Paper 7660-2016

A SAS® Macro for Generating Random Numbers of Skew Normal and Skew *t* Distributions

Alan Ricardo da Silva, Universidade de Brasília, Dep. de Estatística, Brazil
Paulo Henrique Dourado da Silva, Banco do Brasil, Brazil

ABSTRACT

This paper aims to show a SAS® macro for generating random numbers of skew normal and skew *t* distributions as well as the quantiles of these distributions. The results are similar to those generated by 'sn' package of R software.

INTRODUCTION

Skew Normal (SN) and skew *t* (ST) distributions are a generalization of a normal and *t* distributions, respectively, allowing asymmetry by an inclusion of a third parameter λ. When $\lambda = 0$, then skew normal becomes the standard normal, when $\lambda > 0$ the distribution has positively skew, when $\lambda < 0$ the distribution has negatively skew and when $\lambda \to \infty$, then skew normal becomes the half normal distribution. This paper shows macros for generating random numbers of skew normal and skew *t* as well as for generating quantiles of these distributions, because this task it is not possible in any version of SAS so far.

SKEW NORMAL AND SKEW *T* DISTRIBUTIONS

The pdf of skew normal is given by (Azzalini, 1985):

$$f(x) = 2\emptyset(x)\Phi(\lambda x) \tag{1}$$

where $\emptyset(x)$ denotes the standard normal probability density function given by

$$\emptyset(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}} \tag{2}$$

and $\Phi(\,)$ is the normal cumulative distribution function given by

$$\Phi(x)\int_{-\infty}^{x}\emptyset(t)dt \tag{3}$$

This distribution was first introduced by O'Hagan and Leonard (1976).

To add location and scale parameters to (1), just let $x = \frac{y-\mu}{\sigma}$ and (1) becomes

$$f(y) = \frac{2}{\sigma}\emptyset\left(\frac{y-\mu}{\sigma}\right)\Phi\left(\lambda\frac{y-\mu}{\sigma}\right) \tag{4}$$

Let $\delta = \frac{\lambda}{\sqrt{1+\lambda^2}}$, then the mean and variance of (4) is given by:

$$\mu_{SN} = \mu + \sigma\delta\sqrt{\frac{2}{\pi}} \tag{5}$$

$$\sigma_{SN}^2 = \sigma^2\left(1 - \frac{2\delta^2}{\pi}\right) \tag{6}$$

Azzalini (2015) shows a simple way to generate random number of a skew normal distribution, as follows:

1. Sample u_0, u_1 having marginal distribution $N(0,1)$ and correlation δ. A simple way to achieve this is to generate u_0, v as independent $N(0,1)$ variates and define $u_1 = \delta u_0 + \sqrt{1-\delta^2}v$.

2.Then

$$z = \begin{cases} u_1, & if\ u_0 > 0 \\ -u_1, & \text{otherwise} \end{cases} \tag{7}$$

is a random number sampled from the SN distribution with shape parameter $\lambda = \delta/\sqrt{1-\delta^2}$.

3.To change the location and scale from (0,1) to (a, b) with $b > 0$, say, set $y = a + bz$.

The pdf of skew *t* is given by (Azzalini, 1985):

$$f(x) = 2t_v(x)T_{v+1}\left(\lambda x\sqrt{\frac{1+v}{v+x^2}}\right) \tag{8}$$

where $t_v(x)$is the standard *t* density probability function with v degrees of freedom and $T_{v+1}()$ is the cumulative distribution function of a *t* distribution with $v + 1$ degrees of freedom.

To obtain a ST variate, generate $V \sim \chi^2_\vartheta$ and put $w = \frac{z}{\sqrt{V/\vartheta}}$; then w has ST distribution with ϑ degrees of freedom and shape parameter equal to the one of z described in (7).

SAS® MACRO

The SAS® macros for generating random numbers of SN and ST are called `%SN` and `%ST`, respectively, and the parameters of the macros are:

```
%SN(n=,seed1=123,seed2=321,shape=0,mean=0,var=1,out=SN);
%ST(n=,seed1=123,seed2=321,shape=0,df=200,mean=0,var=1, out=ST);
```

n = amount of random numbers to be generated;

`seed1` = seed to be used in the first random normal distribution;

`seed2` = seed to be used in the second random normal distribution;

`shape` = asymmetry parameter λ;

`df` = degree of freedom for *t* distribution;

`mean` = desired average of the distribution;

`var` = desired variance of the distribution;

`out` = output for the random numbers.

Note that, except for the parameter n, all parameters have default values in order to facilitate the macro use.

The SAS® macros for generating quantiles of SN and ST called `%qSN` and `%qST`, respectively, use IML procedure and the parameters of the macros are:

```
%qSN(data=,var=,gamma=,shape=,out=QSN);
%qST(data=,var=,gamma=,shape=,out=QST);
```

`data` = dataset to be analyzed;

`var` = variable with the shape parameter;

`gamma` = confidence level $(0 < \gamma < 1)$;

`shape` = asymmetry parameter λ;

`out` = output for the generated quantile.

