# STAT 512 sp 2018 Lec 11 R Supplement

Karl Gregory

4/18/2018

## MoMs and MLEs for the Gamma, Beta, and Weibull distributions

Thoughout this document: For a random sample $X_1, \ldots, X_n$, let $m_1' = n^{-1}\sum_{i=1}^n X_i$ and $m_2' = n^{-1}\sum_{i=1}^n X_i^2$ denote the first and second sample moments.

## Gamma distribution

If $X_1, \ldots, X_n$ is a random sample from the Gamma$(\alpha, \beta)$ distribution, the MoMs estimators of $\alpha$ and $\beta$ based on $X_1, \ldots, X_n$ are

$$\bar{\alpha} = \frac{(m_1')^2}{m_2' - (m_1')^2} \quad \text{and} \quad \bar{\beta} = \frac{m_2' - (m_1')^2}{m_1'}.$$

The MLEs for $\alpha$ and $\beta$ are

$$(\hat{\alpha}, \hat{\beta}) = \text{argmax}_{\alpha,\beta} \prod_{i=1}^n \frac{1}{\Gamma(\alpha)\beta^\alpha} X_i^{\alpha-1} e^{-X_i/\beta} \mathbf{1}(X_i > 0).$$

We cannot get a closed-form solution for $(\hat{\alpha}, \hat{\beta})$; if we try to take the derivative of the log-likelihood, we find that the $\alpha$ parameter is stuck inside a gamma function. We can nevertheless compute the MLEs in R using a numerical optimizer.

In the following, we compute the MoMs estimators $(\bar{\alpha}, \bar{\beta})$ and the MLEs $(\hat{\alpha}, \hat{\beta})$ for a data set and compare them how well the Gamma distribution with the estimated parameters fits the data.

### Birth data set

The following R code reads in a data set containing, for each of 7 days, the lengths of time in hours spent by women in the delivery suite while giving birth (without a ceasarian section) at John Radcliffe Hospital in Oxford, England. The data are taken from Davison (2003).

```
# Read in the data for each day
day1 <- c(2.1,3.4,4.25,5.6,6.4,7.3,8.5,8.75,8.9,9.5,9.75,10,10.4,10.4,16,19)
day2 <- c(4,4.1,5,5.5,5.7,6.5,7.25,7.3,7.5,8.2,8.5,9.75,11,11.2,15,16.5)
day3 <- c(2.6,3.6,3.6,6.4,6.8,7.5,7.5,8.25,8.5,10.4,10.75,14.25,14.5)
day4 <- c(1.5,4.7,4.7,7.2,7.25,8.1,8.5,9.2,9.5,10.7,11.5)
day5 <- c(2.5,2.5,3.4,4.2,5.9,6.25,7.3,7.5,7.8,8.3,8.3,10.25,12.9,14.3)
day6 <- c(4,4,5.25,6.1,6.5,6.9,7,8.45,9.25,10.1,10.2,12.75,14.6)
day7 <- c(2,2.7,2.75,3.4,4.2,4.3,4.9,6.25,7,9,9.25,10.7)

# Combine the data together into X
X <- c(day1,day2,day3,day4,day5,day6,day7)
n <- length(X)
```

To compute the MLEs in R, we will use the R function `optim()`, which minimizes a function using numerical methods. We feed the `optim()` function the negative of the log-likelihood function in order to get the MLEs.

```r
# Define the negative log-likelihood function
negll <- function(theta,X)
{
  alpha <- theta[1]
  beta <- theta[2]
  negll <- - sum( dgamma(X, shape = alpha, scale = beta, log = TRUE) )
  return(negll)
}

# Use the optim() function to find the MLEs of alpha and beta
mle <- optim(par=c(1.4,2),fn = negll, X = X)$par
alpha.hat <- mle[1]
beta.hat <- mle[2]

alpha.hat
```

```
## [1] 4.388365
```

```r
beta.hat
```

```
## [1] 1.75994
```

```r
# Compute the MoMs estimators of alpha and beta
m1 <- mean(X)
m2 <- mean(X^2)
alpha.bar <- m1^2 / (m2 - m1^2)
beta.bar <- (m2 - m1^2) / m1

alpha.bar
```

```
## [1] 4.734915
```

```r
beta.bar
```

```
## [1] 1.631108
```

