Truncated Multivariate Student & Normal Toolbox
Help Documentation

Z. I. Botev

Feb 2016

Truncated Multivariate Student & Normal Toolbox

Main functions in this toolbox include.

1. `mvNcdf(l,u,Sig,n)`, which uses a Monte Carlo sample of size n to estimate the cumulative distribution function, $\Pr(\mathbf{l} < \mathbf{X} < \mathbf{u})$, of the d-dimensional multivariate normal with zero-mean and covariance Σ, that is, $\mathbf{X} \sim N(\mathbf{0}, \Sigma)$;
2. `mvNqmc(l,u,Sig,n)` provides a Quasi Monte Carlo algorithm for medium dimensions (say, $d < 50$), in addition to the faster Monte Carlo algorithm in `mvNcdf`;
3. `mvrandn(l,u,Sig,n)` simulates n random vectors $\mathbf{X} \sim N(\mathbf{0}, \Sigma)$, conditional on $\mathbf{l} < \mathbf{X} < \mathbf{u}$;
4. `norminvp(p,l,u)` computes the quantile function at $p \in [0, 1]$ of the univariate $N(0, 1)$ distribution truncated to $[l, u]$, and with high precision in the tails;
5. `trandn(l,u)` is a fast random number generator from the univariate $N(0, 1)$ distribution truncated to $[l, u]$.
6. `mvTcdf(l,u,Sig,nu,n)`, which uses a Monte Carlo sample of size n to estimate the cumulative distribution function, $\Pr(\mathbf{l} < \mathbf{X} < \mathbf{u})$, of the d-dimensional multivariate student with zero-mean and covariance Σ and degrees of freedom ν, that is, $\mathbf{X} \sim t_\nu(\mathbf{0}, \Sigma)$;
7. `mvTqmc(l,u,Sig,nu,n)` provides a Quasi Monte Carlo algorithm for medium dimensions (say, $d < 50$), in addition to the faster Monte Carlo algorithm in `mvTcdf`;
8. `mvrandt(l,u,Sig,nu,n)` simulates n random vectors $\mathbf{X} \sim t_\nu(\mathbf{0}, \Sigma)$, conditional on $\mathbf{l} < \mathbf{X} < \mathbf{u}$;
9. `tregress(l,u,Sig,df,n)` simulates n pairs, $(\mathbf{Z}, R)$, so that $\frac{\sqrt{\nu}\mathbf{Z}}{R} \sim t_\nu(\mathbf{0}, \Sigma)$, conditional on $\mathbf{l} < \mathbf{X} < \mathbf{u}$;

Reference: Z. I. Botev (2017), *The Normal Law Under Linear Restrictions: Simulation and Estimation via Minimax Tilting*, Journal of the Royal Statistical Society, Series B, Volume 79, Part 1, pp. 1-24

Contents

- `mvNcdf(l,u,Sig,n)` - multivariate normal cumulative distribution
- `mvNqmc(l,u,Sig,n)` - multivariate normal cumulative distribution (Quasi Monte Carlo)
- `mvrandn(l,u,Sig,n)` - truncated multivariate normal generator
- `norminvp(p,l,u)` - normal quantile function with tail-precision
- `trandn(l,u)` - fast truncated normal generator
- `mvTcdf(l,u,Sig,nu,n)` - multivariate student cumulative distribution
- `mvTqmc(l,u,Sig,df,n)` - multivariate student cumulative distribution (Quasi Monte Carlo)
- `mvrandt(l,u,Sig,df,n)` - truncated multivariate normal generator
- `tregress(l,u,Sig,df,n)` - truncated student for Bayesian regression simulation

`mvNcdf(l,u,Sig,n)` - multivariate normal cumulative distribution

- Suppose we wish to estimate $\ell = \Pr(\mathbf{l} < A\mathbf{X} < \mathbf{u})$, where A is a full rank matrix and $\mathbf{X} \sim N(\mu, \Sigma)$.

```
d=10;Sig=gallery('randcorr',d);mu = ones(d,1);l=-rand(d,1);u=rand(d,1);A=rand(d,d);
```

We simply compute $\ell = \Pr(\mathbf{l} - A\mu < \mathbf{Y} < \mathbf{u} - A\mu)$, where $\mathbf{Y} \sim N(\mathbf{0}, A\Sigma A^\top)$

```
est=mvNcdf(l-A*mu,u-A*mu,A*Sig*A',10^4)


est =

      prob: 1.1630e-06
    relErr: 0.0039
     upbnd: 1.7293e-06
```