In this case, note that if one desires to create a series of quantiles based on a dataset, then it is possible to use the parameters `data` and `var`. Otherwise, one can just specify the parameters `gamma` and `shape` and leave blank the parameter `data`. The parameter `out` is the only one that has default value.

ILLUSTRATION

To illustrate the use of `%qSN` and `%qST` macros, we consider some specific cases and we compare the results with 'sn' package of R software. The macro calls are:

```
%qSN(gamma=0.975,shape=0);
%qSN(gamma=0.025,shape=0);
%qSN(gamma=0.975,shape=2);
%qSN(gamma=0.025,shape=2);
%qSN(gamma=0.975,shape=-2);
%qSN(gamma=0.025,shape=-2);
%qSN(gamma=0.975,shape=5);
%qSN(gamma=0.025,shape=5);
```

Shape	%qSN Macro		'sn' Package	
	0.975	**0.025**	**0.975**	**0.025**
0	1.959964	-1.959964	1.959964	-1.959964
2	2.241402	-0.503389	2.241402	-0.503389
-2	0.503389	-2.241402	0.503389	-2.241402

Table 1. Results of `%qSN` macro and 'sn' Package for quantiles 0.975 and 0.025

```
%qST(gamma=0.975,shape=0,df=200);
%qST(gamma=0.025,shape=0,df=200);
%qST(gamma=0.975,shape=2,df=200);
%qST(gamma=0.025,shape=2,df=200);
%qST(gamma=0.975,shape=0,df=10);
%qST(gamma=0.025,shape=0,df=10);
%qST(gamma=0.975,shape=-2,df=10);
%qST(gamma=0.025,shape=-2,df=10);
```

df	Shape	%qST Macro		'sn' Package	
		0.975	**0.025**	**0.975**	**0.025**
200	0	1.9718963	-1.9718963	1.971896	-1.971896
	2	2.2584026	-0.505154	2.258403	-0.5051538
10	0	2.2281389	-2.228139	2.228139	-2.228139
	-2	0.5412621	-2.633543	0.5412621	-2.633543

Table 2. Results of `%qSN` macro and 'sn' Package for quantiles 0.975 and 0.025

To illustrate the use of `%SN` and `%ST` macros, we consider some specific cases and we compare the results with the theoretical mean and variance of SN, given in (5) and (6). The macro calls are:

```
%SN(n=1000);
```

```
                The MEANS Procedure

Variable        Mean        Variance
--------------------------------------------------------
z           0.0225024     1.0048708
y           0.0225024     1.0048708
--------------------------------------------------------
```

Note that in this case, $\mu_{SN} = 0 + 1 \times 0\sqrt{\frac{2}{\pi}} \approx 0$ and $\sigma^2_{SN} = 1\left(1 - \frac{2\times 0}{\pi}\right) \approx 1$, for both Z and Y, once the default values are `SHAPE=0`, `MEAN=0` and `VAR=1`.

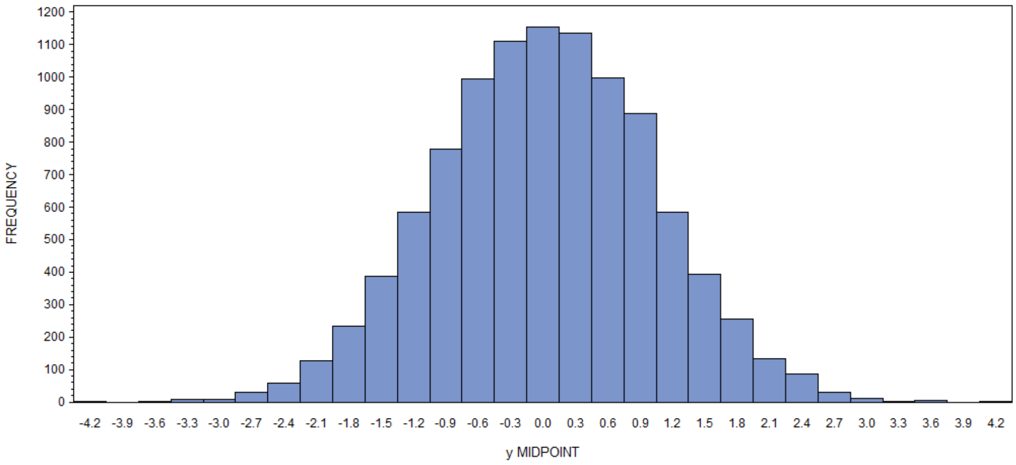

Figure 1. Distribution of Skew Normal with SHAPE = 0

```
%SN(n=1000, shape=-2);
```

The MEANS Procedure

Variable	Mean	Variance
z	-0.7024212	0.4899496
y	-0.7024212	0.4899496

In this case, $\mu_{SN} = 0 + 1 \times -\frac{2}{\sqrt{1+2^2}}\sqrt{\frac{2}{\pi}} = -0.71365$ and $\sigma_{SN}^2 = 1\left(1 - \frac{2\times(-2)^2}{\pi}\right) = 0.4907$, for both Z and Y, once the default values are `MEAN=0` and `VAR=1`.

