2.2.2 Flexibility for unimodal and multimodal data

Figure 1 and Table 1

```
par(mfrow = c(1,1))
x = seq(from = 0, to = 10, by = 0.2)
arg1 = list(shape1 = 0.5, shape2 = 2, scale = 1)
arg2 = list(shape1 = 0.1, shape2 = 0.5, shape3 = 2, scale = 1)
plot(x, dmixt(x, phi = 1, spec1 = "invtrgamma", arg1 = arg1,
              spec2 = "trbeta", arg2 = arg2),
     type = "l", col = "darkblue",
     ylab = "PDF", ylim = c(0, 0.75),
     main = "Unimodality and bimodality of the mixture of
     inverse transformed gamma and transformed beta")

arg1 = list(shape1 = 0.1, shape2 = 0.5, scale = 1)
arg2 = list(shape1 = 0.9, shape2 = 1.5, shape3 = 5, scale = 1)
lines(x, dmixt(x, phi = 2.5, spec1 = "invtrgamma", arg1 = arg1,
               spec2 = "trbeta", arg2 = arg2), col = "darkgreen")

arg1 = list(shape1 = 10, shape2 = 3, scale = 10)
arg2 = list(shape1 = 0.2, shape2 = 5, shape3 = 2, scale = 0.5)
lines(x, dmixt(x, phi = 5, spec1 = "invtrgamma", arg1 = arg1,
               spec2 = "trbeta", arg2 = arg2), type = "l", col = "darkred")
legend("topright", c("Model A", "Model B", "Model C"), col = c("darkblue", "darkgreen", "darkred"),lty=1,
box.lwd = 2)
```

Figure 2 and Table 2

```
par(mfrow = c(1,1))
x = seq(from = 0, to = 10, by = 0.2)
arg1 = list(shape1 = 0.1, shape2 = 2, scale = 1)
arg2 = list(shape1 = 2, shape2 = 5, scale = 1)
plot(x, dmixt(x, phi = 1, spec1 = "burr", arg1 = arg1,
              spec2 = "burr", arg2 = arg2), type = "l", col = "darkblue",
     ylab = "PDF",
     main = "Unimodality and bimodality of the mixture of two Burr distributions")

x = seq(from = 0, to = 10, by = 0.2)
arg1 = list(shape1 = 2, shape2 = 0.6, scale = 1)
arg2 = list(shape1 = 3, shape2 = 0.2, scale = 1)
lines(x, dmixt(x, phi = 2, spec1 = "burr", arg1 = arg1,
               spec2 = "burr", arg2 = arg2), type = "l", col = "darkgreen")

x = seq(from = 0, to = 10, by = 0.2)
arg1 = list(shape1 = 10, shape2 = 10, scale = 3)
arg2 = list(shape1 = 2.5, shape2 = 8, scale = 8)
lines(x, dmixt(x, phi = 1, spec1 = "burr", arg1 = arg1,
               spec2 = "burr", arg2 = arg2), type = "l", col = "darkred")
legend("topright", c("Model A", "Model B", "Model C"), col = c("darkblue", "darkgreen", "darkred"), lty=1,
box.lwd = 2)
```

3. Empirical analysis

```
library(SMPracticals)
library(composite)
#"http://ifas.jku.at/gruen/composite/composite_0.9-3.tar.gz", repos = NULL, type = "source"
library(moments)
library(fitdistrplus)
library(actuar)
library(readxl)
library(gendist)
library(pracma)
library(numDeriv)

Taxi_Claims <- read_excel("C:/Users/anesu/Downloads/Taxi Claims Data.xlsx", col_names = FALSE)
colnames(Taxi_Claims) <- "Claims"
t <- Taxi_Claims$Claims/100
t <- sort(t)
head(t)
length(t)


data("danish", package = "SMPracticals")
str(danish)
#Set danish to a numeric series if it is a time series.
if(is.timeSeries(danish))  danish <- series(danish)
danish <- as.numeric(danish)
d <- sort(danish[1:2492])
n = length(d)
```

Table 3

```
summary(t)
mean(t)
sd(t)
#Coefficient of variation
sd(t)/mean(t)
skewness(t)
kurtosis(t)
```

Table 4

```
summary(d)
mean(d)
sd(d)
#Coefficient of variation
sd(d)/mean(d)
skewness(d)
kurtosis(d)
```

Figure 3

```
boxplot(t, xlab = "Losses in hundreds of Rand", main = "Boxplot of taxi claims data", horizontal = TRUE)
abline(v = mean(t), lty = 4, col = "darkgreen")
boxplot(danish, xlab = "Losses in millions of Danish Krones", main = "Boxplot of Danish fire losses", horizontal
= TRUE)
abline(v = mean(danish), lty = 2, cex = 1.5, col = "darkgreen")
```

Figure 4

```
hist(t, prob = T, ylim = c(0, 0.015), xlab = "", main = "Histogram of taxi claims", breaks = 100)
lines(density(t), lty = 4, type = "l", col = "cyan")
hist(d, prob = T, ylim = c(0, 0.75), xlab = "", main = "Histogram of Danish fire insurance claims", breaks =
100)
lines(density(d), lty = 4, col = "cyan")
```

Figure 5

```
par(mfrow=c(1,2))
ReIns::MeanExcess(t, k=FALSE, main = “Mean excess plot for taxi claims”)
ReIns::MeanExcess(d, k=FALSE, main = “Mean excess plot for Danish losses”)
```

Table 5 and Table A2

```
#1. Gamma-Weibull
set.seed(1234)
a <- CompDist("gamma", "weibull")
a_fit_t <- FitParameters(a, t, iterations = 100)
a_loglik_t <- logLik(a_fit_t, t)
a_NLL_t <- as.numeric((-1)*a_loglik_t)
a_AIC_t <- 2*a_NLL_t + 2*(length(a@dist1@pars) + length(a@dist2@pars))
a_BIC_t <- 2*a_NLL_t + log(length(t))*(length(a@dist1@pars) + length(a@dist2@pars))
a_Refit_t <- RefitParameters(a_fit_t, t)
c(a_NLL_t, a_AIC_t, a_BIC_t)
a_Refit_t

#2. Paralogistic-Inverse Gaussian
set.seed(1234)
b <- CompDist("paralogis", "invgauss")
b_fit_t <- FitParameters(b, t, iterations = 100)
b_loglik_t <- logLik(b_fit_t, t)
b_NLL_t <- as.numeric((-1)*b_loglik_t)
b_AIC_t <- 2*b_NLL_t + 2*(length(b@dist1@pars) + length(b@dist2@pars))
b_BIC_t <- 2*b_NLL_t + log(length(t))*(length(b@dist1@pars) + length(b@dist2@pars))
b_Refit_t <- RefitParameters(b_fit_t, t)
c(b_NLL_t, b_AIC_t, b_BIC_t)
b_Refit_t

#3. Loglogistic-Inverse Gaussian
set.seed(1234)
c <- CompDist("llogis", "invgauss")
c_fit_t <- FitParameters(c, t, iterations = 100)
c_loglik_t <- logLik(c_fit_t, t)
c_NLL_t <- as.numeric((-1)*c_loglik_t)
c_AIC_t <- 2*c_NLL_t + 2*(length(c@dist1@pars) + length(c@dist2@pars))
c_BIC_t <- 2*c_NLL_t + log(length(t))*(length(c@dist1@pars) + length(c@dist2@pars))
c(c_NLL_t, c_AIC_t, c_BIC_t)
c_Refit_t <- RefitParameters(c_fit_t, t)

#4. Paralogistic-Weibull
set.seed(1234)
e <- CompDist("paralogis", "weibull")
e_fit_t <- FitParameters(e, t, iterations = 100)
e_loglik_t <- logLik(e_fit_t, t)
e_NLL_t <- as.numeric((-1)*e_loglik_t)
e_AIC_t <- 2*e_NLL_t + 2*(length(e@dist1@pars) + length(e@dist2@pars))
e_BIC_t <- 2*e_NLL_t + log(length(t))*(length(e@dist1@pars) + length(e@dist2@pars))
c(e_NLL_t, e_AIC_t, e_BIC_t)
e_Refit_t <- RefitParameters(e_fit_t, t)

#5. Inverse Paralogistic-Inverse Gaussian
set.seed(1234)
f <- CompDist("invparalogis", "invgauss")
f_fit_t <- FitParameters(f, t, iterations = 100)
f_loglik_t <- logLik(f_fit_t, t)
f_NLL_t <- as.numeric((-1)*f_loglik_t)
f_AIC_t <- 2*f_NLL_t + 2*(length(f@dist1@pars) + length(f@dist2@pars))
f_BIC_t <- 2*f_NLL_t + log(length(t))*(length(f@dist1@pars) + length(f@dist2@pars))
c(f_NLL_t, f_AIC_t, f_BIC_t)
f_Refit_t <- RefitParameters(f_fit_t, t)

#6. Weibull-Weibull
set.seed(1234)
g <- CompDist("weibull", "weibull")
g_fit_t <- FitParameters(g, t, iterations = 100)
g_loglik_t <- logLik(g_fit_t, t)
g_NLL_t <- as.numeric((-1)*g_loglik_t)
g_AIC_t <- 2*g_NLL_t + 2*(length(g@dist1@pars) + length(g@dist2@pars))
g_BIC_t <- 2*g_NLL_t + log(length(t))*(length(g@dist1@pars) + length(g@dist2@pars))
c(g_NLL_t, g_AIC_t, g_BIC_t)
g_Refit_t <- RefitParameters(g_fit_t, t)

#7. Gamma-Burr
set.seed(1234)
h <- CompDist("gamma", "burr")
h_fit_t <- FitParameters(h, t, iterations = 100)
h_loglik_t <- logLik(h_fit_t, t)
h_NLL_t <- as.numeric((-1)*h_loglik_t)
h_AIC_t <- 2*h_NLL_t + 2*(length(h@dist1@pars) + length(h@dist2@pars))
h_BIC_t <- 2*h_NLL_t + log(length(t))*(length(h@dist1@pars) + length(h@dist2@pars))
c(h_NLL_t, h_AIC_t, h_BIC_t)
h_Refit_t <- RefitParameters(h_fit_t, t)
```

```
#8. Loglogistic-Weibull
set.seed(1234)
i = CompDist("llogis", "weibull")
i_fit = FitParameters(i,t,iterations=100)
i_loglik_t <- logLik(i_fit, t)
i_NLL_t <- as.numeric((-1)*i_loglik_t)
i_AIC_t <- 2*i_NLL_t + 2*(length(i@dist1@pars) + length(i@dist2@pars))
i_BIC_t <- 2*i_NLL_t + log(length(t))*(length(i@dist1@pars) + length(i@dist2@pars))
c(i_NLL_t, i_AIC_t, i_BIC_t)
i_Refit_t <- RefitParameters(i_fit, t)

#9. Paralogistic-Burr
set.seed(1234)
j <- CompDist("paralogis", "burr")
j_fit_t <- FitParameters(j, t, iterations = 100)
j_loglik_t <- logLik(j_fit_t, t)
j_NLL_t <- as.numeric((-1)*j_loglik_t)
j_AIC_t <- 2*j_NLL_t + 2*(length(j@dist1@pars) + length(j@dist2@pars))
j_BIC_t <- 2*j_NLL_t + log(length(t))*(length(j@dist1@pars) + length(j@dist2@pars))
c(j_NLL_t, j_AIC_t, j_BIC_t)
i_Refit_t <- RefitParameters(i_fit_t, t)

#10. Weibull-Burr
set.seed(1234)
k <- CompDist("weibull","burr")
k_fit = FitParameters(k,t,iterations=50)
k_loglik_t <- logLik(k_fit, t)
k_NLL_t <- as.numeric((-1)*k_loglik_t)
k_AIC_t <- 2*k_NLL_t + 2*(length(k@dist1@pars) + length(k@dist2@pars))
k_BIC_t <- 2*k_NLL_t + log(length(t))*(length(k@dist1@pars) + length(k@dist2@pars))
c(k_NLL_t, k_AIC_t, k_BIC_t)
k_Refit_t <- RefitParameters(k_fit, t)

#11. Inverse Burr-Weibull
set.seed(1234)
l <- CompDist("invburr","weibull")
l_fit = FitParameters(l,t,iterations=100)
l_loglik_t <- logLik(l_fit, t)
l_NLL_t <- as.numeric((-1)*l_loglik_t)
l_AIC_t <- 2*l_NLL_t + 2*(length(l@dist1@pars) + length(l@dist2@pars))
l_BIC_t <- 2*l_NLL_t + log(length(t))*(length(l@dist1@pars) + length(l@dist2@pars))
c(l_NLL_t, l_AIC_t, l_BIC_t)
l_Refit_t <- RefitParameters(l_fit, t)

#12. Loglogistic-Burr
set.seed(1234)
m <- CompDist("llogis", "burr")
m_fit_t <- FitParameters(m, t, iterations = 200)
m_loglik_t <- logLik(m_fit_t, t)
m_NLL_t <- as.numeric((-1)*m_loglik_t)
m_AIC_t <- 2*m_NLL_t + 2*(length(m@dist1@pars) + length(m@dist2@pars))
m_BIC_t <- 2*m_NLL_t + log(length(t))*(length(m@dist1@pars) + length(m@dist2@pars))
c(m_NLL_t, m_AIC_t, m_BIC_t)
m_Refit_t <- RefitParameters(m_fit_t, t)

#13. Inverse Burr-Burr
set.seed(1234)
n <- CompDist("invburr", "burr")
n_fit_t <- FitParameters(n, t, iterations = 100)
n_loglik_t <- logLik(n_fit_t, t)
n_NLL_t <- as.numeric((-1)*n_loglik_t)
n_AIC_t <- 2*n_NLL_t + 2*(length(n@dist1@pars) + length(n@dist2@pars))
n_BIC_t <- 2*n_NLL_t + log(length(t))*(length(n@dist1@pars) + length(n@dist2@pars))
c(n_NLL_t, n_AIC_t, n_BIC_t)
n_Refit_t <- RefitParameters(n_fit_t, t)

#14. Inverse Paralogistic-Weibull
set.seed(1234)
o <- CompDist("invparalogis","weibull")
o_fit = FitParameters(o,t,iterations=100)
o_loglik_t <- logLik(o_fit, t)
o_NLL_t <- as.numeric((-1)*o_loglik_t)
o_AIC_t <- 2*o_NLL_t + 2*(length(o@dist1@pars) + length(o@dist2@pars))
o_BIC_t <- 2*o_NLL_t + log(length(t))*(length(o@dist1@pars) + length(o@dist2@pars))
c(o_NLL_t, o_AIC_t, o_BIC_t)
o_Refit_t <- RefitParameters(o_fit, t)

#15. Inverse Paralogistic-Burr
set.seed(1234)
p <- CompDist("invparalogis", "burr")
p_fit_t <- FitParameters(p, t, iterations = 100)
p_loglik_t <- logLik(p_fit_t, t)
p_NLL_t <- as.numeric((-1)*p_loglik_t)
p_AIC_t <- 2*p_NLL_t + 2*(length(p@dist1@pars) + length(p@dist2@pars))
p_BIC_t <- 2*p_NLL_t + log(length(t))*(length(p@dist1@pars) + length(p@dist2@pars))
c(p_NLL_t, p_AIC_t, p_BIC_t)
p_Refit_t <- RefitParameters(p_fit_t, t)

#16. Burr-Pareto
set.seed(1234)
q <- CompDist("burr", "pareto")
q_fit_t <- FitParameters(q, t, iterations = 100)
q_loglik_t <- logLik(q_fit_t, t)
q_NLL_t <- as.numeric((-1)*q_loglik_t)
q_AIC_t <- 2*q_NLL_t + 2*(length(q@dist1@pars) + length(q@dist2@pars))
q_BIC_t <- 2*q_NLL_t + log(length(t))*(length(q@dist1@pars) + length(q@dist2@pars))
c(q_NLL_t, q_AIC_t, q_BIC_t)
q_Refit_t <- RefitParameters(q_fit_t, t)

#17. Weibull-Lognormal
set.seed(1234)
r <- CompDist("weibull", "lnorm")
r_fit_t <- FitParameters(r, t, iterations = 100)
r_loglik_t <- logLik(r_fit_t, t)
r_NLL_t <- as.numeric((-1)*r_loglik_t)
r_AIC_t <- 2*r_NLL_t + 2*(length(r@dist1@pars) + length(r@dist2@pars))
r_BIC_t <- 2*r_NLL_t + log(length(t))*(length(r@dist1@pars) + length(r@dist2@pars))
```

```
c(r_NLL_t, r_AIC_t, r_BIC_t)
r_Refit_t <- RefitParameters(r_fit_t, t)

#18. Gamma-Lognormal
set.seed(1234)
s <- CompDist("gamma", "lnorm")
s_fit_t <- FitParameters(s, t, iterations = 100)
s_loglik_t <- logLik(s_fit_t, t)
s_NLL_t <- as.numeric((-1)*s_loglik_t)
s_AIC_t <- 2*s_NLL_t + 2*(length(s@dist1@pars) + length(s@dist2@pars))
s_BIC_t <- 2*s_NLL_t + log(length(t))*(length(s@dist1@pars) + length(s@dist2@pars))
c(s_NLL_t, s_AIC_t, s_BIC_t)
s_Refit_t <- RefitParameters(s_fit_t, t)

#19. Gamma-Generalized Pareto
set.seed(1234)
u <- CompDist("gamma", "genpareto")
u_fit_t <- FitParameters(u, t, iterations = 100)
u_loglik_t <- logLik(u_fit_t, t)
u_NLL_t <- as.numeric((-1)*u_loglik_t)
u_AIC_t <- 2*u_NLL_t + 2*(length(u@dist1@pars) + length(u@dist2@pars))
u_BIC_t <- 2*u_NLL_t + log(length(t))*(length(u@dist1@pars) + length(u@dist2@pars))
c(u_NLL_t, u_AIC_t, u_BIC_t)
u_Refit_t <- RefitParameters(u_fit_t, t)

#20. Paralogistic-Lognormal
set.seed(1234)
v <- CompDist("paralogis", "lnorm")
v_fit_t <- FitParameters(v, t, iterations = 100)
v_loglik_t <- logLik(v_fit_t, t)
v_NLL_t <- as.numeric((-1)*v_loglik_t)
v_AIC_t <- 2*v_NLL_t + 2*(length(v@dist1@pars) + length(v@dist2@pars))
v_BIC_t <- 2*v_NLL_t + log(length(t))*(length(v@dist1@pars) + length(v@dist2@pars))
c(v_NLL_t, v_AIC_t, v_BIC_t)
v_Refit_t <- RefitParameters(v_fit_t, t)
```