- Consider the following large-scale example with known probability of $1/(d+1)$

```
d=10^3;l=zeros(d,1);u=Inf(d,1);Sig=0.5*eye(d)+.5*ones(d,d);
est=mvNcdf(l,u,Sig,10^4)


est =

      prob: 9.9555e-04
    relErr: 0.0103
     upbnd: 0.0030
```

compare `est.prob` with exact value by computing relative error

```
abs(est.prob-1/(d+1))*(d+1)
```

```
ans =

    0.0034
```

`mvNqmc(l,u,Sig,n)` - multivariate normal cumulative distribution (Quasi Monte Carlo)

Compare errors using pseudo-random and quasi-random implementation for small to medium d.

```
d=20;l=zeros(d,1);u=Inf(d,1);Sig=randn(d,d);Sig=Sig*Sig';
estqmc=mvNqmc(l,u,Sig,10^5), est=mvNcdf(l,u,Sig,10^5)
```

```
estqmc =

      prob: 6.5066e-09
    relErr: 5.6823e-04
     upbnd: 1.5817e-08


est =

      prob: 6.5056e-09
    relErr: 0.0017
     upbnd: 1.5817e-08
```

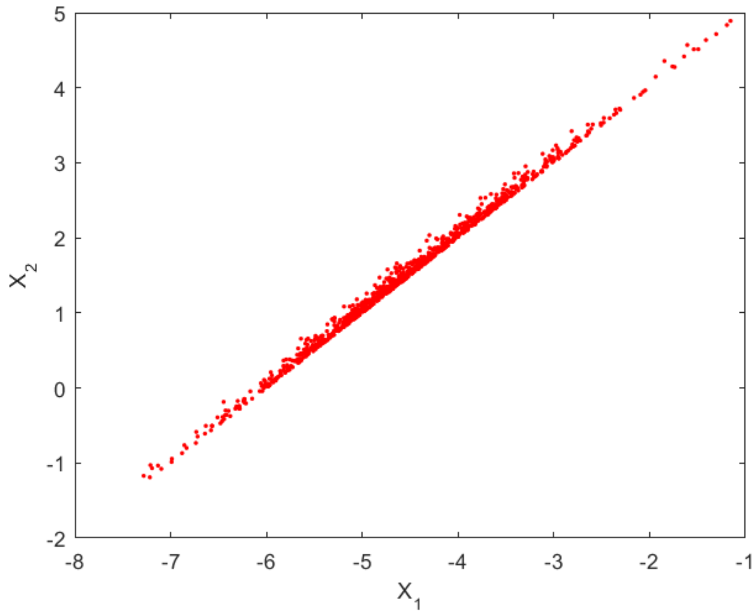
`mvrandn(l,u,Sig,n)` - truncated multivariate normal generator

- Suppose we wish to simulate a bivariate $\mathbf{X} \sim N(\mu, \Sigma)$, conditional on $X_1 - X_2 < -6$

```
Sig=[1,0.9;0.9,1];mu=[-3;0];l=[-Inf;-Inf];u=[-6;Inf];A=[1,-1;0,1];
```

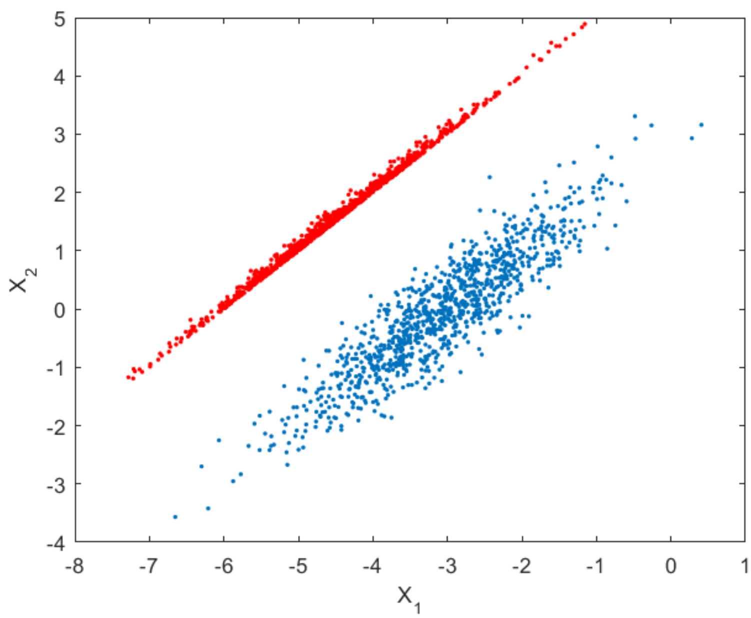
Simulate $\mathbf{Y} \sim N(\mathbf{0}, A\Sigma A^\top)$ conditional on $\mathbf{l} - A\mu < \mathbf{Y} < \mathbf{u} - A\mu$ and then set $\mathbf{X} = \mu + A^{-1}\mathbf{Y}$.

```
n=10^3;Y=mvrandn(l-A*mu,u-A*mu,A*Sig*A',n); X=repmat(mu,1,n)+A\Y;
plot(X(1,:),X(2,:),'r.'), xlabel('X_1'), ylabel('X_2'), hold on
```