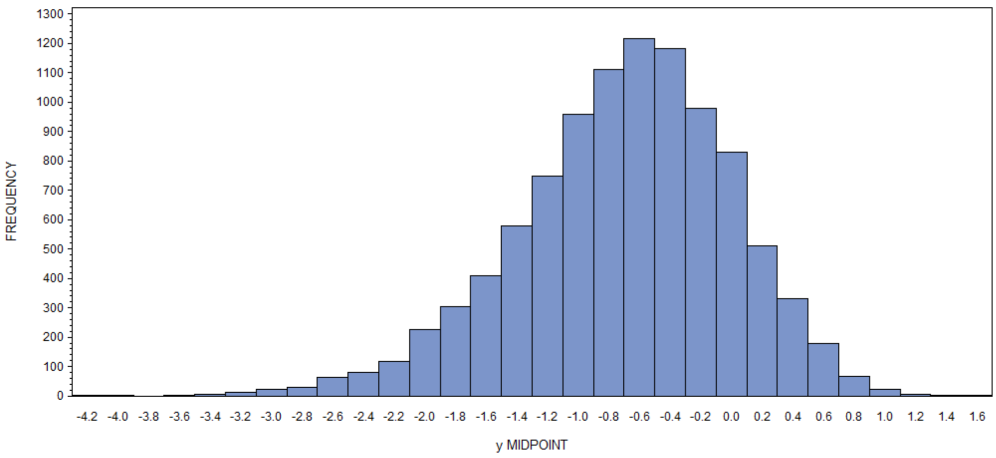

Figure 2. Distribution of Skew Normal with SHAPE $= -2$

```
%SN(n=1000, shape =-5);
```

The MEANS Procedure

Variable	Mean	Variance
z	-0.7766998	0.3852703
y	-0.7766998	0.3852703

Here, $\mu_{SN} = 0 + 1 \times -\frac{5}{\sqrt{1+5^2}}\sqrt{\frac{2}{\pi}} = -0.7824$ and $\sigma_{SN}^2 = 1\left(1 - \frac{2\times(-5)^2}{\pi}\right) = 0.3878$, for both Z and Y, again because the default values are `MEAN=0` and `VAR=1`.

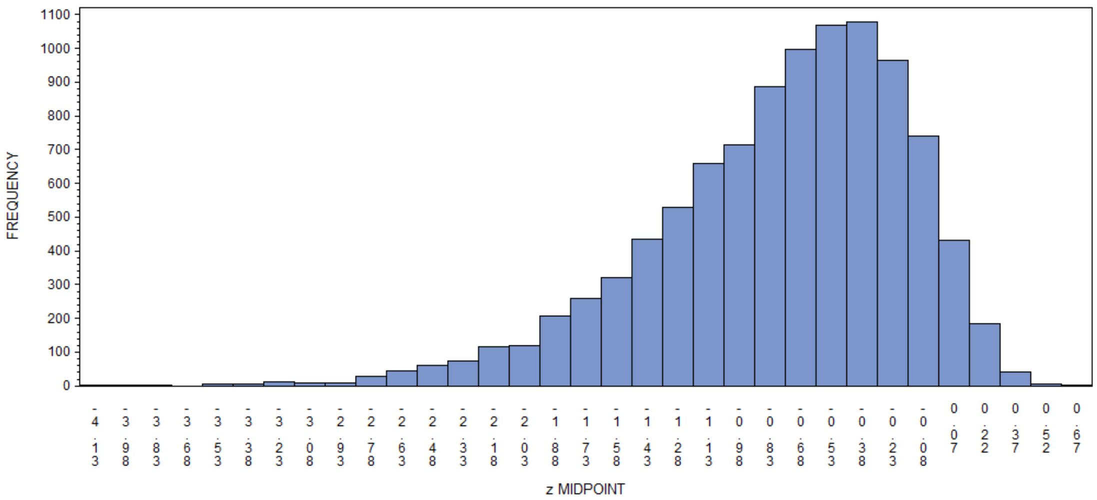

Figure 3. Distribution of Skew Normal with SHAPE $= -5$

```
%SN(n=1000, shape =5);
```

The MEANS Procedure

Variable	Mean	Variance
z	0.7855260	0.3843306
y	0.7855260	0.3843306

Using `SHAPE=5`, the value are the same when `SHAPE=-5` but with opposite signs.