Table 6

```
#1.Gamma-Weibull

pweibull(35.52616, shape = 0.3324826, scale = 7.1856879)

(var_95 <- qweibull(0.8175457 + ((1-0.8175457)/1.395733)*(0.95*(1 + 1.395733) - 1), shape = 0.3324826, scale =
7.1856879))
(var_99 <- qweibull(0.8175457 + ((1-0.8175457)/1.395733)*(0.99*(1 + 1.395733) - 1), shape = 0.3324826, scale =
7.1856879))

f = function(x) qweibull(0.8175457 + ((1-0.8175457)/1.395733)*(x*(1 + 1.395733) - 1), shape = 0.3324826, scale =
7.1856879)
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.28833)
(es_99 <- (1/(1-0.99))*24.22041)

#2.Paralogistic-Inverse Gaussian

pinvgauss(37.11934, mean = 101.31497, shape = 17.46168)

(var_95 <- qinvgauss(0.5780225 + ((1-0.5780225)/1.298232)*(0.95*(1 + 1.298232) - 1), mean = 101.31497, shape =
17.46168))
(var_99 <- qinvgauss(0.5780225 + ((1-0.5780225)/1.298232)*(0.99*(1 + 1.298232) - 1), mean = 101.31497, shape =
17.46168))

f = function(x) qinvgauss(0.5780225 + ((1-0.5780225)/1.298232)*(x*(1 + 1.298232) - 1), mean = 101.31497, shape =
17.46168)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*53.43002)
(es_99 <- (1/(1-0.99))*20.55102)

#3.Loglogistic-Inverse Gaussian

pinvgauss(40.51527, mean = 99.64726, shape = 16.87046)

(var_95 <- qinvgauss(0.6062747 + ((1-0.6062747)/1.159389)*(0.95*(1 + 1.159389) - 1), mean = 99.64726, shape =
16.87046))
(var_99 <- qinvgauss(0.6062747 + ((1-0.606274)/1.159389)*(0.99*(1 + 1.159389) - 1), mean = 99.64726, shape =
16.87046))

f = function(x) qinvgauss(0.6062747 + ((1-0.6062747)/1.159389)*(x*(1 + 1.159389) - 1), mean = 99.64726, shape =
16.87046)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*53.4402)
(es_99 <- (1/(1-0.99))*20.56093)

#4.Paralogistic-Weibull

pweibull(34.78929, shape = 0.3338412, scale = 7.3820175)

(var_95 <- qweibull(0.8132338 + ((1-0.8132338)/1.431932)*(0.95*(1 + 1.431932)-1), shape = 0.3338412, scale =
7.3820175))
(var_99 <- qweibull(0.8132338 + ((1-0.8132338)/1.431932)*(0.99*(1 + 1.431932)-1), shape = 0.3338412, scale =
7.3820175))

f = function(x) qweibull(0.8132338 + ((1-0.8132338)/1.431932)*(x*(1 + 1.431932)-1), shape = 0.3338412, scale =
7.3820175)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.22276)
(es_99 <- (1/(1-0.99))*24.16225)

#5.Inverse Paralogistic-Inverse Gaussian
```

```
pinvgauss(51.2797, mean = 91.28592, shape = 14.06192)

(var_95 <- qinvgauss(0.6905617 + ((1-0.6905617)/0.8695379)*(0.95*(1+0.8695379)-1), mean = 91.28592, shape =
14.06192))
(var_99 <- qinvgauss(0.6905617 + ((1-0.6905617)/0.8695379)*(0.99*(1+0.8695379)-1), mean = 91.28592, shape =
14.06192))

f = function(x) qinvgauss(0.6905617 + ((1-0.6905617)/0.8695379)*(x*(1+0.8695379)-1), mean = 91.28592, shape =
14.06192)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*53.50899)
(es_99 <- (1/(1-0.99))*20.61908)

#6.Weibull-Weibull

pweibull(30.88943, shape = 0.3376205, scale = 7.9323997)

(var_95 <- qweibull(0.7945311 + ((1-0.7945311)/1.679067)*(0.95*(1+1.679067)-1), shape = 0.3376205, scale =
7.9323997))
(var_99 <- qweibull(0.7945311 + ((1-0.7945311)/1.679067)*(0.99*(1+1.679067)-1), shape = 0.3376205, scale =
7.9323997))

f = function(x) qweibull(0.7945311 + ((1-0.7945311)/1.679067)*(x*(1+1.679067)-1), shape = 0.3376205, scale =
7.9323997)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*55.988574)
(es_99 <- (1/(1-0.99))*23.96799)

#7.Gamma-Burr
pburr(35.5336, shape1 = 9.290451e+06, shape2 = 3.324042e-01, rate = 1.519089e-22)

(var_95 <- qburr(0.8176692 + ((1-0.8176692)/1.395305)*(0.95*(1+1.395305)-1), shape1 = 9.290451e+06, shape2 =
3.324042e-01, rate = 1.519089e-22))
(var_99 <- qburr(0.8176692 + ((1-0.8176692)/1.395305)*(0.99*(1+1.395305)-1), shape1 = 9.290451e+06, shape2 =
3.324042e-01, rate = 1.519089e-22))

f = function(x) qburr(0.8176692 + ((1-0.8176692)/1.395305)*(x*(1+1.395305)-1), shape1 = 9.290451e+06, shape2 =
3.324042e-01, rate = 1.519089e-22)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.30491)
(es_99 <- (1/(1-0.99))*24.23174)

#8. Loglogistic-Weibull

pweibull(37.162, shape = 0.3335, scale = 7.3185)

(var_95 <- qweibull(0.8208025+((1-0.8208025)/1.314)*(0.95*(1+1.314)-1), shape = 0.3335, scale = 7.3185))
(var_99 <- qweibull(0.8208025+((1-0.8208025)/1.314)*(0.99*(1+1.314)-1), shape = 0.3335, scale = 7.3185))

f = function(x) qweibull(0.8208025+((1-0.8208025)/1.314)*(x*(1+1.314)-1), shape = 0.3335, scale = 7.3185)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.19262)
(es_99 <- (1/(1-0.99))*24.15197)

#9.Paralogistic-Burr
pburr(34.78983, shape1 = 132231.5, shape2 = 0.3338478, rate = 6.186885e-17)

(var_95 <- qburr(0.8132307 + ((1-0.8132307)/1.431911)*(0.95*(1+1.431911)-1),  shape1 = 132231.5, shape2 =
0.3338478, rate = 6.186885e-17))
(var_99 <- qburr(0.8132307 + ((1-0.8132307)/1.431911)*(0.99*(1+1.431911)-1),  shape1 = 132231.5, shape2 =
0.3338478, rate = 6.186885e-17))

f = function(x) qburr(0.8132307 + ((1-0.8132307)/1.431911)*(x*(1+1.431911)-1),  shape1 = 132231.5, shape2 =
0.3338478, rate = 6.186885e-17)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.22318)
(es_99 <- (1/(1-0.99))*24.16255)

#10. Weibull-Burr

pburr(30.89098, shape1=873910.9, shape2=0.3385995, scale=3.173873e-19)

(var_95 <- qburr(1+((1-1)/1.678947)*(0.95*(1+1.678947)-1),shape1=873910.9,shape2=0.3375995,scale=3.173873e-19))
(var_99 <- qburr(1+((1-1)/1.678947)*(0.99*(1+1.678947)-1),shape1=873910.9,shape2=0.3375995,scale=3.173873e-19))

#11. Inverse Burr-Weibull

pweibull(35.0194, shape = 0.3341, scale = 7.4239)

(var_95 <- qweibull(0.8134573+((1-0.8134573)/1.4186)*(0.95*(1+1.4186)-1),shape = 0.3341, scale = 7.4239))
(var_99 <- qweibull(0.8134573+((1-0.8134573)/1.4186)*(0.99*(1+1.4186)-1),shape = 0.3341, scale = 7.4239))

f = function(x) qweibull(0.8134573+((1-0.8134573)/1.4186)*(x*(1+1.4186)-1),shape = 0.3341, scale = 7.4239)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.23277)
(es_99 <- (1/(1-0.99))*24.16118)

#12. Loglogistic-Burr
pburr(37.16212, shape1 = 229050.8, shape2 = 0.33348, scale = 1.155759e-17)

(var_95 <- qburr(1+((1-1)/1.314023)*(0.95*(1+1.314023)-1),shape1 = 229050.8, shape2 = 0.33348, scale =
1.155759e-17))
(var_95 <- qburr(1+((1-1)/1.314023)*(0.99*(1+1.314023)-1),shape1 = 229050.8, shape2 = 0.33348, scale =
1.155759e-17))

#13. Inverse Burr-Burr
pburr(35.01608, shape1 = 18358.92, shape2 = 3.342370e-01 , rate = 2.354395e-14)

(var_95 <- qburr(0.8133716 + ((1-0.8133716)/1.418816)*(0.95*(1+1.418816)-1), shape1 = 18358.92, shape2 =
3.342370e-01 , rate = 2.354395e-14))
```

```
(var_99 <- qburr(0.8133716 + ((1-0.8133716)/1.418816)*(0.99*(1+1.418816)-1), shape1 = 18358.92, shape2 =
3.342370e-01 , rate = 2.354395e-14))

f = function(x) qburr(0.8133716 + ((1-0.8133716)/1.418816)*(x*(1+1.418816)-1), shape1 = 18358.92, shape2 =
3.342370e-01 , rate = 2.354395e-14)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.19621)
(es_99 <- (1/(1-0.99))*24.14262)

#14. Inverse Paralogistic-Weibull

pweibull(44.1095, shape = 0.3313, scale = 6.9658)

(var_95 <- qweibull(0.8416797 + ((1-0.8416797)/1.0634)*(0.95*(1+1.0634)-1),shape = 0.3313, scale = 6.9658))
(var_99 <- qweibull(0.8416797 + ((1-0.8416797)/1.0634)*(0.99*(1+1.0634)-1),shape = 0.3313, scale = 6.9658))

f = function(x) qweibull(0.8416797 + ((1-0.8416797)/1.0634)*(x*(1+1.0634)-1),shape = 0.3313, scale = 6.9658)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.18658)
(es_99 <- (1/(1-0.99))*24.18417)

#15. Inverse Paralogistic-Burr
pburr(44.10841, shape1 = 29397.29, shape2 = 0.331288, rate = 4.668775e-15)

(var_95 <- qburr(0.8416297 + ((1-0.8416297)/1.063423)*(0.95*(1+1.063423)-1), shape1 = 29397.29, shape2 =
0.331288, rate = 4.668775e-15))
(var_99 <- qburr(0.8416297 + ((1-0.8416297)/1.063423)*(0.99*(1+1.063423)-1), shape1 = 29397.29, shape2 =
0.331288, rate = 4.668775e-15))

f = function(x) qburr(0.8416297 + ((1-0.8416297)/1.063423)*(x*(1+1.063423)-1), shape1 = 29397.29, shape2 =
0.331288, rate = 4.668775e-15)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.22393)
(es_99 <- (1/(1-0.99))*24.20341)

#16. Burr-Pareto

ppareto(371.0913, shape = 2.682678, scale = 443.996072)

(var_95 <- qpareto(0.8040062 + ((1-0.8040062)/0.08827937)*(0.95*(1+0.08827937)-1) , shape = 2.682678, scale =
443.996072))
(var_99 <- qpareto(0.8040062 + ((1-0.8040062)/0.08827937)*(0.99*(1+0.08827937)-1) , shape = 2.682678, scale =
443.996072))

f = function(x) qpareto(0.8040062 + ((1-0.8040062)/0.08827937)*(x*(1+0.08827937)-1) , shape = 2.682678, scale =
443.996072)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*55.61768)
(es_99 <- (1/(1-0.99))*23.91679)

#17. Weibull-Lognormal

plnorm(27.36936, meanlog = 3.299477, sdlog = 1.652108)

(var_95 <- qlnorm(0.502402 + ((1-0.502404)/2.012194)*(0.95*(1+2.012194)-1), meanlog = 3.299477, sdlog =
1.652108))
(var_99 <- qlnorm(0.502402 + ((1-0.502404)/2.012194)*(0.99*(1+2.012194)-1), meanlog = 3.299477, sdlog =
1.652108))

f = function(x) qlnorm(0.502402 + ((1-0.502404)/2.012194)*(x*(1+2.012194)-1), meanlog = 3.299477, sdlog =
1.652108)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*63.53909)
(es_99 <- (1/(1-0.99))*30.67212)

#18. Gamma-Lognormal

plnorm(32.33345, meanlog = 3.249564, sdlog = 1.668629)

(var_95 <- qlnorm(0.5539957 + ((1-0.5539957)/1.602943)*(0.95*(1+1.602943)-1), meanlog = 3.249564, sdlog =
1.668629))
(var_99 <- qlnorm(0.5539957 + ((1-0.5539957)/1.602943)*(0.99*(1+1.602943)-1), meanlog = 3.249564, sdlog =
1.668629))

f = function(x) qlnorm(0.5539957 + ((1-0.5539957)/1.602943)*(x*(1+1.602943)-1), meanlog = 3.249564, sdlog =
1.668629)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*64.32513)
(es_99 <- (1/(1-0.99))*31.3302)

#19. Gamma-Generalized Pareto

pgenpareto(33.02724, shape1 = 2.00855, shape2 = 0.00000007749274, rate = 0.002625036)

(var_95 <- qgenpareto(0.9999999 + ((1-0.9999999)/1.561624)*(0.95*(1+1.561624)-1), shape1 = 2.00855, shape2 =
0.00000007749274, rate = 0.002625036))
(var_99 <- qgenpareto(0.9999999 + ((1-0.9999999)/1.561624)*(0.99*(1+1.561624)-1), shape1 = 2.00855, shape2 =
0.00000007749274, rate = 0.002625036))

f = function(x) qgenpareto(0.9999999 + ((1-0.9999999)/1.561624)*(x*(1+1.561624)-1), shape1 = 2.00855, shape2 =
0.00000007749274, rate = 0.002625036)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*69.80659)
(es_99 <- (1/(1-0.99))*34.04882)

#20. Paralogistic-Lognormal

plnorm(30.95523, meanlog = 3.255658, sdlog = 1.663654)
```

```
(var_95 <- qlnorm(0.5423368 + ((1-0.5423368)/1.697679)*(0.95*(1+1.697679)-1), meanlog = 3.255658, sdlog =
1.663654))
(var_99 <- qlnorm(0.5423368 + ((1-0.5423368)/1.697679)*(0.99*(1+1.697679)-1), meanlog = 3.255658, sdlog =
1.663654))

f = function(x) qlnorm(0.5423368 + ((1-0.5423368)/1.697679)*(x*(1+1.697679)-1), meanlog = 3.255658, sdlog =
1.663654)
integrate(f , lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*63.74534)
(es_99 <- (1/(1-0.99))*30.98643)
```

