### Functions by name

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>abbrev(s,n)</code></td>
<td>name s, abbreviated to a length of n</td>
</tr>
<tr>
<td><code>abs(x)</code></td>
<td>the absolute value of x</td>
</tr>
<tr>
<td><code>acos(x)</code></td>
<td>the radian value of the arccosine of x</td>
</tr>
<tr>
<td><code>acosh(x)</code></td>
<td>the inverse hyperbolic cosine of x</td>
</tr>
<tr>
<td><code>asin(x)</code></td>
<td>the radian value of the arcsine of x</td>
</tr>
<tr>
<td><code>asinh(x)</code></td>
<td>the inverse hyperbolic sine of x</td>
</tr>
<tr>
<td><code>atan(x)</code></td>
<td>the radian value of the arctangent of x</td>
</tr>
<tr>
<td><code>atan2(y,x)</code></td>
<td>the radian value of the arctangent of y/x, where the signs of the parameters y and x are used to determine the quadrant of the answer</td>
</tr>
<tr>
<td><code>atanh(x)</code></td>
<td>the inverse hyperbolic tangent of x</td>
</tr>
<tr>
<td><code>autocode(x,n,x0,x1)</code></td>
<td>partitions the interval from x0 to x1 into n equal-length intervals and returns the upper bound of the interval that contains x</td>
</tr>
<tr>
<td><code>betaden(a,b,x)</code></td>
<td>the probability density of the beta distribution, where a and b are the shape parameters; 0 if x &lt; 0 or x &gt; 1</td>
</tr>
<tr>
<td><code>binomial(n,k,\theta)</code></td>
<td>the probability of observing ( \text{floor}(k) ) or fewer successes in ( \text{floor}(n) ) trials when the probability of a success on one trial is ( \theta ); 0 if ( k &lt; 0 ); or 1 if ( k &gt; n )</td>
</tr>
<tr>
<td><code>binomialp(n,k,p)</code></td>
<td>the probability of observing ( \text{floor}(k) ) successes in ( \text{floor}(n) ) trials when the probability of a success on one trial is ( p )</td>
</tr>
<tr>
<td><code>binomialtail(n,k,\theta)</code></td>
<td>the probability of observing ( \text{floor}(k) ) or more successes in ( \text{floor}(n) ) trials when the probability of a success on one trial is ( \theta ); 1 if ( k &lt; 0 ); or 0 if ( k &gt; n )</td>
</tr>
<tr>
<td><code>binormal(h,k,\rho)</code></td>
<td>the joint cumulative distribution ( \Phi(h,k,\rho) ) of bivariate normal with correlation ( \rho )</td>
</tr>
<tr>
<td><code>bofd(&quot;cal&quot;,e_d)</code></td>
<td>the ( e_b ) business date corresponding to ( e_d )</td>
</tr>
<tr>
<td><code>byteorder()</code></td>
<td>1 if your computer stores numbers by using a hilo byte order and evaluates to 2 if your computer stores numbers by using a lohi byte order</td>
</tr>
<tr>
<td><code>c(name)</code></td>
<td>the value of the system or constant result ( c(name) ) (see [P] creturn)</td>
</tr>
<tr>
<td><code>_caller()</code></td>
<td>version of the program or session that invoked the currently running program; see [P] version</td>
</tr>
<tr>
<td><code>cauchy(a,b,x)</code></td>
<td>the cumulative Cauchy distribution with location parameter ( a ) and scale parameter ( b )</td>
</tr>
<tr>
<td><code>cauchyden(a,b,x)</code></td>
<td>the probability density of the Cauchy distribution with location parameter ( a ) and scale parameter ( b )</td>
</tr>
<tr>
<td><code>cauchytail(a,b,x)</code></td>
<td>the reverse cumulative (upper tail or survivor) Cauchy distribution with location parameter ( a ) and scale parameter ( b )</td>
</tr>
<tr>
<td><code>Cdhms(e_d,h,m,s)</code></td>
<td>the ( e_{\Delta C} ) datetime (ms. with leap seconds since 01jan1960 00:00:00.000) corresponding to ( e_d, h, m, s )</td>
</tr>
<tr>
<td><code>ceil(x)</code></td>
<td>the unique integer ( n ) such that ( n - 1 &lt; x \leq n; x ) (not “.&quot; ) if ( x ) is missing, meaning that <code>ceil(.a) = .a</code></td>
</tr>
</tbody>
</table>
2 Functions by name

`char(n)`  
the character corresponding to ASCII or extended ASCII code `n`; "" if `n` is not in the domain

`chi2(df, x)`  
the cumulative $\chi^2$ distribution with `df` degrees of freedom; 0 if $x < 0$

`chi2den(df, x)`  
the probability density of the chi-squared distribution with `df` degrees of freedom; 0 if $x < 0$

`chi2tail(df, x)`  
the reverse cumulative (upper tail or survivor) $\chi^2$ distribution with `df` degrees of freedom; 1 if $x < 0$

`Chms(h, m, s)`  
the $e_{tc}$ datetime (ms. with leap seconds since 01jan1960 00:00:00.000) corresponding to `h`, `m`, `s` on 01jan1960

`chop(x, \epsilon)`  
round(`x`) if $\text{abs}(x - \text{round}(x)) < \epsilon$; otherwise, `x`; or `x` if `x` is missing

`cholesky(M)`  
the Cholesky decomposition of the matrix: if $R = \text{cholesky}(S)$, then $RR^\top = S$

`clip(x,a,b)`  
`x` if $a < x < b$, $b$ if $x \geq b$, $a$ if $x \leq a$, or missing if `x` is missing

or `a > b` if $x$ is missing

`Clock(s_1, s_2[Y])`  
the $e_{tc}$ datetime (ms. with leap seconds since 01jan1960 00:00:00.000) corresponding to `s_1` based on `s_2` and `Y`

`clock(s_1, s_2[Y])`  
the $e_{tc}$ datetime (ms. since 01jan1960 00:00:00.000) corresponding to `s_1` based on `s_2` and `Y`

`cloglog(x)`  
the complementary log-log of `x`

`Cmdyhms(M, D, Y, h, m, s)`  
the $e_{tc}$ datetime (ms. with leap seconds since 01jan1960 00:00:00.000) corresponding to `M`, `D`, `Y`, `h`, `m`, `s`

`Cofc(e_{tc})`  
the $e_{tc}$ datetime (ms. with leap seconds since 01jan1960 00:00:00.000) of $e_{tc}$ (ms. without leap seconds since 01jan1960 00:00:00.000)

`cofC(e_{tc})`  
the $e_{tc}$ datetime (ms. without leap seconds since 01jan1960 00:00:00.000) of $e_{tc}$ (ms. with leap seconds since 01jan1960 00:00:00.000)

`Cofd(e_d)`  
the $e_{tc}$ datetime (ms. with leap seconds since 01jan1960 00:00:00.000) of date $e_d$ at time 00:00:00.000

`cofd(e_d)`  
the $e_{tc}$ datetime (ms. since 01jan1960 00:00:00.000) of date $e_d$ at time 00:00:00.000

`coleqnumb(M, s)`  
the equation number of `M` associated with column equation `s`; `missing` if the column equation cannot be found

`collatorlocale(loc, type)`  
the most closely related locale supported by ICU from `loc` if `type` is 1; the actual locale where the collation data comes from if `type` is 2

