MATH5743M: Statistical Learning

Dr Seppo Virtanen, School of Mathematics, University of Leeds

Semester 2: 2022

Week 1: Introduction

Statistical learning is the practice of using statistical methods to recognise patterns in data, to identify the processes that produced the data and to make predictions using that knowledge. Statistical learning is important across many areas of science and industry. Data science is one of the fastest growing sectors of academia and many large corporations now use sophisticated data analytics to understand what their customers want, to predict trends in demand for certain products and to identify new market opportunities. Data science consists of the fields of computer science, mathematics and statistics and domain/application knowledge. Exciting new developments in artificial intelligence such as automated vehicles and digital personal assistants are all based on principles of statistical modelling. This course will introduce you to the foundations of statistical learning. You will learn how to select the right statistical model to analyse a data set, how to use statistical models to understand and predict data and see how real world examples make use of statistical models to solve both academic and industrial problems.

Examples

Who uses statistical models? Why and how?

- Stockmarket traders use statistical models to *predict* the value of stocks in the future based on previous stock values, company earnings and national economic trends.
- Companies like Facebook, Google and Amazon use statistical models to *predict* which adverts or products you are most likely to find interesting.
- Police try to *predict* where, when and what type of crime will occur.
- Security services such as GCHQ use statistical models to *infer* possible threats from voice recordings, email records and video tracking.
- Criminologists aim to *infer* causes of crime based on socio-demographic and descriptive spatial features.
- Biologists and social scientists use statistical models to *infer* what drives human and animal behaviour, and to *predict* what individuals will do when their situation changes.
- Doctors use statistical models to *infer* which patients are likely to respond to a specific drug, and to *predict* which individuals are likely to get sick based on previous medical records.
- Self-driving cars use statistical models to *decide* on the best action to take, using cameras, GPS and data from human drivers.

Figure 1: Statistical models are used in finance, astronomy, biology and robotics

Course outline and assessment

Over this semester we are going to learn about a variety of statistical modelling techniques: how they work, when they are used, and how complex statistical models can be constructed from a few fundamental ingredients. Each week will be focused on a particular topic, with two weekly lectures followed by a practical session. The lectures will be kept weekly on Mondays at 9am; the first lecture is on 24 Jan, week 1. The practicals will be arranged on Mondays at 11am on even semester weeks; the first practicals is on Monday 31 Jan, week 2. Lectures are kept online via MS Teams and the team channel is also used for providing information and posting/answering questions. Practicals are kept live at Cohen A and B cluster (1.40). All material will also be made available on Minerva on a weekly schedule.

In the practical sessions you will be expected to work through the problems provided, and complete these outside of class if needed before the next practical class. During the practicals, you will have opportunity to ask questions from me and I will be there to give guidance.

The goal of this course is for you to understand the underlying principles of statistical learning, illustrated through several key types of statistical model. You will also learn how to apply this understanding practically and efficiently using the R statistical software package. At times we will initially forgo using the standard R tool for a particular analysis so that we can see how the analysis works 'under the hood' - how all the details of the analysis are performed. But we will also return to the standard tools to perform analyses so that you gain experience in using the most efficient methods available.

The notes I provide you will be focused mostly on how to perform all of the statistical modelling techniques we will learn about in R. It is important to note that I will not always go through detailed instructions about using an R command in lectures. Instead I will focus that time on teaching how a method is developed, how it works and important factors to consider when using it. That means you will need to read through the notes provided in your own time as well, try the example code I use, and research each new command by reading the help files and online R tutorials. I have written these notes in R itself, so everything you see should run when copied and pasted onto the command line.

I have tried to provide every step necessary to produce most of the figures and analyses you see presented in the notes, but please give me feedback on anything you find hard to understand or which seems incomplete. Nonetheless, we will inevitably find some errors as we go along - please do point these out to me if you find them! I will update the notes held on the Minerva whenever I make any changes so you can always find the most recent version there.