Table 7 and Table A3

```
nlogl.mixt <- function(p, phi, spec1, arg1, spec2, arg2){
  tt <- 1.0e20
  if(all(p>0)){
    tt <- -sum(log( dmixt(t, phi, spec1, arg1, spec2, arg2) ))
  }
  return(tt)
}
#1.Inverse Gamma-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invgamma",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[4], sdlog=p[5]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1))
fun = nlminb(start = c(100,1,1,1,1), objective = ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#2.Inverse Gaussian-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invgauss",
                          arg1=c(mean=p[2],shape=p[3]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[4], sdlog=p[5]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1))
fun = nlm(f=ff,p=c(0.0014,147.5,28.85,1.07,18))
fun = nlm(f=ff,p=c(5,1,1,1,1))
fun = nlm(f=ff,p=c(0.003,1,1,1,1))
fun = nlm(f=ff,p=c(42.5,19.1,2646.85,3.96,1.3))
fun = nlminb(start = c(100,1,1,1,1), objective = ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#3. Generalized Pareto-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="genpareto",
                          arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[5], sdlog=p[6]))
}
fun = nlm(f=ff,p=c(10,1,1,1,1,1))
fun = nlm(f=ff,p=c(4.5,4.7,20.9,5.6,4.1,1.4))
fun = nlm(f=ff,p=c(4.4,4.2,45.3,2.3,4.1,1.4))
fun = nlminb(start=c(10,1,1,1,1,1),objective=ff)
fun = nlminb(start=c(4.436,4.05,64.925,1.5,4.075,1.4),objective=ff)
cat(formatC(fun$objective,digits=3,format="f")," & ",
    formatC(2*fun$objective+2*6,digits=3,format="f")," & ",
    formatC(2*fun$objective+6*log(length(t)),digits=3,format="f"),"","\n")

#4. Inverse Paralogistic-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invparalogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[4], sdlog=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlminb(start=c(10,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#5. Inverse Weibull-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invweibull",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[4], sdlog=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#6. Inverse Burr-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invburr",
                          arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[5], sdlog=p[6]))
}
fun = nlm(f=ff,p=c(5,1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(t)),digits=3,format="f"),"","\n")

#7. Loglogistic-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="llogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[4], sdlog=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlminb(start=c(10,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#8. Burr-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="burr",
                          arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[5], sdlog=p[6]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1,1))
fun = nlminb(start=c(10,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(t)),digits=3,format="f"),"","\n")

#9. Gamma-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="gamma",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[4], sdlog=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlm(f=ff,p=c(2.1,2,14.3,4.3,1.3))
fun = nlminb(start=c(100,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#10. Paralogistic-Lognormal
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="paralogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "lnorm",
                          arg2 = c(meanlog=p[4], sdlog=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#11. Lognormal-Weibull
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="lnorm",
                          arg1=c(meanlog=p[2],sdlog=p[3]),
                          spec2="weibull",
                          arg2=c(shape=p[4], scale=p[5]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1))
fun = nlm(f=ff,p=c(0.2575,4.2,1.36,1.9,29.6))
fun = nlminb(start=c(1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#12. Loglogistic-Generalized Pareto
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="llogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "genpareto",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
fun = nlm(f=ff,p=c(89.6,8.3,3.4,1.3,26,24.3))
fun = nlm(f=ff,p=c(4.2,2.9,22.8,1.45,1.5,58.1))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
fun = nlminb(start=c(0.45,1.759,30.066,2.18,1.3,192.77),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(t)),digits=3,format="f"),"","\n")

#13. Generalized Pareto-Paralogistic
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="genpareto",
                          arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
                          spec2="paralogis",
                          arg2=c(shape=p[5], scale=p[6]))
}
fun = nlm(f=ff,p=c(10,1,1,1,1,1))
fun = nlm(f=ff,p=c(2.25,1.8,13.4,22.4,1.7,43))
fun = nlm(f=ff,p=c(1.2,2,1.7,156,1.7,39))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(t)),digits=3,format="f"),"","\n")

#14. Loglogistic-Paralogistic
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="llogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2="paralogis",
                          arg2=c(shape=p[4], scale=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlm(f=ff,p=c(0.5,1.8,29.4,1.3,199.2))
fun = nlminb(start=c(0.100,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#15. Burr-Loglogistic
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="burr",
                          arg1=c(shape1=p[2],shape2=p[3],scale=p[4]),
                          spec2="llogis",
                          arg2=c(shape=p[5], scale=p[6]))
}
```

```
fun = nlm(f=ff,p=c(2,1,1,1,1,1))
fun = nlm(f=ff,p=c(21.6,0.06,17.8,169.1,1.4,44.5))
fun = nlm(f=ff,p=c(1.7,1.2,1.4,172.7,1.8,28.8))
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(t)),digits=3,format="f"),"","\n")

#16. Paralogistic-Paralogistic
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="paralogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2="paralogis",
                          arg2=c(shape=p[4], scale=p[5]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1))
fun = nlminb(start=c(5,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#17. Burr-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="burr",
                          arg1=c(shape=p[2],shape2=p[3],scale=p[4]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[5], shape2 = p[6], scale = p[7]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1,1,1))
fun = nlm(f=ff,p=c(1.4,6.6,1.7,84.6,1.15,1.4,116.8))
fun = nlminb(start=c(2,1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*7,digits=3,format="f")," & ",
    formatC(2*fun$minimum+7*log(length(t)),digits=3,format="f"),"","\n")

#18. Inverse Gamma-Paralogistic
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invgamma",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2="paralogis",
                          arg2=c(shape=p[4], scale=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(t)),digits=3,format="f"),"","\n")

#19. Inverse Gamma-Generalized Pareto
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invgamma",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "genpareto",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
fun = nlm(f=ff,p=c(0.52,1.2,67,6.65,2.4,49.8))
fun = nlm(f=ff,p=c(2.3,1.8,304.5,8,2,133.8))
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(t)),digits=3,format="f"),"","\n")

#20. Paralogistic-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="paralogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
fun = nlm(f=ff,p=c(0.5,1.7,43.6,0.8,1.9,163.3))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(t)),digits=3,format="f"),"","\n")
```

