Lund University
Mathematical Sciences
Mathematical Statistics

HT2018

Computer Lab 1
Probability Theory

1 Introduction

The purpose of this computer lab is to give an introduction to R, and to illustrate some important concepts in probability theory. The probability theory part of the computer lab consists of three parts.

1. Random number generation.
2. The law of large numbers (LLN)
3. The central limit theorem (CLT)

Try to give answers to all questions. Do not continue until you have understood a part/moment, and ask when you do not understand.

2 Introduction to R

R is the most commonly used software package in mathematical statistics. New methods are often implemented in R with small delays and uploaded on the server. R is freely available (open source and GNU license), and can be downloaded for installation on UNIX machines, Windows and MacOS at http://www.r-project.org/. R is related to S-Plus, which however requires license fee.

First start R. This is done on the command line in UNIX environment by typing R, or Windows and MacOS by starting GUI'n R (click on the appropriate icon or the link).

One can find a good introduction on R at http://www.r-project.org/ under "Manuals" (left) and under the link "An introduction to R", try the html-version. However, note that much of this is in much details. However, sections 2 and 5 are general, and it is recommended to go through them.

In case of confusion, a useful command in R is "?" or "help". For example

```
help(sum)
```

or

```
?sum
```

If you do not know exactly what you are looking for try "help.search" och specify what you are looking for as a string (i.e. a series of characters). Try

```
help.search("sum")
help.search("normal")
```

R can be used for simple calculation. For example

```
1+2
34.7/23
```

One can define variables as:

```
a<-2
```

One can also try

```
a<-2
b<-13
c<-(a+b)/(1-2*b)
```

There is a large collection of the commonly used mathematical functions in-built in R. Try

```
log(10)
cos(0)
```

Vectors can be defined as:

```
x <- c(1, 2, 7.1, 4.4, -23.7)
```

Elementary vector and scalar operations can be done, for example

```
x*x
2*x+1
```

and functional values for variables can be defined as

```
sin(x)
exp(x)
sum(x)
```

Scalar products of two vectors can be evaluated as

```
y<-c(2,3,2,1,5)
sum(x*y)
x%*%y
```

Simple logical operations can be done, for example

```
x>=y
x[x>=y]
```

Sometimes instead of writing TRUE or FALSE, you may want to assign values zero or one. This can be done using the command "as.double", for instance

```
ind<-as.double(x>=y)
```

Note that this can be used to define indicator variables, and even random or stochastic variables (thus Bernoulli random variables). Matrices can be defined as "array", for example

```
array(c(x,y),dim=c(5,2))
array(c(x,y),dim=c(2,5))
array(1:10,dim=c(2,5))
array(1:5,dim=c(2,5))
array(1:5,dim=c(5,2))
```

One can index the arrays of the matrix as:

```
z<-array(c(x,y),dim=c(5,2))
z[2,1]
z[,1]
z[,2]
z[1,]
```

One can obtain the transpose of a matrix using "t()"

```
z
t(z)
```

In R one can apply functions on matrices just as on variables.

```
sin(z)
exp(z)
```

Operations are element-wise applications, for example

```
z*z
z^2
```

Matrix multiplications can be done by "$\% * \%$", for example

```
z%*%t(z)
t(z)%*%z
```

while multiplication of a matrix and a scalar can be done as:

```
x%*%z
t(z)%*%x
```

Matrix inversion can be done using "solve", (also see help for the solution of systems of linear equations)

```
u<-t(z)%*%z
solve(u)
```

3 Generation of random numbers

In principle, to generate random numbers (that is realization of one-dimensional random variables) it is enough to generate random numbers from a uniform distribution on $[0, 1]$. One can then use these random numbers to generate other random numbers from any other distribution. In R there are also in-built random numbers from different distributions. To read more on different algorithms to generate random numbers, try ".Random.seed".

In R you can get $Un(0, 1)$-distributed random variables using the command "runif(n)"; this gives a random vector of length n. One can use the"array" appropriately to obtain a matrix from this. To observe these random numbers one can also draw a histogram from them (more on this later). To make a vector of 100, $Un(0, 1)$ random numbers, and draw a histogram try the following

```
stoch<-runif(100);
hist(stoch);
```

Type "hist" and draw a histogram of different widths. Try using the command "summary" on the sequence of random numbers:

```
summary(stoch)
sd(stoch)
```

3.1 Generating random numbers from discrete distributions.

Let Y be a discrete random variable taking values in $\{0, \ldots, n\}$ with known density function $f_Y(k), k = 0, \ldots, n$ for some n. We want to generate random numbers from this distribution by using the random number generator $Un(0, 1)$. One can try :

1. Divide the interval $[0, 1]$ into $n + 1$ smaller disjoint intervals I_k. The interval I_k should be of length $f_Y(k)$.

