2008-11-20

Parameter Estimation for the Beta Distribution

Claire Elayne Bangerter Owen
Brigham Young University - Provo

PARAMETER ESTIMATION FOR THE BETA DISTRIBUTION

by

Claire B. Owen

A project submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Statistics

Brigham Young University

December 2008

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a project submitted by

Claire B. Owen

This project has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Natalie J. Blades, Chair
Date	David G. Whiting
Date	Scott D. Grimshaw

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the project of Claire B. Owen in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Natalie J. Blades
Chair, Graduate Committee

Accepted for the Department

Scott D. Grimshaw
Graduate Coordinator

Accepted for the College

Thomas W. Sederberg
Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

PARAMETER ESTIMATION FOR THE BETA DISTRIBUTION

Claire B. Owen

Department of Statistics

Master of Science

The beta distribution is useful in modeling continuous random variables that lie between 0 and 1, such as proportions and percentages. The beta distribution takes on many different shapes and may be described by two shape parameters, α and β, that can be difficult to estimate. Maximum likelihood and method of moments estimation are possible, though method of moments is much more straightforward. We examine both of these methods here, and compare them to three more proposed methods of parameter estimation: 1) a method used in the Program Evaluation and Review Technique (PERT), 2) a modification of the two-sided power distribution (TSP), and 3) a quantile estimator based on the first and third quartiles of the beta distribution. We find the quantile estimator performs as well as maximum likelihood and method of moments estimators for most beta distributions. The PERT and TSP estimators do well for a smaller subset of beta distributions, though they never outperform the maximum likelihood, method of moments, or quantile estimators. We apply these estimation techniques to two data sets to see how well they approximate real data from Major League Baseball (batting averages) and the U.S. Department of Energy (radiation exposure). We find the maximum likelihood, method of moments, and

quantile estimators perform well with batting averages (sample size 160), and the method of moments and quantile estimators perform well with radiation exposure proportions (sample size 20). Maximum likelihood estimators would likely do fine with such a small sample size were it not for the iterative method needed to solve for α and β, which is quite sensitive to starting values. The PERT and TSP estimators do more poorly in both situations. We conclude that in addition to maximum likelihood and method of moments estimation, our method of quantile estimation is efficient and accurate in estimating parameters of the beta distribution.

ACKNOWLEDGEMENTS

I would like to thank the professors of the BYU Statistics Department and my husband, daughter, and parents for their support throughout this journey.

CONTENTS

CHAPTER

1 The Beta Distribution 1
1.1 Introduction 1
1.2 Literature Review 1

2 Parameter Estimation 8
2.1 Maximum Likelihood Estimators 8
2.2 Method of Moments Estimators 10
2.3 PERT Estimators 12
2.4 Two-Sided Power Distribution Estimators 14
2.5 Quantile Estimators 16

3 Simulation 18
3.1 Simulation Procedure 18
3.2 Simulation Results 19
3.3 Asymptotic Properties of Estimators 32

4 Applications 39
4.1 Batting Averages 39
4.2 Radiation Exposure 43

APPENDIX

A Simulation Code 52

B Simulation Analysis 57

C Simulation Graphics Code 64

D Application Code 71

TABLES

Table

1.1 Parameter Combinations Used in Simulation Study 4

3.1 Parameter estimates for Beta(2,2) distribution 22

3.2 Parameter estimates for Beta(2,6) distribution 23

3.3 Parameter estimates for Beta(0.5,0.5) distribution 24

3.4 Parameter estimates for Beta(0.2,0.5) distribution 26

3.5 Parameter estimates for Beta(0.2,2) distribution 29

3.6 Parameter estimates for Beta(1,1) distribution 31

3.7 CRLB of MLE variance for Beta(2,2). 34

3.8 CRLB of MLE variance for Beta(2,6). 35

3.9 CRLB of MLE variance for Beta(0.5,0.5). 35

3.10 CRLB of MLE variance for Beta(0.2,0.5). 35

3.11 CRLB of MLE variance for Beta(0.2,2). 35

3.12 CRLB of MLE variance for Beta(1,1). 36

3.13 Variance estimates for Beta(2,2). 36

3.14 Variance estimates for Beta(2,6). 36

3.15 Variance estimates for Beta(0.5,0.5). 37

3.16 Variance estimates for Beta(0.2,0.5). 37

3.17 Variance estimates for Beta(0.2,2). 37

3.18 Variance estimates for Beta(1,1). 38

4.1 Parameter Estimates from five estimation methods 40

4.2 Proportion of Major League players below Mendoza line. 41

4.3 Proportion fo Major League players batting .400 42

4.4 Mean proportion of workers exposed to each level of radiation 44

4.5 Parameter estimates for the < 100 mrem exposure group. 45
4.6 Parameter estimates for the $100 - 250$ mrem exposure group. 45
4.7 Parameter estimates for the $250 - 500$ mrem exposure group. 48
4.8 Parameter estimates for the $500 - 750$ mrem exposure group. 48
4.9 Parameter estimates for the $750 - 1000$ mrem exposure group. 49
4.10 Parameter estimates for the > 1000 mrem exposure group. 49

FIGURES

Figure

1.1 Beta distributions to be studied in simulation. 3

1.2 Unimodal, symmetric about 0.5 beta pdfs. 5

1.3 Unimodal, skewed beta pdfs. 5

1.4 U-shaped, symmetric about 0.5 beta pdfs. 6

1.5 U-shaped, skewed beta pdfs. 6

1.6 J-shaped beta pdfs. 7

3.1 Legend for Figures 3.2 through 3.7 21

3.2 Results for Beta(2,2) symmetric unimodal distribution. 21

3.3 Results for Beta(2,6) skewed unimodal distribution. 23

3.4 Results for Beta(0.5,0.5) symmetric U-shaped distribution. 25

3.5 Results for Beta(0.2,0.5) skewed U-shaped distribution. 27

3.6 Results for Beta(0.2,2) reverse J-shaped distribution. 29

3.7 Results for Beta(1,1) uniform distribution. 30

4.1 Beta densities from estimated parameters with batting average data . 41

4.2 Proportion of workers exposed to < 100 millirems 45

4.3 Proportion of workers exposed to $100 - 250$ millirems 46

4.4 Proportion of workers exposed to $250 - 500$ millirems 46

4.5 Proportion of workers exposed to $500 - 750$ millirems 47

4.6 Proportion of workers exposed to $750 - 1000$ millirems 47

4.7 Proportion of workers exposed to > 1000 millirems 48

1. THE BETA DISTRIBUTION

1.1 Introduction

The beta distribution is characterized by two shape parameters, α and β, and is used to model phenomena that are constrained to be between 0 and 1, such as probabilities, proportions, and percentages. The beta is also used as the conjugate prior distribution for binomial probabilities in Bayesian statistics (Gelman and Carlin 2004). When used in this Bayesian setting, $\alpha - 1$ may be thought of as the prior number of successes and $\beta - 1$ may be thought of as the prior number of failures for the phenomena of interest. With the widespread applicability of the beta distribution, it is important to estimate, with some degree of accuracy, the parameters of the observed data. We present a simulation study to explore the efficacy of five different estimation methods for determining the parameters of beta-distributed data. We apply the same five estimation techniques to batting averages from the Major League Baseball 2006 season and to proportions of workers exposed to detectable levels of radiation at Department of Energy facilities.

1.2 Literature Review

The two-parameter probability density function of the beta distribution with shape parameters α and β is

$$f(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad 0 \le x \le 1, \quad \alpha > 0, \quad \beta > 0. \tag{1.1}$$

The parameters α and β are symmetrically related by

$$f(x|\alpha,\beta) = f(1-x|\beta,\alpha); \tag{1.2}$$

that is, if X has a beta distribution with parameters α and β, then $1 - X$ has a beta distribution with parameters β and α (Kotz 2006).

The shape of the beta distribution can change dramatically with changes in the parameters, as described below.

- When $\alpha = \beta$ the distribution is unimodal and symmetric about 0.5. Note that $\alpha = \beta = 1$ is equivalent to the Uniform (0,1) distribution. The distribution becomes more peaked as α and β increase. (See Figure 1.2.)

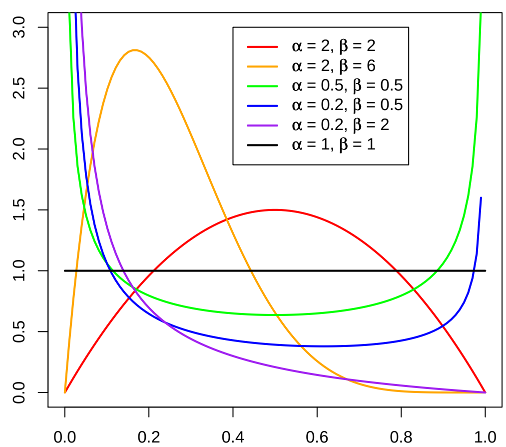
- When $\alpha > 1$ and $\beta > 1$ the distribution is unimodal and skewed with its single mode at $x = (\alpha - 1)/(\alpha + \beta - 2)$. The distribution is strongly right-skewed when β is much greater than α, but the distribution gets less skewed and the mode approaches 0.5 as α and β approach each other. (See Figure 1.3.) The distributions would be left-skewed if the α and β values were switched.

- When $\alpha = \beta < 1$ the distribution is U-shaped and symmetric about 0.5. The case where $\alpha = \beta = 0.5$ is known as the arc-sine distribution, used in statistical communication theory and in the study of the simple random walk. The distribution pushes mass out from the center to the tails as α and β decrease. (See Figure 1.4.)

- When $\alpha < 1$ and $\beta < 1$ the distribution is U-shaped and skewed with an antimode at $x = (\alpha - 1)/(\alpha + \beta - 2)$. The distribution gets less skewed and the antimode approaches 0.5 as α and β approach each other. (See Figure 1.5.) Switching the α and β values would reverse the direction of the skew.

- When $\alpha > 1$, $\beta \leq 1$ the distribution is strictly increasing, a J-shaped beta distribution with no mode or antimode. The distribution becomes more curved as β decreases. (See Figure 1.6.) Switching the α and β values yields a reverse J-shaped beta distribution.

Figure 1.1: Beta distributions to be studied in simulation; these parameter combinations were chosen for their representation of the shapes outlined previously.



To study the parameter estimation of the beta distribution, we consider a variety of parameter combinations, representing each of the previously outlined shapes of the beta distribution. Table 1.1 contains the parameter combinations that we will use in our simulations; Figure 1.1 illustrates the chosen distributions.

Table 1.1: Parameter Combinations Used in Simulation Study

	1	2	3	4	5	6
Alpha	2	2	0.5	0.2	0.2	1
Beta	2	6	0.5	0.5	2	1

Figure 1.2: Unimodal, symmetric about 0.5 beta pdfs. Note that $\alpha = \beta = 1$ is equivalent to the Uniform (0,1) distribution. The distribution becomes more peaked as α and β increase.

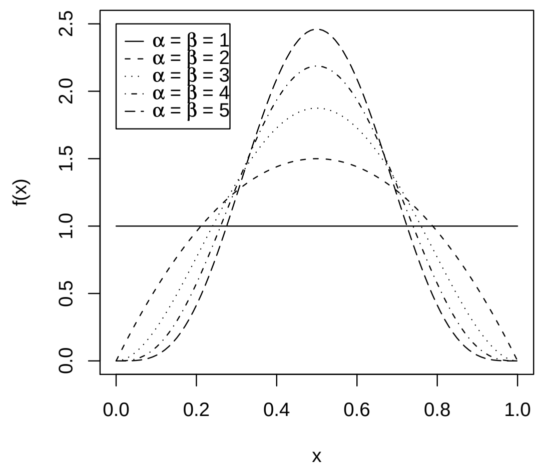


Figure 1.3: Unimodal, skewed beta pdfs. The mode of these distributions occurs at $x = (\alpha - 1)/(\alpha + \beta - 2)$. The distribution is strongly right-skewed when $\beta >> \alpha$, but the distribution gets less skewed and the mode approaches 0.5 as α and β approach each other. The distributions would be left-skewed if the α and β values were switched.

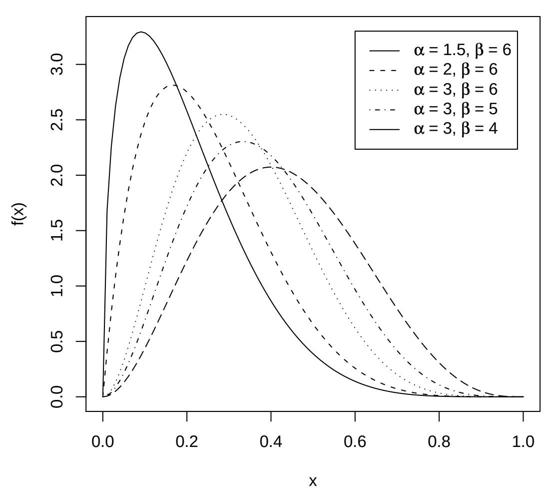


Figure 1.4: U-shaped, symmetric about 0.5 beta pdfs. The distribution pushes mass out from the center to the tails as α and β decrease.

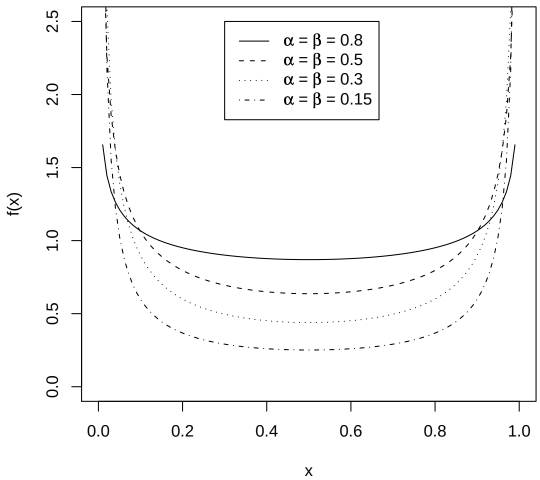


Figure 1.5: U-shaped, skewed beta pdfs with an antimode at $x = (\alpha - 1)/(\alpha + \beta - 2)$. The distribution gets less skewed and the antimode approaches 0.5 as α and β approach each other. Switching the α and β values would reverse the direction of the skew.

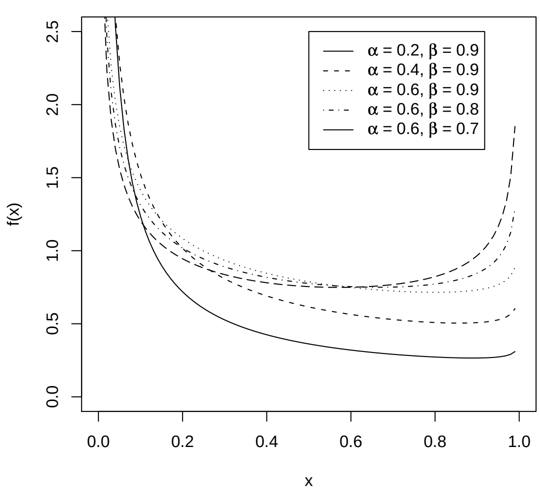
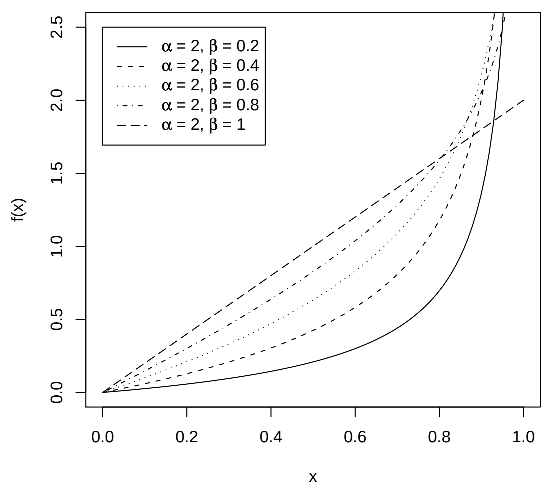


Figure 1.6: J-shaped beta pdfs with no mode or antimode. The distribution becomes more curved as β decreases. Switching the α and β values yields a reverse J-shaped beta distribution.



2. PARAMETER ESTIMATION

Common methods of estimation of the parameters of the beta distribution are maximum likelihood and method of moments. The maximum likelihood equations for the beta distribution have no closed-form solution; estimates may be found through the use of an iterative method. The method of moments estimators are more straightforward and in closed form. We examine both of these estimators here, along with three other proposed estimators. Specifically, we will look at the program evaluation and review technique (PERT) estimators, a method of moments type estimator that we developed using the mean and variance estimates of the two-sided power (TSP) distribution, and an estimator based on the quartiles of the beta distribution.

2.1 Maximum Likelihood Estimators

A well-known method of estimating parameters is the maximum likelihood approach. We define the likelihood function for an iid sample $X_1, \ldots, X_n$ from a population with pdf $f(x|\theta_1, \ldots, \theta_k)$ as $L(\theta_1, \ldots, \theta_k|x_1, \ldots, x_n) = \prod_{i=1}^{n} f(x_i|\theta_1, \ldots, \theta_k)$. The maximum likelihood estimator (MLE) is the parameter value for which the observed sample is most likely. Possible MLEs are solutions to $\frac{\partial}{\partial\theta_i} L(\theta|\mathbf{X}) = 0$, $i = 1, \ldots, k$. We may verify that the points we find are maxima, as opposed to minima, by checking the second derivative of the likelihood function to make sure it is less than zero. Many times it is easier to work with the log likelihood function, $\log L(\theta|\mathbf{X})$, as derivatives of sums are more appealing than derivatives of products (Casella and Berger 2002). MLEs are desirable estimators because they are consistent and asymptotically efficient; that is, they converge in probability to the parameter they are estimating and achieve the lower bound on variance.