Table 8

```
#1. Mixture of Inverse Gamma and Lognormal
arg1 = list(shape = 3.817001, scale = 93.639645)
arg2 = list(meanlog = 4.074110, sdlog = 1.40406)
(var_95 <- qmixt(0.95, 4.427584, spec1 = "invgamma", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 4.427584, spec1 = "invgamma", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 4.427584, spec1 = "invgamma", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 513.978, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1382.649, xb = Inf)$Q)

f = function(x) qmixt(x, 4.427584, spec1 = "invgamma", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*57.11469)
(es_99 <- (1/(1-0.99))*25.58952)

#2. Mixture of Inverse Gaussian-Lognormal
arg1 = list(mean = 29.833773, shape = 107.979847)
arg2 = list(meanlog = 4.079389, sdlog = 1.396125)
(var_95 <- qmixt(0.95, 4.726666, spec1 = "invgauss", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 4.726666, spec1 = "invgauss", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 4.726666, spec1 = "invgauss", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 514.5513, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1373.992, xb = Inf)$Q)
```

```
f = function(x) qmixt(x, 4.726666, spec1 = "invgauss", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.74265)
(es_99 <- (1/(1-0.99))*25.29364)

#3. Generalized Pareto-Lognormal
arg1 = list(shape1 = 4.17122, shape2 = 45.298997, scale = 2.285418)
arg2 = list(meanlog = 4.076518, sdlog = 1.403076)

(var_95 <- qmixt(0.95, 4.443455, spec1 = "genpareto", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 4.443455, spec1 = "genpareto", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 4.443455, spec1 = "genpareto", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 514.6313, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1383.396, xb = Inf)$Q)

f = function(x) qmixt(x, 4.443455, spec1 = "genpareto", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*57.14275)
(es_99 <- (1/(1-0.99))*25.58811)

#4. Inverse Paralogistic-Lognormal
arg1 = list(shape = 2.416601, scale = 17.1673347)
arg2 = list(meanlog = 4.081333, sdlog = 1.40688)

(var_95 <- qmixt(0.95, 4.079912, spec1 = "invparalogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 4.079912, spec1 = "invparalogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 4.079912, spec1 = "invparalogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 515.065, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1390.588, xb = Inf)$Q)

f = function(x) qmixt(x, 4.079912, spec1 = "invparalogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*57.4507)
(es_99 <- (1/(1-0.99))*25.80522)

#5. Inverse Weibull-Lognormal
arg1 = list(shape = 2.003406, scale = 23.086563)
arg2 = list(meanlog = 4.060686, sdlog = 1.409812)

(var_95 <- qmixt(0.95, 4.186258, spec1 = "invweibull", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0, 5000)))
(var_99 <- qmixt(0.99, 4.186258, spec1 = "invweibull", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0, 5000)))

f = function(x) x*dmixt(x, 4.186258, spec1 = "invweibull", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 510.5537, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1378.039, xb = Inf)$Q)

f = function(x) qmixt(x, 4.186258, spec1 = "invweibull", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0, 500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.96529)
(es_99 <- (1/(1-0.99))*25.6072)

#6. Inverse Burr-Lognormal
arg1 = list(shape1 = 4.538114, shape2 = 2.194075, scale = 12.092573)
arg2 = list(meanlog = 4.073553, sdlog = 1.410138)

(var_95 <- qmixt(0.95, 4.041081, spec1 = "invburr", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 4.041081, spec1 = "invburr", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 4.041081, spec1 = "invburr", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 513.7763, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1389.68, xb = Inf)$Q)

f = function(x) qmixt(x, 4.041081, spec1 = "invburr", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*57.42941)
(es_99 <- (1/(1-0.99))*25.83769)

#7. Loglogistic-Lognormal
arg1 = list(shape = 2.933196, scale = 26.404735)
arg2 = list(meanlog = 4.108482, sdlog = 1.397402)

(var_95 <- qmixt(0.95, 3.906552, spec1 = "llogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 3.906552, spec1 = "llogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 3.906552, spec1 = "llogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 517.6727, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1391.508, xb = Inf)$Q)

f = function(x) qmixt(x, 3.906552, spec1 = "llogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*57.4527)
(es_99 <- (1/(1-0.99))*25.70782)
```

```
#8. Burr-Lognormal
arg1 = list(shape1 = 0.5816701, shape2 = 3.4262997, scale = 21.4809239)
arg2 = list(meanlog = 4.0956802, sdlog = 1.4086898)

(var_95 <- qmixt(0.95, 3.6252924, spec1 = "burr", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 3.6252924, spec1 = "burr", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 3.6252924, spec1 = "burr", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 516.7761, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1400.85, xb = Inf)$Q)

f = function(x) qmixt(x, 3.6252924, spec1 = "burr", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*57.88985)
(es_99 <- (1/(1-0.99))*26.07824)

#9. Gamma-Lognormal
arg1 = list(shape = 4.242796, scale = 6.583723)
arg2 = list(meanlog = 4.091793, sdlog = 1.385785)

(var_95 <- qmixt(0.95,4.769101, spec1 = "gamma", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99,4.769101, spec1 = "gamma", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 4.769101, spec1 = "gamma", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 513.2486, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1360.228, xb = Inf)$Q)

f = function(x) qmixt(x, 4.769101, spec1 = "gamma", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.1523)
(es_99 <- (1/(1-0.99))*24.89426)

#10. Paralogistic-Lognormal
arg1 = list(shape = 2.402993, scale = 40.259897)
arg2 = list(meanlog = 4.126997, sdlog = 1.382258)

(var_95 <- qmixt(0.95, 4.002015, spec1 = "paralogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 4.002015, spec1 = "paralogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 4.002015, spec1 = "paralogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 516.781, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1373.679, xb = Inf)$Q)

f = function(x) qmixt(x, 4.002015, spec1 = "paralogis", arg1 = arg1, spec2 = "lnorm", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*56.68135)
(es_99 <- (1/(1-0.99))*25.15685)

#11. Lognormal-Weibull
arg1 = list(meanlog = 4.1530586, sdlog = 1.3612375)
arg2 = list(shape = 1.9636305, scale = 29.5290399)

(var_95 <- qmixt(0.95, 0.2536428, spec1 = "lnorm", arg1 = arg1, spec2 = "weibull", arg2 = arg2, interval = c(0,
5000)))
(var_99 <- qmixt(0.99, 0.2536428, spec1 = "lnorm", arg1 = arg1, spec2 = "weibull", arg2 = arg2, interval = c(0,
5000)))

f = function(x) x*dmixt(x, 0.2536428, spec1 = "lnorm", arg1 = arg1, spec2 = "weibull", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 512.5239, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1342.931, xb = Inf)$Q)

f = function(x) qmixt(x, 0.2536428, spec1 = "lnorm", arg1 = arg1, spec2 = "weibull", arg2 = arg2, interval =
c(0, 500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*55.3707)
(es_99 <- (1/(1-0.99))*24.3157)

#12. Loglogistic-Generalized Pareto
arg1 = list(shape = 1.7759239, scale = 29.8995942)
arg2 = list(shape1 = 2.2800372, shape2 = 1.5140997, scale = 247.5429968)

(var_95 <- qmixt(0.95, 0.5237485, spec1 = "llogis", arg1 = arg1, spec2 = "genpareto", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 0.5237485, spec1 = "llogis", arg1 = arg1, spec2 = "genpareto", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 0.5237485, spec1 = "llogis", arg1 = arg1, spec2 = "genpareto", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 518.1971, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1349.011, xb = Inf)$Q)

f = function(x) qmixt(x, 0.5237485, spec1 = "llogis", arg1 = arg1, spec2 = "genpareto", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*57.8629)
(es_99 <- (1/(1-0.99))*26.64527)

#13. Generalized Pareto-Paralogistic
arg1 = list(shape1 = 2.057002, shape2 = 1.614187, scale = 168.379147)
arg2 = list(shape = 1.732624, scale = 39.130163)

(var_95 <- qmixt(0.95, 1.172619, spec1 = "genpareto", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0,5000)))
```

```
(var_99 <- qmixt(0.99, 1.172619, spec1 = "genpareto", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 1.172619, spec1 = "genpareto", arg1 = arg1, spec2 = "paralogis", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 511.4846, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1363.65, xb = Inf)$Q)

f = function(x) qmixt(x, 1.172619, spec1 = "genpareto", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval
= c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*59.52253)
(es_99 <- (1/(1-0.99))*28.4636)

#14. Loglogistic-Paralogistic
arg1 = list(shape = 1.7640922, scale = 30.5826145)
arg2 = list(shape = 1.3814343, scale = 228.333892)

(var_95 <- qmixt(0.95, 0.4576104, spec1 = "llogis", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 0.4576104, spec1 = "llogis", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 0.4576104, spec1 = "llogis", arg1 = arg1, spec2 = "paralogis", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 513.2153, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1372.248, xb = Inf)$Q)

f = function(x) qmixt(x, 0.4576104, spec1 = "llogis", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*61.18144)
(es_99 <- (1/(1-0.99))*30.0293)

#15. Burr-Loglogistic
arg1 = list(shape1 = 1.634153, shape2 = 1.255751, scale = 256.478155)
arg2 = list(shape = 1.783554, scale = 30.326036)

(var_95 <- qmixt(0.95, 1.951761, spec1 = "burr", arg1 = arg1, spec2 = "llogis", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 1.951761, spec1 = "burr", arg1 = arg1, spec2 = "llogis", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 1.951761, spec1 = "burr", arg1 = arg1, spec2 = "llogis", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 515.4531, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1351.424, xb = Inf)$Q)

f = function(x) qmixt(x, 1.951761, spec1 = "burr", arg1 = arg1, spec2 = "llogis", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*59.0708)
(es_99 <- (1/(1-0.99))*27.99213)

#16. Paralogistic-Paralogistic
arg1 = list(shape = 1.7332581, scale = 40.174668)
arg2 = list(shape = 1.3464215, scale = 180.1914829)

(var_95 <- qmixt(0.95, 0.7779231, spec1 = "paralogis", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0, 5000)))
(var_99 <- qmixt(0.99, 0.7779231, spec1 = "paralogis", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0, 5000)))

f = function(x) x*dmixt(x, 0.7779231, spec1 = "paralogis", arg1 = arg1, spec2 = "paralogis", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 507.1849, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1384.181, xb = Inf)$Q)

f = function(x) qmixt(x, 0.7779231, spec1 = "paralogis", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval
= c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*62.77795)
(es_99 <- (1/(1-0.99))*31.76835)

#17. Burr-Burr
arg1 = list(shape1 = 0.9576539, shape2 = 1.7883638, scale = 29.6828387)
arg2 = list(shape1 = 1.6629942, shape2 = 1.2497661, scale = 264.2923267)

(var_95 <- qmixt(0.95, 0.4903487, spec1 = "burr", arg1 = arg1, spec2 = "burr", arg2=arg2, interval = c(0,5000)))
(var_99 <- qmixt(0.99, 0.4903487, spec1 = "burr", arg1 = arg1, spec2 = "burr", arg2=arg2, interval = c(0,5000)))

f = function(x) x*dmixt(x, 0.4903487, spec1 = "burr", arg1 = arg1, spec2 = "burr", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 516.1666, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1351.721, xb = Inf)$Q)

f = function(x) qmixt(x, 0.4903487, spec1 = "burr", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*59.09217)
(es_99 <- (1/(1-0.99))*27.98084)

#18. Inverse Gamma-Paralogistic
arg1 = list(shape = 1.749536, scale = 294.635837)
arg2 = list(shape = 1.683582, scale = 43.796042)

(var_95 <- qmixt(0.95, 2.431252, spec1 = "invgamma", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 2.431252, spec1 = "invgamma", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 2.431252, spec1 = "invgamma", arg1 = arg1, spec2 = "paralogis", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 504.4198, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1436.604, xb = Inf)$Q)

f = function(x) qmixt(x, 2.431252, spec1 = "invgamma", arg1 = arg1, spec2 = "paralogis", arg2 = arg2, interval =
c(0,500000))
```

```
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*66.37499)
(es_99 <- (1/(1-0.99))*34.96704)

#19. Inverse Gamma-Generalized Pareto
arg1 = list(shape = 1.838566, scale = 334.995954)
arg2 = list(shape1 = 6.273924, shape2 = 1.970403, scale = 107.747145)

(var_95 <- qmixt(0.95, 2.562574, spec1 = "invgamma", arg1 = arg1, spec2 = "genpareto", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 2.562574, spec1 = "invgamma", arg1 = arg1, spec2 = "genpareto", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x,2.562574, spec1 = "invgamma", arg1 = arg1, spec2 = "genpareto", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 507.2773, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1402.212, xb = Inf)$Q)

f = function(x) qmixt(x, 2.562574, spec1 = "invgamma", arg1 = arg1, spec2 = "genpareto", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*63.41137)
(es_99 <- (1/(1-0.99))*32.20809)

#20. Paralogistic-Burr
arg1 = list(shape = 1.737833, scale = 40.1805425)
arg2 = list(shape1 = 1.3951233, shape2 = 1.3193864, scale = 184.5420102)

(var_95 <- qmixt(0.95, 0.7970739, spec1 = "paralogis", arg1 = arg1, spec2 = "burr", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 0.7970739, spec1 = "paralogis", arg1 = arg1, spec2 = "burr", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 0.7970739, spec1 = "paralogis", arg1 = arg1, spec2 = "burr", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 507.4376, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 1376.181, xb = Inf)$Q)

f = function(x) qmixt(x, 0.7970739, spec1 = "paralogis", arg1 = arg1, spec2 = "burr", arg2 = arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*62.04721)
(es_99 <- (1/(1-0.99))*31.09602)
```