Now superimpose the samples from the unconstrained Gaussian.

```
x=repmat(mu,1,n)+chol(Sig,'lower')*randn(2,n); plot(x(1,:),x(2,:),'.')
```

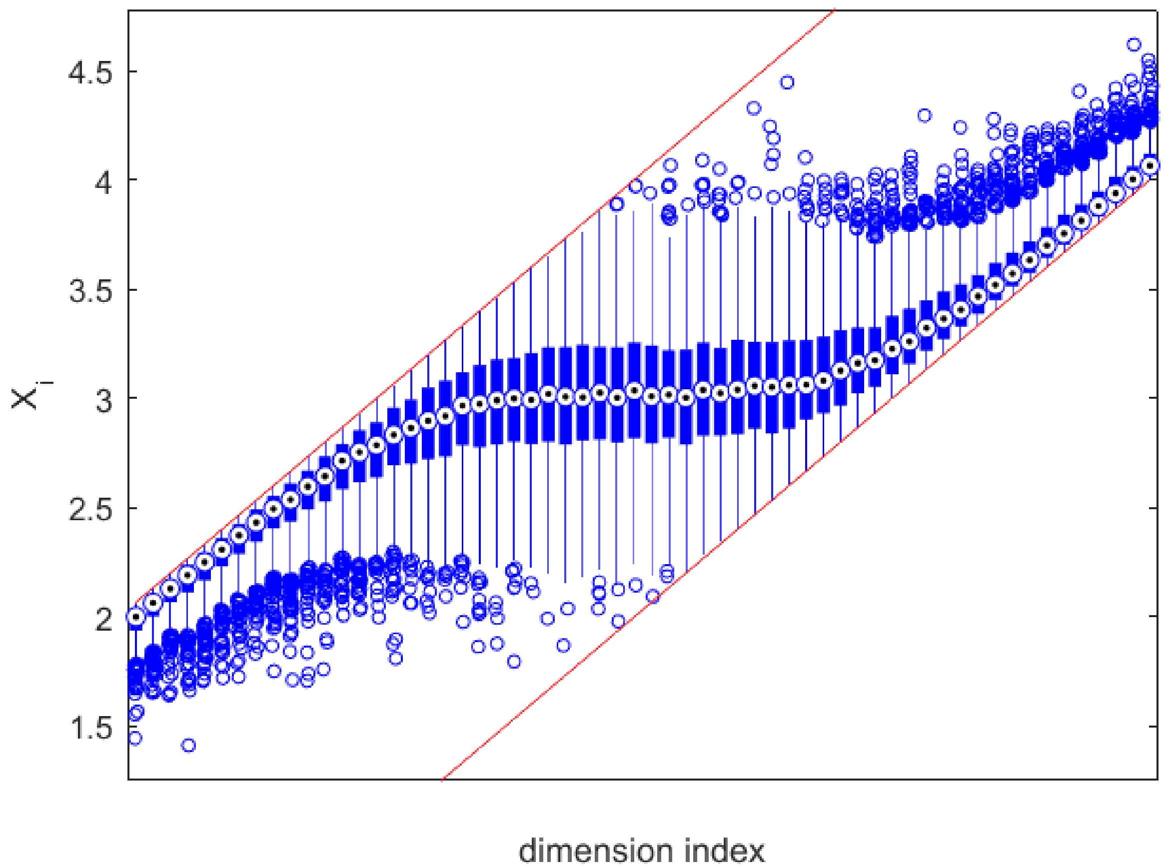


- Large-scale example with strong positive correlation.

```
d=60;n=10^3;
Sig=0.9*ones(d,d)+.1*eye(d);
l=(1:d)/d*4;u=l+2;
X=mvrandn(l,u,Sig,n);
```

