Appendix: R codes

```
library(actuar)
library(gendist)
library(SMPracticals)
library(DescTools)


nlogl.mixt <- function(p, phi, spec1, arg1, spec2, arg2){
tt <- 1.0e20

if(all(p>0)){
tt <- -sum(log( dmixt(x, phi, spec1, arg1, spec2, arg2) ))
}
return(tt)
}



############################################################
# Fitting of inverse transformed gamma-transformed beta #
# models to complete and incomplete data                 #
#
############################################################

x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="trbeta",
arg2=c(shape1=p[5],shape2=p[6],shape3=p[7],scale=p[8]))}
fun = optim(fn=ff,par=c(1,1,19,1,1,5,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*8,digits=3,format="f")," & ",
formatC(2*fun$value+8*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="trbeta", arg2=c(shape1=p[5],shape2=p[6],shape3=p[7],scale=p[8]))}
p = optim(fn=ff,par=c(1,1,19,1,1,5,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="trbeta", arg2=c(shape1=p[5],shape2=p[6],shape3=p[7],scale=p[8]))
cat(formatC(ll,digits=3,format="f")," & ",
```

```
formatC(2*ll+2*8,digits=3,format="f")," & ",
formatC(2*ll+8*log(length(x)),digits=3,format="f"),"\\\\","\n")
#


x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",
arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
fun = optim(fn=ff,par=c(1,1,20,1,1,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*7,digits=3,format="f")," & ",
formatC(2*fun$value+7*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",
arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
p=optim(fn=ff,par=c(1,1,20,1,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",
arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*7,digits=3,format="f")," & ",
formatC(2*ll+7*log(length(x)),digits=3,format="f"),"\\\\","\n")
xx=min(x)+(max(x)-min(x))*seq(0,1,0.01)
d1=dmixt(xx, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))
p1=function (yy) {pmixt(yy, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
ks.test(x,p1)
AndersonDarlingTest(x,p1)
q1=qmixt(0.99, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))
c1=mean(x[x>q1])
#
```

```
x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invburr", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
fun = optim(fn=ff,par=c(1,1,1,1,1,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*7,digits=3,format="f")," & ",
formatC(2*fun$value+7*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invburr", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
p=optim(fn=ff,par=c(1,1,1,1,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invburr", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*7,digits=3,format="f")," & ",
formatC(2*ll+7*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
```

```
x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="genpareto", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
fun = optim(fn=ff,par=c(1,1,15,1,10,10,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*7,digits=3,format="f")," & ",
formatC(2*fun$value+7*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="genpareto", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
p=optim(fn=ff,par=c(1,1,15,1,10,10,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="genpareto", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))
cat(formatC(ll,digits=3,format="f")," & ",
```

```
formatC(2*ll+2*7,digits=3,format="f")," & ",
formatC(2*ll+7*log(length(x)),digits=3,format="f"),"\\\\","\n")
#


x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="pareto", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="pareto", arg2=c(shape=p[5],scale=p[6]))}
p=optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="pareto", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#




x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invpareto", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(0.01,1,20,1,10,0.03),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invpareto", arg2=c(shape=p[5],scale=p[6]))}
p=optim(fn=ff,par=c(0.01,1,20,1,10,0.03),
```

```
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invpareto", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#



x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="paralogis", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,10,1,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="paralogis", arg2=c(shape=p[5],scale=p[6]))}
p=optim(fn=ff,par=c(1,1,10,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="paralogis", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
d3=dmixt(xx, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="paralogis",arg2=c(shape=p[5],scale=p[6]))
p3=function (yy) {pmixt(yy, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="paralogis",arg2=c(shape=p[5],scale=p[6]))}
ks.test(x,p3)
AndersonDarlingTest(x,p3)
q3=qmixt(0.99, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="paralogis",arg2=c(shape=p[5],scale=p[6]))
c3=mean(x[x>q3])
#
```

```
x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,20,1,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis", arg2=c(shape=p[5],scale=p[6]))}
p=optim(fn=ff,par=c(1,1,20,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
d2=dmixt(xx, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis",arg2=c(shape=p[5],scale=p[6]))
p2=function (yy) {pmixt(yy, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis",arg2=c(shape=p[5],scale=p[6]))}
ks.test(x,p2)
AndersonDarlingTest(x,p2)
q2=qmixt(0.99, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis",arg2=c(shape=p[5],scale=p[6]))
c2=mean(x[x>q2])
#
```

```
x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="llogis", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,1,1,1,1),
```

```
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="llogis", arg2=c(shape=p[5],scale=p[6]))}

p=optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="llogis", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")


par(mfrow=c(1,1),mai=c(0.8,0.8,0,0))
hist(x,nclass=200,freq=FALSE,xlab="Data",
ylab="Histogram and Fitted PDFs",main="",xlim=c(0,60),ylim=c(0,0.2))
par(new=TRUE)
plot(xx,d1,xlab="",ylab="",xlim=c(0,60),
ylim=c(0,0.2),type="l",col="red")
par(new=TRUE)
plot(xx,d2,xlab="",ylab="",xlim=c(0,60),
ylim=c(0,0.2),type="l",col="blue")
par(new=TRUE)
plot(xx,d3,xlab="",ylab="",xlim=c(0,60),
ylim=c(0,0.2),type="l",col="black")







##########################################################
# Fitting of inverse transformed gamma-transformed gamma#
# models to complete and incomplete data                #
#
##########################################################


x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
```

```
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="trgamma", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
fun = optim(fn=ff,par=c(1,1,19,1,1,5,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*8,digits=3,format="f")," & ",
formatC(2*fun$value+8*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="trgamma", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