Table 9 and Table A4

```
#1. Weibull-Inverse Weibull
set.seed(1234)
a <- CompDist("weibull", "invweibull")
a_fit_d <- FitParameters(a, d, iterations = 100)
a_loglik_d <- logLik(a_fit_d, d)
a_NLL_d <- as.numeric((-1)*a_loglik_d)
a_AIC_d <- 2*a_NLL_d + 2*(length(a@dist1@pars) + length(a@dist2@pars))
a_BIC_d <- 2*a_NLL_d + log(length(d))*(length(a@dist1@pars) + length(a@dist2@pars))
c(a_NLL_d, a_AIC_d, a_BIC_d)
Refit_a <- RefitParameters(a_fit_d, d)

#2. Paralogistic-Inverse Weibull
set.seed(1234)
b <- CompDist("paralogis", "invweibull")
b_fit_d <- FitParameters(b, d, iterations = 100)
b_loglik_d <- logLik(b_fit_d, d)
b_NLL_d <- as.numeric((-1)*b_loglik_d)
b_AIC_d <- 2*b_NLL_d + 2*(length(b@dist1@pars) + length(b@dist2@pars))
b_BIC_d <- 2*b_NLL_d + log(length(d))*(length(b@dist1@pars) + length(b@dist2@pars))
c(b_NLL_d, b_AIC_d, b_BIC_d)
b_Refit_d <- RefitParameters(b_fit_d, d)

#3. Inverse Burr-Inverse Weibull
set.seed(1234)
c <- CompDist("invburr","invweibull")
c_fit_d <- FitParameters(c, d, iterations = 100)
c_loglik_d <- logLik(c_fit_d, d)
c_NLL_d <- as.numeric((-1)*c_loglik_d)
c_AIC_d <- 2*c_NLL_d + 2*(length(c@dist1@pars) + length(c@dist2@pars))
c_BIC_d <- 2*c_NLL_d + log(length(d))*(length(c@dist1@pars) + length(c@dist2@pars))
c(c_NLL_d, c_AIC_d, c_BIC_d)
c_Refit_d <- RefitParameters(c_fit_d, d)

#4. Weibull-Inverse Paralogistic
set.seed(1234)
e <- CompDist("weibull", "invparalogis")
e_fit_d <- FitParameters(e, d, iterations = 100)
e_loglik_d <- logLik(e_fit_d, d)
e_NLL_d <- as.numeric((-1)*e_loglik_d)
e_AIC_d <- 2*e_NLL_d + 2*(length(b@dist1@pars) + length(e@dist2@pars))
e_BIC_d <- 2*e_NLL_d + log(length(d))*(length(e@dist1@pars) + length(b@dist2@pars))
c(e_NLL_d, e_AIC_d, e_BIC_d)
e_Refit_d <- RefitParameters(e_fit_d, d)

#5. Inverse Burr-Inverse Paralogistic: 1 hour
set.seed(1234)
f <- CompDist("invburr", "invparalogis")
f_fit_d <- FitParameters(f, d, iterations = 100)
f_loglik_d <- logLik(f_fit_d, d)
f_NLL_d <- as.numeric((-1)*f_loglik_d)
f_AIC_d <- 2*f_NLL_d + 2*(length(f@dist1@pars) + length(f@dist2@pars))
f_BIC_d <- 2*f_NLL_d + log(length(d))*(length(f@dist1@pars) + length(f@dist2@pars))
c(f_NLL_d, f_AIC_d, f_BIC_d)
f_Refit_d <- RefitParameters(f_fit_d, d)

#6. Paralogistic-Inverse Paralogistic:
set.seed(1234)
```

```
g <- CompDist("paralogis", "invparalogis")
g_fit_d <- FitParameters(g, d, iterations = 100)
g_loglik_d <- logLik(g_fit_d, d)
g_NLL_d <- as.numeric((-1)*g_loglik_d)
g_AIC_d <- 2*g_NLL_d + 2*(length(g@dist1@pars) + length(g@dist2@pars))
g_BIC_d <- 2*g_NLL_d + log(length(d))*(length(g@dist1@pars) + length(g@dist2@pars))
c(g_NLL_d, g_AIC_d, g_BIC_d)
g_Refit_d <- RefitParameters(g_fit_d, d)

#7. Weibull-Loglogistic
set.seed(1234)
h <- CompDist("weibull", "llogis")
h_fit_d <- FitParameters(h, d, iterations = 100)
h_loglik_d <- logLik(h_fit_d, d)
h_NLL_d <- as.numeric((-1)*h_loglik_d)
h_AIC_d <- 2*h_NLL_d + 2*(length(h@dist1@pars) + length(h@dist2@pars))
h_BIC_d <- 2*h_NLL_d + log(length(d))*(length(h@dist1@pars) + length(h@dist2@pars))
c(h_NLL_d, h_AIC_d, h_BIC_d)
h_Refit_d <- RefitParameters(h_fit_d, d)

#8. Inverse Burr-Loglogistic: First set of parameters not same as Grun and Miljkovic.
set.seed(1234)
i <- CompDist("invburr", "llogis")
i_fit_d <- FitParameters(i, d, iterations = 100)
i_loglik_d <- logLik(i_fit_d, d)
i_NLL_d <- as.numeric((-1)*i_loglik_d)
i_AIC_d <- 2*i_NLL_d + 2*(length(i@dist1@pars) + length(i@dist2@pars))
i_BIC_d <- 2*i_NLL_d + log(length(d))*(length(i@dist1@pars) + length(i@dist2@pars))
c(i_NLL_d, i_AIC_d, i_BIC_d)
i_Refit_d <- RefitParameters(i_fit_d, d)

#9. Paralogistic-Loglogistic
set.seed(1234)
j <- CompDist("paralogis", "llogis")
j_fit_d <- FitParameters(j, d, iterations = 100)
j_loglik_d <- logLik(j_fit_d, d)
j_NLL_d <- as.numeric((-1)*j_loglik_d)
j_AIC_d <- 2*j_NLL_d + 2*(length(j@dist1@pars) + length(j@dist2@pars))
j_BIC_d <- 2*j_NLL_d + log(length(d))*(length(j@dist1@pars) + length(j@dist2@pars))
c(j_NLL_d, j_AIC_d, j_BIC_d)
j_Refit_d <- RefitParameters(j_fit_d, d)

#10. Loglogistic-Inverse Weibull
set.seed(1234)
k <- CompDist("llogis", "invweibull")
k_fit_d <- FitParameters(k, d, iterations = 100)
k_loglik_d <- logLik(k_fit_d, d)
k_NLL_d <- as.numeric((-1)*k_loglik_d)
k_AIC_d <- 2*k_NLL_d + 2*(length(k@dist1@pars) + length(k@dist2@pars))
k_BIC_d <- 2*k_NLL_d + log(length(d))*(length(k@dist1@pars) + length(k@dist2@pars))
c(k_NLL_d, k_AIC_d, k_BIC_d)
k_Refit_d <- RefitParameters(k_fit_d, d)

#11. Weibull-Burr
set.seed(1234)
l <- CompDist("weibull", "burr")
l_fit_d <- FitParameters(l, d, iterations = 100)
l_loglik_d <- logLik(l_fit_d, d)
l_NLL_d <- as.numeric((-1)*l_loglik_d)
l_AIC_d <- 2*l_NLL_d + 2*(length(l@dist1@pars) + length(l@dist2@pars))
l_BIC_d <- 2*l_NLL_d + log(length(d))*(length(l@dist1@pars) + length(l@dist2@pars))
c(l_NLL_d, l_AIC_d, l_BIC_d)
l_Refit_d <- RefitParameters(l_fit_d, d)

#12. Paralogistic-Burr
set.seed(1234)
m <- CompDist("paralogis", "burr")
m_fit_d <- FitParameters(m, d, iterations = 100)
m_loglik_d <- logLik(m_fit_d, d)
m_NLL_d <- as.numeric((-1)*m_loglik_d)
m_AIC_d <- 2*m_NLL_d + 2*(length(m@dist1@pars) + length(m@dist2@pars))
m_BIC_d <- 2*m_NLL_d + log(length(d))*(length(m@dist1@pars) + length(m@dist2@pars))
c(m_NLL_d, m_AIC_d, m_BIC_d)
m_Refit_d <- RefitParameters(m_fit_d, d)

#13. Inverse Burr-Burr
set.seed(1234)
n <- CompDist("invburr", "burr")
n_fit_d <- FitParameters(n, d, iterations = 100)
n_loglik_d <- logLik(n_fit_d, d)
n_NLL_d <- as.numeric((-1)*n_loglik_d)
n_AIC_d <- 2*n_NLL_d + 2*(length(n@dist1@pars) + length(n@dist2@pars))
n_BIC_d <- 2*n_NLL_d + log(length(d))*(length(n@dist1@pars) + length(n@dist2@pars))
c(n_NLL_d, n_AIC_d, n_BIC_d)
n_Refit_d <- RefitParameters(n_fit_d, d)

#14. Log-logistic-Inverse Paralogistic
set.seed(1234)
o <- CompDist("llogis", "invparalogis")
o_fit_d <- FitParameters(o, d, iterations = 100)
o_loglik_d <- logLik(o_fit_d, d)
o_NLL_d <- as.numeric((-1)*o_loglik_d)
o_AIC_d <- 2*o_NLL_d + 2*(length(o@dist1@pars) + length(o@dist2@pars))
o_BIC_d <- 2*o_NLL_d + log(length(d))*(length(o@dist1@pars) + length(o@dist2@pars))
c(o_NLL_d, o_AIC_d, o_BIC_d)
o_Refit_d <- RefitParameters(o_fit_d, d)

#15. Inverse Burr-Inverse Gamma
set.seed(1234)
p <- CompDist("invburr", "invgamma")
p_fit_d <- FitParameters(p, d, iterations = 100)
p_loglik_d <- logLik(p_fit_d, d)
p_NLL_d <- as.numeric((-1)*p_loglik_d)
p_AIC_d <- 2*p_NLL_d + 2*(length(p@dist1@pars) + length(p@dist2@pars))
p_BIC_d <- 2*p_NLL_d + log(length(d))*(length(p@dist1@pars) + length(p@dist2@pars))
c(p_NLL_d, p_AIC_d, p_BIC_d)
p_Refit_d <- RefitParameters(p_fit_d, d)
```

```
#16. Paralogistic-Inverse Gamma
set.seed(1234)
q <- CompDist("paralogis", "invgamma")
q_fit_d <- FitParameters(q, d, iterations = 100)
q_loglik_d <- logLik(q_fit_d, d)
q_NLL_d <- as.numeric((-1)*q_loglik_d)
q_AIC_d <- 2*q_NLL_d + 2*(length(q@dist1@pars) + length(q@dist2@pars))
q_BIC_d <- 2*q_NLL_d + log(length(d))*(length(q@dist1@pars) + length(q@dist2@pars))
c(q_NLL_d, q_AIC_d, q_BIC_d)
q_Refit_d <- RefitParameters(q_fit_d, d)

#17. Loglogistic-Loglogistic
set.seed(1234)
r <- CompDist("llogis", "llogis")
r_fit_d <- FitParameters(r, d, iterations = 100)
r_loglik_d <- logLik(r_fit_d, d)
r_NLL_d <- as.numeric((-1)*r_loglik_d)
r_AIC_d <- 2*r_NLL_d + 2*(length(r@dist1@pars) + length(r@dist2@pars))
r_BIC_d <- 2*r_NLL_d + log(length(d))*(length(r@dist1@pars) + length(r@dist2@pars))
c(r_NLL_d, r_AIC_d, r_BIC_d)
r_Refit_d <- RefitParameters(r_fit_d, d)

#18. Weibull-Paralogistic
set.seed(1234)
s <- CompDist("weibull", "paralogis")
s_fit_d <- FitParameters(s, d, iterations = 100)
s_loglik_d <- logLik(s_fit_d, d)
s_NLL_d <- as.numeric((-1)*s_loglik_d)
s_AIC_d <- 2*s_NLL_d + 2*(length(s@dist1@pars) + length(s@dist2@pars))
s_BIC_d <- 2*s_NLL_d + log(length(d))*(length(s@dist1@pars) + length(s@dist2@pars))
c(s_NLL_d, s_AIC_d, s_BIC_d)
s_Refit_d <- RefitParameters(s_fit_d, d)

#19. Paralogistic-Paralogistic
set.seed(1234)
u <- CompDist("paralogis", "paralogis")
u_fit_d <- FitParameters(u, d, iterations = 100)
u_loglik_d <- logLik(u_fit_d, d)
u_NLL_d <- as.numeric((-1)*u_loglik_d)
u_AIC_d <- 2*u_NLL_d + 2*(length(u@dist1@pars) + length(u@dist2@pars))
u_BIC_d <- 2*u_NLL_d + log(length(d))*(length(u@dist1@pars) + length(u@dist2@pars))
c(u_NLL_d, u_AIC_d, u_BIC_d)
u_Refit_d <- RefitParameters(u_fit_d, d)

#20. Inverse Burr-Paralogistic
set.seed(1234)
v <- CompDist("invburr", "paralogis")
v_fit_d <- FitParameters(v, d, iterations = 100)
v_loglik_d <- logLik(v_fit_d, d)
v_NLL_d <- as.numeric((-1)*v_loglik_d)
v_AIC_d <- 2*u_NLL_d + 2*(length(v@dist1@pars) + length(v@dist2@pars))
v_BIC_d <- 2*u_NLL_d + log(length(d))*(length(v@dist1@pars) + length(v@dist2@pars))
c(v_NLL_d, v_AIC_d, v_BIC_d)
v_Refit_d <- RefitParameters(v_fit_d, d)
```