2. Choose a random number x from $Un(0,1)$.
3. Look for the intervals $I_0, \ldots, I_n$ in which x belongs. Let this interval be I_l.
4. Set $y = l$.

Therefore, y is a random number with density p_Y, as

$$\begin{aligned} P(Y=k) &= P(X \in I_k) \\ &= \int_{I_k} f_X(x)dx \\ &= \int_{I_k} dx \\ &= \text{length of } I_k \\ &= f_Y(k), \end{aligned}$$

for $k = 0, \ldots, r$.

Example 1 *(Coin toss model) Suppose we want to generate a random variable to model the outcomes of an biased coin. Then*

$$Y = \begin{cases} 0 \textit{ in case of a tail }, \\ 1 \textit{ in case of head }. \end{cases}$$

The fact that the coin is biased implies that $f_Y(0) \neq f_Y(1)$, for example $f_Y(0) = 0.4, f_Y(1) = 0.6$. Then one can try the following algorithm:

```
x<-runif(1)
y<-as.double(x>0.4)
```

To generate 100 *outcomes of coin toss try:*

```
x<-runif(100)
y<-as.double(x>0.4)
```

When we generate discrete random variables, it is less convenient to draw histograms, since we need not make a grouping of the data. Instead we can count the relative frequencies. For example, in Example 1 we can count the number of times we get 1 and divide it by 100 and do the same for 0. This gives the relative frequencies, and then we can plot this data. Try

```
y1<-sum(y)/100
y0<-1-y1
plot(c(0,1),c(y0,y1),type="p",xlim=c(0,1),ylim=c(0,1))
```

Why are the heights of these bars not exactly 0.4 or 0.6? Repeat the same task with 1000 observations of the coin toss. Do you see any difference, what is it?

Example 2 *(Rolling of a fair die). We will generate random numbers distributed uniformly on $\{1, 2, \ldots, 6\}$, i.e., observations of a discrete random variable X with density function $f(k) = 1/6$ for $k = 1, \ldots, 6$.*

```
x<-runif(1)
y<-as.double(x<=c(1/6,2/6,3/6,4/6,5/6,1))*(1:6)
x<-min(y[y>0])
```

3.2 Random numbers from continuous distribution

Let Y be a random variable with a known distribution function $F = F_Y$. We want to generate random numbers from this distribution, even though we have only random numbers from $Un(0,1)$. Let $X \in Un(0,1)$. Let $Y = F^{-1}(X)$. Then Y has the desired distribution function! This is because

$$\begin{aligned} F_Y(y) &= P(Y \leq y) \\ &= P(F^{-1}(X) \leq y) \\ &= P(X \leq F(y)) \\ &= F(y), \end{aligned}$$

where the last step follows from $P(X \leq u) = F_X(u) = u$ for a random variable X with uniform distribution on $[0,1]$.

1. Generate 100 random numbers having distribution $F(x) = 1 - e^{-x}, x \geq 0$. Plot a histogram from the observed data. Do they appear to be distributed according to an exponential distribution?

In mathematical statistics a very important distribution is the Gaussian distribution. The method mentioned described above cannot be employed to generate random numbers from this distribution, since there is no explicit expression for either F or F^{-1} in this case. But it is possible to use other methods. In R there is an in-built function "rnorm". Go to help and use this to generate random numbers from Gaussian distribution.

1. Generate 200 random numbers from $N(1,2)$ distribution. Plot a histogram from this. Do the data look like it has Normal distribution?

2. Generate 100 random numbers from $N(0,1)$, $N(0,2)$ and $N(0,3)$ distributions, and plot a histogram. Do you observe any difference?

4 The law of large numbers

The law of large numbers is arguably the most important result in mathematical statistics. One version of it says that if $X_1, X_2, \ldots$ is a sequence of independent identically distributed r.v.'s with expectation m, then for every interval $(m - \epsilon, m + \epsilon)$, the r.v. $\bar{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$ lie in this interval with high probability, if n is large enough. This part of the lab consists of illustrating this.

1. Generate 10000 $Un(0,1)$ distributed r.v.'s and plot the sequence of averages $\bar{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$ against n.

```
n<-10000
x<-runif(n)
x_n<-cumsum(x)/(1:n)
plot(1:n,x_n)
```

   Q1 What does the graph seem to converge to? Calculate (analytically) $E(X_1)$.

   Q2 How does this relate to the statement of the LLN?

2. Generate 10000 $N(0,1)$-distributed r.v.'s and form $Z_n = n^{-1}\sum_{i=1}^n X_i^2$, and plot Z_n versus n.

```
n<-10000
x<-rnorm(n,0,1)
z_n<-cumsum(x^2)/(1:n)
plot((1:n),z_n)
```

   Q1 What does the graph seem to converge to? Calculate $E(X_1^2)$.