`collatorversion(loc)`  
the version string of a collator based on locale `loc`

`colnfreeparms(M)`  
the number of free parameters in columns of `M`

`colnumb(M, s)`  
the column number of `M` associated with column name `s`; `missing` if the column cannot be found

`colsof(M)`  
the number of columns of `M`

`comb(n, k)`  
the combinatorial function $n!/\{k!(n-k)\}$

`cond(x,a,b[,c])`  
a if `x` is true and nonmissing, `b` if `x` is false, and `c` if `x` is missing;

`a` if `c` is not specified and `x` evaluates to `missing`

`corr(M)`  
the correlation matrix of the variance matrix
cos(x)  
the cosine of x, where x is in radians

cosh(x)  
the hyperbolic cosine of x

daily(s₁, s₂[, Y])  
a synonym for date(s₁, s₂[, Y])
date(s₁, s₂[, Y])  
the e_d date (days since 01jan1960) corresponding to s₁ based on
  s₂ and Y
day(e_d)  
the numeric day of the month corresponding to e_d
det(M)  
the determinant of matrix M
dgammapda(a, x)  
\frac{\partial P(a, x)}{\partial a}, where P(a, x) = \gammamap(a, x); 0 if x < 0
dgammapdada(a, x)  
\frac{\partial^2 P(a, x)}{\partial a^2}, where P(a, x) = \gammamap(a, x); 0 if x < 0
dgammapdadx(a, x)  
\frac{\partial^2 P(a, x)}{\partial a \partial x}, where P(a, x) = \gammamap(a, x); 0 if x < 0
dgammapdax(a, x)  
\frac{\partial P(a, x)}{\partial x}, where P(a, x) = \gammamap(a, x); 0 if x < 0
dgammapdxdx(a, x)  
\frac{\partial^2 P(a, x)}{\partial x^2}, where P(a, x) = \gammamap(a, x); 0 if x < 0
dhms(e_d, h, m, s)  
the e_d date (days since 01jan1960) of time (ms. since 01jan1960 00:00:00.000) corresponding
to e_d, h, m, and s
diag(M)  
the square, diagonal matrix created from the row or column vector
the number of zeros on the diagonal of M
diag0cnt(M)  
digamma() function, \frac{d \ln \Gamma(x)}{dx}
dofb(e_b, "cal")  
the e_d date (days since 01jan1960) of datetime e_b
dofC(e_tC)  
the e_d date (days since 01jan1960) of datetime e_tC (ms. with leap
  seconds since 01jan1960 00:00:00.000)
dofc(e_tC)  
the e_d date (days since 01jan1960) of datetime e_tC (ms. since
  01jan1960 00:00:00.000)
dofh(e_h)  
the e_d date (days since 01jan1960) of the start of half-year e_h
dofm(e_m)  
the e_d date (days since 01jan1960) of the start of month e_m
dofq(e_q)  
the e_d date (days since 01jan1960) of the start of quarter e_q
dofw(e_w)  
the e_d date (days since 01jan1960) of the start of week e_w
dofy(e_y)  
the e_d date (days since 01jan1960) of 01jan in year e_y
dow(e_d)  
the numeric day of the week corresponding to date e_d; 0 = Sunday,
  1 = Monday, ..., 6 = Saturday
doy(e_d)  
the numeric day of the year corresponding to date e_d
dunnettprob(k, df, x)  
the cumulative multiple range distribution that is used in Dunnett’s
  multiple-comparison method with k ranges and df degrees of
  freedom; 0 if x < 0
e(name)  
the value of stored result e(name); see [U] 18.8 Accessing results
calculated by other programs
el(s, i, j)  
s[floor(i), floor(j)], the i, j element of the matrix named s;
  missing if i or j are out of range or if matrix s does not exist
e(sample)  
1 if the observation is in the estimation sample and 0 otherwise
epsdouble()  
the machine precision of a double-precision number
epsfloat()  
the machine precision of a floating-point number
exp(x)  
the exponential function \( e^x \)
expm1(x)  
e^x - 1 with higher precision than \( \exp(x) - 1 \) for small values of
  \( |x| \)
4 Functions by name

exponential(b,x) the cumulative exponential distribution with scale b
exponentialden(b,x) the probability density function of the exponential distribution with scale b
exponentialtail(b,x) the reverse cumulative exponential distribution with scale b
F(df1,df2,f) the cumulative $F$ distribution with $df_1$ numerator and $df_2$ denominator degrees of freedom: $F(df_1, df_2, f) = \int_0^f F_{\text{den}}(df_1, df_2, t) \, dt$; 0 if $f < 0$
Fden(df1,df2,f) the probability density function of the $F$ distribution with $df_1$ numerator and $df_2$ denominator degrees of freedom; 0 if $f < 0$
fileexists(f) 1 if the file specified by $f$ exists; otherwise, 0
fileread(f) the contents of the file specified by $f$
filereaderror(s) 0 or positive integer, said value having the interpretation of a return code
filewrite(f,s[,r]) writes the string specified by $s$ to the file specified by $f$ and returns the number of bytes in the resulting file
float(x) the value of $x$ rounded to float precision
floor(x) the unique integer $n$ such that $n \leq x < n + 1$; $x$ (not “.”) if $x$ is missing, meaning that $\text{floor}(\cdot.a) = \cdot.a$
fmtwidth(fmtstr) the output length of the \%fmt contained in fmtstr; missing if fmtstr does not contain a valid \%fmt
frval() returns values of variables stored in other frames
__frval() programmer’s version of frval()
Ftail(df1,df2,f) the reverse cumulative (upper tail or survivor) $F$ distribution with $df_1$ numerator and $df_2$ denominator degrees of freedom; 1 if $f < 0$
gammadens(a,b,g,x) the probability density function of the gamma distribution; 0 if $x < g$
gammaph(a,x) the cumulative gamma distribution with shape parameter $a$; 0 if $x < 0$
gammapartail(a,x) the reverse cumulative (upper tail or survivor) gamma distribution with shape parameter $a$; 1 if $x < 0$
get(systemname) a copy of Stata internal system matrix systemname
hadamard(M,N) a matrix whose $i, j$ element is $M[i,j] \cdot N[i,j]$ (if $M$ and $N$ are not the same size, this function reports a conformability error)
halfyear(ed) the numeric half of the year corresponding to date $e_d$
halfyearly(s1,s2[,Y]) the $e_h$ half-yearly date (half-years since 1960h1) corresponding to $s_1$ based on $s_2$ and $Y$; $Y$ specifies topyear; see date() has_eiprop(name) 1 if name appears as a word in e(properties); otherwise, 0
hh(ed) the hour corresponding to datetime $e_{tc}$ (ms. since 01jan1960 00:00:00.000)
hhC(ed) the hour corresponding to datetime $e_{tc}$ (ms. with leap seconds since 01jan1960 00:00:00.000)
hms(h,m,s) the $e_{tc}$ datetime (ms. since 01jan1960 00:00:00.000) corresponding to $h$, $m$, $s$ on 01jan1960
hofd(ed) the $e_h$ half-yearly date (half years since 1960h1) containing date $e_d$
hours(ms) $ms/3,600,000$
hyergeometric($N,K,n,k$) the cumulative probability of the hypergeometric distribution

hyergeometricp($N,K,n,k$) the hypergeometric probability of $k$ successes out of a sample of size $n$, from a population of size $N$ containing $K$ elements that have the attribute of interest