Table 10

```
#1. Weibull-Inverse Weibull
pinvweibull(0.955, shape = 1.555, rate = 1.102)

(var_95 <- qinvweibull(0.3970714+((1-0.3970714)/9.854)*(0.95*(1+9.854)-1), shape = 1.555, rate = 1.102))
(var_99 <- qinvweibull(0.3970714+((1-0.3970714)/9.854)*(0.99*(1+9.854)-1), shape = 1.555, rate = 1.102))

f = function(x) qinvweibull(0.3970714+((1-0.3970714)/9.854)*(x*(1+9.854)-1), shape = 1.555, rate = 1.102)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.132366)
(es_99 <- (1/(1-0.99))*0.6389962)

#2. Paralogistic-Inverse Weibull
pinvweibull(0.957, shape = 1.554, rate = 1.105)

(var_95 <- qinvweibull(0.3997936+((1-0.3997936)/9.688)*(0.95*(1+9.688)-1),shape = 1.554, rate = 1.105))
(var_99 <- qinvweibull(0.3997936+((1-0.3997936)/9.688)*(0.99*(1+9.688)-1),shape = 1.554, rate = 1.105))

f = function(x) qinvweibull(0.3997936+((1-0.3997936)/9.688)*(x*(1+9.688)-1),shape = 1.554, rate = 1.105)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.134356)
(es_99 <- (1/(1-0.99))*0.6405415)

#3. Inverse Burr-Inverse Weibull
pinvweibull(0.934, shape = 1.557, rate = 1.096)

(var_95 <- qinvweibull(0.3812731+((1-0.3812731)/12.609)*(0.95*(1+12.609)-1), shape = 1.557, rate = 1.096))
(var_99 <- qinvweibull(0.3812731+((1-0.3812731)/12.609)*(0.99*(1+12.609)-1), shape = 1.557, rate = 1.096))

f = function(x) qinvweibull(0.3812731+((1-0.3812731)/12.609)*(x*(1+12.609)-1), shape = 1.557, rate = 1.096)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.12871)
(es_99 <- (1/(1-0.99))*0.6360959)

#4. Weibull-Inverse Paralogistic
pinvparalogis(0.961, shape = 1.567, rate = 1.777)

(var_95 <- qinvparalogis(0.5694846+((1-0.5694846)/9.256)*(0.95*(1+9.256)-1), shape = 1.567, rate = 1.777))
(var_99 <- qinvparalogis(0.5694846+((1-0.5694846)/9.256)*(0.99*(1+9.256)-1), shape = 1.567, rate = 1.777))

f = function(x) qinvparalogis(0.5694846+((1-0.5694846)/9.256)*(x*(1+9.256)-1), shape = 1.567, rate = 1.777)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
```

```
(es_95 <- (1/(1-0.95))*1.120001)
(es_99 <- (1/(1-0.99))*0.6272805)

#5. Inverse Burr-Inverse Paralogistic
pinvparalogis(0.928, shape = 1.567, rate = 1.775)

(var_95 <- qinvparalogis(0.554168+((1-0.554168)/14.086)*(0.95*(1+14.086)-1), shape = 1.567, rate = 1.775))
(var_99 <- qinvparalogis(0.554168+((1-0.554168)/14.086)*(0.99*(1+14.086)-1), shape = 1.567, rate = 1.775))

f = function(x) qinvparalogis(0.554168+((1-0.554168)/14.086)*(x*(1+14.086)-1), shape = 1.567, rate = 1.775)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.120577)
(es_99 <- (1/(1-0.99))*0.6276047)

#6. Paralogistic-Inverse Paralogistic
pinvparalogis(0.964, shape = 1.566, rate = 1.787)

(var_95 <- qinvparalogis(0.5732206+((1-0.5732206)/9.054)*(0.95*(1+9.054)-1), shape = 1.566, rate = 1.787))
(var_99 <- qinvparalogis(0.5732206+((1-0.5732206)/9.054)*(0.99*(1+9.054)-1), shape = 1.566, rate = 1.787))

f = function(x) qinvparalogis(0.5732206+((1-0.5732206)/9.054)*(x*(1+9.054)-1), shape = 1.566, rate = 1.787)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.121136)
(es_99 <- (1/(1-0.99))*0.6283171)

#7. Weibull-Loglogistic
pllogis(0.964, shape = 1.568, scale = 0.68)

(var_95 <- qllogis(0.6334926+((1-0.6334926)/9.03)*(0.95*(1+9.03)-1), shape = 1.568, scale = 0.68))
(var_99 <- qllogis(0.6334926+((1-0.6334926)/9.03)*(0.99*(1+9.03)-1), shape = 1.568, scale = 0.68))

f = function(x) qllogis(0.6334926+((1-0.6334926)/9.03)*(x*(1+9.03)-1), shape = 1.568, scale = 0.68)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.121059)
(es_99 <- (1/(1-0.99))*0.627531)

#8. Inverse Burr-Loglogistic
pllogis(0.928, shape = 1.57, scale = 0.688)

(var_95 <- qllogis(0.6153391+((1-0.6153391)/14.02)*(0.95*(1+14.02)-1),shape = 1.57, scale = 0.688))
(var_99 <- qllogis(0.6153391+((1-0.6153391)/14.02)*(0.99*(1+14.02)-1),shape = 1.57, scale = 0.688))

f = function(x) qllogis(0.6153391+((1-0.6153391)/14.02)*(x*(1+14.02)-1),shape = 1.57, scale = 0.688)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.119344)
(es_99 <- (1/(1-0.99))*0.6257742)

#9. Paralogistic-Loglogistic
pllogis(0.965, shape = 1.567, scale = 0.678)

(var_95 <- qllogis(0.6348594+((1-0.6348594)/8.906)*(0.95*(1+8.906)-1), shape = 1.567, scale = 0.678))
(var_99 <- qllogis(0.6348594+((1-0.6348594)/8.906)*(0.99*(1+8.906)-1), shape = 1.567, scale = 0.678))

f = function(x) qllogis(0.6348594+((1-0.6348594)/8.906)*(x*(1+8.906)-1), shape = 1.567, scale = 0.678)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.122486)
(es_99 <- (1/(1-0.99))*0.6287367)

#10. Loglogistic-Inverse Weibull
pinvweibull(0.976, shape = 1.547, rate = 1.13)

(var_95 <- qinvweibull(0.4234048+((1-0.4234048)/8.216)*(0.95*(1+8.216)-1), shape = 1.547, rate = 1.13))
(var_99 <- qinvweibull(0.4234048+((1-0.4234048)/8.216)*(0.99*(1+8.216)-1), shape = 1.547, rate = 1.13))

f = function(x) qinvweibull(0.4234048+((1-0.4234048)/8.216)*(x*(1+8.216)-1), shape = 1.547, rate = 1.13)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.146925)
(es_99 <- (1/(1-0.99))*0.6506441)

#11. Weibull-Burr
pburr(0.976, shape1 = 0.395, shape2 = 3.646, rate = 1.182)

(var_95 <- qburr(0.3229169+((1-0.3229169)/11.135)*(0.95*(1+11.135)-1), shape1 = 0.395, shape2 = 3.646, rate =
1.182))
(var_99 <- qburr(0.3229169+((1-0.3229169)/11.135)*(0.99*(1+11.135)-1), shape1 = 0.395, shape2 = 3.646, rate =
1.182))

f = function(x) qburr(0.3229169+((1-0.3229169)/11.135)*(x*(1+11.135)-1), shape1 = 0.395, shape2 = 3.646, rate =
1.182)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.36833)
(es_99 <- (1/(1-0.99))*0.8366859)

#12. Paralogistic-Burr
pburr(0.947, shape1 = 0.394, shape2 = 3.649, rate = 1.182)

(var_95 <- qburr(0.3040174+((1-0.3040174)/11.043)*(0.95*(1+11.043)-1), shape1 = 0.394, shape2 = 3.649, rate =
1.182))
(var_99 <- qburr(0.3040174+((1-0.3040174)/11.043)*(0.99*(1+11.043)-1), shape1 = 0.394, shape2 = 3.649, rate =
1.182))

f = function(x) qburr(0.3040174+((1-0.3040174)/11.043)*(x*(1+11.043)-1), shape1 = 0.394, shape2 = 3.649, rate =
1.182)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.352281)
(es_99 <- (1/(1-0.99))*0.8284575)

#13. Inverse Burr-Burr
pburr(0.932, shape1 = 0.406, shape2 = 3.549, rate = 1.19)
```

```
(var_95 <- qburr(0.3042881+((1-0.3042881)/13.251)*(0.95*(1+13.251)-1), shape1 = 0.406, shape2 = 3.549, rate =
1.19))
(var_99 <- qburr(0.3042881+((1-0.3042881)/13.251)*(0.99*(1+13.251)-1), shape1 = 0.406, shape2 = 3.549, rate =
1.19))

f = function(x) qburr(0.3042881+((1-0.3042881)/13.251)*(x*(1+13.251)-1), shape1 = 0.406, shape2 = 3.549, rate =
1.19)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.343002)
(es_99 <- (1/(1-0.99))*0.8207399)

#14. Loglogistic-Inverse Paralogistic
pinvparalogis(0.98, shape = 1.561, rate = 1.819)

(var_95 <- qinvparalogis(0.5877439+((1-0.5877439)/7.876)*(0.95*(1+7.876)-1), shape = 1.561, rate = 1.819))
(var_99 <- qinvparalogis(0.5877439+((1-0.5877439)/7.876)*(0.99*(1+7.876)-1), shape = 1.561, rate = 1.819))

f = function(x) qinvparalogis(0.5877439+((1-0.5877439)/7.876)*(x*(1+7.876)-1), shape = 1.561, rate = 1.819)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.129276)
(es_99 <- (1/(1-0.99))*0.6349331)

#15. Inverse Burr-Inverse Gamma
pinvgamma(0.928, shape = 1.641, scale = 1.148)

(var_95 <- qinvgamma(0.5309094+((1-0.5309094)/13.945)*(0.95*(1+13.945)-1), shape = 1.641, scale = 1.148))
(var_99 <- qinvgamma(0.5309094+((1-0.5309094)/13.945)*(0.99*(1+13.945)-1), shape = 1.641, scale = 1.148))

f = function(x) qinvgamma(0.5309094+((1-0.5309094)/13.945)*(x*(1+13.945)-1), shape = 1.641, scale = 1.148)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.07234)
(es_99 <- (1/(1-0.99))*0.5792421)

#16. Paralogistic-Inverse Gamma
pinvgamma(0.967, shape = 1.635, scale = 1.119)

(var_95 <- qinvgamma(0.5589081+((1-0.5589081)/8.753)*(0.95*(1+8.753)-1), shape = 1.635, scale = 1.119))
(var_99 <- qinvgamma(0.5589081+((1-0.5589081)/8.753)*(0.99*(1+8.753)-1), shape = 1.635, scale = 1.119))

f = function(x) qinvgamma(0.5589081+((1-0.5589081)/8.753)*(x*(1+8.753)-1), shape = 1.635, scale = 1.119)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.077436)
(es_99 <- (1/(1-0.99))*0.5839005)

#17. Loglogistic-Loglogistic
pllogis(0.981, shape = 1.562, scale = 0.666)

(var_95 <- qllogis(0.6467847+((1-0.6467847)/7.761)*(0.95*(1+7.761)-1), shape = 1.562, scale = 0.666))
(var_99 <- qllogis(0.6467847+((1-0.6467847)/7.761)*(0.99*(1+7.761)-1), shape = 1.562, scale = 0.666))

f = function(x) qllogis(0.6467847+((1-0.6467847)/7.761)*(x*(1+7.761)-1), shape = 1.562, scale = 0.666)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.131076)
(es_99 <- (1/(1-0.99))*0.6356011)

#18. Weibull-Paralogistic
pparalogis(0.968, shape = 1.267, rate = 1.607)

(var_95 <- qparalogis(0.7224799+((1-0.7224799)/8.66)*(0.95*(1+8.66)-1), shape = 1.267, rate = 1.607))
(var_99 <- qparalogis(0.7224799+((1-0.7224799)/8.66)*(0.99*(1+8.66)-1), shape = 1.267, rate = 1.607))

f = function(x) qparalogis(0.7224799+((1-0.7224799)/8.66)*(x*(1+8.66)-1), shape = 1.267, rate = 1.607)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.096683)
(es_99 <- (1/(1-0.99))*0.6019009)

#19. Paralogistic-Paralogistic
pparalogis(0.969, shape = 1.266, rate = 1.611)

(var_95 <- qparalogis(0.7230963+((1-0.7230963)/8.551)*(0.95*(1+8.551)-1), shape = 1.266, rate = 1.611))
(var_99 <- qparalogis(0.7230963+((1-0.7230963)/8.551)*(0.99*(1+8.551)-1), shape = 1.266, rate = 1.611))

f = function(x) qparalogis(0.7230963+((1-0.7230963)/8.551)*(x*(1+8.551)-1), shape = 1.266, rate = 1.611)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.02008)
(es_99 <- (1/(1-0.99))*0.6057635)

#20. Inverse Burr-Paralogistic
pparalogis(0.928, shape = 1.27, rate = 1.559)

(var_95 <- qparalogis(0.7026201+((1-0.7026201)/13.775)*(0.95*(1+13.775)-1),shape = 1.27, rate = 1.559))
(var_99 <- qparalogis(0.7026201+((1-0.7026201)/13.775)*(0.99*(1+13.775)-1),shape = 1.27, rate = 1.559))

f = function(x) qparalogis(0.7026201+((1-0.7026201)/13.775)*(x*(1+13.775)-1),shape = 1.27, rate = 1.559)
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.087297)
(es_99 <- (1/(1-0.99))*0.5940785)
```