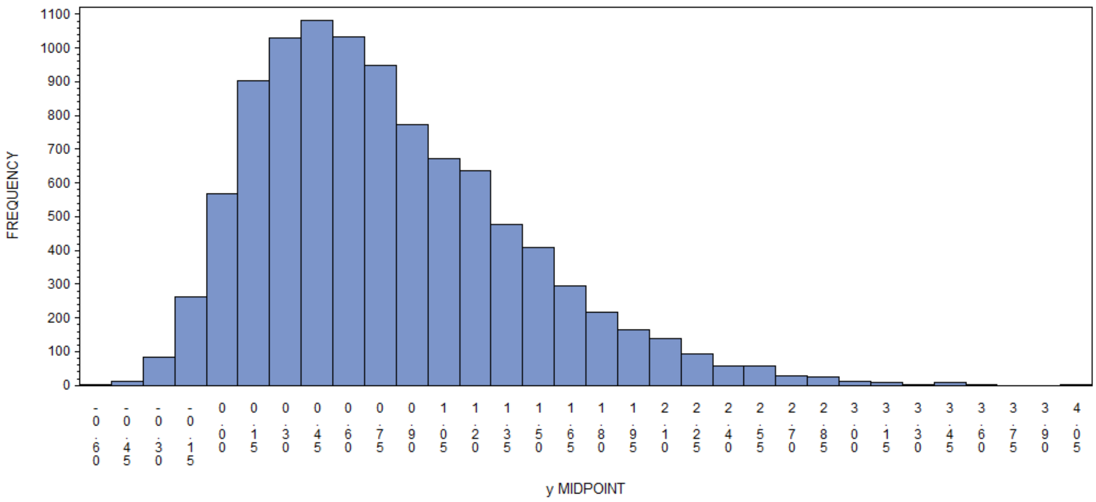

Figure 4. Distribution of Skew Normal with SHAPE = 5

```
%SN(n=1000, shape =5, mean=10, var=5);
```

The MEANS Procedure

Variable	Mean	Variance
z	0.7855260	0.3843306
y	11.7564896	1.9216532

Now using `MEAN=10` and `VAR=5`, the mean and variance for Z are the same than above, but for Y these values are computed as: $\mu_{SN} = 10 + \sqrt{5} \times -\frac{5}{\sqrt{1+5^2}}\sqrt{\frac{2}{\pi}} = 11.7495$ and $\sigma_{SN}^2 = 5\left(1 - \frac{2\times(-5)^2}{\pi}\right) = 1.9393.$

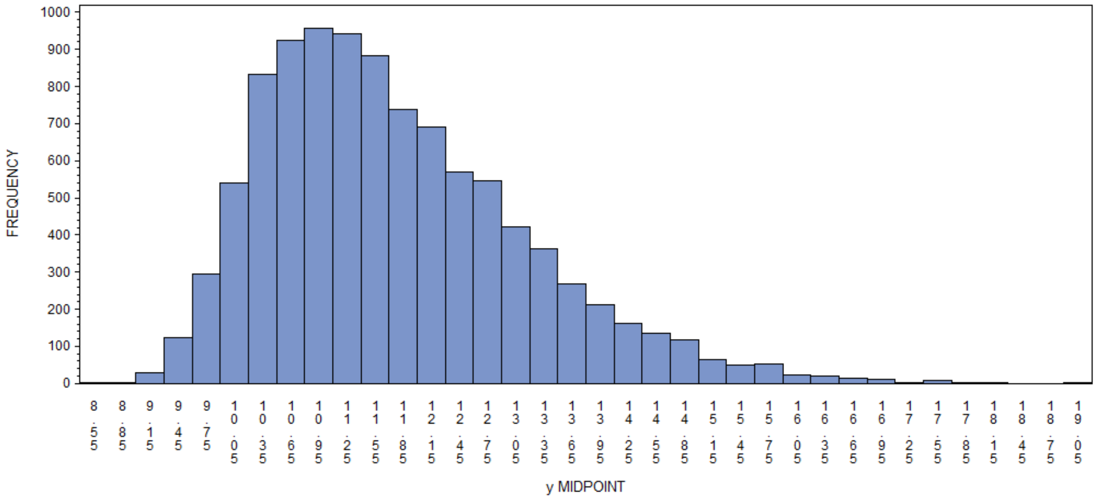

Figure 5. Distribution of Skew Normal with SHAPE = 5, MEAN = 10 and VAR = 5

Table 3 shows the quantiles from random numbers in order to check if the quantiles from random numbers converge to those quantiles generated by %qSN macro. We can see that the number are close, showing a good approximation of %SN macro**.**

Shape	%qSN Macro		%SN Macro	
	0.975	**0.025**	**0.975**	**0.025**
0	1.959964	-1.959964	1.987137	-1.925217
2	2.241402	-0.503389	2.230642	-0.505133
-5	0.125441	-2.241402	0.140201	-2.247990

