Random numbers, sampling and simulations: I

M P Gururajan, Hina A Gokhale and Dayadeep Monder

Indian Institute of Technology Bombay, Mumbai

In this session, we are going to learn some R commands for sampling and simulations. One of the good resources for what follows is *simpleR: using R for introductory statistics* by John Verzani which is available, among other places, at the following URL:

`https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf`.

1 Sampling

Let us first use the `sample` command. As you can see from the examples below, this command can be used to either generate results of the toss of a coin or the throw of a die. This example also shows how to write functions in R. Note that for coin toss, we represent the head and tail by +1 and -1, respectively.

```
Coin <- c(-1,1)
Toss <- function(n) sample(Coin,n,replace=TRUE)
Toss(5)

## [1] -1  1  1  1  1

Die <- 1:6
Throw <- function(n) sample(Die,n,replace=TRUE)
Throw(7)

## [1] 2 6 4 5 6 2 1
```

Questions

1. Write an R script for tossing a fair coin and an unfair coin which has a 0.55 probability for heads. Toss these two coins for 10, 100, 1000, and 10000 times. Compare the mean values in these two cases for these different number of tosses and comment on your result.

   Hint: The coin can be made unfair by attaching different weights for the probability of heads or tails as shown below.

   ```
   Coin <- c(-1,1)
   p <- c(0.55,0.45)
   Toss <- function(n) sample(Coin,n,replace=TRUE,prob=p)
   ```

2. Let us say that there are 100 students in a statistics class. From these students, we want to randomly choose 10 students for a given task. Obviously, the same student can not be chosen more than once. Write a R script to do the choosing exercise.

   Hint: Use the help command for sample to identify how to simulate sampling without replacement.

2 Sampling from different distributions

The following table consists of some of the special random variables the and the R commands for generating the density, distribution function, quantile function and the random deviate. This is to serve as a ready-reckoner and the help command can be used to generate more information – such as the inputs to these function calls and the parameters. Note that some of these commands have already been used in the scripts for demonstrating central limit theorem.

Random Variable	**Density**	**Distribution function**	**Quantile function**	**Random Variate**
Binomial	dbinom	pbinom	qbinom	rbinom
Poisson	dpois	ppois	qpois	rpois
Hypergeometric	dhyper	phyper	qhyper	rhyper
Uniform	dunif	punif	qunif	runif
Normal	dnorm	pnorm	qnorm	rnorm
Exponential	dexp	pexp	qexp	rexp

In order to better understand the density, distribution function, generation of random deviates and the quantile functions, let us consider some of the solved problems and the plots and figures from the special random variables chapter of the textbook (Ross) and solve the problems / generate the figures using R.

