Multivariate Statistics

Prof. Richard Wilkinson

Spring 2021

Contents

Introduction

Warning: these lecture notes are still in preparation. Chapters 1-4 have been finished. Chapters 5-6 are being worked on. Later chapters will appear once a reasonable draft is available.

This module is concerned with the analysis of multivariate data, in which the response is a vector of random variables rather than a single random variable.

Part I of the module describes some basic concepts in Multivariate Analysis and then recaps and introduces some key ideas needed from linear algebra. Chapter 1 defines notation, introduces some datasets, and discusses exploratory data analysis. Chapter 2 provides a recap on some matrix algebra. Much of this will be familiar to you, but if not, we take the time to introduce the key mathematical concepts that will be relied upon during the module. Chapter 3 introduces matrix decompositions. We start with the spectral decomposition of square symmetric matrices (which you will have studied previously), and then introduce the singular value decomposition (SVD). The SVD is one of the most important concepts in this module, and is the key linear algebra technique behind many of the methods we will study.

A theme running through the module is that of dimension reduction. In Part II we consider three types of dimension reduction: Principal Components Analysis (in Chapter 4), whose purpose is to identify the main modes of variation in a multivariate dataset; Canonical Correlation Analysis (Chapter 5), whose purpose is to describe the association between two sets of variables; and Multidimensional Scaling (Chapter 6), in which the starting point is a set of pairwise distances, suitably defined, between the objects under study.

In Part III, we focus on methods of inference for multivariate data whose distribution is multivariate normal.

Finally, in Part IV, we focus on different methods of classification, i.e. allocating the observations in a sample to different subsets (or groups).

If you find any typos or mistakes, please email me at `r.d.wilkinson@nottingham.ac.uk` The notes have been significantly rewritten this year in order to adapt them for remote learning, and I am keen to fix as many of the mistakes as I can!

PART I: Prerequisites

Much of modern multivariate statistics (and machine learning) relies upon linear algebra. Consequently, we will spend some time reminding you of the basics of linear algebra (vector spaces, matrices etc), and introducing a few additional concepts that you may not have seen before. It is worth spending time familiarizing yourself with these ideas, as we will rely heavily upon this material in later chapters.

In Chapter 1 we explain what we mean by multivariate analysis and give some examples of multivariate data. We introduce basic definitions and concepts such as the sample covariance matrix, the sample correlation matrix and describe some simple exploratory data analysis techniques.

In Chapter 2 we summarise the definitions, ideas and results from matrix algebra that will be needed later in the module, most of which will be familiar to you. In particular, we will introduce vector spaces and the concept of a basis for a vector space, discuss the column, row and null space of matrices, and discuss inner product spaces and projections. We also define the centering matrix.

In Chapter 3 we recap the eigen or spectral decomposition of square symmetric matrices, and introduce the singular value decomposition (SVD) which generalises the concept of eigenvalues for non-square matrices. We will rely upon this material inlater chapters.

Chapter 1

Statistical Preliminaries

In this chapter we will define some notation, and recap some basic statistical properties and results.

There are recorded videos for the following topics in this chapter:

- Notation and datasets
- Exploratory data analysis
- Random vectors

1.1 Notation

We will think of datasets as consisting of measurements of p different **variables** for n different **cases/subjects**. We organise the data into an $n \times p$ **data matrix**.

Multivariate analysis (MVA) refers to data analysis methods where there are two or more **response** variables for each case (you are familiar with situations where there is more than one explanatory variable, e.g., multiple linear regression).

We shall often write the data matrix as $\mathbf{X}$ $(n \times p)$ where

$$\mathbf{X} = \begin{bmatrix} - & x_1^\top & - \\ - & x_2^\top & - \\ - & \vdots & - \\ - & x_n^\top & - \end{bmatrix}$$

The vectors $x_1, \ldots, x_n \in \mathbb{R}^p$ are the observation vectors for each of the n subjects.

- The n *rows* of $\mathbf{X}$ are $x_1^\top, \ldots, x_n^\top$ - each row contains the p observations on a single subject.

- The p columns of $\mathbf{X}$ correspond to the p variables being measured, i.e., they contain the measurements of the same variable across all n subjects.

