NOTAT NR. 3 – 2011

ERIK MØNNESS

The Power-Normal Distribution and Johnsons System bounded distribution: Computing details and programs

Høgskolen i **Hedmark**

Erik Mønness

The Power-Normal Distribution and Johnsons System bounded distribution: Computing details and programs

Høgskolen i Hedmark
Notat nr. 3 – 2011

Fulltekstutgave

Utgivelsessted: Elverum

Forfatteren er selv ansvarlig for sine konklusjoner. Innholdet gir
derfor ikke nødvendigvis uttrykk for Høgskolens syn.

I notatserien fra Høgskolen i Hedmark publiseres f.eks. milepældokumentasjon
av et forsknings- og/eller utviklingsprosjekt, eller annen dokumentasjon på at
et arbeid er i gang eller utført, samt interne HH-rapporter med allmenn interesse.

Notat nr. 3 – 2011

ISBN: 978-82-7671-830-0
ISSN: 1501-8555

Høgskolen i **Hedmark**

Tittel: Den Potensnormale fordeling og Johnsons system begrenset fordeling. Beregninger og programmer			
Forfatter: Erik Mønness			
Nummer: 3	**År:** 2011	**Sider:** 22	**ISBN:** 978-82-7671-830-0 **ISSN:** 1501-8555
Oppdragsgiver:			
Emneord: Potensnormalfordeling, Inverse Box-Cox, Johnsons System begrenset fordelingen, sannsynlighetsmaksimeringsestimering, fraktiler, diameter og høyde fordelinger			

Sammendrag:
Dette notatet presenterer den potensnormale (PN) fordelingen og Johnsons System begrenset fordelingen (SB). PN har sitt utgangspunkt i den inverse Box-Cox transformasjonen. Formler, estimeringsprosedyrer og programmer dokumenteres. Notatet er det tekniske grunnlaget for artikkelen “The Power-Normal Distribution: Application to forest stands” (Mønness 2011).

Hedmark University College

Title: The Power-Normal Distribution and Johnsons System bounded distribution: Computing details and programs

Author: Erik Mønness

Number: 3	**Year:** 2011	**Pages:** 22	**ISBN:** 978-82-7671-830-0 **ISSN:** 1501-8555

Financed by:

Keywords: Power-Normal distribution, Inverse Box-Cox, Johnson's System bounded distribution, Maximum likelihood estimation, Percentiles, Diameter and height distributions.

Summary:
This note presents the Power-Normal (PN) distribution and the Johnson System bounded distribution (SB). PN originate from the inverse Box-Cox transformation. Formulas, estimation procedures and programs are given.

The material is the technical basis for the article "The Power-Normal Distribution: Application to forest stands" (Mønness 2011).

THE POWER-NORMAL DISTRIBUTION AND JOHNSONS SYSTEM BOUNDED DISTRIBUTION: COMPUTING DETAILS AND PROGRAMS.

Erik Mønness
Hedmark University College, ØSIR
P.O. box 104, 2450 Rena Norway
Phone: +47 62430598
fax: +47 62430500
e-mail: erik.monness@hihm.no

Contents

SYSTEMS OF DISTRIBUTIONS

The Johnson Distributions

If a theoretical distribution is to estimate some empirical distribution, it must be able to, to some degree, to recreate location, scale and shape, and it must be possible to estimate the parameters of the theoretical distribution from the data. The Johnson's system (Johnson and Kotz 1972) consist of three non-linear transformations that cover the entire space of shapes, when shape is defined as a point in the skewness X kurtosis space.
Let *Z* be a standard Normal variate. Let X be the non-linear transformations given by:

$$Z = \gamma + \delta \cdot \log\left(\frac{X-\xi}{\lambda+\xi-X}\right) \qquad \text{With inverse} \qquad X = \xi + \lambda \cdot \left(1 + e^{\left(\frac{\gamma - Z}{\delta}\right)}\right)^{-1}$$

$$Z = \gamma + \delta \cdot \log\left(X - \xi\right) \qquad \text{With inverse} \qquad X = \xi + e^{\left(\frac{Z-\gamma}{\delta}\right)}$$

$$Z = \gamma + \delta \cdot \sinh^{-1}\left(\frac{X-\xi}{\lambda}\right) \qquad \text{With inverse} \qquad X = \xi + \lambda \cdot \sinh\left(\frac{Z-\gamma}{\delta}\right)$$

Johnson called these three transformations System bounded (SB), System lognormal (SL) and system unbounded (SU), respectively. SL does not have a λ parameter since it could not be separated from γ. The linear transformation $Z = \gamma + \delta \cdot X$ (Normal distribution) is a limiting case of all three. The idea of this system is to take benefit of the normal theory on the *Z* scale. SU has not been relevant in forestry.
(Lambert 1970) and partly revisited by(Rennolls and Wang 2005) introduced the following parameterization of SB, which is statistically better in (μ,σ) and numerically symmetric in lower and upper border (τ,θ). This parameterization is used here.

$$Z = \frac{\log\left(\frac{X-\tau}{\theta-X}\right) - \mu}{\sigma} \qquad \text{With inverse} \qquad X = \frac{\theta e^{Z\sigma+\mu} + \tau}{1 + e^{Z\sigma+\mu}} = \tau + (\theta - \tau)\left(1 + e^{-(Z\sigma+\mu)}\right)^{-1}$$

