MAS2602: Computing for Statistics

Newcastle University
lee.fawcett@ncl.ac.uk

Semester 1, 2019/20

MAS2602 Arrangements

- Statistics classes run in teaching weeks 5–9
- Lecturer is Dr. Lee Fawcett (lee.fawcett@ncl.ac.uk)
- Schedule: 4 lectures, 4 practicals, 1 revision class, 1 class test
- Office hours: Tuesdays 3-4; room 2.07 Herschel
  (also MAS3902: Tuesdays 2-3 and Thursdays 3-4)

Schedule

Week 5	Mon 28 Oct	11–12	Lecture (Herschel LT1)
	Fri 1 Nov	10–12	Practical (Herschel PC)
Week 6	Mon 4 Nov	11–12	Lecture (Herschel LT1)
	Wed 6 Nov	11–1	Practical (Herschel PC)
Week 7	Mon 11 Nov	11–12	Lecture (Herschel LT1)
	Thu 14 Nov	11–1	Practical (Herschel PC)
Week 8	Mon 18 Nov	11–12	Lecture (Herschel LT1)
	Thu 21 Nov	11–1	Practical (Herschel PC)
	Thu 21 Nov	*3pm*	*Assignment due*
	Fri 22 Nov	1–2	Revision (Herschel LT2)
Week 9	*Tue 26 Nov*	*9–10*	*Test (Herschel PC)*

Assessment

Assignment (a.k.a. "mini-project")

- Due **Thursday 21 November, 3pm**
- Worth 10% of the module marks

Class test

- **Tuesday 26 November, 9am** – 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

Friday's practical:

1. Revision of R from MAS1802
2. Putting today's material into practice

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

Friday's practical:

1. Revision of R from MAS1802
2. Putting today's material into practice

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

Friday's practical:

1. Revision of R from MAS1802
2. Putting today's material into practice

The binomial distribution

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

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

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

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

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

R commands:

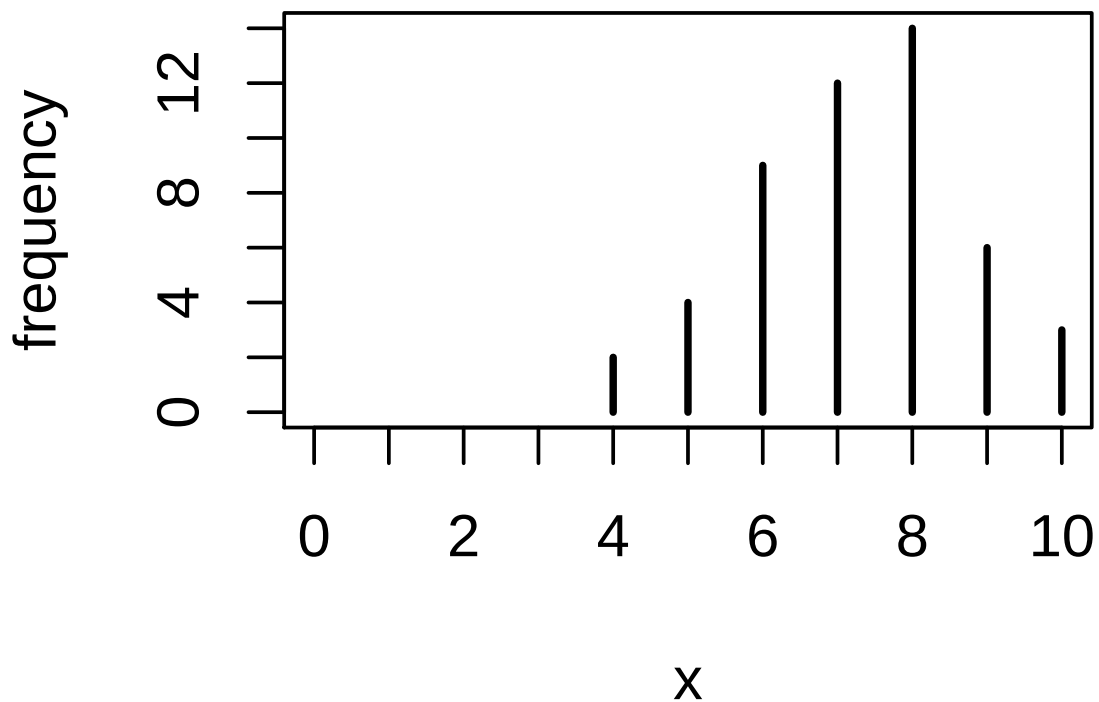
- `dbinom` – calculate PMF
- `pbinom` – calculate CDF
- `rbinom` – generate random sample

R commands for the binomial distribution

```
dbinom(5, 10, 0.7)      # Pr(X = 5), X ~ Bin(10, 0.7)
pbinom(4, 10, 0.7)      # Pr(X ≤ 4), X ~ Bin(10, 0.7)
rbinom(50, 10, 0.7)     # Sample a 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',
      ylab = 'frequency')
3 axis(1, at = seq(0, 10))
```



The geometric distribution

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

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

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

- 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 $0, 1, 2, \ldots$.