The likelihood function for the beta distribution is

$$
\begin{aligned}
L(\alpha, \beta|\mathbf{X}) &= \prod_{i=1}^{n} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x_i^{\alpha-1}(1-x_i)^{\beta-1} \\
&= \left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right)^n \prod_{i=1}^{n}(x_i)^{\alpha-1} \prod_{i=1}^{n}(1-x_i)^{\beta-1} \qquad (2.1)
\end{aligned}
$$

yielding the log likelihood

$$
\begin{aligned}
\log L(\alpha, \beta|\mathbf{X}) =& n\log(\Gamma(\alpha+\beta)) - n\log(\Gamma(\alpha)) - n\log(\Gamma(\beta)) \\
&+ (\alpha-1)\sum_{i=1}^{n}\log(x_i) + (\beta-1)\sum_{i=1}^{n}\log(1-x_i). \qquad (2.2)
\end{aligned}
$$

To solve for our MLEs of α and β we take the derivative of the log likelihood with respect to each parameter, set the partial derivatives equal to zero, and solve for $\hat{\alpha}_{MLE}$ and $\hat{\beta}_{MLE}$:

$$
\begin{aligned}
\frac{\partial}{\partial\alpha}\log L(\alpha, \beta|\mathbf{X}) =& \frac{n\Gamma'(\alpha+\beta)}{\Gamma(\alpha+\beta)} - \frac{n\Gamma'(\alpha)}{\Gamma(\alpha)} + \sum_{i=1}^{n}\log(x_i) \overset{set}{=} 0 \qquad (2.3) \\
\frac{\partial}{\partial\beta}\log L(\alpha, \beta|\mathbf{X}) =& \frac{n\Gamma'(\alpha+\beta)}{\Gamma(\alpha+\beta)} - \frac{n\Gamma'(\beta)}{\Gamma(\beta)} + \sum_{i=1}^{n}\log(1-x_i) \overset{set}{=} 0. \qquad (2.4)
\end{aligned}
$$

There is no closed-form solution to this system of equations, so we will solve for $\hat{\alpha}_{MLE}$ and $\hat{\beta}_{MLE}$ iteratively, using the Newton-Raphson method, a tangent method for root finding. In our case we will estimate $\hat{\theta} = (\hat{\alpha}, \hat{\beta})$ iteratively:

$$\hat{\theta}_{i+1} = \hat{\theta}_i - \mathbf{G^{-1}g}, \qquad (2.5)$$

where $\mathbf{g}$ is the vector of normal equations for which we want

$$\mathbf{g} = \begin{bmatrix} g_1 & g_2 \end{bmatrix},$$

with

$$
\begin{aligned}
g_1 =& \psi(\alpha) - \psi(\alpha+\beta) - \frac{1}{n}\sum_{i=1}^{n}\log(x_i) \quad \text{and} \qquad (2.6) \\
g_2 =& \psi(\beta) - \psi(\alpha+\beta) - \frac{1}{n}\sum_{i=1}^{n}\log(1-x_i), \qquad (2.7)
\end{aligned}
$$

and **G** is the matrix of second derivatives

$$\mathbf{G} = \begin{bmatrix} \frac{dg_1}{d\alpha} & \frac{dg_1}{d\beta} \\ \frac{dg_2}{d\alpha} & \frac{dg_2}{d\beta} \end{bmatrix}$$

where

$$\frac{dg_1}{d\alpha} =\psi'(\alpha) - \psi'(\alpha+\beta), \tag{2.8}$$

$$\frac{dg_1}{d\beta} =\frac{dg_2}{d\alpha} = -\psi'(\alpha+\beta), \tag{2.9}$$

$$\frac{dg_2}{d\beta} =\psi'(\beta) - \psi'(\alpha+\beta), \tag{2.10}$$

and $\psi(\alpha)$ and $\psi'(\alpha)$ are the di- and tri-gamma functions defined as

$$\psi(\alpha) =\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} \quad \text{and} \tag{2.11}$$

$$\psi'(\alpha) =\frac{\Gamma''(\alpha)}{\Gamma(\alpha)} - \frac{\Gamma'(\alpha)^2}{\Gamma(\alpha)^2}. \tag{2.12}$$

The Newton-Raphson algorithm converges, as our estimates of α and β change by less than a tolerated amount with each successive iteration, to $\hat{\alpha}_{MLE}$ and $\hat{\beta}_{MLE}$.

2.2 Method of Moments Estimators

The method of moments estimators of the beta distribution parameters involve equating the moments of the beta distribution with the sample mean and variance (Bain and Engelhardt 1991).

The moment generating function for a moment of order t is

$$\begin{aligned} E(X^t) &= \int_0^1 \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}x^t dx \\ &=1 + \sum_{k=1}^{\infty} \left(\prod_{r=0}^{k-1} \frac{\alpha+r}{\alpha+\beta+r} \right) \frac{t^k}{k!} \\ &=\frac{\Gamma(\alpha+t)\Gamma(\alpha+\beta)}{\Gamma(\alpha+\beta+t)\Gamma(\alpha)}. \end{aligned} \tag{2.13}$$

The first moment of the beta distribution is, then,

$$E(X) = \frac{\alpha}{\alpha+\beta}. \tag{2.14}$$

The second moment of the beta distribution is

$$E(X^2) = \frac{(\alpha+1)\alpha}{(\alpha+\beta+1)(\alpha+\beta)}. \tag{2.15}$$

We can then solve for the variance of the beta distribution as

$$\begin{aligned} Var(X) =& E(X^2) - (E(X))^2 \\ =& \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}. \end{aligned} \tag{2.16}$$

Our method of moments estimators are found by setting the sample mean, $\bar{X}$, and variance, $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$, equal to the population mean and variance

$$\bar{X} = \frac{\alpha}{\alpha+\beta} \quad \text{and} \tag{2.17}$$

$$S^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}. \tag{2.18}$$

To obtain estimators of α and β, we solve equations 2.17 and 2.18 for α and β in terms of $\bar{X}$ and S^2. First, we solve for β (in terms of α):

$$\begin{aligned} &(\alpha+\beta)\bar{X} = \alpha \\ &\Rightarrow \beta\bar{X} = \alpha - \alpha\bar{X} \\ &\Rightarrow \beta = \frac{\alpha}{\bar{X}} - \alpha. \end{aligned}$$

Next we solve for α:

$$
\begin{aligned}
&\alpha\beta = (\alpha+\beta)^2(\alpha+\beta+1)S^2 \\
&\Rightarrow \frac{\alpha^2}{\bar{X}} - \alpha^2 = \left(\alpha + \frac{\alpha}{\bar{X}} - \alpha\right)^2 \left(\alpha + \frac{\alpha}{\bar{X}} - \alpha + 1\right) S^2 \\
&\Rightarrow \frac{\alpha^2}{\bar{X}} - \alpha^2 = \left(\frac{\alpha}{\bar{X}}\right)^2 \left(\frac{\alpha}{\bar{X}} + 1\right) S^2 \\
&\Rightarrow \alpha^2 \left(\frac{1}{\bar{X}} - 1\right) = \alpha^2 \left(\frac{1}{\bar{X}^2}\right) \left(\frac{\alpha}{\bar{X}} + 1\right) S^2 \\
&\Rightarrow \frac{1}{\bar{X}} - 1 = \frac{1}{\bar{X}^2} \left(\frac{\alpha}{\bar{X}} + 1\right) S^2 \\
&\Rightarrow \left(\frac{1}{\bar{X}} - 1\right) \left(\frac{\bar{X}^2}{S^2}\right) = \frac{\alpha}{\bar{X}} + 1 \\
&\Rightarrow \left(\frac{\bar{X}(1-\bar{X})}{S^2}\right) = \frac{\alpha}{\bar{X}} + 1 \\
&\Rightarrow \alpha = \bar{X} \left(\frac{\bar{X}(1-\bar{X})}{S^2} - 1\right).
\end{aligned}
$$

Finally, we express β in terms of $\bar{X}$ and S^2:

$$
\begin{aligned}
&\beta = \frac{\alpha}{\bar{X}} - \alpha \\
&\Rightarrow \beta = \alpha \left(\frac{1-\bar{X}}{\bar{X}}\right) \\
&\Rightarrow \beta = \left(\frac{1-\bar{X}}{\bar{X}}\right) \bar{X} \left(\frac{\bar{X}(1-\bar{X})}{S^2} - 1\right) \\
&\Rightarrow \beta = \left(1-\bar{X}\right) \left(\frac{\bar{X}(1-\bar{X})}{S^2} - 1\right).
\end{aligned}
$$

Thus our method of moments (MOM) estimates of α and β are

$$
\hat{\alpha}_{MOM} = \bar{X} \left(\frac{\bar{X}(1-\bar{X})}{S^2} - 1\right) \quad \text{and} \tag{2.19}
$$

$$
\hat{\beta}_{MOM} = (1-\bar{X}) \left(\frac{\bar{X}(1-\bar{X})}{S^2} - 1\right). \tag{2.20}
$$

2.3 PERT Estimators

The beta distribution is useful in many areas of application because it gives a distributional form to events over a finite interval. It has been widely applied in

PERT analysis to model variable acitivity times (Farnum and Stanton 1987; Johnson 1997). In PERT a 'most likely' modal value, $\hat{m}$, is determined and converted to $\hat{\mu}$ and $\hat{\sigma}$ via the following equations:

$$\text{If } 0.13 \leq \hat{m} \leq 0.87, \qquad \text{then } \hat{\mu} = \frac{4\hat{m}+1}{6}, \quad \hat{\sigma} = \frac{1}{6}. \tag{2.21}$$

$$\text{If } \hat{m} < 0.13, \qquad \text{then } \hat{\mu} = \frac{2}{2+1/\hat{m}}, \quad \hat{\sigma} = \left[\frac{\hat{m}^2(1-\hat{m})}{(1+\hat{m})}\right]^{1/2}. \tag{2.22}$$

$$\text{If } \hat{m} > 0.87, \qquad \text{then } \hat{\mu} = \frac{1}{3-2\hat{m}}, \quad \hat{\sigma} = \left[\frac{\hat{m}(1-\hat{m})^2}{(2-\hat{m})}\right]^{1/2}. \tag{2.23}$$

To estimate the parameters of a beta distribution using this PERT method, we set

$$\hat{\mu} = \frac{\alpha}{\alpha+\beta} \quad \text{and} \tag{2.24}$$

$$\hat{\sigma}^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}. \tag{2.25}$$

PERT variance estimates are rather crude, often resulting in wildly inaccurate parameter estimates. For our simulation purposes we will use the variance of our data, S^2, as our estimate of $\hat{\sigma}^2$ as often as possible. Note that this approach is valid only when $\hat{\mu}(1-\hat{\mu})$ is greater than $\hat{\sigma}^2 = S^2$, as negative estimates of α and β will result otherwise. In the case of $\hat{\mu}(1-\hat{\mu})$ being less than S^2 we will use the variance estimates recommended above.

Our PERT estimates of α and β can then be obtained similarly to the MOM estimates, by solving for α and β in terms of $\hat{\mu}$ and $\hat{\sigma}^2$ in equations 2.24 and 2.25:

$$\hat{\alpha}_{PERT} = \hat{\mu}\left(\frac{\hat{\mu}(1-\hat{\mu})}{\hat{\sigma}^2} - 1\right) \tag{2.26}$$

$$\hat{\beta}_{PERT} = (1-\hat{\mu})\left(\frac{\hat{\mu}(1-\hat{\mu})}{\hat{\sigma}^2} - 1\right). \tag{2.27}$$

2.4 Two-Sided Power Distribution Estimators

VanDorp and Kotz suggest that the two-sided power (TSP) distribution has many attributes similar to the beta distribution due to its ability to be a symmetric or skewed unimodal distribution. This distribution is a modification of the triangular distribution that also allows for $J-$ and $U-$shaped pdfs (van Dorp and Kotz 2002a). The TSP probability density function is

$$f(x|\nu,\theta) = \begin{cases} \nu\left(\frac{x}{\theta}\right)^{\nu-1} & 0 < x \leq \theta, \\ \nu\left(\frac{1-x}{1-\theta}\right)^{\nu-1} & \theta \leq x < 1. \end{cases} \tag{2.28}$$

Note that for $\nu = 2$, the TSP distribution reduces to a triangular distribution.

In the literature the TSP distribution has been used in lieu of the beta distribution because its parameters may be obtained more easily and have better interpretation than those of the beta distribution. We propose use of the mean and variance estimators of this distribution to estimate the parameters of the beta distribution because it is hoped that the TSP distribution's similarities to the beta distribution will allow it to accurately capture the center and spread of the data in our simulations.

The two parameters of the TSP distribution are ν, the shape parameter, and θ, the threshold parameter that coincides with the mode of the distribution. The mean and variance of the TSP distribution are

$$E(X) = \frac{(\nu-1)\theta+1}{\nu+1} \tag{2.29}$$

and

$$Var(X) = \frac{\nu-2(\nu-1)\theta(1-\theta)}{(\nu+2)(\nu+1)^2}. \tag{2.30}$$

To obtain maximum likelihood estimates of ν and θ we consider a random sample $\mathbf{X} = (X_1,\ldots,X_s)$ with size s from a TSP distribution. The order statistics for this sample are $X_{(1)} < X_{(2)} < \cdots < X_{(s)}$. The probability density function in 2.28 gives us the likelihood for $\mathbf{X}$

$$L(\mathbf{X}|\nu,\theta) = \nu^s H(\mathbf{X}|\theta)^{(\nu-1)}, \tag{2.31}$$

where

$$H(\mathbf{X}|\theta) = \frac{\prod_{i=1}^{r} X_{(i)} \prod_{i=r+1}^{s}(1 - X_{(i)})}{\theta^r (1-\theta)^{s-r}}, \tag{2.32}$$

and r is defined by $X_{(r)} \leq \theta < X_{(r+1)}$. The MLEs for the TSP distribution may be found in two steps; first determine $\hat{\theta}$ for which equation 2.32 is maximized, then calculate $\hat{n}$ by maximizing $L(\mathbf{X}|\hat{m}, n)$. van Dorp and Kotz (2002b) proved that equation 2.32 attains its maximum at the order statistic

$$\hat{\theta} = X_{(\hat{r})}, \tag{2.33}$$

where

$$\hat{r} = \max_{r \in \{1,\ldots,s\}} \{M(r)\} \tag{2.34}$$

and

$$M(r) = \prod_{i=1}^{r} \frac{X_i}{X_r} \prod_{i=r+1}^{s} \frac{1 - X_i}{1 - X_r}. \tag{2.35}$$

Next, noting that $H(\mathbf{X}|\hat{\theta}) = M(\hat{r})$, we obtain $\hat{\nu}$ by maximizing the log likelihood with respect to ν:

$$\log L(\mathbf{X}|\nu, \hat{\theta}) = s \log(\nu) + (\nu - 1) \log(H(\mathbf{X}|\hat{\theta})) \tag{2.36}$$

$$\frac{\partial \log L(\mathbf{X}|\nu, \hat{\theta})}{\partial \nu} = \frac{s}{\nu} + \log(H(\mathbf{X}|\hat{\theta})) \overset{set}{=} 0 \tag{2.37}$$

$$\hat{\nu} = \frac{-s}{\log(M(\hat{r}))}. \tag{2.38}$$

Substituting these maximum likelihood estimates of the TSP parameters into our equations for the mean and variance, we obtain

$$\hat{\mu} = \frac{(\hat{\nu} - 1)\hat{\theta} + 1}{\hat{\nu} + 1} \quad \text{and} \tag{2.39}$$

$$\hat{\sigma}^2 = \frac{\hat{\nu} - 2(\hat{\nu} - 1)\hat{\theta}(1 - \hat{\theta})}{(\hat{\nu} + 2)(\hat{\nu} + 1)^2}. \tag{2.40}$$

Our TSP estimates of α and β are then found using equations similar to our MOM and PERT estimators,

$$\hat{\alpha}_{TSP} = \hat{\mu}\left(\frac{\hat{\mu}(1-\hat{\mu})}{\hat{\sigma}^2} - 1\right) \tag{2.41}$$

$$\hat{\beta}_{TSP} = (1-\hat{\mu})\left(\frac{\hat{\mu}(1-\hat{\mu})}{\hat{\sigma}^2} - 1\right). \tag{2.42}$$

2.5 Quantile Estimators

Quantile estimators may be obtained using an approach similar to the method of moments: if we have two equations and two unknowns, we should be able to solve for our unknown parameters. This suggests we should be able to choose two quantiles of the beta distribution, set the quantile functions equal to the sample quantiles, and solve for α and β. The quantile function for the beta distribution has no closed form, which makes directly solving for α and β impossible. Instead, we make use of the beta distribution's quantile function in `R`, `qbeta`. For our simulations we have chosen to use the first and third quartiles of our simulated data to estimate the parameters that were used to create the data.

The `qbeta` function takes as arguments a specified quantile and two shape parameters, corresponding to α and β. The function then returns the value of a random variable that corresponds to the specified quantile of the beta distribution defined by the two shape parameters given. Our code first determines the values of Q1 and Q3 for a simulated data set, then passes several combinations of α and β to the `qbeta` function. We know the true values of α and β used to generate the data, so we search an interval centered around the true parameter values to see if our quantile estimator can capture the true value. The values returned by `qbeta` are then compared to the actual Q1 and Q3 values obtained from the data. The α and β combination that results in the closest estimate of Q1 and Q3 to the actual values of the quartiles is selected as our $\hat{\alpha}_{QNT}$ and $\hat{\beta}_{QNT}$. When using this method for data with unknown parameters, we propose using $\hat{\alpha}_{MOM} \pm 1$ and $\hat{\beta}_{MOM} \pm 1$ as possible parameter values to pass to our function.

3. SIMULATION

In this simulation we study the beta distribution parameter estimators outlined in the previous chapters, namely maximum likelihood, method of moments, PERT, TSP-based estimators, and quantile estimators.

We simulated data from beta distributions with different parameter combinations, and therefore different shapes, to examine the performance of five parameter estimation methods. These parameter estimates adequately estimated the parameters of the beta distribution if they obtained estimates close to the actual values used to generate the data. We also looked at the effect of sample size on the performance of each of our estimators by simulating samples of size 25, 50, 100, and 500. We calculated the mean squared error (MSE) and bias of each estimator for every combination of parameters and sample size used in this simulation. We here present the methodology of our simulations and the conclusions that we reached based on these simulations.

3.1 Simulation Procedure

For this simulation study, realizations from the beta distribution were obtained using the `R` command `rbeta(n,shape1,shape2)`, where n is the desired sample size, $shape_1$ is the desired α, and $shape_2$ is the desired β. The six parameter combinations we examine here were chosen to capture the range of profiles of the beta distribution. These six combinations may be found in Table 1.1 and Figure 1.1.

The number of simulations that we ran was determined using `power.anova.test` in `R`, where the sample size was determined for 120 estimators we wished to compare (number of estimation methods $\times$ number of parameter combinations $\times$ number of sample sizes) at a power of 0.95. The result of this power analysis was to simulate our parameter estimation 13000 times.