Q2 How does this relate to the statement of the LLN.

Q3 Explain how this can be used to give approximations for $E(g(X))$ for functions g (see also part 4 below.) For which functions g can this be done (give a restriction on g that ensures that this can be done.)

3. Simulate 10000 coin tosses, with

$$X_i = \begin{cases} 0 \text{ if tail up in toss number } i, \\ 1 \text{ if head up in toss number } i, \end{cases}$$

for $i = 1, \ldots, 10000$. The model is that $P(\text{head}) = 0.39$ och $P(\text{tail}) = 0.61$. Form $X_n = n^{-1} \sum_{i=1}^{n} X_i$ and plot versus n.

```
n<-10000
x<-runif(n)
x_i<-as.double(x<0.39)
z_n<-cumsum(x_i)/(1:n)
plot((1:n),z_n)
```

Q1 What does the graph converge to? Calculate (analytically) $E(X_1)$.

Q2 How does the result relate to what the LLN says?

Q3 Explain how this can be used to get approximations to $P(A)$ for arbitrary events A.

4. Let us study the integral $I = \int_0^1 e^{-x^2}\, dx$. There is no closed form expression for I, since we lack an elementary expression for the anti-derivative of e^{-x^2}. We will do a so called Monte-Carlo approximation of I. We can write $I = E(g(X))$, with $X \in Un(0,1)$ and $g(x) = e^{-x^2}$. Run the following commands.

```
n<-10000
x<-runif(n)
y<-exp(-x^2)
z_n<-cumsum(x)/(1:n)
plot((1:n),z_n)
```

Q1 What numerical value does the graph seem to converge to?

Q2 Use the LLN to motivate the approximation to I.

5 The central limit theorem

The central limit theorem says that if $X_1, X_2, \ldots$ is a sequence of independent equally distributed r.v.'s then their sum $S_n = \sum_{i=1}^{n} X_i$ and their average $\bar{X}_n = n^{-1} S_n$ are approximately distributed as a Gaussian (or normal) r.v. if n is large. This second part of the lab consist of illustrating this.

In order to do so we will use an estimate F_n of the d.f. F of X (which you will study in great detail in the inference theory part of the course) called the empirical distribution function, defined as

$$F_n(x) \quad = \quad \frac{1}{n} \sum_{i=1}^{n} 1\{X_i \leq x\}.$$

Q1 Derive an analytic expression for $E(F_n(x))$?

Q2 Use the LLN to motivate why F_n can be used as approximation of F.

Q3 For a fixed x_0: In what sense does $F_n(x_0)$ converge to $F(x_0)$ when $n \to \infty$?

Do help on “qqnorm” and on "apply" before you continue.

1. Generate $n = 100$ exponentially distributed r.v.'s with expesctation 1 and form their average $\bar{X}_n$. Do this $m = 200$ times so that you get 200 averages $\bar{X}_n^1, \ldots, \bar{X}_n^{200}$. Do an normal probability plot of these.

```
n<-100
m<-200
x<-rexp(n*m)
x2<-matrix(x,m,n)
x_n<-apply(x2,1,mean)
qqnorm(x_n)
qqline(x_n)
```

   Q1 Do the above with values $n = 10, 20, 200, 1000$. What is the outcome?

   Q2 Do the above with varying m. What is the outcome?

   Q3 Explain how your results relate to the CLT. Explain what happens qualitatively as $n \to \infty$ and as $m \to \infty$.

2. Do the above with $Un(0, 1)$-distributed r.v.'s and varying $n = 5, 10, 50$.

```
n<-100
m<-200
x<-runif(n*m)
x2<-matrix(x,m,n)
x_n<-apply(x2,1,mean)
qqnorm(x_n)
qqline(x_n)
```

   Answer Q1, Q2 and Q3 as above for this case.

   Q4 What, if any, is the difference with the previous exercise?

3. Generate $n = 100$ $Bin(nn, pp)$-distributed r.v.'s with $nn = 3$ and $pp = 0.5$ and their average $\bar{X}_n$. Do this $m = 200$ times. Do a normal probability plot on these.

```
n<-100
m<-200
nn<-3
pp<-0.5
x<-rbinom(n*m,nn,pp)
x2<-matrix(x,m,n)
x_n<-apply(x2,1,mean)
qqnorm(x_n)
qqline(x_n)
```

   Q1 Vary $n = 10, 50, 200$. What is the result?

   Q2 What happens if you change to $nn = 100$, and vary $n = 10, 50, 200$?

   Q3 Can you use the CLT to explain the qualitative difference between the distributions $Bin(3, 0.5)$ and $Bin(100, 0.5)$?

4. Try to imagine what can happen if you change the binomial distribution in the previous exercise to $Bin(100, 0.1)$. Think first, and then do a run and answer Q1 and Q2 in the previous exercise.

(The End)