlotLabel → {ν, δ}]
F7[y_] := F7[y] = NIntegrate[f7[x], {x, a11, y}]
ECDF = Plot[F7[y], {y, a11, a22}, PlotStyle → Orange, PlotRange → All];
Show[Lp, ECDF, PlotRange → Automatic, PlotLabel → {ν, δ, "F7"}, AxesOrigin → {0, 0}]
EstimCDF7 = Table[F7[X1[[i]]], {i, 1, size}];
SSD = Sum[(EmCDF1[[j]] - EstimCDF7[[j]])^2, {j, 1, size}]
```

3.2.2 The Old Faithful geyser data

```
X = {1.6`, 1.667`, 1.7`, 1.733`, 1.75`, 1.75`, 1.75`, 1.75`, 1.75`, 1.75`, 1.783`, 1.783`,
   1.8`, 1.8`, 1.8`, 1.8`, 1.817`, 1.817`, 1.817`, 1.833`, 1.833`, 1.833`, 1.833`,
   1.833`, 1.833`, 1.833`, 1.85`, 1.85`, 1.867`, 1.867`, 1.867`, 1.867`, 1.867`,
   1.867`, 1.867`, 1.867`, 1.883`, 1.883`, 1.883`, 1.883`, 1.917`, 1.917`, 1.933`,
   1.933`, 1.95`, 1.967`, 1.967`, 1.967`, 1.983`, 1.983`, 1.983`, 2.`, 2.`, 2.`,
   2.`, 2.017`, 2.017`, 2.017`, 2.033`, 2.033`, 2.067`, 2.083`, 2.083`, 2.1`, 2.1`,
   2.1`, 2.133`, 2.15`, 2.167`, 2.167`, 2.183`, 2.2`, 2.2`, 2.2`, 2.217`, 2.233`,
   2.233`, 2.25`, 2.25`, 2.267`, 2.283`, 2.3`, 2.317`, 2.333`, 2.35`, 2.367`, 2.383`,
   2.4`, 2.4`, 2.417`, 2.417`, 2.483`, 2.617`, 2.633`, 2.8`, 2.883`, 2.9`, 3.067`,
   3.317`, 3.333`, 3.333`, 3.367`, 3.417`, 3.45`, 3.5`, 3.5`, 3.567`, 3.567`, 3.6`,
   3.6`, 3.6`, 3.6`, 3.683`, 3.717`, 3.733`, 3.75`, 3.767`, 3.767`, 3.817`, 3.833`,
   3.833`, 3.833`, 3.833`, 3.833`, 3.85`, 3.85`, 3.883`, 3.917`, 3.917`, 3.917`,
   3.95`, 3.95`, 3.966`, 3.967`, 4.`, 4.`, 4.`, 4.`, 4.`, 4.`, 4.033`, 4.033`, 4.05`,
   4.067`, 4.067`, 4.083`, 4.083`, 4.083`, 4.083`, 4.083`, 4.1`, 4.1`, 4.117`,
   4.117`, 4.133`, 4.133`, 4.15`, 4.15`, 4.15`, 4.15`, 4.167`, 4.167`, 4.167`,
   4.167`, 4.183`, 4.2`, 4.233`, 4.233`, 4.233`, 4.25`, 4.25`, 4.25`, 4.25`, 4.267`,
   4.267`, 4.283`, 4.283`, 4.3`, 4.3`, 4.317`, 4.333`, 4.333`, 4.333`, 4.333`,
   4.333`, 4.35`, 4.35`, 4.35`, 4.35`, 4.366`, 4.367`, 4.367`, 4.367`, 4.383`, 4.4`,
   4.417`, 4.417`, 4.417`, 4.417`, 4.433`, 4.433`, 4.45`, 4.45`, 4.45`, 4.467`,
   4.467`, 4.483`, 4.5`, 4.5`, 4.5`, 4.5`, 4.5`, 4.5`, 4.5`, 4.5`, 4.517`, 4.533`,
   4.533`, 4.533`, 4.533`, 4.533`, 4.55`, 4.567`, 4.567`, 4.567`, 4.583`, 4.583`,
   4.583`, 4.583`, 4.6`, 4.6`, 4.6`, 4.6`, 4.617`, 4.633`, 4.633`, 4.633`, 4.65`,
   4.667`, 4.667`, 4.7`, 4.7`, 4.7`, 4.7`, 4.7`, 4.7`, 4.716`, 4.733`, 4.75`, 4.767`,
   4.783`, 4.8`, 4.8`, 4.8`, 4.8`, 4.8`, 4.8`, 4.817`, 4.817`, 4.833`, 4.833`,
   4.85`, 4.883`, 4.9`, 4.9`, 4.933`, 4.933`, 4.933`, 5.`, 5.033`, 5.067`, 5.1`};
```

```
size = Length[X];
sd = StandardDeviation[X];
a1 = Min[X] - sd;
a2 = Max[X] + sd;
{size, sd};
{Min[X], Max[X]};
{a1, a2};
𝒟0 = SmoothKernelDistribution[X];
H0 = Histogram[X, {a1, a2, 0.3}, "PDF",
  ChartElementFunction → "FadingRectangle", ChartStyle → Orange]
edis = EmpiricalDistribution[X];
Lp0 = ListPlot[
   Table[{X[[j]], n CDF[edis, X[[j]]] / (n + 1)}, {j, 1, size}], PlotStyle → Black];
X1 = (X - a1)/(a2 - a1);
H1 = Histogram[X1, {0, 1, 0.3/(a2 - a1)}, "PDF",
   ChartElementFunction → "FadingRectangle", ChartStyle → Orange];
𝒟 = SmoothKernelDistribution[X1];
KDE = Plot[PDF[𝒟, x], {x, 0, 1}, PlotStyle -> Dashed, PlotLegends → "KDE"];
KCDF = Plot[CDF[𝒟, x], {x, 0, 1}, PlotStyle → Orange, PlotLegends → {"KCDF"}];
edis1 = EmpiricalDistribution[X1];
EmCDF1 = Table[size * CDF[edis1, X1[[j]]] / (size + 1), {j, 1, size}];
SSD = Sum[(EmCDF1[[j]] - CDF[𝒟, X1[[j]]])^2, {j, 1, size}];
Lp = ListPlot[Table[{X1[[j]], size CDF[edis1, X1[[j]]] / (size + 1)}, {j, 1, size}],
   PlotStyle → Black];
Show[H1, KDE]
Show[Lp, KCDF]
ClearAll[μ, b]
μ[h_] := μ[h] = Rationalize[Mean[X1^h], 10^-100];
α = μ[1] (μ[1] - μ[2])/(μ[2] - μ[1]^2);
β = (1 - μ[1]) α/μ[1];
b[y_] := b[y] = PDF[BetaDistribution[α, β], y]
Plot[{PDF[𝒟, y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"KDE", "Base density"}, PlotRange → All]
```

f1[*x*] = *b*[*x*] p1[*x*], n=12

```
n = 12;
ClearAll[p1, f1];
M1 = Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := p1[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}];
f1[y_] := f1[y] = b[y] p1[y];
ClearAll[f11, f12]
f11[x_] := f11[x] = If[0.15 < x < 0.85, If[f1[x] > 0, f1[x], 0], 0]
k = NIntegrate[f11[x], {x, 0, 1}];
f12[x_] := f12[x] = f11[x] / k
ν = 5;
ClearAll[μ10, r, IR, fν, f5]
μ10[h_] := μ10[h] = NIntegrate[x^h f12[x], {x, 0, 1}]
δ = 0;
M5 = Table[μ10[i + j], {i, 0, ν}, {j, 0, ν}];
μ5 = Prepend[Table[-h * μ10[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Integrate[r[y], {y, 0, x}]
fν[x_] := fν[x] = Exp[IR[x]]
k5 = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f5[x_] := f5[x] = fν[x] / k5
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
  PlotStyle → {Gray, {Dashed, Red}}, PlotLegends → {"KDE", "f1"}], PlotLabel → {n}]
ClearAll[F5]
F5[y_] := F5[y] = NIntegrate[f5[x], {x, 0, y}]
ECDF = Plot[F5[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F1"}]
EstimCDF5 = Table[F5[X1[[i]]], {i, 1, size}];
SSD = Sum[(EmCDF1[[j]] - EstimCDF5[[j]])^2, {j, 1, size}]
```

f2[x] = p2[x], n=9