For each of the 13000 iterations a beta distribution having each possible combination of sample size and parameter combination was simulated and the five parameter estimation algorithms were applied. The result was 13000 estimates of α and β for the 120 possible estimators. Tables 3.1 through 3.6 contain the parameter estimates for each of the 120 combinations. For each of these 120 estimates, the bias and MSE were computed as:

$$Bias(\hat{\theta}) = \hat{\theta} - \theta \quad \text{and} \tag{3.1}$$

$$MSE(\hat{\theta}) = Var(\hat{\theta}) + Bias(\hat{\theta})^2 \tag{3.2}$$

where $\theta = (\alpha, \beta)$ and $\hat{\theta} = (\hat{\alpha}, \hat{\beta})$ (Casella and Berger 2002).

3.2 Simulation Results

Figures 3.2 through 3.7 illustrate the results of this simulation. Three graphs are presented for each of the six parameter combinations. The first of the three portrays the log of the absolute value of the bias for each estimator by sample size. The dotted lines portray the bias of the β estimates while the solid lines portray the bias of the α estimates. Because these bias values are presented on the log scale, the lower the value, the smaller the bias, and therefore the better the estimator. Each estimator has its own color that is used for all three graphs. The second graph is of the log of the MSE for each estimator by sample size. Solid lines indicate α estimates while dotted lines indicate β estimates. Again, the log scale maintains the property of lower values indicating smaller MSE. Thus, the best estimators in terms of MSE are lowest on this graph. The third graph plots the twenty beta distributions estimated by the five estimation methods (five methods times four sample sizes). The solid black line plots the true shape of the beta distribution and therefore covers up any estimated distributions that were close to the actual values. Therefore, when interpreting this third graph, the visible colors correspond to the estimation methods that did the most poorly. The legend corresponding to these graphs may be found in Figure 3.1.

Figure 3.2 portrays the results for the symmetric, unimodal Beta(2,2) distribution. The bias and MSE of the α and β estimates are nearly the same for each estimation method. When there appears to be only one line, it is because the α and β bias or MSE values are approximately equal for that sample size. Note that the PERT, TSP, and QNT estimators had lower biases than the MLE and MOM estimators for small sample sizes. Going from samples of size 25 to size 50 resulted in a sharp decrease in bias for the PERT and TSP estimators, with the bias of the $\hat{\beta}_{PERT}$ dropping lower than any estimator at any sample size. The bias of the PERT and TSP estimators increased as sample size increased from 50 to 500, though the bias of the MOM and ML estimates decreased steadily as sample size increased. The QNT estimator had the lowest bias for samples of size 25, but was unique in that its bias increased for samples of size 50 and 100, then decreased again for samples of size 500, though still not getting quite as small as its bias for samples of size 25. The MSE graph shows that MLE and MOM had the lowest MSE for sample sizes greater than 25, with their values being nearly indistinguishable from one another. The QNT estimator had the lowest MSE for samples of size 25. The MSE of all the estimators decreased as sample size increased. All the estimators appear to approximate the Beta(2,2) distribution well according to their density plots. The most visible colors are purple (TSP), which is a little too flat when n=500, and blue (MLE), which is a little too peaked when n=25. Thus, for a distribution that is unimodal and symmetric, we would recommend any of the estimation methods, though for small sample sizes, the QNT estimator appears to be the best, and for large sample sizes, the TSP estimator appears to do the most poorly.

Figure 3.3 portrays the results for the skewed, unimodal Beta(2,6) distribution. The QNT estimator had the lowest bias and MSE for both α and β at all sample sizes. Looking at the density plot, the red QNT estimate is not visible because it

Figure 3.1: Legend for Parameter Combination Graphs in Figures 3.2 through 3.7

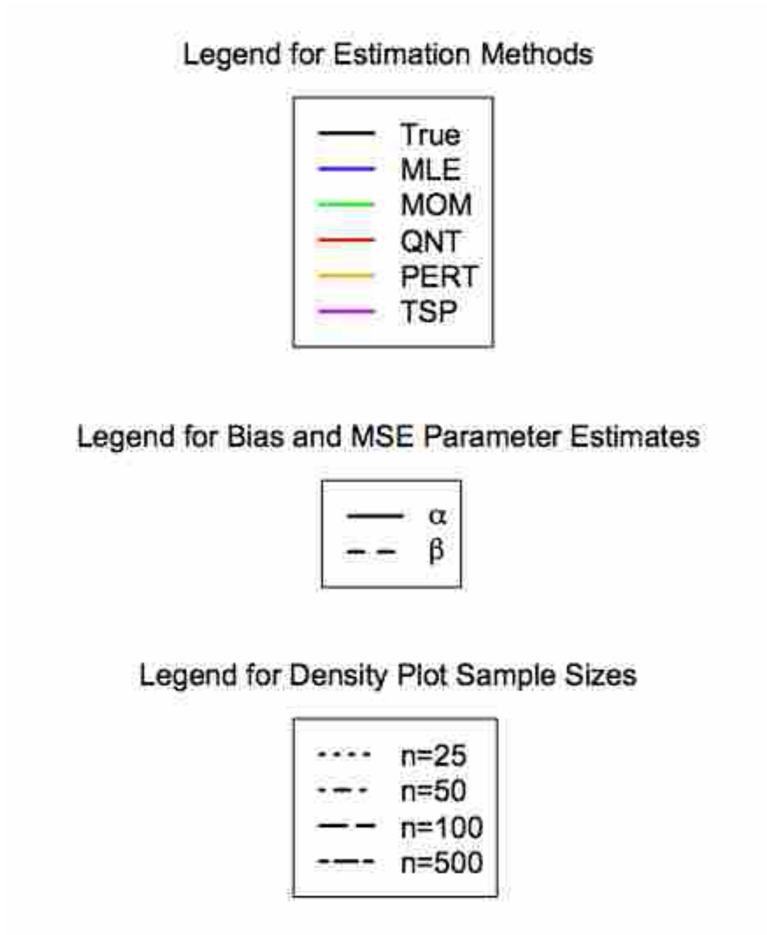


Figure 3.2: Results for Beta(2,2) symmetric unimodal distribution. A legend for these graphs may be found in Figure 3.1.

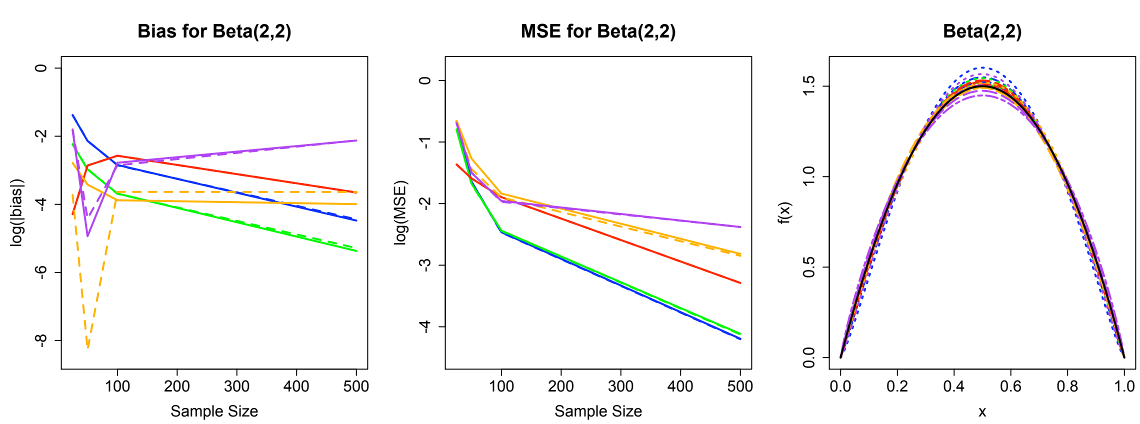


Table 3.1: Parameter estimates for Beta(2,2) distribution

	$\alpha = 2$					$\beta = 2$				
n	MLE	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT
25	2.251	2.107	2.062	2.161	2.014	2.252	2.107	2.024	2.166	2.014
50	2.118	2.051	1.967	2.007	2.057	2.118	2.052	2.000	2.012	2.057
100	2.058	2.025	1.979	1.938	2.076	2.058	2.025	2.026	1.942	2.076
500	2.011	2.005	1.982	1.881	2.026	2.012	2.005	1.974	1.880	2.026

appears to exactly trace the true black density line. The bias of the MOM and ML estimates of α decreased as sample size increased and were consistently lower than the bias of the PERT and TSP α estimates, which decreased from samples of size 25 to 100, but seemed to level off after samples of size 100. The bias of the $\hat{\beta}_{TSP}$ was smaller than the bias of the MOM, ML, and PERT estimates for samples of size 25 and 50. The bias of $\hat{\beta}_{TSP}$ increased for samples of size 100 and 500 while all other β estimate biases decreased as sample size increased. The MSE of all estimates of α and β decreased as sample size increased. The MSE of the MOM and ML α estimates were again quite close to each other, though not as low as the $\hat{\alpha}_{QNT}$ MSE. The MSE of the β estimates for MOM, MLE, PERT, and TSP were much larger than the MSE of $\hat{\beta}_{QNT}$. The density plot highlights the biased nature of the PERT (orange) and TSP (purple) estimates whereas the MOM and MLE estimates are hardly visible. The QNT estimator (red) is completely obscured by the true density, again reflecting the excellent performance of this estimator for this parameter combination. Therefore, for a skewed, unimodal distribution, we recommend the QNT estimator for any sample size, noting that the MOM and ML estimators are also quite good, especially for large sample sizes. The PERT and TSP estimators are not recommended for data with this shape.

Figure 3.4 portrays the results for the symmetric, U-shaped Beta(0.5,0.5) distribution. $\hat{\alpha}_{MOM}$ and $\hat{\beta}_{MOM}$ were the least biased of all the estimators while $\hat{\alpha}_{PERT}$ and

Figure 3.3: Results for Beta(2,6) skewed unimodal distribution. A legend for these graphs may be found in Figure 3.1.

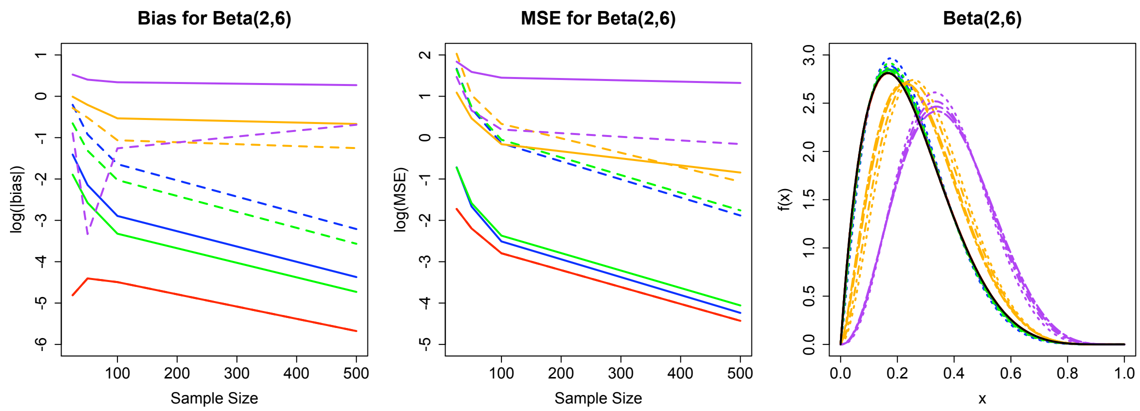


Table 3.2: Parameter estimates for Beta(2,6) distribution

	$\alpha = 2$					$\beta = 6$				
n	MLE	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT
25	2.2445	2.1503	2.9903	3.6906	1.9919	6.8142	6.5200	6.7540	6.4088	5.9919
50	2.1171	2.0761	2.8129	3.4959	2.0122	6.3978	6.2687	6.5969	5.9642	6.0122
100	2.0554	2.0361	2.5868	3.4043	2.0112	6.1938	6.1320	6.3454	5.7158	6.0112
500	2.0126	2.0088	2.5122	3.3097	2.0034	6.0403	6.0283	6.2856	5.4977	6.0034

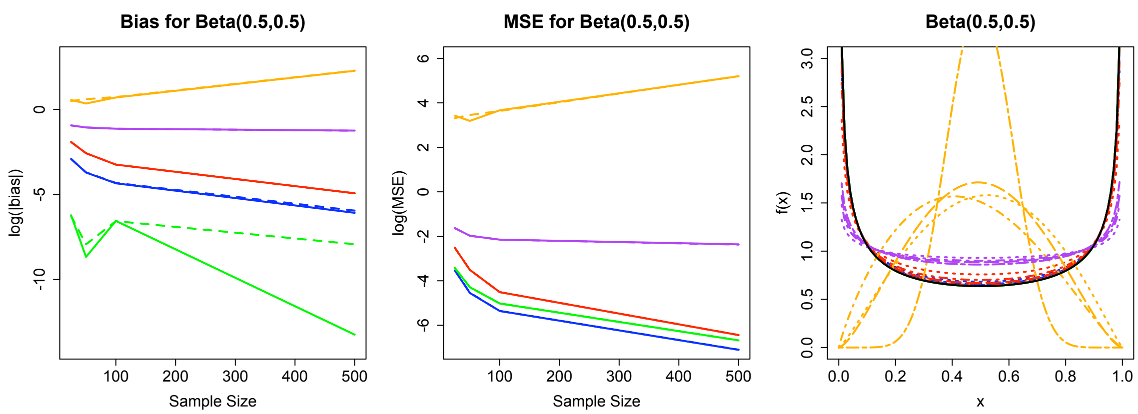
$\hat{\beta}_{PERT}$ were the most biased at all sample sizes. The MOM, ML, and QNT estimators decreased in bias as sample size increased. The TSP estimator had fairly consistent biased across all sample sizes. The PERT estimator actually increased in bias as the sample size increased. The MLEs had the lowest MSE at all sample sizes, though the MOM and QNT estimators also did quite well in terms of MSE. The TSP estimator reduced in MSE slightly as sample size increased. Note that the MSE for the PERT estimator actually increased as the sample size increased. The density plot reveals that PERT (orange) had a hard time reflecting the U shape of the distribution. This likely led to the increasing bias and MSE of the PERT estimators as sample size increased. The MOM, MLE, QNT, and TSP density estimates all reflected the U shape of the distribution. The TSP distribution (purple) did not curve low enough at any sample size and the QNT distribution (red) did not curve low enough at small sample sizes. The MLE and MOM distributions (blue and green) are not visible because the true black density line traces directly over the top of them. Therefore, for a symmetric, U-shaped distribution we recommend the MOM and MLE estimation techniques for any sample size, and even QNT estimation for large sample sizes. The TSP distribution does not quite dip as low as it needs to, but at least reflects the correct shape of the distribution. The PERT estimator would be the worst choice for this type of distribution.

Table 3.3: Parameter estimates for Beta(0.5,0.5) distribution

	$\alpha = 0.5$					$\beta = 0.5$				
n	MLE	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT
25	0.5541	0.5020	2.2444	0.8896	0.6470	0.5544	0.5019	2.1401	0.8898	0.6470
50	0.5245	0.4998	1.9131	0.8457	0.5760	0.5247	0.5004	2.3232	0.8454	0.5760
100	0.5130	0.5014	2.5152	0.8214	0.5389	0.5133	0.5014	2.5686	0.8212	0.5389
500	0.5023	0.5000	10.1782	0.7875	0.5072	0.5026	0.5004	10.2845	0.7861	0.5072

Figure 3.5 portrays the results for the skewed, U-shaped Beta(0.2,0.5) distri-

Figure 3.4: Results for Beta(0.5,0.5) symmetric U-shaped distribution. A legend for these graphs may be found in Figure 3.1.



bution. MOM estimates of α and β had the lowest bias for all sample sizes, with bias decreasing as sample size increased. The bias of the ML estimates of α and β likewise decreased in bias as sample size increased. The QNT estimators increased slightly in bias as sample size increased from 25 to 50, but leveled off for all larger sample sizes. The PERT and TSP estimators also had fairly constant biases across all sample sizes. PERT estimators had the highest biases across the board. MLE estimates had slightly lower MSE than MOM estimates, though both types of estimates decreased in MSE as sample size increased. The QNT, TSP, and PERT estimates had fairly consistent levels of MSE across all sample sizes. The PERT estimates had the largest MSE at all sample sizes for both α and β. The density plot reveals that both the PERT and TSP estimators had a hard time reflecting the U shape of the distribution. The PERT density (orange) centered most of its mass around the lower bound of the distribution while the TSP density (purple) centered its mass closer to 0.45. The QNT distribution (red) was U-shaped, but too shallow at all sample sizes. MLE and MOM estimators (blue and green) are hard to see because they follow the true density line so closely. Therefore, for a skewed, U-shaped distribution, we recommend using ML or MOM parameter estimation. Using QNT estimation will yield parameters corresponding to the correct shape of the distribution, but will not dip as deeply as the distribution should. TSP and PERT estimation will not accurately reflect the true distribution, regardless of sample size.

Table 3.4: Parameter estimates for Beta(0.2,0.5) distribution

	$\alpha = 0.2$					$\beta = 0.5$				
n	MLE	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT
25	0.2168	0.1971	2.9626	1.1079	0.6527	0.5846	0.5183	47.8057	1.2762	0.8159
50	0.2078	0.1987	3.3261	1.1445	0.7091	0.5362	0.5056	48.1907	1.3007	0.8864
100	0.2039	0.1997	3.4425	1.1676	0.7385	0.5170	0.5033	53.1515	1.3166	0.9231
500	0.2009	0.2000	3.5792	1.1873	0.7528	0.5033	0.5003	71.2389	1.3358	0.9410

Figure 3.5: Results for Beta(0.2,0.5) skewed U-shaped distribution. A legend for these graphs may be found in Figure 3.1.