$I(n)$ an $n \times n$ identity matrix if $n$ is an integer; otherwise, a $\text{round}(n) \times \text{round}(n)$ identity matrix

ibeta($a,b,x$) the cumulative beta distribution with shape parameters $a$ and $b$; 0 if $x < 0$; or 1 if $x > 1$

ibetatail($a,b,x$) the reverse cumulative (upper tail or survivor) beta distribution with shape parameters $a$ and $b$; 1 if $x < 0$; or 0 if $x > 1$

igaussian($m,a,x$) the cumulative inverse Gaussian distribution with mean $m$ and shape parameter $a$; 0 if $x \leq 0$

igaussianden($m,a,x$) the probability density of the inverse Gaussian distribution with mean $m$ and shape parameter $a$; 0 if $x \leq 0$

igaussiantail($m,a,x$) the reverse cumulative (upper tail or survivor) inverse Gaussian distribution with mean $m$ and shape parameter $a$; 1 if $x \leq 0$

indexnot($s_1,s_2$) the position in ASCII string $s_1$ of the first character of $s_1$ not found in ASCII string $s_2$, or 0 if all characters of $s_1$ are found in $s_2$

inlist($z,a,b,...$) 1 if $z$ is a member of the remaining arguments; otherwise, 0

inrange($z,a,b$) 1 if it is known that $a \leq z \leq b$; otherwise, 0

int($x$) the integer obtained by truncating $x$ toward 0 (thus, int(5.2) = 5 and int(-5.8) = -5); $x$ (not “.”) if $x$ is missing, meaning that int(.a) = .a

inv($M$) the inverse of the matrix $M$

invbinomial($n,k,p$) the inverse of the cumulative binomial; that is, $\theta$ ($\theta$ = probability of success on one trial) such that the probability of observing floor($k$) or fewer successes in floor($n$) trials is $p$

invbinomialtail($n,k,p$) the inverse of the right cumulative binomial; that is, $\theta$ ($\theta$ = probability of success on one trial) such that the probability of observing floor($k$) or more successes in floor($n$) trials is $p$

invcauchy($a,b,p$) the inverse of cauchy(): if cauchy($a,b,x$) = $p$, then invcauchy($a,b,p$) = $x$

invcaucytail($a,b,p$) the inverse of cauchytail(): if cauchytail($a,b,x$) = $p$, then invcaucytail($a,b,p$) = $x$

invchi2($df,p$) the inverse of chi2(): if chi2($df,x$) = $p$, then invchi2($df,p$) = $x$

invchi2tail($df,p$) the inverse of chi2tail(): if chi2tail($df,x$) = $p$, then invchi2tail($df,p$) = $x$

invcloglog($x$) the inverse of the complementary log-log function of $x$

invdunnettprob($k,df,p$) the inverse cumulative multiple range distribution that is used in Dunnett’s multiple-comparison method with $k$ ranges and $df$ degrees of freedom

inexponential($b,p$) the inverse cumulative exponential distribution with scale $b$: if exponential($b,x$) = $p$, then inexponential($b,p$) = $x$

inexponentialtail($b,p$) the inverse reverse cumulative exponential distribution with scale $b$: if exponentialtail($b,x$) = $p$, then inexponentialtail($b,p$) = $x$
invF(df1, df2, p)
the inverse cumulative $F$ distribution: if $F(df_1, df_2, f) = p$, then invF(df1, df2, p) = f

invFtail(df1, df2, p)
the inverse reverse cumulative (upper tail or survivor) $F$ distribution: if $Ftail(df_1, df_2, f) = p$, then invFtail(df1, df2, p) = f

invgammap(a, p)
the inverse cumulative gamma distribution: if gammap(a, x) = p, then invgammap(a, p) = x

invgammaptail(a, p)
the inverse reverse cumulative (upper tail or survivor) gamma distribution: if gammaptail(a, x) = p, then invgammaptail(a, p) = x

invbeta(a, b, p)
the inverse cumulative beta distribution: if ibeta(a, b, x) = p, then invbeta(a, b, p) = x

invbetatail(a, b, p)
the inverse reverse cumulative (upper tail or survivor) beta distribution: if betatail(a, b, x) = p, then invbetatail(a, b, p) = x

invgaussian(m, a, p)
the inverse of igaussian(): if igaussian(m, a, x) = p, then invgaussian(m, a, x) = x

invgaussianantail(m, a, p)
the inverse of igaussianantail(): if igaussianantail(m, a, x) = p, then invgaussianantail(m, a, p) = x

inlaplace(m, b, p)
the inverse of laplace(): if laplace(m, b, x) = p, then inlaplace(m, b, p) = x

inlaplacetail(m, b, p)
the inverse of laplacetail(): if laplacetail(m, b, x) = p, then inlaplacetail(m, b, p) = x

invlogistic(p)
the inverse cumulative logistic distribution: if logistic(x) = p, then invlogistic(p) = x

invlogistic(s, p)
the inverse cumulative logistic distribution: if logistic(s, x) = p, then invlogistic(s, p) = x

invlogistic(m, s, p)
the inverse cumulative logistic distribution: if logistic(m, s, x) = p, then invlogistic(m, s, p) = x

invlogistictail(p)
the inverse reverse cumulative logistic distribution: if logistictail(x) = p, then invlogistictail(p) = x

invlogistictail(s, p)
the inverse cumulative logistic distribution: if logistic(s, x) = p, then invlogistic(s, p) = x

invlogistictail(m, s, p)
the inverse cumulative logistic distribution: if logistic(m, s, x) = p, then invlogistic(m, s, p) = x

invlogit(x)
the inverse of the logit function of x

invnbinomial(n, k, q)
the value of the negative binomial parameter, $p$, such that $q = nbinomial(n, k, p)$

invnbinomialtail(n, k, q)
the value of the negative binomial parameter, $p$, such that $q = nbinomialtail(n, k, p)$

invchi2(df, np, p)
the inverse cumulative noncentral $\chi^2$ distribution: if $nchi2(df, np, x) = p$, then invchi2(df, np, p) = x

invchi2tail(df, np, p)
the inverse reverse cumulative (upper tail or survivor) noncentral $\chi^2$ distribution: if $nchi2tail(df, np, x) = p$, then invchi2tail(df, np, p) = x

invF(df1, df2, np, p)
the inverse cumulative noncentral $F$ distribution: if $nF(df_1, df_2, np, f) = p$, then invnF(df1, df2, np, p) = f
The following are the inverse cumulative distribution functions:

- **invnFtail(df1, df2, np, p)**: The inverse reverse cumulative (upper tail or survivor) noncentral F distribution: if \( \text{nFtail}(df_1, df_2, np, f) = p \), then \( \text{invnFtail}(df_1, df_2, np, p) = f \)

- **invnibeta(a, b, np, p)**: The inverse cumulative noncentral beta distribution: if \( \text{nibeta}(a, b, np, x) = p \), then \( \text{invnibeta}(a, b, np, p) = x \)

- **invnt(df, np, p)**: The inverse cumulative noncentral Student’s t distribution: if \( \text{nt}(df, np, t) = p \), then \( \text{invnt}(df, np, p) = t \)

- **invnttail(df, np, p)**: The inverse reverse cumulative (upper tail or survivor) noncentral Student’s t distribution: if \( \text{nntail}(df, np, t) = p \), then \( \text{invnttail}(df, np, p) = t \)

- **invnFtail(df1, df2, np, x)**: The inverse cumulative Student’s F distribution: if \( \text{F}(df_1, df_2, np, f) = x \), then \( \text{invnFtail}(df_1, df_2, np, x) = f \)

