Supplement: Inferring skeletal production from time-averaged assemblages: timing of production pulses is pulled towards the modern times

Adam Tomašových, Susan M. Kidwell, Rina Foygel Barber

Source code for model likelihoods and figures written in R language

Six density functions for fitting parameters of models with optimizing likelihoods are followed by age data of four depth assemblages. Fitting of age data with models occurs within the loop, followed by (1) figures showing the relation between the observed medians and IQRs and the values expected under different models, (2) likelihood surfaces for the Weibull and two-phase model, (3) simulations showing the effect of loss rate on the shape of AFD under one-phase and two-phase models, and (4) simulations showing the effects of sample size and fluctuating sinusoidal production on loss-rate parameters.

```
#################################################################
#WEIBULL MODEL
#################################################################
weibull=function(x) {
        age=age
        x[1]=exp(x[1])
        x[2]=exp(x[2])
        C<-(1/x[1])*gamma(1+1/x[2])
        lik<-(1/C)*exp(-(x[1]*age)^x[2])
#if likelihood is zero, it can be replaced by very small values
#to avoid infinities with something as
lik=ifelse(lik<0.00000001,0.00000001,lik)
        lik=log(lik)
return(-sum(lik))
}


#################################################################
#TWO-PHASE EXPONENTIAL MODEL
#################################################################
```

```
TwoPhase=function(age,niters=10000,startat=5000,a=2,b=0.001,rmean
=mean(X),OutputType="best"){
# prior on beta=Beta(a,a); posterior = Beta(a+N1,a+N2)
# prior on r1 & r2 =Gamma(b,1/(rmean*b);
# posterior = Gamma(b+n,1/(rmean*b+mean(X1 or X2)*n))
# if OutputType is set to "all" then the output will be all the
# parameter values sampled in the MCMC,
# starting at interation startat
X=age
N=length(X)
d=function(x){
      r1*exp(-r1*x)*beta+r2*exp(-r2*x)*(1-beta)
      }
beta=0.5;r1=1;r2=1;best=c(beta,r1,r2);dbest=sum(log(d(X)))
cluster=rbinom(N,1,beta);c=which(cluster==1);StorePars=NULL

for(iter in 1:niters){
      beta=rbeta(1,a+length(c),a+N-length(c))
      r1=rgamma(1,b+length(c),rate=rmean*b+sum(X[c]));
      r2=rgamma(1,b+N-length(c),rate=rmean*b+sum(X[-c]))
      L1=beta*r1*exp(-r1*X)
      L2=(1-beta)*r2*exp(-r2*X)
      cluster=rbinom(N,1,L1/(L1+L2));
      c=which(cluster==1)
      dnew=sum(log(d(X)));
      if(dnew>dbest){
            dbest=dnew;best=c(beta,r1,r2)
            }

      if(iter>=startat){
      StorePars=rbind(StorePars,c(beta,r1,r2,dnew))
      }
}

beta=best[1];r1=best[2];r2=best[3]
if(r1>r2){beta=1-beta;r1temp=r1;r1=r2;r2=r1temp}
out=list();
out$beta=beta;
out$r1=r1;
out$r2=r2
out$alpha=(beta*r1)/(beta*r1+(1-beta)*r2)
out$tau=out$alpha*(r2-r1);
out$lambda1=r2-out$tau;
out$lambda2=r1
out$lik=dnew

if(OutputType=="all"){
beta=StorePars[1];
r1=StorePars[2];
r2=StorePars[3];
alpha=(beta*r1)/(beta*r1+(1-beta)*r2);
tau=alpha*(r2-r1);
```

```
lambda1=r2-tau;
lambda2=r1
StorePars=cbind(StorePars[,1:3],alpha,tau,lambda1,lambda2,
StorePars[,4])
colnames(StorePars)=c("beta","r1","r2","alpha","tau",
"lambda1","lambda2","logLikelihood")
return(StorePars)
}
else{return(out)}
}


#####################################################################
#ONE-PHASE TRUNCATED-NORMAL MODEL
#####################################################################
OnePhaseTN=function(age){
x=age
n=length(x)
s0=-sum(x^2)/2/n
s1=sum(x)/n
# log likelihood: s0*a+s1*b-A(a,b), here a=1/sig^2 and
# b=mu/sig^2
A=function(a,b){
      b^2/2/a-1/2*log(a)+log(pnorm(b/sqrt(a)))
      }
neglogLik=function(ab){
      -s0*ab[1]-s1*ab[2]+A(ab[1],ab[2])
      }
opt=optim(c(0.0001,0),neglogLik);
a=opt$par[1];
b=opt$par[2]
out=list();
out$mu=b/a;
out$sig2=1/a;
out$lik=opt$value
out
}

#####################################################################
#TWO-PHASE TRUNCATED-NORMAL MODEL
#####################################################################

TwoPhaseTN=function(theta){
# p is alpha in two-phase exponential model (which is related to
# beta in the right-censored equation)
# sigma is standard deviation - the same for both truncated
# normals,
# mu1 and mu2 - means of two truncated normals
# theta=(p,mu1,mu2,sigma^2)
logitp=log(theta[1]/(1-theta[1]));
sig2=theta[4];
mu1=theta[2];
```

```
mu2=theta[3]
lik=exp(logitp)/(exp(logitp)+1)*1/2/sig2*exp(-
1/2/sig2*age^2+age*mu1/sig2-mu1^2/2/sig2)/pnorm(mu1/sqrt(sig2))
+1/(exp(logitp)+1)*1/2/sig2*
exp(-1/2/sig2*age^2+age*mu2/sig2-
mu2^2/2/sig2)/pnorm(mu2/sqrt(sig2))
lik=ifelse(lik<10e-50, 10e-50,lik)
-sum(log(lik))
}

#################################################################
#ONE-PHASE MODEL WITH PRIOR ON ONSET IN PRODUCTION AND RECENT
#TERMINATION IN PRODUCTION#
#################################################################

OnePhaseTruncated=function(theta){
lambda1=exp(theta)
N=length(age)
first01=0
first02=(lambda1*(exp(-lambda1*offset)-
exp(-lambda1*onset)))/lambda1
first=(1/(first01+first02))
second=lambda1*exp(-lambda1*age)
prsecond=sum(log(second))
loglik=N*log(first)+prsecond
-(loglik)
}


#################################################################
#TWO-PHASE MODEL WITH PRIOR ON ONSET IN PRODUCTION AND RECENT
#TERMINATION IN PRODUCTION#
#################################################################
TwoPhaseTruncated=function(theta){
lambda1=exp(theta[1])
lambda2=exp(theta[2])
tau=exp(theta[3])
N=length(age)
first01=(tau*(exp(-lambda2*offset)-exp(-lambda2*onset)))/lambda2
first02=((lambda1-lambda2)*(exp(-(tau+lambda1)*offset)-exp(-
(tau+lambda1)*onset)))/(tau+lambda1)
first=(1/(first01+first02))
second=(tau*exp(-lambda2*age) + (lambda1-lambda2)*exp(-
(tau+lambda1)*age))
prsecond=sum(log(second))
loglik=N*log(first)+prsecond
if (lambda2>=(lambda1)) {loglik=-100000}
-(loglik)
}

#################################################################
#EXAMPLES WITH FOUR ASSEMBLAGES
```

```
#################################################################

######
#DATA#