R commands for the geometric distribution

Adjust the arguments to account for different definition:

```
dgeom(4, 0.2)          # Pr(Y = 5), Y ~ Geom(0.2)
pgeom(2, 0.2)          # Pr(Y ≤ 3), Y ~ Geom(0.2)
1 + rgeom(100, 0.2)    # Sample a Geom(0.2) distribution 100 times
```

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

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

The Poisson distribution

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

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

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

```
dpois(5, 3.5)          # Pr(Z = 5), Z ~ Po(3.5)
ppois(2, 3.5)          # Pr(Z ≤ 2), Z ~ Po(3.5)
rpois(100, 3.5)        # Sample a Po(3.5) distribution 100 times
```

Summary

Distribution	Binomial	Poisson	Geometric 
PMF	`dbinom(...)`	`dpois(...)`	`dgeom(...)`
CDF	`pbinom(...)`	`ppois(...)`	`pgeom(...)`
sample	`rbinom(...)`	`rpois(...)`	`rgeom(...)`

Continuous random variables

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

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

Continuous random variables – R examples

```
dunif(3, 2, 5)      # f_X(3), X ~ U(2,5)
pexp(1.5, 5)        # F_Y(1.5), Y ~ Exp(5)
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:

```
runif(20)           # Samples a U(0,1) distribution 20 times
pnorm(1.96)         # Pr(Z < 1.96), Z ~ N(0,1)
[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