Plot the boxplots of the d-marginal distributions together with their truncation limits.

```
boxplot(X','plotstyle','compact'),set(gca,'XTickLabel',{''}),
xlabel('dimension index'),ylabel('X_i'),hold on, plot(1:d,l,'r',1:d,u,'r')
```



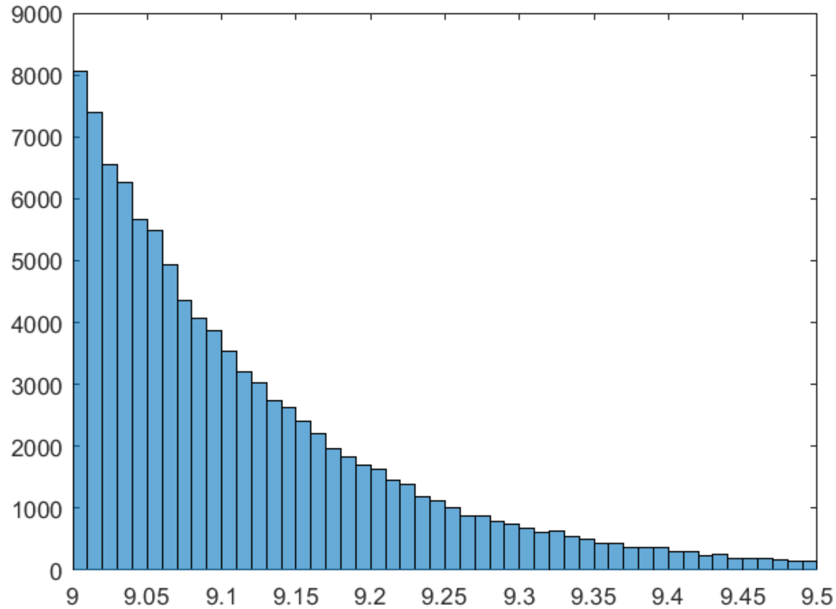
`norminvp(p,l,u)` - normal quantile function with tail-precision

Suppose we wish to simulate a random variable $Z \sim N(\mu, \sigma^2)$ conditional on $l < Z < u$ using the inverse transform method:

```
d=10^5;l=9*ones(d,1);u=9.5*ones(d,1);mu=1;sigma=1;
X=norminvp(rand(d,1),(l-mu)/sigma,(u-mu)/sigma);
Z=mu+sigma*X;
```

Now plot a histogram of the result.

```
hold off, histogram(Z)
```



`trandn(l,u)` - fast truncated normal generator

Simulate 10^6 samples with different truncation points.

```
l=rand(10^6,1)*70; u=Inf(10^6,1);
tic
trandn(l,u);
toc
```

```
Elapsed time is 0.322315 seconds.
```

Compare speed of fast generator with that of `norminvp.m`, the latter being useful for Quasi Monte Carlo estimation.

```
tic
norminvp(rand(size(l)),l,u);
toc
```

```
Elapsed time is 2.237242 seconds.
```

`mvTcdf(l,u,Sig,nu,n)` - multivariate student cumulative distribution

- Comparison with Matlab's default routine

```
d=20;l=ones(d,1)/2;u=ones(d,1);df=400;Sig=inv(0.5*eye(d)+.5*ones(d,d));
est=mvTcdf(l,u,Sig,df,10^4) % output of our method


est =

      prob: 1.7846e-37
    relErr: 0.0048
     upbnd: 2.8537e-37
```

Now execute Matlab's toolbox\stats\stats\mvtcdf.m and verify that with $n = 10^6$ it is slow and inaccurate.

```
options=optimset('TolFun',0,'MaxFunEvals',10^6,'Display','iter');
[prob,err]=mvtcdf(l,u,Sig,df,options)


estimate     error estimate    function evaluations
---------------------------------------------------
3.2071e-49      1.12248e-48           8650
3.3169e-49      4.47392e-49           21800
3.3649e-49      4.47109e-49           41650
3.3864e-49      4.47051e-49           71300
3.6281e-49      4.34551e-49           116650
3.6289e-49      4.34551e-49           184700
3.9017e-49      4.29858e-49           287350
4.3965e-49      4.21329e-49           441300
4.4061e-49      4.21321e-49           672350
Warning: Unable to achieve error tolerance of 0 in 1000000 function evaluations.
Increase the maximum number of function evaluations, or the error tolerance.

prob =

   4.4061e-49


err =

   4.2132e-49
```

mvTqmc(l,u,Sig,df,n) - multivariate student cumulative distribution (Quasi Monte Carlo)

