Program 1

```
data calkowanie;

x=0;
t=0;
n=100;
h=sqrt(n);


 do i=1 to n;
   z=rand('Uniform');
   t=t+1;
  output;
 end;
run;

title"A sequence of random numbers drawn from uniform distribution";
proc print data=calkowanie noobs;
var z ;
run;
```

Program 1a

```
data calkowanie;

x=0;
t=0;
n=100;
h=sqrt(n);


 do i=1 to n;
   z=rand('Uniform');
   t=t+1;
  output;
 end;
```

```
run;


symbol value=none interpol=sms line=1 width=2;
title"z=Uniform distribution";
legend1 order=( 'z');
 proc gplot data=calkowanie;
   plot z*t /overlay legend=legend1;
 run;

 title 'Uniform Distribution ';
    proc univariate data=calkowanie;
      var z;
      histogram / midpoints=0.05 to 0.95 by 0.1

                  vaxis   = axis1
                  name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 2

```
data calkowanie;

x=0;
calka=0;
n=100000;

do i=1 to n;
  z=rand('Uniform');
  x=sin(z);
  calka=calka+x;
end;
wynik=calka/n;
output;
run;

Title'Value of integral';
proc print data=calkowanie;
```

```
var wynik;
run;
```

Program 2a

```
data calkowanie;

x=0;
calka=0;
n=100000;

do i=1 to n;
    z=rand('Uniform');
    x=sin(z)*4*(1-z)**3;
    calka=calka+x;
  end;
wynik=calka/n;
output;
run;

Title'Value of integral';
proc print data=calkowanie;
var wynik;
run;
```

Program 3

```
data beta;

x=0;
t=0;

n=1000;
h=sqrt(n);


do i=1 to n;
 z=rand('Beta',1,4);
 t=t+1;
 output;
end;
```

```
run;

symbol value=none interpol=sms line=1 width=2;
title"z=Beta distribution";
legend1 order=( 'z');
proc gplot data=beta;
plot z*t /overlay legend=legend1;
run;

 title 'Distribution uniform';

   proc univariate data=beta;
      var z;
      histogram / midpoints=0.05 to 0.95 by 0.1
                   beta
                  vaxis   = axis1
                  name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 4

```
data beta1;

x=0;
calka=0;
n=10000;

do i=1 to n;
 z=rand('Beta',1,4);
 x=sin(z);
 calka=calka+x;
  end;
value=calka/n;
output;
run;

Title'Value of integral';
```

```
proc print data=beta1;
var value;
run;
```

Program 5

```
data beta2;

a=3.8;
b=4.7;
n=1000;

do i=1 to n;
 z=rand('Beta',a,b);
  output;
  end;
run;




ods graphics on;

proc kde data=beta2; univar z/ gridl=0 gridu=1 bwm=1; run;

ods graphics off;
```

Program 6

```
data beta3;

a=3.8;
b=4.7;
x=0;
y=0;
obs=0;
calka1=0;
calka2=0;
n=1000;

do i=1 to n;
```

```
 z=rand('Beta',a,b);
 x=z;
 y=z**2;
 calka1=calka1+x;
 calka2=calka2+y;
 obs=obs+1;

  end;
meancalculated=calka1/n;
variancecalculated=calka2/n-meancalculated**2;
meantrue=a/(a+b);
variancetrue=a*b/((a+b+1)*(a+b)**2);
output;
run;

title 'Expectation and Variance of Beta Distribution';
proc print data=beta3;

var meantrue meancalculated variancetrue variancecalculated;
run;
```

Program 7

```
data gamma;

 k=1000;
 n=5;

 do i=1 to k;
  z=0;
  do j=1 to n;
  z=z+rand('exponential');
  end;
  r=z;
  output;
 end;

title 'Gamma Distribution';

   proc univariate data=gamma;
```

```
      var r;
      histogram / midpoints=0.05 to 5 by 1
                  gamma
                  vaxis   = axis1
                  name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 8

```
data poisson10;

 array T[20] T1-T20;
 array S[20] S1-S20;


     b=0;
     do k=1 to 20;
      T[k]= rand('Exponential');
      b=b+T[k];
      S[k]=b;
     end;

   p=0;
   r=0;
   x=0;
   delta=0.01;
           do i = 1 to 1500;/* czas obserwacji procesu 15 */
           r =r+delta;
           p=20;
               do j=1 to 20;
                 if r< S[j] then
                 do;
                     p=j-1;
                    leave;
                 end;
                end;
              x=r;
```

```
            output;
           end;
   run;