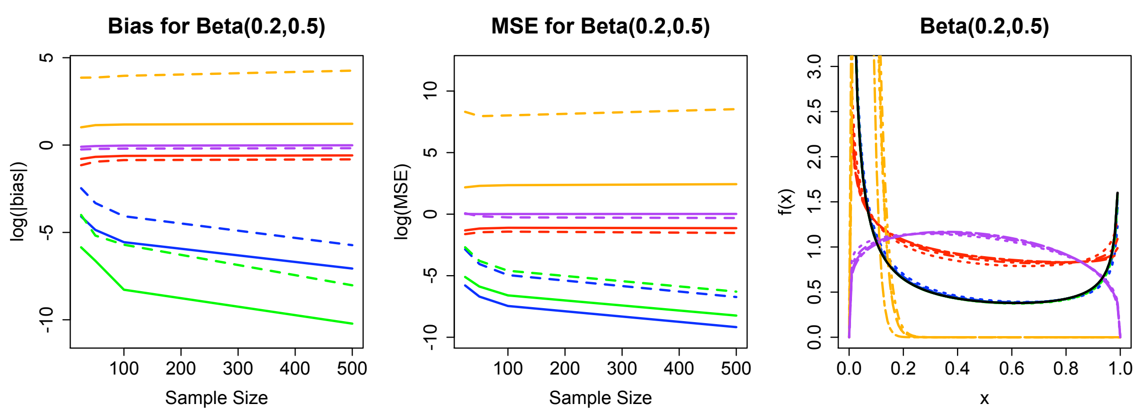


Figure 3.6 contains the results for the reverse J-shaped Beta(0.2,2) distribution. $\hat{\alpha}_{MLE}$ had the lowest bias at all sample sizes, though the bias of $\hat{\alpha}_{MLE}$ and $\hat{\alpha}_{MOM}$ both decreased as sample size increased. $\hat{\beta}_{MLE}$ and $\hat{\beta}_{MOM}$ had nearly identical bias at all sample sizes. Bias of the QNT estimators increased slightly as sample size increased from 25 to 50, but remained fairly constant for larger sample sizes. The TSP estimators likewise increased in bias for small sample sizes, but leveled off for larger sample sizes. The bias of $\hat{\alpha}_{PERT}$ increased from sample size 25 to sample size 100, but remained constant through samples of size 500. The bias of $\hat{\beta}_{PERT}$ decreased from samples of size 25 to samples of size 50, but then increased through samples of size 500. The bias of PERT's estimates were the largest for all sample sizes. The ML estimates had the lowest MSE for all sample sizes, with MSE getting smaller as sample size increased. The MOM estimates likewise decreased in MSE as sample size increased. The QNT estimator decreased slightly in MSE as sample size increased, while the PERT and TSP distributions increased slightly in MSE as sample size increased. $\hat{\alpha}_{TSP}$ had the largest α MSE for all sample sizes while $\hat{\beta}_{PERT}$ had the largest β MSE for all sample sizes. The PERT and TSP estimators once again had a hard time reflecting the true shape of the density at all sample sizes. The density plot reveals that the QNT estimator was quite close to the true density, while once again the ML and MOM estimators are hard to see due to their closeness to the true density. Therefore, for a J-shaped distribution we recommend ML, MOM, or QNT estimation. The PERT and TSP estimators do not yield a distribution that accurately reflects the true distribution.

Figure 3.7 displays the results for the uniform Beta(1,1) distribution. MOM estimators had the lowest bias, with bias decreasing as sample size increased. Likewise, the ML, QNT, and TSP estimators decreased in bias as sample size increased. Only the PERT estimates increased in bias as sample size increased, though for samples of size 25, 50, and 100, PERT did not have the highest bias. PERT actually had a

Figure 3.6: Results for Beta(0.2,2) reverse J-shaped distribution. A legend for these graphs may be found in Figure 3.1.

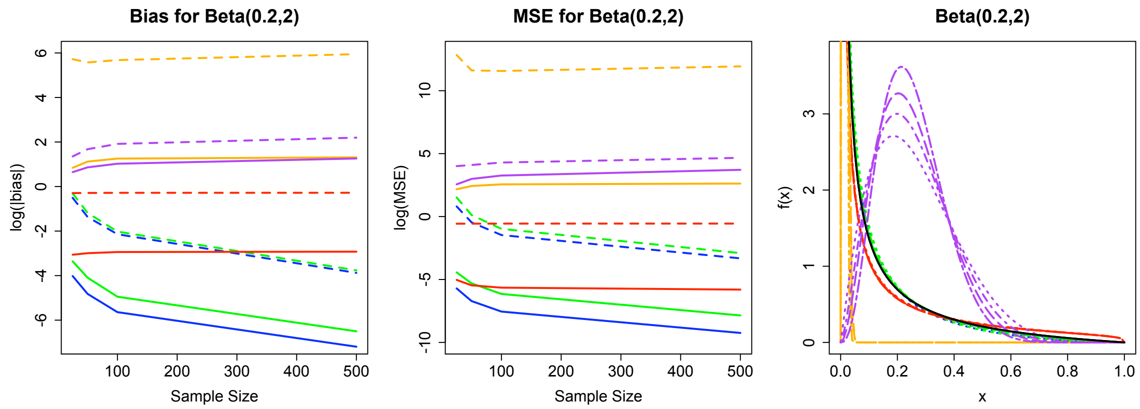


Table 3.5: Parameter estimates for Beta(0.2,2) distribution

	$\alpha = 0.2$					$\beta = 2$				
n	MLE	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT
25	0.2180	0.2349	2.5179	2.1115	0.1531	2.6029	2.7362	306.8130	5.8659	1.2552
50	0.2081	0.2167	3.2675	2.5677	0.1499	2.2554	2.3066	265.3329	7.3482	1.2498
100	0.2036	0.2071	3.7108	2.9878	0.1472	2.1174	2.1322	294.8030	8.7927	1.2453
500	0.2008	0.2015	3.9066	3.7226	0.1463	2.0207	2.0232	383.8551	10.9837	1.2439

lower bias than MLE, QNT, or TSP for samples of size 25. ML, MOM, and TSP estimators all had small MSE, which decreased as sample size increased. The PERT and QNT estimators also decreased in MSE as sample size increased, though the PERT estimates had the largest MSE for all sample sizes. The density plot reveals that the TSP estimators created a density that was too peaked, especially for small sample sizes. It is interesting to note that the PERT estimators created a U-shaped density in response to the generated data at all sample sizes. This is the first time we have seen PERT create a density that was not unimodal. The other estimators, ML, MOM, and QNT, appear to do quite well in the middle of the distribution, though we see them dip a little too low near the boundaries of the distribution. Therefore, for a uniform distribution we recommend ML or MOM estimation, though QNT estimation also does well at large sample sizes. The TSP and PERT estimators over and undershoot the middle part of the distribution (0.2 to 0.8), respectively.

Figure 3.7: Results for Beta(1,1) uniform distribution. A legend for these graphs may be found in Figure 3.1.

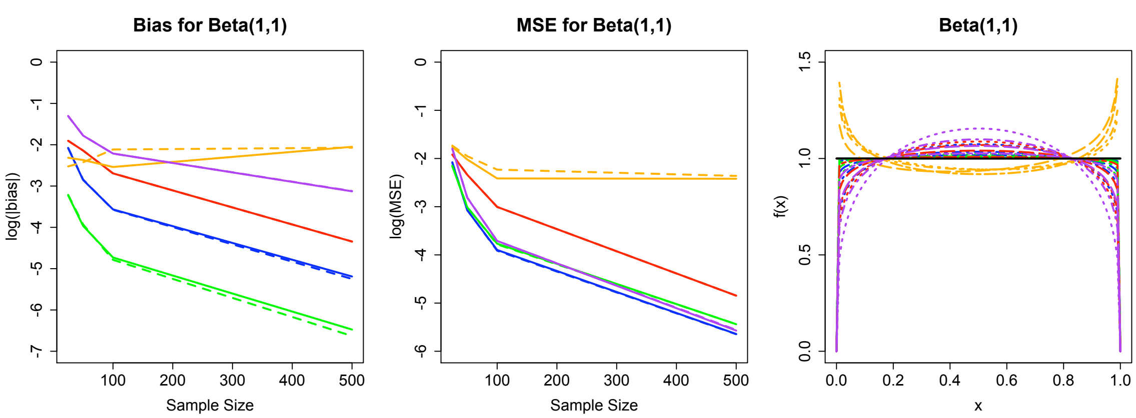


Table 3.6: Parameter estimates for Beta(1,1) distribution

	$\alpha = 1$					$\beta = 1$				
n	MLE	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT
25	1.1246	1.0397	0.9013	1.2714	1.1489	1.1254	1.0402	0.9200	1.2706	1.1489
50	1.0576	1.0190	0.9065	1.1689	1.1176	1.0582	1.0196	0.9115	1.1692	1.1176
100	1.0283	1.0088	0.9211	1.1094	1.0677	1.0280	1.0083	0.8794	1.1093	1.0677
500	1.0056	1.0015	0.8713	1.0438	1.0130	1.0053	1.0013	0.8741	1.0441	1.0130

3.3 Asymptotic Properties of Estimators

Under regularity conditions, the MLE is consistent and asymptotically efficient. For $X_1, X_2, \ldots, X_n$ are iid $f(x|\theta)$. Let $\hat{\theta}$ denote the MLE of θ, and let $\tau(\theta)$ be a continuous function of θ. For members of the exponential family,

$$\sqrt{n}[\tau(\hat{\theta}) - \tau(\theta)] \xrightarrow{L} N[0, \nu(\theta)], \tag{3.3}$$

where $\nu(\theta)$ is the Cramer-Rao Lower Bound (CRLB). That is, $\tau(\theta)$ is a consistent and asymptotically efficient estimator of $\tau(\theta)$ (Casella and Berger 2002).

The log-likelihood of the beta distribution is

$$\begin{aligned}\log L(\alpha, \beta|\mathbf{X}) =& n \log \Gamma(\alpha+\beta) - n \log \Gamma(\alpha) - n \log \Gamma(\beta) \\ &+ (\alpha - 1) \sum_{i=1}^{n} \log(x_i) + (\beta - 1) \sum_{i=1}^{n} \log(1 - x_i)\end{aligned} \tag{3.4}$$

and the partial derivatives with respect to α and β are

$$\begin{aligned}\frac{\partial}{\partial\alpha} \log L(\alpha, \beta|\mathbf{X}) =& \frac{n\Gamma'(\alpha+\beta)}{\Gamma(\alpha+\beta)} - \frac{n\Gamma'(\alpha)}{\Gamma(\alpha)} + \sum_{i=1}^{n} \log(x_i) \\ =& n\psi(\alpha+\beta) - n\psi(\alpha) + \sum_{i=1}^{n} \log(x_i)\end{aligned} \tag{3.5}$$

$$\begin{aligned}\frac{\partial}{\partial\beta} \log L(\alpha, \beta|\mathbf{X}) =& \frac{n\Gamma'(\alpha+\beta)}{\Gamma(\alpha+\beta)} - \frac{n\Gamma'(\beta)}{\Gamma(\beta)} + \sum_{i=1}^{n} \log(1 - x_i) \\ =& n\psi(\alpha+\beta) - n\psi(\beta) + \sum_{i=1}^{n} \log(1 - x_i).\end{aligned} \tag{3.6}$$

Calculation of the CRLB,

$$\nu(\theta) = \frac{\tau'(\theta)^2}{-E[\frac{\partial^2}{\partial\theta^2} \log L(\theta|\mathbf{X})]}, \tag{3.7}$$

requires second partial derivatives with respect to α and β:

$$\frac{\partial^2}{\partial\alpha^2} \log L(\alpha, \beta|\mathbf{X}) = n\psi'(\alpha+\beta) - n\psi'(\alpha) \tag{3.8}$$

$$\frac{\partial^2}{\partial\beta^2} \log L(\alpha, \beta|\mathbf{X}) = n\psi'(\alpha+\beta) - n\psi'(\beta). \tag{3.9}$$

This gives

$$\nu(\alpha) = \frac{1}{-(n\psi'(\alpha+\beta) - n\psi'(\alpha))} \tag{3.10}$$

and

$$\nu(\beta) = \frac{1}{-(n\psi'(\alpha+\beta) - n\psi'(\beta))}. \tag{3.11}$$

Consequently, the MLEs for α and β are consistent with asymptotic variance approaching $\nu(\alpha)$ and $\nu(\beta)$.

Tables 3.7 through 3.12 contain the variance of our simulated maximum likelihood estimates compared with the asymptotically efficient variance. Note that for every parameter combination the variance of our MLE estimates decreases as sample size increases. The variances of our simulated MLEs never quite reach the computed asymptotic variance for each parameter combination, though the variances of the α MLEs for the Beta(0.2,0.5) and Beta(0.2,2) distributions are quite close to the CRLB when $n = 500$ (see Tables 3.10 and 3.11).

Our quantile estimator utilizes the quantile function, $Q(u) = F^{-1}(u)$, $0 < u < 1$. $\hat{Q}(u) = \hat{F}^{-1}(u)$, where $\hat{F}(x)$ is the empirical cdf. If $X_1, \ldots, X_n$ are iid for $Q(u)$, then

$$\hat{Q}(u) \xrightarrow{L} N\left[Q(u), \frac{u(1-u)}{nf^2(Q(u))}\right]. \tag{3.12}$$

For our estimator we employ a function of $Q(u)$ when $u = (0.25, 0.75)$, the first and third quartiles of the beta distribution, to obtain estimates of α and β. The quantile estimator does well for the symmetric distributions, but would perhaps perform better with the skewed, U-shaped, and J-shaped distributions if quantiles other than the 25^{th} and 75^{th} were selected. There is no closed form for the cdf or quantile function of the beta distribution so we use an iterative method to solve for estimates of α and β. The asymptotic variance of these estimates is non-trivial, so we estimate it via simulation. The same is true of our MOM, TSP, and PERT estimators. Tables 3.13 through 3.18 contain the variance of the QNT, MOM, TSP, and PERT estimates of

α and β. Note that most of the variances get smaller as sample size increases, as we would expect. The variances of the PERT estimates of α and β for the Beta(0.5,0.5) distribution, however, increase with sample size. (See Table 3.15.) Likewise, the variance of $\hat{\alpha}_{TSP}$ for the Beta(0.2,2) distribution increases with sample size. (See Table 3.17.)

Table 3.7: Variance of maximum likelihood parameter estimates from simulation compared to the computed Cramer-Rao Lower Bound on variance for the Beta(2,2) distribution.

n	$\widehat{Var(\hat{\alpha})}$	$Var(\hat{\alpha})$	$\widehat{Var(\hat{\beta})}$	$Var(\hat{\beta})$
25	0.4503	0.1108	0.4534	0.1108
50	0.1792	0.0554	0.1789	0.0554
100	0.0816	0.0277	0.0813	0.0277
500	0.0150	0.0055	0.0148	0.0055

Table 3.8: Variance of maximum likelihood parameter estimates from simulation compared to the computed Cramer-Rao Lower Bound on variance for the Beta(2,6) distribution.

n	$\widehat{Var(\hat{\alpha})}$	$Var(\hat{\alpha})$	$\widehat{Var(\hat{\beta})}$	$Var(\hat{\beta})$
25	0.4281	0.0782	4.6326	0.8301
50	0.1756	0.0391	1.8577	0.4151
100	0.0783	0.0195	0.8356	0.2075
500	0.0143	0.0039	0.1506	0.0415

Table 3.9: Variance of maximum likelihood parameter estimates from simulation compared to the computed Cramer-Rao Lower Bound on variance for the Beta(0.5,0.5) distribution.

n	$\widehat{Var(\hat{\alpha})}$	$Var(\hat{\alpha})$	$\widehat{Var(\hat{\beta})}$	$Var(\hat{\beta})$
25	0.0260	0.0122	0.0262	0.0122
50	0.0100	0.0061	0.0101	0.0061
100	0.0045	0.0030	0.0045	0.0030
500	0.0008	0.0006	0.0008	0.0006

Table 3.10: Variance of maximum likelihood parameter estimates from simulation compared to the computed Cramer-Rao Lower Bound on variance for the Beta(0.2,0.5) distribution.

n	$\widehat{Var(\hat{\alpha})}$	$Var(\hat{\alpha})$	$\widehat{Var(\hat{\beta})}$	$Var(\hat{\beta})$
25	0.0028	0.0017	0.0503	0.0190
50	0.0012	0.0009	0.0161	0.0095
100	0.0006	0.0004	0.0068	0.0048
500	0.0001	0.0001	0.0012	0.0010

Table 3.11: Variance of maximum likelihood parameter estimates from simulation compared to the computed Cramer-Rao Lower Bound on variance for the Beta(0.2,2) distribution.

n	$\widehat{Var(\hat{\alpha})}$	$Var(\hat{\alpha})$	$\widehat{Var(\hat{\beta})}$	$Var(\hat{\beta})$
25	0.0030	0.0016	1.8788	0.5555
50	0.0011	0.0008	0.5653	0.2778
100	0.0005	0.0004	0.2163	0.1389
500	0.0001	0.0001	0.0357	0.0278

Table 3.12: Variance of maximum likelihood parameter estimates from simulation compared to the computed Cramer-Rao Lower Bound on variance for the Beta(1,1) distribution.

n	$\widehat{Var(\hat{\alpha})}$	$Var(\hat{\alpha})$	$\widehat{Var(\hat{\beta})}$	$Var(\hat{\beta})$
25	0.1084	0.0400	0.1095	0.0400
50	0.0429	0.0200	0.0431	0.0200
100	0.0195	0.0100	0.0192	0.0100
500	0.0035	0.0020	0.0035	0.0020

Table 3.13: Variance of α and β estimates for Beta(2,2) distribution computed from simulation.

	$Var(\hat{\alpha})$					$Var(\hat{\beta})$				
n	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT	MLE
25	0.438	0.514	0.472	0.255	0.450	0.441	0.500	0.473	0.255	0.453
50	0.184	0.280	0.223	0.200	0.179	0.183	0.237	0.224	0.200	0.179
100	0.087	0.159	0.137	0.145	0.082	0.086	0.149	0.136	0.145	0.081
500	0.016	0.059	0.078	0.037	0.015	0.016	0.057	0.078	0.037	0.015

Table 3.14: Variance of α and β estimates for Beta(2,6) distribution computed from simulation.

	$Var(\hat{\alpha})$					$Var(\hat{\beta})$				
n	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT	MLE
25	0.462	1.992	3.423	0.178	0.428	4.908	7.038	4.174	0.178	4.633
50	0.199	0.935	2.664	0.111	0.176	2.027	2.445	1.928	0.111	1.858
100	0.092	0.509	2.289	0.061	0.078	0.933	1.276	1.136	0.061	0.836
500	0.017	0.168	2.038	0.012	0.014	0.171	0.263	0.604	0.012	0.151

Table 3.15: Variance of α and β estimates for Beta(0.5,0.5) distribution computed from simulation.

	$Var(\hat{\alpha})$					$Var(\hat{\beta})$				
n	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT	MLE
25	0.032	27.60	0.043	0.059	0.026	0.032	24.82	0.043	0.059	0.026
50	0.014	22.24	0.019	0.024	0.010	0.014	28.31	0.019	0.024	0.010
100	0.007	35.00	0.013	0.009	0.005	0.007	33.63	0.013	0.009	0.005
500	0.001	86.85	0.012	0.002	0.001	0.001	85.92	0.011	0.002	0.001

Table 3.16: Variance of α and β estimates for Beta(0.2,0.5) distribution computed from simulation.