- **invnormal(p)**: The inverse cumulative standard normal distribution: if \( \text{normal}(z) = p \), then \( \text{invnormal}(p) = z \)

- **invpoisson(k, p)**: The Poisson mean such that the cumulative Poisson distribution evaluated at \( k \) is \( p \): if \( \text{poisson}(m, k) = p \), then \( \text{invpoisson}(k, p) = m \)

- **invpoissontail(k, q)**: The Poisson mean such that the reverse cumulative Poisson distribution evaluated at \( k \) is \( q \): if \( \text{poissontail}(m, k) = q \), then \( \text{invpoissontail}(k, q) = m \)

- **invnormal(p)**: The inverse of \( M \) if \( M \) is positive definite

- **invnt(df, np, p)**: The inverse cumulative Student’s t distribution: if \( t(df, t) = p \), then \( \text{invnt}(df, np, p) = t \)

- **invnttail(df, np, p)**: The inverse reverse cumulative (upper tail or survivor) Student’s t distribution: if \( ttail(df, t) = p \), then \( \text{invnttail}(df, np, p) = t \)

- **invtukeyprob(k, df, p)**: The inverse cumulative Tukey’s Studentized range distribution with \( k \) ranges and \( df \) degrees of freedom

- **invweibull(a, b, p)**: The inverse cumulative Weibull distribution with shape \( a \) and scale \( b \): if \( \text{weibull}(a, b, x) = p \), then \( \text{invweibull}(a, b, p) = x \)

- **invweibull(a, b, g, p)**: The inverse cumulative Weibull distribution with shape \( a \), scale \( b \), and location \( g \): if \( \text{weibull}(a, b, g, x) = p \), then \( \text{invweibull}(a, b, g, p) = x \)

- **invweibullph(a, b, p)**: The inverse cumulative Weibull (proportional hazards) distribution with shape \( a \) and scale \( b \): if \( \text{weibullph}(a, b, x) = p \), then \( \text{invweibullph}(a, b, p) = x \)

- **invweibullph(a, b, g, p)**: The inverse cumulative Weibull (proportional hazards) distribution with shape \( a \), scale \( b \), and location \( g \): if \( \text{weibullph}(a, b, g, x) = p \), then \( \text{invweibullph}(a, b, g, p) = x \)

- **invweibullphtail(a, b, p)**: The inverse reverse cumulative Weibull (proportional hazards) distribution with shape \( a \) and scale \( b \): if \( \text{weibullphtail}(a, b, x) = p \), then \( \text{invweibullphtail}(a, b, p) = x \)

- **invweibullphtail(a, b, g, p)**: The inverse reverse cumulative Weibull (proportional hazards) distribution with shape \( a \), scale \( b \), and location \( g \): if \( \text{weibullphtail}(a, b, g, x) = p \), then \( \text{invweibullphtail}(a, b, g, p) = x \)

- **invweibulltail(a, b, p)**: The inverse reverse cumulative Weibull distribution with shape \( a \) and scale \( b \): if \( \text{weibulltail}(a, b, x) = p \), then \( \text{invweibulltail}(a, b, p) = x \)