```
n = 9;
ClearAll[p2, f2, F2];
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Table[μ[i], {i, 0, n}];
coe2 = Inverse[M2].μ2;
p2[y_] := p2[y] = Sum[coe2[[i]] y^(i-1), {i, 1, n+1}]
k2 = Rationalize[NIntegrate[p2[y], {y, 0, 1}], 10^-10];
f2[y_] := f2[y] = p2[y] / k2
ClearAll[f11, f12]
f11[x_] := f11[x] = If[0.15 < x < 0.85, If[f2[x] > 0, f2[x], 0], 0]
k = NIntegrate[f11[x], {x, 0, 1}];
f12[x_] := f12[x] = f11[x] / k
ν = 4;
ClearAll[μ10, r, IR, fν, f5, F5]
μ10[h_] := μ10[h] = NIntegrate[x^h f12[x], {x, 0, 1}]
δ = 0;
M5 = Rationalize[Table[μ10[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ5 = Prepend[Table[-h * μ10[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Integrate[r[y], {y, 0, x}]
fν[x_] := fν[x] = Exp[IR[x]]
k5 = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f5[x_] := f5[x] = fν[x] / k5
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
  PlotStyle → {Gray, {Dashed, Red}}, PlotLegends → {"KDE", "f2"}], PlotLabel → {n}]
ClearAll[F5]
F5[y_] := F5[y] = NIntegrate[f5[x], {x, 0, y}]
ECDF = Plot[F5[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F2"}]
EstimCDF5 = Table[F5[X1[[i]]], {i, 1, size}];
SSD = NumberForm[Sum[(EmCDF1[[j]] - EstimCDF5[[j]])^2, {j, 1, size}], 7]
```

f3[x] = b[x] + p3[x], n=19

```
n = 19;
ClearAll[f31, p3, f3, F3];
M3 = Table[1 / (i + j + 1), {i, 0, n}, {j, 0, n}];
μ3 = Table[μ[h] - Moment[BetaDistribution[α, β], h], {h, 0, n}];
coe3 = Inverse[M3].μ3;
p3[x_] := p3[x] = Sum[coe3[[i]] x^(i-1), {i, 1, n+1}];
f31[y_] := f31[y] = b[y] + p3[y];
k3 = Rationalize[NIntegrate[f31[y], {y, 0, 1}], 10^-10];
f3[y_] := f3[y] = f31[y] / k3;
ClearAll[f11, f12, F3]
f11[x_] := f11[x] = If[0.15 < x < 0.825, If[f3[x] > 0, f3[x], 0], 0]
k = NIntegrate[f11[x], {x, 0, 1}];
f12[x_] := f12[x] = f11[x] / k
Show[H1, Plot[{PDF[𝒟, y], f12[y]}, {y, 0, 1},
  PlotStyle → {Gray, {Red, Dashed}}, PlotLegends → {"KDE", "f3"}], PlotLabel → {n}]
F3[y_] := F3[y] = NIntegrate[f12[x], {x, 0, y}]
ECDF = Plot[F3[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F3"}]
EstimCDF3 = Table[F3[X1[[i]]], {i, 1, size}];
SSD = Sum[(EmCDF1[[j]] - EstimCDF3[[j]])^2, {j, 1, size}]
```

$f5[x] = k e^{\int_\alpha^x r[y]\,dy}$, v=7

```
ν = 7;
ClearAll[r, IR, fν, f5, F5];
δ = 0;
M5 = Table[μ[i + j], {i, 0, ν}, {j, 0, ν}];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := fν[x] = Exp[IR[x]];
k5 = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f5[x_] := f5[x] = fν[x] / k5;
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
  PlotStyle → {Gray, {Red, Dashed}}, PlotLegends → {"KDE", "f5"}], PlotLabel → {ν, δ}]
F5[y_] := F5[y] = NIntegrate[f5[x], {x, 0, y}]
ECDF = Plot[F5[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {ν, δ, "F5"}]
EstimCDF5 = Table[F5[X1[[i]]], {i, 1, size}];
SSD = Sum[(EmCDF1[[j]] - EstimCDF5[[j]])^2, {j, 1, size}]
```

3.2.3 A large data set: US household income in 2016

```
X = Flatten[Import[
    "F:\\Google Drive\\Thesis\\Thesis writing\\notebooks final version\\chap-density
      estimation\\Big data\\kaggle income big data.xlsx", {"Data", 1}]];
size = Length[X];
sd = StandardDeviation[X];
a1 = 0;
a2 = Max[X] + sd;
{size, sd};
{Min[X], Max[X]};
{a1, a2};
𝒟0 = SmoothKernelDistribution[X];
H0 = Histogram[X, {a1, a2, 10000}, "PDF"]
edis = EmpiricalDistribution[X];
Lp0 = ListPlot[Table[{X[[j]], size CDF[edis, X[[j]]] / (size + 1)}, {j, 1, size}],
   PlotStyle → Black];
X1 = (X - a1)/(a2 - a1);
H1 = Histogram[X1, {0, 1, 10000/(a2 - a1)}, "PDF"];
𝒟 = SmoothKernelDistribution[X1];
KDE = Plot[PDF[𝒟, x], {x, 0, 1}, PlotStyle -> Dashed, PlotLegends → "KDE"];
KCDF = Plot[CDF[𝒟, x], {x, 0, 1}, PlotStyle → Orange, PlotLegends → {"KCDF"}];
edis1 = EmpiricalDistribution[X1];
EmCDF1 = Table[size * CDF[edis1, X1[[j]]] / (size + 1), {j, 1, size}];
SSD = Sum[(EmCDF1[[j]] - CDF[𝒟, X1[[j]]])^2, {j, 1, size}];
Lp = ListPlot[Table[{X1[[j]], size CDF[edis1, X1[[j]]] / (size + 1)}, {j, 1, size}],
   PlotStyle → Black];
Show[H1, KDE]
Show[Lp, KCDF]

ClearAll[μ, b]
μ[h_] := μ[h] = Mean[X1^h];
α = μ[1] (μ[1] - μ[2])/(μ[2] - μ[1]^2);
β = (1 - μ[1]) α/μ[1];
b[y_] := PDF[BetaDistribution[α, β], y]
Plot[{PDF[𝒟, y], b[y]}, {y, 0, 1}, PlotStyle → {Gray, Dashed},
 PlotLegends → {"KDE", "Base density"}, PlotRange → All]
```

f1[*x*] = *b*[*x*] p1[*x*], n=19

```
n = 19;
ClearAll[p1, f10, f1];
M1 =
  Rationalize[Table[Moment[BetaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}], 10^-10];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
p1[y_] := p1[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}];
f10[y_] := f10[y] = b[y] p1[y];

k = NIntegrate[f10[x], {x, 0, 1}];
f1[y_] := f1[y] = f10[y] / k;
ClearAll[f11, f12]
a11 = FindRoot[f1[y] == 0, {y, 0.05}][[1, 2]];
b11 = FindRoot[f1[y] == 0, {y, 0.7}][[1, 2]];
{a11, b11};
f11[y_] := f11[y] = If[a11 < y < b11, f1[y], 0]
k = NIntegrate[f11[y], {y, 0, 1}];
f12[y_] := f12[y] = f11[y] / k
Show[H1, Plot[{PDF[𝒟, y], f12[y]}, {y, 0, 1},
  PlotStyle → {Gray, {Red, Dashed}}, PlotLegends → {"KDE", "f1"}], PlotLabel → {n}]
ClearAll[F1]
F1[y_] := NIntegrate[f12[x], {x, 0, y}];
ECDF = Plot[F1[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F1"}]
```

f2[*x*] = p2[*x*], n=20