Important remark on notation: Throughout the module we shall use

- non-bold letters, whether upper or lower case, to indicate scalar (i.e. real-valued) quantities, e.g., x, y
- lower-case letters in bold to signify column vectors, e.g., $\mathbf{x}, \mathbf{y}$
- upper case letters in bold to signify matrices, e.g., $\mathbf{X}, \mathbf{Y}$.

This convention for bold letters will also apply to random quantities. So, in particular, for a random vector we always use (bold) lower case, and for a random matrix we always use bold upper-case, regardless of whether we are referring to (i) the unobserved random quantity or (ii) its observed value. It should always be clear from the context which of these two interpretations (i) or (ii) is appropriate.

1.1.1 Example datasets

Example 1.1. The football league table is an example of multivariate data. Here W = number of wins, D = number of draws, F = number of goals scored and A = number of goals conceded for four teams. In this example we have $p = 4$ variables $(W, D, F, A)^\top$ measured on $n = 4$ cases (teams).

Team	W	D	F	A
USA	1	2	4	3
England	1	2	2	1
Slovenia	1	1	3	3
Algeria	0	1	0	2

The data vector for the USA is

$$\mathbf{x}_1^\top = (1, 2, 4, 3)$$

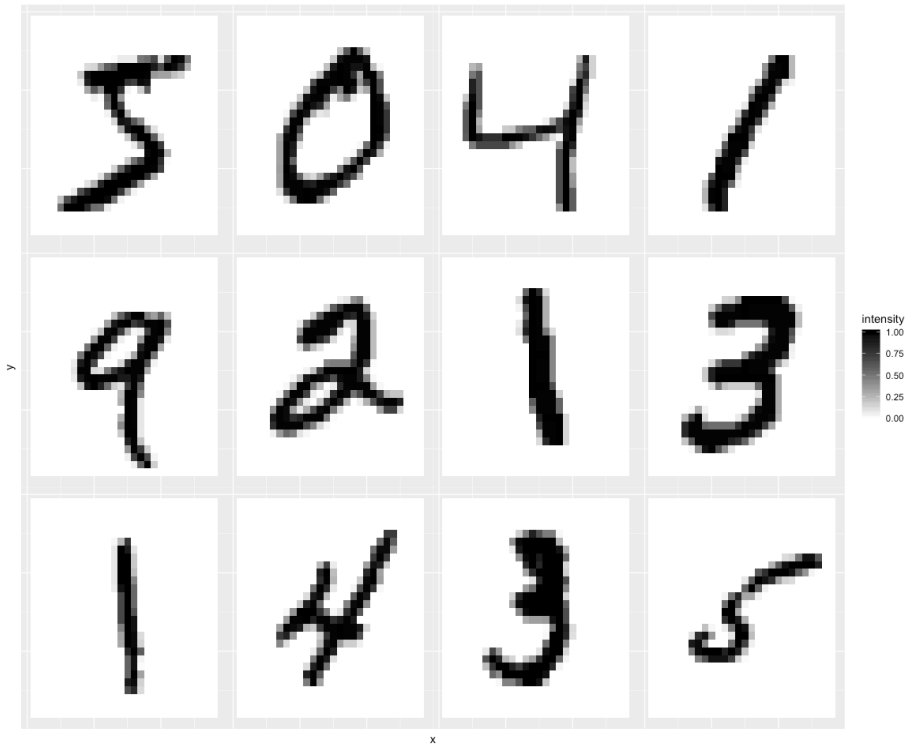
Example 1.2. Exam marks for a set of n students where P = mark in probability and S = mark in statistics. Let x_{ij} denote the jth variable measured on the ith subject.

Student	P	S
1	x_{11}	x_{12}
2	x_{21}	x_{22}
$\vdots$	$\vdots$	$\vdots$
n	x_{n1}	x_{n2}

Example 1.3. The `iris` dataset is a famous set of measurements collected on the sepal length and width, and the petal length and width, of 50 flowers for each of 3 species of iris (setosa, versicolor, and virginica). The dataset is built into R (try typing `iris` in R) and is often used to demonstrate multivariate statistical methods. For these data, $p = 5$, and $n = 150$.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica

Example 1.4. The MNIST dataset is a collection of handwritten digits that is widely used in statistics and machine learning to test algorithms. It contains 60,000 images of hand-written digits. Here are the first 12 images:



Each digit has been converted to a grid of 28×28 pixels, with a grayscale intensity level specified for each pixel. When we store these on a computer, we flatten each grid to a vector of length 784. So for this dataset, $n = 60,000$ and $p = 784$. As an example of what the data look like, the intensities (times 100) for the first image above are shown in the plot below:

Intensity*100

Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	1	7	7	7	49	53	69	10	65	100	97	50	0	0	0	0
7	0	0	0	0	0	0	0	0	12	14	37	60	67	99	99	99	99	99	88	67	99	95	76	25	0	0	0	0
8	0	0	0	0	0	0	0	19	93	99	99	99	99	99	99	99	99	98	36	32	32	22	15	0	0	0	0	0
9	0	0	0	0	0	0	0	7	86	99	99	99	99	99	78	71	97	95	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	31	61	42	99	99	80	4	0	17	60	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	5	0	60	99	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	55	99	75	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	4	75	99	27	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	14	95	88	63	42	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	32	94	99	99	47	10	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	73	99	99	59	11	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	36	99	99	73	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	99	98	25	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	51	72	99	99	81	1	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	15	58	90	99	99	99	98	71	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	9	45	87	99	99	99	99	79	31	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	9	26	84	99	99	99	99	78	32	1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	7	67	86	99	99	99	99	76	31	4	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	22	67	89	99	99	99	99	96	52	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	53	99	99	99	83	53	52	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Column

1.1.2 Aims of multivariate data analysis

The aim of multivariate statistical analysis is to answer questions such as:

- How can we visualise the data?
- What is the joint distribution of marks?
- Can we simplify the data? For example, we rank football teams using $3W + D$ and we rank students by their average module mark. Is this fair? Can we reduce the dimension in a better way?
- Can we use the data to discriminate, for example, between male and female students?
- Are the different iris species different shapes?
- Can we build a model to predict the intended digit from an image of someones handwriting? Or predict the species of iris from measurements of its sepal and petal?

We could just apply standard univariate techniques to each variable in turn, but this ignores possible dependencies between the variables which we must represent to draw valid conclusions.

What is the difference between MVA and standard linear regression?

- In standard linear regression we have a scalar response variable, y say, and

a vector of covariates, x, say. The focus of interest is on how knowledge of x influences the distribution of y (in particular, the mean of y). In contrast, in MVA the focus is a vector y, in which all the components of y are viewed as responses rather than covariates, possibly with additional covariate information x. We will discuss this further in Chapter **??**.

1.2 Exploratory data analysis (EDA)

A picture is worth a thousand words

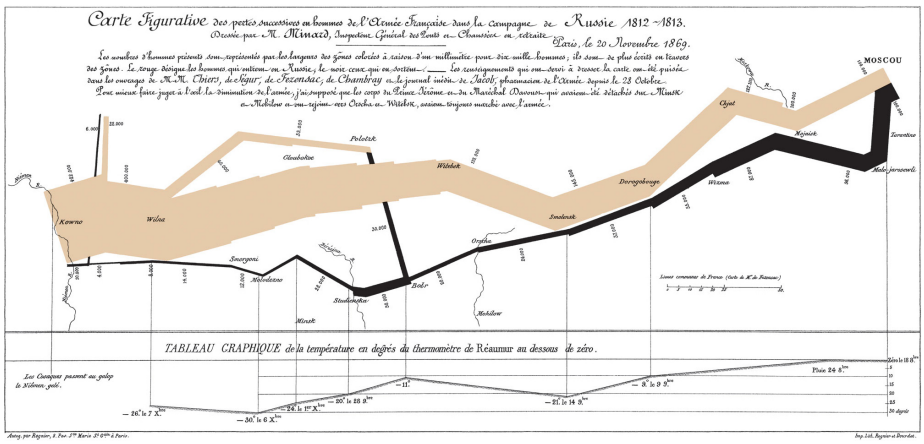


Figure 1.1: Charles Joseph Minard's famous map of Napoleon's 1812 invasion of Russian. It displays six types of data in two dimensions.

Before trying any form of statistical analysis, it is always a good idea to do some form of exploratory data analysis to understand the challenges presented by the data. As a minimum, this usually involves finding out whether each variable is continuous, discrete, or categorical, doing some basic visualization (plots), and perhaps computing a few summary statistics such as the mean and variance.

1.2.1 Data visualization

Visualising datasets before fitting any models can be extremely useful. It allows us to see obvious patterns and relationships,and may suggest a sensible form of analysis. With multivariate data, finding the right kind of plot is not always simple, and many different approaches have been proposed.