Compare errors using pseudo-random and quasi-random implementation for small to medium d.

```
est=mvTqmc(l,u,Sig,df,10^4) % QMC version
est=mvTcdf(l,u,Sig,df,10^4) % ordinary Monte Carlo version


est =

      prob: 1.7820e-37
    relErr: 7.0824e-04
     upbnd: 2.8537e-37


est =

      prob: 1.7727e-37
    relErr: 0.0048
     upbnd: 2.8537e-37
```

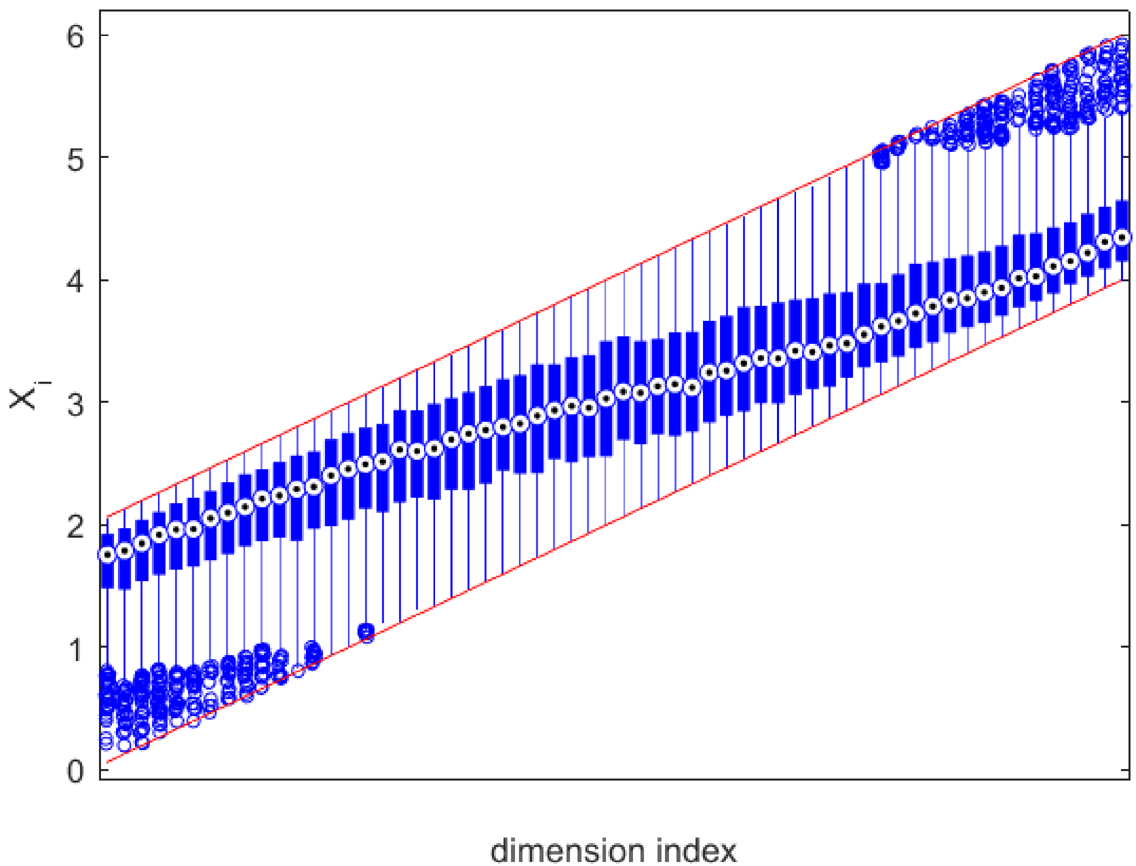
mvrandt(l,u,Sig,df,n) - truncated multivariate normal generator

- Large-scale example with strong positive correlation.

```
d=60;n=10^3;
Sig=0.9*ones(d,d)+.1*eye(d);
l=(1:d)/d*4;u=l+2; df=10;
X=mvrandt(l,u,Sig,df,n);
```