######
NT30mage=c(5,2437,53,9,5,6,6,9,7,10,17,8703,1516,6,10316,2780,
573,12,695,16,2423,2518,1076,469,3270,5631,3836,14,
6299,1073,5308,4384,6225,5856,6,20,14,21,27,1836,3285,3548,3,
20,7540,18,14,20,17,18,282,4523,6,11334,2,6,5058,926,9557,21,3,
3662,3,3,2,2576,10,8,3118,2727,14,7,25,9,9,14,22,16,8,5)

NT4050mage=c(1,0,2414,0,1,0,1,3176,0,0,1,0,1,39,91,62,28,0,7,0,0,
0,1,82,3023,304,26,3,10,2117,1946,4,1524,151,3,2648,4500,1102,
3270,1477,3770,1850,44,130,1727,1374,701,2,1766,3,940,5173,7838,
6928,6,5080,5703,2,11903,10688,5228,7146,5875,8024,4348,8113)

NT5060mage=c(4637,3320,7,2793,39,7597,120,5,4968,9902,1454,5685,
5752,4885,10762,7708,5092,7412,5314,6518,3766,5864,8140,6652,
4693,10,4,4,8,136,4660,5666,9733,4348,8418,2167,7610,4778,7187,
7966,1465,7044,2972,5681,2761,2488,8684,1171,24,7421,9028,6014,
4843,9505,5799,10,2,1,7,4,11)

NT90mage=c(11100,12808,5676,21202,15108,18961,15454,18517,10292,
15074,15659,9626,15591,13052,12020,12606,1962,8341,9385,10771,
10763,10219,10159,11949,13666,13666,8656,11573,12204,18091,13,
10183,20595,13969,14278,17248,13486,5703,10706,8607,9941,9644,
13434,7541,17517,12739)

#########
#OBJECTS#
#########
samplesize=numeric(); obsmed=numeric(); obsIQR=numeric()
obsmedCI=array(NA,dim=c(4,2))
obsIQRCI=array(NA,dim=c(4,2))
likONEPHASETN=numeric();ONEPHASETNmean=numeric();
ONEPHASETNSD=numeric();like=numeric();onephaseAIC=numeric();
onephaserate=numeric();
weibullrate=numeric();weibullshape=numeric();
likweibull=numeric();weibullAIC=numeric();
twophaserate1=numeric();
twophaserate2=numeric();twophasealpha=numeric();
twophasebeta=numeric();liktwophase=numeric();
twophaseAIC=numeric();tau=numeric();
lambda1=numeric();lambda2=numeric();
TWOPHASETNvariance=numeric();TWOPHASETNmean1=numeric();
TWOPHASETNmean2=numeric();TWOPHASETNproportion=numeric();
onephaseratePRIOR=numeric();lambda1PRIOR=numeric();
lambda2PRIOR=numeric();tauPRIOR=numeric();
onephaseAICw=numeric();weibullAICw=numeric();
twophaseAICw=numeric()

expmed=array(NA,dim=c(4,5))
```

```
expIQR=array(NA,dim=c(4,5))
expmedCI=array(NA,dim=c(4,10))
expIQRCI=array(NA,dim=c(4,10))

#########################
#OBSERVED STATISTICS  #
#########################

for (j in 1:4) {
if (j==1) {age=NT30mage}
if (j==2) {age=NT4050mage}
if (j==3) {age=NT5060mage}
if (j==4) {age=NT90mage}

bootmedian<-numeric()
bootIQR<-numeric()
samplesize[j]=length(age)
obsmed[j]<-round(median(age),digits=1)
obsIQR[j]<-round(IQR(age),digits=0)
for (i in 1:5000) {
largesample<-sample(age,5000, replace=T)
bootmedian[i]=median(sample(largesample,length(age), replace=F))
bootIQR[i]=IQR(sample(largesample,length(age), replace=F))
}

obsmedCI[j,1:2]=quantile(bootmedian,c(0.025,0.975))
obsIQRCI[j,1:2]=quantile(bootIQR,c(0.025,0.975))

#add one to avoid zero ages when fitting data to density
functions
age=age+0.1

################################
#ONE-PHASE EXPONENTIAL MODEL    #
################################
lkexp=function(rho) {-length(age)*log(rho) + rho*sum(age)}
#95% credible interval according to Clark (2007), Statistical
#computation for environmental sciences in R
#a conjugate prior for the exponential likelihood is the gamma
#density
alpha=1; beta=1
rseq=seq(0.0001,0.5,length=1000)
prior=dgamma(rseq, alpha, beta)
lik=exp(-lkexp(rseq))
post=dgamma(rseq, (alpha+length(age)), (beta+sum(age)))
onephaserate[j]<-1/mean(age)
onephaserateLCI<-qgamma(c(0.025), (alpha+length(age)),
(beta+sum(age)))
onephaserateUCI<-qgamma(c(0.975), (alpha+length(age)),
(beta+sum(age)))
#negative loglikelihood for exponential
like[j]<--(sum(log(onephaserate[j])-onephaserate[j]*age))
```

```
AIC=2*1-(-2*like[j])
AICc=AIC+((2*1*(1+1))/(length(age)-1-1))
onephaseAIC[j]<-AICc

###################
#WEIBULL MODEL
###################
start1=c(log(0.0005),log(0.1))
fit1=optim(par=start1, fn=weibull)
weibullrate[j]<-exp(fit1$par[1])
weibullshape[j]<-exp(fit1$par[2])
likweibull[j]<-fit1$value
AIC=2*2-(-2*likweibull[j])
AICc=AIC+((2*2*(2+1))/(length(age)-2-1))
weibullAIC[j]<-AICc
#likelihood ratio test comparing weibull model with one-phase
#exponential
dev<-2*(like[j]-likweibull[j])
likratio<-1-pchisq(dev,1)

################################
#TWO-PHASE EXPONENTIAL MODEL    #
################################
out<-TwoPhase(age)
twophaserate1[j]<-out$r1
twophaserate2[j]<-out$r2
twophasealpha[j]<-out$alpha
twophasebeta[j]<-out$beta
lik<--(out$lik)
liktwophase[j]=lik
rAIC=2*3-(-2*lik)
twophaseAIC[j]=rAIC+((2*3*(3+1))/(length(age)-3-1))
tau[j]<-twophasealpha[j]*(twophaserate2[j]-twophaserate1[j])
lambda1[j]<-twophaserate1[j]*twophasealpha[j]
+twophaserate2[j]*(1-twophasealpha[j])
lambda2[j]<-twophaserate1[j]

#############
#AIC WEIGHTS#
#############
AICvector=c(onephaseAIC[j], weibullAIC[j], twophaseAIC[j])
minAIC=which.min(AICvector)
deltaAIC=AICvector-AICvector[minAIC]
onephaseAICw[j]=exp((-1/2)*deltaAIC[1])/sum(exp((-1/2)*deltaAIC))
weibullAICw[j]=exp((-1/2)*deltaAIC[2])/sum(exp((-1/2)*deltaAIC))
twophaseAICw[j]=exp((-1/2)*deltaAIC[3])/sum(exp((-1/2)*deltaAIC))

################################
#ONE-PHASE TRUNCATED NORMAL MODEL#
################################
out=OnePhaseTN(age)
ONEPHASETNmean[j]=out$mu
```

```
ONEPHASETNSD[j]=sqrt(out$sig2)

###################################
#TWO-PHASE TRUNCATED NORMAL MODEL #
###################################
#obtain initial estimates
MEAN=mean(age)
SD=sd(age)
start1=c(0.5,MEAN,MEAN-10,SD^2)
out=optim(start1,TwoPhaseTN)
theta=out$par
TWOPHASETNvariance[j]=theta[4];
TWOPHASETNmean1[j]=theta[2]
TWOPHASETNmean2[j]=theta[3]
TWOPHASETNproportion[j]=theta[1]
#logitp=log(theta[1]/(1-theta[1]))

#####################################################################
#PRIORS ON TIME OF PRODUCTION ONSET AND OFFSET - RECTANGULAR
#PRODUCTION TRAJECTORY#
#####################################################################
###################
offset=0   #Tmin e.g. here set to zero if production remains
constant until recent
onset=25000     #Tmax e.g. can be set to time of
transgression/flooding after the last glacial maximum

#These values can be compared with fits of one and two-phase
#model