The weekly themes of this module will be:

1. Introduction
2. Optimisation and learning
3. Linear regression 1
4. Linear regression 2
5. Model selection
6. Logistic regression
7. Decision trees
8. Ensemble learning and collective intelligence
9. Random Forests
10. Further topics/Revision

These themes are designed to build up your knowledge in stages. It is therefore essential that you attend all lectures and complete all practical assignments if you wish to perform well in the final assessment.

Assessment will be by a mixture of an exam (50%) and three pieces of assessed practical work (50%). The deadlines for the three assessed practicals are:

1. 11:59pm Friday March 18
2. 11:59pm Friday April 8
3. 11:59pm Friday May 13

You need to return each **assessed** practical report by the deadline in Minerva using Turnitin. I will return to you marks and feedback. Details about the exam will be provided later.

If you have a good reason for being unable to complete the work on time please let me know *in advance*. Late submission without warning will reduce the maximum mark available.

The R statistical package

This course is both applied and theoretical. You will need to perform statistical analyses on a computer to complete exercises and to better understand the methods we cover in theory. For this we will use the free software *R*, a general-purpose statistical software package. You should have already encountered R in the previous semester. Throughout the course notes you will find instructions on perfoming operations in R, which will be written in text like this:

```
a = 3
x = 1:10
y = a*x + rnorm(10)
```

You should use copy-and-paste to run these code segments yourself between lectures and in practicals. Experiment with changing the numbers used in examples to see how the results change. Check that you understand what each function is doing as you run the code. You can get help on any function in R by typing help(function_name), replacing function_name with the function you want help on. E.g.

```
help(rnorm)
```

If you don't know the name of the function you need to do a specific job, you will find a wealth of information online. Try Googling **R function for X**, or **how to X in R**, where X describes the task you want. For instance, Googling for **how to generate random numbers in R** gives the top hits shown in Figure 2. These vary from web versions of the R help files to blogs and dedicated R cookbooks.

When you get an error message and an R command you tried to run fails (this will happen, a lot!), take the following four-step routine to fixing the problem:

- Pause. Think carefully about the command you have entered. A moment's refelction often reveals an easily fixed mistake
- Check the help files for how the command you are using works. Can you spot any differences with your code? Try the examples given in the help file.
- Google the error message – someone has probably had the same problem as you before and fixed it. As the saying goes, if you're not Googling Stack Overflow you're working too hard!
- Ask me! In practicals I can help you with any problems quickly, but don't spend time waiting for me to reply to an email before you've tried to fix the problem yourself.