- **invweibulltail(a, b, g, p)**: The inverse reverse cumulative Weibull distribution with shape \( a \), scale \( b \), and location \( g \): if \( \text{weibulltail}(a, b, g, x) = p \), then \( \text{invweibulltail}(a, b, g, p) = x \)
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{irecode(x, x_1, \ldots, x_n)}</td>
<td>\textit{missing} if (x) is missing or (x_1, \ldots, x_n) is not weakly increasing; 0 if (x \leq x_1); 1 if (x_1 &lt; x \leq x_2); 2 if (x_2 &lt; x \leq x_3); \ldots; (n) if (x &gt; x_n)</td>
</tr>
<tr>
<td>\texttt{issymmetric(M)}</td>
<td>1 if the matrix is symmetric; otherwise, 0</td>
</tr>
<tr>
<td>\texttt{J(r, c, z)}</td>
<td>the (r \times c) matrix containing elements (z)</td>
</tr>
<tr>
<td>\texttt{laplace(m, b, x)}</td>
<td>the cumulative Laplace distribution with mean (m) and scale parameter (b)</td>
</tr>
<tr>
<td>\texttt{laplaceden(m, b, x)}</td>
<td>the probability density of the Laplace distribution with mean (m) and scale parameter (b)</td>
</tr>
<tr>
<td>\texttt{laplacetail(m, b, x)}</td>
<td>the reverse cumulative (upper tail or survivor) Laplace distribution with mean (m) and scale parameter (b)</td>
</tr>
<tr>
<td>\texttt{ln(x)}</td>
<td>the natural logarithm, (\ln(x))</td>
</tr>
<tr>
<td>\texttt{ln1m(x)}</td>
<td>the natural logarithm of (1 - x) with higher precision than (\ln(1 - x)) for small values of (</td>
</tr>
<tr>
<td>\texttt{ln1p(x)}</td>
<td>the natural logarithm of (1 + x) with higher precision than (\ln(1 + x)) for small values of (</td>
</tr>
<tr>
<td>\texttt{lncauchyden(a, b, x)}</td>
<td>the natural logarithm of the density of the Cauchy distribution with location parameter (a) and scale parameter (b)</td>
</tr>
<tr>
<td>\texttt{lnfactorial(n)}</td>
<td>the natural log of (n) factorial = (\ln(n!))</td>
</tr>
<tr>
<td>\texttt{lngamma(x)}</td>
<td>(\ln{\Gamma(x)})</td>
</tr>
<tr>
<td>\texttt{lnigammaden(a, b, x)}</td>
<td>the natural logarithm of the inverse gamma density, where (a) is the shape parameter and (b) is the scale parameter</td>
</tr>
<tr>
<td>\texttt{lngaussianden(m, a, x)}</td>
<td>the natural logarithm of the inverse Gaussian density with mean (m) and shape parameter (a)</td>
</tr>
<tr>
<td>\texttt{lniwishartden(df, V, X)}</td>
<td>the natural logarithm of the density of the inverse Wishart distribution; missing if (df \leq n - 1)</td>
</tr>
<tr>
<td>\texttt{lnlplaceden(m, b, x)}</td>
<td>the natural logarithm of the density of the Laplace distribution with mean (m) and scale parameter (b)</td>
</tr>
<tr>
<td>\texttt{lnmvnormalden(M, V, X)}</td>
<td>the natural logarithm of the multivariate normal density</td>
</tr>
<tr>
<td>\texttt{lnnormal(z)}</td>
<td>the natural logarithm of the cumulative standard normal distribution</td>
</tr>
<tr>
<td>\texttt{lnnormalden(z)}</td>
<td>the natural logarithm of the standard normal density, (N(0, 1))</td>
</tr>
<tr>
<td>\texttt{lnnormalden(x, \sigma)}</td>
<td>the natural logarithm of the normal density with mean 0 and standard deviation (\sigma)</td>
</tr>
<tr>
<td>\texttt{lnnormalden(x, \mu, \sigma)}</td>
<td>the natural logarithm of the normal density with mean (\mu) and standard deviation (\sigma), (N(\mu, \sigma^2))</td>
</tr>
<tr>
<td>\texttt{lnwishartden(df, V, X)}</td>
<td>the natural logarithm of the density of the Wishart distribution; missing if (df \leq n - 1)</td>
</tr>
<tr>
<td>\texttt{log(x)}</td>
<td>a synonym for (\ln(x))</td>
</tr>
<tr>
<td>\texttt{log10(x)}</td>
<td>the base-10 logarithm of (x)</td>
</tr>
<tr>
<td>\texttt{log1m(x)}</td>
<td>a synonym for (\ln1m(x))</td>
</tr>
<tr>
<td>\texttt{log1p(x)}</td>
<td>a synonym for (\ln1p(x))</td>
</tr>
<tr>
<td>\texttt{logistic(x)}</td>
<td>the cumulative logistic distribution with mean 0 and standard deviation (\pi/\sqrt{3})</td>
</tr>
<tr>
<td>\texttt{logistic(s, x)}</td>
<td>the cumulative logistic distribution with mean 0, scale (s), and standard deviation (s\pi/\sqrt{3})</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>logistic($m, s, x$)</td>
<td>the cumulative logistic distribution with mean $m$, scale $s$, and standard deviation $s\pi/\sqrt{3}$</td>
</tr>
<tr>
<td>logisticden()</td>
<td>the density of the logistic distribution with mean 0 and standard deviation $\pi/\sqrt{3}$</td>
</tr>
<tr>
<td>logisticden($s, x$)</td>
<td>the density of the logistic distribution with mean 0, scale $s$, and standard deviation $s\pi/\sqrt{3}$</td>
</tr>
<tr>
<td>logisticden($m, s, x$)</td>
<td>the density of the logistic distribution with mean $m$, scale $s$, and standard deviation $s\pi/\sqrt{3}$</td>
</tr>
<tr>
<td>logistictail()</td>
<td>the reverse cumulative logistic distribution with mean 0 and standard deviation $\pi/\sqrt{3}$</td>
</tr>
<tr>
<td>logistictail($s, x$)</td>
<td>the reverse cumulative logistic distribution with mean 0, scale $s$, and standard deviation $s\pi/\sqrt{3}$</td>
</tr>
<tr>
<td>logistictail($m, s, x$)</td>
<td>the reverse cumulative logistic distribution with mean $m$, scale $s$, and standard deviation $s\pi/\sqrt{3}$</td>
</tr>
<tr>
<td>logit()</td>
<td>the log of the odds ratio of $x$, $\text{logit}(x) = \ln{x/(1 - x)}$</td>
</tr>
<tr>
<td>matmissing($M$)</td>
<td>1 if any elements of the matrix are missing; otherwise, 0</td>
</tr>
<tr>
<td>matrix(exp)</td>
<td>restricts name interpretation to scalars and matrices; see scalar()</td>
</tr>
<tr>
<td>matuniform($r, c$)</td>
<td>the $r \times c$ matrices containing uniformly distributed pseudorandom numbers on the interval (0, 1)</td>
</tr>
<tr>
<td>max($x_1, x_2, \ldots, x_n$)</td>
<td>the maximum value of $x_1, x_2, \ldots, x_n$</td>
</tr>
<tr>
<td>maxbyte()</td>
<td>the largest value that can be stored in storage type byte</td>
</tr>
<tr>
<td>maxdouble()</td>
<td>the largest value that can be stored in storage type double</td>
</tr>
<tr>
<td>maxfloat()</td>
<td>the largest value that can be stored in storage type float</td>
</tr>
<tr>
<td>maxint()</td>
<td>the largest value that can be stored in storage type int</td>
</tr>
<tr>
<td>maxlong()</td>
<td>the largest value that can be stored in storage type long</td>
</tr>
<tr>
<td>mdy($M, D, Y$)</td>
<td>the $e_d$ date (days since 01jan1960) corresponding to $M, D, Y$</td>
</tr>
<tr>
<td>mdyhms($M, D, Y, h, m, s$)</td>
<td>the $e_{tc}$ datetime (ms. since 01jan1960 00:00:00.000) corresponding to $M, D, Y, h, m, s$</td>
</tr>
<tr>
<td>mi($x_1, x_2, \ldots, x_n$)</td>
<td>a synonym for missing($x_1, x_2, \ldots, x_n$)</td>
</tr>
<tr>
<td>min($x_1, x_2, \ldots, x_n$)</td>
<td>the minimum value of $x_1, x_2, \ldots, x_n$</td>
</tr>
<tr>
<td>minbyte()</td>
<td>the smallest value that can be stored in storage type byte</td>
</tr>
<tr>
<td>mindouble()</td>
<td>the smallest value that can be stored in storage type double</td>
</tr>
<tr>
<td>minfloat()</td>
<td>the smallest value that can be stored in storage type float</td>
</tr>
<tr>
<td>minint()</td>
<td>the smallest value that can be stored in storage type int</td>
</tr>
<tr>
<td>minlong()</td>
<td>the smallest value that can be stored in storage type long</td>
</tr>
<tr>
<td>minutes($ms$)</td>
<td>$ms/60,000$</td>
</tr>
<tr>
<td>missing($x_1, x_2, \ldots, x_n$)</td>
<td>1 if any $x_i$ evaluates to missing; otherwise, 0</td>
</tr>
<tr>
<td>mm($e_{tc}$)</td>
<td>the minute corresponding to datetime $e_{tc}$ (ms. since 01jan1960 00:00:00.000)</td>
</tr>
<tr>
<td>mmC($e_{tc}$)</td>
<td>the minute corresponding to datetime $e_{tc}$ (ms. with leap seconds since 01jan1960 00:00:00.000)</td>
</tr>
<tr>
<td>mod($x, y$)</td>
<td>the modulus of $x$ with respect to $y$</td>
</tr>
<tr>
<td>mofd($e_d$)</td>
<td>the $e_m$ monthly date (months since 1960m1) containing date $e_d$</td>
</tr>
</tbody>
</table>
functions by name

10 Functions by name

- **month(\(e_d\))**
  - the numeric month corresponding to date \(e_d\)

- **monthly(\(s_1, s_2, Y\))**
  - the \(e_m\) monthly date (months since 1960m1) corresponding to \(s_1\) based on \(s_2\) and \(Y\); \(Y\) specifies topyear; see date()

- **mreldif(\(X, Y\))**
  - the relative difference of \(X\) and \(Y\), where the relative difference is defined as \(
  \max_{i,j} \{|x_{ij} - y_{ij}|/(|y_{ij}| + 1)\}
  \)

- **msofhours(h)**
  - \(h \times 3,600,000\)

- **msofminutes(m)**
  - \(m \times 60,000\)

- **msofseconds(s)**
  - \(s \times 1,000\)

- **nbetaden(a, b, np, x)**
  - the probability density function of the noncentral beta distribution; \(0\) if \(x < 0\) or \(x > 1\)

- **nbinomial(n, k, p)**
  - the cumulative probability of the negative binomial distribution

- **nbinomialp(n, k, p)**
  - the negative binomial probability

- **nbinomialtail(n, k, p)**
  - the reverse cumulative probability of the negative binomial distribution

- **nchi2(df, np, x)**
  - the cumulative noncentral \(\chi^2\) distribution; \(0\) if \(x < 0\)

- **nchi2den(df, np, x)**
  - the probability density of the noncentral \(\chi^2\) distribution; \(0\) if \(x < 0\)

- **nchi2tail(df, np, x)**
  - the reverse cumulative (upper tail or survivor) noncentral \(\chi^2\) distribution; \(1\) if \(x < 0\)

- **nF(df1, df2, np, f)**
  - the cumulative noncentral \(F\) distribution with \(df_1\) numerator and \(df_2\) denominator degrees of freedom and noncentrality parameter \(np\); \(0\) if \(f < 0\)