p=optim(fn=ff,par=c(1,1,19,1,1,5,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="trgamma", arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*8,digits=3,format="f")," & ",
formatC(2*ll+8*log(length(x)),digits=3,format="f"),"\\\\","\n")
#


x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invtrgamma",
arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
fun = optim(fn=ff,par=c(1,1,20,1,1,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*7,digits=3,format="f")," & ",
formatC(2*fun$value+7*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invtrgamma",
arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
p=optim(fn=ff,par=c(1,1,20,1,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invtrgamma",
arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))
cat(formatC(ll,digits=3,format="f")," & ",
```

```
formatC(2*ll+2*7,digits=3,format="f")," & ",
formatC(2*ll+7*log(length(x)),digits=3,format="f"),"\\\\","\n")
#



x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="gamma", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="gamma", arg2=c(shape=p[5],scale=p[6]))}
p=optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="gamma", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#



x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invgamma", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,15,1,10,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invgamma", arg2=c(shape=p[5],scale=p[6]))}
p=optim(fn=ff,par=c(1,1,15,1,10,1),
```

```
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invgamma", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#


x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="weibull", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="weibull", arg2=c(shape=p[5],scale=p[6]))}
p=optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="weibull", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#




x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invweibull", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(0.01,1,20,1,10,0.03),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*6,digits=3,format="f")," & ",
formatC(2*fun$value+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
```

```
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invweibull", arg2=c(shape=p[5],scale=p[6]))}
p=optim(fn=ff,par=c(0.01,1,20,1,10,0.03),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invweibull", arg2=c(shape=p[5],scale=p[6]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*6,digits=3,format="f")," & ",
formatC(2*ll+6*log(length(x)),digits=3,format="f"),"\\\\","\n")
#



x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="exp", arg2=c(rate=p[5]))}
fun = optim(fn=ff,par=c(1,1,10,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*5,digits=3,format="f")," & ",
formatC(2*fun$value+5*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="exp", arg2=c(rate=p[5]))}
p=optim(fn=ff,par=c(1,1,10,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="exp", arg2=c(rate=p[5]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*5,digits=3,format="f")," & ",
formatC(2*ll+5*log(length(x)),digits=3,format="f"),"\\\\","\n")
#



x = sort(danish)[1:2492]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
```

```
spec2="invexp", arg2=c(scale=p[5]))}
fun = optim(fn=ff,par=c(1,1,20,1,1),
control=list(maxit=100000))
cat(formatC(fun$value,digits=3,format="f")," & ",
formatC(2*fun$value+2*5,digits=3,format="f")," & ",
formatC(2*fun$value+5*log(length(x)),digits=3,format="f"),"\\\\","\n")
#
x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invexp", arg2=c(scale=p[5]))}
p=optim(fn=ff,par=c(1,1,20,1,1),
control=list(maxit=100000))$par
x = sort(danish)[1:2492]
ll=nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invexp", arg2=c(scale=p[5]))
cat(formatC(ll,digits=3,format="f")," & ",
formatC(2*ll+2*5,digits=3,format="f")," & ",
formatC(2*ll+5*log(length(x)),digits=3,format="f"),"\\\\","\n")
#


##############################################
# Sensitivity analysis for the ITG-B model  #
##############################################

code = data.frame(matrix(,1,13))
fse = data.frame(matrix(,7,13))

for (i in seq(-6,0,by=0.5)){
x = danish[log(danish[1:2492]/max(danish[1:2492]))<=i]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
fun = optim(fn=ff,par=c(1,1,19,1,1,5,1),
control=list(maxit=100000),hessian=TRUE)
code[,13+i/0.5] = fun$convergence

hess = fun$hessian
se = 1/hess^0.5
diag = diag(se)
fse[,13+i/0.5] = diag
}

par(mfrow=c(4,2),oma=rep(0,4),mai=c(0.6,0.6,0,0))
plot(seq(-6,0,by=0.5), as.numeric(fse[1,]), xlab="d",
ylab="Standard error",type="l")
```

```
plot(seq(-6,0,by=0.5), as.numeric(fse[2,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[3,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[4,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[5,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[6,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[7,]), xlab="d",
ylab="Standard error",type="l")



##############################################
# Sensitivity analysis for the ITG-PL model #
##############################################

code = data.frame(matrix(,1,13))
fse = data.frame(matrix(,6,13))

for (i in seq(-6,0,by=0.5)){
x = danish[log(danish[1:2492]/max(danish[1:2492]))<=i]
ff=function(p){nlogl.mixt(p, phi=p[1],
spec1="invtrgamma", arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="paralogis",arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,20,1,1,1),
control=list(maxit=100000),hessian=TRUE)
code[,13+i/0.5] = fun$convergence

hess = fun$hessian
se = 1/hess^0.5
diag = diag(se)
fse[,13+i/0.5] = diag
}

par(mfrow=c(3,2),oma=rep(0,4),mai=c(0.6,0.6,0,0))
plot(seq(-6,0,by=0.5), as.numeric(fse[1,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[2,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[3,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[4,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[5,]), xlab="d",
ylab="Standard error",type="l")
```

```
plot(seq(-6,0,by=0.5), as.numeric(fse[6,]), xlab="d",
ylab="Standard error",type="l")



##############################################
# Sensitivity analysis for the ITG-IPL model #
##############################################

code = data.frame(matrix(,1,13))
fse = data.frame(matrix(,6,13))

for (i in seq(-6,0,by=0.5)){
x = danish[log(danish[1:2492]/max(danish[1:2492]))<=i]
ff=function(p){nlogl.mixt(p, phi=p[1],
spec1="invtrgamma", arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000),hessian=TRUE)
code[,13+i/0.5] = fun$convergence

hess = fun$hessian
se = 1/hess^0.5
diag = diag(se)
fse[,13+i/0.5] = diag
}

par(mfrow=c(3,2),oma=rep(0,4),mai=c(0.6,0.6,0,0))
plot(seq(-6,0,by=0.5), as.numeric(fse[1,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[2,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[3,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[4,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[5,]), xlab="d",
ylab="Standard error",type="l")
plot(seq(-6,0,by=0.5), as.numeric(fse[6,]), xlab="d",
ylab="Standard error",type="l")


t1=rep(0,100)
t2=t1
t3=t1


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