Plot the boxplots of the d-marginal distributions together with their truncation limits.

```
boxplot(X','plotstyle','compact'),set(gca,'XTickLabel',{''}),
xlabel('dimension index'),ylabel('X_i'),hold on, plot(1:d,l,'r',1:d,u,'r')
```



`tregress(l,u,Sig,df,n)` - truncated student for Bayesian regression simulation

- simulates n random pairs, $(\mathbf{Z}, R)$, such that $\frac{\sqrt{\nu}\mathbf{Z}}{R}$ has the same distribution as $\mathbf{X} \sim t_\nu(\mathbf{0}, \Sigma)$, conditional on $\mathbf{l} < \mathbf{X} < \mathbf{u}$. For example, we can repeat the above experiment as follows.

```
d=60;n=10^3;
Sig=0.9*ones(d,d)+.1*eye(d);
l=(1:d)/d*4;u=l+2; df=10;
[Z,R]=tregress(l,u,Sig,df,n);
X=bsxfun(@rdivide,sqrt(df)*Z,R);
```

`mvrorth(l,u,Sig,n)` - exact simulations from posterior of Probit regression

Example uses the **extramarital affairs** dataset from Ray C. Fair, *Journal of Political Economy* Vol. 86, No. 1 (Feb., 1978), pp. 45-61

Let the prior be $\beta \sim N(\mathbf{0}, \nu^2 I)$. We first simulate

$$\mathbf{Z} \sim N(0, \Sigma), \text{ where } \Sigma = I + \nu^2 X X^\top,$$

conditional on $\mathbf{Z} \geq \mathbf{0}$. Then, we simulate the posterior regression coefficients, β, of the Probit regression

$$(\beta | \mathbf{Z}) \sim N(C X^\top \mathbf{Z}, C), \text{ where } C^{-1} = I/\nu^2 + X^\top X.$$

```
load('affairs.csv'); % load data
Y = affairs(:,1); X = affairs(:,2:end); % response and design matrix
[m, d] = size(X); % dimensions of problem
X=diag(2*Y-1)*X; % incorporate response into design matrix
nu=sqrt(5); % prior scale parameter
C=inv(eye(d)/nu^2+X'*X);L=chol(C,'lower');Sig=eye(m)+nu^2*X*X';
l=zeros(m,1);u=inf(m,1);est=mvNcdf(l,u,Sig,10^3);
```

estimate the reciprocal of acceptance probability

```
est.upbnd/est.prob


ans =

  182.5406
```

sample $\mathbf{Z}$ from the truncated multivariate normal

```
tic
z=mvrandn(l,u,Sig,10^2);
toc


Elapsed time is 26.027357 seconds.
```

