

```

# PDF
dAPTZLD <- function(x, alpha, theta) {
  if (alpha <= 0 || theta <= 0) {
    return("The parameter must be positive")
  }
  else if (alpha==1) {
    return((0.5*theta*(1+2*theta+theta*x)*exp(-theta*x)/(theta+1)))
  }
  else {
    term1 <- log(alpha) / (alpha - 1)
    term2 <- theta / (2 * (1 + theta))
    term3 <- (1 + 2 * theta + theta * x) * exp(-theta * x)
    exponent <- 1 - (1 + (theta * x)) / (2 * (1 + theta)) * exp(-theta * x)
    term4 <- alpha^exponent
    return(term1 * term2 * term3 * term4)
  }
}
# CDF
pAPTZLD <- function(x, alpha, theta) {
  if (alpha <= 0 || theta <= 0) {
    return("The parameter must be positive")
  }
  else if (alpha==1) {
    return((1 + theta * x) / (2 * (1 + theta))) * exp(-theta * x))
  }
  else {
    c = 1 - (1 + theta * x) / (2 * (1 + theta)) * exp(-theta * x)
    return((exp(c*log(alpha))-1)/(alpha - 1))
  }
}
# SF
sAPTZLD <- function(x, alpha, theta){
  if (alpha <= 0 || theta <= 0) {
    return("The parameter must be positive")
  }
  else if (alpha==1) return(pAPTZLD(x,alpha,theta))
  else return(1 - pAPTZLD(x,alpha,theta))
}
# HRF
hAPTZLD <- function(x, alpha, theta){
  if (alpha <= 0 || theta <= 0) {
    return("The parameter must be positive")
  } else
    return(dAPTZLD(x,alpha,theta)/sAPTZLD(x,alpha,theta))
}
# quantile function
qAPTZLD=function(p,alpha,theta){
  if (alpha <= 0 || theta <= 0) {
    return("The parameter must be positive")
  } else {
    u1 = function(x,alpha,theta) pAPTZLD(x,alpha,theta) - p
    return(uniroot(u1, c(0,1e06), tol = 0.0000000001, alpha=alpha, theta=theta)
$root)
  }
}
# pseudo-random number generator
rAPTZLD =function(n,alpha,theta) {
  x=numeric(n)
  for (i in 1:n) x[i]=qAPTZLD(runif(1,0,1),alpha,theta)
  return(sort(x))
}
# ordinary moments
mAPTZLD=function(k,alpha,theta) {
  return(integral(function(x) x^k*dAPTZLD(x,alpha,theta), 0, Inf, reltol =
1e-12, method = "Simpson"))
}

```

```

# incomplete moments
imAPTZLD=function(k,t, alpha,theta) {
  return(integral(function(x) x^k*dAPTZLD(x,alpha,theta), 0, t, reltol = 1e-12,
method = "Simpson"))
}
# characteristics
chAPTZLD=function(alpha,theta){
  x=numeric(5)
  # mean
  x[1]=mAPTZLD(1,alpha,theta)
  #variance
  x[2]=mAPTZLD(2,alpha,theta)-x[1]^2
  # coefficient of variation
  x[3]=sqrt(x[2])/x[1]
  w1=mAPTZLD(3,alpha,theta)-3*mAPTZLD(1,alpha,theta)*mAPTZLD(2,alpha,theta)+2*
x[1]^3
  #skewness
  x[4]=w1/(x[2])^(1.5)
  w2=mAPTZLD(4,alpha,theta)-4*mAPTZLD(1,alpha,theta)*mAPTZLD(3,alpha,theta)
+6*mAPTZLD(1,alpha,theta)^2*
    mAPTZLD(2,alpha,theta)-3*x[1]^4
  #kurtosis
  x[5]=w2/(x[2])^2
  return(x)
}
#moment generating function
mgfAPTZLD=function(t,alpha,theta) {
  return(integral(function(x) exp(t*x)*dAPTZLD(x,alpha,theta), 0, Inf, reltol =
1e-12, method = "Simpson"))
}
#pdf of order statistics
dosAPTZLD=function(x,r,n,alpha,theta) {
  pdf=dAPTZLD(x,alpha,theta)
  cdf=pAPTZLD(x,alpha,theta)
  return(fact(n)/fact(r-1)/fact(n-r)*pdf*cdf^(r-1)*(1-cdf)^(n-r))
}
# moments of order statistics
mosAPTZLD=function(k,r,n,alpha,theta) {
return(integral(function(x) x^k*dosAPTZLD(x,r,n,alpha,theta), 0, Inf, reltol =
1e-12, method = "Simpson"))
}

```