	$Var(\hat{\alpha})$					$Var(\hat{\beta})$				
n	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT	MLE
25	0.006	1.184	0.194	0.063	0.003	0.067	1.8e+3	0.470	0.098	0.050
50	0.003	0.189	0.114	0.052	0.001	0.022	5.9e+2	0.200	0.081	0.016
100	0.001	0.015	0.076	0.040	5.6e-4	0.010	2.7e+2	0.103	0.063	0.007
500	2.6e-4	0.001	0.044	0.015	1.0e-4	0.002	6.3e+1	0.032	0.023	0.001

Table 3.17: Variance of α and β estimates for Beta(0.2,2) distribution computed from simulation.

	$Var(\hat{\alpha})$					$Var(\hat{\beta})$				
n	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT	MLE
25	0.011	3.373	9.28	0.004	0.003	4.039	2.8e+5	40.02	0.012	1.879
50	0.005	1.941	14.31	0.002	0.001	1.037	4.1e+4	31.57	0.005	0.565
100	0.002	0.606	18.26	0.001	0.001	0.358	2.0e+4	27.10	0.002	0.216
500	3.9e-4	2.9e-4	28.59	1.4e-4	9.6e-5	0.054	6.2e+3	25.65	3.9e-4	0.036

Table 3.18: Variance of α and β estimates for Beta(1,1) distribution computed from simulation.

	$Var(\hat{\alpha})$					$Var(\hat{\beta})$				
n	MOM	PERT	TSP	QNT	MLE	MOM	PERT	TSP	QNT	MLE
25	0.114	0.162	0.092	0.125	0.108	0.115	0.170	0.092	0.125	0.109
50	0.049	0.124	0.031	0.083	0.043	0.049	0.134	0.032	0.083	0.043
100	0.023	0.083	0.012	0.045	0.019	0.023	0.093	0.012	0.045	0.019
500	0.004	0.072	0.002	0.008	0.004	0.004	0.078	0.002	0.008	0.004

4. APPLICATIONS

4.1 Batting Averages

Batting averages, computed by dividing a player's number of hits by number of at-bats, are proportions and may be modeled by a beta distribution. For this application we considered the batting averages of the 160 Major League Baseball players with 500 or more at-bats for the 2006 season (ESPN.com 2007).

Of special note in Major League Baseball are batting averages of .200, the lower bound on an acceptable batting average for a Major League player, and .400, a seemingly unreachable batting average in modern day Major League Ball. A batting average of .200 is known as the Mendoza line, named for Mario Mendoza who hit under .200 five of nine seasons in his career. Historically, batting averages listed in the Sunday paper that fell below Mendoza's were said to "fall below the Mendoza line." Today, the Mendoza line is considered by some to be the standard which Major League players should surpass in order to stay in the league. On the other end of the spectrum is the ".400 hitter." The last ballplayer to hit .400 in a season was Ted Williams in 1941; no one else has gotten closer than .390 in the last 66 years. This is largely attributed to changes in strategies of the game.

For the 2006 data we note that no players fall below the Mendoza line or achieve a .400 season. We are interested in the estimated proportion of Major League players that fall below the Mendoza line or achieve a .400 season according to the five estimation methods we have presented above. We would hope that a good estimation technique would estimate parameters for the distribution of the data that accurately reflect the data. In other words, we would expect a good estimator to allow zero players to receive either a .200 or .400 batting average for the season.

A histogram of the data reveals that it is unimodal and fairly symmetric about

.290. According to our earlier simulations, we would think that the QNT estimator will do extremely well, with MOM and ML estimators performing quite well also.

The estimated α and β parameters may be found in Table 4.1. Figure 4.1 illustrates what the distribution of batting averages would look like if the estimated parameters from each estimation method were the true parameters for the distribution. The MOM, MLE, and QNT densities seem to mimic the data better than the other estimators. Both the PERT and TSP estimated densities are shifted to the right of the data.

Table 4.1: Parameter Estimates from five estimation methods

	$\hat{\alpha}$	$\hat{\beta}$
MLE	108.642	271.838
MOM	107.550	269.108
PERT	151.653	272.556
TSP	88.575	196.257
QNT	106.550	268.108

Each of these estimation methods calculates the proportion of Major League players with more than 500 at-bats falling below the Mendoza line to be nearly zero, see Table 4.2. These estimates are believable, as the actual minimum batting average of the data set is 0.220. Looking at the estimated densities, however, we are concerned that the PERT and TSP distributions are shifted to the right of the data. When we look at the proportions of players achieving a .400 batting average, the TSP density estimates 0.09% of the major league players will hit a .400 for the season, while the PERT density estimates that 3.5% of the players will surpass this mark (see Table 4.3). In order to accurately reflect the data, these estimates should be close to zero, like the MLEs, MOM, and QNT estimates, as the highest batting average in the data set was .347. For a data set of this size and shape, it appears that the MLE, MOM, and QNT methods of estimation most accurately reflect the data.

Table 4.2: Proportion of Major League players falling below the Mendoza line according to estimated distributions.

	Pr(BA< .200)
MLE	3.644e-05
MOM	3.967e-05
PERT	2.896e-14
QNT	5.121e-05
TSP	5.304e-06

Figure 4.1: Beta densities from estimated parameters with batting average data

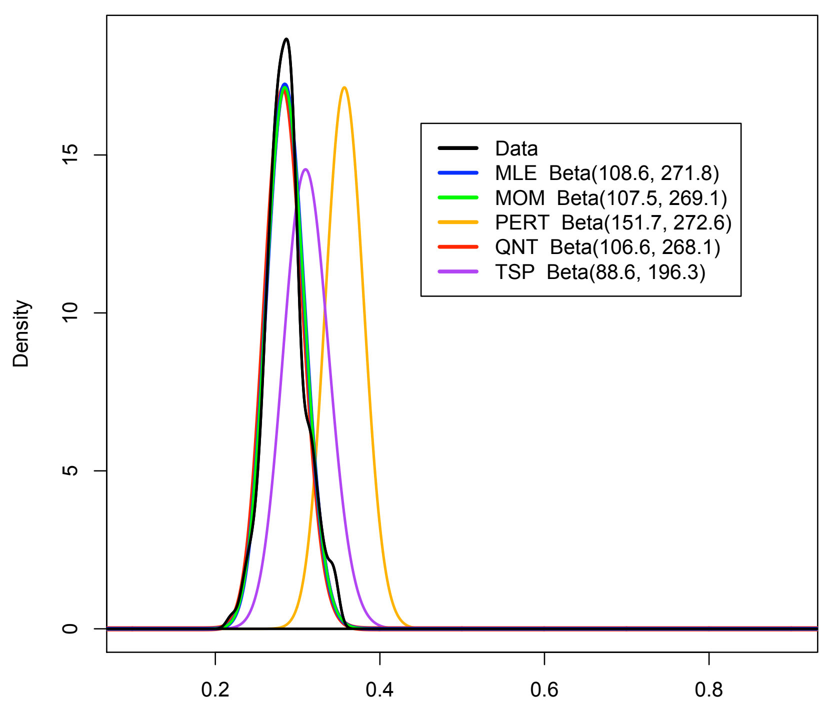


Table 4.3: Proportion of Major League players surpassing a .400 batting average according to estimated distributions.

	Pr(BA> .400)
MLE	1.512e-06
MOM	1.693e-06
PERT	0.03535
QNT	1.426e-06
TSP	0.0008782

4.2 Radiation Exposure

Radiation exposure for workers at the Department of Energy has been monitored since 1987. We have data from 1987 to 2007 on the number of exposed workers and the level of radiation, in millirem, that they were exposed to (energy.gov 2008). There were some workers whose level of exposure was not measureable. For those workers whose level of radiation exposure was detectable, their levels of exposure are divided into the following categories: $<$ 100 millirem, 100-250 millirem, 250-500 millirem, 500-750 millirem, 750-1000 millirem, and $>$ 1000 millirem. We have applied our five estimation methods to these six categories of exposed workers. For each category we have the proportion of exposed workers whose radiation measured in the ranges specified for the 21 years from 1987 to 2007. The estimated densities for each of these ranges have been overlaid on histograms of the proportion data for each range (see Figures 4.2 to 4.7). The estimated parameters for each of these distributions may be found in Table 4.5 through Table 4.10.

Table 4.4 contains the mean proportion of exposed workers with measurable radiation in each category for 1987 to 2007 as estimated by the five estimation methods. The final column in this table contains the average total proportion of workers exposed to a measurable amount of radiation according to the parameters estimated by each technique. Note that the data indicate that 23.96% of workers were exposed to measurable amounts of radiation. ML, MOM, and QNT estimates of the same quantity are within 2% of this value. The PERT and TSP methods, on the other hand, estimate 34.98% and 55.97% of the workers to be exposed to measurable levels of radiation annually on average.

An examination of Figure 4.2 reveals that the QNT, MOM, and ML estimated densities capture the shape of the empirical density better than the PERT and TSP estimated densities. We see in Figure 4.3 that the same is again true, though the TSP estimated density appears to have a support that matches the data better than the PERT estimated density. Figure 4.4 also shows that the PERT density peaks in a different place than the data and that the TSP estimator assigns a somewhat uniform probability to all values of X in the range of the data. In Figure 4.5 we see that the QNT and MOM densities have a peak closest to the peak in the data, while the ML estimated density is strictly decreasing; the PERT estimated density peaks later than the data, and the TSP estimated density looks uniform yet again. Figure 4.6 is one case where the PERT density is the only one to peak near where the data peaks, as the MOM, ML, and QNT densities are all strictly decreasing and the TSP distribution is uniform. Finally, Figure 4.7 shows that the QNT, MOM, ML, and PERT densities all approximate a strictly decreasing pdf and the TSP density is uniform. It is difficult to see the PERT esimate (orange) because it closely follows the QNT (red) estimated density line.

We therefore conclude that the QNT and MOM estimation techniques reflect this data most accurately. The ML technique performed well, but was extremely sensitive to starting values. The PERT technique did well for the > 1000 millirem group, but not very well on all the others. Thus, for data of this size and shape, we would recommend using the QNT or MOM estimation techniques.

Table 4.4: Mean proportion of workers exposed to each level of radiation each year from 1987 to 2007 with mean total proportion of workers exposed as calculated by each estimation method.

	< 100	100-250	250-500	500-750	750-1000	> 1000	Total
Data	0.1956	0.0270	0.0109	0.0033	0.0014	0.0015	0.2396
MLE	0.1976	0.0270	0.0124	0.0055	0.0041	0.0039	0.2504
MOM	0.1956	0.0270	0.0109	0.0033	0.0014	0.0015	0.2396
PERT	0.2738	0.0469	0.0194	0.0063	0.0022	0.0012	0.3498
TSP	0.2621	0.0638	0.0540	0.0746	0.0513	0.0539	0.5597
QNT	0.1846	0.0241	0.0096	0.0029	0.0010	0.0011	0.2233

Table 4.5: Parameter estimates for the < 100 mrem exposure group.

	$\hat{\alpha}$	$\hat{\beta}$
MLE	3.2446	13.1744
MOM	2.5316	10.4110
PERT	4.5507	12.0690
TSP	3.9475	11.1158
QNT	2.3055	10.1849

Figure 4.2: Distribution of proportion of exposed workers with radiation < 100 millirems from 1987 to 2007

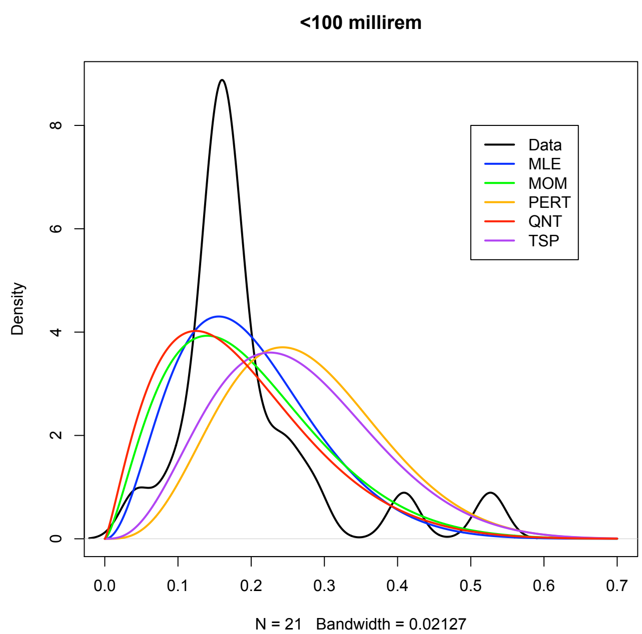


Table 4.6: Parameter estimates for the $100 - 250$ mrem exposure group.

	$\hat{\alpha}$	$\hat{\beta}$
MLE	3.3554	120.8716
MOM	3.1697	114.1887
PERT	9.3810	190.7955
TSP	1.8875	27.6893
QNT	2.8129	113.8319

Figure 4.3: Distribution of proportion of exposed workers with radiation $100 - 250$ millirems from 1987 to 2007

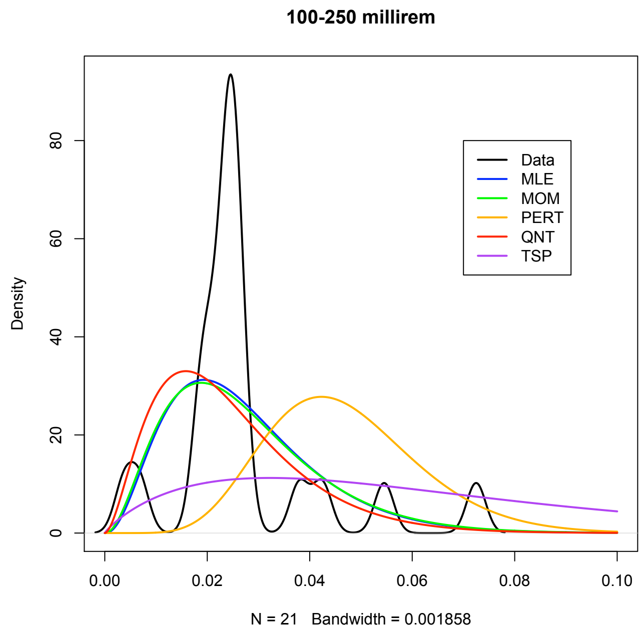


Figure 4.4: Distribution of proportion of exposed workers with radiation $250 - 500$ millirems from 1987 to 2007

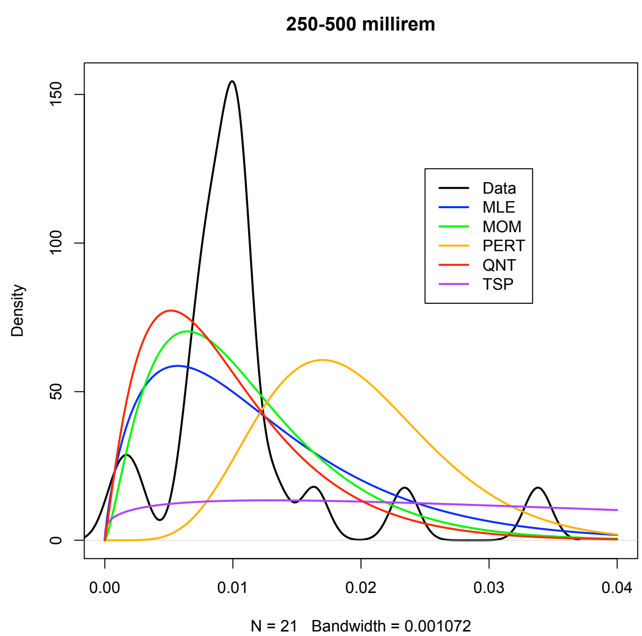


Figure 4.5: Distribution of proportion of exposed workers with radiation $500 - 750$ millirems from 1987 to 2007

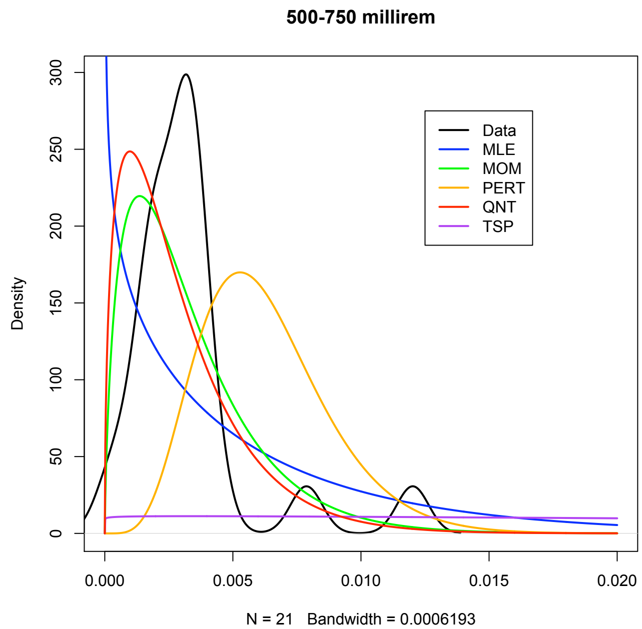


Figure 4.6: Distribution of proportion of exposed workers with radiation $750 - 1000$ millirems from 1987 to 2007

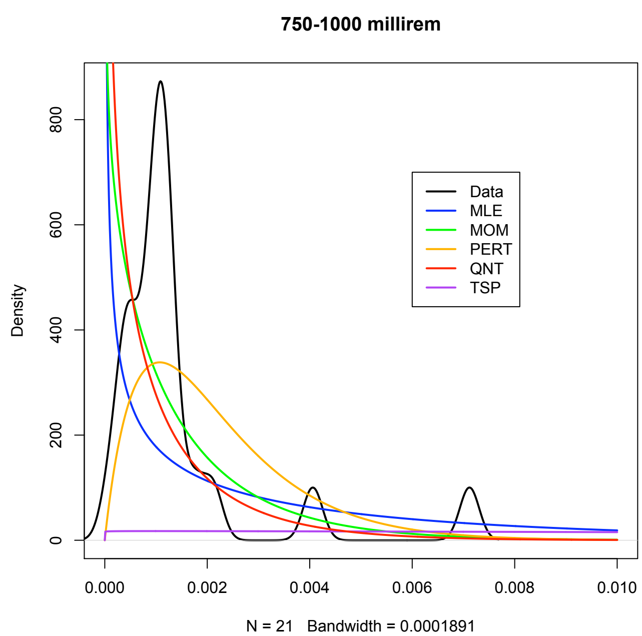


Figure 4.7: Distribution of proportion of exposed workers with radiation > 1000 millirems from 1987 to 2007

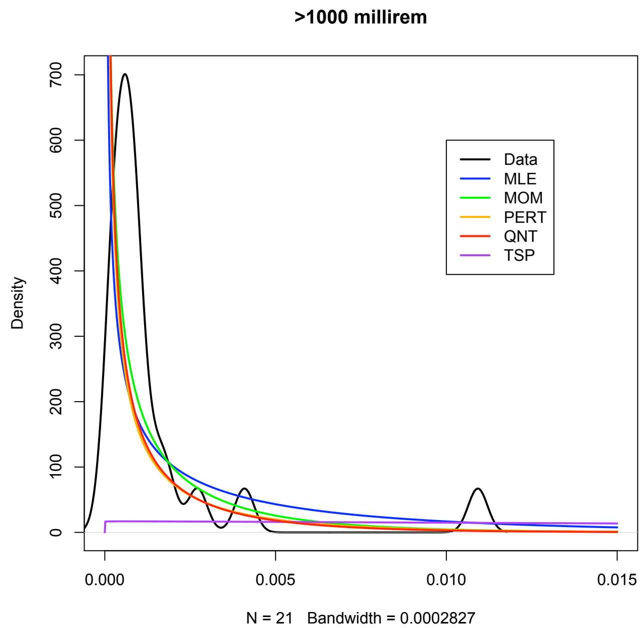


Table 4.7: Parameter estimates for the $250 - 500$ mrem exposure group.