Table 11 and Table A5

```
nlogl.mixt <- function(p, phi, spec1, arg1, spec2, arg2){
  tt <- 1.0e20
  if(all(p>0)){
    tt <- -sum(log( dmixt(d, phi, spec1, arg1, spec2, arg2) ))
  }
  return(tt)
}

#1. Burr-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="burr",
                          arg1=c(shape=p[2],shape2=p[3],scale=p[4]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[5], shape2 = p[6], scale = p[7]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*7,digits=3,format="f")," & ",
    formatC(2*fun$minimum+7*log(length(d)),digits=3,format="f"),"","\n")

#2. Inverse Weibull-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invweibull",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(0.5,1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#3. Loglogistic-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="llogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1,1))
fun = nlminb(start = c(2,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#4. Inverse Paralogistic-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invparalogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#5. Paralogistic-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="paralogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$objective,digits=3,format="f")," & ",
    formatC(2*fun$objective+2*6,digits=3,format="f")," & ",
    formatC(2*fun$objective+6*log(length(d)),digits=3,format="f"),"","\n")

#6. Inverse Burr-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invburr",
                          arg1=c(shape=p[2],shape2=p[3],scale=p[4]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[5], shape2 = p[6], scale = p[7]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1,1,1))
fun = nlm(f=ff,p=c(4.06,1.34,4.24,1.6,0.03,43.15,0.86))
fun = nlm(f=ff,p=c(4.36,136.275,10.65,0.6,0.1575,9.08,1.165))
fun = nlminb(start=c(1,1,1,1,1,1,1),objective=ff)
fun = nlminb(start=c(4.35,647.3,10.59,0.5135,0.1576,9.1,1.265),objective=ff)
cat(formatC(fun$objective,digits=3,format="f")," & ",
    formatC(2*fun$objective+2*7,digits=3,format="f")," & ",
    formatC(2*fun$objective+7*log(length(d)),digits=3,format="f"),"","\n")

#7. Gamma-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="gamma",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(10,1,1,1,1,1))
fun = nlminb(start=c(100,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#8. Lognormal-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="lnorm",
                          arg1=c(meanlog=p[2],sdlog=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(20,1,1,1,1,1))
fun = nlminb(start=c(5,1,1,1,1,1), objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#9. Generalized Pareto-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="genpareto",
                          arg1=c(shape=p[2],shape2=p[3],scale=p[4]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[5], shape2 = p[6], scale = p[7]))
```

```
}
fun = nlm(f=ff,p=c(2,1,1,1,1,1,1))
fun = nlm(f=ff,p=c(4.75,12.25,10.915,2.09,0.03,39.96,0.868))
fun = nlminb(start=c(1,1,1,1,1,1,1),objective = ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*7,digits=3,format="f")," & ",
    formatC(2*fun$minimum+7*log(length(d)),digits=3,format="f"),"","\n")

#10. Inverse Gaussian-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invgauss",
                          arg1=c(mean=p[2],shape=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(0.050,1,1,1,1,1))
fun = nlm(f=ff,p=c(2,1,1,1,1,1))
fun = nlm(f=ff,p=c(4.78,2.15,8.79,0.03,38.87,0.8698))
fun = nlminb(start=c(10,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#11. Inverse Gamma-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invgamma",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1,1))
fun = nlminb(start=c(10,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#12. Inverse Exponential-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invexp",
                          arg1=c(rate=p[2]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[3], shape2 = p[4], scale = p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(d)),digits=3,format="f"),"","\n")

#13. Exponential-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="exp",
                          arg1=c(rate=p[2]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[3], shape2 = p[4], scale = p[5]))
}
fun = nlm(f=ff,p=c(2,1,1,1,1))
fun = nlminb(start=c(0.1000,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(d)),digits=3,format="f"),"","\n")

#14. Inverse Pareto-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invpareto",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#15. Weibull-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="weibull",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")

#16. Pareto-Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="pareto",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "burr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
fun = nlm(f=ff,p=c(121.669,37.7,100.9595,0.05,25.25,0.89))
fun = nlminb(start=c(5,1,1,1,1,1), objective=ff)
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$objective,digits=3,format="f")," & ",
    formatC(2*fun$objective+2*6,digits=3,format="f")," & ",
    formatC(2*fun$objective+6*log(length(d)),digits=3,format="f"),"","\n")

#17. Inverse Weibull-Inverse Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invweibull",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2 = "invburr",
                          arg2 = c(shape1 = p[4], shape2 = p[5], scale = p[6]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1,1))
fun = nlminb(start=c(1,1,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*6,digits=3,format="f")," & ",
    formatC(2*fun$minimum+6*log(length(d)),digits=3,format="f"),"","\n")
```

```
#18.Inverse Paralogistic-Inverse Weibull
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invparalogis",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2="invweibull",
                          arg2=c(shape=p[4], scale=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
fun = nlminb(start=c(5,1,1,1,1),objective=ff)
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(d)),digits=3,format="f"),"","\n")

#19.Inverse Weibull-Inverse Gamma
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invweibull",
                          arg1=c(shape=p[2],scale=p[3]),
                          spec2="invgamma",
                          arg2=c(shape=p[4], scale=p[5]))
}
fun = nlm(f=ff,p=c(1,1,1,1,1))
cat(formatC(fun$minimum,digits=3,format="f")," & ",
    formatC(2*fun$minimum+2*5,digits=3,format="f")," & ",
    formatC(2*fun$minimum+5*log(length(d)),digits=3,format="f"),"","\n")

#20. Inverse Burr-Inverse Burr
ff=function(p){nlogl.mixt(p,phi=p[1],spec1="invburr",
                          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 = nlm(f=ff,p=c(2,1,1,1,1,1,1))
fun = nlm(f=ff,p=c(0.94,26.2,1.65,0.285,26.85,4.4,0.535))
fun = nlm(f=ff,p=c(1.3,5.9,1.7,0.855,77.3,4,0.39))
fun = nlm(f=ff, p=c(1.3,5.9265,1.72,0.855,130,4,0.34))
fun = nlminb(start=c(1,1,1,1,1,1,1),objective=ff)
fun = nlminb(start=c(1.3418175,5.7896785,1.7163448,0.8736045,327.2263893,3.9578678,0.2676891),objective=ff)
fun = nlminb(start=c(1.3390541,5.8468006,1.7147778,0.8659774,4119.4996996952,3.9533317,0.1408207),objective=ff)
cat(formatC(fun$objective,digits=3,format="f")," & ",
    formatC(2*fun$objective+2*7,digits=3,format="f")," & ",
    formatC(2*fun$objective+7*log(length(d)),digits=3,format="f"),"","\n")
```