Estimating SB is elaborate. However, once (τ,θ) is found, (μ,σ) is readily available as in normal distribution. (Mønness 1982) proposed starting values for the iterative maximum likelihood procedure. (Johnson 1949) introduced estimating (τ,θ,μ,σ) by fractiles. However, the fractiles have to obey certain restrictions (Mønness 1982), revisited in (Siekierski 1992), so that not any set of fractiles yields a SB distribution. The full maximum likelihood method is complicated (appendix A1). (Hahn and Shapiro 1967) indicate that the quality of the fit is relatively unaffected by the choice of lower and upper bound as long as they are reasonable. However, the $\log\left(\frac{x}{1-x}\right)$ transformation on $x \in \langle 0,1 \rangle$ (basis of SB) is symmetric and rather linear in the interior, say $x \in \langle 0.2, 0.8 \rangle$. If upper and lower bound is chosen symmetric and somehow away from the data the transformation is close to linear and yields no change of shape. Thus quality estimates of (τ,θ) is crucial for SB. The log-Likelihood surface on (τ,θ) is rather flat around the maximizing value, this can imply convergence problems . Fixing the

lower border τ improves the numeric convergence. A SAS program for ML estimation is given in appendix A4.

The Power-Normal distribution

The Box-Cox transformation is applicable for positive data, *X≥0.*

$$Z=\frac{\frac{X^{\lambda}-1}{\lambda}-\mu}{\sigma} \qquad \lambda\neq 0 \qquad \text{With inverse} \qquad X=\left(1+\lambda\left(Z\sigma+\mu\right)\right)^{\frac{1}{\lambda}}$$

$$Z=\frac{\log\left(X\right)-\mu}{\sigma} \qquad \lambda=0 \qquad \text{With inverse} \qquad X=e^{(Z\sigma+\mu)}$$

The transformation is continuous in λ so estimation involves only the first functional form. The transformation can always be done, but Z can only be N(0,1) in the log() case, and the obvious case λ=1. However, X is near normal (Hernandez and Johnson 1980), actually it is a truncated normal (Freeman and Modarres 2006).
The transformed value is supposed to be normal (λ≠0):

$$P\left(X\le x\right)=P\left(\frac{X^{\lambda}-1}{\lambda}\le\frac{x^{\lambda}-1}{\lambda}\right)=\Phi\left(\frac{\frac{x^{\lambda}-1}{\lambda}-\mu}{\sigma}\right)=\Phi\left(\frac{x^{\lambda}-\left(1+\lambda\mu\right)}{\lambda\sigma}\right) \quad , \quad x\ge 0.$$

However, this is not a true cumulative distribution since the Φ() does not cover (0,1) since:

$\frac{X^{\lambda}-1}{\lambda}\in\left\langle-\infty,-\frac{1}{\lambda}\right\rangle$ when λ<0 and

$\frac{X^{\lambda}-1}{\lambda}\in\left\langle-\frac{1}{\lambda},\infty\right\rangle$ when λ>0

(When λ=0, the entire real line is covered).
The distribution is

$$F(x)=\frac{1}{K}\Phi\left(\frac{x^{\lambda}-\left(1+\lambda\mu\right)}{\lambda\sigma}\right) \qquad ,x\ge 0.$$

With density

$$f\left(x\right)=\frac{x^{\lambda-1}}{K\sqrt{2\pi}\sigma}e^{-\frac{1}{2}\left(\frac{x^{\lambda}-(1+\lambda\mu)}{\lambda\sigma}\right)^{2}} \quad , \; x\ge 0.$$

Where

$$K=\begin{cases}\Phi\left(-\left(\frac{1+\lambda\mu}{\lambda\sigma}\right)\right) & \text{When } \lambda<0\\ \Phi\left(\left(\frac{1+\lambda\mu}{\lambda\sigma}\right)\right) & \text{When } \lambda>0\end{cases}$$

When λ>0, K is typically close to 1 (often on real data μ>>σ). When λ<0, but close to zero, K is still close to 1. If λ<0 and moves away from zero, K becomes small and the truncation does

influence. However, large absolute values of λ are of little practical value since then the transformation becomes close to one vertical and one horizontal line intersection in (1,0).
The PN distribution has only one functional form, thus engaging only one maximum likelihood estimation procedure. If the estimated λ is close to zero one can change to the log-normal. Typically one ignores including K in the estimation step. A maximum likelihood procedure ignoring K is described in (Madansky 1988). The estimation is well-established and with good numerical properties. A SAS program is given in appendix A5.
When prediction a distribution, one has to include K in order to have a proper distribution. The shapes of the PN is explored in the main article (Mønness 2011).

Both SB and PN has the lognormal distribution as a limiting case

Prediction a distribution from a set of stands..

A typical forestry task is to first estimate parameters on each stand. Second, one uses the estimated distribution as a dependent observation in a regression, with stand characteristics as independent variables, in order to make predictions on future development and on other stands. Independent characteristics are typically site index, age, density, location/climate, basal area mean diameter etc. (Basal area mean diameter is typically an independent characteristics since it can be judged on site without knowing all the diameters, but it is also a property of the diameter distribution; $\sqrt{E\left(D^2\right)}$.)

