```
> restart;
  read("c:/appl/appl7.txt");
```

PROCEDURES:

AllPermutations(n), AllCombinations(n, k), Benford(X), BootstrapRV(Data),
CDF:CHF:HF:IDF:PDF:SF(X, [x])), CoefOfVar(X), Convolution(X, Y),
ConvolutionIID(X, n), CriticalPoint(X, prob), Determinant(MATRIX), Difference(X, Y),
Display(X), ExpectedValue(X, [g]), KSTest(X, Data, Parameters), Kurtosis(X),
Maximum(X, Y), MaximumIID(X, n), Mean(X), MGF(X), Minimum(X, Y),
MinimumIID(X, n), Mixture(MixParameters, MixRVs),
MLE(X, Data, Parameters, [Rightcensor]), MLENHPP(X, Data, Parameters, obstime),
MLEWeibull(Data, [Rightcensor]), MOM(X, Data, Parameters),
NextCombination(Previous, size), NextPermutation(Previous), OrderStat(X, n, r, ["wo"]),
PlotDist(X, [low], [high]), PlotEmpCDF(Data, [low], [high]),
PlotEmpCIF(Data, [low], [high]), PlotEmpSF(Data, Censor),
PlotEmpVsFittedCDF(X, Data, Parameters, [low], [high]),
PlotEmpVsFittedCDF(X, Data, Parameters, [low], [high]),
PlotEmpVsFittedSF(X, Data, Parameters, Censor, low, high),
PPPlot(X, Data, Parameters), Product(X, Y), ProductIID(X, n),
QQPlot(X, Data, Parameters), RangeStat(X, n, ["wo"]), Skewness(X), Transform(X, g),
Truncate(X, low, high), Variance(X), VerifyPDF(X)

Procedure Notation:
X and Y are random variables
Greek letters are numeric or symbolic parameters
x is numeric or symbolic
n and r are positive integers, n >= r
low and high are numeric
g is a function
Brackets [] denote optional parameters
"double quotes" denote character strings
MATRIX is a 2 x 2 array of random variables
A capitalized parameter indicates that it must be
entered as a list --> ex. Data := [1, 12.4, 34, 52.45, 63]

Variate Generation:

ArcTanVariate(alpha, phi), BinomialVariate(n, p, m), ExponentialVariate(lambda),
NormalVariate(mu, sigma), UniformVariate(), WeibullVariate(lambda, kappa, m)

DATA SETS:

BallBearing, HorseKickFatalities, Hurricane, MP6, RatControl, RatTreatment, USSHalfBeak

ArcSinRV(), ArcTanRV(alpha, phi), BetaRV(alpha, beta), CauchyRV(a, alpha), ChiRV(n),

ChiSquareRV(n), ErlangRV(lambda, n), ErrorRV(mu, alpha, d), ExponentialRV(lambda), ExponentialPowerRV(lambda, kappa), ExtremeValueRV(alpha, beta), FRV(n1, n2), GammaRV(lambda, kappa), GeneralizedParetoRV(gamma, delta, kappa), GompertzRV(delta, kappa), HyperbolicSecantRV(), HyperExponentialRV(p, l), HypoExponentialRV(l), IDBRV(gamma, delta, kappa), InverseGaussianRV(lambda, mu), InvertedGammaRV(alpha, beta), KSRV(n), LaPlaceRV(omega, theta), LogGammaRV(alpha, beta), LogisticRV(kappa, lambda), LogLogisticRV(lambda, kappa), LogNormalRV(mu, sigma), LomaxRV(kappa, lambda), MakehamRV(gamma, delta, kappa), MuthRV(kappa), NormalRV(mu, sigma), ParetoRV(lambda, kappa), RayleighRV(lambda), StandardCauchyRV(), StandardNormalRV(), StandardTriangularRV(m), StandardUniformRV(), TRV(n), TriangularRV(a, m, b), UniformRV(a, b), WeibullRV(lambda, kappa)

```
Error, attempting to assign to `DataSets` which is protected.
Try declaring `local DataSets`; see ?protect for details.
```

```
> bf := InvertedGammaRV(2,3);
  bfname := "InvertedGammaRV(2,3)";
```

$$bf := \left[\left[x \to \frac{1}{9}\ \frac{e^{-\frac{1}{3x}}}{x^3}\right], [0, \infty], ["\text{Continuous}", "\text{PDF}"]\right]$$

$$bfname := "\text{InvertedGammaRV(2,3)}" \quad \textbf{(1)}$$

```
> #plot(1/csch(t)+1, t = 0..0.0010);
  #plot(diff(1/csch(t),t), t=0..0.0010);
  #limit(1/csch(t), t=0);
> solve(exp(-t) = y, t);
```