Table 12

```
#1. Burr-Burr
arg1 = list(shape1 = 0.27061939, shape2 = 6.6542979, scale = 1.28692176)
arg2 = list(shape1 = 0.02566555, shape2 = 49.30787319, scale = 0.85734704)

(var_95 <- qmixt(0.95, 2.15651789, spec1 = "burr", arg1 = arg1, spec2 = "burr", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 2.15651789, spec1 = "burr", arg1 = arg1, spec2 = "burr", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 2.15651789, spec1 = "burr", arg1 = arg1, spec2 = "burr", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.259708, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 27.00048, xb = Inf)$Q)

f = function(x) qmixt(x, 2.15651789, spec1 = "burr", arg1 = arg1, spec2 = "burr", arg2 = arg2, interval =
c(0,500000))
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.736799)
(es_99 <- (1/(1-0.99))*1.197213)

#2. Inverse Weibull-Burr
arg1 = list(shape = 10.570056, scale = 0.9465657)
arg2 = list(shape1 = 0.1576963, shape2 = 9.0711111, scale = 1.1657858)
(var_95 <- qmixt(0.95, 4.3468277, spec1="invweibull", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 4.3468277, spec1="invweibull", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x, 4.3468277, spec1="invweibull", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.18958, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 25.22828, xb = Inf)$Q)

f = function(x) qmixt(x, 4.3468277, spec1="invweibull", arg1=arg1, spec2="burr", arg2=arg2,
interval=c(0,500000))
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.360692)
(es_99 <- (1/(1-0.99))*0.8383318)

#3. Loglogistic-Burr
arg1 = list(shape = 5.10281163, scale = 1.71858111)
arg2 = list(shape1 = 0.02822428, shape2 = 41.88349844, scale = 0.86454294)

(var_95 <- qmixt(0.95, 4.65623928, spec1="llogis", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 4.65623928, spec1="llogis", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x,4.65623928, spec1="llogis", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 9.249825, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 36.07215, xb = Inf)$Q)

f = function(x) qmixt(x, 4.65623928, spec1="llogis", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*3.000218)
(es_99 <- (1/(1-0.99))*2.34127)

#4. Inverse Paralogistic-Burr
arg1 = list(shape = 11.8502534,scale = 0.7665989)
arg2 = list(shape1 = 0.1540596, shape2 = 9.2682745, scale = 1.1496672)
(var_95 <- qmixt(0.95, 4.778478, spec1="invparalogis", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 4.778478, spec1="invparalogis", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x, 4.778478, spec1="invparalogis", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.202522, xb = Inf)$Q)
```

```
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 25.32025, xb = Inf)$Q)

f = function(x)  qmixt(x, 4.778478, spec1 = "invparalogis", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*1.3686663)
(es_99 <- (1/(1-0.99))*0.8449814)

#5. Paralogistic-Burr
arg1 = list(shape=4.81129696,scale=2.43002926)
arg2 = list(shape1 = 0.03058942, shape2 = 38.62887878, scale = 0.86826956)
(var_95 <- qmixt(0.95, 6.52149259, spec1="paralogis", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 6.52149259, spec1="paralogis", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x,6.52149259, spec1="paralogis", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 9.710989, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 37.91328, xb = Inf)$Q)

f = function(x) qmixt(x, 6.52149259, spec1="paralogis", arg1=arg1, spec2="burr", arg2=arg2,
interval=c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*3.158765)
(es_99 <- (1/(1-0.99))*2.466466)

#6. Inverse Burr-Burr
arg1=list(shape1=136.2773441,shape2=10.6678338,scale=0.5970769)
arg2=list(shape1=0.1574276,shape2=9.0857699,scale=1.1645700)
(var_95 <- qmixt(0.95, 4.3781875, spec1 = "invburr", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 4.3781875, spec1 = "invburr", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x,4.3781875, spec1 = "invburr", arg1 = arg1, spec2 = "burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.190157, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 25.23263, xb = Inf)$Q)

f = function(x)  qmixt(x, 4.3781875, spec1 = "invburr", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*1.361075)
(es_99 <- (1/(1-0.99))*0.8386531)

#7. Gamma-Burr
arg1=list(shape=8.77784174,scale=0.20292783)
arg2=list(shape1=0.02982605,shape2=39.776500105,scale=0.86769652)
(var_95 <- qmixt(0.95, 5.40768286, spec1="gamma", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 5.40768286, spec1="gamma", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x, 5.40768286, spec1="gamma", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 9.395001, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 36.48051, xb = Inf)$Q)

f = function(x) qmixt(x, 5.40768286, spec1="gamma", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000000))
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*2.990194)
(es_99 <- (1/(1-0.99))*2.322167)

#8. Lognormal-Burr
arg1 = list(meanlog = 0.6348901,sdlog=0.4416963)
arg2=list(shape1=0.0308202,shape2=39.4882545,scale=0.868625)
(var_95 <- qmixt(0.95, 4.7594083, spec1="lnorm", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 4.7594083, spec1="lnorm", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x,4.7594083, spec1="lnorm", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.712113, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 32.6674, xb = Inf)$Q)

f = function(x)  qmixt(x, 4.7594083, spec1="lnorm", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000000))
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*2.440865)
(es_99 <- (1/(1-0.99))*1.831835)

#9. Generalized Pareto-Burr
arg1 = list(shape1 = 12.24999991, shape2 = 10.9150002, scale = 2.09000093)
arg2 = list(shape1 = 0.03028511, shape2 = 39.96000014, scale = 0.86791042)
(var_95 <- qmixt(0.95, 4.74999997, spec1 = "genpareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 4.74999997, spec1 = "genpareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 4.74999997, spec1 = "genpareto", arg1 = arg1, spec2 = "burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.814147, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 33.30742, xb = Inf)$Q)

f = function(x) qmixt(x, 4.74999997, spec1 = "genpareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,5000000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*2.536204)
(es_99 <- (1/(1-0.99))*1.917686)

#10. Inverse Gaussian-Burr
arg1 = list(mean=2.15659222,shape=8.79238099)
arg2 = list(shape1=0.03159803,shape2=38.82372936,scale=0.86988248)
(var_95 <- qmixt(0.95, 4.78374957, spec1="invgauss", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 4.78374957, spec1="invgauss", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x,4.78374957, spec1="invgauss", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.57802, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 31.81101, xb = Inf)$Q)

f = function(x) qmixt(x,4.78374957, spec1="invgauss", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,500000))
```

```
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*2.31736)
(es_99 <- (1/(1-0.99))*1.721001)

#11. Burr-Inverse Gamma
arg1 = list(shape1=0.07510464,shape2=16.83561383,scale=0.90207502)
arg2 = list(shape=52.27825116,scale=89.52247241)
(var_95 <- qmixt(0.95, 0.06986236, spec1="burr", arg1=arg1, spec2="invgamma", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 0.06986236, spec1="burr", arg1=arg1, spec2="invgamma", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x,0.06986236, spec1="burr", arg1=arg1, spec2="invgamma", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 9.140959, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 32.64263, xb = Inf)$Q)

f = function(x) qmixt(x, 0.06986236, spec1="burr", arg1=arg1, spec2="invgamma", arg2=arg2,
interval=c(0,5000000))
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*2.185457)
(es_99 <- (1/(1-0.99))*1.560866)

#12. Inverse Exponential-Burr
arg1=list(rate=0.98115112)
arg2=list(shape1=0.04907394,shape2=25.7279575,scale=0.89032348)
(var_95 <- qmixt(0.95, 134.93741153, spec1="invexp", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 134.93741153, spec1="invexp", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x,134.93741153, spec1="invexp", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 9.607268, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 34.55494, xb = Inf)$Q)

f = function(x) qmixt(x, 134.93741153, spec1="invexp", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,50000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95)))
(es_99 <- (1/(1-0.99)))

#13. Exponential-Burr
arg1=list(rate=0.3686359)
arg2=list(shape1=0.0499864,shape2=25.2806679,scale=0.8907614)
(var_95 <- qmixt(0.95, 119.5613637, spec1="exp", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 119.5613637, spec1="exp", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x,119.5613637, spec1="exp", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 9.509637, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 33.85066, xb = Inf)$Q)

f = function(x) qmixt(x, 119.5613637, spec1="exp", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*2.270363)
(es_99 <- (1/(1-0.99))*1.622234)

#14. Inverse Pareto-Burr
arg1 = list(shape=66.4230655,scale=0.01557516)
arg2 = list(shape1=0.04911131,shape2=25.71051278,scale=0.89034719)
(var_95 <- qmixt(0.95, 134.98233103, spec1 = "invpareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 134.98233103, spec1 = "invpareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x,134.98233103, spec1 = "invpareto", arg1 = arg1, spec2 = "burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 9.707152, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 34.55399, xb = Inf)$Q)

f = function(x) qmixt(x, 134.98233103, spec1 = "invpareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)

#15. Weibull-Burr
arg1 = list(shape=19.9053707,scale=0.955047)
arg2 = list(shape1=0.1291055,shape2=10.7568911,scale=1.0426773)
(var_95 <- qmixt(0.95, 10.3812404, spec1="weibull", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))
(var_99 <- qmixt(0.99, 10.3812404, spec1="weibull", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,5000)))

f = function(x) x*dmixt(x, 10.3812404, spec1="weibull", arg1=arg1, spec2="burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.437421, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 26.88499, xb = Inf)$Q)

f = function(x) qmixt(x, 10.3812404, spec1="weibull", arg1=arg1, spec2="burr", arg2=arg2, interval=c(0,500000))
integrate(f,lower=0.95,upper=1)
integrate(f,lower=0.99,upper=1)
(es_95 <- (1/(1-0.95))*1.507003)
(es_99 <- (1/(1-0.99))*0.9603827)

#16. Pareto-Burr
arg1 = list(shape=37.699999999,scale=100.95950001)
arg2 = list(shape1=0.05005141,shape2=25.25000166,scale=0.89081915)
(var_95 <- qmixt(0.95, 121.669, spec1 = "pareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval = c(0,5000)))
(var_99 <- qmixt(0.99, 121.669, spec1 = "pareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval = c(0,5000)))

f = function(x) x*dmixt(x,121.669, spec1 = "pareto", arg1 = arg1, spec2 = "burr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 9.513091, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 33.84644, xb = Inf)$Q)

f = function(x) qmixt(x, 121.669, spec1 = "pareto", arg1 = arg1, spec2 = "burr", arg2=arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*2.269716)
(es_99 <- (1/(1-0.99))*1.621489)

#17. Inverse Weibull-Inverse Burr
arg1 = list(shape=3.9523126,scale=1.1563687)
arg2 = list(shape1=5.8475764,shape2=1.7146380,scale=0.8659477)
```

```r
(var_95 <- qmixt(0.95,0.7464899,spec1="invweibull",arg1=arg1,spec2="invburr",arg2=arg2,interval=c(0,5000)))
(var_99 <- qmixt(0.99,0.7464899,spec1="invweibull",arg1=arg1,spec2="invburr",arg2=arg2,interval=c(0,5000)))

f = function(x) x*dmixt(x,0.7464899,spec1="invweibull",arg1=arg1,spec2="invburr",arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.15435, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 21.50685, xb = Inf)$Q)

f = function(x) qmixt(x,0.7464899,spec1="invweibull",arg1=arg1,spec2="invburr",arg2=arg2,interval=c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*1.005594)
(es_99 <- (1/(1-0.99))*0.5188418)

#18. Inverse Paralogistic-Inverse Weibull
arg1 = list(shape = 1.901227, scale = 2.383503)
arg2 = list(shape = 3.468841, scale = 1.209924)

(var_95 <- qmixt(0.95, 2.287888, spec1 = "invparalogis", arg1 = arg1, spec2 = "invweibull", arg2 = arg2,
interval = c(0,5000)))
(var_99 <- qmixt(0.99, 2.287888, spec1 = "invparalogis", arg1 = arg1, spec2 = "invweibull", arg2 = arg2,
interval = c(0,5000)))

f = function(x) x*dmixt(x, 2.287888, spec1 = "invparalogis", arg1 = arg1, spec2 = "invweibull", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.133526, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 19.91652, xb = Inf)$Q)

f = function(x) qmixt(x, 2.287888, spec1 = "invparalogis", arg1 = arg1, spec2 = "invweibull", arg2 = arg2,
interval = c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*0.8943955)
(es_99 <- (1/(1-0.99))*0.4233884)

#19. Inverse Weibull-Inverse Gamma
arg1 = list(shape = 3.6923749, scale = 1.1829716)
arg2 = list(shape = 1.9112942, scale = 4.8491822)

(var_95 <- qmixt(0.95, 0.6334155, spec1 = "invweibull", arg1 = arg1, spec2 = "invgamma", arg2 = arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 0.6334155, spec1 = "invweibull", arg1 = arg1, spec2 = "invgamma", arg2 = arg2, interval =
c(0,5000)))

f = function(x) x*dmixt(x, 0.6334155, spec1 = "invweibull", arg1 = arg1, spec2 = "invgamma", arg2 = arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.534143, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 22.15739, xb = Inf)$Q)

f = function(x) qmixt(x, 0.6334155, spec1 = "invweibull", arg1 = arg1, spec2 = "invgamma", arg2 = arg2, interval
= c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*0.9919471)
(es_99 <- (1/(1-0.99))*0.4834541)

#20. Inverse Burr-Inverse Burr
arg1 = list(shape1 = 5.8996729, shape2 = 1.7150508, scale = 0.8599175)
arg2 = list(shape1 = 249.9669934, shape2 = 3.9671754, scale = 0.2875419)
(var_95 <- qmixt(0.95, 1.332008, spec1 = "invburr", arg1 = arg1, spec2 = "invburr", arg2=arg2, interval =
c(0,5000)))
(var_99 <- qmixt(0.99, 1.332008, spec1 = "invburr", arg1 = arg1, spec2 = "invburr", arg2=arg2, interval =
c(0,5000)))
f = function(x) x*dmixt(x, 1.332008, spec1 = "invburr", arg1 = arg1, spec2 = "invburr", arg2=arg2)
(es_95 <- (1/(1-0.95))*quadinf(f, xa = 8.151446, xb = Inf)$Q)
(es_99 <- (1/(1-0.99))*quadinf(f, xa = 21.49243, xb = Inf)$Q)

f = function(x) qmixt(x, 1.332008, spec1 = "invburr", arg1 = arg1, spec2 = "invburr", arg2=arg2, interval =
c(0,500000))
integrate(f, lower = 0.95, upper = 1)
integrate(f, lower = 0.99, upper = 1)
(es_95 <- (1/(1-0.95))*1.004816)
(es_99 <- (1/(1-0.99))*0.5183102)
```