Table 3. Quantiles 0.975 and 0.025 Generated by %qSN and %SN macros

The exercise is the same for %ST macro and some results are:

```
%ST(n=1000, shape =0, df=10);
```

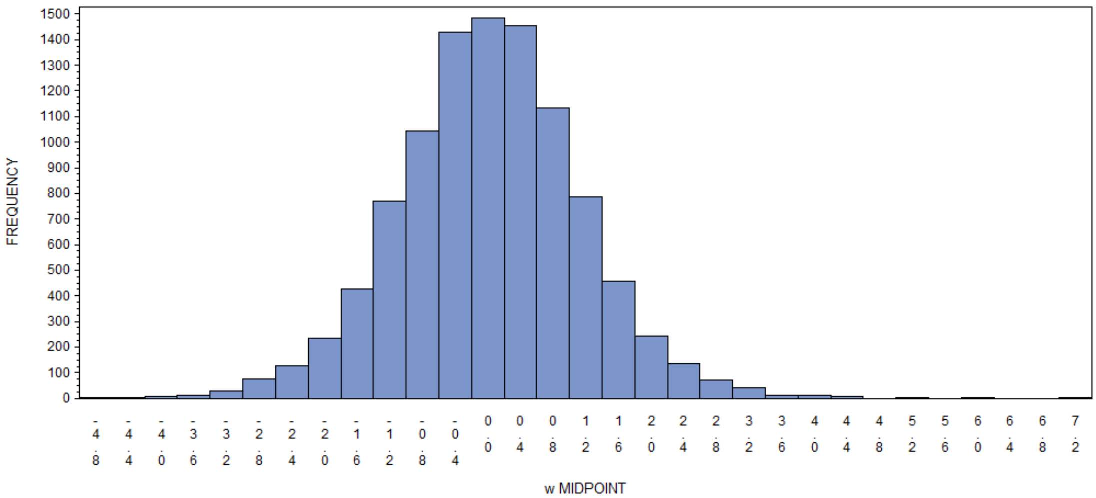

Figure 6. Distribution of Skew *t* with SHAPE = 0 and DF = 10

```
%ST(n=1000, shape =-2, df=10);
```

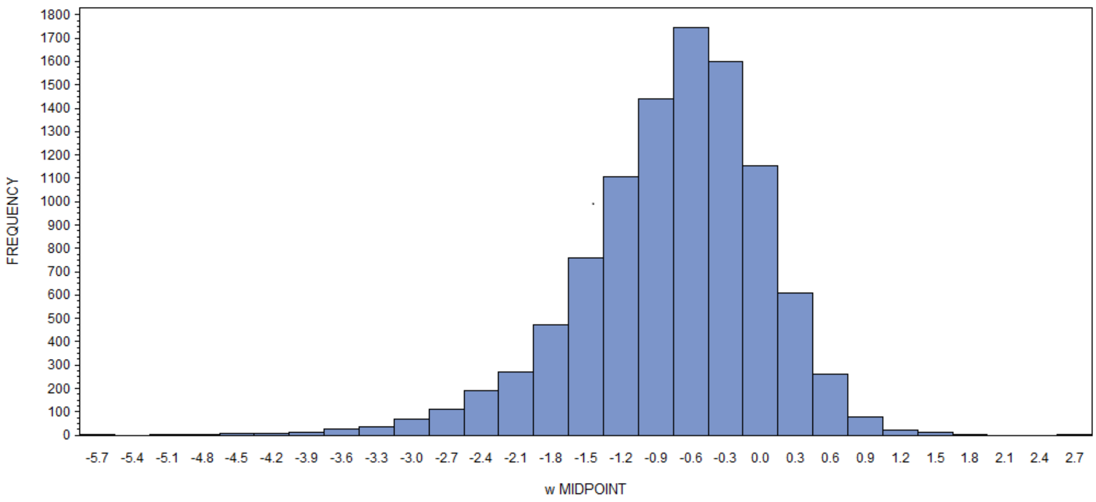

Figure 7. Distribution of Skew *t* with SHAPE = −2 and DF = 10.

```
%ST(n=1000, shape =0, df=200);
```