out=optim(par=log(0.01), OnePhaseTruncated)
onephaseratePRIOR[j]=exp(out$par)

start1=log(c(0.001,0.00001,0.0001))
out=optim(par=start1, TwoPhaseTruncated)
lambda1PRIOR[j]=exp(out$par[1])
lambda2PRIOR[j]=exp(out$par[2])
tauPRIOR[j]=exp(out$par[3])

###################################
#PREDICT DENSITIES AND FREQUENCIES#
###################################
breaks=seq(0,max(age)+250, by=250)
hdata<-hist(age, breaks=breaks, plot=F)
observedfrequency=hdata$counts
times=seq(min(age),max(age),by=1)

ONEPHASE=dexp(times,rate=onephaserate[j])
C=((1/weibullrate[j])*gamma(1+1/weibullshape[j]))
WEIBULL=(1/C)*exp(-(weibullrate[j]*times)^weibullshape[j])
TWOPHASE=twophaserate1[j]*exp(-twophaserate1[j]*times)*
twophasebeta[j]+twophaserate2[j]*exp(-twophaserate2[j]*times)*
(1-twophasebeta[j])
```

```
ONEPHASETRUNCATED=dnorm(times,ONEPHASETNmean[j], ONEPHASETNSD[j])
TWOPHASETRUNCATED=TWOPHASETNproportion[j]*
dnorm(times,TWOPHASETNmean1[j],
sqrt(TWOPHASETNvariance[j]))+(1-TWOPHASETNproportion[j])*
dnorm(times,TWOPHASETNmean2[j],
sqrt(TWOPHASETNvariance[j]))

exp1phasefrequency<-hist(sample(times, 1000000, T,ONEPHASE),
breaks, plot=F)$counts
expweibfrequency<-hist(sample(times,1000000, T,WEIBULL), breaks,
plot=F)$counts
exp2phasefrequency<-hist(sample(times, 1000000, T,TWOPHASE),
breaks, plot=F)$counts
exp2phasetruncatedfrequency<-hist(sample(times,  1000000,
T,TWOPHASETRUNCATED), breaks, plot=F)$counts
exp1phasetruncatedfrequency<-hist(sample(times,  1000000,
T,ONEPHASETRUNCATED), breaks, plot=F)$counts

#G-TEST#
#the code for g test is at
#http://www.psych.ualberta.ca/~phurd/cruft/g.test.r
#expFreq = length(age) *
#exp1phasefrequency/sum(exp1phasefrequency)
#out=g.test(cbind(obsfrequency,expFreq), correct="none",
#parameter=1)

exp1phasemedian=numeric()
exp1phaseIQR=numeric()
exp2phasemedian=numeric()
exp2phaseIQR=numeric()
exp1phasetruncmedian=numeric()
exp1phasetruncIQR=numeric()
expweibmedian=numeric()
expweibIQR=numeric()
exp2phasetruncmedian=numeric()
exp2phasetruncIQR=numeric()

for (k in 1:1000) {
exp1phasemedian[k]<-
median(rexp(length(age),rate=onephaserate[j]))
exp1phaseIQR[k]<-IQR(rexp(length(age),rate=onephaserate[j]))
sampling<-sample(times, length(age), T, TWOPHASE)
exp2phasemedian[k]<-median(sampling)
exp2phaseIQR[k]<-IQR(sampling)
sampling<-sample(times, length(age), T, ONEPHASETRUNCATED)
exp1phasetruncmedian[k]=median(sampling)
exp1phasetruncIQR[k]=IQR(sampling)
sampling<-sample(times, length(age), T, WEIBULL)
expweibmedian[k]<-median(sampling)
expweibIQR[k]<-IQR(sampling)
sampling<-sample(times, length(age), T, TWOPHASETRUNCATED)
exp2phasetruncmedian[k]<-median(sampling)
```

```
exp2phasetruncIQR[k]<-IQR(sampling)
}

expmed[j,1:5]<-c(mean(exp1phasemedian),mean(expweibmedian),
mean(exp2phasemedian),mean(exp1phasetruncmedian),
mean(exp2phasetruncmedian))
expIQR[j,1:5]<-c(mean(exp1phaseIQR),mean(expweibIQR),
mean(exp2phaseIQR),mean(exp1phasetruncIQR),
mean(exp2phasetruncIQR))
expmedCI[j,1:10]<-c(quantile(exp1phasemedian,c(0.025,0.975)),
quantile(expweibmedian,c(0.025,0.975)),
quantile(exp2phasemedian,c(0.025,0.975)),
quantile(exp1phasetruncmedian,c(0.025,0.975)),
quantile(exp2phasetruncmedian,c(0.025,0.975)))
expIQRCI[j,1:10]<-c(quantile(exp1phaseIQR,c(0.025,0.975)),
quantile(expweibIQR,c(0.025,0.975)),
quantile(exp2phaseIQR,c(0.025,0.975)),
quantile(exp1phasetruncIQR,c(0.025,0.975)),
quantile(exp2phasetruncIQR,c(0.025,0.975)))
}

#######################################################
#CONFIDENCE INTERVALS FOR TWO-PHASE MODEL PARAMETERS#
#######################################################
#BOOTSTRAPPED CONFIDENCE INTERVALS
#######################################################
#here, just 50 resampling runs are repeated to reduce time
simulations=50
REStau=array(NA, dim=c(4, simulations))
RESlambda1=array(NA, dim=c(4, simulations))
RESlambda2=array(NA, dim=c(4, simulations))

for (i in 1:simulations) {
for (j in 1:4) {
if (j==1) {age=NT30mage}
if (j==2) {age=NT4050mage}
if (j==3) {age=NT5060mage}
if (j==4) {age=NT90mage}

age<-age+0.1
resampled=sample(age, length(age), replace=T)
X=resampled
out<-TwoPhase(X)
r1<-out$r1
r2<-out$r2
REStau[j,i]=out$tau
RESlambda1[j,i]=out$lambda1
RESlambda2[j,i]=r1
}
print(i)
}
```

```
lambda1LCI=apply(RESlambda1,MARGIN=1, quantile, c(0.025))
lambda2LCI=apply(RESlambda2,MARGIN=1, quantile, c(0.025))
tauLCI=apply(REStau,MARGIN=1, quantile, c(0.025))
lambda1UCI=apply(RESlambda1,MARGIN=1, quantile, c(0.975))
lambda2UCI=apply(RESlambda2,MARGIN=1, quantile, c(0.975))
tauUCI=apply(REStau,MARGIN=1, quantile, c(0.975))


#########################################################
#SUMMARIZING OUTPUTS
#########################################################
options(scipen=20)
Output=rbind(round(onephaserate,digits=6),
round(weibullrate,digits=1),
round(weibullshape,digits=2),
round(lambda1,digits=3),
round(lambda2,digits=6),
round(tau,digits=5),
round(ONEPHASETNmean,digits=0), round(ONEPHASETNSD,digits=0),
round(TWOPHASETNmean1,digits=0), round(TWOPHASETNmean2,digits=0),
round(sqrt(TWOPHASETNvariance),digits=0))
colnames(Output)=c("30 m","40-50 m","50-60 m","90 m")

########################################
#OBSERVED MEDIAN VERSUS PREDICTED MEDIAN#
########################################
par(cex=1.4)
par(mfrow=c(3,3))
par(mar=c(4,1,1,1))

par(pty="s")
plot(obsmed,expmed[,1], type="n",asp=1, pch=21, bg="gray51",
cex=1.3,xlim=c(1,30000), ylim=c(1,30000), main="",cex.main=0.9,
xlab="Observed median age (years)",
ylab="Expected median age (years)",log="xy")
lines(c(1,30000), c(1,30000))
for (i in 1:length(obsmed)) {
lines(c(obsmed[i],obsmed[i]),expmedCI[i,1:2], lty=2)
lines(c(obsmed[i],obsmed[i]),expmedCI[i,3:4], lty=2)
lines(c(obsmed[i],obsmed[i]),expmedCI[i,5:6], lty=2)
lines(c(obsmed[i],obsmed[i]),expmedCI[i,7:8], lty=2)
lines(c(obsmed[i],obsmed[i]),expmedCI[i,9:10], lty=2)
lines(obsmedCI[i,1:2],c(expmed[i,1],expmed[i,1]), lty=2)
lines(obsmedCI[i,1:2],c(expmed[i,2],expmed[i,2]), lty=2)
lines(obsmedCI[i,1:2],c(expmed[i,3],expmed[i,3]), lty=2)
lines(obsmedCI[i,1:2],c(expmed[i,4],expmed[i,4]), lty=2)
