

# The GIG\_GNIG Workspace

You will need to install and `load` the `Mpfr` package to run the functions in this Workspace

In this workspace there is a function `GIG_help`, which may be used to obtain help on the user functions, which are `GIGpdf`, `GIGcdf`, `GIGquant`, `plotGIGpdf`, `plotGIGcdf`, `GNIGpdf`, `GNIGcdf`, `GNIGquant`, `plotGNIGpdf`, `plotGNIGcdf`

```
> GIG_help(GIGpdf)
*****
* ==> GIGpdf(r,l,z,prec,precp) <==
* Computes values of the GIG pdf
* arguments are (first 5 are mandatory):
*   r - list of integer shape parameters for the GNIG distribution
*   l - list of rate parameters for the GNIG distribution
*   z - value where the pdf is to be computed
*   prec - optional argument that indicates the number of precision digits
*           for computations (default value: 50)
*   precp - optional argument that indicates the number of digits for
*            printing the result (default value: 20)
*****
> GIGpdf(c(7,4,6),c(3.4,1.2,5.6),4.6)
0.16898659825321382267
> GIGpdf(c(7,4,6),c(3.4,1.2,5.6),4.6,,30)
0.168986598253213822668320246907
> GIGpdf(c(7,4,6),c(3.4,1.2,5.6),4.6,,55)
0.1689865982532138226683202469067962498636429438084936650

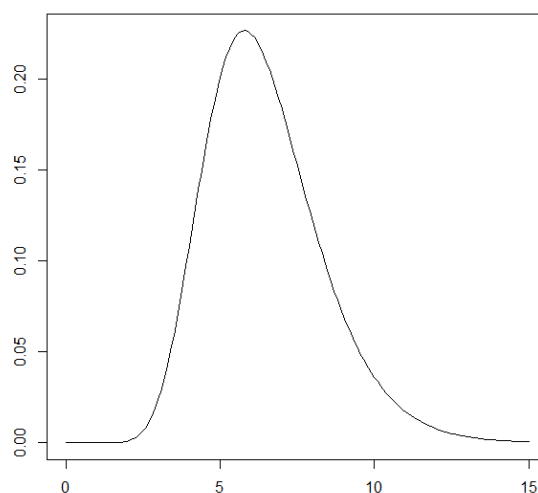
> # If the required printing number of digits exceeds the number of precision digits
> #   for computations, the last digits that exceed this value are in general not correct
> GIGpdf(c(7,4,6),c(3.4,1.2,5.6),4.6,60,55)
0.1689865982532138226683202469067962498636429438084813297

> # All user functions, while they give a printed value, they also compute
> # a value which may be stored;
> # this value is a multiple precision value, with 'prec' precision digits
> value<-GIGpdf(c(7,4,6),c(3.4,1.2,5.6),4.6)
0.1689865982532138227
> value
An object of class "mpfr"
[[1]]
'mpfr1' 0.168986598253213822668320246906796249863642943808494

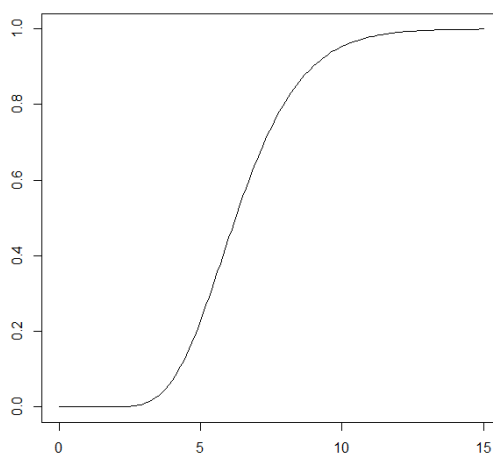
> GIG_help(GIGcdf)
*****
* ==> GIGcdf(r,l,z,prec,precp) <==
* Computes values of the GIG cdf
* arguments are (first 5 are mandatory):
*   r - list of integer shape parameters for the GNIG distribution
*   l - list of rate parameters for the GNIG distribution
*   z - value where the pdf is to be computed
*   prec - optional argument that indicates the number of precision digits
*           for computations (default value: 50)
*   precp - optional argument that indicates the number of digits for
*            printing the result (default value: 20)
*****
```

```
> GIGcdf(c(7,4,6),c(3.4,1.2,5.6),14.6)
0.99913058737075271510
> GIGcdf(c(7,4,6),c(3.4,1.2,5.6),25.6)
0.99999998750072964845
> GIGcdf(c(7,4,6),c(3.4,1.2,5.6),55.6)
1.00000000000000000000
> GIGcdf(c(7,4,6),c(3.4,1.2,5.6),55.6,,50)
0.999999999999999999996185377757127161136115034726
```

```
> GIG_help(plotGIGpdf)
*****
* ==> plotGIGpdf(r,l,ll,ul,step,prec=50) *
* Plots GIG pdfs *
* Its has 6 arguments (the first 5 are mandatory): *
*   r - list of shape parameters for the GIG distribution *
*   l - list of rate parameters for the GIG distribution *
*   ll - lower limit for plot range (usually 0) *
*   ul - upper limit for plot range *
*   step - step for plotting points between ll and ul *
*   prec - (optional argument) number of precision digits for computations *
*           (default value: 50) *
*****
> plotGIGpdf(c(7,4,6),c(3.4,1.2,5.6),0,15,.1)
```



```
> GIG_help(plotGIGcdf)
*****
* ==> plotGIGcdf(r,l,ll,ul,step,prec=50)
* Plots GIG cdfs
* Its has 6 arguments (the first 5 are mandatory):
*   r - list of shape parameters for the GIG distribution
*   l - list of rate parameters for the GIG distribution
*   ll - lower limit for plot range (usually 0)
*   ul - upper limit for plot range
*   step - step for plotting points between ll and ul
*   prec - (optional argument) number of precision digits for computations
*           (default value: 50)
*****
> plotGIGcdf(c(7,4,6),c(3.4,1.2,5.6),0,15,.1)
```



```
> GIG_help(GIGquant)
*****
* ==> GIGquant(r,l,quant,xo,eps=10^-10,prec=50,precpr=10)
* Computes quantiles for the GIG distribution
* It has 7 arguments (the first 4 are mandatory)
*   r - list of integer shape parameters
*   l - list of corresponding rate parameters
*   quant - quantile to be computed (value between 0 and 1)
*   xo - initial value for the quantile
*   eps - (optional argument) maximum error for the quantile
*           (default value: 10^-10)
*   prec - (optional argument) number of precision digits for computations
*           (default value: 50)
*   precpr - (optional argument) number of digits for printing the result
*             (default value: maximum of -log10(eps) and the given value)
*****
> GIGquant(c(7,4,6),c(3.4,1.2,5.6),.05,2)
3.793759456
> q05<-GIGquant(c(7,4,6),c(3.4,1.2,5.6),.05,2)
3.793759456
> q05
An object of class "mpfr"
[[1]]
'mpfr1' 3.7937594550137866276315089080542162561292272932861
```