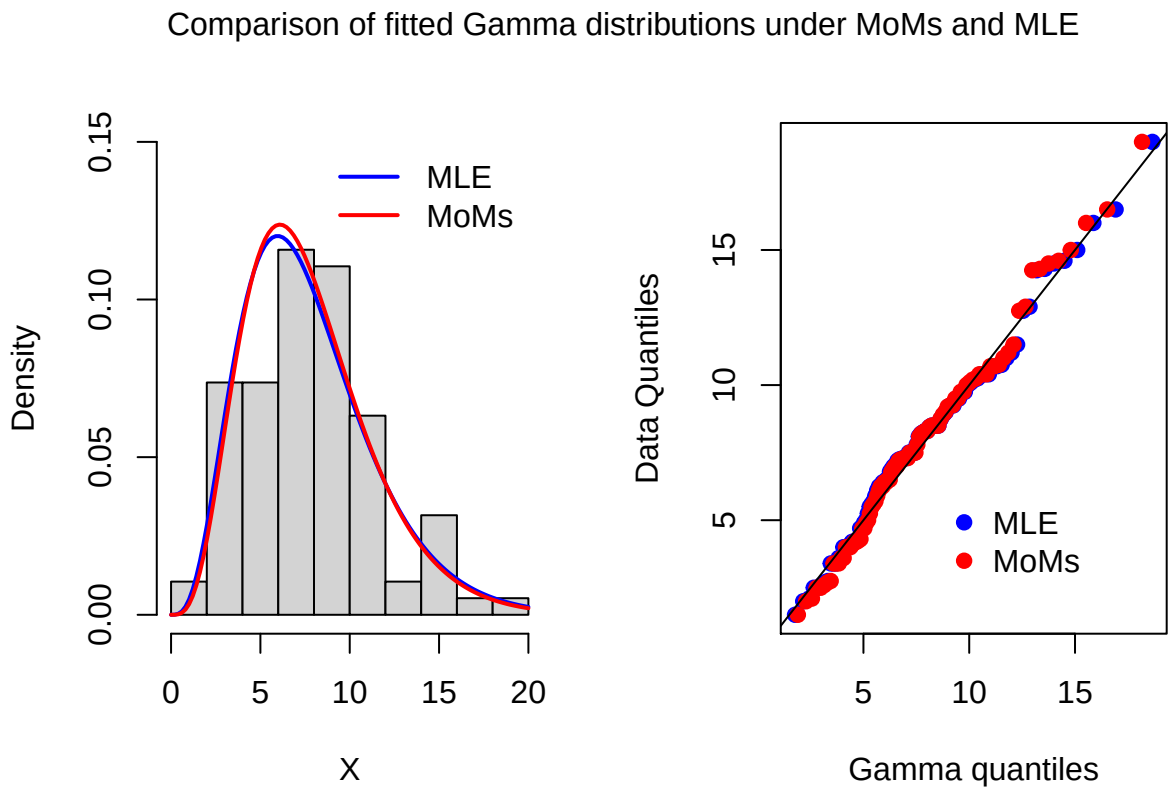
```r
# Specify that the next two plots should be side-by-side
par(mfrow=c(1,2))

# Plot histogram of the data and overlay gamma densities based on the MoMs and the MLEs
x.seq <- seq(0,20,length=100)
hist(X,freq=FALSE,ylim=c(0,0.15),main="")
lines(dgamma(x.seq,shape=alpha.hat,scale=beta.hat)~x.seq,col="blue",lwd=2)
lines(dgamma(x.seq,shape=alpha.bar,scale=beta.bar)~x.seq,col="red",lwd=2)
x.pos <- grconvertX(.5,from="nfc",to="user")
y.pos <- grconvertY(.8,from="nfc",to="user")
legend(x = x.pos, y = y.pos, legend = c("MLE","MoMs") ,
       col = c("blue","red"), lty=c(1,1), lwd=c(2,2), bty="n")

# Plot data quantiles against the corresponding quantiles of the fitted gamma dists
gamma.quantiles.mle <- qgamma((1:n)/(n+1),shape=alpha.hat,scale=beta.hat)
gamma.quantiles.mom <- qgamma((1:n)/(n+1),shape=alpha.bar,scale=beta.bar)
plot(sort(X)~gamma.quantiles.mle,xlab=paste("Gamma quantiles",sep=""),
     ylab="Data Quantiles",pch=19,col="blue")
points(sort(X)~gamma.quantiles.mom,pch=19,col="red")
```

```r
abline(0,1)
x.pos <- grconvertX(.5,from="nfc",to="user")
y.pos <- grconvertY(.4,from="nfc",to="user")
legend(x = x.pos, y = y.pos, legend = c("MLE","MoMs") ,
       col = c("blue","red"), pch=c(19,19), bty="n")

# Add title to plot
mtext(side=3,outer=TRUE, "Comparison of fitted Gamma distributions under MoMs and MLE",
      line=-2)
```




