Distributions

void re_init

This function reinitializes the library. This basically initializes the seed for the random numbers once again if the user wants to deal with pseudo random distributions.

---

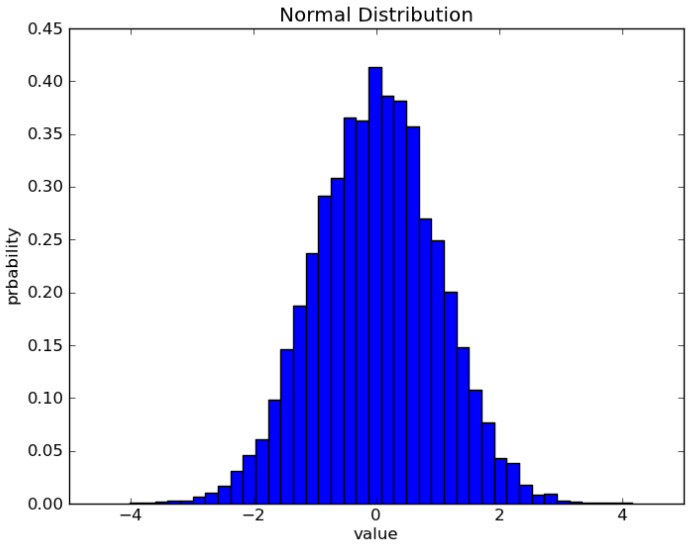
float normal(float mu, float sigma)

This function returns normally distributed random numbers with the mean as **mu** and standard deviation as **sigma**.

A sample code is shown here:

```
#include <stdio.h>
#include "distibutions"

int main(void) {


   for (int i = 0; i < 100; ++i) {

       //Generate a normal random variate with
       //mean = 0 and standard deviation = 1

       printf("%f\n",normal(0.0, 1.0));

   }

}
```

Here is a plot of the normal distribution

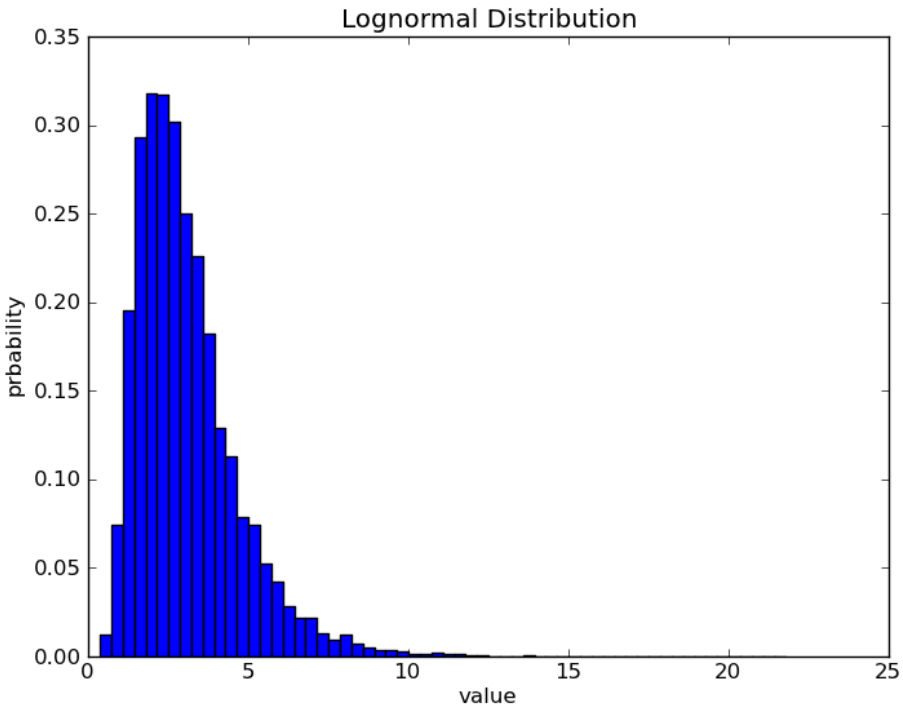


`float lognormal(float mu, float sigma);`

This function returns a lognormal variate. A sample implementation is shown here with the use of a graph library.

```
#include <stdio.h>
#include "distribution.h"
#include "grapher.h"
#define POINTS 10000



int main(void) {

   float data[POINTS];
   int i;
   for (i = 0; i < POINTS; ++i) {

       //Generate a lognormal random variate with
underlying normal parameters as
       //mean = 0 and standard deviation = 1
```