The estimated parameters typically perform poorly as dependent variables since they may not act well in a linear model with stand characteristics. Also, different sets of parameters can yield nearly equal shapes. An alternative is to, from the estimated distributions, to calculate a certain set of fractiles that are used as dependent observations in the regression. After this second estimation, one must invert the calculations to get parameters.

With SB this was done by (Mønness 1982). A problem, mentioned earlier, is that not every set of 4 fractiles (minimum, median and two symmetrically chosen fractiles) can yield a SB distribution (appendix A2). With PN, any set of three fractiles (median and two symmetrically chosen fractiles) yields PN parameter estimates (appendix A3).

A1. MAXIMUM LIKELIHOOD OF SB.

Here the complete ML equations are presented. The parameterization is from (Lambert 1970). The likelihood of a set of n independent observations from a SB is

$$L(x_1,...,x_n;\mu,\sigma,\tau,\theta) = \frac{(\theta-\tau)^n}{(2\pi\sigma)^{\frac{n}{2}}\prod_i (x_i-\tau)(\theta-x_i)} \cdot \exp\left\{\frac{-1}{2\sigma}\sum_i\left(\log\left(\frac{x_i-\tau}{\theta-x_i}\right)-\mu\right)^2\right\}$$

Partial derivatives of σ and μ, given θ and τ are the usual mean and standard deviation of a normal distributed data set:

$$\hat{\sigma}(\tau,\theta) = \frac{1}{n}\sum\left[\log(x_i-\tau)-\log(\theta-x_i)-\hat{\mu}(\tau,\theta)\right]^2$$

and

$$\hat{\mu}(\tau,\theta) = \frac{1}{n}\sum\left[\log(x_i-\tau)-\log(\theta-x_i)\right]$$

The likelihood with these maximizing values inserted is

$$L^{**}(x_1,...,x_n;\tau,\theta) = \frac{(\theta-\tau)^n}{(\hat{\sigma}(\tau,\theta))^{\frac{n}{2}}(2\pi e)^{\frac{n}{2}}\prod_i (x_i-\tau)(\theta-x_i)}$$

The log likelihood to be maximized by Newton's method is then

$$\log L^{**}(\tau,\theta) = n\log(\theta-\tau)-\frac{n}{2}\log\hat{\sigma}(\tau,\theta)-\sum(x_i-\tau)-\sum\log(\theta-x_i)-\frac{n}{2}\log(2\pi e)$$

Equations to determine maximum (The ^ is omitted):

$$\frac{\partial \log L^{**}}{\partial\tau} = -n(\theta-\tau)^{-1}-\frac{n}{2}\sigma^{-1}\frac{\partial\sigma}{\partial\tau}+\sum(x_i-\tau)^{-1}=0$$

and

$$\frac{\partial \log L^{**}}{\partial\theta} = -n(\theta-\tau)^{-1}-\frac{n}{2}\sigma^{-1}\frac{\partial\sigma}{\partial\theta} \quad -\sum(\theta-x_i)^{-1}=0$$

2. order derivatives are:

$$\frac{\partial^2 \log L^{**}}{\partial\tau^2} = -n(\theta-\tau)^{-2}-\frac{n}{2}\left(-\sigma^{-2}\left(\frac{\partial\sigma}{\partial\tau}\right)^2+\sigma^{-1}\frac{\partial^2\sigma}{\partial\tau^2}\right) \quad +\sum(x_i-\tau)^{-2}$$

$$\frac{\partial^2 \log L^{**}}{\partial\theta^2} = -n(\theta-\tau)^{-2}-\frac{n}{2}\left(-\sigma^{-2}\left(\frac{\partial\sigma}{\partial\theta}\right)^2+\sigma^{-1}\frac{\partial^2\sigma}{\partial\theta^2}\right) \quad +\sum(\theta-x_i)^{-2}$$

$$\frac{\partial^2 \log L^{**}}{\partial\tau\partial\theta} = \frac{\partial^2 \log L^{**}}{\partial\theta\partial\tau} = -n(\theta-\tau)^{-2}-\frac{n}{2}\left(-\sigma^{-2}\frac{\partial\sigma}{\partial\theta}\frac{\partial\sigma}{\partial\tau}+\sigma^{-1}\frac{\partial^2\sigma}{\partial\theta\partial\tau}\right)$$