lines(obsmedCI[i,1:2],c(expmed[i,5],expmed[i,5]), lty=2)
}
points(obsmed,expmed[,1], pch=21, bg="white", cex=1.7, lwd=2)
points(obsmed,expmed[,2], pch=21, bg="gray51", cex=1.7, lwd=2)
points(obsmed,expmed[,3], pch=21,bg="black", cex=1.7, lwd=2)
points(obsmed,expmed[,4], pch=3, bg="black", cex=1.7, lwd=2)
```

```
points(obsmed,expmed[,5], pch=8, bg="black", cex=1.7, lwd=2)
legend(x="bottomright", legend=c("1-phase trunc.","2-phase
trunc."), pch=c(3,8),cex=1.1, bty="n", pt.lwd=c(2,2))



###################################
#OBSERVED IQR VERSUS PREDICTED IQR#
###################################

par(pty="s")
plot(obsIQR,expIQR[,1], type="n",asp=1, pch=16, cex=1.3,
xlim=c(500,20000), ylim=c(100,20000), main="",cex.main=0.9,
xlab="Observed IQR (years)",
ylab="Expected IQR (years)",log="xy")
lines(c(1,50000), c(1,50000))
for (i in 1:length(obsIQR)) {
lines(c(obsIQR[i],obsIQR[i]),expIQRCI[i,1:2], lty=2)
lines(c(obsIQR[i],obsIQR[i]),expIQRCI[i,3:4], lty=2)
lines(c(obsIQR[i],obsIQR[i]),expIQRCI[i,5:6], lty=2)
lines(c(obsIQR[i],obsIQR[i]),expIQRCI[i,7:8], lty=2)
lines(c(obsIQR[i],obsIQR[i]),expIQRCI[i,9:10], lty=2)
lines(obsIQRCI[i,1:2],c(expIQR[i,1],expIQR[i,1]), lty=2)
lines(obsIQRCI[i,1:2],c(expIQR[i,2],expIQR[i,2]), lty=2)
lines(obsIQRCI[i,1:2],c(expIQR[i,3],expIQR[i,3]), lty=2)
lines(obsIQRCI[i,1:2],c(expIQR[i,4],expIQR[i,4]), lty=2)
lines(obsIQRCI[i,1:2],c(expIQR[i,5],expIQR[i,5]), lty=2)
}
points(obsIQR,expIQR[,1], pch=21, bg="white", cex=1.7, lwd=2)
points(obsIQR,expIQR[,2], pch=21, bg="gray51", cex=1.7, lwd=2)
points(obsIQR,expIQR[,3], pch=21,bg="black", cex=1.7, lwd=2)
points(obsIQR,expIQR[,4], pch=3,bg="black", cex=1.7, lwd=2)
points(obsIQR,expIQR[,5], pch=8,bg="black", cex=1.7, lwd=2)
legend(x="topleft", pch=c(21,21,21),
pt.bg=c("white","gray51", "black"),
legend=c("1-phase","Weibull","2-phase"), cex=1.1, bty="n",
pt.lwd=c(2,2))


#################################################################
#LOSS-RATE PARAMETER ESTIMATES
#################################################################
options(scipen=20)

plot(c(1:4)-0.1,lambda1, ylim=c(0.00001,1), ylab="Rate", pch=16,
cex=1.5, log="y", xlim=c(0.5,4.5), xlab="", xaxt="n")
axis(1,at=1:4, labels=c("23-31 m","40-48 m","51-58 m","89 m"),
las=2)
for (i in 1:4) {
lines(c(i,i)-0.1,c(lambda1LCI[i],lambda1UCI[i]), lty=2)
lines(c(i,i),c(lambda2LCI[i],lambda2UCI[i]), lty=2)
lines(c(i,i)+0.1,c(tauLCI[i],tauUCI[i]), lty=2)
```

```
}
points(c(1:4),lambda2, pch=21, bg="gray",cex=1.5)
points(c(1:4)+0.1,tau, pch=21, bg="white",cex=1.5)
legend(x="bottomleft",pch=c(21,21,21), pt.cex=1.5,
pt.bg=c("black","gray","white"),legend=c("l1","l2","tau"),
bty="n",ncol=3)

#################################################################
#DISTRIBUTIONS
#################################################################

par(mfrow=c(4,4))
par(mar=c(1,2,2,1))

for (j in 1:4) {
if (j==1) {age=NT30mage;title="23-31 m"; maxylim=70}
if (j==2) {age=NT4050mage; title="40-48 m"; maxylim=50}
if (j==3) {age=NT5060mage;  title="51-58 m"; maxylim=20}
if (j==4) {age=NT90mage; title="89 m"; maxylim=15}

age=age+0.1
breaks=seq(0,22500, by=500)
hdata<-hist(age, breaks=breaks,  xlab="",
ylab="Number of individuals", col="white", freq=T, xaxt="n",
main=title, ylim=c(0,maxylim),
cex.main=1.3, cex.axis=1.3, cex.lab=1.5)
times=hdata$mids-hdata$mids[1]
ataxis=seq(times[1],times[length(times)],by=2500)
axis(1,at=ataxis, labels=ataxis/1000,las=1, tick=TRUE,
cex.axis=1.3)

#prediction for two phase model
predictRANDOM<-twophaserate1[j]*exp(-
twophaserate1[j]*times)*twophasebeta[j]+twophaserate2[j]*exp(-
twophaserate2[j]*times)*(1-twophasebeta[j])
if (j<4) {lines(times, predictRANDOM/sum(predictRANDOM)*
length(age), lwd=2, col="black", lty=1)}

#prediction for two phase truncated-normal model
MEANt2=TWOPHASETNmean2[j]
MEANt1=TWOPHASETNmean1[j]
proportion=TWOPHASETNproportion[j]
sdt=sqrt(TWOPHASETNvariance[j])
predictRANDOMTRUNCATED<-
(dnorm(times,mean=MEANt1,sd=sdt)*proportion)/
pnorm(MEANt1/sdt,0,1) + (dnorm(times,mean=MEANt2,sd=sdt)*(1-
proportion))/pnorm(MEANt2/sdt,0,1)
if (j>2)  {lines(times,
predictRANDOMTRUNCATED/sum(predictRANDOMTRUNCATED)* length(age),
col="gray31",lty=1, lwd=2)}
}
```

```
#####################################################################
#LIKELIHOOD SURFACE AND PROFILES
#####################################################################
#example
age=NT5060mage+0.1    #data

#WEIBULL
start1=c(log(0.0005),log(0.1))
fit1=optim(par=start1, fn=weibull)
weibullrate<-exp(fit1$par[1])
weibullshape<-exp(fit1$par[2])
likweibull<-fit1$value
lw<-weibullrate
cw<-weibullshape
C<-(1/weibullrate)*gamma(1+1/weibullshape)
lik<-(1/C)*exp(-(weibullrate*age)^weibullshape)
likw=-sum(log(lik))


#LIKELIHOOD SURFACE FOR WEIBULL
z<-matrix(NA, nrow=400, ncol=400)
rp1<-seq(0.01,80,length=nrow(z))
rp2<-seq(0.1,0.5,length=nrow(z))
for (i in 1:nrow(z)) {
      lp<-rp1[i]                  #scale
      for (j in 1:ncol(z)) {
            cp<-rp2[j]            #shape
            C<-(1/lp)*gamma(1+1/cp)
            lik<-(1/C)*exp(-(lp*age)^cp)
            lik=ifelse(lik<0.00000001,0.00000001,lik)
            z[i,j]<--sum(log(lik))
            }
      }

options(scipen=20)
par(cex=1.4)
par(oma=c(2,1,1,1))
par(mar=c(4,4,1,1))
par(mfrow=c(3,3))
contour(rp1, rp2,z, levels=round(seq(likw, likw+40,by=2)),
xlab="Weibull rate", ylab="Weibull shape", main="",log="x",
xlim=c(0.01,80), ylim=c(0.1,0.3), method="edge")
points(weibullrate, weibullshape, pch=21, bg="gray51", cex=1.4)

#TWO FUNCTIONS THAT OPTIMIZE THE LIKELIHOOD FOR ONE PARAMETER
#WITH RESPECT TO THE OTHER
#LIKELIHOOD PROFILE FOR SCALE PARAMETER (INVERSE OF RATE)
profscale<-function(cp) {
C<-(1/lp)*gamma(1+1/cp)
lik<-(1/C)*exp(-(lp*age)^cp)
```

```
lik=ifelse(lik<0.00000001,0.00000001,lik)
return(-sum(log(lik)))
}

#LIKELIHOOD PROFILE FOR SHAPE PARAMETER
profshape<-function(lp) {
C<-(1/lp)*gamma(1+1/cp)
lik<-(1/C)*exp(-(lp*age)^cp)
lik=ifelse(lik<0.00000001,0.00000001,lik)
return(-sum(log(lik)))
}

#LIKELIHOOD PROFILE
ll=numeric() #Likelihood profile for Weibull scale (1/rate)
lc=numeric() #Likelihood profile for Weibull shape