  Symbol value=none interpol=sms line=1  width=2;
title"Poisson";
proc gplot data=poisson10;
plot p*r x*r /overlay ;
run;
```

Program 9

```
data poisson;

   t=0;
   p=0;
   lambda=5;
   delta=0.01;
   output;

       do i = 1 to 1000;
     t =t+delta;
     p = p+ rand('Poisson',delta*lambda);
     x=lambda*t;
    output;
   end;
   run;


   Symbol value=none interpol=sms line=1  width=2;
   title"Trajectory Poisson";
   proc gplot data=poisson;
   plot p*t x*t /overlay ;
   run;
```

Program 10

```
data poisson1;

   t=0;
```

```
   p=0;
   lambda=1;
   delta=0.01;
   output;

       do i = 1 to 10000;
     t =t+delta;
     p = p+ rand('Poisson',delta*lambda);
     x=lambda*t;
 y=lambda*t+sqrt(2*t*log(log(t)));
 z=lambda*t-sqrt(2*t*log(log(t)));
    output;
   end;
   run;


Symbol value=none interpol=sms line=1  width=2;
title"Poisson";
proc gplot data=poisson1;
plot p*t x*t y*t z*t/overlay ;
run;
```

Program 11

```
 data poissonmix2;

   x=0;
   t=0;
   p=0;
   delta=0.01;
   c=delta**2/2;
   output;

       do i = 1 to 1000;
         t =t+delta;
         x=t**2/2;
         p = p+ rand('Poisson',c+t*delta);
        output;
       end;
   run;
```

```
   Symbol value=none interpol=sms line=1  width=2;
title"Poisson with a square  intensity function";
proc gplot data=poissonmix2;
plot p*t x*t /overlay ;
run;
```

Program 12

```
data walk;

   k=0;
   X=0;
   p=0.6;


   output;

       do i = 1 to 100;
     k=i;
     X = X+ 2*rand('binomial',p,1)-1;

    output;
   end;
   run;

   Symbol value=none interpol=sms line=1  width=2;
title"Random walk";
proc gplot data=walk;
plot X*k ;
run;
```

Program 13

```
data limit;
x=0; z=0; n=100;

     do i = 1 to 2000;
      x=0;
       do k=1 to n;
        x = rand('Exponential')+x;
```

```
        end;
       z=(x-n)/sqrt(n);
       output;
      end;
     run;

ods graphics on;

proc kde data=limit; univar z/ gridl=-3 gridu=3 bwm=1; run;

ods graphics off;
```

Program 14

```
data limit1;

n=200;
a=3;
b=10;
/* Mean and Variance of Beta distribution*/
ES=n*a/(a+b);
VarS=n*a*b/(((a+b)**2)*(a+b+1));

     do i = 1 to 200;
      S=0;
       do k=1 to n;
        S = rand('Beta',a,b)+S;
       end;
      z=(S-ES)/sqrt(VarS);
      output;
     end;
    run;


title 'Normal Distribution';

   proc univariate data=limit1;
      var z;
      histogram / midpoints=-3 to 3 by 0.5
```

```
                normal
                vaxis   = axis1
                name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 15

```
data cauchy;

n=2;

   do i = 1 to 2000;

  x=0;
      do k=1 to n;
        x = Rand('Cauchy')+x;
   end;
  Sn =x/n;
  T = rand('Cauchy');
  output;
    end;

  ods graphics on;

proc kde data=cauchy;
univar Sn T/gridl=-8 gridu=8
bwm=0.01;
run;

ods graphics off;
```

Program 16

```
data gamma;

x=0;
t=0;
n=1000;
h=sqrt(n);


 do i=1 to n;
   z=rand('gamma',16);
   t=t+1;
  output;
 end;
run;




 title 'Gamma Distribution ';
    proc univariate data=gamma;
      var z;
      histogram / midpoints=0.05 to 10 by 1

                  vaxis   = axis1
                  name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 17

```
data cramer;

/* Strong Laws of Large Numbers for S(t)*/

 lambdat=2;
```

```
 a=16;
 n=20000;
 array r[20000] r1-r20000;

   do i = 1 to n;
   poss=Rand('Poisson',lambdat);
   x=0;

    do k=1 to poss;
      x = Rand('gamma',a)+x;
    end;
     r[i]=x;
    end;

suma=0;

    do i=1 to n;
       suma=r[i]+suma;
    end;
    s=suma/n;
true_value=a*lambdat;
    run;


    title 'ES(t) ';
proc print data=cramer;
var s true_value ;
run;
```

Program 18

```
data central;

lambda=10;
a=16;

 n=20000;
```

```
   do i = 1 to n;
     poss=Rand('Poisson',lambda);
     s=0;