When $p = 1$ or $p = 2$ we can simply draw histograms and scatter plots (respectively) to view the distribution. For $p \geq 3$ the task is harder. One solution is a matrix of pair-wise scatter plots using the `pairs` command in R. The graph below shows the relationship between goals scored (F), goals against (A) and points (PT) for 20 teams during a recent Premiership season.

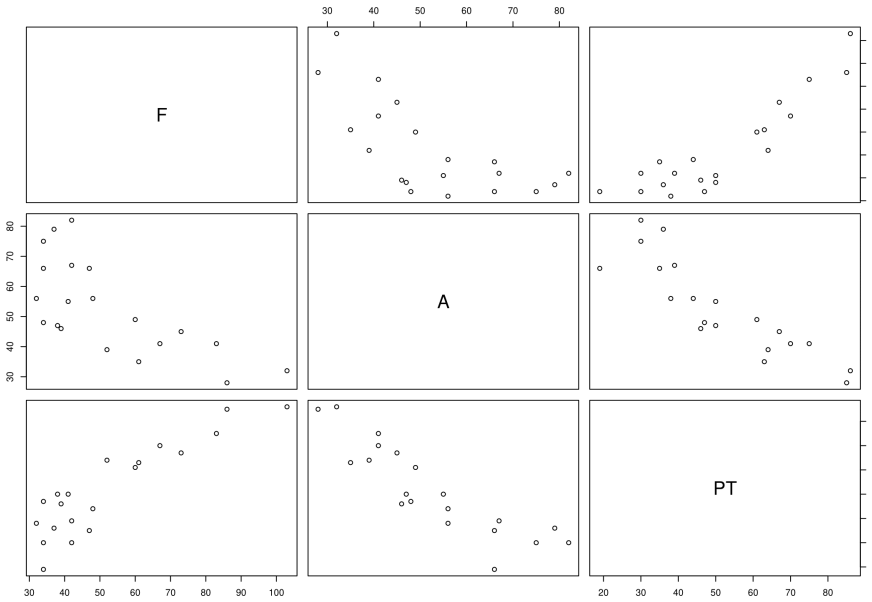


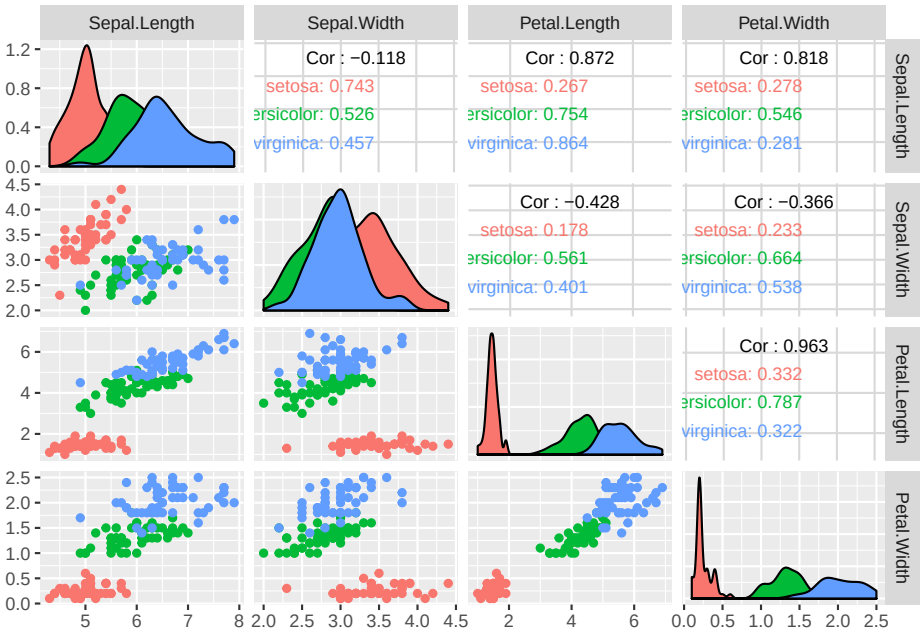
Figure 1.2: Scatter plots of goals for (F), goals against (A) and points (PT) for a recent Premier League Season

We can instantly see that points and goals scored are positively correlated, and that points and goals conceded (A) are negatively correlated (this is not a surprise of course).