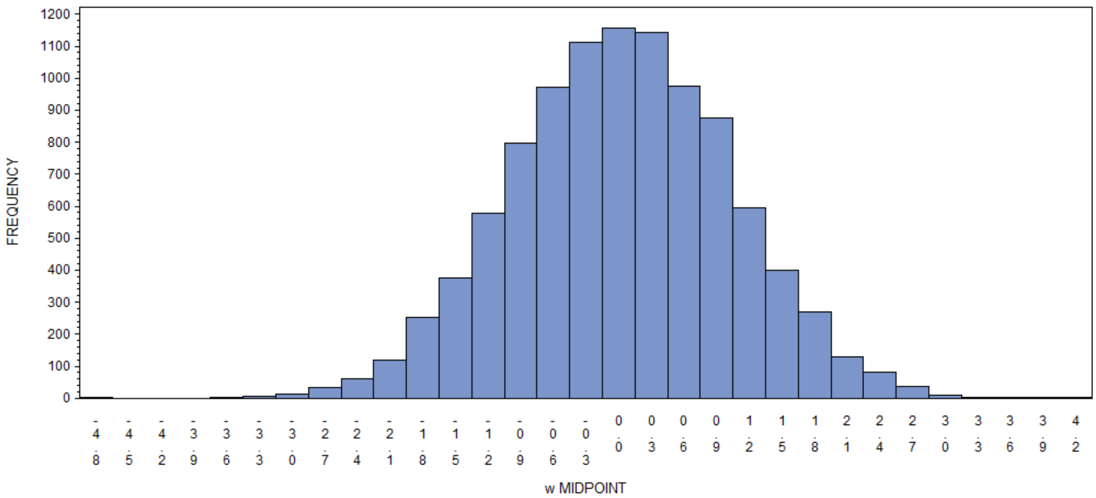

Figure 8. Distribution of Skew *t* with SHAPE $= 0$ and DF $= 200$

```
%ST(n=1000, shape =-2, df=200);
```

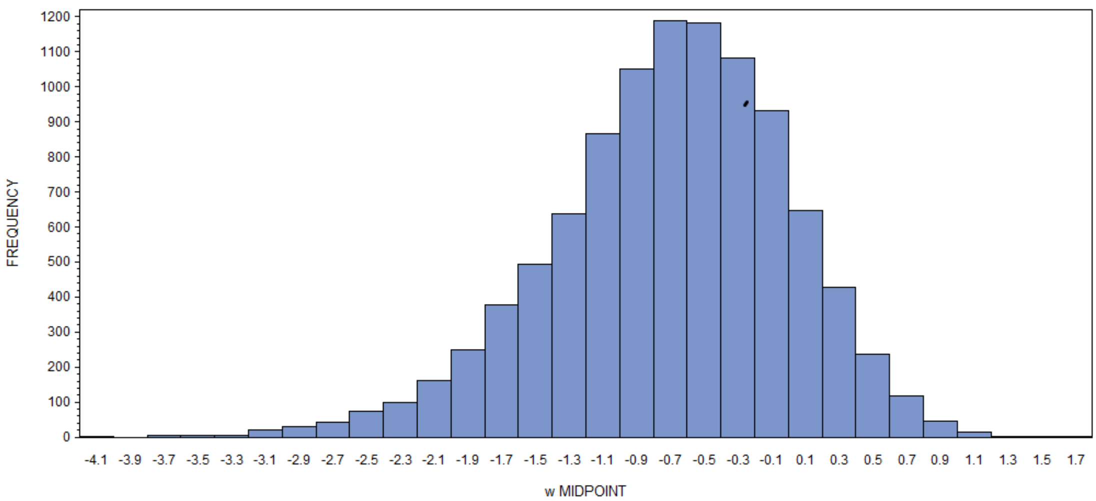

Figure 9. Distribution of Skew *t* with SHAPE $= -2$ and DF $= 200$

Table 4 shows the quantiles from random numbers in order to check if the quantiles from random numbers converge to those quantiles generated by %qST macro. We can see that the number are close, showing a good approximation of %ST macro**.**

df	Shape	%qST Macro		%ST Macro	
		0.975	**0.025**	**0.975**	**0.025**
200	0	1.9718963	-1.971896	1.961276	-1.8744130
	-2	0.5051542	-2.258402	0.530769	-2.5815742
10	0	2.2281389	-2.228139	1.338404	-2.256904
	-2	0.5412621	-2.633543	0.446271	-2.467777

Table 4. Quantiles 0.975 and 0.025 Generated by %qST and %ST macros

CONCLUSION

This paper showed SAS® macros for generating random numbers and quantiles of skew normal and skew *t* distributions. The results were close that those generated by 'sn' package of R software.

REFERENCES

Azzalini, A. 1985. "A Class of Distributions which Includes the Normal Ones". *Scand. J. Statist* 12: 171-178.

Azzalini, A. and Capitanio, A. 2003. "Distributions generated by perturbation of symmetry with emphasis on a multivariate skew-t distribution". *J.Roy. Statist. Soc. B*. 65: 367–389.

Azzalini, A. and Capitanio, A. 2014. *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Azzalini, A. (2015), "Random numbers with SN or ST distribution". Accessed August 7, 2015. http://azzalini.stat.unipd.it/SN/faq-r.html.

Jamalizadeh, A., Khosravi, M., and Balakrishnan, N. 2009. "Recurrence relations for distributions of a skew-t and a linear combination of order statistics from a bivariate-t". *Comp. Statist. Data An.* 53: 847–852.

