5634 - Stochastic Methods

Michelmas Term 2017-2018

Homework 1

1. **Download and compile (or install the package as part of the GNU Scientific Library) the DieHarder battery of random number tests from http://www.phy.duke.edu/~rgb/General/dieharder.php.**

Please find below the screenshot of the downloaded package "Dieharder"

```
saumya@saumya-VirtualBox:~/Downloads/dieharder-3.31.1$ sudo apt-get install dieharder
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.10.0-28 linux-headers-4.10.0-28-generic
  linux-image-4.10.0-28-generic linux-image-extra-4.10.0-28-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libdieharder3 libgsl2
Suggested packages:
  gsl-ref-psdoc | gsl-doc-pdf | gsl-doc-info | gsl-ref-html
The following NEW packages will be installed:
  dieharder libdieharder3 libgsl2
0 upgraded, 3 newly installed, 0 to remove and 119 not upgraded.
Need to get 1,132 kB of archives.
After this operation, 4,148 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ie.archive.ubuntu.com/ubuntu xenial/main amd64 libgsl2 amd64 2.1+dfsg-2 [840 kB]
Get:2 http://ie.archive.ubuntu.com/ubuntu xenial/universe amd64 libdieharder3 amd64 3.31.1-7 [121 kB]
Get:3 http://ie.archive.ubuntu.com/ubuntu xenial/universe amd64 dieharder amd64 3.31.1-7 [170 kB]
Fetched 1,132 kB in 1s (961 kB/s)
Selecting previously unselected package libgsl2:amd64.
(Reading database ... 243529 files and directories currently installed.)
Preparing to unpack .../libgsl2_2.1+dfsg-2_amd64.deb ...
Unpacking libgsl2:amd64 (2.1+dfsg-2) ...
Selecting previously unselected package libdieharder3.
Preparing to unpack .../libdieharder3_3.31.1-7_amd64.deb ...
Unpacking libdieharder3 (3.31.1-7) ...
Selecting previously unselected package dieharder.
Preparing to unpack .../dieharder_3.31.1-7_amd64.deb ...
Unpacking dieharder (3.31.1-7) ...
Processing triggers for libc-bin (2.23-0ubuntu9) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libgsl2:amd64 (2.1+dfsg-2) ...
Setting up libdieharder3 (3.31.1-7) ...
Setting up dieharder (3.31.1-7) ...
Processing triggers for libc-bin (2.23-0ubuntu9) ...
saumya@saumya-VirtualBox:~/Downloads/dieharder-3.31.1$ apt-cache policy dieharder
dieharder:
  Installed: 3.31.1-7
  Candidate: 3.31.1-7
  Version table:
 *** 3.31.1-7 500
        500 http://ie.archive.ubuntu.com/ubuntu xenial/universe amd64 Packages
        100 /var/lib/dpkg/status
saumya@saumya-VirtualBox:~/Downloads/dieharder-3.31.1$
```

The suggested packages are compiled as below:

```
saumya@saumya-VirtualBox:~/Desktop$ sudo apt-get install libgsl0-dev
[sudo] password for saumya:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libgsl-dev' instead of 'libgsl0-dev'
The following packages were automatically installed and are no longer required:
  linux-headers-4.10.0-28 linux-headers-4.10.0-28-generic linux-image-4.10.0-28-generic linux-image-extra-4.10.0-28-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  libgsl-dev
0 upgraded, 1 newly installed, 0 to remove and 125 not upgraded.
Need to get 962 kB of archives.
After this operation, 6,240 kB of additional disk space will be used.
Get:1 http://ie.archive.ubuntu.com/ubuntu xenial/main amd64 libgsl-dev amd64 2.1+dfsg-2 [962 kB]
Fetched 962 kB in 1s (842 kB/s)
Selecting previously unselected package libgsl-dev.
(Reading database ... 247251 files and directories currently installed.)
Preparing to unpack .../libgsl-dev_2.1+dfsg-2_amd64.deb ...
Unpacking libgsl-dev (2.1+dfsg-2) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libgsl-dev (2.1+dfsg-2) ...
saumya@saumya-VirtualBox:~/Desktop$ gcc -o try_1 try_1.c
```

Which random number generators can be considered good/bad according to this set of tests (name a few).