The function $\sigma(\tau,\theta)$ must be calculated and differentiated (The ^ is omitted):

$$\sigma(\tau,\theta) = \frac{1}{n}\sum\left\{\log(x_i-\tau)-\log(\theta-x_i)-\hat{\mu}(\tau,\theta)\right\}^2$$

$$\frac{\partial\sigma}{\partial\tau}=\frac{2}{n}\left[-\sum\frac{\log(x_i-\tau)}{(x_i-\tau)}+\sum\frac{\log(\theta-x_i)}{(x_i-\tau)}+\frac{1}{n}\sum\log(x_i-\tau)\sum(x_i-\tau)^{-1}-\frac{1}{n}\sum\log(\theta-x_i)\sum(x_i-\tau)^{-1}\right]$$

$$\frac{\partial\sigma}{\partial\theta}=\frac{2}{n}\left[\sum\frac{\log(\theta-x_i)}{(\theta-x_i)}-\sum\frac{\log(x_i-\tau)}{(\theta-x_i)}-\frac{1}{n}\sum\log(\theta-x_i)\sum(\theta-x_i)^{-1}+\frac{1}{n}\sum\log(x_i-\tau)\sum(\theta-x_i)^{-1}\right]$$

$$\frac{\partial^2\sigma}{\partial\theta^2}=\frac{2}{n}\left[\sum(\theta-x_i)^{-2}-\sum\frac{\log(\theta-x_i)}{(\theta-x_i)^2}+\sum\frac{\log(x_i-\tau)}{(\theta-x_i)^2}\right]$$
$$-\frac{2}{n^2}\left[\left(\sum(\theta-x_i)^{-1}\right)^2-\sum\log(\theta-x_i)\sum(\theta-x_i)^{-2}+\sum\log(x_i-\tau)\sum(\theta-x_i)^{-2}\right]$$

$$\frac{\partial^2\sigma}{\partial\theta^2}=\frac{2}{n}\left[\sum(x_i-\tau)^{-2}-\sum\frac{\log(x_i-\tau)}{(x_i-\tau)^2}+\sum\frac{\log(\theta-x_i)}{(x_i-\tau)^2}\right]$$
$$-\frac{2}{n^2}\left[\left(\sum(x_i-\tau)^{-1}\right)^2-\sum\log(x_i-\tau)\sum(x_i-\tau)^{-2}+\sum\log(\theta-x_i)\sum(x_i-\tau)^{-2}\right]$$

$$\frac{\partial^2\sigma}{\partial\theta\partial\tau}=\frac{\partial^2\sigma}{\partial\tau\partial\theta}=\frac{2}{n}\left[\sum(x_i-\tau)^{-1}(\theta-x_i)^{-1}-\frac{1}{n}\sum(\theta-x_i)^{-1}\sum(x_i-\tau)^{-1}\right]$$

These formulas involve 18 different sums of the observations (Appendix A4). Initial values are discussed in (Mønness 1982). If τ is forced to 0 or another fixed value, this value can be substituted everywhere and then only the θ derivatives are used.

A2. SB FROM FRACTILES.

Johnson (1949) indicated the estimating procedure that also can be used as a means to go from predicted fractiles to a predicted distribution. Suppose four fractiles is given or is predicted from a regression. Then $\left(X_{(0)},X_{(p)},X_{(0.5)},X_{(1-p)}\right)$ can give a closed solution of θ as given below.

Let $f_p=X_{(p)}-X_{(0)}$, (the lower p-percentile minus the lower bound), $f_{1-p}=X_{(1-p)}-X_{(0)}$ and $f_{0.5}=X_{(0.5)}-X_{(0)}$ (the median minus the lower bound). Then $\tau=X_{(0)}$ and

$$\theta=\tau+f_{0.5}\cdot\frac{f_{0.5}f_p+f_{0.5}f_{1-p}-2f_pf_{1-p}}{f_{0.5}^2-f_pf_{1-p}}.$$

The denominator has to be positive. Thus not all possible set of percentiles can give raise to a SB. μ and σ are readily available.

Let $x=f_{0.5}/f_{1-p}$ and $y=f_{0.5}/f_p$. Then the θ-τ expression can be reformulated as a function of the kind $\frac{x+y-2}{x\cdot y-1}$ with $0<x<1<y$. This function has singularities along xy=1 and can give unstable solutions.

A3. PN FROM FRACTILES.

Suppose the (diameter or height) fractiles $X_{(p)}, X_{(0.50)}, X_{(1-p)}$ are given (from prediction regressions or from actual real data). In order to find the parameter set λ, μ, σ the following equations have to be solved:

$$X_{(p)} = \left(1 + \lambda\left[z_p\sigma + \mu\right]\right)^{\frac{1}{\lambda}}$$

$$X_{(0.50)} = \left(1 + \lambda\left[z_{0.50}\sigma + \mu\right]\right)^{\frac{1}{\lambda}} = \left(1 + \lambda\mu\right)^{\frac{1}{\lambda}}$$

$$X_{(1-p)} = \left(1 + \lambda\left[z_{1-p}\sigma + \mu\right]\right)^{\frac{1}{\lambda}}$$

where the z's are fractiles from the standard normal distribution.
λ can be solved from

$\left(\frac{X_{(1-p)}}{X_{(0.50)}}\right)^{\lambda} + \left(\frac{X_{(p)}}{X_{(0.50)}}\right)^{\lambda} = 2$, which has a zero solution in addition to the solution of interest.

Then

$$\mu = \frac{X_{(0.50)}{}^{\lambda} - 1}{\lambda} \text{ and } \sigma = \frac{X_{(1-p)}{}^{\lambda} - X_{(0.50)}{}^{\lambda}}{z_{(1-p)}\lambda}$$

The first equation has no explicit solution but can be solved numerically.