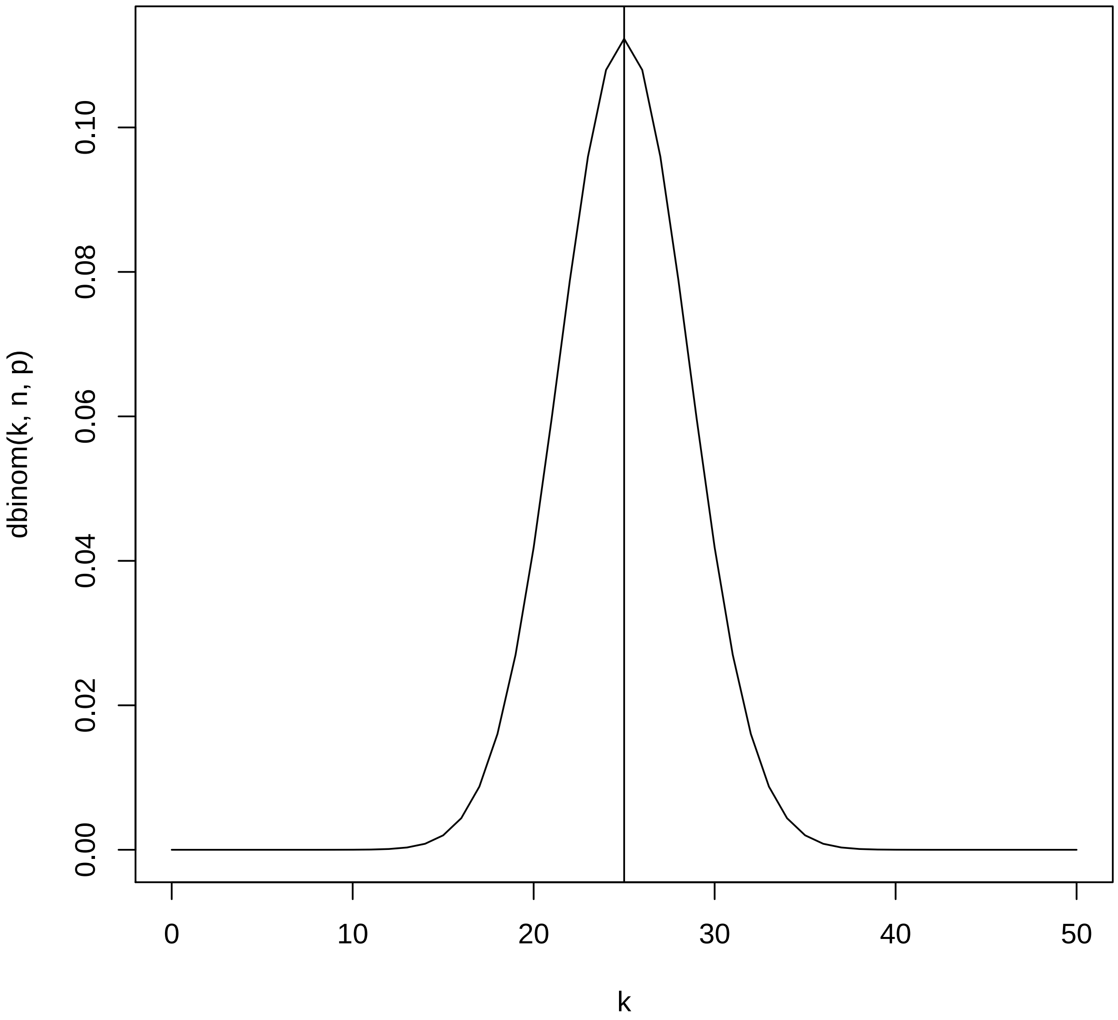
1. The colour of one's eyes is determined by a single pair of genes, with the gene for brown eyes being dominant over the one for blue eyes. This means that an individual having two blue-eyed genes will have blue eyes, while one having either two brown-eyed genes or one brown-eyes and one blue-eyed genes will have brown eyes. A baby inherits one randomly chosen gene from each of its parents' gene pair. If the eldest child of a pair of brown-eyed parents has blue eyes, what is the probability that exactly two of the other four children (none of whom is a twin) of this couple also have blue eyes.

   The eldest child has blue eyes while the parents are brown-eyed. This implies that both the parents have one blue-eyed gene. Hence, the probability that the child gets these blue eyed genes from the parents is $p = 0.5 * 0.5 = 0.25$. Given this p, we are looking for 2 blue eyed out of 4. Hence, we can use the `dbinom` command to get the answer:

```
p = 0.25
dbinom(2,4,0.25)

## [1] 0.2109375
```