[illegible]

```

> GNIGpdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,4.6)
0.060513926134947235470
> GNIGpdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,4.6,,30)
0.0605139261349472354700402981119
> GNIGpdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,4.6,60,55)
0.06051392613494723547004029811187270191036986315462832261
> GNIGpdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,4.6,,50)
0.060513926134947235470040298111872701910291495463928
> # Computations for the GNIG distribution are quite heavy and quite tough
> # and as such, some of the last digits in the computation, for a given
> # value of precision digits for computations, may be not precise enough;
> # anyway, giving a large value for prec, will get a more precise value

> GIG_help(GNIGcdf)
*****
* ==> GNIGcdf(r,ro,l,lo,z,prec,prec) <==
* Computes values of the GNIG cdf
* arguments are (first 5 are mandatory):
*   r - list of integer shape parameters for the GNIG distribution
*   ro - non-integer shape parameter
*   l - list of rate parameters corresponding to the integer shape parameters
*   lo - rate parameter corresponding to the non-integer shape parameter
*   z - value where the pdf is to be computed
*   prec - optional argument that indicates the number of precision digits
*           for computations (default value: 50)
*   precp - optional argument that indicates the number of digits for
*            printing the result (default value: 20)
*****
> GNIGcdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,4.6)
0.033737765820166216771
> GNIGcdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,4.6,,30)
0.0337377658201662167710882990179
> GNIGcdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,4.6,,50)
0.033737765820166216771088299017891187918886021421138

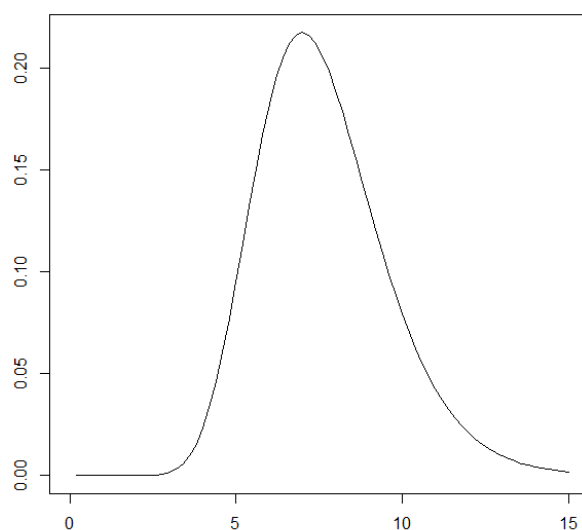
> GNIGcdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,4.6,60,55)
0.03373776582016621677108829901789119221000817933581527981

> GIG_help(plotGNIGpdf)
*****
* ==> plotGNIGpdf(r,ro,l,lo,ll,ul,step,prec=50) <==
* Plots GNIG pdfs
* arguments are (first 5 are mandatory):
*   r - list of integer shape parameters for the GNIG distribution
*   ro - non-integer shape parameter
*   l - list of rate parameters for the GIG distribution
*   lo - rate parameter corresponding to the non-integer shape parameter
*   ll - lower limit for plot range (usually 0)
*   ul - upper limit for plot range
*   step - step for plotting points between ll and ul
*   prec - (optional argument) number of precision digits for computations
*****

> # As computations for the GNIG distribution are quite tough, plotting the pdf
> # and mainly pltting the cdf, may take a while, even on faster computers

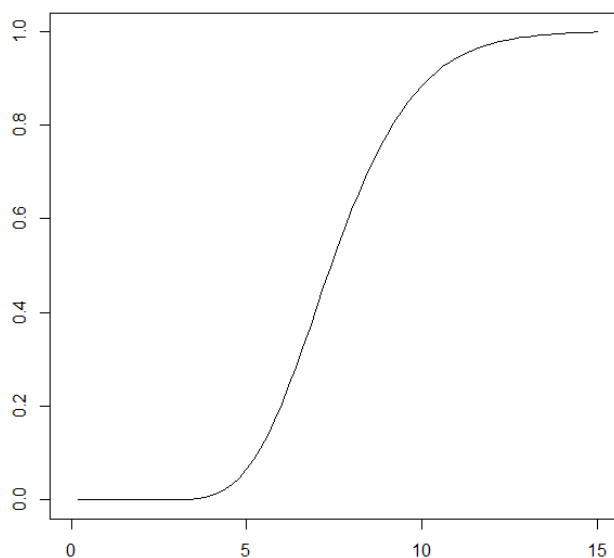
```

```
> plotGNIGpdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,0,15,.2)
```



```
> GIG_help(plotGNIGcdf)
```

```
*****
* ==> plotGNIGcdf(r,ro,l,lo,ll,ul,step,prec=50) <==*
* Plots GNIG cdfs*
* arguments are (first 5 are mandatory):*
*   r - list of integer shape parameters for the GNIG distribution*
*   ro - non-integer shape parameter*
*   l - list of rate parameters for the GIG distribution*
*   lo - rate parameter corresponding to the non-integer shape parameter*
*   ll - lower limit for plot range (usually 0)*
*   ul - upper limit for plot range*
*   step - step for plotting points between ll and ul*
*   prec - (optional argument) number of precision digits for computations*
*****
> plotGNIGcdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,0,15,.2)
```