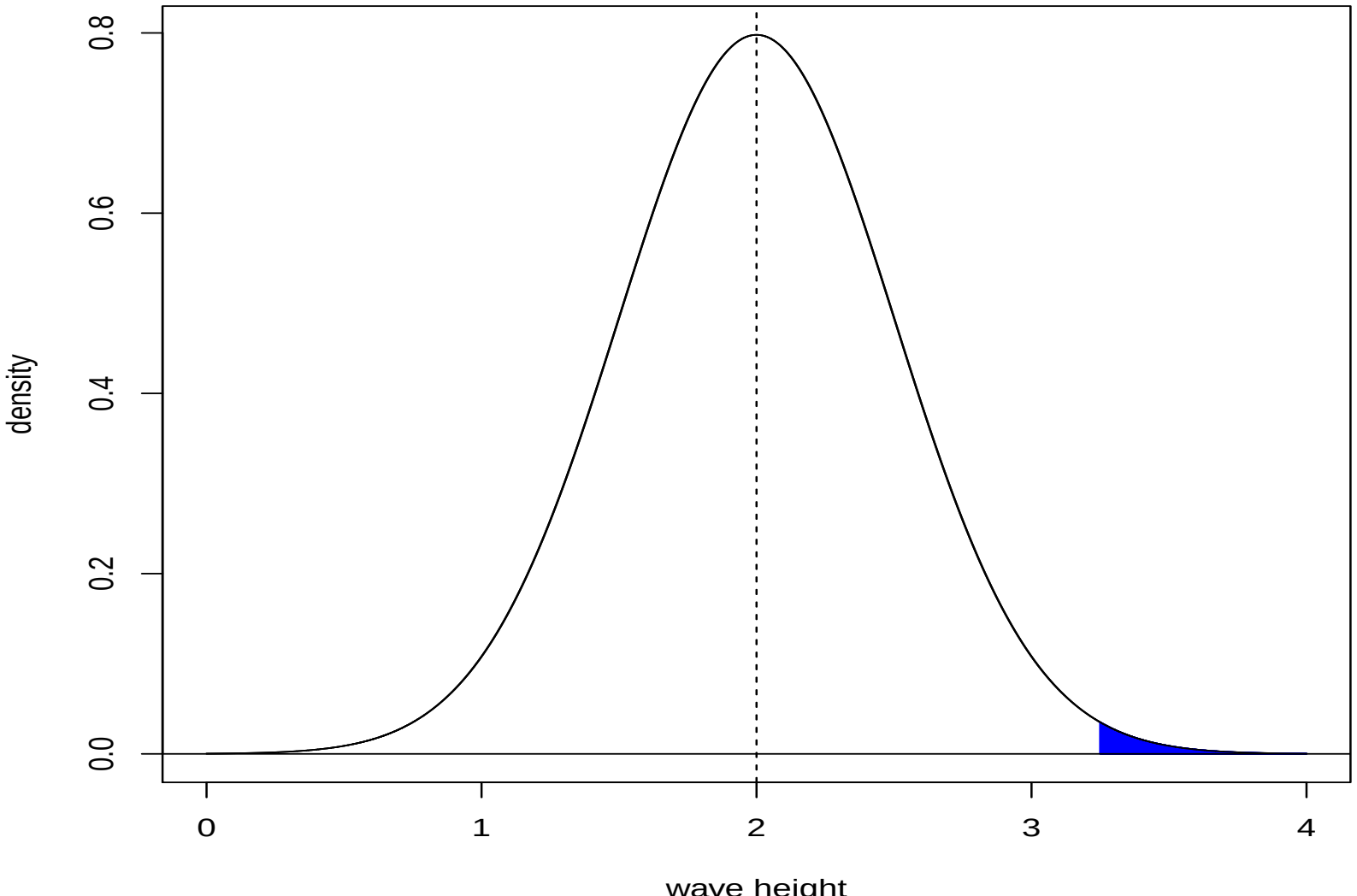
Example

Suppose annual maximum wave heights observed off the coast at a flood-prone town are assumed Normally distributed, with mean 2 metres and standard deviation 0.5 metres.

(a) Write down the R command to find the probability that the largest wave height next year will exceed 3.25 metres.

(b) Write down the R command to estimate the height of a new sea wall such that we might expect the town to be flooded, on average, once per century. Why might our modelling assumption be invalid?

Example 1.1: Solution (a)



Example 1.1: Solution (a)

Could work out the old-fashioned way:

$$\begin{aligned}
\Pr(X > 3.25) &= \Pr\left(Z > \frac{3.25 - 2}{0.5}\right) \\
&= \Pr(Z > 2.5) \\
&= 1 - \Pr(Z \leq 2.5) = 1 - 0.994 = 0.006.
\end{aligned}$$

Or could just use R:

```
1 1-pnorm(3.25, 2, 0.5)
2 [1] 0.006209665
```

Example 1.1: Solution (a)

Could work out the old-fashioned way:

$$\begin{aligned}
\Pr(X > 3.25) &= \Pr\left(Z > \frac{3.25 - 2}{0.5}\right) \\
&= \Pr(Z > 2.5) \\
&= 1 - \Pr(Z \leq 2.5) = 1 - 0.994 = 0.006.
\end{aligned}$$

Or could just use R:

```
1 1-pnorm(3.25, 2, 0.5)
2 [1] 0.006209665
```

Example 1.1: Solution (a)