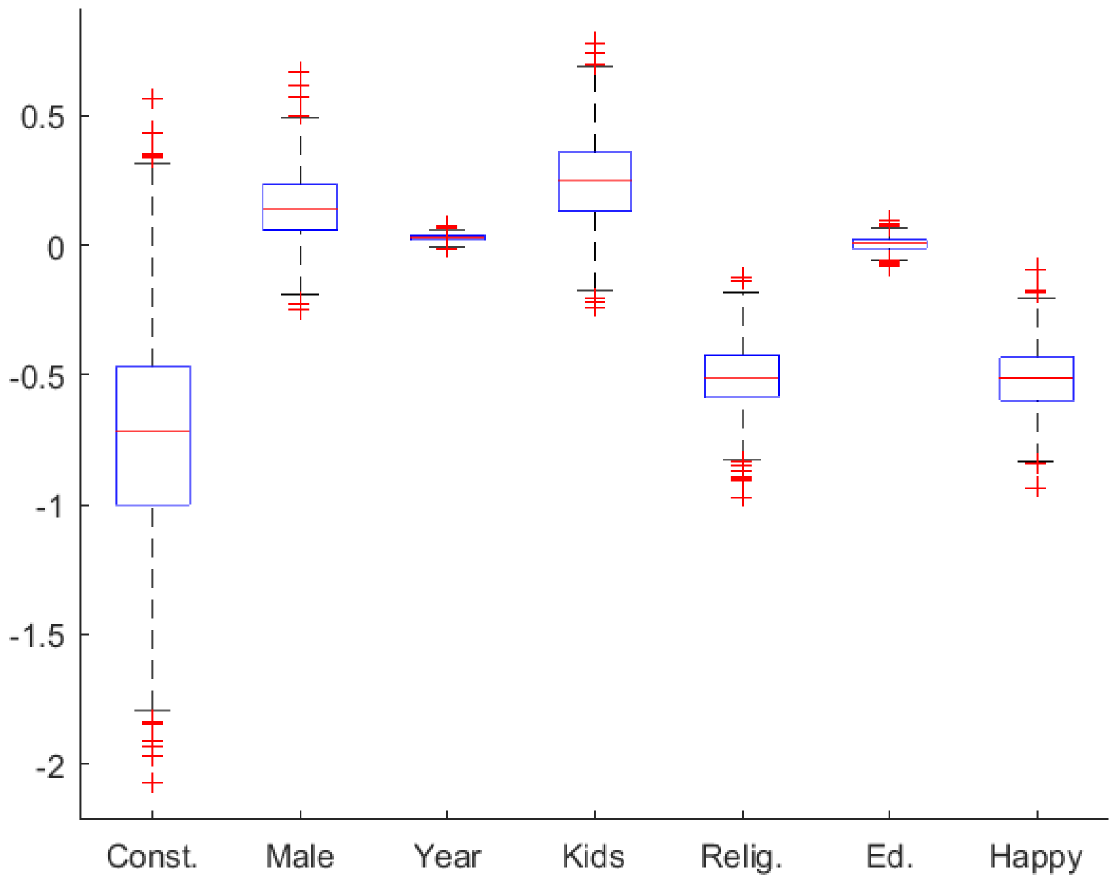
to speed up the simulation apply a different variable reordering via `cholorth.m`

```
tic
z=mvrorth(l,u,Sig,10^3);
toc


Elapsed time is 40.361316 seconds.
```

simulate β given $\mathbf{Z}$ and plot boxplots of marginals

```
beta=L*randn(d,size(z,2))+C*X'*z;
boxplot(beta','labels', ...
    {'Const.' 'Male' 'Year' 'Kids' 'Relig.' 'Ed.','Happy'}), box off
```



Exact Simulations for the Bayesian Posterior of the Tobit Regression

Example uses the **women's wage** dataset from T. A. Mroz, *Econometrica: Journal of the Econometric Society* Vol. 55, No. 4 (Jul., 1987), pp. 765-799

The response variables $\mathbf{y} = (y_1, \ldots, y_m)^\top$ in the Tobit model is modelled via:

$$Y_i = W_i \mathrm{I}\{u_i < W_i\} + u_i \mathrm{I}\{W_i \leq u_i\}, \text{ where } \mathbf{W} \sim N(X\beta, \sigma^2 I),$$

where $\mathbf{W}$ are *hidden* or *latent* variables; (β, σ) are the model parameters; and $X = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_d^\top]^\top$ is the matrix with predictors. We wish to sample from the Bayesian posterior with priors: $p(\beta)$ proportional to 1, and $p(\sigma)$ proportional to σ^{-2}. This gives the posterior:

$$f(\beta,\sigma) = \text{const.} \times \exp\left(-\sum_{i:y_i>u_i}\left(\frac{(y_i-\mathbf{x}_i^\top\beta)^2}{2\sigma^2}+\ln\sigma\right)+\sum_{i:y_i=u_i}\ln\Phi((u_i-\mathbf{x}_i^\top\beta)/\sigma)\right)\times\sigma^{-2}$$

An appropriate coordinate tranformation, $(\beta,\sigma)\mapsto(\mathbf{z},r)$, shows that simulating from the above posterior is equivelent to simulating from the truncated pdf:

$$f(\mathbf{z},r) = \text{const.}\times\exp\left(-\frac{\mathbf{z}^\top\mathbf{z}}{2}-\frac{r^2}{2}+(\nu-1)\ln r\right)\mathrm{I}\{\sqrt{\nu}\,L\mathbf{z}\geq r\mathbf{l}\}$$