```
n = 20;
ClearAll[p2, f2];
M2 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ2 = Rationalize[Table[μ[i], {i, 0, n}], 10^-100];
coe2 = Inverse[M2].μ2;
p2[y_] := p2[y] = Sum[coe2[[i]] y^(i-1), {i, 1, n+1}];
k2 = NIntegrate[p2[x], {x, 0, 1}];
f2[y_] := f2[y] = p2[y] / k2;
ClearAll[f11, f12]
a11 = FindRoot[f2[y] == 0, {y, 0.05}][[1, 2]];
b11 = FindRoot[f2[y] == 0, {y, 0.7}][[1, 2]];
{a11, b11};
f11[y_] := f11[y] = If[a11 < y < b11, f2[y], 0]
k = NIntegrate[f11[y], {y, 0, 1}];
f12[y_] := f12[y] = f11[y] / k
Show[H1, Plot[{PDF[𝒟, y], f12[y]}, {y, 0, 1},
  PlotStyle → {Gray, {Dashed, Red}}, PlotLegends → {"KDE", "f2"}], PlotLabel → {n}]
ClearAll[F2]
F2[y_] := F2[y] = NIntegrate[f12[x], {x, 0, y}];
ECDF = Plot[F2[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F2"}]
```

f3[*x*] = *b*[*x*] + p3[*x*], n=12

```
n = 12;
ClearAll[f30, p3, f3];
M3 = Table[1/(i + j + 1), {i, 0, n}, {j, 0, n}];
μ3 = Table[μ[h] - Moment[BetaDistribution[α, β], h], {h, 0, n}];
coe3 = Inverse[M3].μ3;
p3[x_] := p3[x] = Sum[coe3[[i]] x^(i-1), {i, 1, n+1}];
f30[y_] := f30[y] = b[y] + p3[y];
k3 = NIntegrate[f30[x], {x, 0, 1}];
f3[y_] := f3[y] = f30[y] / k3;

ClearAll[f11, f12]
a11 = FindRoot[f3[y] == 0, {y, 0.05}][[1, 2]];
b11 = FindRoot[f3[y] == 0, {y, 0.7}][[1, 2]];
{a11, b11};
f11[y_] := f11[y] = If[a11 < y < b11, f3[y], 0]
k = NIntegrate[f11[y], {y, 0, 1}];
f12[y_] := f12[y] = f11[y] / k
Show[H1, Plot[{PDF[𝒟, y], f12[y]}, {y, 0, 1},
  PlotStyle → {Gray, {Dashed, Red}}, PlotLegends → {"KDE", "f3"}], PlotLabel → {n}]
ClearAll[F3]
F3[y_] := F3[y] = NIntegrate[f12[x], {x, 0, y}]
ECDF = Plot[F3[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {n, "F3"}]
```

f5[x] = $ke^{\int_\alpha^x r[y]\,dy}$, $v=8$

```
ν = 5;
ClearAll[r, IR, fν, f5, F5]
δ = 0;
M5 = Rationalize[Table[μ[i + j], {i, 0, ν}, {j, 0, ν}], 10^-100];
μ5 = Prepend[Table[-h * μ[h - 1], {h, 1, ν}], 0];
coe5 = Inverse[M5].μ5;
r[x_] := r[x] = Sum[coe5[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10]
fν[x_] := fν[x] = Exp[IR[x]]
k5 = NIntegrate[fν[x], {x, 0, 1}];
f5[x_] := f5[x] = fν[x] / k5
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f5[y]]}, {y, 0, 1}, PlotRange → All,
  PlotStyle → {Gray, {Dashed, Red}}, PlotLegends → {"KDE", "f5"}], PlotLabel → {ν, δ}]
ClearAll[F5]
F5[y_] := F5[y] = NIntegrate[f5[x], {x, 0, y}]
ECDF = Plot[F5[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {ν, δ, "F5"}]
```

f7[x] = w b[x] + (1 − w) $e^{\int_\alpha^x r[y]\,dy}$, $v=5$

```
ν = 5;
δ = 0;
w = 1 / 10;
ClearAll[μ70, r, IR, fν, f7];
μ70[h_] :=
  μ70[h] = Rationalize[1/(1 - w) (μ[h] - w * Moment[BetaDistribution[α, β], h]), 10^-100];
M7 = Table[μ70[i + j], {i, 0, ν}, {j, 0, ν}];
μ7 = Prepend[Table[-h * μ70[h - 1], {h, 1, ν}], 0];
coe7 = Inverse[M7].μ7;
r[x_] := r[x] = Sum[coe7[[i]] x^(i-1), {i, 1, ν+1}];
IR[x_] := IR[x] = Rationalize[Integrate[r[y], {y, 0, x}], 10^-10];
fν[x_] := w b[x] + (1 - w) Exp[IR[x]];
k = Rationalize[NIntegrate[fν[x], {x, 0, 1}], 10^-10];
f7[x_] := f7[x] = fν[x] / k;
Show[H1, Plot[{PDF[𝒟, y], Evaluate[f7[y]]}, {y, 0, 1}, PlotRange → All,
  PlotStyle → {Gray, {Dashed, Red}}, PlotLegends → {"KDE", "f7"}], PlotLabel → {ν, δ}]
ClearAll[F7]
F7[y_] := F7[y] = NIntegrate[f7[x], {x, 0, y}]
ECDF = Plot[F7[y], {y, 0, 1}, PlotStyle → Orange];
Show[Lp, ECDF, PlotLabel → {ν, δ, "F7"}]
```

Chapter 4

4.2.1 Sample generated from a beta(2,5) distribution

```
α = 2; β = 5;
ClearAll[f, F]
f[x_] := f[x] = PDF[BetaDistribution[α, β], x];
F[x_] := F[x] = CDF[BetaDistribution[α, β], x];
Plot[f[x], {x, 0, 1}, PlotStyle → Gray]
ClearAll[fo, p1, data1, data2, p2, data3, p3, data4, data5]
fo[n_, k_, x_] := fo[n, k, x] = n! / ((k - 1)! (n - k)!) F[x]^(k-1) (1 - F[x])^(n-k) f[x]
𝒟[n_, k_] = ProbabilityDistribution[fo[n, k, x], {x, 0, 1}];
p1[n_, k_] := p1[n, k] = Integrate[x * fo[n, k, x], {x, 0, 1}]
data1[n_] := data1[n] = Table[p1[n, i], {i, 1, n}]
p2[n_, k_] := p2[n, k] = Quiet[ArgMax[{fo[n, k, x], 0 < x < 1}, x]]
data2[n_] := data2[n] = Table[p2[n, i], {i, 1, n}]
p3[n_, k_] := p3[n, k] = Median[𝒟[n, k]]
data3[n_] := data3[n] = Table[p3[n, i], {i, 1, n}]
data4[n_] := data4[n] = Table[Quantile[BetaDistribution[α, β], (2 i - 1)/(2 n)], {i, 1, n}]
SeedRandom[1]
data5[n_] := data5[n] = Sort[RandomVariate[BetaDistribution[α, β], n]]

moment0[h_] := moment0[h] = Moment[BetaDistribution[α, β], h]
moment1[h_, n_] := moment1[h, n] = Moment[data1[n], h]
moment2[h_, n_] := moment2[h, n] = Moment[data2[n], h]
moment3[h_, n_] := moment3[h, n] = Moment[data3[n], h]
moment4[h_, n_] := moment4[h, n] = Moment[data4[n], h]
moment5[h_, n_] := moment4[h, n] = Moment[data5[n], h]
n = 5;
ListLinePlot[{data1[n], data2[n], data3[n], data4[n], data5[n]},
 PlotLegends → {1, 2, 3, 4, 5}, PlotLabel → n]

m[0] = Table[moment0[h], {h, 1, n}] // N;
m[1] = Table[moment1[h, n], {h, 1, n}] // N;
m[2] = Table[moment2[h, n], {h, 1, n}] // N;
m[3] = Table[moment3[h, n], {h, 1, n}] // N;
m[4] = Table[moment4[h, n], {h, 1, n}] // N;
m[5] = Table[moment5[h, n], {h, 1, n}] // N;
com = {m[0], m[1], m[2], m[3], m[4], m[5]};
com1 = com;
Table[com1[[i, j]] = NumberForm[(100 * ((com[[i, j]] - com[[1, j]]) / com[[1, j]])) "%",
     {5, 2}], {i, 2, 6}, {j, 1, n}] // N;
Grid[
 Transpose[Prepend[Transpose[Prepend[com1, Range[n]]],
   {"Moments", "exact", "method 1", "method 2", "method 3", "method 4", "method 5"}]],
 Frame → All,
 Background → {None, {{White, Lighter[Blend[{Blue, Green}], 0.8`]}}},
 Dividers → All,
 Spacings → 1.5` {1, 1}
]
```

Criterion 1: SSD (moments)