[illegible]

```

*****
* ==> GNIGquant(r1,r2,l1,l2,quant,xo,eps=10^-10,prec=50,precp=10)
* Computes quantiles for the GNIG distribution
* It has 9 arguments (the first 6 are mandatory)
*   r1 - list of integer shape parameters
*   r2 - non-integer shape parameter
*   l1 - list of rate parameters corresponding to the integer shape parameters
*   l2 - rate parameter corresponding to the non-integer shape parameter
*   quant - quantile to be computed (value between 0 and 1)
*   xo - initial value for the quantile
*   eps - (optional argument) maximum error for the quantile
*         (default value: 10^-10)
*   prec - (optional argument) number of precision digits for computations
*         (default value: 50)
*   precp - (optional argument) number of digits for printing the result
*         (default value: maximum of -log10(eps) and the given value)
*****

```

```
> GNIGquant(c(7, 4, 6), 5.6, c(3.4, 1.2, 5.6), 4.8, .05, 2)
4.832361180
```

```
> q05<-GNIGquant(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,.05,2)
4.832361180
```

```
> q05
An object of class "mpfr"
[[1]]
'mpfr1' 4.83236118001985006837940527628052795884834063524859
```

```
> GNIGcdf(c(7, 4, 6), 5.6, c(3.4, 1.2, 5.6), 4.8, q05)
      0.05000000000000000000
```

```
> GNIGcdf(c(7, 4, 6), 5.6, c(3.4, 1.2, 5.6), 4.8, q05, , 30)
      0.05000000000000000000000000568032
```

```
> q05<-GNIGquant(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,.05,5)
4.832361180 q05
```

```
An object of class "mpfr"
[[1]]
'mpfr1' 4.83236118001985006837940456466293726415995969748488
```

```
> GNIGcdf(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,q05,,30)
      0.05000000000000000000000000000000
```

[illegible]

```
> # Note that the minimum precision required for the quantile that is being used in
> # only 10^-10
```

```
> q05<-GNIGquant(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,.05,5,10^-25)
4.832361180019850068379405
```

[illegible][illegible]

```
> # Note also that to obtain a more precise quantile, in cases that eps
> # is very small, it may required to increase prec
> # and then also to increase prec when computing the cdf
```

```
> q05<-GNIGquant(c(7,4,6),5.6,c(3.4,1.2,5.6),4.8,.05,5,10^-25,100)
4.832361180019850068379405
```

[illegible]

[illegible]