R has a good basic plotting functionality. However, we will sometimes use packages that provide additional functionality. The first time you use a package you may need to install it. We can use `ggplot2` and `GGally` (which adds

functionality to ggplot2) to add colour and detail to pairs plots. For example

```
data(iris)
library(ggplot2)
library(GGally)
# pairs(iris) # - try the pairs command for comparison
ggpairs(iris, columns=1:4,  mapping=ggplot2::aes(colour = Species),
        upper = list(continuous = wrap("cor", size = 3)))  # fix the font size
```



This plot allows us to instantly see that there are clear differences between the three species of iris, at least when we look at the pairs plots. The benefit of adding colour in this case is that we can see the differences between the different species. Note how the sepal length and width are (weakly) negatively correlated across the entire dataset, but are positively correlated when we look at a single species at a time. We would have missed this information if we only used the `pairs` command (try it!).

Note that it is possible to miss key relationships when looking at *marginals* plots such as these, as they only show two variables at a time. More complex relationships between three or more variables will not be visible. It is difficult visualize data in three or more dimensions. Many different types of plot have been proposed (e.g. Google Chernoff faces). One approach is to use a *parallel line* plot

```
ggparcoord(iris, 1:4, groupColumn=5)
```

Each case is represented by a single line, and here we have the information shown for the four continuous variables. The fifth variable `Species` is a discrete factor, and is shown by colouring the lines.

If you not familiar with `ggplot2`, a nice introduction can be found here. Details about 'GGally can be found here. A good way to see the variety of plots that are possible, and to find code to create them, is to browse plot galleries such as those available here and here.

1.2.2 Summary statistics

It is often useful to report a small number of numerical summaries of the data. In univariate statistics we define the sample mean and sample variance of samples $x_1, \ldots, x_n$ to be

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad s_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

and for two samples, $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$, we define the sample covariance to be

$$s_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).$$

Analogous multivariate quantities can be defined as follows:

Definition 1.1. For a sample of n points, each containing p variables, $x_1, x_2, \ldots, x_n \in \mathbb{R}^p$, the **sample mean** and **sample covariance matrix** are

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1.1}$$

$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top \tag{1.2}$$

where $x_i \in \mathbb{R}^p$ denotes the p variables observed on the ith subject.

Note that

- $\bar{x} \in \mathbb{R}^p$. The jth entry in $\bar{x}$ is simply the (univariate) sample mean of the jth variable.
- $S \in \mathbb{R}^{p\times p}$. Note that the ij^{th} entry of S is s_{ij}, the sample covariance between variable i and variable j. The i^{th} diagonal element is the (univariate) sample variance of the ith variable.

- S is symmetric since $s_{ij} = s_{ji}$.
- an alternative formula for S is

$$S = \frac{1}{n}\left(\sum_{i=1}^{n} x_i x_i^\top\right) - \bar{x}\bar{x}^\top.$$

- We have divided by n rather than $n-1$ here, which gives the maximum likelihood estimator of the variance, rather than the unbiased variance estimator that is often used.

Definition 1.2. The **sample correlation matrix**, R, is the matrix with ij^{th} entry r_{ij} equal to the sample correlation between variables i and j, that is

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}s_{jj}}}.$$

Note that

- If $D = \text{diag}(\sqrt{s_{11}}, \ldots, \sqrt{s_{pp}})$, then

$$R = D^{-1}SD^{-1}$$

- R is symmetric
- the diagonal entries of R are exactly 1 (each variable is perfectly correlated with itself)
- $|r_{ij}| \leq 1$ for all i, j

Note that if we change the unit of measurement for the x_i's then S will change but R will not.

Definition 1.3. The **total variation** in a data set is usually measured by $\text{tr}(S)$ where tr() is the trace function that sums the diagonal elements of the matrix. That is,

$$\text{tr}(S) = s_{11} + s_{22} + ... + s_{pp}.$$

In other words, it is the sum of the univariate variances of each of the p variables.