Let $F(\lambda) = \left(\frac{X_{(1-p)}}{X_{(0.50)}}\right)^{\lambda} + \left(\frac{X_{(p)}}{X_{(0.50)}}\right)^{\lambda} - 2$

The minimum value is explicitly determined by

$$\lambda_0 = \frac{\log\left(\frac{\log\left(\frac{X_{(0.50)}}{X_{(p)}}\right)}{\log\left(\frac{X_{(1-p)}}{X_{(0.50)}}\right)}\right)}{\log\left(\frac{\frac{X_{(1-p)}}{X_{(0.50)}}}{\frac{X_{(p)}}{X_{(0.50)}}}\right)} = \frac{\log\left(\frac{\log\left(X_{(0.50)}\right) - \log\left(X_{(p)}\right)}{\log\left(X_{(1-p)}\right) - \log\left(X_{(0.50)}\right)}\right)}{\log\left(X_{(1-p)}\right) - \log\left(X_{(p)}\right)}$$

If

$\lambda_0 < 0$ then $\lambda_{solution} < \lambda_0$

$\lambda_0 = 0$ then $\lambda_{solution} = 0$

$\lambda_0 > 0$ then $\lambda_{solution} > \lambda_0$

Thus starting values of the numeric solution are determined. Newton's method (Henrici 1964) is

$\lambda_{n+1} = \lambda_n - \frac{F(\lambda_n)}{F'(\lambda_n)}$ until convergences.

The use of the median is critical to solve the system.