for some lower triangular matrix L and threshold vector $\mathbf{l}$. In fact, the distribution of $\mathbf{X}=\sqrt{\nu}\frac{L\mathbf{Z}}{R}$ is the multivariate student $t_\nu(\mathbf{0},LL^\top)$ truncated to $\mathbf{X}\geq\mathbf{l}$. We can thus use `tregress.m` to perform this simulation.

```
Wage=csvread('private\WomenWage.csv',1,0); % load data
Y = Wage(:,1); m=length(Y); % response and design matrix
X = [ones(m,1),Wage(:,2:end)];
[m, d] = size(X); % dimensions of problem
Yl=Y(Y==0);Yu=Y(Y>0);Xl=X(Y==0,:); Xu=X(Y>0,:);
ml=length(Yl);Inv=inv(Xu'*Xu);Sig=eye(ml)+Xl*Inv*Xl';
s=Yu'*(eye(m-ml)-Xu*Inv*Xu')*Yu; % least squares residuals
s=sqrt(s);
nu=m-d-ml+1; % degrees of freedom
wh=Xl*Inv*Xu'*Yu; % w hat
l=sqrt(nu)*wh/s; % upper threshold for censoring is zero
```

Simulate $(\mathbf{Z},R)$ from a truncated student-type distribution:

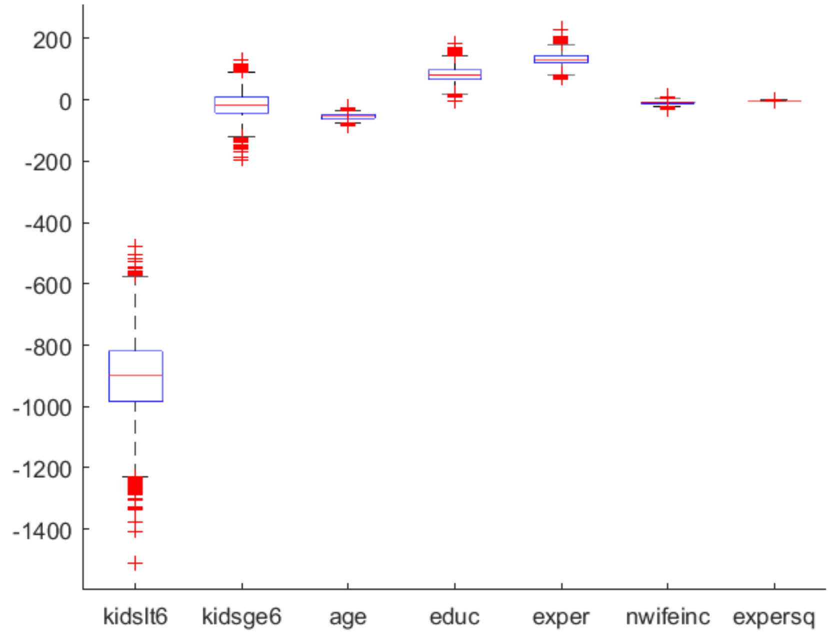
```
n=10^4;
[Z,R]=tregress(l,Inf(size(wh)),Sig,nu,n);
```

Reverse the mapping $(\beta,\sigma)\mapsto(\mathbf{z},r)$ to obtain samples from the posterior of β:

```
sig=s./R; % posterior distr. of sigma
C=inv(Xu'*Xu+Xl'*Xl);L=chol(C,'lower');
beta=nan(d,n);
for k=1:n
    W=wh-sig(k)*Z(:,k); % auxiliary variables
    beta(:,k)=C*(Xu'*Yu+Xl'*W)+sig(k)*L*randn(d,1);
end
```

Boxplot the marginal distributions of the posterior to assess statistical significance.

```
boxplot(beta(2:d,:)','labels', ...
    {'kidslt6','kidsge6','age','educ','exper','nwifeinc','expersq'}), box off
```



Plot marginal means and standard deviations

```
[mean(beta,2),prctile(beta,2.5,2),prctile(beta,97.5,2),std(beta,[],2)]
```

```
ans =

   1.0e+03 *

    0.9559    0.0307    1.8422    0.4607
   -0.9040   -1.1480   -0.6772    0.1200
   -0.0159   -0.0935    0.0610    0.0396
   -0.0549   -0.0701   -0.0397    0.0078
    0.0820    0.0397    0.1272    0.0226
    0.1330    0.0987    0.1695    0.0182
   -0.0090   -0.0181   -0.0000    0.0046
   -0.0019   -0.0030   -0.0008    0.0006
```


Reference: Z. I. Botev and P. L'Ecuyer (2015), *Efficient probability estimation and simulation of the truncated multivariate student-t distribution*, Proceedings of the 2015 Winter Simulation Conference, pages 380-391, (L. Yilmaz, W. Chan, I. Moon, T. Roeder, C. Macal, and M. Rossetti, eds.)