	$\hat{\alpha}$	$\hat{\beta}$
MLE	1.8348	146.6562
MOM	2.4247	220.5724
PERT	7.6952	388.2310
TSP	1.2808	22.4406
QNT	2.1383	220.2859

Table 4.8: Parameter estimates for the $500 - 750$ mrem exposure group.

	$\hat{\alpha}$	$\hat{\beta}$
MLE	0.8101	146.9451
MOM	1.6996	514.4712
PERT	6.1821	977.2371
TSP	1.0389	12.8801
QNT	1.5036	514.2752

Table 4.9: Parameter estimates for the $750 - 1000$ mrem exposure group.

	$\hat{\alpha}$	$\hat{\beta}$
MLE	0.5304	130.1833
MOM	0.8410	593.3511
PERT	1.9751	908.5387
TSP	1.0172	18.8073
QNT	0.6198	593.0245

Table 4.10: Parameter estimates for the > 1000 mrem exposure group.

	$\hat{\alpha}$	$\hat{\beta}$
MLE	0.4258	109.9806
MOM	0.3774	258.9674
PERT	0.2551	212.8918
TSP	1.0094	17.7096
QNT	0.2769	258.3694

BIBLIOGRAPHY

Bain, L. J. and Engelhardt, M. (1991), *Introduction to Probability and Mathematical Statistics*, Duxbury Press.

Casella, G. and Berger, R. L. (2002), *Statistical Inference*, Duxbury Press.

energy.gov (2008), "Radiation Exposure Monitoring System," World Wide Web electronic publication.

ESPN.com (2007), "MLB Player Batting Stats: 2006," World Wide Web electronic publication.

Farnum, N. R. and Stanton, L. W. (1987), "Some Results Concerning the Estimation of Beta Distribution Parameters in PERT," *Journal of the Operational Research Society*, 38, 287–290.

Gelman, A. and Carlin, J. B. (2004), *Bayesian Data Analysis*, Chapman & Hall CRC.

Johnson, D. (1997), "The Triangular Distribution As a Proxy for the Beta Distribution in Risk Analysis," *The Statistician: Journal of the Institute of Statisticians*, 46, 387–398.

Kotz, S. (2006), *Handbook of Beta distribution and its applications.*

R Development Core Team (2008), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Rahman, N. A. (1977), *A Course in Theoretical Statistics*, Charles Griffin & Co.

van Dorp, J. R. and Kotz, S. (2002a), "A Novel Extension of the Triangular Distribution and Its Parameter Estimation," *Journal of the Royal Statistical Society, Series D: The Statistician*, 51, 63–79.