O'Hagan, A. and Leonard, T. 1976. "Bayes estimation subject to uncertainty about parameter constraints". *Biometrika.* 63: 201-202.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Alan Ricardo da Silva
Enterprise: Universidade de Brasília
Address: Campus Universitário Darcy Ribeiro, Departamento de Estatística, Prédio CIC/EST sala A1 35/28
City, State ZIP: Brasília, DF, Brazil, 70910-900
Work Phone: +5561 3107 3672
E-mail: alansilva@unb.br
Web: www.est.unb.br

Name: Paulo Henrique Dourado da Silva
Enterprise: Banco do Brasil
Address: SAUN Quadra 5 Lote B - Torre I - 7º Andar
City, State ZIP: Brasília, DF, Brazil, 70040-912
Work Phone: +5561 3493 2263
E-mail: paulodourado@bb.com.br

APPENDIX I – SAS® MACROS

```
%macro qSN(data=,var=,gamma=,shape=,out=QSN);
proc iml;
       start dsn(x, shape);
    pdf = pdf("NORMAL",x);
    cdf = cdf("NORMAL", shape # x);
       sn=2 * pdf * cdf;
       return (sn);
       finish dsn;

       start dsn1(x) global(shape);
       shape=shape;
    pdf = pdf("NORMAL",x);
    cdf = cdf("NORMAL", shape # x);
       sn=2 # pdf # cdf;
       return(sn);
       finish dsn1;

       start psn(x);
       a = .M;
       b = x;
       call quad(t, "dsn1", a||b);
       return(t);
       finish psn;

       start cumulants_half_norm(n);
              n=max(n,2);
              n=int(2#ceil(n/2));
              half_n=int(n/2);
              m=0:(half_n-1);
              pi=constant('pi');
              a=sqrt(2/pi)/(gamma(m+1)#(2##m)#(2#m+1));
              signs=repeat({1 -1},half_n)[1:half_n];
              b=signs`#a;
              a=b//t(repeat(0,half_n));
              a=a[,1]//a[,2]; /*MODIFICAR*/
              coeff=repeat(a[1,],n);
              do k=2 to n;
```

```
                ind=1:(k-1);
                coeff[k,]=a[k,]-sum((ind`# coeff[ind,])#a[(k-1):1,]/k);
        end;
        kappa=coeff`#gamma((1:n)+1);
        kappa=kappa`;
        kappa[2,]=1+kappa[2,];
        return(kappa);
        finish;

   start sncumulants(shape);
   n=4;
   nrow=nrow(shape);
   delta=shape/sqrt(1+shape##2);
   *print delta;
   n0=n;
   n = max(n, 2);
   kv=cumulants_half_norm(n);
   if (nrow(kv)>n) then do;
        kv=kv[-(n+1),];
   end;
   kv[2,]=kv[2,]-1;
   seq=1:n;
   outer=delta##seq;
   kappa=outer`#kv;
   kappa[2,]=kappa[2,]+1;
   outer1=1;
   kappa=kappa#outer1`;
   kappa[1,]=kappa[1,];
   return(kappa);
   finish sncumulants;

   start qsn(p, shape);
   shape=shape;
   cum = sncumulants(shape);
   maxq = sqrt(quantile("CHISQUARE", p, 1));
minq = -sqrt(quantile("CHISQUARE", 1 - p, 1));
   g1 = cum[3]/cum[2]##(3/2);
g2 = cum[4]/cum[2]##2;
x = quantile("NORMAL", p);
x = (x + (x##2 - 1) # g1/6 + x # (x##2 - 3) # g2/24 - x #
```

```
        (2 # x##2 - 5) # g1##2/36);
    x = cum[1] + sqrt(cum[2]) # x;

       maxerr = 1;
       maxint = 1;
       do while (maxerr > 1E-6 & maxint<200);
        x1 = x - (psn(x) - p)/dsn(x, shape);
        x1 = min(x1, maxq);
        x1 = max(x1, minq);
        maxerr = max(abs(x1 - x)/(1 + abs(x)));
        x = x1;
         maxint = maxint + 1;
    end;
       y=x;
       return(y);
       finish qsn;
       %if &data= %then %do;
       shape=&shape;
       qsn=qsn(&gamma, shape[1]);
       print shape qsn;
       %end;
       %else %do;
       use &data;read all var{&var} into shapef;
       qsn =j(nrow(shapef),1,0);
       do i=1 to nrow(shapef);
       shape=shapef[i];
       qsn [i]=qsn(&gamma, shape[1]);
       end;
       %end;
create &out var{qsn};