Please find below the tests that are considered good/bad in Dieharder:

```
saumya@saumya-VirtualBox:~/Desktop$ dieharder -l
#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
Installed dieharder tests:
 Test Number                          Test Name                 Test Reliability
==============================================================================
   -d 0                           Diehard Birthdays Test              Good
   -d 1                             Diehard OPERM5 Test               Good
   -d 2                     Diehard 32x32 Binary Rank Test            Good
   -d 3                       Diehard 6x8 Binary Rank Test            Good
   -d 4                           Diehard Bitstream Test              Good
   -d 5                                    Diehard OPSO             Suspect
   -d 6                                  Diehard OQSO Test          Suspect
   -d 7                                   Diehard DNA Test          Suspect
   -d 8                 Diehard Count the 1s (stream) Test             Good
   -d 9                   Diehard Count the 1s Test (byte)             Good
   -d 10                        Diehard Parking Lot Test               Good
   -d 11          Diehard Minimum Distance (2d Circle) Test            Good
   -d 12          Diehard 3d Sphere (Minimum Distance) Test            Good
   -d 13                          Diehard Squeeze Test                 Good
   -d 14                             Diehard Sums Test          Do Not Use
   -d 15                             Diehard Runs Test               Good
   -d 16                           Diehard Craps Test                Good
   -d 17                    Marsaglia and Tsang GCD Test             Good
   -d 100                             STS Monobit Test               Good
   -d 101                               STS Runs Test                Good
   -d 102                   STS Serial Test (Generalized)            Good
   -d 200                       RGB Bit Distribution Test            Good
   -d 201           RGB Generalized Minimum Distance Test            Good
   -d 202                         RGB Permutations Test              Good
   -d 203                           RGB Lagged Sum Test              Good
   -d 204                RGB Kolmogorov-Smirnov Test Test            Good
   -d 205                           Byte Distribution                Good
   -d 206                                    DAB DCT                 Good
   -d 207                          DAB Fill Tree Test               Good
   -d 208                        DAB Fill Tree 2 Test               Good
   -d 209                          DAB Monobit 2 Test               Good
saumya@saumya-VirtualBox:~/Desktop$
```

Explain one of the tests in more detail.

Diehard Birthday Test:

Step 1: Consider a year of "n" days. The test considers n of the order of 2^{24} (Keeping n of the order of 2^{18} or more, so then the results could be compared with Poisson distribution).

Step 2: Select "m" birthdays out of those "n" days.

Step 3: Place the birthdays at intervals and list the spacings. Let's call the spacing as "j".

j should be of the sample size of the order of ~500.

The distribution is assumed to be chi-squared distribution.

Step 4: Check if j is getting repeated.

If j is getting repeated, then it is asymptotically Poisson distributed with mean, $\lambda = 2^{(3/4n)}$

So, using $m = 2^{10}$.

$\lambda = 2^{30}/2^{26}$

⇨ $\lambda = 16$ (Poisson distribution with a mean value of 16)

Step 5: Pearson's Chi-square test is performed on the sampled j values, each test providing a p-value.

The first test uses bits 1–24 (counting from the left) from integers in the specified file. Then the file is closed and reopened.

Next, bits 2–25 are used to provide birthdays, then 3–26 and so on to bits 9–32.

Each set of bits provides a p-value, and the nine p-values provide a sample for a Kolmogorov–Smirnov test (K-S Test).

Step 6: K-S Test is performed to compare the samples.

2. Install Martin Luscher's RANLUX random number generator, obtainable (together with documentation and the link to his original paper) from his webpage http://luscher.web.cern.ch/luscher/

Please find below the compiled version of Ranlux random number generator.