```
Table[Sum[(com[[i, j]] - com[[1, j]])^2, {j, 1, n}], {i, 2, 6}] // N // TableForm
```

Criterion 2: ISD (density)

```
KD1 = SmoothKernelDistribution[data1[n]];
KD2 = SmoothKernelDistribution[data2[n]];
KD3 = SmoothKernelDistribution[data3[n]];
KD4 = SmoothKernelDistribution[data4[n]];
KD5 = SmoothKernelDistribution[data5[n]];
{ISD1 = NIntegrate[(PDF[KD1, y] - f[y])^2, {y, 0, 1}],
  ISD2 = NIntegrate[(PDF[KD2, y] - f[y])^2, {y, 0, 1}],
  ISD3 = NIntegrate[(PDF[KD3, y] - f[y])^2, {y, 0, 1}],
  ISD4 = NIntegrate[(PDF[KD4, y] - f[y])^2, {y, 0, 1}],
  ISD5 = NIntegrate[(PDF[KD5, y] - f[y])^2, {y, 0, 1}]} // TableForm
GraphicsGrid[
 {{Plot[{f[x], PDF[KD1, x], PDF[KD2, x]}, {x, -0.3, 1}, PlotLegends → {"Exact", 1, 2}],
   Plot[{f[x], PDF[KD3, x], PDF[KD4, x], PDF[KD5, x]},
    {x, -0.3, 1}, PlotLegends → {"Exact", 3, 4, 5}]}}]
```

Criterion 3: ISD (CDF)

```
ED1 = EmpiricalDistribution[data1[n]];
ED2 = EmpiricalDistribution[data2[n]];
ED3 = EmpiricalDistribution[data3[n]];
ED4 = EmpiricalDistribution[data4[n]];
ED5 = EmpiricalDistribution[data5[n]];
{ISE1 = NIntegrate[(CDF[ED1, y] - F[y])^2, {y, 0, 1}],
  ISE2 = NIntegrate[(CDF[ED2, y] - F[y])^2, {y, 0, 1}],
  ISE3 = NumberForm[
    NIntegrate[(CDF[ED3, y] - F[y])^2, {y, 0, 1}, WorkingPrecision → 300], 6],
  ISE4 = NumberForm[NIntegrate[(CDF[ED4, y] - F[y])^2,
     {y, 0, 1}, WorkingPrecision → 100], 6],
  ISE5 = NIntegrate[(CDF[ED5, y] - F[y])^2, {y, 0, 1}]} // TableForm
C1 = Plot[{F[x], CDF[ED1, x]}, {x, 0, 1},
   PlotRange → All, PlotLegends → {"Exact", "ECDF1"}];
C2 = Plot[{F[x], CDF[ED2, x]}, {x, 0, 1}, PlotRange → All,
   PlotLegends → {"Exact", "ECDF2"}];
C3 = Plot[{F[x], CDF[ED3, x]}, {x, 0, 1}, PlotRange → All,
   PlotLegends → {"Exact", "ECDF3"}];
C4 = Plot[{F[x], CDF[ED4, x]}, {x, 0, 1}, PlotRange → All,
   PlotLegends → {"Exact", "ECDF4"}];
C5 = Plot[{F[x], CDF[ED5, x]}, {x, 0, 1}, PlotRange → All,
  PlotLegends → {"Exact", "ECDF5"}];
GraphicsGrid[{{C1, C2, C3}, {C4, C5}}]
```

Criterion 4: SSD (corrected CDF)

```
Lp1 = ListPlot[Table[{data1[n][[j]], n * CDF[ED1, data1[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
Lp2 = ListPlot[Table[{data2[n][[j]], n * CDF[ED2, data2[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
Lp3 = ListPlot[Table[{data3[n][[j]], n * CDF[ED3, data3[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
Lp4 = ListPlot[Table[{data4[n][[j]], n * CDF[ED4, data4[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
Lp5 = ListPlot[Table[{data5[n][[j]], n * CDF[ED5, data5[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
{SSD1 = NumberForm[Sum[(n * CDF[ED1, y] / (n + 1) - F[y])^2, {y, data1[n]}] // N, {6, 6}],
  SSD2 =
   NumberForm[Sum[(n * CDF[ED2, y] / (n + 1) - F[y])^2, {y, data2[n]}] // N, {6, 6}],
  SSD3 = NumberForm[Sum[(n * CDF[ED3, y] / (n + 1) - F[y])^2, {y, data3[n]}] // N,
    {6, 6}],
  SSD4 = NumberForm[Sum[(n * CDF[ED4, y] / (n + 1) - F[y])^2, {y, data4[n]}] // N,
    {6, 6}],
  SSD5 = NumberForm[Sum[(n * CDF[ED5, y] / (n + 1) - F[y])^2, {y, data5[n]}] // N,
    {6, 6}]} // TableForm
S1 = Show[Plot[F[x], {x, 0, 1}], Lp1, PlotLabel → 1];
S2 = Show[Plot[F[x], {x, 0, 1}], Lp2, PlotLabel → 2];
S3 = Show[Plot[F[x], {x, 0, 1}], Lp3, PlotLabel → 3];
S4 = Show[Plot[F[x], {x, 0, 1}], Lp4, PlotLabel → 4];
S5 = Show[Plot[F[x], {x, 0, 1}], Lp5, PlotLabel → 5];
GraphicsGrid[{{S1, S2, S3}, {S4, S5}}]
```

4.2.2 Samples generated from a mixture of beta pdf's

```
α1 = 8; β1 = 12;
α2 = 3; β2 = 15;
ClearAll[f, F]
f[x_] :=
  f[x] = 1/2 (PDF[BetaDistribution[α1, β1], x] + PDF[BetaDistribution[α2, β2], x]);
F[x_] := F[x] = 1/2 (CDF[BetaDistribution[α1, β1], x] +
      CDF[BetaDistribution[α2, β2], x]);
Plot[f[x], {x, 0, 1}, PlotStyle → Gray]
ClearAll[fo, p1, data1, data2, p2, data3, p3, data4, data5]
fo[n_, k_, x_] := fo[n, k, x] = n! / ((k - 1)! (n - k)!) F[x]^(k-1) (1 - F[x])^(n-k) f[x]
𝒟[n_, k_] = 𝒟[n, k] = ProbabilityDistribution[fo[n, k, x], {x, 0, 1}];
p1[n_, k_] := p1[n, k] = Integrate[x * fo[n, k, x], {x, 0, 1}]
data1[n_] := data1[n] = Table[p1[n, i], {i, 1, n}]
p2[n_, k_] := p2[n, k] = Quiet[ArgMax[{fo[n, k, x], 0 < x < 1}, x]]
data2[n_] := data2[n] = Table[p2[n, i], {i, 1, n}]
p3[n_, k_] := p3[n, k] = Median[𝒟[n, k]]
data3[n_] := data3[n] = Table[p3[n, i], {i, 1, n}]
data4[n_] := data4[n] = Table[Quantile[MixtureDistribution[{1/2, 1/2},
      {BetaDistribution[α1, β1], BetaDistribution[3, 15]}], (2 i - 1)/(2 n)], {i, 1, n}]
SeedRandom[1]
data5[n_] := data5[n] = Sort[RandomVariate[MixtureDistribution[
      {1/2, 1/2}, {BetaDistribution[8, 12], BetaDistribution[3, 15]}], 5]]