```
# Simulation for the ITG-B model #
###################################


x=danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1], spec1="invtrgamma",
arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
par1=optim(fn=ff,par=c(1,1,19,1,1,5,1),
control=list(maxit=100000))$par
est = data.frame(matrix(,7,100))
fse = data.frame(matrix(,7,100))

for(i in 1:100){
cat(i,"\n")
tt=proc.time()
r = runif(2489)
x = qmixt(r, phi=par1[1], spec1="invtrgamma",
arg1=c(shape1=par1[2],shape2=par1[3],scale=par1[4]),
spec2="burr",
arg2=c(shape1=par1[5],shape2=par1[6],scale=par1[7]),
interval=c(0,10000))
ff=function(p){nlogl.mixt(p, phi=p[1],
spec1="invtrgamma", arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="burr",arg2=c(shape1=p[5],shape2=p[6],scale=p[7]))}
fun = optim(fn=ff,par=c(1,1,19,1,1,5,1),
control=list(maxit=100000000),hessian=TRUE)
tt=proc.time()-tt
t1[i]=tt[3]
est[,i] = fun$par
hess = fun$hessian
se = 1/hess^0.5
diag = diag(se)
fse[,i] = diag
}


par(mfrow=c(7,2),oma=rep(0,4),mai=c(0.6,0.6,0,0))
nn=20
hist(as.numeric(est[1,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[1,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[2,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[2,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[3,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[3,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[4,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[4,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[5,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
```

```
hist(as.numeric(fse[5,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[6,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[6,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[7,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[7,]), xlab="Standard error",ylab="Count",main="",nclass=nn)


quantile(as.numeric(est[1,]), prob=c(0.025,0.975))
xx=as.numeric(fse[1,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))
quantile(as.numeric(est[2,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[2,]), prob=c(0.025,0.975))
quantile(as.numeric(est[3,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[3,]), prob=c(0.025,0.975))
quantile(as.numeric(est[4,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[4,]), prob=c(0.025,0.975))
quantile(as.numeric(est[5,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[5,]), prob=c(0.025,0.975))
quantile(as.numeric(est[6,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[6,]), prob=c(0.025,0.975))
quantile(as.numeric(est[7,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[7,]), prob=c(0.025,0.975))