2. Consider the following script:

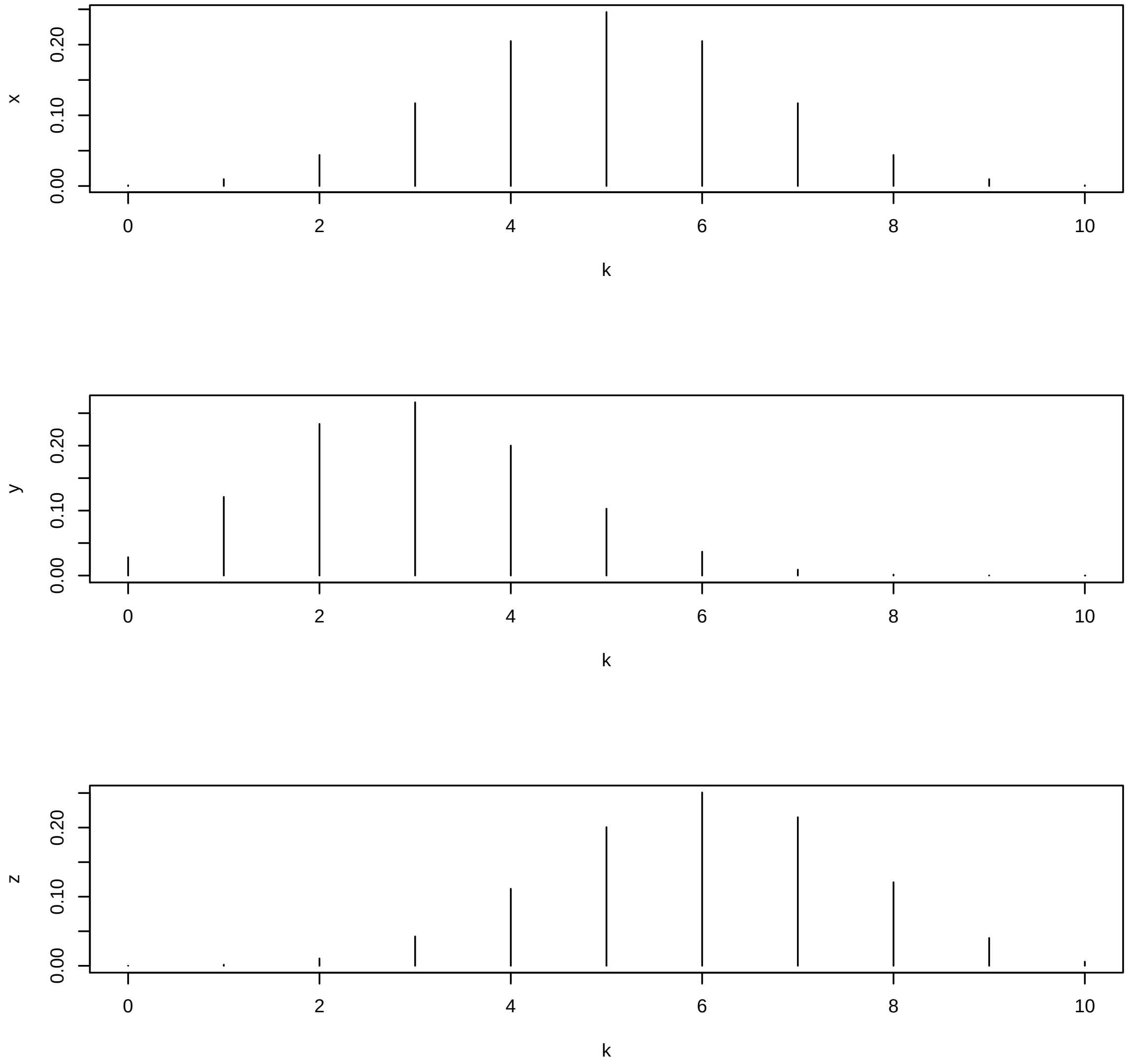
```
n <- 50
k <- seq(0,n)
p = 0.5
plot(k,dbinom(k,n,p),type="l")
abline(v=25)
```



How does the plot look? What happens if you use p value of 0.3 or 0.6?

3. Let us consider the probability mass functions shown in Figure 5.1 of Ross. Can you generate the same figure using an R script?

```
n <- 10
k <- seq(0,n)
p = 0.5
x <- dbinom(k,n,p)
p = 0.3
y <- dbinom(k,n,p)
p = 0.6
z <- dbinom(k,n,p)
par(mfrow=c(3,1))
plot(k,x,type="h")
plot(k,y,type="h")
plot(k,z,type="h")
```