1.3 Random vectors and matrices

Definition 1.4. The **population mean vector** of the random vector x is

$$\mu = \mathbb{E}(x).$$

The **population covariance matrix** of x is

$$\Sigma = \mathbb{V}\text{ar}(x) = \mathbb{E}\left((x - \mathbb{E}(x))(x - \mathbb{E}(x))^\top\right).$$

The **covariance** between x $(p \times 1)$ and y $(q \times 1)$ is

$$\mathbb{C}\text{ov}(x, y) = \mathbb{E}\left((x - \mathbb{E}(x))(y - \mathbb{E}(y))^\top\right).$$

Let A denote a $q \times p$ constant matrix, and let b a constant vector of size $q \times 1$. Expectation is a linear operator in the sense that

$$\mathbb{E}(Ax + b) = A\mathbb{E}(x) + b = A\mu + b.$$

The following properties follow:

- $\mathbb{V}\text{ar}(x) = \mathbb{E}(xx^\top) - \mu\mu^\top$.
- $\mathbb{V}\text{ar}(Ax + b) = A\Sigma A^\top$
- $\mathbb{C}\text{ov}(x, y) = \mathbb{E}(xy^\top) - \mathbb{E}(x)\mathbb{E}(y)^\top$.
- $\mathbb{C}\text{ov}(x, x) = \Sigma$.
- $\mathbb{C}\text{ov}(x, y) = \mathbb{C}\text{ov}(y, x)^\top$.
- $\mathbb{C}\text{ov}(Ax, By) = A\mathbb{C}\text{ov}(x, y)B^\top$
- If $p = q$ then

$$\mathbb{V}\text{ar}(x + y) = \mathbb{V}\text{ar}(x) + \mathbb{V}\text{ar}(y) + \mathbb{C}\text{ov}(x, y) + \mathbb{C}\text{ov}(y, x).$$

Finally, note that if x and y are independent (in which case I will write $x \perp\!\!\!\perp y$) then $\mathbb{C}\text{ov}(x, y) = \mathbf{0}_{p,q}$, i.e., a $p \times q$ matrix of zeros.

1.3.1 Estimators

The population mean vector μ and population covariance matrix Σ will usually be unknown. We can use data to **estimate** these quantities.

- The sample mean $\bar{x}$ is often used as an estimator of μ.
- The sample covariance matrix S is often used as an estimator of Σ.

Equation (1.1) gives an unbiased estimator of the sample mean. The sample covariance matrix (1.2) is a biased estimator of the population covariance matrix. An unbiased estimate is obtained by dividing by $n-1$ rather than n in Equation (1.2).

1.4 Computer tasks

If you haven't done so already, please download and install R and Rstudio. R is the programming language, and Rstudio is an integrated development environment that makes using R much more pleasurable. My advice is to always use Rstudio and never run code in R itself.

0. **For complete beginners:** For those who are completely new to R (or those who want a refresher), I recommend working through an online tutorial. This tutorial looks good, but contains more than you'll need.

1. **Warm-up:** The most important aspects of R to focus on for this module are

   - Basic plotting
   - Manipulation of matrices and data frames.

   Let's look at the `iris` dataset.

   - Can you plot the sepal length against the sepal width?

   We'll now do some exercises on data manipulation. Note that there are several ways to do basic data manipulation in R. You can use base R commands or if you prefer, you can use the dplyr commands which are part of the `tidyverse` packages. For example, to select columns, in base you can do:

```
iris[,2] # selects column 2
iris$Sepal.Width # selects the same column by name
```

or using dplyr you can do

```
library(dplyr)
select(iris, "Sepal.Width")
```

   - Can you select the column of the `iris` data that contains just the sepal length and add it to the sepal width?

   To select only certain rows of the data (i.e. to filter it), we can again use either base R or dplyr.

```
iris[iris[,3]<5,] # select all rows that have a petal length less than 5.
filter(iris, Petal.Length<5) # do the same thing using dplyr
```

- Can you now select all the rows of the `iris` data frame that are for species *setosa*? What is the mean petal length for these flowers?
- Can you select all the flowers that have a sepal length greater than 5? What is the proportion of each species of iris in this set?

A nice aspect of dplyr is that you can chain commmands together. So for example, to select the versicolour flowers with petal width less than 1.5, we can do

```
iris %>% filter(Species=='versicolor') %>% filter(Petal.Width<1.5)
```

- Can you select all the flowers that have a sepal length greater than 6, and a petal length less than 5? What is the proportion of each species in this set?

