

MAS2602: Computing for Statistics

Newcastle University
`lee.fawcett@ncl.ac.uk`

Semester 1, 2018/19

MAS2602 Arrangements

- Statistics classes run in teaching weeks 5–9
- Lecturer is Dr. Lee Fawcett (lee.fawcett@ncl.ac.uk)
- Schedule: 6 lectures, 3 practicals, 1 class test
- Office hours: Mondays 2-3 and Wednesdays 12-1, room 2.07 Herschel

Schedule

Week 5	Tues 30 Oct	15.00	Intro Lecture (Herschel, LT2)
Week 6	Mon 5 Nov	14.00	Lecture (Herschel, LT2)
	Tues 6 Nov	15.00	Lecture (Herschel, LT2)
	Thurs 8 Nov	16:00-18.00	Practical (Herschel cluster)
Week 7	Tues 13 Nov	15.00	Lecture (Herschel, LT2)
	Thurs 15 Nov	16.00-18.00	Practical (Herschel cluster)
Week 8	Tues 20 Nov	15.00	Lecture (Herschel, LT2)
	Thurs 22 Nov	16:00-18:00	Practical (Herschel cluster)
	Fri 23 Nov	16.00	Assignment due
Week 9	Mon 26 Nov	14.00	Lecture (Herschel, LT2)
	Thurs 29 Nov	16:00	Class test (Herschel cluster)

Assignment (a.k.a. “mini-project”)

- Due **Friday 23 November, 4pm**
- Worth 10% of the module marks

Class test

- **4pm, 29 November** – one hour long
- Worth 30% of module marks
- Open-book test
- You will write one or two short R programs during the test

Help and Support

Help available:

- Office hours
- Demonstrators in practical sessions
- Books – see recommendations in booklet
- Email Lee, or just pop in!
- Blackboard and dedicated webpage

Late work policy

In this module, deadline extensions can be requested for the final project (by means of submitting a PEC form), and work submitted within 7 days of the deadline without good reason will be marked for reduced credit. This module also contains tests worth more than 10% for which rescheduling can be requested (by means of submitting a PEC form). There are mini-projects (worth 10% each) for which it is not possible to extend deadlines and for which no late work can be accepted. For details of the policy (including procedures in the event of illness etc.) please look at the School web site:

<http://www.ncl.ac.uk/maths/students/teaching/homework/>

For problems with deadlines, speak to your personal tutor and prepare a PEC form

Lecture 1: Introduction and Simulation of Random Variables

Introduction

In this part of the module we will do statistics with R:

- R is the foremost tool in modern computational statistics
- Using R teaches general concepts in programming
- It can be used to illustrate mathematical ideas in probability and statistics

Today's lecture: simulating random variables

- 1 Simulating random variables seen in MAS1604
- 2 Using this to simulate more complicated probability models

The binomial distribution

If $X \sim \text{Bin}(n, p)$ then X has PMF (probability mass function) given by

$$p_X(k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad \text{for } k = 0, 1, \dots, n.$$

There is no closed formula for the CDF (cumulative distribution function).

R commands:

- `dbinom` – calculate PMF
- `pbinom` – calculate CDF
- `rbinom` – generate random sample

The binomial distribution

If $X \sim \text{Bin}(n, p)$ then X has PMF (probability mass function) given by

$$p_X(k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad \text{for } k = 0, 1, \dots, n.$$

There is no closed formula for the CDF (cumulative distribution function).

R commands:

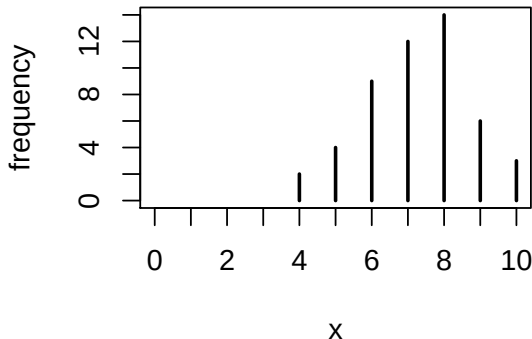
- `dbinom` – calculate PMF
- `pbinom` – calculate CDF
- `rbinom` – generate random sample

R commands for the binomial distribution

```
1 dbinom(5, 10, 0.7)      #  $\Pr(X = 5), X \sim \text{Bin}(10, 0.7)$   
2 pbinom(4, 10, 0.7)      #  $\Pr(X \leq 4), X \sim \text{Bin}(10, 0.7)$   
3 rbinom(50, 10, 0.7)     # Sample a  $\text{Bin}(10, 0.7)$  distribution 50 times
```

Creating a bar plot

```
1 x = rbinom(50, 10, 0.7)
2 plot(table(x), xlim = c(0,10), xaxt = 'n', xlab = 'x',
3      ylab = 'frequency')
4 axis(1, at = seq(0, 10))
```



The geometric distribution

If $Y \sim \text{Geom}(p)$ then Y has PMF and CDF given by

$$p_Y(k) = (1 - p)^{k-1}p,$$

$$F_Y(k) = 1 - (1 - p)^k, \quad \text{for } k = 1, 2, \dots$$

Note that Y takes values in $1, 2, 3, \dots$:

- R uses a slightly different definition
- We use the definition that Y is the number of Bernoulli trials with up to and including first success
- R counts number of trials up to, but **not** including the first success, so in R geometric random variables take values $0, 1, 2, \dots$

R commands for the geometric distribution

Adjust the arguments to account for different definition:

```
1 dgeom(4, 0.2)           #  $\Pr(Y = 5), Y \sim \text{Geom}(0.2)$   
2 pgeom(2, 0.2)           #  $\Pr(Y \leq 3), Y \sim \text{Geom}(0.2)$   
3 1 + rgeom(100, 0.2)     # Sample a  $\text{Geom}(0.2)$  distribution 100 times
```

Here's a function to replace `dgeom` with our definition of the geometric distribution:

```
1 mydgeom = function(x, p) {  
2   dgeom(x-1, p)}  
}
```

The Poisson distribution

If $Z \sim Po(\lambda)$ then Z has PMF given by

$$p_Z(k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad \text{for } k = 0, 1, 2, \dots$$

There is no closed formula for the CDF (cumulative distribution function).

R commands:

- `dpois` – calculate PMF
- `ppois` – calculate CDF
- `rpois` – generate random sample

R commands for the Poisson distribution

```
1 dpois(5, 3.5)      #  $\Pr(Z = 5), Z \sim Po(3.5)$   
2 ppois(2, 3.5)      #  $\Pr(Z \leq 2), Z \sim Po(3.5)$   
3 rpois(100, 3.5)    # Sample a  $Po(3.5)$  distribution 100 times
```

Summary

Distribution	Binomial	Poisson	Geometric 
PMF	<code>dbinom(...)</code>	<code>dpois(...)</code>	<code>dgeom(...)</code>
CDF	<code>pbinom(...)</code>	<code>ppois(...)</code>	<code>pgeom(...)</code>
sample	<code>rbinom(...)</code>	<code>rpois(...)</code>	<code>rgeom(...)</code>

Continuous random variables

R has functions for the uniform, exponential and normal distributions:

Distribution	Uniform	Exponential	Normal
PDF	<code>dunif(...)</code>	<code>dexp(...)</code>	<code>dnorm(...)</code>
CDF	<code>punif(...)</code>	<code>pexp(...)</code>	<code>pnorm(...)</code>
quantile	<code>qunif(...)</code>	<code>qexp(...)</code>	<code>qnorm(...)</code>
sample	<code>runif(...)</code>	<code>rexp(...)</code>	<code>rnorm(...)</code>

Continuous random variables – R examples

```
1 dunif(3, 2, 5)      #  $f_X(3), X \sim U(2, 5)$   
2 pexp(1.5, 5)        #  $F_Y(1.5), Y \sim \text{Exp}(5)$   
3 rnorm(10, 0, 2)     # Sample a  $N(0, 2^2)$  distribution 10 times
```

For the standard uniform ($U(0, 1)$) and standard normal ($N(0, 1)$) distributions you don't need to provide the parameters $a = 0$, $b = 1$ and $\mu = 0$, $\sigma = 1$ respectively. For example:

```
1 runif(20)           # Samples a  $U(0, 1)$  distribution 20 times  
2 pnorm(1.96)         #  $\Pr(Z < 1.96), Z \sim N(0, 1)$   
3 [1] 0.9750021
```

Quantiles

The quantile functions `qunif`, `qexp`, `qnorm` solve equations like

$$F_X(\alpha) = p$$

for α given a probability p . For example:

```
1 qnorm(0.9750021)
2 [1] 1.96
```

Quantiles

Example

Suppose $X \sim N(5, 2^2)$. Give the R command to find α such that (i) $X \leq \alpha$ with 90% probability, and (ii) $X \geq \alpha$ with 95% probability

A more advanced model

- Number of arrivals per day at an IT help-desk is modelled using a Poisson distribution
- Mean of the Poisson distribution might vary from day-to-day
- Suppose the number of arrivals $X \sim Po(\Lambda)$
- Λ is itself a random variable, with $\Lambda \sim Exp(c)$ for a constant $c = 0.05$

What can we say about the distribution of X ?

- 1 What are the expectation and variance of X ?
- 2 What is $\Pr(X > 30)$?