## Beta distribution

If $X_1, \ldots, X_n$ is a random sample from the Beta$(\alpha, \beta)$ distribution, the MoMs estimators of $\alpha$ and $\beta$ based on $X_1, \ldots, X_n$ are

$$\bar{\alpha} = m_1' \left[ \frac{m_1'(1-m_1')}{m_2' - (m_1')^2} - 1 \right] \quad \text{and} \quad \bar{\beta} = (1-m_1') \left[ \frac{m_1'(1-m_1')}{m_2' - (m_1')^2} - 1 \right].$$

The MLEs for $\alpha$ and $\beta$ are

$$(\hat{\alpha}, \hat{\beta}) = \text{argmax}_{\alpha,\beta} \prod_{i=1}^{n} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} X_i^{\alpha-1}(1-X_i)^{\beta-1} \mathbf{1}(0 < X_i < 1).$$

We cannot get a closed-form solution for $(\hat{\alpha}, \hat{\beta})$; if we try to take the derivative of the log-likelihood, we find that the parameters are stuck inside gamma functions. We can nevertheless compute the MLEs in R using numerical methods.

In the following, we compute the MoMs estimators $(\bar{\alpha}, \bar{\beta})$ and the MLEs $(\hat{\alpha}, \hat{\beta})$ for a data set and compare them how well the Beta distribution with the estimated parameters fits the data.

**P-values from prostate cancer data set**

The following R code loads a data set containing $z$-values for testing, for each of 6033 genes, whether the gene is involved with prostate cancer. If none of the genes are involved, then the $p$-values should have a distribution which is approximately uniform. The data are taken from Efron (2012).

```
# Load the data from Efron's website
load(url("http://statweb.stanford.edu/~ckirby/brad/LSI/datasets-and-programs/data/prostz.RData"))

# Convert the test statistics to p-values
X <- 2*(1-pnorm(abs(prostz)))
n <- length(X)
```

As before, to compute the MLEs in R, we will use the R function `optim()`, which minimizes a function using numerical methods. We feed the `optim()` function the negative of the log-likelihood function in order to get the MLEs.

```
# Define the negative log-likelihood function
negll <- function(theta,X)
{
  alpha <- theta[1]
  beta <- theta[2]
  negll <- - sum( dbeta(X, shape1 = alpha, shape2 = beta, log = TRUE ))
  return(negll)
}

# Use the optim() function to find the MLEs of alpha and beta
mle <- optim(par=c(1,1),fn = negll, X = X)$par
alpha.hat <- mle[1]
beta.hat <- mle[2]

alpha.hat
```

```
## [1] 0.8222642
```

```
beta.hat
```

```
## [1] 0.9594244
```

```
# Compute the MoMs estimators of alpha and beta
m1 <- mean(X)
m2 <- mean(X^2)
alpha.bar <- m1 * ( m1 * ( 1 - m1 ) / ( m2 - m1^2 )  - 1)
beta.bar <- (1 - m1) * ( m1 * ( 1 - m1 ) / ( m2 - m1^2 )  - 1)

alpha.bar
```

```
## [1] 0.8852412
```

```
beta.bar
```

```
## [1] 0.9972068
```