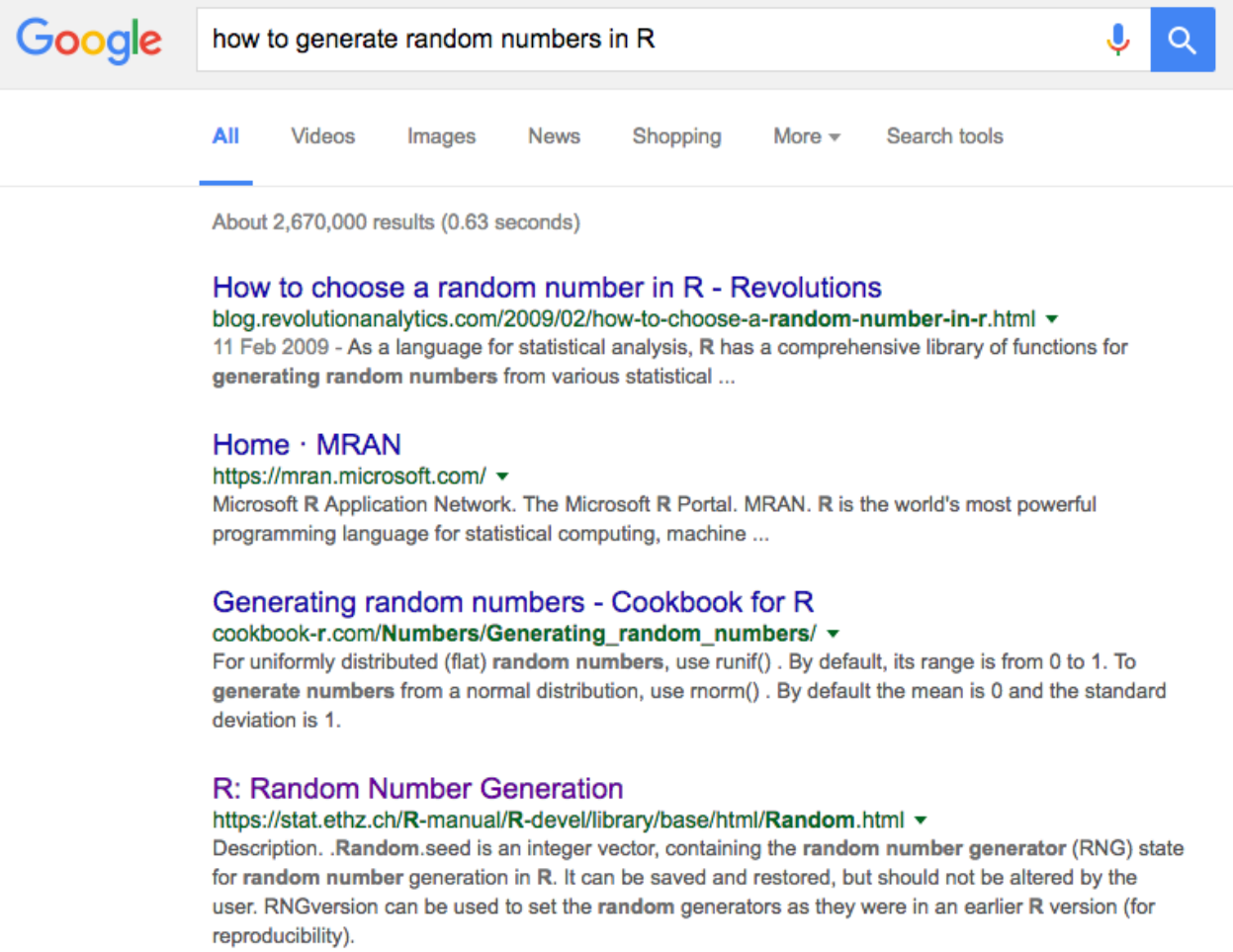


Figure 2: Top hits for 'how to generate random numbers in R'

IMPORTANT: before the first practical, download the practical worksheet and read the section on using R Markdown to document your work. This will make your practical work much easier and save you time in class.

Figure 3: There are many powerful statistical tools available through R and other software. But, as this tweet from computational social science conference points out, the key to using these wisely is to know understand how each works.

Supervised learning

There are many different types of statistical modelling suited to different practical problems. In this course we will focus on models that belong to a class known as *supervised learning*. The goal of supervised learning is to find a model that can *predict* the values of some *outputs* based on matching sets of *inputs*. In classical statistics the inputs are often known as *independent variables*, and the outputs as *dependent variables*. In this course we will use the more generic terms, inputs and outputs.

In general, a supervised learning model uses observed inputs to predict the conditional probability that the matching output takes a specific value, using a function $f()$ that specifies the statistical model. The functional form of the model usually involves some free parameters, which we have to *learn* in order to make the model perform well. In general we denote these parameters as θ.

$$P(\text{OUTPUT} = y \mid \text{INPUT} = x, \theta) = f(y, x, \theta) \tag{1}$$

The function $f()$ is like a machine that takes in a specific set of inputs (x), specified values of the free parameters (θ) and tells us how likely a given output (y) is. In general our task as statisticians can be broken down into a few key components:

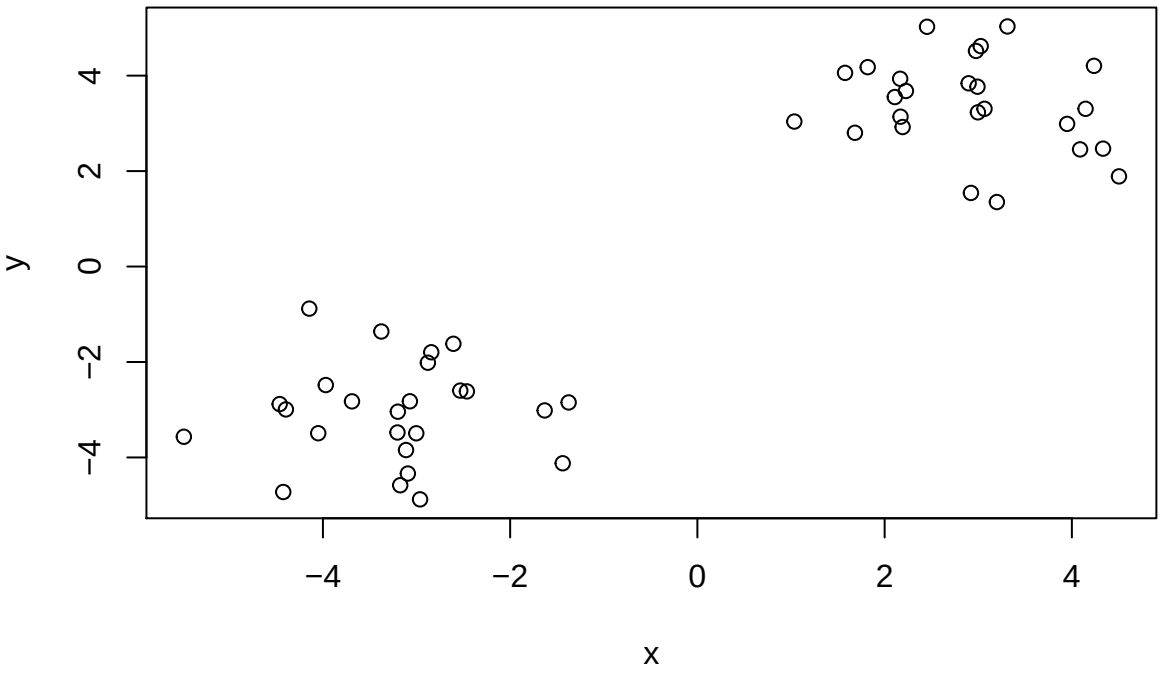
- *Choose* an appropriate functional form for $f()$, alternatively, a probability density function for the outputs conditioning on the inputs and parameters
- Using a data set of matched inputs and outputs, *learn* the best values for the free parameters, θ
- *Interpret* the values of θ - what do they tell us about the real system we are studying?
- Use $f(\cdot, \cdot, \theta)$ to *predict* the values of new outputs, given their matching inputs.
- Model evaluation and selection. Which model performs best?

This course will take us through these different steps. You will learn about different types of function we can use, and how these can be split into *regression* models and *classification* models based on the domain/type of the outputs. Few relevant probability distributions are normal for continuous and numeric outputs, Bernoulli

for binary yes/no outputs and multinomial/categorical for one-out-of-K classes/outputs. We will see how we can use data to *infer* the values of free parameters in the model, and how to intepret the values that we obtain. Then you will learn how to use a statistical model with these parameters to make predictions, and judge how good those predictions are. When we have finished you will be able to use statistical modelling to solve many different real world problems.