- **nFden(df1, df2, np, f)**
  - the probability density function of the noncentral \(F\) distribution with \(df_1\) numerator and \(df_2\) denominator degrees of freedom and noncentrality parameter \(np\); \(0\) if \(f < 0\)

- **nFtail(df1, df2, np, f)**
  - the reverse cumulative (upper tail or survivor) noncentral \(F\) distribution with \(df_1\) numerator and \(df_2\) denominator degrees of freedom and noncentrality parameter \(np\); \(1\) if \(f < 0\)

- **nibeta(a, b, np, x)**
  - the cumulative noncentral beta distribution; \(0\) if \(x < 0\); or \(1\) if \(x > 1\)

- **normal(z)**
  - the cumulative standard normal distribution

- **normalden(z)**
  - the standard normal density, \(N(0, 1)\)

- **normalden(x, o)**
  - the normal density with mean 0 and standard deviation \(\sigma\)

- **normalden(x, m, o)**
  - the normal density with mean \(\mu\) and standard deviation \(\sigma\), \(N(\mu, \sigma^2)\)

- **npnchi2(df, x, p)**
  - the noncentrality parameter, \(np\), for noncentral \(\chi^2\): if \(nchi2(df, np, x) = p\), then \(npnchi2(df, x, p) = np\)

- **npnF(df1, df2, f, p)**
  - the noncentrality parameter, \(np\), for the noncentral \(F\): if \(nF(df1, df2, np, f) = p\), then \(npnF(df1, df2, f, p) = np\)

- **npnt(df, t, p)**
  - the noncentrality parameter, \(np\), for the noncentral Student’s \(t\) distribution: if \(npnt(df, np, t) = p\), then \(npnt(df, t, p) = np\)

- **nt(df, np, t)**
  - the cumulative noncentral Student’s \(t\) distribution with \(df\) degrees of freedom and noncentrality parameter \(np\)

- **ntden(df, np, t)**
  - the probability density function of the noncentral Student’s \(t\) distribution with \(df\) degrees of freedom and noncentrality parameter \(np\)

- **nttail(df, np, t)**
  - the reverse cumulative (upper tail or survivor) noncentral Student’s \(t\) distribution with \(df\) degrees of freedom and noncentrality parameter \(np\)
nullmat(matname)
nullmat(matname)
use with the row-join (,) and column-join (\) operators
plurals(n, s)
the plural of s if n ≠ ±1
plurals(n, s1, s2)
the plural of s1, as modified by or replaced with s2, if n ≠ ±1
poisson(m, k)
the probability of observing floor(k) or fewer outcomes that are
distributed as Poisson with mean m
poissonp(m, k)
the probability of observing floor(k) outcomes that are distributed
as Poisson with mean m
poisontail(m, k)
the probability of observing floor(k) or more outcomes that are
distributed as Poisson with mean m
qofd(\(e_d\))
the \(e_q\) quarterly date (quarters since 1960q1) containing date \(e_d\)
quarter(\(e_d\))
the numeric quarter of the year corresponding to date \(e_d\)
quarterly(s1, s2[], Y]
the \(e_q\) quarterly date (quarters since 1960q1) corresponding to \(s_1\)
based on \(s_2\) and \(Y\); \(Y\) specifies toyear; see date()
r(name)
the value of the stored result r(name); see [U] 18.8 Accessing
results calculated by other programs
rbeta(a, b)
beta(a, b) random variates, where a and b are the beta distribution
shape parameters
rbinomial(n, p)
binomial(n, p) random variates, where n is the number of trials and
p is the success probability
rcauchy(a, b)
Cauchy(a, b) random variates, where a is the location parameter and
b is the scale parameter
rchisq(df)
chi-squared, with \(df\) degrees of freedom, random variates
recode(x, x1, ..., x_n)
missing if \(x_1, x_2, \ldots, x_n\) is not weakly increasing; \(x\) if \(x\) is missing;
\(x_1\) if \(x \leq x_1\); \(x_2\) if \(x \leq x_2, \ldots\); otherwise, \(x_n\) if \(x > x_1, x_2, \ldots, x_{n-1}\).
\(x_i\) ≥ . is interpreted as \(x_i = +\infty\)
real(s)
s converted to numeric or missing
regexm(s, re)
performs a match of a regular expression and evaluates to 1 if regular
expression re is satisfied by the ASCII string s; otherwise, 0
regexr(s1, re, s2)
replaces the first substring within ASCII string \(s_1\) that matches re
with ASCII string \(s_2\) and returns the resulting string
regexs(n)
subexpression n from a previous regexm() match, where \(0 \leq n < 10\)
reldif(x, y)
the “relative” difference \(|x - y|/(|y| + 1)\); 0 if both arguments are
the same type of extended missing value; missing if only one
argument is missing or if the two arguments are two different
types of missing
replay()
1 if the first nonblank character of local macro ‘0’ is a comma, or
if ‘0’ is empty
return(name)
the value of the to-be-stored result r(name); see [P] return
exponential(b)
exponential random variates with scale b
rgamma(a, b)
gamma(a, b) random variates, where a is the gamma shape parameter
and b is the scale parameter
rhypergeometric(N, K, n)
hypergeometric random variates
rigaussian(m, a)
inverse Gaussian random variates with mean m and shape param-
eter a
rlaplace(m, b)
Laplace(m, b) random variates with mean m and scale parameter b
rlogistic()
logistic variates with mean 0 and standard deviation \(\pi/\sqrt{3}\)
rlogistic()  logistic variates with mean 0, scale \( s \), and standard deviation \( s\pi/\sqrt{3} \)
rlogistic(\( m, s \))  logistic variates with mean \( m \), scale \( s \), and standard deviation \( s\pi/\sqrt{3} \)
rnbinomial(\( n, p \))  negative binomial random variates
rnormal()  standard normal (Gaussian) random variates, that is, variates from a normal distribution with a mean of 0 and a standard deviation of 1
rnormal(\( m \))  normal(\( m, 1 \)) (Gaussian) random variates, where \( m \) is the mean and the standard deviation is 1
rnormal(\( m, s \))  normal(\( m, s \)) (Gaussian) random variates, where \( m \) is the mean and \( s \) is the standard deviation
round(\( x, y \)) or round(\( x \))  \( x \) rounded in units of \( y \) or \( x \) rounded to the nearest integer if the argument \( y \) is omitted; \( x \) (not ‘.’) if \( x \) is missing (meaning that round(.\( a \)) = .\( a \) and that round(.\( a, y \)) = .\( a \) if \( y \) is not missing) and if \( y \) is missing, then ‘.’ is returned
roweqnumb(\( M, s \))  the equation number of \( M \) associated with row equation \( s \); missing if the row equation cannot be found
rownfreeparms(\( M \))  the number of free parameters in rows of \( M \)
rownumb(\( M, s \))  the row number of \( M \) associated with row name \( s \); missing if the row cannot be found
rowsof(\( M \))  the number of rows of \( M \)
rpoisson(\( m \))  Poisson(\( m \)) random variates, where \( m \) is the distribution mean
rt(\( df \))  Student’s \( t \) random variates, where \( df \) is the degrees of freedom
runiform()  uniformly distributed random variates over the interval \((0, 1)\)
runiform(\( a, b \))  uniformly distributed random variates over the interval \((a, b)\)
runiformint(\( a, b \))  uniformly distributed random integer variates on the interval \([a, b]\)
rweibull(\( a, b \))  Weibull variates with shape \( a \) and scale \( b \)
rweibull(\( a, b, g \))  Weibull variates with shape \( a \), scale \( b \), and location \( g \)
rweibullph(\( a, b \))  Weibull (proportional hazards) variates with shape \( a \) and scale \( b \)
rweibullph(\( a, b, g \))  Weibull (proportional hazards) variates with shape \( a \), scale \( b \), and location \( g \)
\( s(name) \)  the value of stored result \( s(name) \); see [U] 18.8 Accessing results calculated by other programs
scalar(\( exp \))  restricts name interpretation to scalars and matrices
seconds(\( ms \))  \( ms/1,000 \)
sign(\( x \))  the sign of \( x \): \(-1 \) if \( x < 0 \), \( 0 \) if \( x = 0 \), \( 1 \) if \( x > 0 \), or missing if \( x \) is missing
sin(\( x \))  the sine of \( x \), where \( x \) is in radians
sinh(\( x \))  the hyperbolic sine of \( x \)
smallestdouble()  the smallest double-precision number greater than zero
soundex(\( s \))  the soundex code for a string, \( s \)
soundex_nara(\( s \))  the U.S. Census soundex code for a string, \( s \)
sqrt(\( x \))  the square root of \( x \)
ss(\( etc \))  the second corresponding to datetime \( etc \) (ms. since 01jan1960 00:00:00.000)