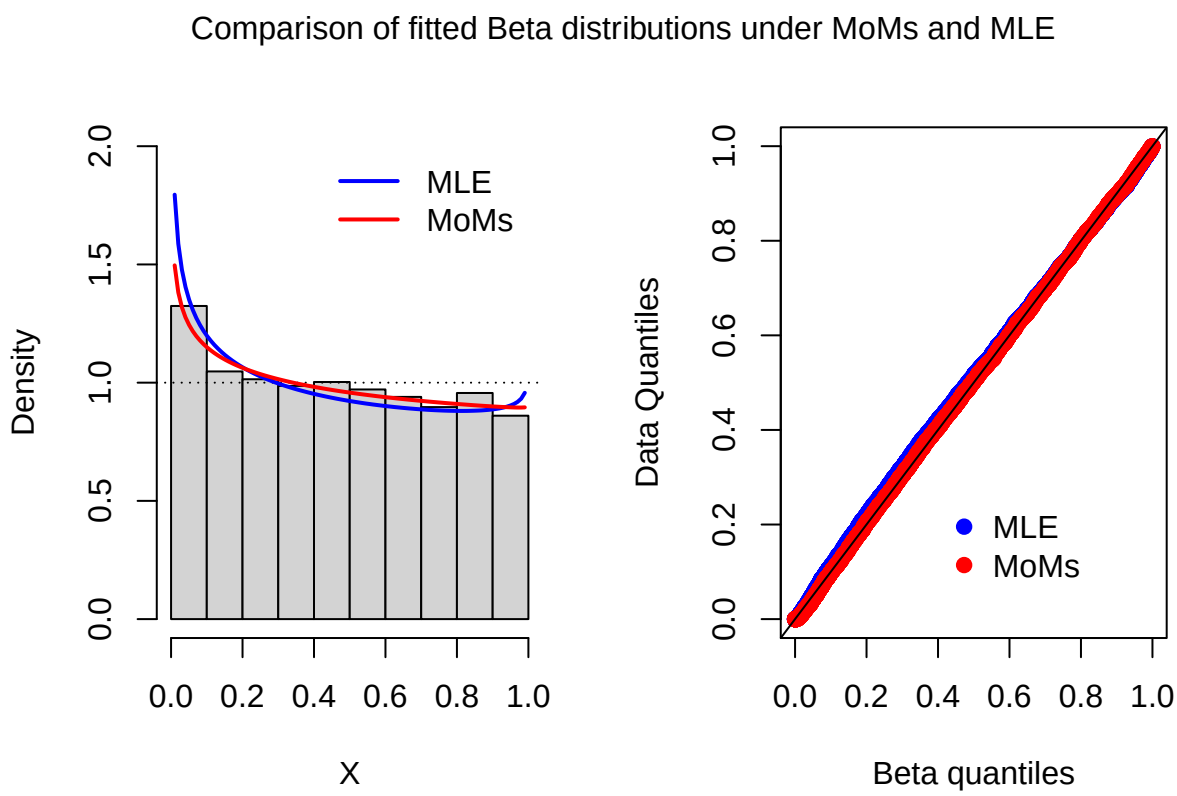
```
# Specify that the next two plots should be side-by-side
par(mfrow=c(1,2))

# Plot histogram of the data and overlay beta densities based on the MoMs and the MLEs:
hist(X,freq=FALSE,main="",ylim=c(0,2))
abline(h = 1,lty=3)
```

```r
x.seq <- seq(0,1,length=100)
lines(dbeta(x.seq,shape1=alpha.hat,shape2=beta.hat)~x.seq,col="blue",lwd=2)
lines(dbeta(x.seq,shape1=alpha.bar,shape2=beta.bar)~x.seq,col="red",lwd=2)
x.pos <- grconvertX(.5,from="nfc",to="user")
y.pos <- grconvertY(.8,from="nfc",to="user")
legend(x = x.pos, y = y.pos, legend = c("MLE","MoMs") ,
       col = c("blue","red"), lty=c(1,1), lwd=c(2,2), bty="n")

# Plot data quantiles against the corresponding quantiles of the fitted beta dists
beta.quantiles.mle <- qbeta((1:n)/(n+1),shape1=alpha.hat,shape2=beta.hat)
beta.quantiles.mom <- qbeta((1:n)/(n+1),shape1=alpha.bar,shape2=beta.bar)
plot(sort(X)~beta.quantiles.mle,xlab=paste("Beta quantiles",sep=""),
     ylab="Data Quantiles",pch=19,col="blue")
points(sort(X)~beta.quantiles.mom,pch=19,col="red")
abline(0,1)
x.pos <- grconvertX(.5,from="nfc",to="user")
y.pos <- grconvertY(.4,from="nfc",to="user")
legend(x = x.pos, y = y.pos, legend = c("MLE","MoMs") ,
       col = c("blue","red"), pch=c(19,19), bty="n")

# Add title to plot
mtext(side=3,outer=TRUE, "Comparison of fitted Beta distributions under MoMs and MLE",
      line=-2)
```



## Weibull distribution

The Weibull$(a, b)$ distribution has probability density function given by

$$f_X(x; a, b) = \left(\frac{a}{b}\right)\left(\frac{x}{b}\right)^{a-1} \exp\left[-\left(\frac{x}{b}\right)^a\right] \mathbf{1}(x > 0).$$