A4. SAS PROGRAM ML FOR SB

The SAS program (SAS 2008) for maximum likelihood estimation. In the progam, *sigma* is named *beta*

```
/*-------------------------------------------------------------------------
                           Beregning av parametre i Johnson Sb.
                             for bivariat fordeling diametre og
                             høyder.
                           estimeringsmetode ML med nedre
                                                (tau)=xmin-faktor0
                             desember 2009.
                             fil= hoydiam.dhparJSbf_09.sas
                            Beregner parametre og legger ut enkeltrær
-------------------------------------------------------------------------*/
data hoydiam.dhparJSbf
(keep=flate dtau dteta dbeta dmy dloglikl ditr dispes
            htau hteta hbeta hmy hloglikl hitr hispes
                           n     ro)
     hoydiam.JSbf_dh (keep=flate lfd lfh)              ;

set hoydiam.dh ;by flate;

array dd{550} 3 _temporary_;                  /*  enkelt-dataverdier */
array hh{550} 3 _temporary_;                  /*  enkelt-dataverdier */
array f{18} _temporary_;                      /*  f(hver obs)  */
array s{18} _temporary_;                      /*   "kvadratsummer" */
retain n dmin dmax hmin hmax   dd hh ;
/*-----------------------------------innlesing av alle trær--------------*/

if first.flate
then           do;                             /*initier tellere */
                 n=1;
                 dmin=diam ; dmax=diam ; dd{1}=diam ;
                 hmin=hoyde; hmax=hoyde; hh{1}=hoyde;
               end;
else           do;                             /* les inn en flate */
                 n=n+1;
                 dd{n}=diam; hh{n}=hoyde;
                        if diam <dmin then dmin=diam ;
                         else  if diam >dmax then dmax=diam ;
                        if hoyde<hmin then hmin=hoyde;
                         else  if hoyde>hmax then hmax=hoyde;
               end;
/*-------------------------------------------------beregn for en flate */
if last.flate then do;
/*---------------------------------------diametre--------------------*/

   ispes=0; itr=0;
                             /* startverdier iterasjon */
   faktor=(dmax - dmin)/sqrt(n);
/*   tau=max(0,dmin-faktor)*/
   tau= 0;
   teta=dmax+2*faktor;

   do while (ispes=0 and itr< 50                 );

   do j=1 to 18; s{j}=0; end;            /* nullstiller summer */
                 do i=1 to n;
                           f{ 1}=1/(dd{i}-tau);
                           f{ 2}=1/(teta - dd{i} );

                           f{ 3}=f{ 1}*f{ 1};
```

```
                          f{ 4}=f{ 2}*f{ 2};
                          f{ 5}=f{ 1}*f{ 2};
                          f{ 6}=  log(dd{i}-tau);
                          f{ 7}=  log(teta-dd{i});
                          f{ 8}=f{ 6}*f{ 7};
                          f{ 9}=f{ 6}*f{ 6};
                          f{10}=f{ 7}*f{ 7};
                          f{11}=f{ 6}*f{1};
                          f{12}=f{ 7}*f{1};
                          f{13}=f{ 6}*f{2};
                          f{14}=f{ 7}*f{2};
                          f{15}=f{ 6}*f{3};
                          f{16}=f{ 7}*f{3};
                          f{17}=f{ 7}*f{4};
                          f{18}=f{ 6}*f{4};

                          do j=1 to 18; s{j}=s{j} + f{j}; end;
                          end;


    beta=( s{9} + s{10} - 2*s{8} -( s{6}*s{6} + s{7}*s{7} -2*s{6}*s{7})
         /n )/n;
    my=    (s{6} - s{7})/n;

              if itr=0 then do;
/*                              tau0= max(0,dmin-faktor)*/
                               tau0= 0;
                               teta0=dmax+2*faktor;
                               my0=my;
                               beta0=beta;
                               s60=s{6};
                               s70=s{7};
                               end;

                                                    /* log(2*pi*e) */
    bDtet  =2*( s{14} - s{13} + (s{6} - s{7})*s{2}/n  )/n;
    bDtettet =2* ( s{4} - s{17} + s{18} -( s{2}*s{2} - s{7}*s{4} +
                                        s{4}*s{6} )/n  )/n;


    LDtet= -s{2} + n/(teta-tau) - n*bDtet/(2*beta);

    LDtettet= s{4} - n/((teta-tau)*(teta-tau))
                   +(n/2)*( bDtet*bDtet/(beta*beta) - bDtettet/beta);

            if LDtettet NE 0 then do;
               del_tet=LDtet/ LDtettet;

                       if      teta - del_tet > dmax +0.5
                           THEN do;
                                endring=sqrt((del_tet**2)
                                             /( teta**2));
                                if endring > 0.001 then do;
                                                    itr=itr+1;
                                                    teta=teta-del_tet;
                                                    end;
                                                ELSE   ispes=1;
                                end;
                           ELSE ispes=2;
                       end;
                       else ispes=3;
     end;  /* end while*/
    if (ispes NE 1) OR (teta>2*teta0) then do;
                                       /* bruker startverdier */
                                       tau= tau0;
                                       teta=teta0;
                                       beta=beta0;
                                       my=my0;
                                       s{6}=s60;
                                       s{7}=s70;
                                       ispes=10+ispes;
                                       end;

    loglikli= n*log(teta-tau) - (n/2)*log(beta) - s{6} -s{7} -
                                (n/2)*2.837877066409;
    dbeta=  beta;
    dmy=    my;
```

```
dtau=tau; dteta=teta;
dn=n;                       dloglikl=loglikli; dispes=ispes; ditr=itr;
/*-----------------------------------------høyder--------------------*/
   ispes=0; itr=0;
                              /* startverdier iterasjon */
   faktor=(hmax - hmin)/sqrt(n);
   tau= max(0,hmin-faktor);
   teta=hmax+faktor;

   do while (ispes=0 and itr< 50                 );

   do j=1 to 18; s{j}=0; end;            /* nullstill summer */
                 do i=1 to n;
                            f{ 1}=1/(hh{i}-tau);
                            f{ 2}=1/(teta - hh{i} );

                            f{ 3}=f{ 1}*f{ 1};
                            f{ 4}=f{ 2}*f{ 2};
                            f{ 5}=f{ 1}*f{ 2};
                            f{ 6}=  log(hh{i}-tau);
                            f{ 7}=  log(teta-hh{i});
                            f{ 8}=f{ 6}*f{ 7};
                            f{ 9}=f{ 6}*f{ 6};
                            f{10}=f{ 7}*f{ 7};
                            f{11}=f{ 6}*f{1};
                            f{12}=f{ 7}*f{1};
                            f{13}=f{ 6}*f{2};
                            f{14}=f{ 7}*f{2};
                            f{15}=f{ 6}*f{3};
                            f{16}=f{ 7}*f{3};
                            f{17}=f{ 7}*f{4};
                            f{18}=f{ 6}*f{4};

                            do j=1 to 18; s{j}=s{j} + f{j}; end;
                            end;


    beta=( s{9} + s{10} - 2*s{8} -( s{6}*s{6} + s{7}*s{7} -2*s{6}*s{7})
         /n )/n;
    my=    (s{6} - s{7})/n;

                if itr=0 then do;
                                 tau0= max(0,hmin-faktor);
                                 teta0=hmax+faktor;
                                 my0=my;
                                 beta0=beta;
                                 s60=s{6};
                                 s70=s{7};
                                 end;

                                                    /* log(2*pi*e) */
    bDtet  =2*( s{14} - s{13} + (s{6} - s{7})*s{2}/n  )/n;
    bDtettet =2* ( s{4} - s{17} + s{18} -( s{2}*s{2} - s{7}*s{4} +
                                        s{4}*s{6} )/n  )/n;


    LDtet= -s{2} + n/(teta-tau) - n*bDtet/(2*beta);

    LDtettet= s{4} - n/((teta-tau)*(teta-tau))
                   +(n/2)*( bDtet*bDtet/(beta*beta) - bDtettet/beta);

            if LDtettet NE 0 then do;
               del_tet=LDtet/ LDtettet;

                       if      teta - del_tet > hmax +0.5
                           THEN do;
                                endring=sqrt((del_tet**2)
                                             /( teta**2));
                                if endring > 0.001 then do;
                                                        itr=itr+1;
                                                        teta=teta-del_tet;
                                                        end;
                                                   ELSE   ispes=1;
                                end;
                           ELSE ispes=2;
                       end;
                       else ispes=3;
```

```
     end;  /* end while*/
    if (ispes NE 1) OR (teta>2*teta0) then do;
                                        /* bruker startverdier */
                                        tau=tau0;
                                        teta=teta0;
                                        beta=beta0;
                                        my=my0;
                                        s{6}=s60;
                                        s{7}=s70;
                                        ispes=10+ispes;
                                        end;

    loglikli= n*log(teta-tau) - (n/2)*log(beta) - s{6} -s{7} -
                                (n/2)*2.837877066409;
    hbeta=  beta;
    hmy=    my;
htau=tau; hteta=teta;
hn=n;                       hloglikl=loglikli; hispes=ispes; hitr=itr;
/*---------------------------bivariate parametre og utlegg av enkelttrær--*/
        ro=0;
        do i=1 to n;
               lfd=(log((dd{i}-dtau)/(dteta-dd{i})) -dmy)/sqrt(dbeta);
               lfh=(log((hh{i}-htau)/(hteta-hh{i}))-hmy)/sqrt(hbeta);;
                ro=ro+lfd*lfh;
                output hoydiam.JSbf_dh;
        end;
        ro=ro/n;
/*        teta=exp((ro*dgamma-hgamma)/hdelta);
        fi=ro*ddelta/hdelta;*/

/*-----------------------------------------utlegging parametre------------*/
output hoydiam.dhparJSbf;
put 'flate= ' flate ' ro= ' ro;
end; /*last.flate*/

 run;
```