append;
quit;
%mend qSN;


%macro qST(data=,var=,gamma=,shape=,df=,out=QST);
proc iml;
       start dst(x, shape, df);
    pdf = pdf("T",x, df);
    cdf = cdf("T", shape # x # sqrt((df + 1)/(x##2 + df)), df + 1);
```

```
   st=2 * pdf * cdf;
   return (st);
   finish dst;

   start dst1(x) global(shape, df);
pdf = pdf("T",x, df);
cdf = cdf("T", shape # x # sqrt((df + 1)/(x##2 + df)), df + 1);
   st=2 # pdf # cdf;
   return(st);
   finish dst1;

   start pst(x);
   a = .M;
   b = x;
   call quad(t, "dst1", a||b);
   return(t);
   finish pst;

   start stcumulants(shape, df1);
   df=max(5,df1);
   n=4;
   pi=constant("pi");
   n = min(n, 4);
   par =shape;
   nrow=nrow(par);

   delta=shape/sqrt(1+shape##2);
   mu=delta#sqrt(df/pi)#exp(log(gamma((df - 1)/2)) - log(gamma(df/2)));
   cum = J(nrow, n,0);
   cum[, 1] = mu;

   if n > 1 then do;
    cum[, 2] = df/(df - 2) - mu##2;
   end;
else; if n > 2 then do;
    cum[, 3] = mu#(df#(3 - delta##2)/(df - 3) - 3#df/(df - 2) + 2# mu##2);
   end;
else; if n > 3 then do;
    cum[, 4] = (3# df##2/((df - 2)#(df - 4)) - 4#mu##2 #
        df # (3 - delta##2)/(df - 3) + 6 # mu##2 # df/(df -
```

```
        2) - 3 # mu##4) - 3 # cum[, 2]##2;
    end;

    seq=1:n;
    outer=1;
    cum = cum # outer;
    cum[, 1] = cum[, 1];
    return(cum);
    finish;

    start qst(p, shape, df);
    cum = stcumulants(shape,df);
    maxq = sqrt(quantile("F", p, 1, 1));
minq = -sqrt(quantile("F", 1 - p, 1, 1));
    g1 = cum[3]/cum[2]##(3/2);
g2 = cum[4]/cum[2]##2;
x = quantile("NORMAL", p);
x = (x + (x##2 - 1) # g1/6 + x # (x##2 - 3) # g2/24 - x #
    (2 # x##2 - 5) # g1##2/36);
x = cum[,1] + sqrt(cum[,2]) # x;

    maxerr = 1;
    maxint = 1;
    do while (maxerr > 1E-6 & maxint<200);
      x1 = x - (pst(x) - p)/dst(x, shape, df);
      x1 = min(x1, maxq);
      x1 = max(x1, minq);
      maxerr = max(abs(x1 - x)/(1 + abs(x)));
      x = x1;
       maxint = maxint + 1;
end;
    y=x;
    return(y);
    finish qst;
    df=&df;
    %if &data= %then %do;
    shape=&shape;
    qst=qst(&gamma,shape[1],df[1]);
    print shape df qst;
    %end;
```

```
      %else %do;
      use &data;read all var{&var} into shapef;
      qst =j(nrow(shapef),1,0);
      do i=1 to nrow(shapef);
      shape=shapef[i];
      qst[i]=qst(&gamma,shape[1],df[1]);
      end;
      %end;
create &out var{qst};
append;
quit;
%mend qST;

%macro SN(n=,seed1=123,seed2=4321,shape=0,mean=0,var=1,out=SN);
data &out;
   do i = 1 to &n;
             mu_0=rannor(&seed1);
             v=rannor(&seed2);
             output;
   end;
 run;

data &out;set &out;
delta=&shape/sqrt(1+(&shape)**2);
mu_1 = delta* mu_0 + v* sqrt(1- delta**2 ) ;
if mu_0>=0 then z = mu_1;
                else z= - mu_1;
lambda= delta/ sqrt(1- delta**2);
y= &mean + sqrt(&var)*z ;
run;
quit;
proc gchart data=&out;vbar z /space=0;run;quit;
proc gchart data=&out;vbar y /space=0;run;quit;
%mend SN;


%macro ST(n=,seed1=123,seed2=4321,shape=0,df=200,mean=0,var=1,out=ST);
data &out;
   call streaminit(&seed1);
   do i = 1 to &n;
```

```
             mu_0=rannor(&seed1);
             v=rannor(&seed2);
             w=RAND('CHISQUARE',&df);
             output;
   end;
 run;

data &out;set &out;
delta=&shape/sqrt(1+(&shape)**2);
mu_1 = delta* mu_0 + v* sqrt(1- delta**2 ) ;
if mu_0>=0 then z = mu_1;
                else z= - mu_1;
lambda= delta/ sqrt(1- delta**2);
y= &mean + sqrt(&var)*z ;
W=y/sqrt(W/&df);
run;
quit;
proc gchart data=&out;vbar W /space=0;run;quit;
%mend ST;
```