ClearAll[moment0, moment1, moment2, moment3, moment4, moment5]
moment0[h_] := moment0[h] = Moment[MixtureDistribution[
     {1/2, 1/2}, {BetaDistribution[α1, β1], BetaDistribution[α2, β2]}], h]
moment1[h_, n_] := moment1[h, n] = Moment[data1[n], h]
moment2[h_, n_] := moment2[h, n] = Moment[data2[n], h]
moment3[h_, n_] := moment3[h, n] = Moment[data3[n], h]
moment4[h_, n_] := moment4[h, n] = Moment[data4[n], h]
moment5[h_, n_] := moment5[h, n] = Moment[data5[n], h]
n = 5;
ListLinePlot[{data1[n], data2[n], data3[n], data4[n], data5[n]},
 PlotLegends → {1, 2, 3, 4, 5}, PlotLabel → n]
```

```
m[0] = Parallelize[Table[moment0[h], {h, 1, n}]] // N;
m[1] = Parallelize[Table[moment1[h, n], {h, 1, n}]] // N;
m[2] = Parallelize[Table[moment2[h, n], {h, 1, n}]] // N;
m[3] = Parallelize[Table[moment3[h, n], {h, 1, n}]] // N;
m[4] = Parallelize[Table[moment4[h, n], {h, 1, n}]] // N;
m[5] = Parallelize[Table[moment5[h, n], {h, 1, n}]] // N;
com = {m[0], m[1], m[2], m[3], m[4], m[5]};
com1 = com;
Table[com1[[i, j]] = NumberForm[(100 * ((com[[i, j]] - com[[1, j]]) / com[[1, j]])) "%",
     {5, 2}], {i, 2, 6}, {j, 1, n}] // N;
Grid[
 Transpose[Prepend[Transpose[Prepend[com1, Range[n]]],
   {"Moments", "exact", "method 1", "method 2", "method 3", "method 4", "method 5"}]],
 Frame → All,
 Background → {None, {{White, Lighter[Blend[{Blue, Green}], 0.8`]}}},
 Dividers → All,
 Spacings → 1.5` {1, 1}
]
```

Criterion 1: SSD (moments)

```
Table[Sum[(com[[i, j]] - com[[1, j]])^2, {j, 1, n}], {i, 2, 6}] // N // TableForm
```

Criterion 2: ISD (density)

```
KD1 = SmoothKernelDistribution[data1[n]];
KD2 = SmoothKernelDistribution[data2[n]];
KD3 = SmoothKernelDistribution[data3[n]];
KD4 = SmoothKernelDistribution[data4[n]];
KD5 = SmoothKernelDistribution[data5[n]];
{ISD1 = NIntegrate[(PDF[KD1, y] - f[y])^2, {y, 0, 1}],
  ISD2 = NIntegrate[(PDF[KD2, y] - f[y])^2, {y, 0, 1}],
  ISD3 = NIntegrate[(PDF[KD3, y] - f[y])^2, {y, 0, 1}],
  ISD4 = NIntegrate[(PDF[KD4, y] - f[y])^2, {y, 0, 1}],
  ISD5 = NIntegrate[(PDF[KD5, y] - f[y])^2, {y, 0, 1}]} // TableForm
GraphicsGrid[
 {{Plot[{f[x], PDF[KD1, x], PDF[KD2, x]}, {x, -0.3, 1}, PlotLegends → {"Exact", 1, 2}],
   Plot[{f[x], PDF[KD3, x], PDF[KD4, x], PDF[KD5, x]},
    {x, -0.3, 1}, PlotLegends → {"Exact", 3, 4, 5}]}}]
```

Criterion 3: ISD (CDF)

```
ED1 = EmpiricalDistribution[data1[n]];
ED2 = EmpiricalDistribution[data2[n]];
ED3 = EmpiricalDistribution[data3[n]];
ED4 = EmpiricalDistribution[data4[n]];
ED5 = EmpiricalDistribution[data5[n]];
{ISE1 = NIntegrate[(CDF[ED1, y] - F[y])^2, {y, 0, 1}],
  ISE2 = NIntegrate[(CDF[ED2, y] - F[y])^2, {y, 0, 1}],
  ISE3 = NumberForm[
    NIntegrate[(CDF[ED3, y] - F[y])^2, {y, 0, 1}, WorkingPrecision → 300], 6],
  ISE4 = NumberForm[NIntegrate[(CDF[ED4, y] - F[y])^2,
     {y, 0, 1}, WorkingPrecision → 100], 6],
  ISE5 = NIntegrate[(CDF[ED5, y] - F[y])^2, {y, 0, 1}]} // TableForm
C1 = Plot[{F[x], CDF[ED1, x]}, {x, 0, 1},
   PlotRange → All, PlotLegends → {"Exact", "ECDF1"}];
C2 = Plot[{F[x], CDF[ED2, x]}, {x, 0, 1}, PlotRange → All,
   PlotLegends → {"Exact", "ECDF2"}];
C3 = Plot[{F[x], CDF[ED3, x]}, {x, 0, 1}, PlotRange → All,
   PlotLegends → {"Exact", "ECDF3"}];
C4 = Plot[{F[x], CDF[ED4, x]}, {x, 0, 1}, PlotRange → All,
   PlotLegends → {"Exact", "ECDF4"}];
C5 = Plot[{F[x], CDF[ED5, x]}, {x, 0, 1}, PlotRange → All,
   PlotLegends → {"Exact", "ECDF5"}];
GraphicsGrid[{{C1, C2, C3}, {C4, C5}}]
```

Criterion 4: SSD (corrected CDF)

```
Lp1 = ListPlot[Table[{data1[n][[j]], n * CDF[ED1, data1[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
Lp2 = ListPlot[Table[{data2[n][[j]], n * CDF[ED2, data2[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
Lp3 = ListPlot[Table[{data3[n][[j]], n * CDF[ED3, data3[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
Lp4 = ListPlot[Table[{data4[n][[j]], n * CDF[ED4, data4[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
Lp5 = ListPlot[Table[{data5[n][[j]], n * CDF[ED5, data5[n][[j]]] / (n + 1)}, {j, 1, n}],
   PlotStyle → Black];
{SSD1 = NumberForm[Sum[(n * CDF[ED1, y] / (n + 1) - F[y])^2, {y, data1[n]}] // N, {6, 6}],
  SSD2 =
   NumberForm[Sum[(n * CDF[ED2, y] / (n + 1) - F[y])^2, {y, data2[n]}] // N, {6, 6}],
  SSD3 = NumberForm[Sum[(n * CDF[ED3, y] / (n + 1) - F[y])^2, {y, data3[n]}] // N,
    {6, 6}],
  SSD4 = NumberForm[Sum[(n * CDF[ED4, y] / (n + 1) - F[y])^2, {y, data4[n]}] // N,
    {6, 6}],
  SSD5 = NumberForm[Sum[(n * CDF[ED5, y] / (n + 1) - F[y])^2, {y, data5[n]}] // N,
    {6, 6}]} // TableForm
S1 = Show[Plot[F[x], {x, 0, 1}], Lp1, PlotLabel → 1];
S2 = Show[Plot[F[x], {x, 0, 1}], Lp2, PlotLabel → 2];
S3 = Show[Plot[F[x], {x, 0, 1}], Lp3, PlotLabel → 3];
S4 = Show[Plot[F[x], {x, 0, 1}], Lp4, PlotLabel → 4];
S5 = Show[Plot[F[x], {x, 0, 1}], Lp5, PlotLabel → 5];
GraphicsGrid[{{S1, S2, S3}, {S4, S5}}]
```

Chapter 5

section 5.3

```
ClearAll[tail]
tail[𝒟_] := (
  μ = Mean[𝒟];
  σ = StandardDeviation[𝒟];
  Qt9 = (2/π) ArcTan[(Quantile[𝒟, .9] - μ) / σ];
  Qt999999 = (2/π) ArcTan[(Quantile[𝒟, .999999] - μ) / σ];
  Print["p=", Qt999999 - Qt9];
  Print[Plot[ PDF[𝒟, σ Tan[(π / 2) y] + μ] σ (π / 2) Sec[(π / 2) y]^2,
    {y, -1, 1}, PlotRange → All, PlotLabel → {𝒟}]]
 )
```