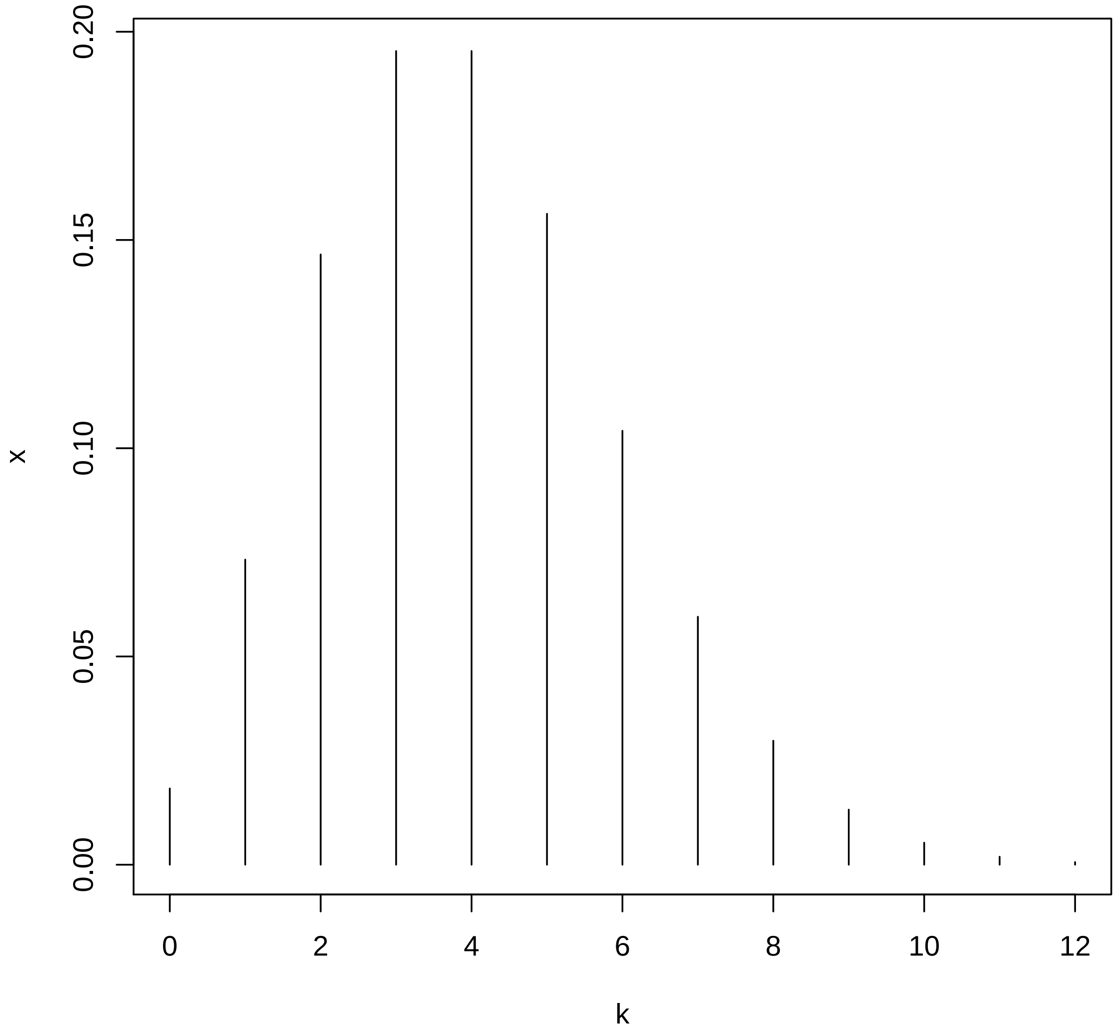
4. If X is a binomial random variable with parameters $n = 100$ and $p = 0.75$, find $P\{X = 70\}$ and $P\{X \leq 70\}$. Compare your results with the results given in Ross: Example 5.1f.

```
n <- 100
p = 0.75
dbinom(70,n,p)

## [1] 0.04575381

pbinom(70,n,p)

## [1] 0.149541
```

Note the use of `pbinom` to obtain the cumulative probability.

5. Plot the Poisson probability mass function with $\lambda = 4$ (in the range 0 to 12). Compare it with Figure 5.3 of Ross.

```
lambda <- 4
n = 12
k <- seq(0,n)
x <- dpois(k,lambda)
par(mfrow=c(1,1))
plot(k,x,type="h")
```



6. Suppose the probability that an item produced by a certain machine will be defective is 0.1. Find the probability that a sample of 10 items will contain at most one defective item. Assume that the quality of successive items is independent. Use Poisson approximation and compare the result with that obtained using binomial.

   We know that in Poisson, $\lambda = np$ where n is the number of independent trials, in each of which, the probability of success is p.

```
p = 0.1
n = 10
pbinom(1,n,p)

## [1] 0.7360989

dpois(0,n*p) + dpois(1,n*p)

## [1] 0.7357589
```

7. If the average number of claims handled daily by an insurance company is 5, what proportion of days have less than 3 claims? What is the probability that there will be 4 claims in exactly 3 of the next 5 days? Assume that the number of claims on different days independent.

```
lambda = 5
dpois(0,lambda)+dpois(1,lambda)+dpois(2,lambda)

## [1] 0.124652

ppois(2,lambda)

## [1] 0.124652

p = dpois(4,lambda)
dbinom(3,5,p)

## [1] 0.03672864
```

   Note that there are two ways of calculating the answer for the first question and both are shown in the script and the answer in both cases is ≈ 0.1247; the answer for the second question is ≈ 0.0367.

8. Consider one of the problems from the previous sub-section – of choosing 10 students from a list of 100 – without repeating any of the students. By choosing random variates from a uniform distribution, one can generate the random list of 10 students:

```
list <- as.integer(runif(10,min=1,max=100))
list

##  [1] 39 85 34 20 45 90 86  6 94 53
```

   Note that the above script sometimes pick the same number more than once and it is not possible to avoid. On the other hand, the sample command has an option not to repeat.

9. Suppose that a number of miles that a car can run before its battery wears out is exponentially distributed with an average value of 10000 miles. Is a person desires to take a 5000 mile trip, what is the probability that she will be able to complete her trip without having to replace her car battery?

```
1 - pexp(5000,1/10000)

## [1] 0.6065307
```

   In the second part of this tutorial, we will learn about some more distributions as well as about simulations and bootstrapping. In the meanwhile, solve as many of the Ross problems as possible by writing R scripts. That will come handy when you do your next R session with the TAs.