Unsupervised learning

You may at this point wonder, if this is supervised learning, what is unsupervised learning? Where do these descriptions come from? First, lets look at an example of unsupervised learning and then discuss the differences. A classic unsupervised learning task is *clustering*, the partitioning on a number of data points into two or more distinct groups. For example, consider the set of data points plotted below:



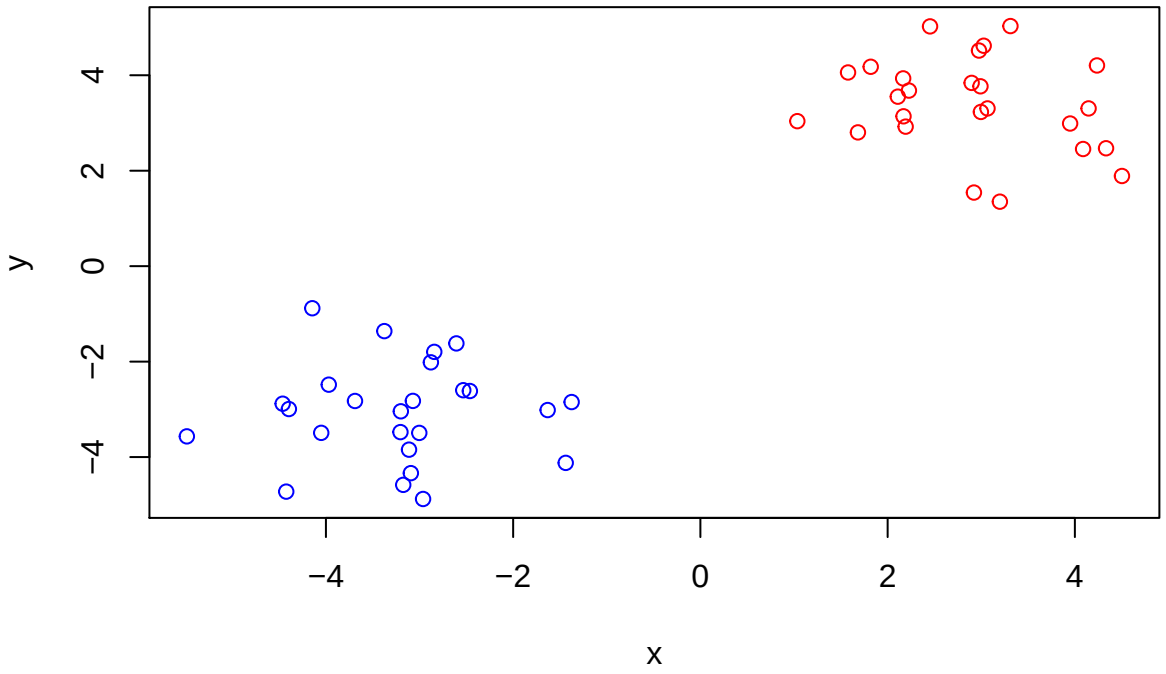
Visually we can see that there are clearly two distinct groups of points, but how could we use a computer to automatically identify where the two clusters are and which points belong to them? One standard algorithm is called k-means clustering, and works as follows:

1. Specify two randomly selected 'centres' for the clusters
2. Group points according to which centre they are closest to
3. Reposition the centre to the mean position of the points in that cluster
4. Go back to step 1 and repeat until the cluster membership stops changing

If we perform this algorithm in R we can see if it works. I previously stored the x and y positions on each point in a 50 x 2 matrix, D. R has a function **kmeans** that will perform the algorithm for us - we need to provide the data and the number of centres we want (which in this case is 2). Once the algorithm has completed we extract the cluster identities from the output and plot the points colour-coded by cluster

```
kmeans_object = kmeans(D, 2)
cluster_id = kmeans_object$cluster
```

```
cols = c("Red", "Blue")
plot(D[, 1], D[, 2], col=cols[cluster_id], xlab="x", ylab="y")
```



In this case we get a perfect clustering, because the problem was set up to be very easy.