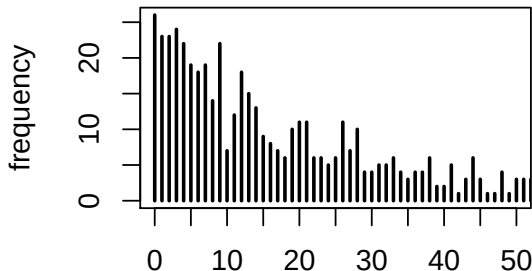
Arrival model – R code

```
1 arrivals = function(n, c = 0.05) {  
2   x = vector(mode = 'numeric', length = n)  
3   for (i in 1:n) {  
4     lambda = rexp(1, rate = c)  
5     x[i] = rpois(1, lambda)  
6   }  
7   return(x)  
8 }
```

Arrival model – R code

Bar plot:

```
1 x = arrivals(500)
2 plot(table(x), xlim = c(0, 50), xaxt = 'n', xlab = 'x'
  , ylab = 'frequency')
3 axis(1, at = seq(0, 50, 5))
```



Using simulated samples

- The **sample mean** is an approximation to the distribution mean
- The same applies for the sample variance
- To calculate $\Pr(X > 30)$ we count the proportion of times this occurs in the sample

We expect the approximation to improve **as we increase the sample size**

Using simulated samples

```
1 mean(x)
2 [1] 19.618
3 var(x)
4 [1] 382.1764
5 sum(x>30)/500
6 [1] 0.202
```

Recap Quiz

Write R code to:

- 1 simulate 10 observations on X_1 , where $X_1 \sim \text{Bin}(20, 0.05)$;
- 2 find $\Pr(X_2 > 3)$, where $X_2 \sim \text{Po}(2)$;
- 3 find $\Pr(X_3 = 4)$, where $X_3 \sim \text{Geom}(0.4)$;
- 4 find the median and IQR of the standard Normal distribution.

Mixtures of normal distributions

Suppose

- $\mu_1, \mu_2, \sigma_1 > 0, \sigma_2 > 0$ are fixed constants,
- w_1, w_2 are positive constants with $w_1 + w_2 = 1$.

Consider the following function:

$$f(x) = w_1 f_1(x) + w_2 f_2(x)$$

where f_1 and f_2 are the density functions for $Z_1 \sim N(\mu_1, \sigma_1^2)$ and $Z_2 \sim N(\mu_2, \sigma_2^2)$ respectively.

Check that $f(x)$ represents a valid probability density function.

Mixtures of normal distributions

First note that $f(x) \geq 0$ everywhere. Also

$$\begin{aligned}\int_{-\infty}^{\infty} f(x) dx &= w_1 \int_{-\infty}^{\infty} f_1(x) dx + w_2 \int_{-\infty}^{\infty} f_2(x) dx \\ &= w_1 + w_2 = 1.\end{aligned}$$

Mixtures of normal distributions

First note that $f(x) \geq 0$ everywhere. Also

$$\begin{aligned}\int_{-\infty}^{\infty} f(x) dx &= w_1 \int_{-\infty}^{\infty} f_1(x) dx + w_2 \int_{-\infty}^{\infty} f_2(x) dx \\ &= w_1 + w_2 = 1.\end{aligned}$$

Mixtures of normal distributions

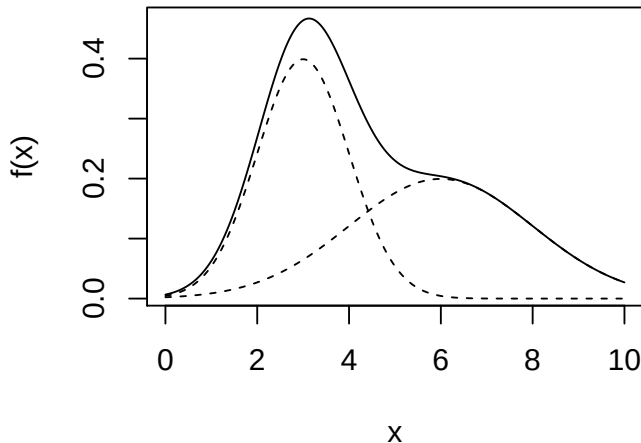
If a random variable X has PDF corresponding to $f(x)$ we say it is a **mixture** of normal distributions $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ with weights w_1, w_2 .

Note that X is **not** the sum of two normal random variables i.e.

$$X \neq w_1 Z_1 + w_2 Z_2 \quad \text{where } Z_i \sim N(\mu_i, \sigma_i^2), \quad i = 1, 2.$$

Example

For example: suppose $\mu_1 = 3, \sigma_1 = 1$ and $\mu_2 = 6, \sigma_2 = 2$ with $w_1 = w_2 = 1/2$.



Sampling from a mixture distribution

- 1 Sample a random variable $J \in \{1, 2\}$ such that $\Pr(J = 1) = w_1$ and $\Pr(J = 2) = w_2$.
- 2 The random variable J tells you which **component** of the mixture to sample X from.
- 3 If $J = 1$ then sample X from $N(\mu_1, \sigma_1^2)$, but if $J = 2$ then sample X from $N(\mu_2, \sigma_2^2)$.

Sampling from a mixture distribution

```
1  normal.mixture = function(n, mu1, sig1, w1, mu2, sig2,
2    w2) {
3    p = c(w1, w2)
4    x = vector(mode = 'numeric', length = n)
5    for (i in 1:n) {
6      j = sample(c(1, 2), 1, prob = p)
7      if (j==1) {
8        x[i] = rnorm(1, mu1, sig1)
9      }
10     else {
11       x[i] = rnorm(1, mu2, sig2)
12     }
13   }
14   return(x)
15 }
```