$$-\ln(y) \quad \textbf{(2)}$$

```
> # discarded  -ln(t + 1),  t-> csch(t),t->arccsch(t),t -> tan(t),
> #name of the file for latex output
  filename := "C:/LatexOutput/Trash.tex";


  glist := [t -> t^2 , t -> sqrt(t),  t -> 1/t,  t -> arctan(t),  t
  -> exp(t), t -> ln(t), t -> exp(-t), t -> -ln(t),  t -> ln(t+1),
  t -> 1/(ln(t+2)), t -> tanh(t), t -> sinh(t), t -> arcsinh(t),
  t-> csch(t+1),t->arccsch(t+1), t-> 1/tanh(t+1), t->  1/sinh(t+1),
   t->  1/arcsinh(t+1), t->  1/csch(t)+1, t-> tanh(1/t), t->csch
  (1/t), t-> arccsch(1/t), t-> arctanh(1/t)  ]:

  base := t -> PDF(bf, t):

  print(base(x)):


  #begin loopint through transformations
  for i from 15 to 15 do
  #for i from 1 to 3 do
```

```
   print( "i is", i, "  ----------------------------------------
----------------------------------------------------------------
-------------");

   g := glist[i]:
   l := bf[2][1];
   u := bf[2][2];
   Temp := Transform(bf, [[unapply(g(x), x)],[l,u]]);

  #terminal output

  PlotDist(PDF(Temp), 0, 40);
  PlotDist(HF(Temp), 0, 40);


od;
```

$$filename := \text{"C:/LatexOutput/Trash.tex"}$$

$$\frac{1}{9}\,\frac{e^{-\frac{1}{3x}}}{x^3}$$

"i is", 15,
" ----------
----------"

$$g := t \to \operatorname{arccsch}(t+1)$$

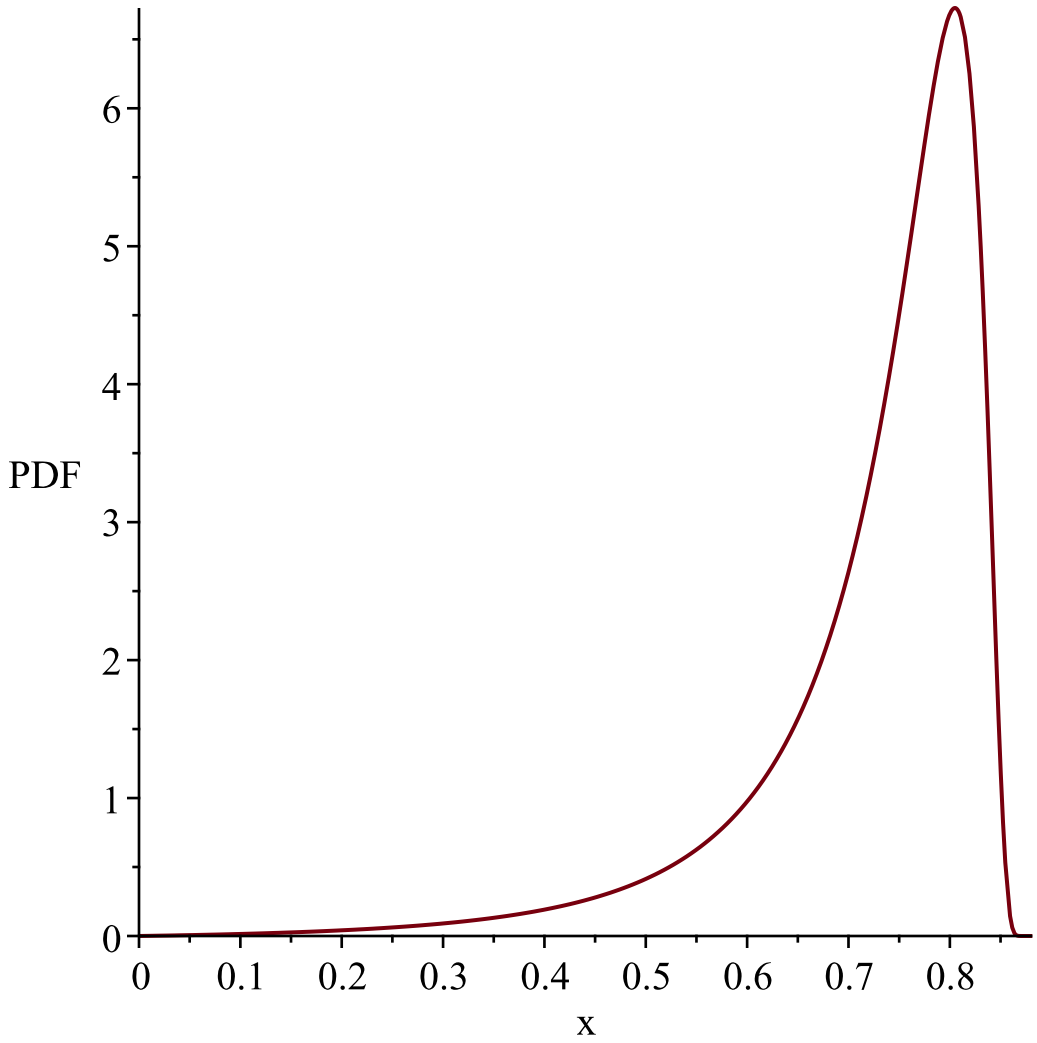
$$l := 0$$

$$u := \infty$$

$$Temp := \left[\left[y\sim \to -\frac{1}{9}\,\frac{e^{\frac{1}{3}\,\frac{\sinh(y\sim)}{\sinh(y\sim)-1}}\cosh(y\sim)\,\sinh(y\sim)}{\cosh(y\sim)^2\sinh(y\sim)-3\cosh(y\sim)^2+2\sinh(y\sim)+2}\right],\left[0,\ln\left(1+\sqrt{2}\right)\right],\left[\text{"Continuous"},\text{"PDF"}\right]\right]$$