```
tail[UniformDistribution[{0, 1}]]
tail[BetaDistribution[5, 2]]
tail[NormalDistribution[0, 1]]
tail[WeibullDistribution[2, 1]]
tail[WeibullDistribution[1, 1]]
tail[WeibullDistribution[0.5, 1]]
tail[ExtremeValueDistribution[0, 1]]
tail[LogisticDistribution[0, 1]]
tail[ExponentialDistribution[1]]
tail[LogNormalDistribution[0, 1]]
```

```
tail[RayleighDistribution[1]]
tail[GammaDistribution[1, 1]]
tail[GammaDistribution[2, 1]]
tail[GammaDistribution[5, 1]]
tail[GammaDistribution[20, 1]]
tail[GammaDistribution[50, 1]]
tail[StudentTDistribution[3]]
tail[StudentTDistribution[5]]
tail[StudentTDistribution[20]]
tail[BetaPrimeDistribution[5, 3]]
tail[BetaPrimeDistribution[2, 5]]
tail[BetaPrimeDistribution[50, 30]]
```

Chapter 6

6.2.1 Quadratic forms in gamma random variables

A={{3,1},{1,2}}

```
A = {{3, 1}, {1, 2}};
α1 = 2; β1 = 2; α2 = 9; β2 = 1;
Plot[{PDF[GammaDistribution[α1, β1], y], PDF[GammaDistribution[α2, β2], y]},
 {y, 0, 25}, PlotRange → All, PlotLegends → {"X1", "X2"}]
```

```
𝒟1 = GammaDistribution[α1, β1];
𝒟2 = GammaDistribution[α2, β2];
size = 10 000;
SeedRandom[1]
data1 = RandomVariate[𝒟1, size];
SeedRandom[2]
data2 = RandomVariate[𝒟2, size];
data =
  Sort[Table[{data1[[i]], data2[[i]]}.A.{data1[[i]], data2[[i]]}, {i, 1, size}]];
ED = EmpiricalDistribution[data];
ClearAll[F]
F[x_] := F[x] = CDF[ED, x];
ub = 2000;
F[ub];
H1 = Histogram[data, {0, ub, ub / 30}, "PDF"];
ClearAll[μ];
μ[h_] :=
 μ[h] = Expand[({X1, X2}.A.{X1, X2})^h] /. {X1^j_. :→ Moment[GammaDistribution[α1, β1], j],
    X2^j_. :→ Moment[GammaDistribution[α2, β2], j]}
{α = μ[1]^2 / (μ[2] - μ[1]^2), β = (μ[2] - μ[1]^2) / μ[1]};
ClearAll[fb, Fb];
fb[y_] := fb[y] = PDF[GammaDistribution[α, β], y]
Fb[y_] := Fb[y] = CDF[GammaDistribution[α, β], y]
pf = Plot[fb[y], {y, 0, ub}];
Show[H1, pf]
Plot[{F[y], Fb[y]}, {y, 0, ub}, PlotRange → All, PlotLegends → {"Empirical", "Base"}]

n = 5;
M1 = Table[Moment[GammaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
ClearAll[p, f1, F1]
p[y_] := p[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = fb[y] p[y]
pf1 = Plot[f1[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf1, PlotLabel → n]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
Plot[{F[y], F1[y]}, {y, 0, 1000}, PlotRange → All,
 PlotLegends → {"Empirical", "Approximated"}, PlotLabel → n]
```

A = {{1, 0, 0, 0}, {0, 2, 1, 0}, {0, 1, 2, 0}, {0, 0, 0, 3}}

```
A = {{1, 0, 0, 0}, {0, 2, 1, 0}, {0, 1, 2, 0}, {0, 0, 0, 3}};
% // TableForm;
Eigenvalues[A] // N;
α1 = 2; β1 = 2; α2 = 9; β2 = 1;
α3 = 2; β3 = 1; α4 = 12; β4 = 1;
Plot[{PDF[GammaDistribution[α1, β1], y], PDF[GammaDistribution[α2, β2], y],
  PDF[GammaDistribution[α3, β3], y], PDF[GammaDistribution[α4, β4], y]},
 {y, 0, 25}, PlotRange → All, PlotLegends → {"X1", "X2", "X3", "X4"}]
```

```
𝒟1 = GammaDistribution[α1, β1];
𝒟2 = GammaDistribution[α2, β2];
𝒟3 = GammaDistribution[α3, β3];
𝒟4 = GammaDistribution[α4, β4];
size = 10000;
SeedRandom[1]
data1 = RandomVariate[𝒟1, size];
SeedRandom[2]
data2 = RandomVariate[𝒟2, size];
SeedRandom[3]
data3 = RandomVariate[𝒟3, size];
SeedRandom[4]
data4 = RandomVariate[𝒟4, size];
data = Sort[Table[{data1[[i]], data2[[i]], data3[[i]], data4[[i]]}.
      A.{data1[[i]], data2[[i]], data3[[i]], data4[[i]]}, {i, 1, size}]];
ED = EmpiricalDistribution[data];
ClearAll[F]
F[x_] := F[x] = CDF[ED, x];
ub = 3000;
F[ub];
H1 = Histogram[data, {0, ub, ub/30}, "PDF"];

ClearAll[μ];
μ[h_] := μ[h] = Expand[({X1, X2, X3, X4}.A.{X1, X2, X3, X4})^h] /.
    {X1^j_. :→ Moment[GammaDistribution[α1, β1], j],
     X2^j_. :→ Moment[GammaDistribution[α2, β2], j],
     X3^j_. :→ Moment[GammaDistribution[α3, β3], j],
     X4^j_. :→ Moment[GammaDistribution[α4, β4], j]}
{α = μ[1]^2/(μ[2] - μ[1]^2), β = (μ[2] - μ[1]^2)/μ[1]};
ClearAll[fb, Fb];
fb[y_] := fb[y] = PDF[GammaDistribution[α, β], y]
Fb[y_] := Fb[y] = CDF[GammaDistribution[α, β], y]
pf = Plot[fb[y], {y, 0, ub}];
Show[H1, pf]
Plot[{F[y], Fb[y]}, {y, 0, ub}, PlotRange → All, PlotLegends → {"Empirical", "Base"}]

n = 8;
M1 = Table[Moment[GammaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
ClearAll[p, f1, F1]
p[y_] := p[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = fb[y] p[y]
pf1 = Plot[f1[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf1, PlotLabel → n]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
Plot[{F[y], F1[y]}, {y, 0, ub}, PlotRange → All,
 PlotLegends → {"Empirical", "Approximated"}, PlotLabel → n]
```

6.2.2 Quadratic forms in inverse Gaussian random variables

A={{3,1},{1,2}}

```
A = {{3, 1}, {1, 2}};
α1 = 2; β1 = 5; α2 = 3; β2 = 6;
Plot[{PDF[InverseGaussianDistribution[α1, β1], y],
   PDF[InverseGaussianDistribution[α2, β2], y]},
  {y, 0, 10}, PlotRange → All, PlotLegends → {"X1", "X2"}]

𝒟1 = InverseGaussianDistribution[α1, β1];
𝒟2 = InverseGaussianDistribution[α2, β2];
size = 10 000;
SeedRandom[1]
data1 = RandomVariate[𝒟1, size];
SeedRandom[2]
data2 = RandomVariate[𝒟2, size];
data =
   Sort[Table[{data1[[i]], data2[[i]]}.A.{data1[[i]], data2[[i]]}, {i, 1, size}]];

ED = EmpiricalDistribution[data];
ClearAll[F]
F[x_] := F[x] = CDF[ED, x];
ub = 700;
F[ub];
H1 = Histogram[data, {0, ub, ub / 30}, "PDF"];