###################################
# Simulation for the ITG-PL model #
###################################

x = danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1],
spec1="invtrgamma", arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="paralogis",arg2=c(shape=p[5],scale=p[6]))}
par1 = optim(fn=ff,par=c(1,1,20,1,1,1),
control=list(maxit=100000))$par
est = data.frame(matrix(,6,100))
fse = data.frame(matrix(,6,100))

for(i in 1:100){
cat(i,"\n")
tt=proc.time()
r = runif(2489)
x = qmixt(r, phi=par1[1], spec1="invtrgamma",
arg1=c(shape1=par1[2],shape2=par1[3],scale=par1[4]),
spec2="paralogis",arg2=c(shape=par1[5],scale=par1[6]),
interval=c(0,1000000))
ff=function(p){nlogl.mixt(p, phi=p[1],
spec1="invtrgamma", arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
```

```
spec2="paralogis",arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,20,1,1,1),
control=list(maxit=100000),hessian=TRUE)
tt=proc.time()-tt
t2[i]=tt[3]
est[,i] = fun$par
hess = fun$hessian
se = 1/hess^0.5
diag = diag(se)
fse[,i] = diag
}


par(mfrow=c(4,3),oma=rep(0,4),mai=c(0.6,0.6,0,0))
nn=20
hist(as.numeric(est[1,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[1,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[2,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[2,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[3,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[3,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[4,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[4,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[5,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[5,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[6,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[6,]), xlab="Standard error",ylab="Count",main="",nclass=nn)



quantile(as.numeric(est[1,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[1,]), prob=c(0.025,0.975))
quantile(as.numeric(est[2,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[2,]), prob=c(0.025,0.975))
quantile(as.numeric(est[3,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[3,]), prob=c(0.025,0.975))
quantile(as.numeric(est[4,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[4,]), prob=c(0.025,0.975))
quantile(as.numeric(est[5,]), prob=c(0.025,0.975))
xx=as.numeric(fse[5,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))
quantile(as.numeric(est[6,]), prob=c(0.025,0.975))
xx=as.numeric(fse[6,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))
```

```
###################################
# Simulation for the ITG-IPL model #
###################################

x = danish[log(danish[1:2492]/max(danish[1:2492]))<=-1]
ff=function(p){nlogl.mixt(p, phi=p[1],
spec1="invtrgamma", arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis", arg2=c(shape=p[5],scale=p[6]))}
par1=optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000))$par
est = data.frame(matrix(,6,100))
fse = data.frame(matrix(,6,100))




for(i in 1:100){
cat(i,"\n")
tt=proc.time()
r = runif(2489)
x = qmixt(r, phi=par1[1], spec1="invtrgamma",
arg1=c(shape1=par1[2],shape2=par1[3],scale=par1[4]),
spec2="invparalogis",
arg2=c(shape=par1[5],scale=par1[6]), interval=c(0,1000000))
ff=function(p){nlogl.mixt(p, phi=p[1],
spec1="invtrgamma", arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
spec2="invparalogis", arg2=c(shape=p[5],scale=p[6]))}
fun = optim(fn=ff,par=c(1,1,1,1,1,1),
control=list(maxit=100000),hessian=TRUE)
tt=proc.time()-tt
t3[i]=tt[3]
est[,i] = fun$par
hess = fun$hessian
se = 1/hess^0.5
diag = diag(se)
fse[,i] = diag
}


par(mfrow=c(4,3),oma=rep(0,4),mai=c(0.6,0.6,0,0))
nn=20
hist(as.numeric(est[1,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[1,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[2,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[2,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[3,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[3,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
```

```
hist(as.numeric(est[4,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[4,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[5,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[5,]), xlab="Standard error",ylab="Count",main="",nclass=nn)
hist(as.numeric(est[6,]), xlab="Parameter est",ylab="Count",main="",nclass=nn)
hist(as.numeric(fse[6,]), xlab="Standard error",ylab="Count",main="",nclass=nn)


xx=as.numeric(est[1,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))
xx=as.numeric(fse[1,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))
quantile(as.numeric(est[2,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[2,]), prob=c(0.025,0.975))
quantile(as.numeric(est[3,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[3,]), prob=c(0.025,0.975))
quantile(as.numeric(est[4,]), prob=c(0.025,0.975))
quantile(as.numeric(fse[4,]), prob=c(0.025,0.975))
xx=as.numeric(est[5,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))
xx=as.numeric(fse[5,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))
xx=as.numeric(est[6,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))
xx=as.numeric(fse[6,])
quantile(xx[!is.na(xx)], prob=c(0.025,0.975))


############################
# Box plots of CPU times #
############################

par(mfrow=c(1,1),mai=c(0.8,0.8,0,0))
boxplot(cbind(t1,t2,t3),names=c("ITG-B","ITG-PL","ITG-IPL"),
ylab="CPU time")
```