Given a random sample $X_1, \ldots, X_n$ from the Weibull$(a, b)$ distribution, the MoMs estimators of $a$ and $b$ are the solutions to

$$m_1' = b\Gamma\left(1 + \frac{1}{a}\right) \quad \text{and} \quad m_2' = b^2\Gamma\left(1 + \frac{2}{a}\right).$$

We cannot write down the MoMs estimators in closed-form; we find that we must do a numerical search for the value of $a$ which satisfies the equation

$$\frac{m_2'}{(m_1')^2} = \frac{\Gamma\left(1 + \frac{2}{a}\right)}{\left[\Gamma\left(1 + \frac{1}{a}\right)\right]^2},$$

and then plug this value into the following expression for $b$:

$$b = \frac{m_1'}{\Gamma\left(1 + \frac{1}{a}\right)}.$$

The MLEs $\hat{a}$ and $\hat{b}$ of $a$ and $b$ are given by

$$(\hat{a}, \hat{b}) = \text{argmax}_{a,b} \prod_{i=1}^{n} \left(\frac{a}{b}\right)\left(\frac{X_i}{b}\right)^{a-1} \exp\left[-\left(\frac{X_i}{b}\right)^a\right] \mathbf{1}(X_i > 0).$$

As with the MoMs estimators, neither is there a closed form for the MLEs, so we must use numerical methods to find them.

### Tree diameter data set

The following R code fits a Weibull distribution to a data set containing the diameter at breast height of 78 Alder trees in Camden, UK (Find the complete data set here). We compute from these data both the MoMs estimators and the MLEs for the parameters $a$ and $b$ of the Weibull$(a, b)$ distribution and compare the fits.

```
# Alder tree diameter at breast height with zeroes removed
X <- c(35, 24, 5, 15, 49, 50, 41, 21, 63, 44, 16, 57, 50, 25, 19, 25, 8, 20,
       16, 28, 41, 30, 38, 41, 5, 20, 24, 20, 5, 24, 24, 26, 36, 27, 23, 55,
       21, 26, 27, 49, 4, 66, 26, 90, 4, 24, 43, 19, 17, 63, 37, 37, 40, 16,
       26, 40, 34, 52, 18, 10, 9, 4, 13, 25, 23, 44, 35, 13, 25, 50, 26, 26,
       56, 40, 41, 28, 20, 4)
n <- length(X)
```

To compute the MoMs estimator of $a$, we will use the R function `uniroot()`, which finds the root of a univariate function; we will define a function of which the root is the MoMs estimator of $a$. Then the MoMs estimator of $b$ is a simple function of the MoMs estimator of $a$. To find the MLEs, we will use, as before, the R function `optim()`.

```
# Define the negative log-likelihood for the Weibull(a,b) distribution
negll <- function(theta,X)
{
  a <- theta[1]
  b <- theta[2]
  negll <- - sum( dweibull(X, shape = a , scale = b, log = TRUE) )
  return(negll)
```

```r
}

# Use the optim() function to find the MLEs of alpha and beta
mle <- optim(par=c(1,2),fn = negll, X = X)$par
a.hat <- mle[1]
b.hat <- mle[2]

a.hat
```

```
## [1] 1.825335
```

```r
b.hat
```

```
## [1] 33.43643
```

```r
# Compute the MoMs estimators of a and b using the uniroot() function
m1 <- mean(X)
m2 <- mean(X^2)

# Define a function, the root of which is the MoMs estimator of a
a.fun <- function(a,m1,m2)
{
  return( m2/m1^2 - gamma(1 + 2/a)/(gamma(1 + 1/a))^2 )
}

# Use the uniroot() function to find the root of the a.fun function
a.bar <- uniroot(a.fun,c(1,20),m1=m1,m2=m2)$root
b.bar <- m1/gamma(1 + 1/a.bar)

a.bar
```

```
## [1] 1.835746
```

```r
b.bar
```

```
## [1] 33.49081
```