— (2002b), “The Standard Two-sided Power Distribution and Its Properties: With Applications in Financial Engineering,” *The American Statistician*, 56, 90–99.

```
N<-20000

saveda<-matrix(rep(NA,N*120),ncol=120)
savedb<-matrix(rep(NA,N*120),ncol=120)

Alpha<-c(2,2,.5,.2,.2,1)
Beta<-c(2,6,.5,.5,2,1)

keep_iters<-matrix(rep(NA,N*24),ncol=24)

n<-c(25,50,100,500)

for(I in 1:N){
count<-I
ind<- -4
ind2<-0

for(k in 1:4){

for(j in 1:6){
betdat<-rbeta(n=n[k],shape1=Alpha[j],shape2=Beta[j])
ind<-ind+5
ind2<-ind2+1

#### MLE: Newton-Raphson ####
nrand<-n[k]
i<-2
alpha<-rep(Alpha[j],2)
beta<-rep(Beta[j],2)
tol<-10^-3
lim<-10^-4
lim2<--5
eps<-1
maxiter<-100
while(tol<eps & i<maxiter){
# create g matrix - 1st derivs

g1<- digamma(alpha[i-1]) - digamma(alpha[i-1]+beta[i-1]) - sum(log(betdat))/nrand

g2<- digamma(beta[i-1]) - digamma(alpha[i-1]+beta[i-1]) - sum(log(1-betdat))/nrand

g<- c(g1,g2)
```

```
if(g1<lim2 | g2<lim2){
num<-i
i<-maxiter
alpha[i]<-alpha[num-1]
beta[i]<-beta[num-1]
}
else{

# create g' matrix - matrix of 2nd derivs

g1a<- trigamma(alpha[i-1]) - trigamma(alpha[i-1] + beta[i-1])

g1b<- g2a<- -trigamma(alpha[i-1] + beta[i-1])

g2b<- trigamma(beta[i-1]) - trigamma(alpha[i-1] + beta[i-1])

gp<- matrix(c(g1a,g1b,g2a,g2b),ncol=2,byrow=T)

# compute next value

temp<- c(alpha[i-1],beta[i-1]) - solve(gp)%*%g

alpha[i]<- temp[1]

beta[i]<- temp[2]

# see if we've reached our tolerance

eps<- max(abs((alpha[i-1]-alpha[i])/alpha[i-1]),abs((beta[i-1]-beta[i])/beta[i-1]))

# increment the loop!

if(abs(g1a)<lim | abs(g1b)<lim | abs(g2b)<lim){
num<-i
i<-maxiter
alpha[i]<-alpha[num-1]
beta[i]<-beta[num-1]
}
}

i<- i + 1
}
keep_iters[I,ind2]<-i-1
saveda[I,ind]<-alpha[i-1]
savedb[I,ind]<-beta[i-1]

#### MOM ####
```

```
xbar<-mean(betdat)
varx<-var(betdat)
amom<-xbar*((xbar*(1-xbar)/varx)-1)
bmom<-(1-xbar)*((xbar*(1-xbar)/varx)-1)

saveda[I,ind+1]<-amom
savedb[I,ind+1]<-bmom

#### modified MOM:  PERT Approx. ####
y<-density(betdat)$y
x<-density(betdat)$x
top<-which(density(betdat)$y==max(density(betdat)$y))
mo<-density(betdat)$x[top]

# to improve est of var, use var of data
sig2x<-varx

if(mo>=0.13 & mo<=0.87){
mux<-(4*mo+1)/6
# sig2x<-(1/6)^2
}
if(mo<0.13){
mux<-2/(2+(1/mo))
# sig2x<-(mo^2*(1-mo))/(1+mo)
}
if(mo>0.87){
mux<-1/(3-2*mo)
# sig2x<-(mo*(1-mo)^2)/(2-mo)
}

if(mux*(1-mux)<sig2x){
if(mo>=0.13 & mo<=0.87){
sig2x<-(1/6)^2
}
if(mo<0.13){
sig2x<-(mo^2*(1-mo))/(1+mo)
}
if(mo>0.87){
sig2x<-(mo*(1-mo)^2)/(2-mo)
}
aprt<-mux*((mux*(1-mux)/sig2x)-1)
bprt<-(1-mux)*((mux*(1-mux)/sig2x)-1)
}
if(mux*(1-mux)>=sig2x){
aprt<-mux*((mux*(1-mux)/sig2x)-1)
bprt<-(1-mux)*((mux*(1-mux)/sig2x)-1)
}
```

```
saveda[I,ind+2]<-aprt
savedb[I,ind+2]<-bprt

### modified two sided power / triangular: tsp ###
s<-length(betdat)
myind<-order(betdat)

m.fun<-function(r){
prod1<-1
prod2<-1
for(i in 1:(r-1)){
prod1<-prod1*betdat[myind[i]]/betdat[myind[r]]
}
for(i in (r+1):s){
prod2<-prod2*(1-betdat[myind[i]])/(1-betdat[myind[r]])
}
M.stat<-prod1*prod2
return(M.stat)
}

test<-matrix(0,s-1)
for(i in 2:(s-1)){
test[i]<-m.fun(i)
}

rhat<-which(test==max(test))
mhat<-betdat[rhat]
nhat<--s/log(m.fun(rhat))

tmux<-((nhat-1)*mhat+1)/(nhat+1)
tsig2x<-(nhat-2*(nhat-1)*mhat*(1-mhat))/((nhat+2)*(nhat+1)^2)

atsp<-tmux*((tmux*(1-tmux)/tsig2x)-1)
btsp<-(1-tmux)*((tmux*(1-tmux)/tsig2x)-1)

saveda[I,ind+3]<-atsp
savedb[I,ind+3]<-btsp

### modified quantile est:  mne ###
q1<-quantile(betdat,.25)
q3<-quantile(betdat,.75)

loa<-ifelse(Alpha[j]-1<0,0,Alpha[j]-1)
hia<-Alpha[j]+1
lob<-ifelse(Beta[j]-1<0,0,Beta[j]-1)
hib<-Beta[j]+1
```

```
acand<-seq(loa,hia,length=200)
bcand<-seq(lob,hib,length=200)

q1est<-qbeta(.25,shape1=acand,shape2=bcand)
q3est<-qbeta(.75,shape1=acand,shape2=bcand)

my.crit<-(q1-q1est)^2+(q3-q3est)^2
my.keep<-which(my.crit==min(my.crit))

amne<-acand[my.keep]
bmne<-bcand[my.keep]

saveda[I,ind+4]<-amne
savedb[I,ind+4]<-bmne

}
}


}

write.table(keep_iters,file="keepiter.txt")
write.table(saveda,file="allsaveda.txt")
write.table(savedb,file="allsavedb.txt")
```

```
all_alpha<-read.table("all_alpha.txt",header=T)
all_beta<-read.table("all_beta.txt",header=T)
all_iter<-read.table("all_iter.txt",header=T)


mles<-seq(1,120,by=5)
moms<-seq(2,120,by=5)
prts<-seq(3,120,by=5)
tsps<-seq(4,120,by=5)
mnes<-seq(5,120,by=5)

amle<-all_alpha[,mles]
amom<-all_alpha[,moms]
aprt<-all_alpha[,prts]
atsp<-all_alpha[,tsps]
amne<-all_alpha[,mnes]

bmle<-all_beta[,mles]
bmom<-all_beta[,moms]
bprt<-all_beta[,prts]
btsp<-all_beta[,tsps]
bmne<-all_beta[,mnes]

### Identify all obs that didn't converge:
## for MLE, if all_iter=100, that iteration didn't converge...
for(i in 1:ncol(all_iter)){
outMLE<-which(all_iter[,i]==100)
amle[outMLE,i]<-0
bmle[outMLE,i]<-0
outMLE<-which(amle[,i]<0)
amle[outMLE,i]<-0
outMLE<-which(bmle[,i]<0)
bmle[outMLE,i]<-0
}




outPRT<-NULL
for(i in 1:ncol(aprt)){
outPRT[i]<-length(which(aprt[,i]==0))
}
```

```
ns<-kronecker(c(25,50,100,500),rep(1,6))
Alpha<-c(2,2,.5,.2,.2,1)
Beta<-c(2,6,.5,.5,2,1)
cas<-kronecker(rep(1,4),Alpha)
cbs<-kronecker(rep(1,4),Beta)

rbind(ns,cas,cbs)
combs<-paste(ns,cas,cbs,sep=",")

for(i in 1:ncol(amne)){
outMNE<-c(which(amne[,i]==(cas[i]+1)),
which(amne[,i]==(cas[i]-1)))
amne[outMNE,i]<-0
bmne[outMNE,i]<-0
}

## now 0's indicate non-converging parameters...
# When I do the analysis, don't include 0-estimates of parameters.

# we have a problem with the PERT estimator... as usual.
# see allperta.txt and allpertb.txt for new pert estimates.
aprt<-read.table("allperta.txt",header=T)
bprt<-read.table("allpertb.txt",header=T)

for(i in 1:ncol(bprt)){
outPRT<-which(bprt[,i]<0)
bprt[outPRT,i]<-0}

# compute bias and MSE
my.bias.mse<-function(datA,datB){
biasa<-biasb<-msea<-mseb<-amean<-bmean<-avar<-bvar<-matrix(NA,24,ncol=1)
for(i in 1:ncol(datA)){
calca<-datA[which(datA[,i]!=0),i]
calcb<-datB[which(datB[,i]!=0),i]
amean[i]<-mean(calca)
bmean[i]<-mean(calcb)
avar[i]<-var(calca)
bvar[i]<-var(calcb)
biasa[i]<-amean[i]-cas[i]
biasb[i]<-bmean[i]-cbs[i]
msea[i]<-avar[i]+biasa[i]^2
mseb[i]<-bvar[i]+biasb[i]^2
}
return(cbind(ns,cas,cbs,biasa,biasb,msea,mseb,amean,bmean,avar,bvar))
}
```

```
res_mle<-my.bias.mse(amle,bmle)
res_mom<-my.bias.mse(amom,bmom)
res_prt<-my.bias.mse(aprt,bprt)
res_tsp<-my.bias.mse(atsp,btsp)
res_mne<-my.bias.mse(amne,bmne)

results<-rbind(res_mle,res_mom,res_prt,res_tsp,res_mne)
#write.table(results,"computed_results.txt",row.names=FALSE)
results<-read.table("computed_results.txt",header=T)
names(results)<-c("ns","cas","cbs","biasa","biasb","msea",
"mseb","amean","bmean","avar","bvar")


c1<-seq(1,24,by=6)
c2<-seq(2,24,by=6)
c3<-seq(3,24,by=6)
c4<-seq(4,24,by=6)
c5<-seq(5,24,by=6)
c6<-seq(6,24,by=6)

esta<-estb<-NULL
for(i in 1:6){
esta<-rbind(esta,t(res_mle[seq(i,24,by=6),8]),
t(res_mom[seq(i,24,by=6),8]),
t(res_prt[seq(i,24,by=6),8]),
t(res_tsp[seq(i,24,by=6),8]),
t(res_mne[seq(i,24,by=6),8]))
estb<-rbind(estb,t(res_mle[seq(i,24,by=6),9]),
t(res_mom[seq(i,24,by=6),9]),
t(res_prt[seq(i,24,by=6),9]),
t(res_tsp[seq(i,24,by=6),9]),
t(res_mne[seq(i,24,by=6),9]))
}

types<-c("MLE","MOM","PERT","TSP","QNT")
library(xtable)
xtable(cbind(c(types,types),round(rbind(esta[26:30,],estb[26:30,]),3)))


results<-read.table("computed_results.txt",header=T)
names(results)<-c("ns","cas","cbs","biasa","biasb","msea",
"mseb","amean","bmean","avar","bvar")


c1<-seq(1,24,by=6)
c2<-seq(2,24,by=6)
```

```
c3<-seq(3,24,by=6)
c4<-seq(4,24,by=6)
c5<-seq(5,24,by=6)
c6<-seq(6,24,by=6)


res_mle<-results[1:24,]
res_mom<-results[25:48,]
res_prt<-results[49:72,]
res_tsp<-results[73:96,]
res_mne<-results[97:120,]

# track the var of each estimator...
m1<-cbind(res_mom[c1,c(1,10)],res_prt[c1,10],res_tsp[c1,10],
res_mne[c1,10], res_mom[c1,11],res_prt[c1,11],res_tsp[c1,11],res_mne[c1,11])

m2<-cbind(res_mom[c2,c(1,10)],res_prt[c2,10],res_tsp[c2,10],
res_mne[c2,10], res_mom[c2,11],res_prt[c2,11],res_tsp[c2,11],res_mne[c2,11])

m3<-cbind(res_mom[c3,c(1,10)],res_prt[c3,10],res_tsp[c3,10],
res_mne[c3,10], res_mom[c3,11],res_prt[c3,11],res_tsp[c3,11],res_mne[c3,11])

m4<-cbind(res_mom[c4,c(1,10)],res_prt[c4,10],res_tsp[c4,10],
res_mne[c4,10], res_mom[c4,11],res_prt[c4,11],res_tsp[c4,11],res_mne[c4,11])

m5<-cbind(res_mom[c5,c(1,10)],res_prt[c5,10],res_tsp[c5,10],
res_mne[c5,10], res_mom[c5,11],res_prt[c5,11],res_tsp[c5,11],res_mne[c5,11])

m6<-cbind(res_mom[c6,c(1,10)],res_prt[c6,10],res_tsp[c6,10],
res_mne[c6,10], res_mom[c6,11],res_prt[c6,11],res_tsp[c6,11],res_mne[c6,11])



#asymptotically, where should these things go?
Alpha<-c(2,2,.5,.2,.2,1)
Beta<-c(2,6,.5,.5,2,1)
n<-c(25,50,100,500)

# MLEs:
# ahat and bhat should go to alpha and beta as n increases
res_mle[c1,c(1,2,8,3,9)]
# what about the var of these estimates? should approach CRLB

my.hessian<-function(i,j){
hess<-matrix(c(n[j]*trigamma(Alpha[i]+Beta[i])-n[j]*trigamma(Alpha[i]),
n[j]*trigamma(Alpha[i]+Beta[i]),n[j]*trigamma(Alpha[i]+Beta[i]),
n[j]*trigamma(Alpha[i]+Beta[i])-n[j]*trigamma(Beta[i])),nrow=2,byrow=T)
```

```
return(c(hess[1,1],hess[2,2]))
}


h1<-rbind(
1/-my.hessian(1,1),
1/-my.hessian(1,2),
1/-my.hessian(1,3),
1/-my.hessian(1,4))
var1<-cbind(res_mle[c1,c(1,10)],h1[,1],res_mle[c1,11],h1[,2])

h2<-rbind(
1/-my.hessian(2,1),
1/-my.hessian(2,2),
1/-my.hessian(2,3),
1/-my.hessian(2,4))
var2<-cbind(res_mle[c2,c(1,10)],h2[,1],res_mle[c2,11],h2[,2])

h3<-rbind(
1/-my.hessian(3,1),
1/-my.hessian(3,2),
1/-my.hessian(3,3),
1/-my.hessian(3,4))
var3<-cbind(res_mle[c3,c(1,10)],h3[,1],res_mle[c3,11],h3[,2])

h4<-rbind(
1/-my.hessian(4,1),
1/-my.hessian(4,2),
1/-my.hessian(4,3),
1/-my.hessian(4,4))
var4<-cbind(res_mle[c4,c(1,10)],h4[,1],res_mle[c4,11],h4[,2])

h5<-rbind(
1/-my.hessian(5,1),
1/-my.hessian(5,2),
1/-my.hessian(5,3),
1/-my.hessian(5,4))
var5<-cbind(res_mle[c5,c(1,10)],h5[,1],res_mle[c5,11],h5[,2])

h6<-rbind(
1/-my.hessian(6,1),
1/-my.hessian(6,2),
1/-my.hessian(6,3),
1/-my.hessian(6,4))
var6<-cbind(res_mle[c6,c(1,10)],h6[,1],res_mle[c6,11],h6[,2])

# param estimates for each distribution
```

```
p1<-cbind(
res_mle[c1,c(1,8)],
res_mom[c1,8],
res_prt[c1,8],
res_tsp[c1,8],
res_mne[c1,8],
res_mle[c1,9],
res_mom[c1,9],
res_prt[c1,9],
res_tsp[c1,9],
res_mne[c1,9])

p2<-cbind(
res_mle[c2,c(1,8)],
res_mom[c2,8],
res_prt[c2,8],
res_tsp[c2,8],
res_mne[c2,8],
res_mle[c2,9],
res_mom[c2,9],
res_prt[c2,9],
res_tsp[c2,9],
res_mne[c2,9])

p3<-cbind(
res_mle[c3,c(1,8)],
res_mom[c3,8],
res_prt[c3,8],
res_tsp[c3,8],
res_mne[c3,8],
res_mle[c3,9],
res_mom[c3,9],
res_prt[c3,9],
res_tsp[c3,9],
res_mne[c3,9])

p4<-cbind(
res_mle[c4,c(1,8)],
res_mom[c4,8],
res_prt[c4,8],
res_tsp[c4,8],
res_mne[c4,8],
res_mle[c4,9],
res_mom[c4,9],
res_prt[c4,9],
res_tsp[c4,9],
res_mne[c4,9])
```

```
p5<-cbind(
res_mle[c5,c(1,8)],
res_mom[c5,8],
res_prt[c5,8],
res_tsp[c5,8],
res_mne[c5,8],
res_mle[c5,9],
res_mom[c5,9],
res_prt[c5,9],
res_tsp[c5,9],
res_mne[c5,9])

p6<-cbind(
res_mle[c6,c(1,8)],
res_mom[c6,8],
res_prt[c6,8],
res_tsp[c6,8],
res_mne[c6,8],
res_mle[c6,9],
res_mom[c6,9],
res_prt[c6,9],
res_tsp[c6,9],
res_mne[c6,9])
```

```
par(mfrow=c(1,3),ps=18)
# bias c1
plot(ns[c1],log(abs(res_mle[c1,4])),lty=1,lwd=2,col="blue",type="l",
ylim=c(-8.5,0),main="Bias for Beta(2,2)",xlab="Sample Size",ylab="log(|bias|)")
lines(ns[c1],log(abs(res_mle[c1,5])),lty=2,lwd=2,col="blue")
lines(ns[c1],log(abs(res_mom[c1,4])),lty=1,lwd=2,col="green")
lines(ns[c1],log(abs(res_mom[c1,5])),lty=2,lwd=2,col="green")
lines(ns[c1],log(abs(res_mne[c1,4])),lty=1,lwd=2,col="red")
lines(ns[c1],log(abs(res_mne[c1,5])),lty=2,lwd=2,col="red")
lines(ns[c1],log(abs(res_prt[c1,4])),lty=1,lwd=2,col="orange")
lines(ns[c1],log(abs(res_prt[c1,5])),lty=2,lwd=2,col="orange")
lines(ns[c1],log(abs(res_tsp[c1,4])),lty=1,lwd=2,col="purple")
lines(ns[c1],log(abs(res_tsp[c1,5])),lty=2,lwd=2,col="purple")

# MSE c1
plot(ns[c1],log(res_mle[c1,6]),lty=1,lwd=2,col="blue",type="l",
ylim=c(-4.5,0.2),main="MSE for Beta(2,2)",xlab="Sample Size",ylab="log(MSE)")
lines(ns[c1],log(res_mle[c1,7]),lty=2,lwd=2,col="blue")
lines(ns[c1],log(res_mom[c1,6]),lty=1,lwd=2,col="green")
lines(ns[c1],log(res_mom[c1,7]),lty=2,lwd=2,col="green")
lines(ns[c1],log(res_mne[c1,6]),lty=1,lwd=2,col="red")
lines(ns[c1],log(res_mne[c1,7]),lty=2,lwd=2,col="red")
lines(ns[c1],log(res_prt[c1,6]),lty=1,lwd=2,col="orange")
lines(ns[c1],log(res_prt[c1,7]),lty=2,lwd=2,col="orange")
lines(ns[c1],log(res_tsp[c1,6]),lty=1,lwd=2,col="purple")
lines(ns[c1],log(res_tsp[c1,7]),lty=2,lwd=2,col="purple")

# Density c1
plot(tt,dbeta(tt,shape1=Alpha[1],shape2=Beta[1]),lwd=2,type="l",
main="Beta(2,2)",xlab="x",ylab="f(x)",ylim=c(0,1.6))
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mle[c1[i],8],shape2=res_mle[c1[i],9]),
lwd=2,col="blue",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mom[c1[i],8],shape2=res_mom[c1[i],9]),
lwd=2,col="green",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mne[c1[i],8],shape2=res_mne[c1[i],9]),
lwd=2,col="red",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_prt[c1[i],8],shape2=res_prt[c1[i],9]),
lwd=2,col="orange",lty=i+2)}
for(i in 1:4){
```

```
lines(tt,dbeta(tt,shape1=res_tsp[c1[i],8],shape2=res_tsp[c1[i],9]),
lwd=2,col="purple",lty=i+2)}
lines(tt,dbeta(tt,shape1=Alpha[1],shape2=Beta[1]),lwd=2)


# bias c2
plot(ns[c2],log(abs(res_mle[c2,4])),lty=1,lwd=2,col="blue",type="l",
ylim=c(-6,1),main="Bias for Beta(2,6)",xlab="Sample Size",ylab="log(|bias|)")
lines(ns[c2],log(abs(res_mle[c2,5])),lty=2,lwd=2,col="blue")
lines(ns[c2],log(abs(res_mom[c2,4])),lty=1,lwd=2,col="green")
lines(ns[c2],log(abs(res_mom[c2,5])),lty=2,lwd=2,col="green")
lines(ns[c2],log(abs(res_mne[c2,4])),lty=1,lwd=2,col="red")
lines(ns[c2],log(abs(res_mne[c2,5])),lty=2,lwd=2,col="red")
lines(ns[c2],log(abs(res_prt[c2,4])),lty=1,lwd=2,col="orange")
lines(ns[c2],log(abs(res_prt[c2,5])),lty=2,lwd=2,col="orange")
lines(ns[c2],log(abs(res_tsp[c2,4])),lty=1,lwd=2,col="purple")
lines(ns[c2],log(abs(res_tsp[c2,5])),lty=2,lwd=2,col="purple")

# MSE c2
plot(ns[c2],log(res_mle[c2,6]),lty=1,lwd=2,col="blue",type="l",
ylim=c(-5,2),main="MSE for Beta(2,6)",xlab="Sample Size",ylab="log(MSE)")
lines(ns[c2],log(res_mle[c2,7]),lty=2,lwd=2,col="blue")
lines(ns[c2],log(res_mom[c2,6]),lty=1,lwd=2,col="green")
lines(ns[c2],log(res_mom[c2,7]),lty=2,lwd=2,col="green")
lines(ns[c2],log(res_mne[c2,6]),lty=1,lwd=2,col="red")
lines(ns[c2],log(res_mne[c2,7]),lty=2,lwd=2,col="red")
lines(ns[c2],log(res_prt[c2,6]),lty=1,lwd=2,col="orange")
lines(ns[c2],log(res_prt[c2,7]),lty=2,lwd=2,col="orange")
lines(ns[c2],log(res_tsp[c2,6]),lty=1,lwd=2,col="purple")
lines(ns[c2],log(res_tsp[c2,7]),lty=2,lwd=2,col="purple")

# Density c2
plot(tt,dbeta(tt,shape1=Alpha[2],shape2=Beta[2]),lwd=2,type="l",
main="Beta(2,6)",xlab="x",ylab="f(x)",ylim=c(0,3))
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mle[c2[i],8],shape2=res_mle[c2[i],9]),
lwd=2,col="blue",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mom[c2[i],8],shape2=res_mom[c2[i],9]),
lwd=2,col="green",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mne[c2[i],8],shape2=res_mne[c2[i],9]),
lwd=2,col="red",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_prt[c2[i],8],shape2=res_prt[c2[i],9]),
lwd=2,col="orange",lty=i+2)}
for(i in 1:4){
```

```
lines(tt,dbeta(tt,shape1=res_tsp[c2[i],8],shape2=res_tsp[c2[i],9]),
lwd=2,col="purple",lty=i+2)}
lines(tt,dbeta(tt,shape1=Alpha[2],shape2=Beta[2]),lwd=2)


# bias c3
plot(ns[c3],log(abs(res_mle[c3,4])),lty=1,lwd=2,col="blue",type="l",
ylim=c(-14,3),main="Bias for Beta(0.5,0.5)",xlab="Sample Size",ylab="log(|bias|)")
lines(ns[c3],log(abs(res_mle[c3,5])),lty=2,lwd=2,col="blue")
lines(ns[c3],log(abs(res_mom[c3,4])),lty=1,lwd=2,col="green")
lines(ns[c3],log(abs(res_mom[c3,5])),lty=2,lwd=2,col="green")
lines(ns[c3],log(abs(res_mne[c3,4])),lty=1,lwd=2,col="red")
lines(ns[c3],log(abs(res_mne[c3,5])),lty=2,lwd=2,col="red")
lines(ns[c3],log(abs(res_prt[c3,4])),lty=1,lwd=2,col="orange")
lines(ns[c3],log(abs(res_prt[c3,5])),lty=2,lwd=2,col="orange")
lines(ns[c3],log(abs(res_tsp[c3,4])),lty=1,lwd=2,col="purple")
lines(ns[c3],log(abs(res_tsp[c3,5])),lty=2,lwd=2,col="purple")

# MSE c3
plot(ns[c3],log(res_mle[c3,6]),lty=1,lwd=2,col="blue",type="l",
ylim=c(-7,6),main="MSE for Beta(0.5,0.5)",xlab="Sample Size",ylab="log(MSE)")
lines(ns[c3],log(res_mle[c3,7]),lty=2,lwd=2,col="blue")
lines(ns[c3],log(res_mom[c3,6]),lty=1,lwd=2,col="green")
lines(ns[c3],log(res_mom[c3,7]),lty=2,lwd=2,col="green")
lines(ns[c3],log(res_mne[c3,6]),lty=1,lwd=2,col="red")
lines(ns[c3],log(res_mne[c3,7]),lty=2,lwd=2,col="red")
lines(ns[c3],log(res_prt[c3,6]),lty=1,lwd=2,col="orange")