ClearAll[μ];
μ[h_] := μ[h] = Expand[({X1, X2}.A.{X1, X2})^h] /.
    {X1^j_. :→ Moment[InverseGaussianDistribution[α1, β1], j],
     X2^j_. :→ Moment[InverseGaussianDistribution[α2, β2], j]}
{α = μ[1], β = μ[1]^3 / (μ[2] - μ[1]^2)};
ClearAll[fb, Fb];
fb[y_] := fb[y] = PDF[InverseGaussianDistribution[α, β], y]
Fb[y_] := Fb[y] = CDF[InverseGaussianDistribution[α, β], y]
pf = Plot[fb[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf]
Plot[{F[y], Fb[y]}, {y, 0, ub}, PlotRange → All, PlotLegends → {"Empirical", "Base"}]
```

```
n = 5;
M1 = Table[Moment[InverseGaussianDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
ClearAll[p, f1]
p[y_] := p[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = fb[y] p[y]

pf1 = Plot[f1[y], {y, 0, ub}, PlotRange → All, PlotLabel → n];
Show[H1, pf1, PlotLabel → n]
ClearAll[F1]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
Plot[{F[y], F1[y]}, {y, 0, ub}, PlotRange → All,
 PlotLegends → {"Empirical", "Approximated"}, PlotLabel → n]
```

A = {{1, 0, 0, 0}, {0, 2, 1, 0}, {0, 1, 2, 0}, {0, 0, 0, 3}}

```
A = {{1, 0, 0, 0}, {0, 2, 1, 0}, {0, 1, 2, 0}, {0, 0, 0, 3}};
% // TableForm;
Eigenvalues[A] // N;
α1 = 2; β1 = 5; α2 = 3; β2 = 6;
α3 = 2; β3 = 2; α4 = 3; β4 = 4;
Plot[{PDF[InverseGaussianDistribution[α1, β1], y],
   PDF[InverseGaussianDistribution[α2, β2], y],
   PDF[InverseGaussianDistribution[α3, β3], y],
   PDF[InverseGaussianDistribution[α4, β4], y]}, {y, 0, 10},
 PlotRange → All, PlotLegends → {"X1", "X2", "X3", "X4"}]

𝒟1 = InverseGaussianDistribution[α1, β1];
𝒟2 = InverseGaussianDistribution[α2, β2];
𝒟3 = InverseGaussianDistribution[α3, β3];
𝒟4 = InverseGaussianDistribution[α4, β4];
size = 10000;
SeedRandom[1]
data1 = RandomVariate[𝒟1, size];
SeedRandom[2]
data2 = RandomVariate[𝒟2, size];
SeedRandom[3]
data3 = RandomVariate[𝒟3, size];
SeedRandom[4]
data4 = RandomVariate[𝒟4, size];
data = Sort[Table[{data1[[i]], data2[[i]], data3[[i]], data4[[i]]}.
      A.{data1[[i]], data2[[i]], data3[[i]], data4[[i]]}, {i, 1, size}]];
ED = EmpiricalDistribution[data];
ClearAll[F]
F[x_] := F[x] = CDF[ED, x];
ub = 900;
F[ub];
H1 = Histogram[data, {0, ub, ub / 30}, "PDF"];
```

```
ClearAll[μ];
μ[h_] := μ[h] = Expand[({X1 , X2, X3, X4}.A. {X1 , X2, X3, X4})^h] /.
    {X1^j_. :> Moment[InverseGaussianDistribution[α1, β1], j],
     X2^j_. :> Moment[InverseGaussianDistribution[α2, β2], j],
     X3^j_. :> Moment[InverseGaussianDistribution[α3, β3], j],
     X4^j_. :> Moment[InverseGaussianDistribution[α4, β4], j]}
{α = μ[1], β = μ[1]^3 / (μ[2] - μ[1]^2)};
ClearAll[fb, Fb];
fb[y_] := fb[y] = PDF[InverseGaussianDistribution[α, β], y]
Fb[y_] := Fb[y] = CDF[InverseGaussianDistribution[α, β], y]
pf = Plot[fb[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf]
Plot[{F[y], Fb[y]}, {y, 0, ub}, PlotRange → All, PlotLegends → {"Empirical", "Base"}]

n = 5;
M1 = Table[Moment[InverseGaussianDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
ClearAll[p, f1, F1]
p[y_] := p[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = fb[y] p[y]
pf1 = Plot[f1[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf1, PlotLabel → n]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
Plot[{F[y], F1[y]}, {y, 0, ub}, PlotRange → All,
 PlotLegends → {"Empirical", "Approximated"}, PlotLabel → n]
```

6.2.3 Quadratic forms in binomial random variables

A = {{3, 1}, {1, 2}}

```
A = {{3, 1}, {1, 2}};
α1 = 20; β1 = 1/4; α2 = 30; β2 = 1/2;
Plot[{CDF[BinomialDistribution[α1, β1], y], CDF[BinomialDistribution[α2, β2], y]},
 {y, 0, 30}, PlotRange → All, PlotLegends → {"X1", "X2"}]
```

```
𝒟1 = BinomialDistribution[α1, β1];
𝒟2 = BinomialDistribution[α2, β2];
size = 10000;
SeedRandom[1]
data1 = RandomVariate[𝒟1, size];
SeedRandom[2]
data2 = RandomVariate[𝒟2, size];
data =
  Sort[Table[{data1[[i]], data2[[i]]}.A.{data1[[i]], data2[[i]]}, {i, 1, size}]];
ED = EmpiricalDistribution[data];
ClearAll[F]
F[x_] := F[x] = CDF[ED, x];
ub = 2000;
F[ub] // N;
H1 = Histogram[data, {0, ub, ub/20}, "PDF"];
ClearAll[μ];
μ[h_] := μ[h] =
  Expand[({X1, X2}.A.{X1, X2})^h] /. {X1^j_. :> Moment[BinomialDistribution[α1, β1], j],
    X2^j_. :> Moment[BinomialDistribution[α2, β2], j]}
{α = μ[1]^2/(μ[2] - μ[1]^2), β = (μ[2] - μ[1]^2)/μ[1]} // N;
ClearAll[fb, Fb];
fb[y_] := fb[y] = PDF[GammaDistribution[α, β], y]
Fb[y_] := Fb[y] = CDF[GammaDistribution[α, β], y]
pf = Plot[fb[y], {y, 0, ub}];
Show[H1, pf]
Plot[{F[y], Fb[y]}, {y, 0, ub}, PlotRange → All, PlotLegends → {"Empirical", "Base"}]

n = 8;
M1 = Table[Moment[GammaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
ClearAll[p, f1, F1]
p[y_] := p[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = N[fb[y] p[y]]
pf1 = Plot[f1[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf1, PlotLabel → n]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
Plot[{F[y], F1[y]}, {y, 0, ub}, PlotRange → All,
 PlotLegends → {"Empirical", "Approximated"}, PlotLabel → n]
```

A = {{1, 0, 0, 0}, {0, 2, -1, 0}, {0, -1, 2, 0}, {0, 0, 0, 3}}

```
A = {{1, 0, 0, 0}, {0, 2, -1, 0}, {0, -1, 2, 0}, {0, 0, 0, 3}};
% // TableForm;
Eigenvalues[A] // N;
α1 = 20; β1 = 1/4; α2 = 30; β2 = 1/2;
α3 = 20; β3 = 1/2; α4 = 30; β4 = 1/3;
Plot[{CDF[BinomialDistribution[α1, β1], y], CDF[BinomialDistribution[α2, β2], y],
  CDF[BinomialDistribution[α3, β3], y], CDF[BinomialDistribution[α4, β4], y]},
 {y, 0, 30}, PlotRange → All, PlotLegends → {"X1", "X2", "X3", "X4"}]
```

```
𝒟1 = BinomialDistribution[α1, β1];
𝒟2 = BinomialDistribution[α2, β2];
𝒟3 = BinomialDistribution[α3, β3];
𝒟4 = BinomialDistribution[α4, β4];
size = 10 000;
SeedRandom[1]
data1 = RandomVariate[𝒟1, size];
SeedRandom[2]
data2 = RandomVariate[𝒟2, size];
SeedRandom[3]
data3 = RandomVariate[𝒟3, size];
SeedRandom[4]
data4 = RandomVariate[𝒟4, size];
data = Sort[Table[{data1[[i]], data2[[i]], data3[[i]], data4[[i]]}.
      A.{data1[[i]], data2[[i]], data3[[i]], data4[[i]]}, {i, 1, size}]];