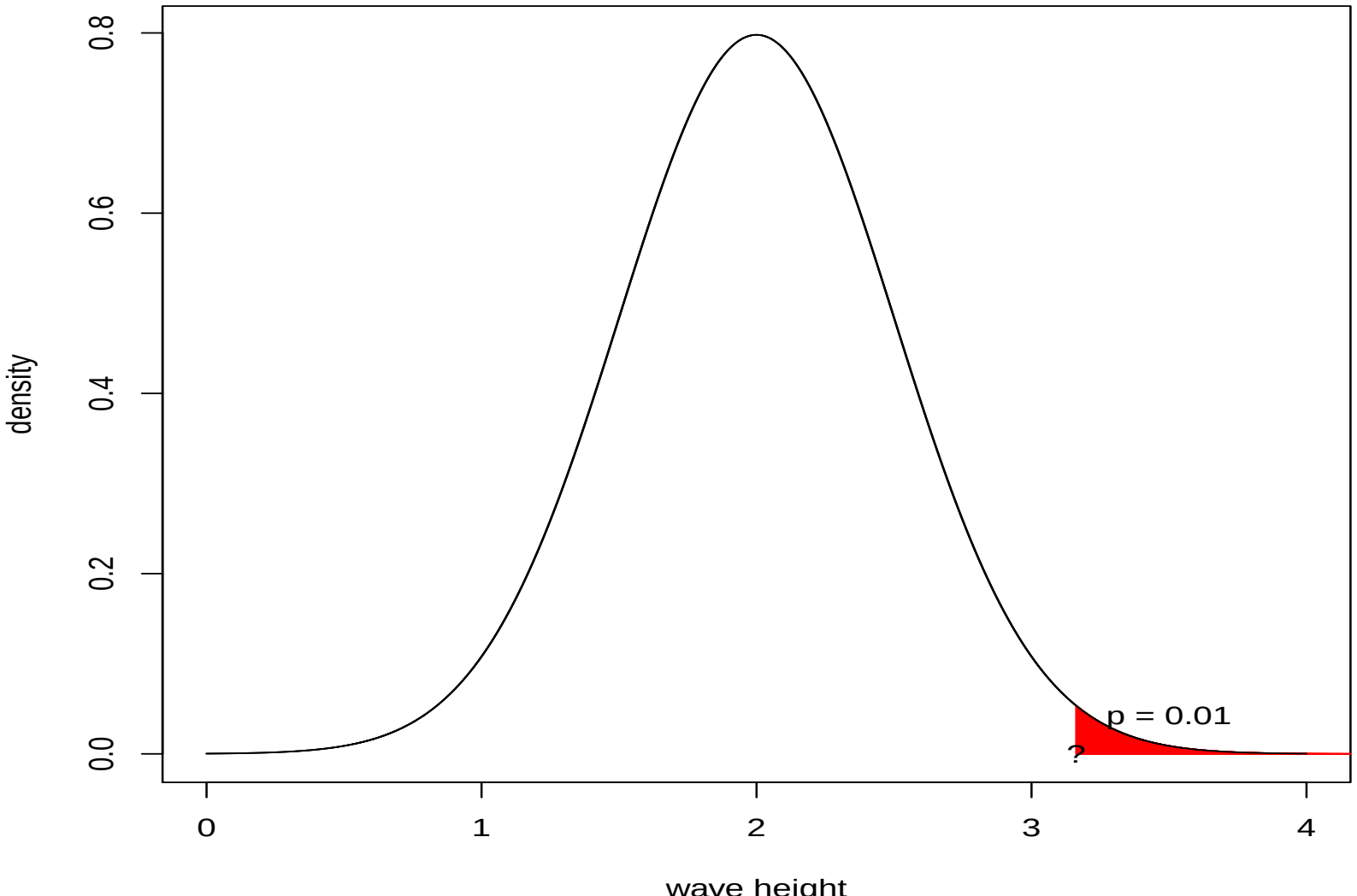
Could work out the old-fashioned way:

$$\begin{aligned}
\Pr(X > 3.25) &= \Pr\left(Z > \frac{3.25 - 2}{0.5}\right) \\
&= \Pr(Z > 2.5) \\
&= 1 - \Pr(Z \leq 2.5) = 1 - 0.994 = 0.006.
\end{aligned}$$

Or could just use R:

```
1 1-pnorm(3.25, 2, 0.5)
2 [1] 0.006209665
```

Example 1.1: Solution (b)



Example 1.1: Solution (b)

Similarly:

```
qnorm(0.99, 2, 0.5)
[1] 3.163174
```

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 $\Pr(X > 30)$?

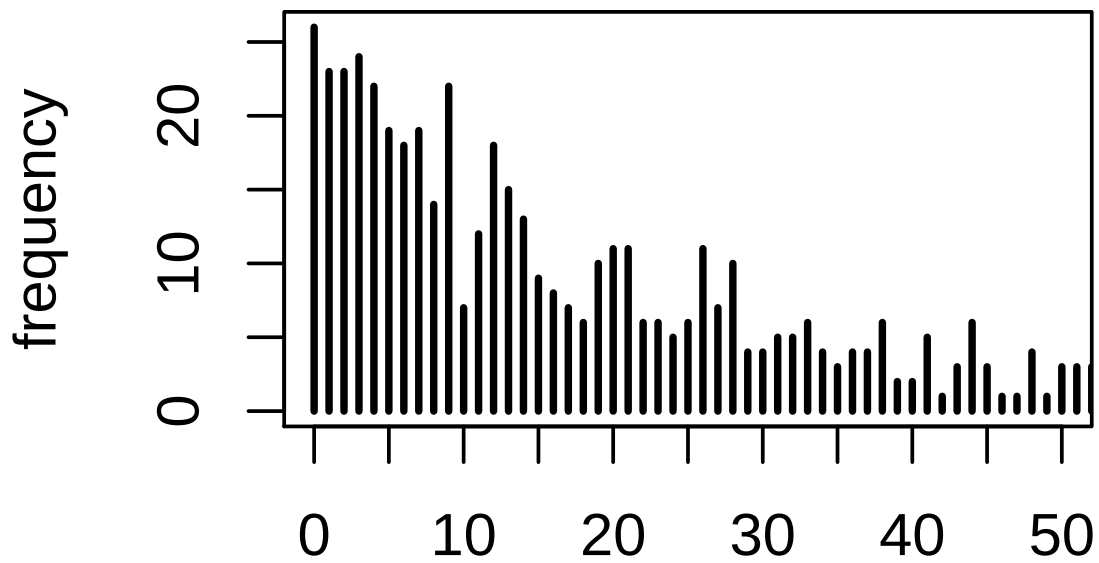
Arrival model – R code

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

Arrival model – R code

Bar plot:

```
x = arrivals(500)
plot(table(x), xlim = c(0, 50), xaxt = 'n', xlab = 'x'
    , ylab = 'frequency')
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