lines(ns[c3],log(res_prt[c3,7]),lty=2,lwd=2,col="orange")
lines(ns[c3],log(res_tsp[c3,6]),lty=1,lwd=2,col="purple")
lines(ns[c3],log(res_tsp[c3,7]),lty=2,lwd=2,col="purple")

# Density c3
plot(tt,dbeta(tt,shape1=Alpha[3],shape2=Beta[3]),lwd=2,type="l",
main="Beta(0.5,0.5)",xlab="x",ylab="f(x)",ylim=c(0,3))
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mle[c3[i],8],shape2=res_mle[c3[i],9]),
lwd=2,col="blue",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mom[c3[i],8],shape2=res_mom[c3[i],9]),
lwd=2,col="green",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mne[c3[i],8],shape2=res_mne[c3[i],9]),
lwd=2,col="red",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_prt[c3[i],8],shape2=res_prt[c3[i],9]),
lwd=2,col="orange",lty=i+2)}
for(i in 1:4){
```

```
lines(tt,dbeta(tt,shape1=res_tsp[c3[i],8],shape2=res_tsp[c3[i],9]),
lwd=2,col="purple",lty=i+2)}
lines(tt,dbeta(tt,shape1=Alpha[3],shape2=Beta[3]),lwd=2)


# bias c4
plot(ns[c4],log(abs(res_mle[c4,4])),lty=1,lwd=2,col="blue",type="l",
ylim=c(-11,4.5),main="Bias for Beta(0.2,0.5)",xlab="Sample Size",ylab="log(|bias|)")
lines(ns[c4],log(abs(res_mle[c4,5])),lty=2,lwd=2,col="blue")
lines(ns[c4],log(abs(res_mom[c4,4])),lty=1,lwd=2,col="green")
lines(ns[c4],log(abs(res_mom[c4,5])),lty=2,lwd=2,col="green")
lines(ns[c4],log(abs(res_mne[c4,4])),lty=1,lwd=2,col="red")
lines(ns[c4],log(abs(res_mne[c4,5])),lty=2,lwd=2,col="red")
lines(ns[c4],log(abs(res_prt[c4,4])),lty=1,lwd=2,col="orange")
lines(ns[c4],log(abs(res_prt[c4,5])),lty=2,lwd=2,col="orange")
lines(ns[c4],log(abs(res_tsp[c4,4])),lty=1,lwd=2,col="purple")
lines(ns[c4],log(abs(res_tsp[c4,5])),lty=2,lwd=2,col="purple")

# MSE c4
plot(ns[c4],log(res_mle[c4,6]),lty=1,lwd=2,col="blue",type="l",
ylim=c(-10,12),main="MSE for Beta(0.2,0.5)",xlab="Sample Size",ylab="log(MSE)")
lines(ns[c4],log(res_mle[c4,7]),lty=2,lwd=2,col="blue")
lines(ns[c4],log(res_mom[c4,6]),lty=1,lwd=2,col="green")
lines(ns[c4],log(res_mom[c4,7]),lty=2,lwd=2,col="green")
lines(ns[c4],log(res_mne[c4,6]),lty=1,lwd=2,col="red")
lines(ns[c4],log(res_mne[c4,7]),lty=2,lwd=2,col="red")
lines(ns[c4],log(res_prt[c4,6]),lty=1,lwd=2,col="orange")
lines(ns[c4],log(res_prt[c4,7]),lty=2,lwd=2,col="orange")
lines(ns[c4],log(res_tsp[c4,6]),lty=1,lwd=2,col="purple")
lines(ns[c4],log(res_tsp[c4,7]),lty=2,lwd=2,col="purple")

# Density c4
plot(tt,dbeta(tt,shape1=Alpha[4],shape2=Beta[4]),lwd=2,type="l",
main="Beta(0.2,0.5)",xlab="x",ylab="Density",ylim=c(0,3))
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mle[c4[i],8],shape2=res_mle[c4[i],9]),
lwd=2,col="blue",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mom[c4[i],8],shape2=res_mom[c4[i],9]),
lwd=2,col="green",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mne[c4[i],8],shape2=res_mne[c4[i],9]),
lwd=2,col="red",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_prt[c4[i],8],shape2=res_prt[c4[i],9]),
lwd=2,col="orange",lty=i+2)}
for(i in 1:4){
```

```
lines(tt,dbeta(tt,shape1=res_tsp[c4[i],8],shape2=res_tsp[c4[i],9]),
lwd=2,col="purple",lty=i+2)}
lines(tt,dbeta(tt,shape1=Alpha[4],shape2=Beta[4]),lwd=2)


# bias c5
plot(ns[c5],log(abs(res_mle[c5,4])),lty=1,lwd=2,col="blue",type="l",
ylim=c(-7,6),main="Bias for Beta(0.2,2)",xlab="Sample Size",ylab="log(|bias|)")
lines(ns[c5],log(abs(res_mle[c5,5])),lty=2,lwd=2,col="blue")
lines(ns[c5],log(abs(res_mom[c5,4])),lty=1,lwd=2,col="green")
lines(ns[c5],log(abs(res_mom[c5,5])),lty=2,lwd=2,col="green")
lines(ns[c5],log(abs(res_mne[c5,4])),lty=1,lwd=2,col="red")
lines(ns[c5],log(abs(res_mne[c5,5])),lty=2,lwd=2,col="red")
lines(ns[c5],log(abs(res_prt[c5,4])),lty=1,lwd=2,col="orange")
lines(ns[c5],log(abs(res_prt[c5,5])),lty=2,lwd=2,col="orange")
lines(ns[c5],log(abs(res_tsp[c5,4])),lty=1,lwd=2,col="purple")
lines(ns[c5],log(abs(res_tsp[c5,5])),lty=2,lwd=2,col="purple")

# MSE c5
plot(ns[c5],log(res_mle[c5,6]),lty=1,lwd=2,col="blue",type="l",
ylim=c(-10,13),main="MSE for Beta(0.2,2)",xlab="Sample Size",ylab="log(MSE)")
lines(ns[c5],log(res_mle[c5,7]),lty=2,lwd=2,col="blue")
lines(ns[c5],log(res_mom[c5,6]),lty=1,lwd=2,col="green")
lines(ns[c5],log(res_mom[c5,7]),lty=2,lwd=2,col="green")
lines(ns[c5],log(res_mne[c5,6]),lty=1,lwd=2,col="red")
lines(ns[c5],log(res_mne[c5,7]),lty=2,lwd=2,col="red")
lines(ns[c5],log(res_prt[c5,6]),lty=1,lwd=2,col="orange")
lines(ns[c5],log(res_prt[c5,7]),lty=2,lwd=2,col="orange")
lines(ns[c5],log(res_tsp[c5,6]),lty=1,lwd=2,col="purple")
lines(ns[c5],log(res_tsp[c5,7]),lty=2,lwd=2,col="purple")

# Density c5
plot(tt,dbeta(tt,shape1=Alpha[5],shape2=Beta[5]),lwd=2,type="l",
main="Beta(0.2,2)",xlab="x",ylab="f(x)",ylim=c(0,3.8))
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mle[c5[i],8],shape2=res_mle[c5[i],9]),
lwd=2,col="blue",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mom[c5[i],8],shape2=res_mom[c5[i],9]),
lwd=2,col="green",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mne[c5[i],8],shape2=res_mne[c5[i],9]),
lwd=2,col="red",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_prt[c5[i],8],shape2=res_prt[c5[i],9]),
lwd=2,col="orange",lty=i+2)}
for(i in 1:4){
```

```
lines(tt,dbeta(tt,shape1=res_tsp[c5[i],8],shape2=res_tsp[c5[i],9]),
lwd=2,col="purple",lty=i+2)}
lines(tt,dbeta(tt,shape1=Alpha[5],shape2=Beta[5]),lwd=2)


# bias c6
plot(ns[c6],log(abs(res_mle[c6,4])),lty=1,lwd=2,col="blue",type="l",
ylim=c(-7,0),main="Bias for Beta(1,1)",xlab="Sample Size",ylab="log(|bias|)")
lines(ns[c6],log(abs(res_mle[c6,5])),lty=2,lwd=2,col="blue")
lines(ns[c6],log(abs(res_mom[c6,4])),lty=1,lwd=2,col="green")
lines(ns[c6],log(abs(res_mom[c6,5])),lty=2,lwd=2,col="green")
lines(ns[c6],log(abs(res_mne[c6,4])),lty=1,lwd=2,col="red")
lines(ns[c6],log(abs(res_mne[c6,5])),lty=2,lwd=2,col="red")
lines(ns[c6],log(abs(res_prt[c6,4])),lty=1,lwd=2,col="orange")
lines(ns[c6],log(abs(res_prt[c6,5])),lty=2,lwd=2,col="orange")
lines(ns[c6],log(abs(res_tsp[c6,4])),lty=1,lwd=2,col="purple")
lines(ns[c6],log(abs(res_tsp[c6,5])),lty=2,lwd=2,col="purple")

# MSE c6
plot(ns[c6],log(res_mle[c6,6]),lty=1,lwd=2,col="blue",type="l",
ylim=c(-6,0),main="MSE for Beta(1,1)",xlab="Sample Size",ylab="log(MSE)")
lines(ns[c6],log(res_mle[c6,7]),lty=2,lwd=2,col="blue")
lines(ns[c6],log(res_mom[c6,6]),lty=1,lwd=2,col="green")
lines(ns[c6],log(res_mom[c6,7]),lty=2,lwd=2,col="green")
lines(ns[c6],log(res_mne[c6,6]),lty=1,lwd=2,col="red")
lines(ns[c6],log(res_mne[c6,7]),lty=2,lwd=2,col="red")
lines(ns[c6],log(res_prt[c6,6]),lty=1,lwd=2,col="orange")
lines(ns[c6],log(res_prt[c6,7]),lty=2,lwd=2,col="orange")
lines(ns[c6],log(res_tsp[c6,6]),lty=1,lwd=2,col="purple")
lines(ns[c6],log(res_tsp[c6,7]),lty=2,lwd=2,col="purple")

# Density c6
plot(tt,dbeta(tt,shape1=Alpha[6],shape2=Beta[6]),lwd=2,type="l",
main="Beta(1,1)",xlab="x",ylab="f(x)",ylim=c(0,1.5))
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mle[c6[i],8],shape2=res_mle[c6[i],9]),
lwd=2,col="blue",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mom[c6[i],8],shape2=res_mom[c6[i],9]),
lwd=2,col="green",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_mne[c6[i],8],shape2=res_mne[c6[i],9]),
lwd=2,col="red",lty=i+2)}
for(i in 1:4){
lines(tt,dbeta(tt,shape1=res_prt[c6[i],8],shape2=res_prt[c6[i],9]),
lwd=2,col="orange",lty=i+2)}
for(i in 1:4){
```

```
lines(tt,dbeta(tt,shape1=res_tsp[c6[i],8],shape2=res_tsp[c6[i],9]),
lwd=2,col="purple",lty=i+2)}
lines(tt,dbeta(tt,shape1=Alpha[6],shape2=Beta[6]),lwd=2)

# legends
plot(-10,-10,xlab="",ylab="",xlim=c(0,10),ylim=c(0,10),axes=FALSE)
text(5,10,labels=c("Legend for Estimation Methods"))
legend(4,9.5,legend=c("True","MLE","MOM","QNT","PERT","TSP"),
col=c("black","blue","green","red","orange","purple"),lty=1,lwd=2)
text(5,5.5,labels=c("Legend for Bias and MSE Parameter Estimates"))
legend(4.3,5,legend=c(expression(alpha),expression(beta)),lty=c(1,2),lwd=2)
text(5,2.7,labels=c("Legend for Density Plot Sample Sizes"))
legend(4,2.2,legend=c("n=25","n=50","n=100","n=500"),lty=c(3:6),lwd=2)
```

```
# Use this function to analyze data: #
application.analysis<-function(x,starta,startb){
betdat<-x

#### MLE: Newton-Raphson ####
nrand<-length(betdat)
i<-2
alpha<-rep(starta,2)
beta<-rep(startb,2)
tol<-10^-3
lim<-10^-4
lim2<--5
eps<-1
maxiter<-100
while(tol<eps & i<maxiter){
# create g matrix - 1st derivs

g1<- digamma(alpha[i-1]) - digamma(alpha[i-1]+beta[i-1]) - sum(log(betdat))/nrand

g2<- digamma(beta[i-1]) - digamma(alpha[i-1]+beta[i-1]) - sum(log(1-betdat))/nrand

g<- c(g1,g2)

if(g1<lim2 | g2<lim2){
num<-i
i<-maxiter
alpha[i]<-alpha[num-1]
beta[i]<-beta[num-1]
}
else{

# create g' matrix - matrix of 2nd derivs

g1a<- trigamma(alpha[i-1]) - trigamma(alpha[i-1] + beta[i-1])

g1b<- g2a<- -trigamma(alpha[i-1] + beta[i-1])

g2b<- trigamma(beta[i-1]) - trigamma(alpha[i-1] + beta[i-1])

gp<- matrix(c(g1a,g1b,g2a,g2b),ncol=2,byrow=T)

# compute next value
```

```
temp<- c(alpha[i-1],beta[i-1]) - solve(gp)%*%g

alpha[i]<- temp[1]

beta[i]<- temp[2]

# see if we've reached our tolerance

eps<- max(abs((alpha[i-1]-alpha[i])/alpha[i-1]),abs((beta[i-1]-beta[i])/beta[i-1]))

# increment the loop!

if(abs(g1a)<lim | abs(g1b)<lim | abs(g2b)<lim){
num<-i
i<-maxiter
alpha[i]<-alpha[num-1]
beta[i]<-beta[num-1]
}
}

i<- i + 1
}
amle<-alpha[i-1]
bmle<-beta[i-1]

#### MOM ####
xbar<-mean(betdat)
varx<-var(betdat)
amom<-xbar*((xbar*(1-xbar)/varx)-1)
bmom<-(1-xbar)*((xbar*(1-xbar)/varx)-1)

#### modified MOM:  PERT Approx. ####
y<-density(betdat)$y
x<-density(betdat)$x
top<-which(density(betdat)$y==max(density(betdat)$y))
mo<-density(betdat)$x[top]

# to improve est of var, use var of data
sig2x<-varx

if(mo>=0.13 & mo<=0.87){
mux<-(4*mo+1)/6
# sig2x<-(1/6)^2
}
if(mo<0.13){
mux<-2/(2+(1/mo))
# sig2x<-(mo^2*(1-mo))/(1+mo)
```

```
}
if(mo>0.87){
mux<-1/(3-2*mo)
# sig2x<-(mo*(1-mo)^2)/(2-mo)
}

if(mux*(1-mux)<sig2x){
if(mo>=0.13 & mo<=0.87){
sig2x<-(1/6)^2
}
if(mo<0.13){
sig2x<-(mo^2*(1-mo))/(1+mo)
}
if(mo>0.87){
sig2x<-(mo*(1-mo)^2)/(2-mo)
}
aprt<-mux*((mux*(1-mux)/sig2x)-1)
bprt<-(1-mux)*((mux*(1-mux)/sig2x)-1)
}
if(mux*(1-mux)>=sig2x){
aprt<-mux*((mux*(1-mux)/sig2x)-1)
bprt<-(1-mux)*((mux*(1-mux)/sig2x)-1)
}

### modified two sided power / triangular: tsp ###
s<-length(betdat)
myind<-order(betdat)

m.fun<-function(r){
prod1<-1
prod2<-1
for(i in 1:(r-1)){
prod1<-prod1*betdat[myind[i]]/betdat[myind[r]]
}
for(i in (r+1):s){
prod2<-prod2*(1-betdat[myind[i]])/(1-betdat[myind[r]])
}
M.stat<-prod1*prod2
return(M.stat)
}

test<-matrix(0,s-1)
for(i in 2:(s-1)){
test[i]<-m.fun(i)
}

rhat<-which(test==max(test))
```

```
mhat<-betdat[rhat]
nhat<--s/log(m.fun(rhat))

tmux<-((nhat-1)*mhat+1)/(nhat+1)
tsig2x<-(nhat-2*(nhat-1)*mhat*(1-mhat))/((nhat+2)*(nhat+1)^2)

atsp<-tmux*((tmux*(1-tmux)/tsig2x)-1)
btsp<-(1-tmux)*((tmux*(1-tmux)/tsig2x)-1)

### modified quantile est:  mne ###
q1<-quantile(betdat,.25)
q3<-quantile(betdat,.75)

loa<-ifelse(amom-1<0,0,amom-1)
hia<-amom+1
lob<-ifelse(bmom-1<0,0,bmom-1)
hib<-bmom+1

acand<-seq(loa,hia,length=200)
bcand<-seq(lob,hib,length=200)

q1est<-qbeta(.25,shape1=acand,shape2=bcand)
q3est<-qbeta(.75,shape1=acand,shape2=bcand)

my.crit<-(q1-q1est)^2+(q3-q3est)^2
my.keep<-which(my.crit==min(my.crit))

amne<-acand[my.keep]
bmne<-bcand[my.keep]

return(cbind(rbind(amle,amom,aprt,atsp[1],amne),
rbind(bmle,bmom,bprt,btsp[1],bmne)) )
}




# for MLB Batting Averages #
batting<-read.table("batting.csv",header=T,sep=",")
batavg<-batting$BA

application.analysis(batavg,100,250)

# Parameter Estimates #
library(xtable)
xtable(cbind(rbind(amle,amom,aprt,atsp[1],amne),
rbind(bmle,bmom,bprt,btsp[1],bmne)),digits=3)
```

```
par(mfrow=c(1,1))
plot(density(batavg),lwd=2,)xlim=c(0,.8),xlab="",main="")
lines(xx<-seq(0,1,length=1000),dbeta(xx,shape1=amle,shape2=bmle),
lwd=4,col="blue",,ylim=c(0,20))
lines(xx,dbeta(xx,shape1=amom,shape2=bmom),lwd=2,col="green")
lines(xx,dbeta(xx,shape1=aprt,shape2=bprt),lwd=2,col="orange")
lines(xx,dbeta(xx,shape1=amne,shape2=bmne),lwd=2,col="red")
lines(xx,dbeta(xx,shape1=atsp[1],shape2=btsp[1]),lwd=2,col="purple")
legend(.45,16,legend=c("Data","MLE","MOM","PERT","QNT","TSP"),
col=c("black","blue","green","orange","red","purple"),lwd=2)



# Probability of a MLB player falling below the Mendoza Line (.200)
MLE1<-pbeta(.2,shape1=amle,shape2=bmle)
MOM1<-pbeta(.2,shape1=amom,shape2=bmom)
PERT1<-pbeta(.2,shape1=aprt,shape2=bprt)
QNT1<-pbeta(.2,shape1=amne,shape2=bmne)
TSP1<-pbeta(.2,shape1=atsp[1],shape2=btsp[1])
xtable(rbind(MLE1,MOM1,PERT1,QNT1,TSP1),digits=4,display=c("g","g"))

MLE2<-1-pbeta(.4,shape1=amle,shape2=bmle)
MOM2<-1-pbeta(.4,shape1=amom,shape2=bmom)
PERT2<-1-pbeta(.4,shape1=aprt,shape2=bprt)
QNT2<-1-pbeta(.4,shape1=amne,shape2=bmne)
TSP2<-1-pbeta(.4,shape1=atsp[1],shape2=btsp[1])
xtable(rbind(MLE2,MOM2,PERT2,QNT2,TSP2),digits=4,display=c("g","g"))

# for exposure data #
dose<-read.csv("rems_out.dat",header=T)

year<-dose[,1]
num.monitored<-dose[,2]
num.with.msr.dose<-dose[,3]
num.with.msr.exposure<-dose[,6]

lt100<-dose[,7]
m100_250<-dose[,8]
m250_500<-dose[,9]
m500_750<-dose[,10]
m750_1000<-dose[,11]
m1000_2000<-dose[,12]
m2000_3000<-dose[,13]
m3000_4000<-dose[,14]
m4000_5000<-dose[,15]
m5000_6000<-dose[,16]
```

```
m6000_7000<-dose[,17]
m7000_8000<-dose[,18]
m8000_9000<-dose[,19]
m9000_10000<-dose[,20]
gt10000<-dose[,21]

names(dose)

exposed<-num.with.msr.exposure
g1<-lt100
g2<-m100_250
g3<-m250_500
g4<-m500_750
g5<-m750_1000
g6<-m1000_2000
g7<-m2000_3000
g8<-m3000_4000
g9<-m4000_5000
g10<-m5000_6000
g11<-m6000_7000
g12<-m7000_8000
g13<-m8000_9000
g14<-m9000_10000
g15<-gt10000

dat<-cbind(exposed,g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,g11,g12,g13,g14,g15)
dat
big<-apply(dat[,7:16],1,sum)
big
newdat<-cbind(dat[,1:6],big)
newdat
propdat<-newdat/exposed
zapsmall(propdat)

#what is the true proportion of workers exposed to each level of REM?
g1est<-application.analysis(propdat[,2],2,10)
g2est<-application.analysis(propdat[,3],1,30)
g3est<-application.analysis(propdat[,4],1,30)
g4est<-application.analysis(propdat[,5],1,99)
g4est
g5est<-application.analysis(propdat[,6],.5,70)
g5est
g6est<-application.analysis(propdat[,7],.15,10)
g6est

#write.table(rbind(g1est,g2est,g3est,g4est,g5est,g6est),
file="exposure_estimates.txt",row.names=FALSE,col.names=FALSE)
```

```
cols<-c("blue","green","orange","purple","red")
plot(density(propdat[,7]),xlim=c(0,.015),main=">1000 millirem",lwd=2)
xx<-seq(0,.015,length=1000)
for(i in 1:5){
lines(xx,dbeta(xx,shape1=g6est[i,1],shape2=g6est[i,2]),col=cols[i],lwd=2)
}
legend(.01,600,legend=c("Data","MLE","MOM","PERT","QNT","TSP"),
col=c("black","blue","green","orange","red","purple"),lwd=2)

plot(density(propdat[,6]),xlim=c(0,.01),main="750-1000 millirem",lwd=2)
xx<-seq(0,.01,length=1000)
for(i in 1:5){
lines(xx,dbeta(xx,shape1=g5est[i,1],shape2=g5est[i,2]),col=cols[i],lwd=2)
}
legend(.006,700,legend=c("Data","MLE","MOM","PERT","QNT","TSP"),
col=c("black","blue","green","orange","red","purple"),lwd=2)
plot(density(propdat[,5]),xlim=c(0,.02),main="500-750 millirem",lwd=2)
xx<-seq(0,.02,length=1000)
for(i in 1:5){
lines(xx,dbeta(xx,shape1=g4est[i,1],shape2=g4est[i,2]),col=cols[i],lwd=2)
}
legend(.0125,275,legend=c("Data","MLE","MOM","PERT","QNT","TSP"),
col=c("black","blue","green","orange","red","purple"),lwd=2)

plot(density(propdat[,4]),xlim=c(0,.04),main="250-500 millirem",lwd=2)
xx<-seq(0,.04,length=1000)
for(i in 1:5){
lines(xx,dbeta(xx,shape1=g3est[i,1],shape2=g3est[i,2]),col=cols[i],lwd=2)
}
legend(.025,125,legend=c("Data","MLE","MOM","PERT","QNT","TSP"),
col=c("black","blue","green","orange","red","purple"),lwd=2)

plot(density(propdat[,3]),xlim=c(0,.1),main="100-250 millirem",lwd=2)
xx<-seq(0,.1,length=1000)
for(i in 1:5){
lines(xx,dbeta(xx,shape1=g2est[i,1],shape2=g2est[i,2]),col=cols[i],lwd=2)
}
legend(.07,80,legend=c("Data","MLE","MOM","PERT","QNT","TSP"),
col=c("black","blue","green","orange","red","purple"),lwd=2)

plot(density(propdat[,2]),xlim=c(0,.7),main="<100 millirem",lwd=2)
xx<-seq(0,.7,length=1000)
for(i in 1:5){
lines(xx,dbeta(xx,shape1=g1est[i,1],shape2=g1est[i,2]),col=cols[i],lwd=2)
}
legend(.5,8,legend=c("Data","MLE","MOM","PERT","QNT","TSP"),
```

```
col=c("black","blue","green","orange","red","purple"),lwd=2)

g1mn<-g1est[,1]/(g1est[,1]+g1est[,2])
g2mn<-g2est[,1]/(g2est[,1]+g2est[,2])
g3mn<-g3est[,1]/(g3est[,1]+g3est[,2])
g4mn<-g4est[,1]/(g4est[,1]+g4est[,2])
g5mn<-g5est[,1]/(g5est[,1]+g5est[,2])
g6mn<-g6est[,1]/(g6est[,1]+g6est[,2])

exposuremeans<-rbind(apply(propdat[,2:7],2,mean),
cbind(g1mn,g2mn,g3mn,g4mn,g5mn,g6mn))

apply(exposuremeans,1,sum)

xtable(cbind(exposuremeans,apply(exposuremeans,1,sum)),digits=4)
```