llam=numeric() #Likelihood profile for One-phase lambda
pl=numeric() #Chi-square probability for Weibull scale
pc=numeric() #Chi-square probability for Weibull shape
plam=numeric() #Chi-square probability for One-phase lambda

#LOOP OVER THE RANGE FROM 0.2 to 20 TIMES THE ML ESTIMATE OF EACH
#PARAMETER AND DETERMINE THE PROBABILITY FOR EACH VALUE USINGA LR
#TEST AGAINST THE ML ESTIMATE
for (i in 1:length(rp1)) {
#set the hypothesized value of scale
     lp<-rp1[i]      #lw is scale
     cp<-weibullshape
     lout<-optimize(profscale, interval=c(.001,1), maximum=F)
     ll[i]<-lout$objective
# given varying scale, where is the minimum -LnL for shape
     points(lp,lout$minimum,cex=0.6,pch=21, bg="gray")
     dl<-2*(ll[i] - likw)                  #LR test
#pl is chi-square probability for scale
     pl[i]<-1 - pchisq(dl,1)
#set the hypothesized value of shape
     lp<-weibullrate
     cp<-rp2[i]
     cout<-optimize(profshape, interval=c(.00001,50), maximum=F)
     lc[i]<-cout$objective
# given varying shape, where is the minimum -LnL for scale
     points(cout$minimum,cp, cex=0.6, pch=21, bg="gray")
#deviances for different shapes, keeping scale constant
     dc<-2*(lc[i] - likw)
#pc is chi-square probability
     pc[i]<- 1 - pchisq(dc,1)
}

abline(v=weibullrate, lty=2)
abline(h=weibullshape, lty=2)

#CONFIDENCE INTERVALS
multiply=(rp1/weibullrate)
```

```
plot(rp1, ll, type="l", xlim=c(min(rp1),max(rp1)),
xlab="Weibull rate", log="x", ylab="Negative log-likelihood",
ylim=c(min(lc),min(lc)+8))
abline(h=.025, lty=2)
l1<-findInterval(.025, pl[multiply<.9])
#interpolate between intersecting points
llo<-mean(rp1[l1:(l1+1)])
abline(v=llo, lty=2)
l2<-length(rp1[multiply < 1.1]) + findInterval(.975, (1-
pl[multiply > 1.1]))
lhi<-mean(rp1[l2:(l2+1)])
abline(v=lhi, lty=2)

#CONFIDENCE INTERVALS
multiply=(rp2/weibullshape)
plot(rp2,lc, xlim=c(min(rp2),max(rp2)), type="l",
xlab="Weibull shape", ylab="Negative log-likelihood",
ylim=c(min(lc),min(lc)+8))
abline(h=.025, lty=2)
c1<-findInterval(.025, pc[multiply<.9])
#interpolate between intersecting points
clo<-mean(rp2[c1:(c1+1)])
abline(v=clo, lty=2)

temp=1-pc[multiply > 1.1]
INDEX=which((round(temp,digits=3))%in%0.975)
#c2<-length(rp2[multiply<1.1]) +
#findInterval(.975, (1-pc[multiply > 1.1]))
c2<-length(rp2[multiply<1.1]) + INDEX

chi<-mean(rp2[c2:(c2+1)])
abline(v=chi, lty=2)


#################################################################
#LIKELIHOOD SURFACE FOR TWO-PHASE MODEL
#################################################################
out=TwoPhase(age)
rlik<--(out$lik)
r1<-out$r1
r2<-out$r2
alpha<-out$alpha
beta<-out$beta
tau<-alpha*(r2-r1)
lambda1<-r1*alpha+r2*(1-alpha)
lambda2<-r1


#LIKELIHOOD SURFACE
z<-matrix(NA, nrow=200, ncol=200)
rp1<-seq(0.01,0.4,length=nrow(z))
rp2<-seq(0.0002,0.004,length=nrow(z))
```

```
for (i in 1:nrow(z)) {
      Rlambda1<-lambda1*rp1[i]
      Rlambda1<-rp1[i]
      for (j in 1:ncol(z)) {
            Rtau<-tau*rp2[j]
            Rtau<-rp2[j]
            N=length(age)
            first=(1/((Rtau/lambda2) + (Rlambda1-lambda2)/
(Rtau+Rlambda1)))^(N)
            second=(Rtau*exp(-lambda2*age) + (Rlambda1-
lambda2)*exp(-(Rtau+Rlambda1)*age))
            prsecond=sum(log(second))
            loglik=log(first)+prsecond
            likr=-(loglik)
            z[i,j]<-likr
            }
print(i)
      }

options(scipen=20)
contour(rp1, rp2,z, levels=round(seq(rlik, max(z),by=2)),
xlab="2-phase lambda 1", ylab="2-phase tau",method="edge",
xlim=c(0.02,0.4), ylim=c(0.0002,0.004),
drawlabels=TRUE,labcex=0.75)
points(lambda1, tau, pch=21, bg="gray51", cex=1.4)

#LIKELIHOOD PROFILE
#SET ARRAYS
lla1=numeric()  #Likelihood profile for simulated lambda 1
lla2=numeric()  #Likelihood profile for simulated tau
pla1=numeric()  #Chi-square probability for simulated lambda 1
pla2=numeric()  #Chi-square probability for simulated tau


#TWO FUNCTIONS THAT OPTIMIZE THE LIKELIHOOD FOR ONE PARAMETER
#WITH RESPECT TO THE OTHER
#LIKELIHOOD PROFILE FOR LAMBDA1
proftau<-function(simlam1) {
lambda1=(simlam1)
lambda2=r1 #observed lambda 2
tau=(simtau)
N=length(age)
first=(1/((tau/lambda2) + (lambda1-lambda2)/(tau+lambda1)))^(N)
second=(tau*exp(-lambda2*age) + (lambda1-lambda2)*exp(-
(tau+lambda1)*age))
prsecond=sum(log(second))
loglik=log(first)+prsecond
support=-(loglik)
}

#LIKELIHOOD PROFILE FOR TAU
```

```
proflam1<-function(simtau) {
lambda1=(simlam1)
lambda2=r1 #observed lambda 2
tau=(simtau)
N=length(age)
first=(1/((tau/lambda2) + (lambda1-lambda2)/(tau+lambda1)))^(N)
second=(tau*exp(-lambda2*age) + (lambda1-lambda2)*exp(-
(tau+lambda1)*age))
prsecond=sum(log(second))
loglik=log(first)+prsecond
support=-(loglik)
}

#LOOP OVER THE RANGE FROM 0.2 to 20 TIMES THE ML ESTIMATE OF EACH
PARAMETER
#AND DETERMINE THE PROBABILITY FOR EACH VALUE USING A LIKELIHOOD-
RATIO TEST AGAINST THE ML ESTIMATE
for (i in 1:length(rp1)) {
#set the hypothesized value of lambda 1 and determine chi-square
#probabilities
      simlam1<-rp1[i]
      simtau=alpha*(r2-r1)  #observed tau
      lout<-optimize(proflam1,interval=c(0.0000001,1), maximum=F)
      lla1[i]<-lout$objective
      points(simlam1,lout$minimum, cex=0.5, pch=21, bg="gray")
      dla1<-2*(lla1[i] - rlik)
      pla1[i]<-1 - pchisq(dla1,1)
#set the hypothesized value of lambda 1 and determine chi-square
#probabilities
      simlam1<-r1*alpha+r2*(1-alpha) #observed lambda1
      simtau=rp2[i]
      cout<-optimize(proftau, interval=c(0.000005,1), maximum=F)
      lla2[i]<-cout$objective
      points(cout$minimum,simtau, cex=0.5, pch=21, bg="gray")
      dca2<-2*(lla2[i] - rlik)   #deviance
      pla2[i]<- 1 - pchisq(dca2,1)
}

abline(v=r1*alpha+r2*(1-alpha), lty=2)
abline(h=alpha*(r2-r1), lty=2)

#CONFIDENCE INTERVALS
multiply=(rp1/(r1*alpha+r2*(1-alpha)))
plot(rp1,lla1, xlim=c(0,max(rp1)), type="l", xlab="2-phase lambda
1",
ylab="Negative log-likelihood", ylim=c(min(lla1),min(lla1)+8))
abline(h=.025, lty=2)
c1<-findInterval(.025, pla1[multiply<.9])
clo<-mean(rp1[c1:(c1+1)])             #interpolate between
intersecting points