<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ssC(e_{tC})</code></td>
<td>the second corresponding to datetime <code>e_{tC}</code> (ms. with leap seconds since 01jan1960 00:00:00.000)</td>
</tr>
<tr>
<td><code>strcat(s_1, s_2)</code></td>
<td>there is no <code>strcat()</code> function; instead the addition operator is used to concatenate strings</td>
</tr>
<tr>
<td><code>strdup(s_1, n)</code></td>
<td>there is no <code>strdup()</code> function; instead the multiplication operator is used to create multiple copies of strings</td>
</tr>
<tr>
<td><code>string(n)</code></td>
<td>a synonym for <code>strofreal(n)</code></td>
</tr>
<tr>
<td><code>string(n, s)</code></td>
<td>a synonym for <code>strofreal(n, s)</code></td>
</tr>
<tr>
<td><code>strittrim(s)</code></td>
<td><code>s</code> with multiple, consecutive internal blanks (ASCII space character <code>char(32)</code>) collapsed to one blank</td>
</tr>
<tr>
<td><code>strlen(s)</code></td>
<td>the number of characters in ASCII <code>s</code> or length in bytes</td>
</tr>
<tr>
<td><code>strlower(s)</code></td>
<td>lowercase ASCII characters in string <code>s</code></td>
</tr>
<tr>
<td><code>strmatch(s_1, s_2)</code></td>
<td>1 if <code>s_1</code> matches the pattern <code>s_2</code>; otherwise, 0</td>
</tr>
<tr>
<td><code>strofreal(n)</code></td>
<td><code>n</code> converted to a string</td>
</tr>
<tr>
<td><code>strofreal(n, s)</code></td>
<td><code>n</code> converted to a string using the specified display format</td>
</tr>
<tr>
<td><code>strpos(s_1, s_2)</code></td>
<td>the position in <code>s_1</code> at which <code>s_2</code> is first found; otherwise, 0</td>
</tr>
<tr>
<td><code>strproper(s)</code></td>
<td>a string with the first ASCII letter and any other letters immediately following characters that are not letters capitalized; all other ASCII letters converted to lowercase</td>
</tr>
<tr>
<td><code>strreverse(s)</code></td>
<td>reverses the ASCII string <code>s</code></td>
</tr>
<tr>
<td><code>strrpos(s_1, s_2)</code></td>
<td>the position in <code>s_1</code> at which <code>s_2</code> is last found; otherwise, 0</td>
</tr>
<tr>
<td><code>strrtrim(s)</code></td>
<td><code>s</code> without trailing blanks (ASCII space character <code>char(32)</code>); equivalent to <code>strtrim(strrtrim(s))</code></td>
</tr>
<tr>
<td><code>strtoname(s[, p])</code></td>
<td><code>s</code> translated into a Stata 13 compatible name</td>
</tr>
<tr>
<td><code>strtrim(s)</code></td>
<td><code>s</code> without leading and trailing blanks (ASCII space character <code>char(32)</code>); equivalent to <code>strtrim(strrtrim(s))</code></td>
</tr>
<tr>
<td><code>strupper(s)</code></td>
<td>uppercase ASCII characters in string <code>s</code></td>
</tr>
<tr>
<td><code>subinstr(s_1, s_2, s_3, n)</code></td>
<td><code>s_1</code>, where the first <code>n</code> occurrences in <code>s_1</code> of <code>s_2</code> have been replaced with <code>s_3</code></td>
</tr>
<tr>
<td><code>subinword(s_1, s_2, s_3, n)</code></td>
<td><code>s_1</code>, where the first <code>n</code> occurrences in <code>s_1</code> of <code>s_2</code> as a word have been replaced with <code>s_3</code></td>
</tr>
<tr>
<td><code>substr(s, n_1, n_2)</code></td>
<td>the substring of <code>s</code>, starting at <code>n_1</code>, for a length of <code>n_2</code></td>
</tr>
<tr>
<td><code>sum(x)</code></td>
<td>the running sum of <code>x</code>, treating missing values as zero</td>
</tr>
<tr>
<td><code>sweep(M, i)</code></td>
<td>matrix <code>M</code> with <code>i</code>th row/column swept</td>
</tr>
<tr>
<td><code>t(df, t)</code></td>
<td>the cumulative Student’s <code>t</code> distribution with <code>df</code> degrees of freedom</td>
</tr>
<tr>
<td><code>tan(x)</code></td>
<td>the tangent of <code>x</code>, where <code>x</code> is in radians</td>
</tr>
<tr>
<td><code>tanh(x)</code></td>
<td>the hyperbolic tangent of <code>x</code></td>
</tr>
<tr>
<td><code>tc(l)</code></td>
<td>convenience function to make typing dates and times in expressions easier</td>
</tr>
<tr>
<td><code>tc(l)</code></td>
<td>convenience function to make typing dates and times in expressions easier</td>
</tr>
<tr>
<td><code>td(l)</code></td>
<td>convenience function to make typing dates and times in expressions easier</td>
</tr>
<tr>
<td><code>tden(df, t)</code></td>
<td>the probability density function of Student’s <code>t</code> distribution</td>
</tr>
<tr>
<td><code>th(l)</code></td>
<td>convenience function to make typing half-yearly dates in expressions easier</td>
</tr>
</tbody>
</table>
tin($d_1, d_2$) \[\text{true if } d_1 \leq t \leq d_2, \text{ where } t \text{ is the time variable previously } \text{tsset}\]

tm($l$) \[\text{convenience function to make typing monthly dates in expressions easier}\]

tobytes($s[, n]$) \[\text{escaped decimal or hex digit strings of up to 200 bytes of } s\]

tq($l$) \[\text{convenience function to make typing quarterly dates in expressions easier}\]

trace($M$) \[\text{the trace of matrix } M\]

trigamma($x$) \[\text{the second derivative of } \text{lngamma}(x) = d^2 \ln\Gamma(x)/dx^2\]

trunc($x$) \[\text{a synonym for } \text{int}(x)\]

ttail($df, t$) \[\text{the reverse cumulative (upper tail or survivor) Student’s } t \text{ distribution; the probability } T > t\]