```
mean(x)
[1] 19.618
var(x)
[1] 382.1764
 sum(x>30)/500
[1] 0.202
```

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

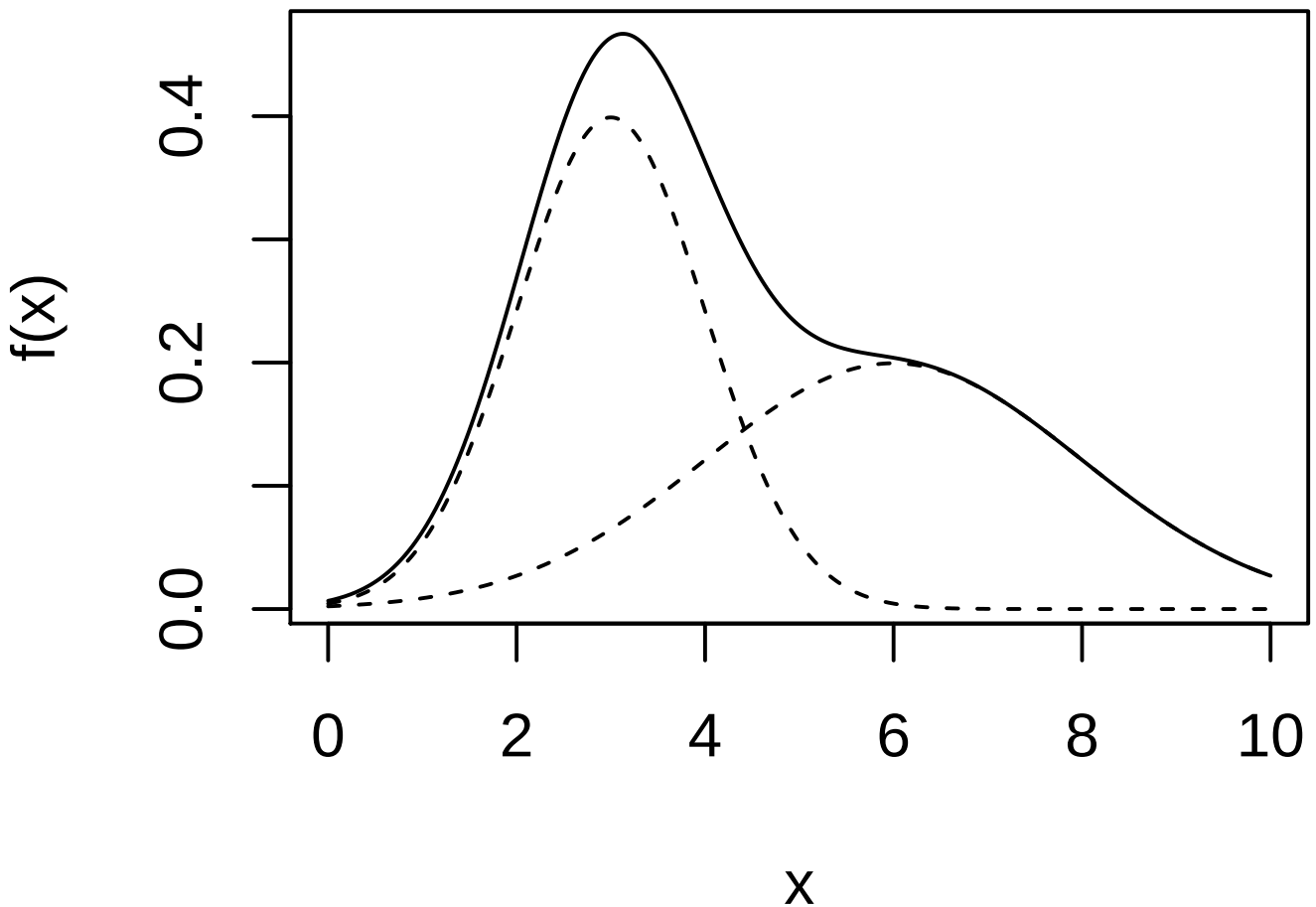
If a random variable X has PDF corresponding to $f(x)$ we say it is a mixture of normal distributions $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ with weights w_1, w_2.

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

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

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