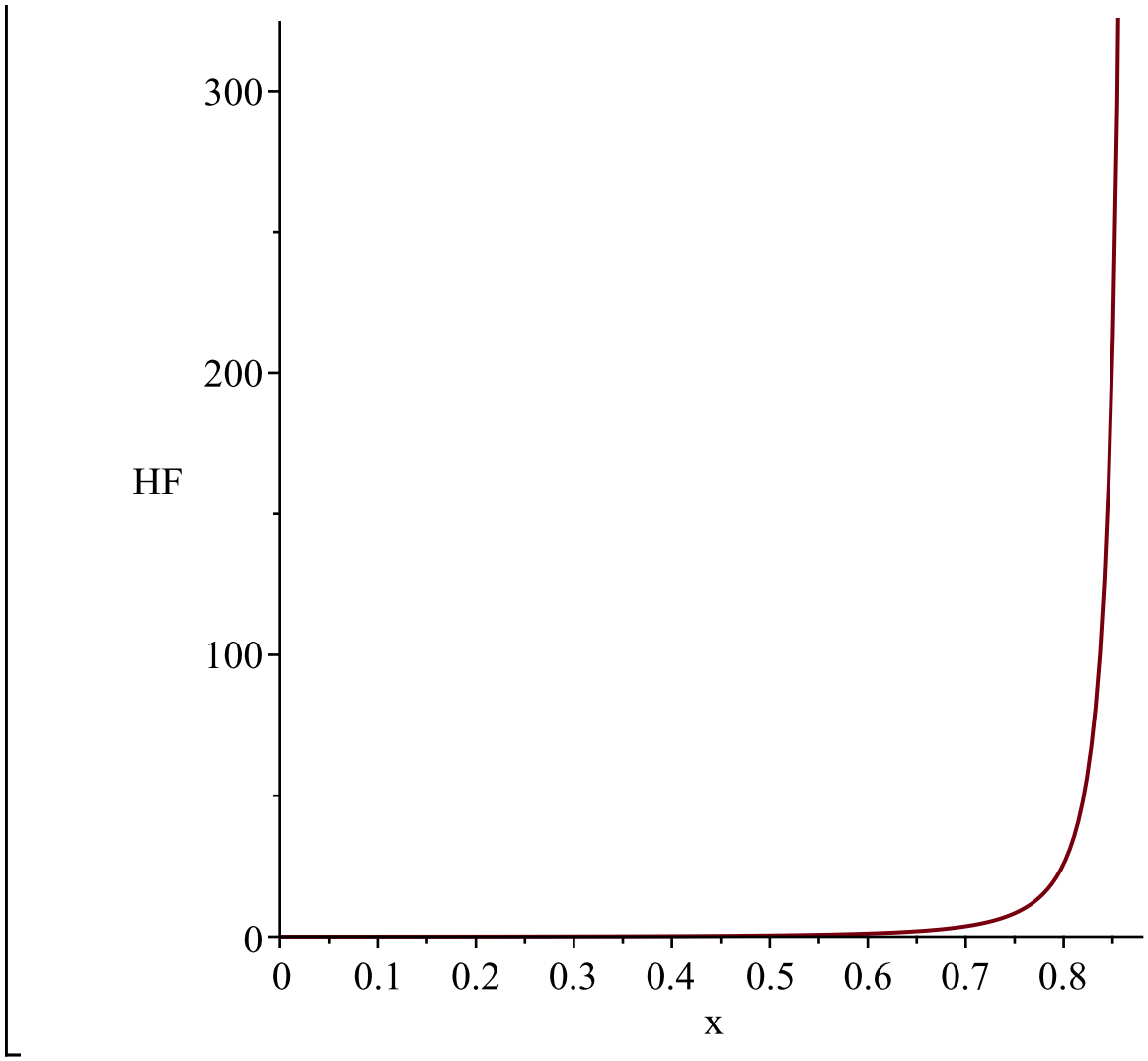
WARNING(PlotDist): High value provided by user, 40 *is greater than maximum support value of the random variable,* $\ln\left(1+\sqrt{2}\right)$

Resetting high to RV's maximum support value



WARNING(PlotDist): High value provided by user, 40 *is greater than maximum support value of the random variable*, $\ln\left(1+\sqrt{2}\right)$

Resetting high to RV's maximum support value


300
200
HF
100
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x