MAS2602

Computing for Statistics

Lecturer: Dr. Lee Fawcett

Semester 1, 2018/19

Lecturer information

The lecturer is Lee Fawcett. If you have any questions about the module he can be contacted via email at `lee.fawcett@newcastle.ac.uk`. Lee is based in room 2.07 on the second floor of the Herschel Building.

Class 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)

Office hours are 2-3 on Mondays and 12-1 on Wednesdays in Herschel room 2.07.

Assessment

The statistics component accounts for 40% of module marks in total. The different assessment components are weighted as follows:

- Assignment – 10% of module marks. Due on **Friday 23 November, 4pm**.
- Class test – 30% of module marks.

The class test will be held in the Herschel cluster at **4pm on Thursday 29 November**. The test will last 1 hour, and will be open-book. You will need to complete an R programming task and submit your results via NESS.

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). 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/

Course Materials

All course materials will be posted on Blackboard and on a dedicated webpage:

www.mas.ncl.ac.uk/∼nlf8/teaching/mas2602

Books

The following books are in the Robinson Library, and are suitable for the material we cover.

- John Braun and Duncan James Murdoch: A First Course in Statistical Programming with R
- Sheldon Ross: A First Course in Probability
- N. A. Weiss, Paul T. Holmes and Hardy Michael: A Course in Probability

Introduction

In this module we will learn about computational tools for performing probability calculations and statistical analysis. We will use the R programming language. R is very widely used in universities, businesses and other organizations due to its many advantages over other similar software packages. It is available free of charge, it is 'open source' and can be used for a very wide range of statistical and scientific applications. Libraries of R functions and data-structures have been created and are readily available on the web. These enable statisticians to perform a very wide range of tasks but with a minimum of additional programming. Learning to use R is important because: (i) it is the foremost tool in modern computational statistics; (ii) it is a great way to learn general concepts in programming which can be used with other languages or in other contexts; and (iii) it can be used to illustrate mathematical ideas in probability and statistics.

The lectures will present a mixture of background theory in probability and statistics, together with elements of the R programming language. Each lecture will therefore contain both some mathematical theory and some R programming concepts. This mixture of material means that notes written only on the board are impractical for this module, and so this booklet of notes has been provided: you need to fill in the blank sections yourself. In addition to the lectures, there are three computer practicals during which the principles discussed in the lectures can be put into practice. The practicals are designed to help students complete the assignment and prepare for the class test. Code written in the practicals can subsequently be used in the class test, which is an open-book exam.

MAS1802 is a prerequisite for this module. You saw how to use R and RStudio in that module, and learned the basics of R programming: how to write functions, and how to use `if` statements and `for` loops. This material will not be covered again during the lectures for this module, but in MAS2602 we will see this material in lots of examples.

1 Simulating random variables

1.1 Discrete distributions: the binomial, geometric and Poisson distributions

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

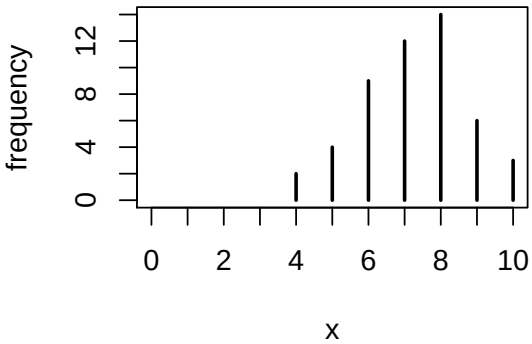
There is no closed formula for the CDF (cumulative distribution function; i.e. $\Pr(X \leq x)$).

The R commands `dbinom`, `pbinom`, and `rbinom` are used respectively to compute the PMF, compute the CDF, and simulate from a binomial distribution. Here are examples:

```
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
```

Here's an example of how to create a bar plot of a sample from a binomial distribution:

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



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

Note that Y takes values in $1, 2, 3, \ldots$. R uses a slightly different definition of the geometric distribution. We use the definition that Y is the number of Bernoulli trials with parameter p up to and including the first trial which is a success. The definition in R is the number of trials up to, but **not** including the first success, so in R geometric random variables take values $0, 1, 2, \ldots$. Like the binomial distribution, the R commands `dgeom`, `pgeom`, and `rgeom` are used respectively to compute the PDF, compute the CDF, and simulate from the geometric distribution. However, they need to be adjusted to match our 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:

There is no closed formula for the CDF.

Examples of R commands:

```
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(...)`

1.2 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(...)`

Examples of R commands:

```
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
```

The quantile functions `qunif, qexp, qnorm` solve equations of the form $F_X(\alpha) = p$, for α, given a probability p. For example:

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

Example 1.1: 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.

1.3 Some more advanced probability models

The main reason to use computers for mathematics and statistics is to solve problems which cannot be done by hand or which require very lengthy calculations. This section contains a simple motivating example to illustrate this.

The number of arrivals per day at a university IT help-desk can be modelled using a Poisson distribution. However, it is thought that the mean of the Poisson distribution might vary from day-to-day. For example, the desk is often busier nearer the end of term due to more students having deadlines. The following model is proposed. Suppose the number of arrivals $X|\Lambda \sim Po(\Lambda)$ where the rate 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?

The random variable X does not have a distribution which we can immediately recognize. In fact, much can be done analytically (as we will see in MAS2903 in semester 2), but we will use simulation to answer the following questions:

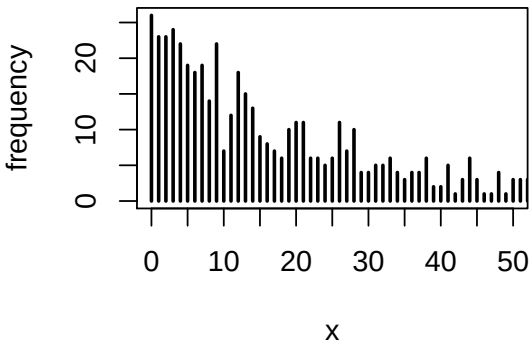
1. What are the expectation and variance of X?
2. What is $\Pr(X > 30)$?

Here is code to simulate a number n of samples from the distribution of X:

```
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)
}
```

Here's how to create a bar plot of the sample and compute the **sample** mean and variance:

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




```
mean(x)
[1] 19.618
var(x)
[1] 382.1764
```

The mean and variance of the **distribution** will be close to the sample mean and variance (see section 6).

To calculate $\Pr(X > 30)$ we count the proportion of times this occurs in the sample:

```
 sum(x>30)/500
[1] 0.202
```

1.4 Mixtures of normals

Suppose $\mu_1, \mu_2, \sigma_1 > 0, \sigma_2 > 0$ are fixed constants, and 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.

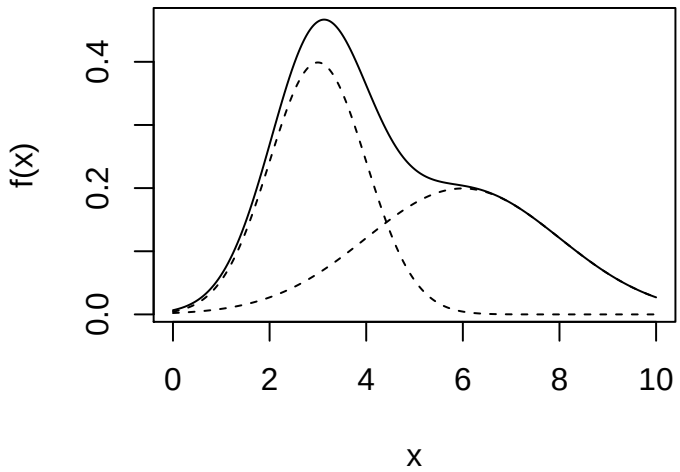
Check that $f(x)$ represents a valid probability density function:

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.

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

The right hand side is actually another normal distribution (as seen in MAS1604), but X **does not have** the PDF of a normal random variable.

For example: suppose $\mu_1 = 3, \sigma_1 = 1$ and $\mu_2 = 6, \sigma_2 = 2$ with $w_1 = w_2 = 1/2$. Here is a plot of $f(x)$:



Consider how to sample X from this distribution:

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

Here is an R function to perform this sampling n times:

```
normal.mixture = function(n, mu1, sig1, w1, mu2, sig2, w2) {
    p = c(w1, w2)
    x = vector(mode = 'numeric', length = n)
    for (i in 1:n) {
        j = sample(c(1, 2), 1, prob = p)
        if (j==1) {
            x[i] = rnorm(1, mu1, sig1)
        }
        else {
            x[i] = rnorm(1, mu2, sig2)
        }
    }
    return(x)
}
```

Write down any notes about this function.

Normal mixture distributions can contain more than two components – the generalization is very straightforward. They offer a very flexible way to model data with different shaped distributions.

2 Further simulation methods

In this section we look at two topics:

1. Use of the `sample` command in R to perform simulations that help solve combinatorial problems.
2. The inverse CDF method for simulating continuous random variables.

2.1 The `sample` command

The `sample` command samples elements repeatedly from a given set, and sampling can either be with or without replacement.

```
sample(x, n)                    # Create a random sample of n elements from x without replacement
sample(x, n, replace = TRUE)    # Create a random sample with replacement
sample(x, n, prob = p)          # Create a random sample from x using probabilities stored in p
```

Example 2.1: Suppose we randomly permute the letters of the word MISSISSIPPI. What is the expected number of times we have consecutive S's in the jumbled up word? What is the probability the word contains no consecutive S's? (This problem is relevant to the design of clinical trials.)

Solution is possible analytically, but simulation is easy:

```
mississippi = function(num_sims) {
    total_pairs = 0
    freq = 0
    letters = c("M","I","S","S","I","S","S","I","P","P","I")
    len = length(letters)
    for (i in 1:num_sims) {
        word = sample(letters, len)
        #Where are the S's?
        s = (word == "S")
        #Where are the pairs?
        pairs = s[2:11] & s[1:10]
        if (sum(pairs)>0) {
            freq = freq + 1;
        }
        total_pairs = total_pairs + sum(pairs)
    }
    cat("Expectation = ",total_pairs/num_sims, "\n")
    cat("Probability = ",freq/num_sims, "\n")
}
```

Make some notes about how this function works.

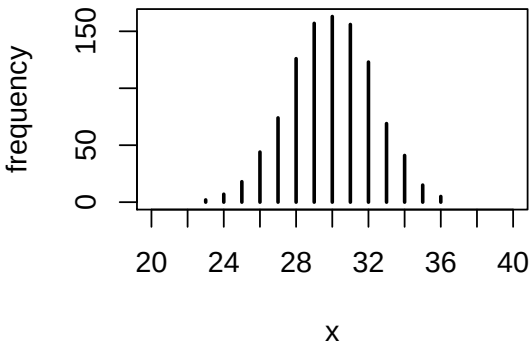
Example 2.2: The Ehrenfest model of diffusion. Suppose we have two urns, urn A initially containing n orange balls and urn B containing n grey balls. At every time-step, a ball is chosen uniformly at random from A and and a ball is chosen uniformly at random from B. The two balls are then swapped (i.e. the ball from A is put in B and the ball from B is put in A). This repeats for k time steps. Suppose $n = 50$ and let X be the number of grey balls in B after $k = 40$ steps. Use sampling to investigate the distribution of X.

```
#This function returns a single value of X
ehrenfest = function(n, k) {
    A = rep(0, n)
    B = rep(1, n)
    for (i in 1:k) {
        ballA = sample(1:n, 1)
        ballB = sample(1:n, 1)

        if ((A[ballA]==1)&&(B[ballB]==0)) {
            A[ballA] = 0
            B[ballB] = 1
        }
        else if ((A[ballA]==0)&&(B[ballB]==1)) {
            A[ballA] = 1
            B[ballB] = 0
        }
    }
    sum(B)
}
```

```
# Sample X many times to see the distribution
n = 50
k = 40
num_iterations = 1000
x = vector(mode = 'numeric', length = num_iterations)
for (j in 1:num_iterations) {
    x[j] = ehrenfest(n,k)
}
```

```
plot(table(x), xlim = c(20, 40), xaxt = 'n', xlab = 'x', ylab = 'frequency')
axis(1, at = seq(20, 40, 2))
```



Make some notes about how this function works.

2.2 Inverse CDF method of simulating univariate random variables

Let $U \sim U(0,1)$. Suppose $F(x)$ is a well-defined CDF which is invertible. Then the random variable $X = F^{-1}(U)$ has CDF $F(x)$.

Proof:

This result enables us to simulate the random variable X very readily provided we can simulate uniform random variables. In R we would make use of the command:

```
x = inv.cdf(runif(1))
```

Here, the function `inv.cdf` is a function that we have written ourselves to return the inverse of the CDF of a particular distribution of interest; see practical 1 for more details.

Example 2.3: Suppose we obtain two observations from a $U(0,1)$ distribution: 0.1 and 0.85. Use these values to obtain two observations from $X \sim Exp(2)$.

Example 2.4: Patients arriving at a doctor's surgery have appointment times 15 minutes apart. The random variable L measures how late a patient is for his/her appointment, with negative values denoting early arrives. The PDF of L is

$$f_L(l) = \begin{cases} 0.0025(l+20), & -20 \leq l \leq 0, \\ 0.05e^{-l/10}, & > 0. \end{cases}$$

(a) Sketch $f_L(l)$.

(b) Show that the CDF of L is

$$F_L(l) = \begin{cases} 0, & l < -20, \\ 0.00125(l+20)^2, & -20 \leq l \leq 0, \\ 1 - \frac{e^{-l/10}}{2}, & l > 0. \end{cases}$$

(c) Suppose that 0.205 and 0.713 are two realized values of $U \sim U(0,1)$. Show that the corresponding realized values of L are -7.19 and 5.55 under the inverse CDF simulation method.

3 Transformation of random variables

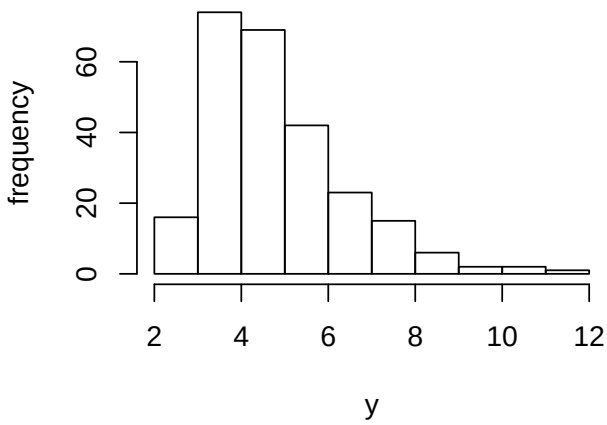
3.1 Transforming a normal random variable

The normal distribution is very widely used to model data. However, normal random variables take values on the entirety of $\mathbb{R}$ and they are symmetric around the mean. This is unrealistic in many situations – for example, if we want to model people's heights or weights. One way to obtain a positive random variable is to define

$$Y = e^X, \quad \text{where } X \sim N(\mu, \sigma^2).$$

Then Y is a random variable which is a **transformation** of X. What does the distribution of Y look like? We can use sampling to find out, at least approximately:

```
mu = 1.5
sigma = 0.3
n = 250
y = exp(rnorm(n, mean = mu, sd = sigma))
hist(y, xlab = 'y', ylab = 'frequency', main = '')
```



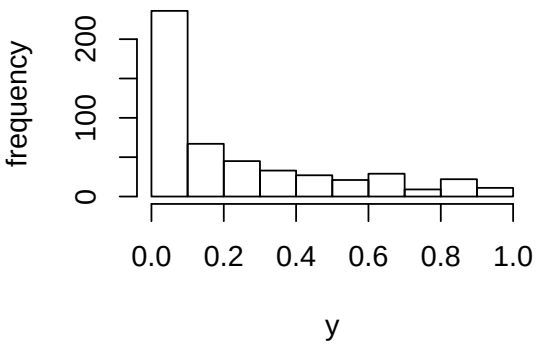
The random variable Y can only adopt positive values, and the distribution is positively skewed (long tail on the right). This kind of distribution is useful to model quantities like height – but what is the PDF of Y? A first guess is $f_Y(y) = f_X(\log(y))$ since log maps Y back to X, but this is **not** the correct answer.

3.2 Computing PDFs for transformed random variables

Example 3.1: If $X \sim U(0,1)$, what is the PDF of $Y = X^3$?

To confirm this result, we can simulate Y. Does the histogram look right?

```
y = (runif(500))^3
hist(y, xlab = 'y', ylab = 'frequency', main = '')
```



Notes:

Example 3.2: If $X \sim N(0,1)$, what is the PDF of $Y = X^2$?

Example 3.3: If $X \sim N(\mu, \sigma^2)$, what is the PDF of $Y = e^X$? This is the example used at the start of this section. We say Y has the **log-normal** distribution and write $Y \sim LN(\mu, \sigma^2)$.

3.3 The transformation method

These examples are all special cases of the following result.

If X has PDF $f_X(x)$ and $h(x)$ is a monotonically increasing or monotonically decreasing function then h is invertible and the PDF of $Y = h(X)$ is

$$f_Y(y) = f_X(h^{-1}(y)) \times \left| \frac{d\left(h^{-1}(y)\right)}{dy} \right|$$

when y is in the range of $h(x)$ and zero otherwise.

This result allows direct calculation of $f_Y(y)$ and is called the **univariate transformation method**. We will see this result again in semester 2, in MAS2903.

Proof:

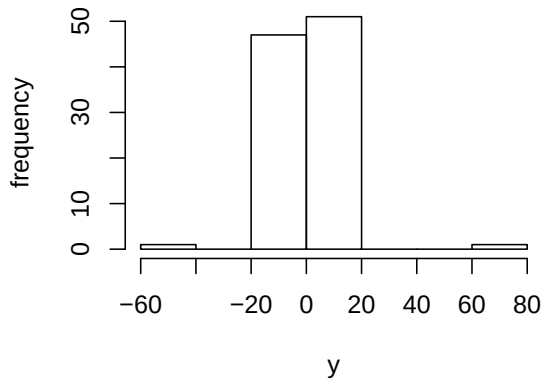
Example 3.4: When $X \sim U(0,1)$, use the transformation method to find the PDF of $Y = X^3$.

Example 3.5: When $X \sim U(-1,1)$ and $Y = e^X$, use the transformation method to find the PDF of Y.

Example 3.6: Use the transformation method to find the PDF of the log-normal distribution.

Example 3.7: Suppose $X \sim U(-\frac{\pi}{2}, \frac{\pi}{2})$ and $Y = \tan(X)$. Use simulation to study the distribution of Y and explicitly compute the PDF.

```
y = tan(runif(100, -pi/2, pi/2))
hist(y, xlab = 'y', ylab = 'frequency', main = '')
```



This distribution looks odd! Calculate the PDF:-

4 Bivariate distributions

In this section we consider two related topics: (i) how to simulate samples from continuous bivariate distributions, and (ii) transformations of bivariate random variables. This leads naturally to consideration of the bivariate normal distribution.

4.1 Sampling from bivariate distributions

Suppose we have a pair of continuous random variables (X, Y) with joint PDF $f_{X,Y}(x, y)$. Recall that

$$f_{X,Y}(x, y) = f_X(x) f_{Y|X}(y \mid X = x)$$

where $f_X(x)$ is the **marginal** PDF of X and $f_{Y|X}(y \mid X = x)$ is the **conditional** PDF of Y given $X = x$. One way to simulate realizations of the pair (X, Y) is:

1. Simulate a realization x of X from the marginal distribution of X.
2. Simulate a realization y of Y from the conditional distribution of Y given $X = x$.

In the **trivariate case** we can add a third step: simulate Z given $Y = y$ and $X = x$.

Example 4.1: Suppose $X \sim U(0.1, 0.5)$ and

$$f_{Y|X}(Y \mid X = x) = \begin{cases} xe^{-xy}, & \text{when } y \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Write an R function to generate a $2 \times n$ **matrix** of samples from this distribution, and produce a scatter plot.

Note that, given that $X = x$, Y has an exponential distribution with parameter $\lambda = X$:

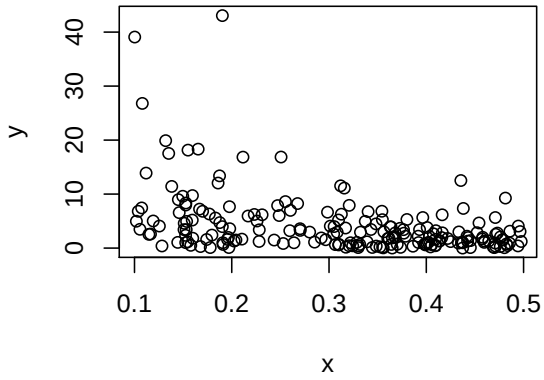
$$Y \mid X = x \sim Exp(x).$$

```
example4.1 = function(n) {
    output = matrix(0, 2, n)
    x = runif(n, 0.1, 0.5)
    y = rexp(n, rate = x)
    output[1, ] = x
    output[2, ] = y
    output
}
```

Make some notes about the R function.

To obtain a scatterplot:

```
test = example4.1(200)
plot(test[1,], test[2,], xlab = 'x', ylab = 'y')
```



Example 4.2: (Trivariate case.) Suppose that $X \sim U(-1,1)$ and $Y \sim Exp(\lambda)$ independently of X and that

$$f_{Z|X,Y}(Z \mid X = x, Y = y) = \frac{1}{\sqrt{2\pi}y} e^{-\frac{1}{2}\left(\frac{z-x}{y}\right)^2}.$$

Write an R function to sample realizations (X, Y, Z) in the case $\lambda = 5$ and produce a scatterplot of (X, Z). What would the plot look like if $\lambda = 1$?

Note that, given that $X = x$, $Y = y$ then Z has a normal distribution:

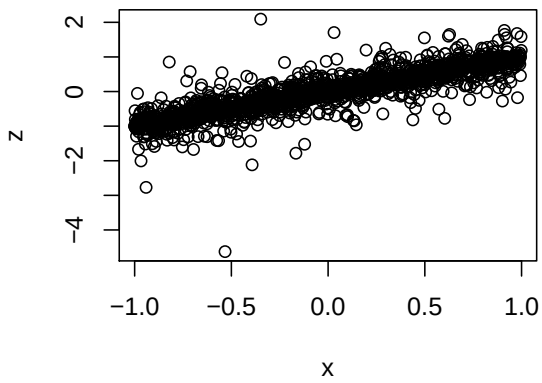
$$Z \mid X = x, Y = y \sim N(x, y^2).$$