A5. SAS PROGRAM ML FOR PN

The SAS program (SAS 2008) for maximum likelihood estimation

```
/*-------------------------------------------------------------------------
                                    Beregning av parametre i BoxCox
                                    for bivariat fordeling diametre og
                                    høyder.
                                    estimeringsmetode s178 Albert Madansky
                                   juni 1993.
                                    fil= hoydiam\dhbox1.sas
-------------------------------------------------------------------------*/
data hoydiam.dhbox1 (keep= flate dbmy dbsigma dlanda hbmy hbsigma hlanda ro)
     hoydiam.box_dh (keep=flate box_d box_h)              ;

set hoydiam.dh (obs=300);by flate;

array x_d{550} 3 _temporary_;                  /*  enkelt-dataverdier */
array x_h{550} 3 _temporary_;                  /*  enkelt-dataverdier */
array yl_d{550} _temporary_;
array yl_h{550} _temporary_;
array dy_d{550} _temporary_;
array dy_h{550} _temporary_;
retain n x_d x_h yl_d yl_h dy_d dy_h yx_d yx_h    ;
/*-----------------------------------innlesing av alle trær--------------*/

if first.flate
then           do;                             /*initier tellere */
                 n=1;
                 x_d{1}=diam ;
                 x_h{1}=hoyde;
                yx_d=log(diam); yx_h=log(hoyde);
               end;
else           do;                             /* les inn en flate */
                 n=n+1;
                 x_d{n}=diam; x_h{n}=hoyde;
                yx_d=yx_d+log(diam); yx_h=yx_h+log(hoyde);
               end;
/*-------------------------------------------------beregn for en flate */
if last.flate then do;
/*-----------------------------------------diametre--------------------*/
xlanda_d=0.25;
wz=1;
do while (wz>0.000005);
        do i=1 to n;
                yl_d{i}=(x_d{i}**xlanda_d -1)/xlanda_d;
                end;
        ym_d=0; ys_d=0;
        do i=1 to n;
                ym_d=ym_d+yl_d{i}/n;                /*y_landa mean */
                ys_d=ys_d+yl_d{i}*yl_d{i};
                end;
                ys_d=(ys_d-n*ym_d*ym_d)/n;              /* var y_landa  */
        z_d=0; xlx_d=0; yxlx_d=0;
        do i=1 to n;
                xlx_d=xlx_d+log(x_d{i})*x_d{i}**xlanda_d;
                yxlx_d=yxlx_d + yl_d{i}*log(x_d{i})*x_d{i}**xlanda_d ;
                z_d=z_d+(yl_d{i}-ym_d)*((1+xlanda_d*yl_d{i})*
                        log(1+xlanda_d*yl_d{i}) - xlanda_d*yl_d{i});
                end;
                z_d=yx_d-z_d/(xlanda_d*xlanda_d*ys_d);
        do i=1 to n;
                dy_d{i}=(-yl_d{i}+x_d{i}**xlanda_d * log(x_d{i}))/xlanda_d;
                end;
        dm_d=0;
        do i =1 to n;
                dm_d=dm_d+dy_d{i}/n;
                end;
        dsig_d=-2*ys_d/xlanda_d +2*(yxlx_d-xlx_d*ym_d)/(xlanda_d*n);
```

```
        g_d= -(z_d-yx_d)*(2*xlanda_d*ys_d +
                xlanda_d*xlanda_d*dsig_d)/(xlanda_d*xlanda_d*ys_d);
        s2_d=0; s1_d=0;
        do i =1 to n;
                wy_d=1+xlanda_d*yl_d{i};
                s1_d=s1_d +
                     (yl_d{i}-ym_d)*log(wy_d)*(yl_d{i}+xlanda_d*dy_d{i});
                s2_d=s2_d +(dy_d{i}-dm_d)*(wy_d*log(wy_d)-xlanda_d*yl_d{i});
                end;
        g_d=g_d-(s1_d+s2_d)/(xlanda_d*xlanda_d*ys_d);
        put flate xlanda_d z_d g_d;
        wz=abs(z_d);
        if wz>0.000005 then xlanda_d=xlanda_d-z_d/g_d;

end;

/*-----------------------------------------høyder--------------------*/
xlanda_h=0.25;
wz=1;
do while (wz>0.000005);
        do i=1 to n;
                yl_h{i}=(x_h{i}**xlanda_h -1)/xlanda_h;
                end;
        ym_h=0; ys_h=0;
        do i=1 to n;
                ym_h=ym_h+yl_h{i}/n;                 /*y_landa mean */
                ys_h=ys_h+yl_h{i}*yl_h{i};
                end;
                ys_h=(ys_h-n*ym_h*ym_h)/n;               /* var y_landa  */
        z_h=0; xlx_h=0; yxlx_h=0;
        do i=1 to n;
                xlx_h=xlx_h+log(x_h{i})*x_h{i}**xlanda_h;
                yxlx_h=yxlx_h + yl_h{i}*log(x_h{i})*x_h{i}**xlanda_h ;
                z_h=z_h+(yl_h{i}-ym_h)*((1+xlanda_h*yl_h{i})*
                        log(1+xlanda_h*yl_h{i}) - xlanda_h*yl_h{i});
                end;
                z_h=yx_h-z_h/(xlanda_h*xlanda_h*ys_h);
        do i=1 to n;
                dy_h{i}=(-yl_h{i}+x_h{i}**xlanda_h * log(x_h{i}))/xlanda_h;
                end;
        dm_h=0;
        do i =1 to n;
                dm_h=dm_h+dy_h{i}/n;
                end;
        dsig_h=-2*ys_h/xlanda_h +2*(yxlx_h-xlx_h*ym_h)/(xlanda_h*n);
        g_h= -(z_h-yx_h)*(2*xlanda_h*ys_h +
                xlanda_h*xlanda_h*dsig_h)/(xlanda_h*xlanda_h*ys_h);
        s2_h=0; s1_h=0;
        do i =1 to n;
                wy_h=1+xlanda_h*yl_h{i};
                s1_h=s1_h +
                     (yl_h{i}-ym_h)*log(wy_h)*(yl_h{i}+xlanda_h*dy_h{i});
                s2_h=s2_h +(dy_h{i}-dm_h)*(wy_h*log(wy_h)-xlanda_h*yl_h{i});
                end;
        g_h=g_h-(s1_h+s2_h)/(xlanda_h*xlanda_h*ys_h);
        put flate xlanda_h z_h g_h;
        wz=abs(z_h);
        if wz>0.000005 then xlanda_h=xlanda_h-z_h/g_h;
end;
/*---------------------------bivariate parametre og utlegg av enkelttrær--*/

        ro=0;
                dlanda=xlanda_d; hlanda=xlanda_h;
                dbsigma=sqrt(ys_d); dbmy=ym_d;
                hbsigma=sqrt(ys_h); hbmy=ym_h;
        do i=1 to n;
                box_d=(yl_d{i}-dbmy)/dbsigma;
                box_h=(yl_h{i}-hbmy)/hbsigma;
               ro=ro+box_d*box_h;
                output hoydiam.box_dh;
        end;
        ro=ro/n;

/*---------------------------------------utlegging parametre------------*/

output hoydiam.dhbox1;
put 'flate= ' flate ' ro>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>= ' ro;
```

```
put dlanda hlanda;
end; /*last.flate*/

 run;
```