      do k=1 to poss;
       s = Rand('gamma',a)+s;
      end;
     z=(s-a*poss)/sqrt(poss*a);
 output;

    end;



title 'Limit distribution ';
    proc univariate data=central;
      var z;
      histogram / midpoints=-3 to 3 by 0.5
                  normal
                  vaxis   = axis1
                  name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 19

```
data limit1;

lambdat=80;
a=16;
n=2000;
 array r[2000] r1-r2000;

   do i = 1 to n;
   poss=Rand('Poisson',lambdat);
   x=0;
```

```
    do k=1 to poss;
      x = Rand('gamma',a)+x;
    end;
     r[i]=x;
    end;
/* Estimation of mean and variance*/
    suma=0;
    sumakw=0;
    do i=1 to n;
       suma=r[i]+suma;
     sumakw=r[i]**2+sumakw;
    end;
    sr=suma/n;
    var=sumakw/n-(sr)**2;
    odch=sqrt(var);

    do i=1 to n;
    z1=(r[i]-sr)/odch;
    output;
    end;
run;


title 'Limit distribution ';
    proc univariate data=limit1;
      var z1;
      histogram / midpoints=-3 to 3 by 0.5
                  normal
                  vaxis   = axis1
                  name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 20

```
data wiener;
```

```
   x=0;
   z=0;
   output;
   delta=0.01;
   n=1000;


       do i = 1 to n;
         x =x+delta;
         z = z+ rand('normal')*sqrt(delta);
         output;
       end;
   run;


Symbol value=none interpol=sms line=1  width=2;
title"Wiener process";
proc gplot data=wiener;
plot z*x ;
run;
```

Program 21

```
data wiener_ciesielski;

   K=10;
   N=1023; /*N=2**K-1  ilosc funkcji*/
   array W[1025] W1-W1025; /*Dimension of Array N+2*/

   W[1]=0;
   W[2**K+1]=rand('normal');

   do j= 1 to K;
     do i=0 to 2**(j-1)-1;
        r=((2*i+1)*(2**(K-j)))+1;
        s1=(i)*(2**(K-j+1))+1;
        s2=(i+1)*(2**(K-j+1))+1;
```

```
        W[r]=(W[s1]+W[s2])/2+(1/sqrt(2))**(j+1)*rand('normal');
      end;
    end;

  do j=0 to N+1;
    x=W[j+1];
    t=j/2**K;
    output;
   end;
run;

Symbol value=none interpol=sms line=1  width=2;
title"Wiener";
proc gplot data=wiener_ciesielski;
plot x*t;
run;
```

Program 22

```
data twonormal;

 do i=1 to 5000;
   z1=rand('normal');
   z2=rand('normal');
   x1=1+ z1-z2;
   x2=2+2*z2;
  output;
 end;

run;

/* a procedure to create a histogram and estimation of density*/

ods graphics on;

 proc kde data=twonormal;
 bivar x1 x2 / plots=all;
 run;
```

```
ods graphics off;
```

Program 23

```
data wiener_skorelowany;
   x=0;
   z1=0;
   z2=0;
   W=0;
   delta=0.01;
   output;
   rho1=0.95; /* correlation parameter  */
    rho2=sqrt(1-rho1**2);
   /* z1 and z2 are independent  Brownian motion */
       do i = 1 to 100;
       x =x+delta;
   z1 = z1+rand('normal')*sqrt(delta);
   z2= z2+rand('normal')*sqrt(delta);
    W=rho1*z1+rho2*z2;
   output;
  end;
   run;
```

```
Symbol value=none interpol=sms line=1  width=2;
title"Wiener correlated";
legend1 order=('z1' 'z2' 'b');
proc gplot data=wiener_skorelowany;
plot z1*x z2*x W*x/overlay legend=legend1 ;
run;
```

Program 24

```
data wiener;
```

```
   x=0;
   z=0;
   output;
   delta=0.01;
   n=100000;


       do i = 1 to n;
         x =x+delta;
         z = z+ rand('normal')*sqrt(delta);
 if x> 1 then y1=sqrt(2*x*log(log(x)));
          else y1=0;
 if x>1 then y2=-sqrt(2*x*log(log(x)));
         else y2=0;
         output;
       end;
   run;


Symbol value=none interpol=sms line=1  width=2;
title"Wiener process";
proc gplot data=wiener;
plot z*x y1*x y2*x /overlay;
run;
```

Program 25

```
data wiener_max;

   p=1000; /* number of  samples */
   K=10;
   N=1023; /*N=2**K-1  ilosc funkcji*/
   array W[1025] W1-W1025; /*Dimension of Array N+2*/