tukeyprob($k, df, x$) \[\text{the cumulative Tukey’s Studentized range distribution with } k \text{ ranges and } df \text{ degrees of freedom; 0 if } x < 0\]

tw($l$) \[\text{convenience function to make typing weekly dates in expressions easier}\]

twithin($d_1, d_2$) \[\text{true if } d_1 < t < d_2, \text{ where } t \text{ is the time variable previously } \text{tsset}\]

uchar($n$) \[\text{the Unicode character corresponding to Unicode code point } n \text{ or an empty string if } n \text{ is beyond the Unicode code-point range}\]

udstrlen($s$) \[\text{the number of display columns needed to display the Unicode string } s \text{ in the Stata Results window}\]

udsubstr($s, n_1, n_2$) \[\text{the Unicode substring of } s \text{, starting at character } n_1 \text{, for } n_2 \text{ display columns}\]

uisdigit($s$) \[1 \text{ if the first Unicode character in } s \text{ is a Unicode decimal digit; otherwise, 0}\]

uisletter($s$) \[1 \text{ if the first Unicode character in } s \text{ is a Unicode letter; otherwise, 0}\]

ustrcompare($s_1, s_2[, loc]$) \[\text{compares two Unicode strings}\]

ustrcompareex($s_1, s_2[, loc, case, cslv, norm, num, alt, fr]$) \[\text{compares two Unicode strings}\]

ustrfix($s[, rep]$) \[\text{replaces each invalid UTF-8 sequence with a Unicode character}\]

ustrfrom($s, enc, mode$) \[\text{converts the string } s \text{ in encoding } enc \text{ to a UTF-8 encoded Unicode string}\]

ustrinvalidcnt($s$) \[\text{the number of invalid UTF-8 sequences in } s\]

ustrleft($s, n$) \[\text{the first } n \text{ Unicode characters of the Unicode string } s\]

ustrlen($s$) \[\text{the number of characters in the Unicode string } s\]

ustrlower($s[, loc]$) \[\text{lowercase all characters of Unicode string } s \text{ under the given locale } loc\]

ustrltrim($s$) \[\text{removes the leading Unicode whitespace characters and blanks from the Unicode string } s\]

ustrnormalize($s, norm$) \[\text{normalizes Unicode string } s \text{ to one of the five normalization forms specified by } norm\]

ustrpos($s_1, s_2[, loc]$) \[\text{the position in } s_1 \text{ at which } s_2 \text{ is first found; otherwise, 0}\]

ustrregexm($s, re[, noc]$) \[\text{performs a match of a regular expression and evaluates to 1 if regular expression } re \text{ is satisfied by the Unicode string } s; \text{ otherwise, 0}\]

ustrregexra($s_1, re, s_2[, noc]$) \[\text{replaces all substrings within the Unicode string } s_1 \text{ that match } re \text{ with } s_2 \text{ and returns the resulting string}\]
ustrregexrf($s_1$, re, $s_2$, [noc]) replaces the first substring within the Unicode string $s_1$ that matches re with $s_2$ and returns the resulting string.

ustrregexs(n) subexpression n from a previous ustrregexm() match.

ustrreverse(s) reverses the Unicode string s.

ustrright(s, n) the last n Unicode characters of the Unicode string s.

ustrrpos($s_1$, $s_2$, [n]) the position in $s_1$ at which $s_2$ is last found; otherwise, 0.

ustrrtrim(s) remove trailing Unicode whitespace characters and blanks from the Unicode string s.

ustrsortkey(s[,loc]) generates a null-terminated byte array that can be used by the sort command to produce the same order as ustrcompare().

ustrsortkeyex(s, loc, st, case, cslv, norm, num, alt, fr) generates a null-terminated byte array that can be used by the sort command to produce the same order as ustrcompare().

ustrtitle(s[,loc]) a string with the first characters of Unicode words titlecased and other characters lowercased.

ustrto(s, enc, mode) converts the Unicode string s in UTF-8 encoding to a string in encoding enc.

ustrtohex(s[,n]) escaped hex digit string of s up to 200 Unicode characters.

ustrtoname(s[,p]) string s translated into a Stata name.

ustrtrim(s) removes leading and trailing Unicode whitespace characters and blanks from the Unicode string s.

ustrunescape(s) the Unicode string corresponding to the escaped sequences of s.

ustrupper(s[,loc]) uppercase all characters in string s under the given locale loc.

ustrword(s, n[,loc]) the n-th Unicode word in the Unicode string s.

ustrwordcount(s[,loc]) the number of nonempty Unicode words in the Unicode string s.

usubinstr($s_1$, $s_2$, $s_3$, n) replaces the first n occurrences of the Unicode string $s_2$ with the Unicode string $s_3$ in $s_1$.

usubstr(s, n1, n2) the Unicode substring of s, starting at n1, for a length of n2.

vec(M) a column vector formed by listing the elements of M, starting with the first column and proceeding column by column.

vecdiag(M) the row vector containing the diagonal of matrix M.

week(e_d) the numeric week of the year corresponding to date e_d, the %td encoded date (days since 01jan1960)

weekly($s_1$, $s_2$, [Y]) the $e_w$ weekly date (weeks since 1960w1) corresponding to $s_1$ based on $s_2$ and Y; Y specifies topyear; see date()

weibull(a, b, x) the cumulative Weibull distribution with shape a and scale b.

weibull(a, b, g, x) the cumulative Weibull distribution with shape a, scale b, and location g.

weibullden(a, b, x) the probability density function of the Weibull distribution with shape a and scale b.

weibullden(a, b, g, x) the probability density function of the Weibull distribution with shape a, scale b, and location g.

weibullph(a, b, x) the cumulative Weibull (proportional hazards) distribution with shape a and scale b.
weibullph(a, b, g, x)  the cumulative Weibull (proportional hazards) distribution with shape a, scale b, and location g
weibullphden(a, b, x)  the probability density function of the Weibull (proportional hazards) distribution with shape a and scale b
weibullphden(a, b, g, x)  the probability density function of the Weibull (proportional hazards) distribution with shape a, scale b, and location g
weibullphtail(a, b, x)  the reverse cumulative Weibull (proportional hazards) distribution with shape a and scale b
weibullphtail(a, b, g, x)  the reverse cumulative Weibull (proportional hazards) distribution with shape a, scale b, and location g
weibulltail(a, b, x)  the reverse cumulative Weibull distribution with shape a and scale b
weibulltail(a, b, g, x)  the reverse cumulative Weibull distribution with shape a, scale b, and location g
wofd(e_d)  the ew weekly date (weeks since 1960w1) containing date e_d
word(s, n)  the nth word in s; missing (""") if n is missing
wordbreaklocale(loc, type)  the most closely related locale supported by ICU from loc if type is 1, the actual locale where the word-boundary analysis data come from if type is 2; or an empty string is returned for any other type
wordcount(s)  the number of words in s
year(e_d)  the numeric year corresponding to date e_d
yearly(s_1, s_2[, Y])  the ey yearly date (year) corresponding to s_1 based on s_2 and Y; Y specifies topyear; see date()
yh(Y, H)  the eh half-yearly date (half-years since 1960h1) corresponding to year Y, half-year H
ym(Y, M)  the em monthly date (months since 1960m1) corresponding to year Y, month M
yofd(e_d)  the ey yearly date (year) containing date e_d
yq(Y, Q)  the eq quarterly date (quarters since 1960q1) corresponding to year Y, quarter Q
yw(Y, W)  the ew weekly date (weeks since 1960w1) corresponding to year Y, week W