abline(v=clo, lty=2)
```

```
c2<-length(rp1[multiply<1.1]) + findInterval(.975, (1-
pla1[multiply > 1.1]))
chi<-mean(rp1[c2:(c2+1)])
abline(v=chi, lty=2)

multiply=(rp2/(alpha*(r2-r1) ))
plot(rp2,lla2, xlim=c(0,max(rp2)), type="l", xlab="2-phase tau",
ylab="Negative log-likelihood", ylim=c(min(lla1),min(lla1)+8))
abline(h=.025, lty=2)
c1<-findInterval(.025, pla2[multiply<.9])
clo<-mean(rp2[c1:(c1+1)])             #interpolate between
intersecting points
abline(v=clo, lty=2)
c2<-length(rp2[multiply<1.1]) + findInterval(.975, (1-
pla2[multiply > 1.1]))
chi<-mean(rp2[c2:(c2+1)])
abline(v=chi, lty=2)




#################################################################
#1-phase truncated-normal model VERSUS original production
#trajectory under varying standard deviation of the trajectory
#and different loss rates
#################################################################
par(mfrow=c(3,3))
par(mar=c(4,2,2,1))
lam=c(0.01,0.005,0.001,0.0005)
sdt=500
ut=5000
times=seq(1,10000,by=10)
times1=seq(1,10000,by=1)
LWD=c(2,2,2,2,2)
COL=c("black","black","gray51","gray71")
LTY=c(1,2,1,2,1)
MAXYLIM=2000
breaks=seq(0,10000,by=250)
SAMPLESIZE=10000
LOG=""

#########
sdt=1000
#########
plot(times,dnorm(times,mean=ut,sd=sdt)/max(dnorm(times,mean=ut,sd
=sdt)),log=LOG,ylim=c(1,MAXYLIM), type="n",
ylab="Probability",xlab="Time (y)",main="sd = 1000 y",
cex.main=0.9)
PROB=dnorm(times1,mean=ut,sd=sdt)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
```

```
lines(out2$mids,out2$counts, col="gray51",lwd=4)
for (i in 1:length(lam)) {
PROB=(dnorm(times1,mean=ut,sd=sdt)*exp(-
times1*lam[i]))/pnorm(ut/sdt,0,1)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
lines(out2$mids,out2$counts, col=COL[i],lwd=LWD[i],lty=LTY[i])
}

#################################################################
#check the performance of one-phase truncated-normal model with
#the simulated data
#################################################################
out=OnePhaseTN(out)
out$mu+(out$sig2)*lam[i]
sqrt(out$sig2)

#########
sdt=1500
#########
plot(times,dnorm(times,mean=ut,sd=sdt)/max(dnorm(times,mean=ut,sd
=sdt)),log=LOG, ylim=c(1,MAXYLIM),type="n",
ylab="Probability",xlab="Time (y)",main="sd = 2500 y",
cex.main=0.9)
PROB=dnorm(times1,mean=ut,sd=sdt)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
lines(out2$mids,out2$counts, col="gray51",lwd=4)
for (i in 1:length(lam)) {
PROB=(dnorm(times1,mean=ut,sd=sdt)*exp(-
times1*lam[i]))/pnorm(ut/sdt,0,1)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
lines(out2$mids,out2$counts, col=COL[i],lwd=LWD[i],lty=LTY[i])
}
#################################################################
#check the performance of one-phase truncated-normal model with
#the simulated data
#################################################################
out=OnePhaseTN(out)
out$mu+(out$sig2)*lam[i]
sqrt(out$sig2)

#########
sdt=2500
#########
plot(times,dnorm(times,mean=ut,sd=sdt)/max(dnorm(times,mean=ut,sd
=sdt)),log=LOG, ylim=c(1,MAXYLIM),type="n",
ylab="Probability",xlab="Time (y)",main="sd = 5000 y",
cex.main=0.9)
PROB=dnorm(times1,mean=ut,sd=sdt)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
```

```
out2=hist(out,breaks=breaks,plot=F)
lines(out2$mids,out2$counts, col="gray51",lwd=4)
for (i in 1:length(lam)) {
PROB=(dnorm(times1,mean=ut,sd=sdt)*exp(-
times1*lam[i]))/pnorm(ut/sdt,0,1)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
lines(out2$mids,out2$counts, col=COL[i],lwd=LWD[i],lty=LTY[i])
}
#################################################################
#check the performance of one-phase truncated-normal model with
#the simulated data
#################################################################
out=OnePhaseTN(out)
out$mu+(out$sig2)*lam[i]
sqrt(out$sig2)


#################################################################
#2-phase truncated-normal model VERSUS original production
#trajectory under varying standard deviation of the trajectory
#and different loss rates
#################################################################
breaks=seq(0,10000,by=250)
SAMPLESIZE=10000
ut=5000
times=seq(1,10000,by=10)
times1=seq(1,10000,by=1)
lam1=c(0.01,0.005,0.001,0.0005)
lam2=c(0.001,0.0005,0.0001,0.00005)
ta=rep(0.001,5)

#########
sdt=1000
#########
plot(times,dnorm(times,mean=ut,sd=sdt)/max(dnorm(times,mean=ut,sd
=sdt)), log=LOG,
ylim=c(1,MAXYLIM),type="n",ylab="Probability",xlab="Time
(y)",main="sd = 500 y", cex.main=0.9)
PROB=dnorm(times1,mean=ut,sd=sdt)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
for (i in (length(lam1)):1) {
MEANt1=ut-lam2[i]*(sdt^2)
MEANt2=ut-(lam1[i]+ta[i])*(sdt^2)
r1=lam2[i]
r2=lam1[i]+ta[i]
alpha=ta[i]/(ta[i]+lam1[i]-lam2[i])
beta=((1/r1)*alpha)/((1/r1)*alpha + (1/r2)*(1-alpha))
proportion=beta
normal<-(dnorm(times1,mean=MEANt1,sd=sdt)*proportion)/
pnorm(MEANt1/sdt,0,1) +
```

```
(dnorm(times1,mean=MEANt2,sd=sdt)*(1-proportion))/
pnorm(MEANt2/sdt,0,1)
normal=normal/max(normal)
PROB=normal
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
lines(out2$mids,out2$counts, col=COL[i],lwd=LWD[i],lty=LTY[i])
}

age=out
MEAN=mean(age)
SD=sd(age)
start1=c(0.5,MEAN,MEAN-10,SD^2)
out1=optim(start1,TwoPhaseTN)
theta=out1$par
sqrt(theta[4])   #variance
theta[2]+  #mean 1
theta[3]   #mean 2
theta[1]   #beta proportion


#########
sdt=1500
#########
plot(times,dnorm(times,mean=ut,sd=sdt)/max(dnorm(times,mean=ut,sd
=sdt)), log=LOG, ylim=c(1,MAXYLIM),type="n",
ylab="Probability",xlab="Time (y)",main="sd = 1000 y",
cex.main=0.9)
PROB=dnorm(times1,mean=ut,sd=sdt)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
for (i in length(lam1):1) {
MEANt1=ut-lam2[i]*(sdt^2)
MEANt2=ut-(lam1[i]+ta[i])*(sdt^2)
r1=lam2[i]
r2=lam1[i]+ta[i]
alpha=ta[i]/(ta[i]+lam1[i]-lam2[i])
beta=((1/r1)*alpha)/((1/r1)*alpha + (1/r2)*(1-alpha))
proportion=beta
normal<-(dnorm(times1,mean=MEANt1,sd=sdt)*proportion)/
pnorm(MEANt1/sdt,0,1) + (dnorm(times1,mean=MEANt2,sd=sdt)*(1-
proportion))/pnorm(MEANt2/sdt,0,1)
normal=normal/max(normal)
PROB=normal
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
lines(out2$mids,out2$counts, col=COL[i],lwd=LWD[i],lty=LTY[i])