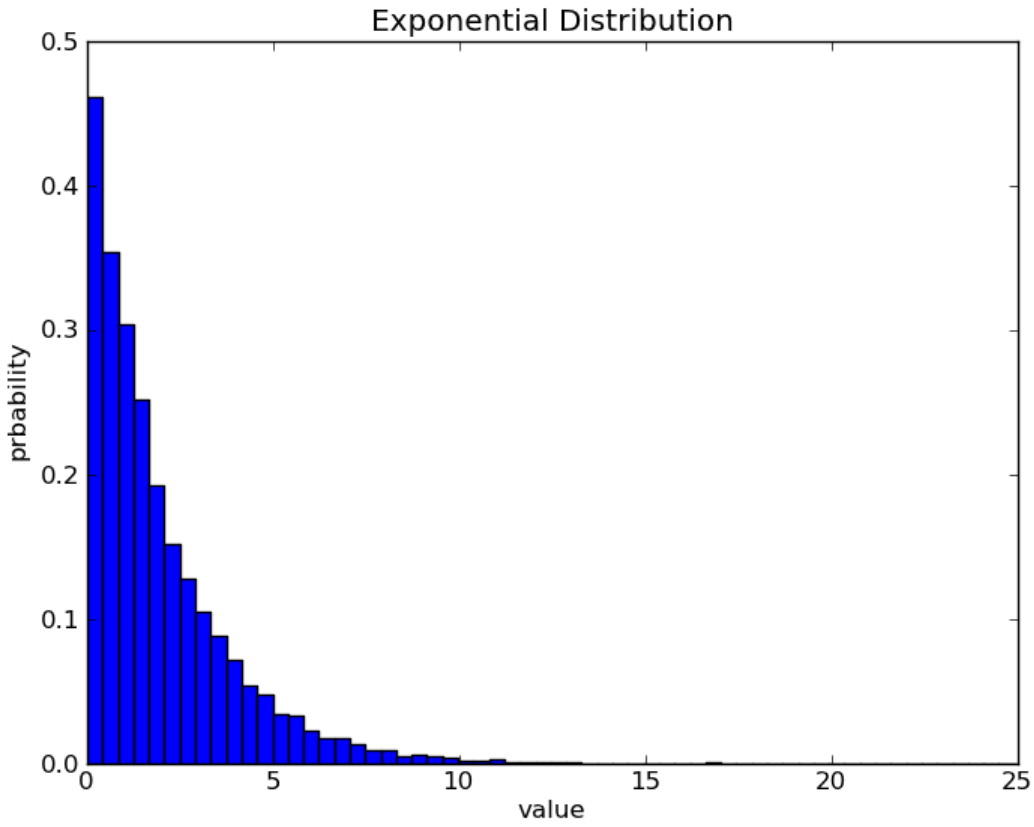
```
        data[i] = lognormal(1.0, 0.5);


    }

    graph_init();
    graph_new(HIST, data, POINTS,
"xlabel=value,ylabel=prbability,title=Lognormal
Distribution");
    graph_show();


}
```



float rexp(float lambda)

This function generates an exponentially distributed random variate with lambda as the rate parameter. Here is a sample code that uses a graphing library for plotting the resultant distribution.

```
#include <stdio.h>
#include "distribution.h"
#include "grapher.h"
#define POINTS 10000

int main(void) {

   float data[POINTS];
   int i;
   for (i = 0; i < POINTS; ++i) {

       //Generate a exponential random variate with
lambda as the rate parameter

       data[i] = rexp(0.5);


   }
   graph_init();
   graph_new(HIST, data, POINTS,
"xlabel=value,ylabel=prbability,title=Exponential
Distribution");
   graph_show();

}
```


Exponential Distribution
0.5
0.4
0.3
0.2
0.1
0.0
prbability
0
5
10
15
20
25
value