ED = EmpiricalDistribution[data];
ClearAll[F]
F[x_] := F[x] = CDF[ED, x];
ub = 1600;
F[ub] // N;
H1 = Histogram[data, {0, ub, ub / 20}, "PDF"];
ClearAll[μ];
μ[h_] := μ[h] = Expand[({X1 , X2, X3, X4}.A. {X1 , X2, X3, X4})^h] /.
    {X1^j_. :→ Moment[BinomialDistribution[α1, β1], j],
     X2^j_. :→ Moment[BinomialDistribution[α2, β2], j],
     X3^j_. :→ Moment[BinomialDistribution[α3, β3], j],
     X4^j_. :→ Moment[BinomialDistribution[α4, β4], j]}
{α = μ[1]^2 / (μ[2] - μ[1]^2), β = (μ[2] - μ[1]^2) / μ[1]};
ClearAll[fb, Fb];
fb[y_] := fb[y] = PDF[GammaDistribution[α, β], y]
Fb[y_] := Fb[y] = CDF[GammaDistribution[α, β], y]
pf = Plot[fb[y], {y, 0, ub}];
Show[H1, pf]
Plot[{F[y], Fb[y]}, {y, 0, ub}, PlotRange → All, PlotLegends → {"Empirical", "Base"}]

n = 9;
M1 = Table[Moment[GammaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
ClearAll[p, f1, F1]
p[y_] := p[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = N[fb[y] p[y]]
pf1 = Plot[f1[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf1, PlotLabel → n]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
Plot[{F[y], F1[y]}, {y, 0, ub}, PlotRange → All,
 PlotLegends → {"Empirical", "Approximated"}, PlotLabel → n]
```

6.2.4 Quadratic forms in Poisson random variables

A={{3,1},{1,2}}

```
A = {{3, 1}, {1, 2}};
α1 = 3; α2 = 5;
Plot[{CDF[PoissonDistribution[α1], y], CDF[PoissonDistribution[α2], y]},
 {y, 0, 20}, PlotRange → All, PlotLegends → {"X1", "X2"}]

𝒟1 = PoissonDistribution[α1];
𝒟2 = PoissonDistribution[α2];
size = 10 000;
SeedRandom[1]
data1 = RandomVariate[𝒟1, size];
SeedRandom[2]
data2 = RandomVariate[𝒟2, size];
data =
  Sort[Table[{data1[[i]], data2[[i]]}.A.{data1[[i]], data2[[i]]}, {i, 1, size}]];
ED = EmpiricalDistribution[data];
ClearAll[F]
F[x_] := F[x] = CDF[ED, x];
ub = Reduce[F[x] == 1, x][[2]];
H1 = Histogram[data, {0, ub, ub / 20}, "PDF"];
ClearAll[μ];
μ[h_] :=
 μ[h] = Expand[({X1, X2}.A.{X1, X2})^h] /. {X1^j_. :→ Moment[PoissonDistribution[α1], j],
    X2^j_. :→ Moment[PoissonDistribution[α2], j]}
{α = μ[1]^2 / (μ[2] - μ[1]^2), β = (μ[2] - μ[1]^2) / μ[1]} // N;
ClearAll[fb, Fb];
fb[y_] := fb[y] = PDF[GammaDistribution[α, β], y]
Fb[y_] := Fb[y] = CDF[GammaDistribution[α, β], y]
pf = Plot[fb[y], {y, 0, ub}];
Show[H1, pf]
Plot[{F[y], Fb[y]}, {y, 0, ub}, PlotRange → All, PlotLegends → {"Empirical", "Base"}]

n = 8;
M1 = Table[Moment[GammaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
ClearAll[p, f1, F1]
p[y_] := p[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = N[fb[y] p[y]]
pf1 = Plot[f1[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf1, PlotLabel → n]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
Plot[{F[y], F1[y]}, {y, 0, ub}, PlotRange → All,
 PlotLegends → {"Empirical", "Approximated"}, PlotLabel → n]
```

A = {{1, 0, 0, 0}, {0, 2, -1, 0}, {0, -1, 2, 0}, {0, 0, 0, 3}}

```
A = {{1, 0, 0, 0}, {0, 2, -1, 0}, {0, -1, 2, 0}, {0, 0, 0, 3}};
% // TableForm;
Eigenvalues[A] // N;
α1 = 3; α2 = 4;
α3 = 5; α4 = 6;
Plot[{CDF[PoissonDistribution[α1], y], CDF[PoissonDistribution[α2], y],
  CDF[PoissonDistribution[α3], y], CDF[PoissonDistribution[α4], y]},
 {y, 0, 20}, PlotRange → All, PlotLegends → {"X1", "X2", "X3", "X4"}]

𝒟1 = PoissonDistribution[α1];
𝒟2 = PoissonDistribution[α2];
𝒟3 = PoissonDistribution[α3];
𝒟4 = PoissonDistribution[α4];
size = 10000;
SeedRandom[1]
data1 = RandomVariate[𝒟1, size];
SeedRandom[2]
data2 = RandomVariate[𝒟2, size];
SeedRandom[3]
data3 = RandomVariate[𝒟3, size];
SeedRandom[4]
data4 = RandomVariate[𝒟4, size];
data = Sort[Table[{data1[[i]], data2[[i]], data3[[i]], data4[[i]]}.
      A.{data1[[i]], data2[[i]], data3[[i]], data4[[i]]}, {i, 1, size}]];
ED = EmpiricalDistribution[data];
ClearAll[F]
F[x_] := F[x] = CDF[ED, x];
ub = 1000;
F[ub];
H1 = Histogram[data, {0, ub, ub/20}, "PDF"];
ClearAll[μ];
μ[h_] := μ[h] = Expand[({X1, X2, X3, X4}.A.{X1, X2, X3, X4})^h] /.
    {X1^j_. :→ Moment[PoissonDistribution[α1], j],
     X2^j_. :→ Moment[PoissonDistribution[α2], j],
     X3^j_. :→ Moment[PoissonDistribution[α3], j],
     X4^j_. :→ Moment[PoissonDistribution[α4], j]}
{α = μ[1]^2/(μ[2] - μ[1]^2), β = (μ[2] - μ[1]^2)/μ[1]};
ClearAll[fb, Fb];
fb[y_] := fb[y] = PDF[GammaDistribution[α, β], y]
Fb[y_] := Fb[y] = CDF[GammaDistribution[α, β], y]
pf = Plot[fb[y], {y, 0, ub}];
Show[H1, pf]
Plot[{F[y], Fb[y]}, {y, 0, ub}, PlotRange → All, PlotLegends → {"Empirical", "Base"}]
```

```
n = 10;
M1 = Table[Moment[GammaDistribution[α, β], i + j], {i, 0, n}, {j, 0, n}];
μ1 = Table[μ[i], {i, 0, n}];
coe1 = Inverse[M1].μ1;
ClearAll[p, f1, F1]
p[y_] := p[y] = Sum[coe1[[i]] y^(i-1), {i, 1, n+1}]
f1[y_] := f1[y] = N[fb[y] p[y]]
pf1 = Plot[f1[y], {y, 0, ub}, PlotRange → All];
Show[H1, pf1, PlotLabel → n]
F1[y_] := F1[y] = NIntegrate[f1[x], {x, 0, y}]
Plot[{F[y], F1[y]}, {y, 0, ub}, PlotRange → All,
 PlotLegends → {"Empirical", "Approximated"}, PlotLabel → n]
```

where the sum in the definition of p is printed as

$$p[y\_] := p[y] = \sum_{i=1}^{n+1} \text{coe1}[[i]]\, y^{i-1}$$

Curriculum Vitae

Name:	Yishan Zang
Post-Secondary Education and Degrees:	University of Western Ontario London, ON 2017 - Present M.Sc. candidate
	Shandong University Jinan, China 2013 - 2017 B.Sc.
Related Work Experience:	Teaching Assistant The University of Western Ontario 2017 - 2018