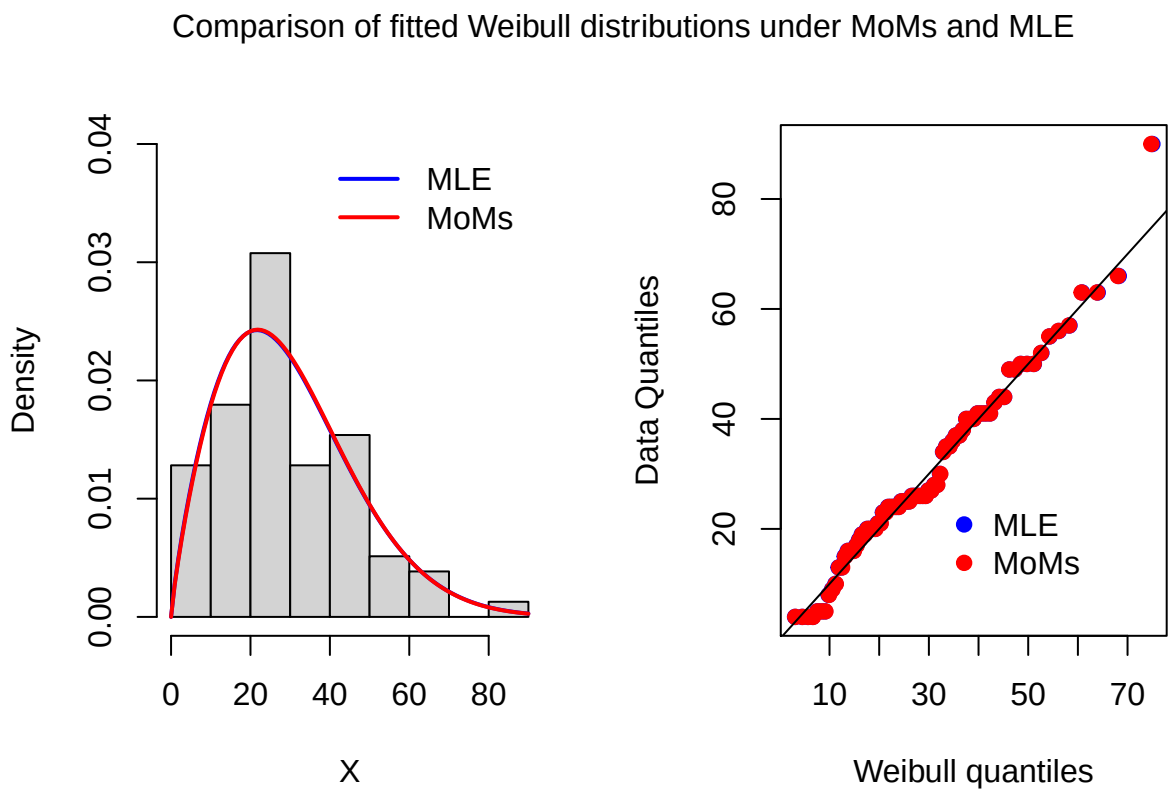
```r
# Specify that the next two plots should be side-by-side
par(mfrow=c(1,2))

# Plot histogram of the data and overlay Weibull densities based on the MoMs and the MLEs:
hist(X,freq=FALSE,main="",ylim=c(0,.04))
x.seq <- seq(0,90,length=100)
lines(dweibull(x.seq, shape = a.hat, scale = b.hat) ~ x.seq, col="blue", lwd=2)
lines(dweibull(x.seq, shape = a.bar, scale = b.bar) ~ x.seq, col="red", lwd=2)
x.pos <- grconvertX(.5,from="nfc",to="user")
y.pos <- grconvertY(.8,from="nfc",to="user")
legend(x = x.pos, y = y.pos, legend = c("MLE","MoMs") ,
       col = c("blue","red"), lty=c(1,1), lwd=c(2,2), bty="n")

# Plot data quantiles against the corresponding quantiles of the fitted Weibull dists
weibull.quantiles.mle <- qweibull((1:n)/(n+1),shape=a.hat,scale=b.hat)
weibull.quantiles.mom <- qweibull((1:n)/(n+1),shape=a.bar,scale=b.bar)
plot(sort(X)~weibull.quantiles.mle,xlab=paste("Weibull quantiles",sep=""),
  ylab="Data Quantiles",pch=19,col="blue")
points(sort(X)~weibull.quantiles.mom,pch=19,col="red")
abline(0,1)
```

```r
x.pos <- grconvertX(.5,from="nfc",to="user")
y.pos <- grconvertY(.4,from="nfc",to="user")
legend(x = x.pos, y = y.pos, legend = c("MLE","MoMs") ,
        col = c("blue","red"), pch=c(19,19), bty="n")

# Add title to plot
mtext(side=3,outer=TRUE, "Comparison of fitted Weibull distributions under MoMs and MLE",
      line=-2)
```




**Bibliography**


Davison, A. C. *Statistical Models.* Cambridge University Press, 2003.

Efron, Bradley. *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction.* Vol. 1. Cambridge University Press, 2012.