This algorithm demonstrates some of the differences between supervised and unsupervised learning. In this case there were no output labels attached to the data points. We were not given any definitive examples of points that belonged to either cluster. Instead we tried to find structure directly in the data to construct those clusters ourselves (or the computer did on our behalf). The term *supervised* in supervised learning relates to the provision of training labels. Historically this is because in such algorithms a human operator was needed to provide these labels to the machine, hence 'supervising' the learning process. Nowadays these labels might be provided automatically, but the distinction between the two methodologies remains.

In some cases researchers use a combination of both methods in their statistical analyses. For example, they might use the k-means clustering algorithm to preprocess input values, before using the cluster labels as inputs for a supervised learning task. In general this is called *semi-supervised* learning.

A note on use of data in this course

During this course you will see me repeat a similar procedure again and again. That is, as I did above, I will create data to analyse myself, by simulating a particular type of process on the computer. I will then use statistical models to analyse this simulated data, to see if these models can recover the relationships that I know are there (because I created it!). This is a very good way to check if your statistical model works the way you hope: if the data really are generated the way that your model assumes they are, it should be able to retrieve the properties that you gave that data accurately.

You may wonder, why don't you always use real data instead? Surely we should be learning how to deal with real data and real problems. Of course, we will look at real data as well. In general I will mostly use simulated data in the lecture notes, and I will more often use real data for the practical sessions. This is because, in most cases, if we use real data we don't necessarily know what the answer 'should' be. Therefore it is harder

to tell if the model we are using is working. So don't panic, you will certainly get plenty of experience dealing with real data. But it is also important that you learn how to test models, how to see when they work and fail. For this, it is important to be familar with the process of creating you own simulated data to perform tests.

Regression and Classification

Supervised learning problems can be split into two broad categories: *regression* and *classification*.

Regression

When we talk about regression in statistics the first thing that probably comes to mind is linear regression - fitting a straight line to data. Linear regression is a very important and surprisingly powerful statistical method that we will address in this course, but regression is a broader concept than just fitting straight lines. In standard terminology a regression model is one where the outputs are numerical and continuous. To illustrate with an example: in the UK, the Met Office is reponsible for making weather forecasts. This includes predicting what the temperature will be tomorrow, and up to several weeks ahead. To do this they employ highly complex models of the atmosphere and oceans, using the current and past state of the climate as inputs to predict the future. Predicting the temperature is a regression task; the output, temperature, is a continuous numerical variable. Errors in the forecast will also be continuous and numerical. In a regression task, because the output is continuous, the model will predict the probability *density* of observing any given output. Conversely, predicting whether or not it will rain at some point tomorrow is not a regression task. The output, rain or no rain, is a discrete, binary variable. Making predictions about discrete variables is the task of *classification* models.

Classification

We previously considered regression problems, where the outputs are real numbers. Another important class of problems is *classification*, where the outputs are one of a series of discrete *classes*. Classification problems are very common in our daily lives. Consider when you add photos to your Facebook account. Facebook will analyse the photo and detect faces within it. Often it will then suggest which of your friends these faces might belong to. Predicting the identity of an individual is a classification task – the statistical model needs to determine the probability that the face belongs to one of many different classes – the identities of all your friends – based on inputs from the pixels in the photgraph. In classification tasks, since the outputs are discrete the model will predict the probability that the output will be of any given class.

Further notes on terminology

Although the definition above are quite standard, you may encounter these terms used somewhat differently at times. Classification is almost always used to describe models where the outputs are discrete classes. Regression is sometimes used to cover all supervised learning models, both continuous and discrete. For example, the main classification model we will investigate in this course is typically called 'logistic regression'. This can obviously be confusing if you are just starting to understand the differences between methods. For the purpose of this course you should work with the definitions above, but the important thing to think about with any model is what type of inputs and outputs can it accept and predict, and do these match the data you are trying to analyse. Almost all supervised learning methods and some unsupervised ones are intimately connected to each other, and so it can be difficult to draw precise boundaries between them. The

purpose of this course is to give you a deep enough understanding of how they work mathematically that you can intelligently select the right tool for the job.