REFERENCES:

Freeman, J. and R. Modarres. 2006. Inverse box-cox: The power-normal distribution. Statistics & Probability Letters 76, no. 8: 764-72.

Hahn, G.J. and S.S. Shapiro. 1967. *Statistical models in engineering*: John Wiley and Sons, Inc.

Henrici, P. 1964. *Elements of numerical analysis*. London: John Wiley & Sons Inc.

Hernandez, F. and R.A. Johnson. 1980. The large-sample behavior of transformations to normality. Journal of the American Statistical Association 75, no. 372: 855-61.

Johnson, N.L. 1949. Systems of frequency curves generated by methods of translation. Biometrica 36: 149-76.

Johnson, N.L. and S. Kotz. 1972. *Distributions in statistics. Continuous univariate distributions*: ohn Wiley & Sons, Inc.

Lambert, J.L. 1970. Estimation of parameters in the four-parameter lognormal distribution. Australian Journal of Statistics 12, no. 1: 33-43.

Madansky, A. 1988. *Prescriptions for working statisticians*: Springer Verlag.

Mønness, E. 1982. Diameter distributions and height curves in even-aged stands of pinus sylvestris l. Medd. Nor. inst. skogforsk 36, no. 15: 1-43.

Mønness, E. 2011. The power-normal distribution: Application to forest stands. Canadian Journal of Forest Research 41, no. 4: 707-14.

Rennolls, K. and M. Wang. 2005. A new parameterization of johnson's sb distribution with application to fitting forest tree diameter data. Can. J. For. Res. 35: 575–79.

Sas. 2008. Sas 9.1. Cary, NC, USA.: SAS.

Siekierski, K. 1992. Comparison and evaluation of three methods of estimation of the johnson s_b distribution. Biometrical Journal 34, no. 7: 16.