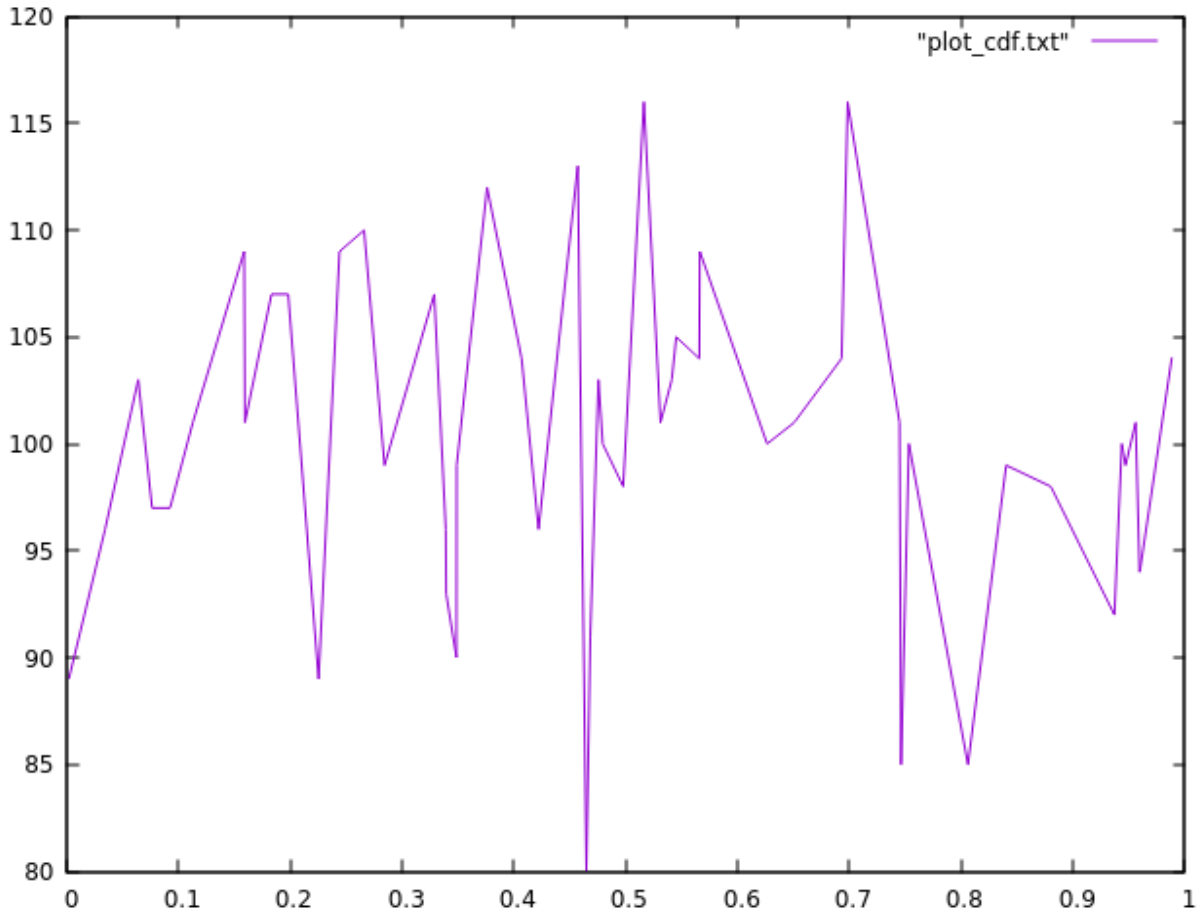
```
saumya@saumya-VirtualBox:~/Downloads/ranlux-3.3$ cc ranlxs.c ranlxd.c testlx.c -o testlx
saumya@saumya-VirtualBox:~/Downloads/ranlux-3.3$ ./testlx

All tests passed
=> ranlxs and ranlxd work correctly on this machine

saumya@saumya-VirtualBox:~/Downloads/ranlux-3.3$
```

3. **Write a program which uses a random number generator (i.e. continuous RV with uniform distribution on [0; 1)) to produce a binomial random variable. Choose values for n and p and plot the empirical CDF for your binomial distri-bution. Does it approach the expected CDF?**

The required program is attached in the file “binomial_random_variable.c”

Taking 500 independent experiments (n=500) & success probability of 0.2 (p = 0.2), 50 binomially distributed values are extracted on randomly generated values.

The graph of the binomial distribution created is as follows:



The graph starts from the value of (0.002, 89) to (0.4571, 113) to (0.9596, 94), approaching the expected CDF

4. **Write a program to estimate integrals of the type using uniform random numbers on the interval (e.g. from RANLUX). Then estimate the integrals of the functions:**
x, x^2, sqrt(x) by averaging over N samples.
The same has been attached in integral_a.c, integral_b.c, integral_c.c
Produce a plot of the results vs. N.

How large does N have to be to reproduce the analytic result to 2,3, or 4 digits, respectively?

X			x^2			x^(1/2)		
N	Value	Error	N	Value	Error	N	Value	Error
50	0.455373	0.040765	100	0.264859	0.025929	100	0.629817	0.021581
500	0.498074	0.012784	1000	0.333881	0.009326	1000	0.668633	0.007407
5000	0.501971	0.00411	10000	0.335008	0.002990	10000	0.668256	0.002351
50000	0.499548	0.00129	100000	0.332588	0.000942	100000	0.666381	0.000744
500000	0.499604	0.00041	100000	0.332588	0.000942	1000000	0.666581	0.000236
1000000	0.499867	0.000289	1000000	0.333152	0.000298	10000000	0.666662	0.000075
5000000	0.500149	0.000129	10000000	0.333331	0.000094	100000000	0.666689	0.000024
10000000	0.499995	0.000091				1000000000	0.666674	0.000007
100000000	0.500029	0.000029						
1E+09	0.500007	0.000009						

Can you estimate how large N would have to be to obtain 8 digit accuracy?

10^13