   W[1]=0;
   W[2**K+1]=rand('normal');


do n=1 to p;
```

```
   do j= 1 to K;
     do i=0 to 2**(j-1)-1;
        r=((2*i+1)*(2**(K-j)))+1;
        s1=(i)*(2**(K-j+1))+1;
        s2=(i+1)*(2**(K-j+1))+1;
        W[r]=(W[s1]+W[s2])/2+(1/sqrt(2))**(j+1)*rand('normal');
     end;
   end;
max=max(of W1-W1025);
output;
end;
```

```
title 'Max  distribution ';
    proc univariate data=wiener_max;
      var max;
      histogram / midpoints=0 to 3 by 0.1

                  vaxis   = axis1
                  name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 26

```
data wiener_Levy;

   delta=0.001;
   n=1000;
   p=2000; /* number of samples*/

   do k=1 to p;
       z=0;

    do i = 1 to n;
```

```
        y = z+ rand('normal')*sqrt(delta);
 if y*z<0 then w=i/n-0.01/n;

 z= y;
       end;
max=w;
output;
end;
   run;



title 'Arcsin distribution';

   proc univariate data=wiener_Levy;
      var max;
      histogram / midpoints=0 to 1 by 0.05
                   beta
                  vaxis   = axis1
                  name    = 'MyHist';
      inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
             / pos = ne  header = 'Summary Statistics';
      axis1 label=(a=90 r=0);
   run;
```

Program 27

```
data wiener_walk;

   z=0;
   y=0;
   output;
   delta=0.01;
   n=100;


       do i = 1 to n;

         z = z+ rand('normal')*sqrt(delta);
  y = y+ rand('normal')*sqrt(delta);
```

```
        output;
      end;
  run;


Symbol value=dot interpol=join line=1  width=2;
title"Wiener process";
proc gplot data=wiener_walk;
plot z*y ;
run;
```

Program 28

```
data black;
 mi=0.08;
 sigma=0.4;

 p=400; /* the number of samples*/

 delta=0.001; /*  delta*n=T in years */
 n=1000; /* the number of steps*/

   do k=1 to p;
     S=5;
       do j = 1 to n;
         y=rand('normal');
         S=S*(1+mi*delta+sigma*y*sqrt(delta));
       end;
      c=S/5;
     output;
   end;
   run;

 title ' Lognormal Distribution';
 ods select ParameterEstimates
        GoodnessOfFit FitQuantiles MyHist;
   proc univariate data=black;
      var c;
      histogram / midpoints=0.1 to 3 by 0.2
                  lognormal
```

```
                vaxis   = axis1
                name    = 'MyHist';
     inset n mean(5.3) std='Std Dev'(5.3) skewness(5.3)
            / pos = ne  header = 'Summary Statistics';
     axis1 label=(a=90 r=0);
  run;
```

Program 29

```
data ornstein;
 theta=0.08;
 sigma=0.4;
 mi=-1;
 krok=0.01;
 n=5000;


    t=0;
    x=5;
      do j = 1 to n;
    t =t+krok;
as=mi;
        y=rand('normal');
        x=x+theta*(mi-x)*krok+sigma*y*sqrt(krok);
output;
      end;

   run;

Symbol value=none interpol=sms line=1  width=2;
title" Ornstein - Uhlenbeck process";
proc gplot data=ornstein;
plot x*t as*t /overlay;
run;
```

Program 30

```
data Heston;


   x=0;
   w=0;
   z1=0;
   z2=0;
   S=5;
   ni=0.08;
   output;

   rho=0.9;
   eta=sqrt(1-rho**2);
   mi=0.05;
   krok=0.01;/* krok czasowy*/
   delta=0.05;
   beta=0.02;



  do i = 1 to 1000;
     y1 = rand('normal');
     y2 = rand('normal');
     x =x+krok;

     z2u=rho*z1+eta*w;

     z1 = z1+ y1*sqrt(krok);
     w= w+y2*sqrt(krok);

     z2=rho*z1+eta*w;
     ni=ni+2*(delta**2-beta*ni)*krok+delta*sqrt(ni)*(z2-z2u);
     S=S*(1+mi*krok+sqrt(ni)*y1*sqrt(krok));
     output;
   end;
   run;

Symbol value=none interpol=sms line=1  width=2;

title"Value of instantaneous variance";
```

```
proc gplot data=Heston;
plot  ni*x ;
run;

Symbol value=none interpol=sms line=1  width=2;
title"Value of an asset";
proc gplot data=Heston;
plot   s*x ;
run;
```