}

#########
sdt=2500
#########
```

```
plot(times,dnorm(times,mean=ut,sd=sdt)/max(dnorm(times,mean=ut,sd
=sdt)), log=LOG, ylim=c(1,MAXYLIM),type="n",
ylab="Probability",xlab="Time (y)",main="sd = 2500 y",
cex.main=0.9)
PROB=dnorm(times1,mean=ut,sd=sdt)
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
for (i in length(lam1):1) {
MEANt1=ut-lam2[i]*(sdt^2)
MEANt2=ut-(lam1[i]+ta[i])*(sdt^2)
r1=lam2[i]
r2=lam1[i]+ta[i]
alpha=ta[i]/(ta[i]+lam1[i]-lam2[i])
beta=((1/r1)*alpha)/((1/r1)*alpha + (1/r2)*(1-alpha))
proportion=beta
normal<-(dnorm(times1,mean=MEANt1,sd=sdt)*proportion)/
pnorm(MEANt1/sdt,0,1) + (dnorm(times1,mean=MEANt2,sd=sdt)*(1-
proportion))/pnorm(MEANt2/sdt,0,1)
normal=normal/max(normal)
PROB=normal
out=sample(times1,SAMPLESIZE,replace=T,prob=PROB)
out2=hist(out,breaks=breaks,plot=F)
lines(out2$mids,out2$counts, col=COL[i],lwd=LWD[i],lty=LTY[i])
}



############################################################
#EFFECTS ON ESTIMATES OF LOSS UNDER SINUSOIDAL PRODUCTION
#AND SAMPLE SIZE EFFECTS
############################################################

#time in years for which prediction is made
times=seq(0,25000,by=1)
#range of sample sizes
size=c(10,20,30,40,50,100,500,1000)
#predetermined rates of loss
#one-phase model with lambda
ONErate=c(0.05,0.005,0.0005)
#two-phase model with lambda1,lambd2 and tau
TWOlambda1=ONErate
TWOlambda2=c(0.0001)
TWOtau=c(0.00025,0.0001)
PERIOD=c(1,20,50,100,250,500,1000)
#periods of fluctuations in years
PERIOD=c(1,100,250,500,1000)
FREQUENCY=2*pi/PERIOD
#number of simulations (set to a small number here)
sims=5
```

```
fit1=array(NA,
dim=c(length(size),length(ONErate),length(PERIOD),sims))
fit2tau=array(NA, dim=c(length(size),length(ONErate),
length(TWOtau),length(PERIOD),sims))
fit2lambda2=array(NA, dim=c(length(size),length(ONErate),
length(TWOtau),length(PERIOD),sims))
fit2lambda1=array(NA, dim=c(length(size),length(ONErate),
length(TWOtau),length(PERIOD),sims))

#############
#SIMULATIONS#
#############
#if offset is at 900, modern population is at minimum
#if offset is at 300, modern population is at maximum
OFFSET=900
for (k in 1:sims) {
for (n in 1:length(FREQUENCY)) {
#ES defines the trajectory in production
ES<-sin(times*(FREQUENCY[n])-OFFSET)
#for variable production
ES=(ES+1)/2
#for constant production
if (n==1) {ES=1}
#loop for sample size values
for (i in 1:length(size)) {
#loop for lambda and lambda1  values
for (j in 1:length(ONErate)) {
###############
#1-PHASE MODEL#
###############
predictSIMPLE<-exp(-ONErate[j]*times)
age<-sample(times, size[i], replace = T, prob = predictSIMPLE*ES)
hist(age, col="gray", main="One-phase model", xlab="Postmortem
age")
fit1[i,j,n,k]=1/mean(age)
###############
#2-PHASE MODEL#
###############
#loop for tau values
for (l in 1:length(TWOtau)) {
Ralpha=TWOtau[l]/(TWOtau[l]+TWOlambda1[j]-TWOlambda2)
Rrate2=TWOlambda1[j]+TWOtau[l]
Rrate1=TWOlambda2
predictRANDOM<-exp(-Rrate1*times)*Ralpha+exp(-Rrate2*times)*(1-
Ralpha)
age=sample(times, size[i], replace = T, prob = predictRANDOM*ES)
hist(age, col="gray", main="Two-phase model", xlab="Postmortem
age")
X=age
start1=log(c(0.0001,0.0001,0.0001))
out<-TwoPhase(X)
fit2tau[i,j,l,n,k]=out$alpha*(out$r2-out$r1)
```

```
fit2lambda2[i,j,l,n,k]=out$r1
fit2lambda1[i,j,l,n,k]=out$r2-out$tau
}
}     #loop for j lambda
}     #loop for i sample size
print(paste("n =",n))
}     #loop for n FREQUENCY
print(paste("k =",k))
}

#################################################################
#FIGURE - EFFECTS OF SAMPLE SIZE AND FLUCTATING PRODUCTION
#################################################################
leg=c("true lam1=0.05","true lam1=0.005","true lam1=0.0005")
options(scipen=20)
par(cex=1.4)
par(mfrow=c(3,4))
par(mar=c(4,2,1,0))
#############################################
#EFFECT OF SAMPLE SIZE ON ONE-PHASE LAMBDA  #
#############################################
PER=1
LWD=3
SUMMARY="median"
plot(size,apply(fit1[,1,PER,],1,SUMMARY), ylab="One-phase
lambda", xlab="Sample size", type="l", lwd=LWD,
ylim=c(0.00001,0.5), log="xy",yaxt="n",
main="", cex.main=1, cex.axis=1.1, cex.lab=1.1, xlim=c(10,1000))
lines(size,apply(fit1[,1,PER,],1,quantile, 0.025), lty=1)
lines(size,apply(fit1[,1,PER,],1,quantile, 0.975), lty=1)
lines(size,apply(fit1[,2,PER,],1,SUMMARY),col="gray25",lwd=LWD,
lty=2)
lines(size,apply(fit1[,2,PER,],1,quantile, 0.025), lty=2)
lines(size,apply(fit1[,2,PER,],1,quantile, 0.975), lty=2)
lines(size,apply(fit1[,3,PER,],1,SUMMARY),col="gray51",lwd=LWD,
lty=3)
lines(size,apply(fit1[,3,PER,],1,quantile, 0.025), lty=3)
lines(size,apply(fit1[,3,PER,],1,quantile, 0.975), lty=3)
axis(2,at=ONErate, labels=c(ONErate), cex.axis=1.1)
axisrates=c(log(2)/c(10,100,1000,10000))

###################################
#EFFECT OF SAMPLE SIZE ON LAMBDA 1#
###################################
TAU=1
plot(size, apply(fit2lambda1[,1,TAU,PER,],1,SUMMARY),
ylab="Estimated lambda 1 (t=0.0001)",xlab="Sample
size",type="l",ylim=c(0.00001,0.5), log="xy",
yaxt="n", lwd=LWD,main="", cex.main=1, cex.axis=1.1, cex.lab=1.1)
lines(size,apply(fit2lambda1[,1,TAU,PER,],1,quantile, 0.025))
lines(size,apply(fit2lambda1[,1,TAU,PER,],1,quantile, 0.975))
```

```
lines(size,apply(fit2lambda1[,2,TAU,PER,],1,SUMMARY), lwd=LWD,
lty=2, col="gray31")
lines(size,apply(fit2lambda1[,2,TAU,PER,],1,quantile, 0.025),
lty=2, col="black")
lines(size,apply(fit2lambda1[,2,TAU,PER,],1,quantile, 0.975),
lty=2, col="black")
lines(size,apply(fit2lambda1[,3,TAU,PER,],1,SUMMARY), lwd=LWD,
lty=3, col="gray51")
lines(size,apply(fit2lambda1[,3,TAU,PER,],1,quantile, 0.025),
lty=3, col="black")
lines(size,apply(fit2lambda1[,3,TAU,PER,],1,quantile, 0.975),
lty=3, col="black")
axis(2,at = TWOlambda1, labels = TWOlambda1, cex.axis=1.1)
axisrates=log(2)/c(10,100,1000,10000)

TAU=2
plot(size, apply(fit2lambda1[,1,TAU,PER,],1,SUMMARY),
ylab="Estimated lambda 1 (t=0.00001)",xlab="Sample
size",type="l",ylim=c(0.00001,0.5), log="xy",
yaxt="n", lwd=LWD,main="", cex.main=1, cex.axis=1.1, cex.lab=1.1)
lines(size,apply(fit2lambda1[,1,TAU,PER,],1,quantile, 0.025))