```
example4.2 = function(n) {
   output = matrix(0, 3, n)
    x = runif(n, -1, 1)
    y = rexp(n, rate = 5)
    z = rnorm(n, mean = x, sd = y)
    output[1, ] = x
    output[2, ] = y
    output[3, ] = z
    output
}
```

Make some notes about the R function.

To obtain a scatterplot:

```
output = example4.2(2000)
plot(output[1,], output[3,], xlab = 'x', ylab = 'z')
```



4.2 Transformations of bivariate random variables

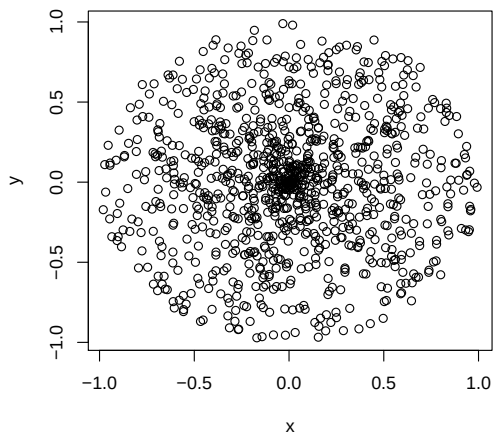
Motivating example. Consider distributions of points that lie on a disc of radius 1. Suppose (R, Θ) are the polar coordinates of a random point. Suppose that $R \sim U(0,1)$ and $\Theta \sim U(0, 2\pi)$. What does this distribution look like? Is it uniform over the disc?

Here is some R code to sample from this distribution and to create a scatterplot:

```
disc = function(n) {
    r = runif(n, 0, 1)
    theta = runif(n, 0,2*pi)
    x = r*sin(theta)
    y = r*cos(theta)
    data.frame(x, y)
}
```

To obtain a scatterplot:

```
plot(disc(1000))
```



This distribution is **not** uniform on the disc: there seem to be more points in the centre than near the edges. So what should the distribution of (R, Θ) be?

The following result relates the PDF of a pair of random variables (X, Y) with the PDF of a pair (U, V) where U and V are **functions** of X and Y.

Let X and Y have joint PDF $f_{X,Y}(x, y)$, and let $A \subseteq \mathbb{R}^2$ be the set

$$A = \{(x, y) : f_{X,Y}(x, y) > 0\}.$$

Suppose $T : A \to B$ is an invertible map where $B \subseteq R^2$ is the image of T. Then define random variables $(U, V) = T(X, Y)$. Then the joint PDF of U, V is

$$f_{U,V}(u, v) = |J|\, f_{X,Y}(T^{-1}(u, v)) \quad \text{when } (u, v) \in B$$

where J is the determinant

$$J = \det \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v}. \end{pmatrix}$$

If $(u, v) \notin B$ then $f_{U,V}(u, v) = 0$.

The proof is omitted.

Example 4.3: If (X, Y) are uniformly distributed on the unit disc, find the joint PDF of (R, Θ) where R and Θ are the corresponding polar coordinates. Also find the marginal PDFs for R and Θ.

Example 4.4: If X and Y are independent $Exp(1)$ random variables, and $U = X + Y$, $V = X/(X + Y)$, then find the distribution of U and V.

5 The multivariate normal distribution

We met the bivariate normal distribution in MAS1604. The bivariate normal distribution is a generalization of the normal distribution to pairs of random variables, or equivalently, to a distribution on vectors in $\mathbb{R}^2$. The multivariate normal distribution is the analogous distribution on vectors in $\mathbb{R}^n$.

5.1 The bivariate normal distribution

Suppose we have constants μ_x, μ_y, σ_x, σ_y and ρ such that $\sigma_x \geq 0$, $\sigma_y \geq 0$ and $-1 \leq \rho \leq 1$. Define the 2×2 matrix Σ by

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix}.$$

Then define a joint probability density function by

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sqrt{\det\Sigma}} \exp\left(-\frac{1}{2}Q(x,y)\right) \tag{1}$$

where

$$Q(x,y) = (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \tag{2}$$

and the column vectors $\underline{x}, \underline{\mu}$ are defined by $\underline{x} = (x,y)^T$, $\underline{\mu} = (\mu_x, \mu_y)^T$. If random variables (X,Y) have joint probability density given by $f_{X,Y}$ above, then we say that (X,Y) have a **bivariate normal distribution** and write

$$(X,Y)^T \sim N_2(\underline{\mu}, \Sigma).$$

Some of the terms in the definition can be expanded:

$$\sqrt{\det\Sigma} = \sigma_x\sigma_y\sqrt{1-\rho^2}$$

and

$$\begin{aligned} Q(x,y) &= \frac{1}{\sigma_x^2\sigma_y^2(1-\rho^2)} \begin{pmatrix} x-\mu_x & y-\mu_y \end{pmatrix} \begin{pmatrix} \sigma_y^2 & -\rho\sigma_x\sigma_y \\ -\rho\sigma_x\sigma_y & \sigma_x^2 \end{pmatrix} \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix} \\ &= \frac{1}{1-\rho^2}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 + \left(\frac{y-\mu_y}{\sigma_y}\right)^2 - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\right]. \end{aligned}$$

However, equations (1) and (2) are **much** more important than the expansions just given.

It can be proved that the function $f_{X,Y}(x,y)$ in equation (1) integrates to 1 and therefore defines a valid joint PDF. The proof is omitted.

Remarks:

1. The vector $\underline{\mu} = (\mu_x, \mu_y)^T$ is called the **mean vector** and the matrix Σ is called the **covariance** matrix (or sometimes **variance-covariance** matrix).

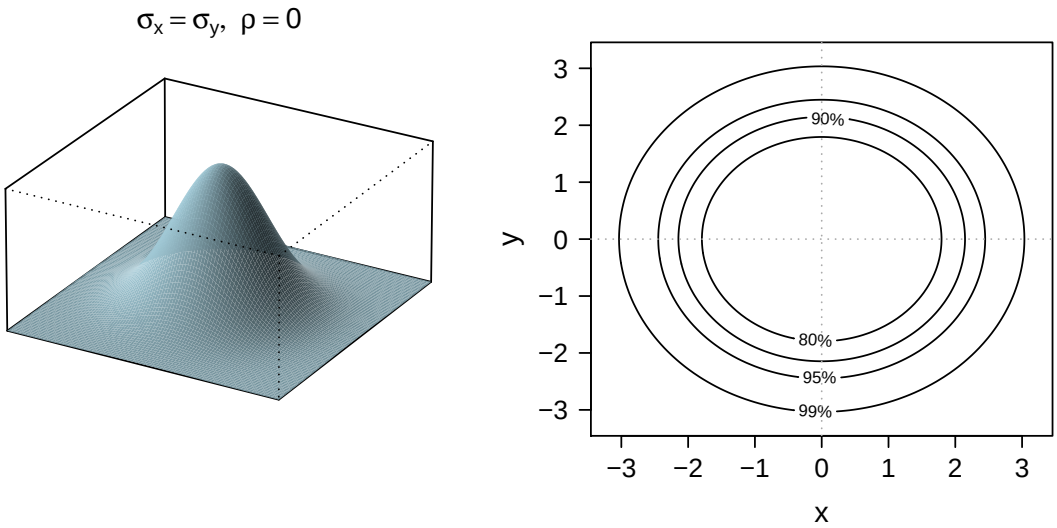
2. Functions of the form $F(\underline{x}) = \underline{x}^T\Sigma^{-1}\underline{x}$ are called **quadratic forms**. Quadratic forms are functions $\mathbb{R}^n \to \mathbb{R}$ which satisfy certain properties. They crop up in several areas of mathematics and statistics.

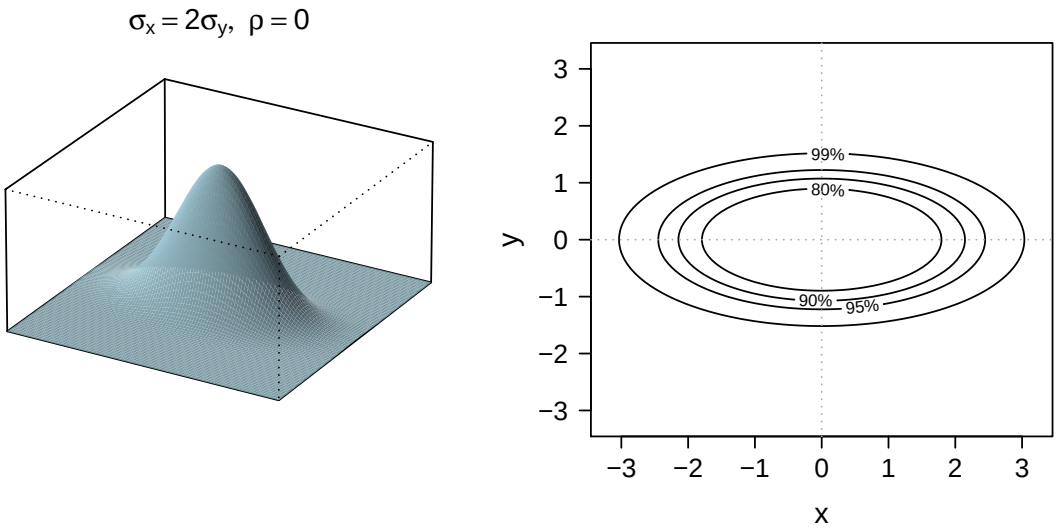
3. The matrix Σ and its inverse Σ^{-1} are **positive definite**. A matrix A is positive definite if

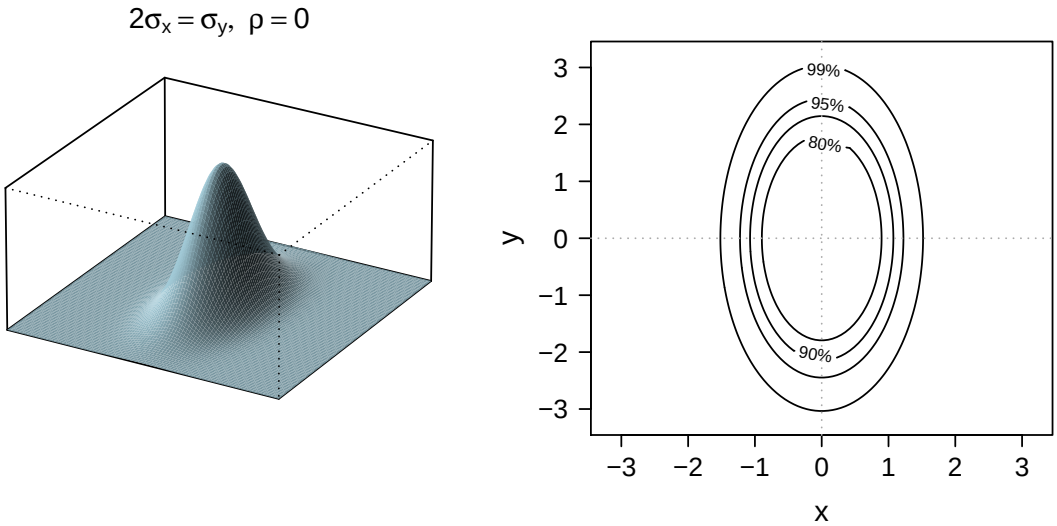
$$\underline{x}^T A \underline{x} \geq 0$$

for all non-zero vectors $\underline{x}$.

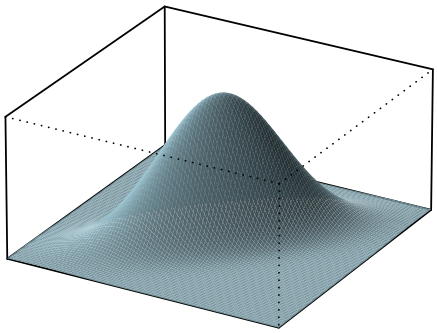
4. It follows that when $\mu_x = \mu_y = 0$, $Q(x, y)$ is of positive definite quadratic form.

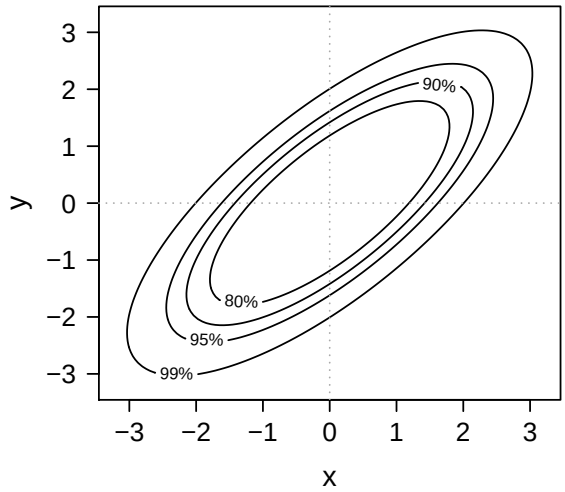

σx = σy, ρ = 0
90%
80%
95%
99%
y
x


σx = 2σy, ρ = 0
99%
80%
90%
95%
y
x

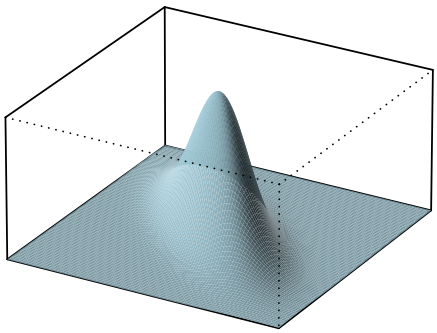

2σx = σy, ρ = 0
99%
95%
80%
90%
y
x

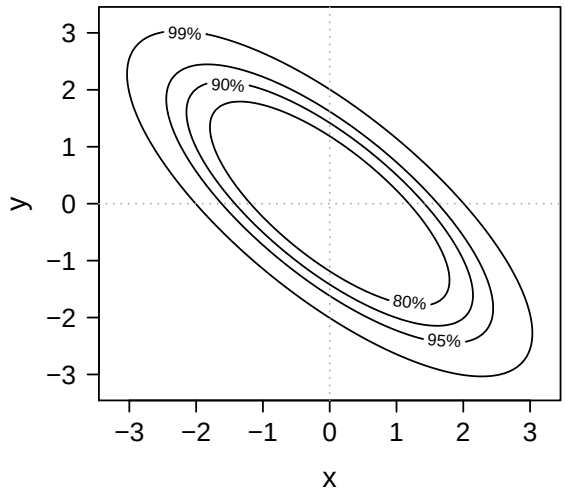
$\sigma_x = \sigma_y,\ \rho = 0.75$



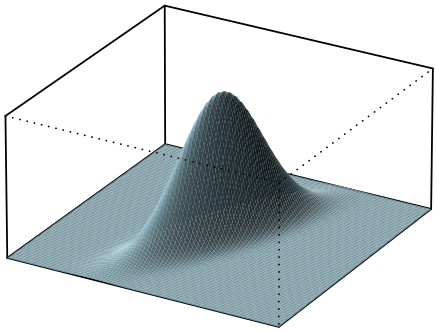


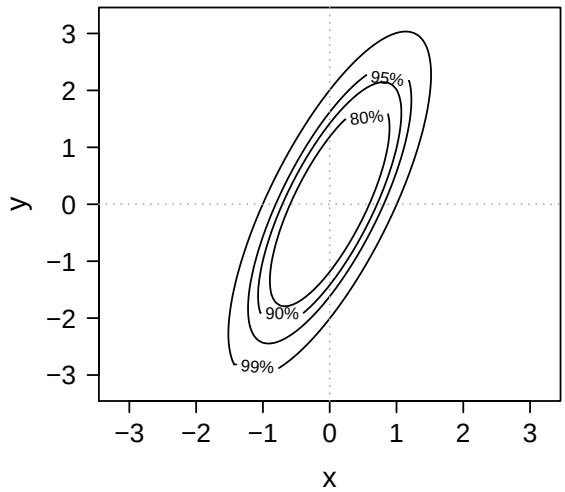
$\sigma_x = \sigma_y,\ \rho = -0.75$





$2\sigma_x = \sigma_y,\ \rho = 0.75$





The previous two pages show contour and surface plots for the density function on the xy-plane for different parameter values. Comments:

1. $Q(x, y) \geq 0$ with equality only when $\underline{x} = \underline{\mu}$. It follows that the density function has its mode at $\underline{x} = \underline{\mu}$. The above plots all have $\mu_x = \mu_y = 0$.
2. Changing the values of μ_x, μ_y does not change the shape of the plots, but corresponds to a translation of the xy-plane i.e. changing μ_x, μ_y just shifts the contours / surface to a new mode position.
3. The contours of equal density are **circular** when $\sigma_x = \sigma_y$ and $\rho = 0$ and **elliptical** when $\sigma_x \neq \sigma_y$ or $\rho \neq 0$.
4. σ_x and σ_y control the extent to which the distribution is **dispersed**.
5. The parameter ρ is the **correlation** of X, Y i.e. $\mathrm{Cor}\,(X, Y) = \rho$. (The proof is omitted.) Thus for non-zero ρ, the contours are at an angle to the axes.

When (X, Y) has a bivariate normal distribution then given a general set $A \subseteq \mathbb{R}^2$, calculation of the probability $\Pr((X, Y) \in A)$ requires a computer. However, the marginal and conditional distributions of the bivariate normal distribution are particularly simple – one of the reasons the distribution is so useful in statistics.

Suppose $(X, Y)^T \sim N_2(\underline{\mu}, \Sigma)$. The following results hold.

1. The marginal distributions are normal:
$$X \sim N(\mu_x, \sigma_x^2) \quad \text{and} \quad Y \sim N(\mu_y, \sigma_y^2).$$
2. The conditional distributions are normal:
$$\begin{aligned} X|Y = y &\sim N(\mu_x + \rho\frac{\sigma_x}{\sigma_y}(y - \mu_y), \sigma_x^2(1 - \rho^2)) \quad \text{and} \\ Y|X = x &\sim N(\mu_y + \rho\frac{\sigma_y}{\sigma_x}(x - \mu_x), \sigma_y^2(1 - \rho^2)). \end{aligned}$$
3. When $\rho = 0$, X and Y are independent.
4. Linear combinations of X and Y are also normally distributed:
$$aX + bY \sim N(a\mu_x + b\mu_y, a^2\sigma_x^2 + b^2\sigma_y^2 + 2ab\rho\sigma_x\sigma_y)$$
where a, b are constants. Thus a **linear combination** of normal random variables is also normally distributed itself.

Proofs are omitted.

5.2 Simulating bivariate normal random variables

R has built-in functions to simulate bivariate and multivariate normal random variables. However, we can simulate using the strategy introduced in section 4.1: first simulate from the marginal for X (or marginal for Y if preferred), and then simulate Y from the conditional $Y|X = x$.

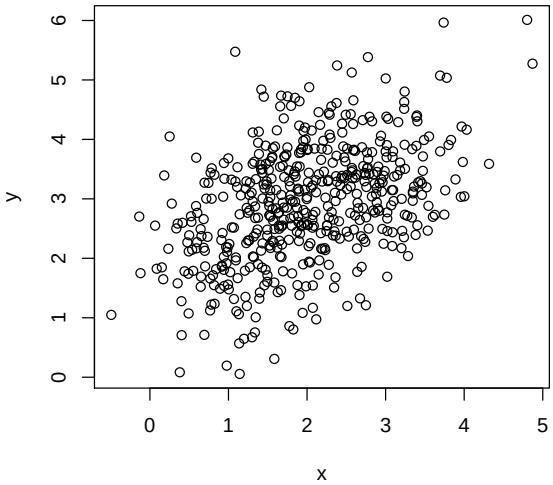
Example 5.1: Suppose $(X, Y)^T \sim N_2(\underline{\mu}, \Sigma)$ where $\mu_x = 2$, $\mu_y = 3$, $\sigma_x = 1$, $\sigma_y = 1$ and $\rho = 0.5$. Simulate a sample of size 500 from this distribution and draw a scatter plot. Use simulation to find $\Pr\big(X^2 + Y^2 < 9\big)$.

The marginal distribution of X is $X \sim N(2, 1^2)$. Using the formula for the conditional

$$\begin{aligned} Y|X = x &\sim N(\mu_y + \rho\frac{\sigma_y}{\sigma_x}(x - \mu_x), \sigma_y^2(1 - \rho^2)) \\ &\sim N(3 + 0.5(x - 2), 0.75). \end{aligned}$$

To sample:

```
npts = 500
x = rnorm(npts, mean = 2, sd = 1)
y = rnorm(npts, mean = 3+0.5*(x-2), sd = sqrt(0.75))
```



To find $\Pr\big(X^2 + Y^2 < 9\big)$ approximately, count the number of points in the region:

```
npts = 10000
x = rnorm(npts, mean = 2, sd = 1)
y = rnorm(npts, mean = 3+0.5*(x-2), sd = sqrt(0.75))
f = x^2+y^2
sum(f<9)/npts
[1] 0.2877
```

5.3 The multivariate normal distribution

The multivariate normal distribution is a distribution on vectors in $\mathbb{R}^n$. Suppose that $\underline{X}$ is a random vector with n entries, i.e. $\underline{X} = (X_1, \ldots, X_n)^T$. Then

$$\underline{X} \sim N_n(\underline{\mu}, \Sigma)$$

if $X_1, \ldots, X_n$ have joint PDF given by

$$f_{\underline{X}}(\underline{x}) = \frac{1}{2\pi\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}Q(\underline{x})\right)$$

where

$$Q(\underline{x}) = (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}).$$

This definition makes sense for any column vector $\underline{\mu} \in \mathbb{R}^n$ and any **positive definite** $n \times n$ matrix Σ. Some remarks:

1. The vector $\underline{\mu}$ is the **mean** of the distribution and Σ is called the **covariance** matrix.
2. All the marginal distributions of $\underline{X}$ are normal. (We do not specify their parameters here, however).
3. Similarly, all the conditional distributions of $\underline{X}$ are normal. (Again, we do not specify the parameters of these distributions here).

Example 5.2: Suppose $\underline{X} \sim N_3(0, \Sigma)$ where

$$\Sigma = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 3 \end{bmatrix}.$$

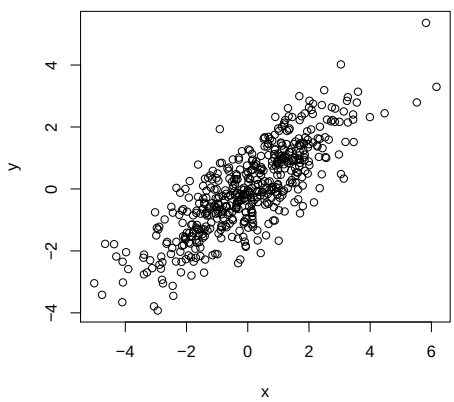
Use R to sample from this distribution. Create a scatterplot of (X, Y) samples and compute the correlation.

In R, you use the `mvrnorm` function to sample:

```
library(MASS)
npts = 500
mu = c(0, 0, 0)
sigma = matrix(data = c(3,2,1,2,2,1,1,1,3), nrow = 3, byrow = TRUE)
x = mvrnorm(npts, mu, sigma)
```

Now `x` has dimension `npts`$\times 3$, so column 1 is X and column 2 is Y:

```
plot(x[,1], x[,2], xlab = 'x', ylab = 'y')
```



The sample correlation is:

```
cor(x[,1],x[,2])
[1] 0.812629
```

6 Samples and populations

In this section we consider the fundamental statistical concepts of **populations** and **samples**. We investigate the properties of samples, particularly as the sample size gets large, via computer simulation. This material is intended to reinforce the ideas studied in MAS2901.

Example 6.1: A factory has been depositing mercuric waste into a river near its estuary. It is feared that this waste may have caused the mercury content in the prawns there to have risen to a level which may be toxic for human consumption. The maximum permissible level is one part per million and the factory claims the average mercury content in the prawns does not exceed this level. This claim is to be tested.

It is obviously impractical to measure the mercury content of all the prawns. A random sample of 20 prawns is collected. The concentration of mercury (parts per million) in each of the 20 prawns is measured, giving the following results: the observed sample mean is $\bar{x} = 1.2$, and the observed sample standard deviation $= s = 0.5$.

Given these measurements do we believe the factory's claim?

6.1 Populations

Suppose we measure some random quantity X: X can adopt a range of possible values, and some values are more likely than others. This is, of course, the familiar idea of a **distribution** for X. Usually we do not know this distribution exactly, but we want to learn something about it. The unknown distribution is called the **population distribution**.

Referring to the prawn example:

- the population consists of the prawns in the estuary;
- the random quantity X is the mercury concentration in a randomly selected prawn; and
- the population distribution is the distribution of X.

We are usually interested in key properties of the population distribution such as:

- the expectation of X – usually called the **population mean**;
- the variance of X – usually called the **population variance**; or
- the 95th percentile of X (for example).

Often we make some simplifying assumptions about the population distribution. For example, we might assume:

(a) X is normally distributed with unknown mean and variance;

(b) X is exponentially distributed with rate parameter λ, where λ is unknown but lies on the interval $(0, 1)$;

(c) X is normally distributed with unknown mean and variance $\sigma^2 = 5$.

A set of assumptions like this is referred to as a **model**. Typically a model involves some unknown parameters. In the examples above we have the unknown rate λ in (b), and unknown mean μ in (c).

In some situations – usually rather artificial ones – we know the population distribution exactly. For example:

- let X be the score obtained from rolling a fair die; or
- let X be the number on a card drawn at random from a full deck. (Assume Jack, Queen, King numbered 11, 12, 13 respectively.)

6.2 Samples

Typically we do not know everything about the population distribution. We learn about the population distribution by drawing a **sample**. Referring to the prawn example: we cannot measure the mercury concentration in every prawn in the estuary, so we sample 20 prawns at random and analyse those.

A sample of size n corresponds to taking n independent measurements from the distribution. Each measurement is itself a random variable with the same distribution as X: the sample measurements denoted $X_1, X_2, \ldots, X_n$. So in our example, X_1 represents the mercury concentration in the first prawn collected – and this is a random quantity. The actual measurements obtained are denoted $x_1, x_2, \ldots, x_n$.

The distinction between the population distribution and how we learn about the population from limited samples is probably the most **important concept** in statistics.

6.3 Estimators

Suppose we wish to learn about some aspect of the population distribution: the population mean or the population variance for example. To do this we construct an **estimator** for the quantity of interest. For example, if we wish to learn about the population mean, a good estimator is the sample mean

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

(We won't address what a 'good' estimator might be in this module.) Formally, an estimator is defined to be some **function** of the sample, i.e. an estimator S is a random variable for the form

$$S = g(X_1, X_2, \ldots, X_n)$$

for some function g. It follows that every estimator is a **random variable**. When we observe some measurements $X_1 = x_1, \ldots, X_n = x_n$ then we can compute the corresponding value of the estimator:

$$s = g(x_1, x_2, \ldots, x_n).$$

This value is called an **estimate**. To summarize: an estimator S is a random quantity which is a function of the sample; whereas an estimate is an observed value of that random quantity.

Since any estimator S is a random variable it makes sense to talk about its **distribution**. The rest of this section is concerned with studying the distribution of some simple estimators.

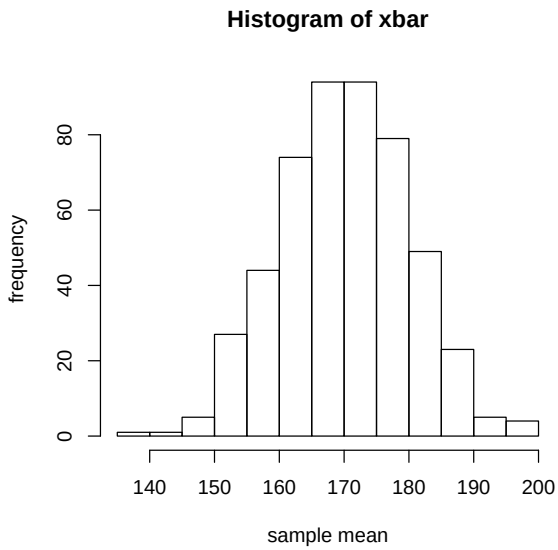
Example 6.2: Suppose the population distribution is normal, and we wish to estimate the population mean. Suppose the sample size is $n = 4$ and our estimator is $\bar{X} = (X_1 + X_2 + X_3 + X_4)/4$. What is the distribution of $\bar{X}$ when the population distribution is $N(170, 20^2)$?

Simulation can be used to answer this question approximately: we can simulate lots of samples and look at the resulting distribution.

```
simulate.xbar = function(n) {
    xbar = vector(mode = 'numeric', length = n)
    for (i in 1:n) {
        x = rnorm(4, 170, 20) # Generate a sample of size 4
        xbar[i] = 0.25*sum(x)
    }
    xbar
}
```

Histogram of $\bar{X}$:

```
test = simulate.xbar(500)
hist(test, xlab = 'sample mean', ylab = 'frequency')
```



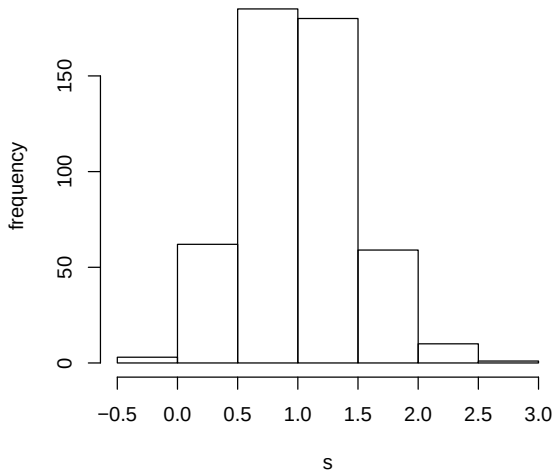
Example 6.3: Suppose the population distribution is normal, and we wish to estimate the 90th percentile using a sample of size 10. A sensible estimator is to define S to be the second largest value in the sample (i.e. the 9th value when the samples are ordered from smallest to largest). What is the distribution of S when the population distribution is $N(0, 1)$?

Code to simulate the distribution of S:

```
simulate.percentile = function(n) {
    s = vector(mode = 'numeric', length = n)
    for (i in 1:n) {
        x = rnorm(10, 0, 1) # Generate a sample of size 10
        x = sort(x)
        s[i] = x[9] # Get 9th value on sorted list
    }
    s
}
```

Histogram of S:

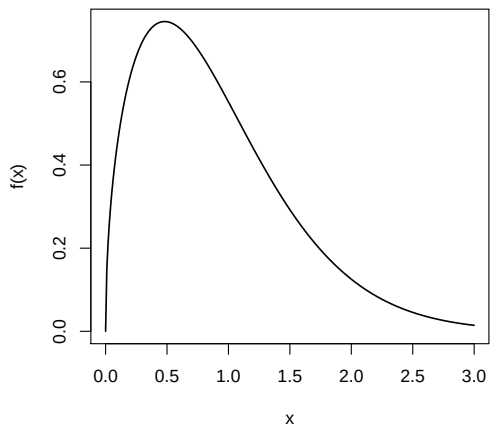
```
S = simulate.percentile(500)
hist(S, xlab = 's', ylab = 'frequency', main = '')
```



6.4 Simulation experiment

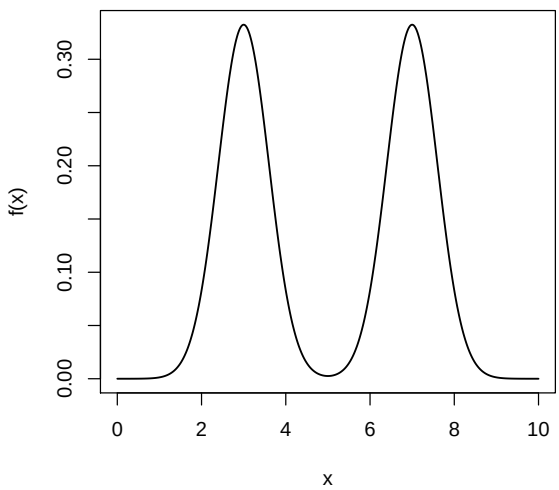
Consider the following two examples for the density of the population distribution. For each example, decide which histogram on the slides (A, B, C or D) is most likely to represent the distribution of the sample mean $\bar{X}$ when the sample size is 10.

Example 6.4: Suppose X (the population quantity) has the following PDF:



Most likely histogram:

Example 6.5: Suppose X (the population quantity) has the following PDF:



Most likely histogram:

Conclusion:

- The sample mean is distributed around the population mean.
- The distribution of sample mean values 'forgets' the underlying shape of the population distribution.
- As n increases we expect the distribution of $\bar{X}$ to become more clustered around the true value.

6.5 The Central Limit Theorem

The Central Limit Theorem is the most important theorem in statistics. It describes the distribution of the sample mean $\bar{X}$ as the sample size increases.

Suppose $X_1, X_2, \ldots, X_n$ are independent and **identically distributed** random variables with common mean μ and variance σ^2 which are both finite. Define

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}.$$

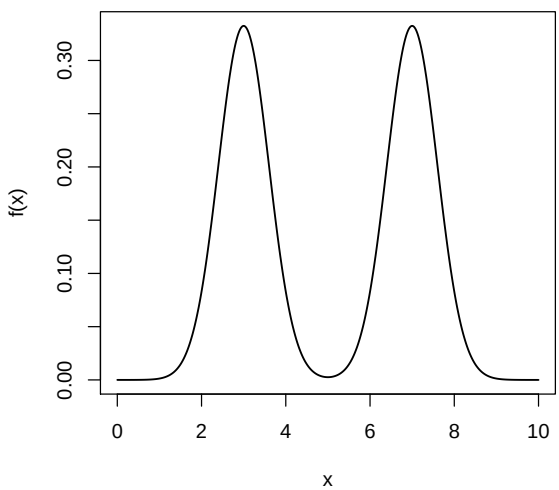
Then as $n \to \infty$ the distribution of Z tends to $N(0, 1)$.

While this results describes the approximate distribution of the sample mean as n gets large, it can also be adapted and extended to understand the asymptotic behaviour of other estimators.

Remarks:

1. Equivalently, the Central Limit Theorem shows that for large n, $\bar{X} \sim N(\mu, \sigma^2/n)$ approximately.
2. It follows that $\bar{X}$ is distributed somewhere around the population mean μ.
3. As n increases, the variance of the distribution of $\bar{X}$ decreases. In other words, $\bar{X}$ is more likely to lie closer to μ – "larger samples give better estimates".
4. The underlying population distribution is forgotten as n increases, as $\bar{X}$ has a normal distribution.

Example 6.6: Suppose X has the bimodal distribution considered in example 6.5:



This is a normal mixture distribution, and the following function can be used to sample values $\bar{X}$ from the distribution when the sample size is n.

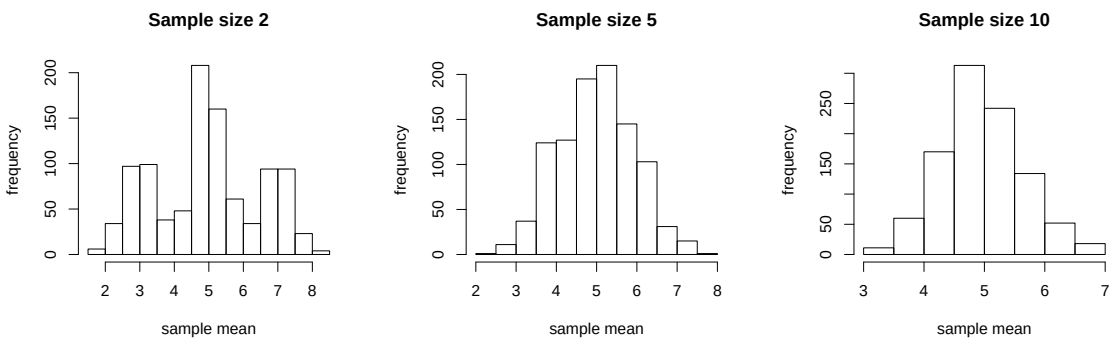
```
simulate.bimod = function(k, n) {
    # Generate k samples of size n
    s = vector(mode = 'numeric', length = k)
    for (i in 1:k) {
        u = rnorm(n, 3, 0.6)
        v = rnorm(n, 7, 0.6)
        r = runif(n)
        x = c(u[r>0.5], v[r<=0.5])
        s[i] = mean(x)
    }
    s
}
```

Use this function to study the distribution of $\bar{X}$ for sample sizes of $n = 2$, $n = 5$ and $n = 10$. The population mean is $\mu = 5$ and $\sigma^2 = 4.3$.

Generate samples:

```
samples.size.2 = simulate.bimod(1000, 2)
samples.size.5 = simulate.bimod(1000, 5)
samples.size.10 = simulate.bimod(1000, 10)
```

Histograms:



We can compare the observed means and variances to the values predicted from the Central Limit Theorem:

Sample size n	μ	σ^2/n	Observed mean of $\bar{X}$	Observed variance of $\bar{X}$
2	5.0	2.15	4.94	2.27
5	5.0	0.86	4.98	0.862
10	5.0	0.43	4.96	0.443

The observed mean and variance of $\bar{X}$ for $n = 2$ are calculated by:

```
 mean(samples.size.2)
[1] 4.939386
 var(samples.size.2)
[1] 2.270115
```

Further examples will be studied in the practicals.