lines(size,apply(fit2lambda1[,1,TAU,PER,],1,quantile, 0.975))
lines(size,apply(fit2lambda1[,2,TAU,PER,],1,SUMMARY), lwd=LWD,
lty=2, col="gray31")
lines(size,apply(fit2lambda1[,2,TAU,PER,],1,quantile, 0.025),
lty=2, col="black")
lines(size,apply(fit2lambda1[,2,TAU,PER,],1,quantile, 0.975),
lty=2, col="black")
lines(size,apply(fit2lambda1[,3,TAU,PER,],1,SUMMARY), lwd=LWD,
lty=3, col="gray51")
lines(size,apply(fit2lambda1[,3,TAU,PER,],1,quantile, 0.025),
lty=3, col="black")
lines(size,apply(fit2lambda1[,3,TAU,PER,],1,quantile, 0.975),
lty=3, col="black")
axis(2,at = TWOlambda1, labels = TWOlambda1, cex.axis=1.1)
axisrates=log(2)/c(10,100,1000,10000)

###################################
#EFFECT OF SAMPLE SIZE ON LAMBDA 2#
###################################
TAU=2
plot(size,apply(fit2lambda2[,1,TAU,PER,],1,SUMMARY), lwd=LWD,
ylab="Estimated lambda 2 (tau=0.0001)",
xlab="Sample size",type="l",ylim=c(0.00001,0.5), log="xy",
main="", cex.main=1, cex.lab=1.1, cex.axis=1.1)
lines(size,apply(fit2lambda2[,1,TAU,PER,],1,SUMMARY), lty=1)
lines(size,apply(fit2lambda2[,1,TAU,PER,],1,quantile, 0.025))
lines(size,apply(fit2lambda2[,1,TAU,PER,],1,quantile, 0.975))
lines(size,apply(fit2lambda2[,2,TAU,PER,],1,SUMMARY), lwd=LWD,
lty=2, col="gray25")
lines(size,apply(fit2lambda2[,2,TAU,PER,],1,quantile, 0.025),
lty=2, col="black")
```

```
lines(size,apply(fit2lambda2[,2,TAU,PER,],1,quantile, 0.975),
lty=2, col="black")
lines(size,apply(fit2lambda2[,3,TAU,PER,],1,SUMMARY), lwd=LWD,
lty=3, col="gray51")
lines(size,apply(fit2lambda2[,3,TAU,PER,],1,quantile, 0.025),
lty=3, col="black")
lines(size,apply(fit2lambda2[,3,TAU,PER,],1,quantile, 0.975),
lty=3, col="black")
axisrates=log(2)/c(5,50,500,5000)

##########################################
#EFFECT OF FLUCTUATION ON ONE-PHASE LAMBDA #
##########################################
CEXAXIS=0.85
SIZE=4
LWD=3
plot(PERIOD,apply(fit1[SIZE,1,,],1,mean), ylab="Estimated one-
phase lambda",
xlab="Period of fluctuation (y)", type="l",lwd=LWD,
ylim=c(0.00001,0.25), log="xy",yaxt="n",
main="", cex.main=1, cex.axis=1.1, cex.lab=1.1, xlim=c(1,1000))
lines(PERIOD,apply(fit1[SIZE,1,,],1,quantile, 0.025), lty=1)
lines(PERIOD,apply(fit1[SIZE,1,,],1,quantile, 0.975), lty=1)
lines(PERIOD,apply(fit1[SIZE,2,,],1,mean),col="gray25",
lwd=LWD,lty=2)
lines(PERIOD,apply(fit1[SIZE,2,,],1,quantile, 0.025), lty=2)
lines(PERIOD,apply(fit1[SIZE,2,,],1,quantile, 0.975), lty=2)
lines(PERIOD,apply(fit1[SIZE,3,,],1,mean),col="gray51",
lwd=LWD,lty=3)
lines(PERIOD,apply(fit1[SIZE,3,,],1,quantile, 0.025), lty=3)
lines(PERIOD,apply(fit1[SIZE,3,,],1,quantile, 0.975), lty=3)
axis(2,at=c(0.0001,ONErate), labels=c(0.0001,ONErate),
cex.axis=CEXAXIS)
axisrates=c(log(2)/c(10,100,1000,10000))


###################################
#EFFECT OF FLUCTUATION ON LAMBDA 1#
###################################
CEXAXIS=0.85
PER=1
TAU=1
plot(PERIOD, apply(fit2lambda1[SIZE,1,TAU,,],1,SUMMARY),
ylab="Estimated lambda 1 (tau=0.0001)",xlab="Period of
fluctuation (y)",type="l",
ylim=c(0.00001,0.25), log="xy",
yaxt="n", lwd=LWD,main="", cex.main=1, cex.axis=1.1, cex.lab=1.1,
xlim=c(1,1000))
lines(PERIOD,apply(fit2lambda1[SIZE,1,TAU,,],1,quantile, 0.025))
lines(PERIOD,apply(fit2lambda1[SIZE,1,TAU,,],1,quantile, 0.975))
lines(PERIOD,apply(fit2lambda1[SIZE,2,TAU,,],1,SUMMARY), lwd=LWD,
lty=2, col="gray31")
```

```
lines(PERIOD,apply(fit2lambda1[SIZE,2,TAU,,],1,quantile, 0.025),
lty=2, col="black")
lines(PERIOD,apply(fit2lambda1[SIZE,2,TAU,,],1,quantile, 0.975),
lty=2, col="black")
lines(PERIOD,apply(fit2lambda1[SIZE,3,TAU,,],1,SUMMARY), lwd=LWD,
lty=3, col="gray51")
lines(PERIOD,apply(fit2lambda1[SIZE,3,TAU,,],1,quantile, 0.025),
lty=3, col="black")
lines(PERIOD,apply(fit2lambda1[SIZE,3,TAU,,],1,quantile, 0.975),
lty=3, col="black")
axis(2,at=c(0.0001,ONErate), labels=c(0.0001,ONErate),
cex.axis=CEXAXIS)
axisrates=log(2)/c(10,100,1000,10000)

TAU=2
plot(PERIOD, apply(fit2lambda1[SIZE,1,TAU,,],1,SUMMARY),
ylab="Estimated lambda (tau=0.00001)",
xlab="Period of fluctuation (y)",type="l",ylim=c(0.00001,0.5),
log="xy",
yaxt="n", lwd=LWD,main="", cex.main=1, cex.axis=1.1, cex.lab=1.1,
xlim=c(1,1000))
lines(PERIOD,apply(fit2lambda1[SIZE,1,TAU,,],1,quantile, 0.025))
lines(PERIOD,apply(fit2lambda1[SIZE,1,TAU,,],1,quantile, 0.975))
lines(PERIOD,apply(fit2lambda1[SIZE,2,TAU,,],1,SUMMARY), lwd=LWD,
lty=2, col="gray25")
lines(PERIOD,apply(fit2lambda1[SIZE,2,TAU,,],1,quantile, 0.025),
lty=2, col="black")
lines(PERIOD,apply(fit2lambda1[SIZE,2,TAU,,],1,quantile, 0.975),
lty=2, col="black")
lines(PERIOD,apply(fit2lambda1[SIZE,3,TAU,,],1,SUMMARY), lwd=LWD,
lty=3, col="gray51")
lines(PERIOD,apply(fit2lambda1[SIZE,3,TAU,,],1,quantile, 0.025),
lty=3, col="black")
lines(PERIOD,apply(fit2lambda1[SIZE,3,TAU,,],1,quantile, 0.975),
lty=3, col="black")
axis(2,at=c(0.0001,ONErate), labels=c(0.0001,ONErate),
cex.axis=CEXAXIS)
axisrates=log(2)/c(10,100,1000,10000)

####################################
#EFFECT OF FLUCTUATIONS ON LAMBDA 2#
####################################
TAU=1
SIZE=5
plot(PERIOD, apply(fit2lambda2[SIZE,1,TAU,,],1,SUMMARY), lwd=LWD,
ylab="Estimated lambda 2", xlab="Period of fluctuation
(y)",type="l",ylim=c(0.00001,0.5), log="xy", yaxt="n",main="",
cex.main=1, cex.axis=1.1, cex.lab=1.1, xlim=c(1,1000))
lines(PERIOD,apply(fit2lambda2[SIZE,1,TAU,,],1,SUMMARY), lwd=LWD)
lines(PERIOD,apply(fit2lambda2[SIZE,1,TAU,,],1,quantile, 0.025),
lty=2)
```

```
lines(PERIOD,apply(fit2lambda2[SIZE,1,TAU,,],1,quantile, 0.975),
lty=2)
lines(PERIOD,apply(fit2lambda2[SIZE,2,TAU,,],1,SUMMARY), lwd=LWD,
lty=2, col="gray25")
lines(PERIOD,apply(fit2lambda2[SIZE,2,TAU,,],1,quantile, 0.025),
lty=2)
lines(PERIOD,apply(fit2lambda2[SIZE,2,TAU,,],1,quantile, 0.975),
lty=2)
lines(PERIOD,apply(fit2lambda2[SIZE,3,TAU,,],1,SUMMARY), lwd=LWD,
lty=3, col="gray51")
lines(PERIOD,apply(fit2lambda2[SIZE,3,TAU,,],1,quantile, 0.025),
lty=3)
lines(PERIOD,apply(fit2lambda2[SIZE,3,TAU,,],1,quantile, 0.975),
lty=3)
axis(2,at=c(0.0001,ONErate), labels=c(0.0001,ONErate),
cex.axis=CEXAXIS)
axisrates=log(2)/c(10,100,1000,10000)
```