Note that `iris` is a data frame

```
is.data.frame(iris)
```

```
## [1] TRUE
```

which is a type of structure used in R. This is convenient for some tasks, but not for others. Let's first extract the four numerical columns and store them as a matrix X.

```
is.matrix(iris)
```

```
## [1] FALSE
```

```
X <- as.matrix(iris[,1:4])
is.matrix(X)
```

```
## [1] TRUE
```

- Select the 4 numerical columns and multiply the first column by 1, the second by 2, the third by 3, and the 4th by 4. One way to do this is by multiplying X by the diagonal matrix

```
diag(1:4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    2    0    0
## [3,]    0    0    3    0
## [4,]    0    0    0    4
```

2. The table below shows the module marks for 5 students on the modules G11PRB (P) and G11STA (S).

Student	P	S
A	41	63
B	72	82
C	46	38
D	77	57
E	59	85

- As an exercise, calculate the sample mean, sample covariance, sample correlation and total variation by hand.
- Now calculate these in R using `colMeans`, `cov`, and `cor`. These commands assume each column is a different variable, and each row a different observation.

```
library(dplyr)
Ex1 <- data.frame(
  Student=LETTERS[1:5],
  P = c(41,72,46,77,59),
  S = c(63,82,38,57,85)
  )

Ex1 %>% select_if(is.numeric) %>% colMeans
```

```
##  P  S
## 59 65
```

```
Ex1 %>% select_if(is.numeric) %>% cov
```

```
##       P     S
## P 246.5 116.0
## S 116.0 371.5
```

Note that by default R uses $n-1$ in the denominator for the variance and covariance commands, whereas we used n in our definition.

We will be using the `dplyr` R package to perform basic data manipulation in R. If you are unfamiliar with `dplyr`, you can read about it at https://dplyr.tidyverse.org/. The pipe command `%>%` is particularly useful for chaining together multiple commands.

You could compute the same quantities using more familiar commands by selecting the numerical columns:

```
colMeans(Ex1[,2:3])
```

```
##  P  S
## 59 65
```

```
cov(Ex1[,2:3])
```

```
##       P     S
## P 246.5 116.0
## S 116.0 371.5
```

- Can you compute the covariance matrix using the definition in Equation (1.2)?

3. The `mtcars` dataset is another built-in dataset in R. You can read about it by typing `?mtcars` in R. Note that some of the variables are factors. You can ensure R treats them as factors by using the following command to create a dataset where they are listed as factors:

```
mtcars2 <- within(mtcars, {
   vs <- factor(vs, labels = c("V", "S"))
   am <- factor(am, labels = c("automatic", "manual"))
   cyl  <- ordered(cyl)
   gear <- ordered(gear)
   carb <- ordered(carb)
})
```

`Work with the `mtcars2` dataframe when you use `ggplot2`.`

- Create some plots to explore the structure of this dataset using ggplot2.
- Try using the `pairs` command from base R and the `ggpairs` command from GGally.
- Try colouring the scatter plots accoring to whether the car is automatic or not. - Create another plot using colour to represent the number of gears.
- Find another type of plot from one of the plot galleries and try to create a similar plot with these data.

4. We can 100 generate samples from the multivariate normal distribution with mean vector

$$\mu = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and covariance matrix

$$\Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

as follows (you may need to install the R package `mvtnorm` first):

```
library(mvtnorm)
mu = c(1,0)
Sigma=matrix(c(2,1,1,2), nr=2)
X <- rmvnorm(n=100, mean=mu, sigma=Sigma)
```

- Compute the sample mean and covariance matrix of these samples.
- Generate a new sample dataset, X, and recompute the sample mean and covariance matrix. What do you notice?

- Try changing n, the number of samples (making it much larger say), and now recomputing the mean and covariance. What do you notice?

5. **Optional** Download the MNIST data from Moodle and load it into R.

```
load('mnist.rda')
```

This loads a list `mnist` that splits the data into two parts

```
mnist$train ## a training set of 60000 images
mnist$test ## a test set of 10000 images
```

Let's just look at the training set. This is also a list containing the image intensities and the image labels

```
mnist$train$x # image intensities
mnist$train$y # image labels
```

If we select just the first image we can see it is a vector of length 784 containing numbers between 0 and 1.

```
mnist$train$x[1,]
```

I've created a function to help you plot these images.

```
library(reshape2)
library(ggplot2)


plot.mnist <- function(im){
  #im[im<0]<-0 # set any negative intensities to zero
  #im[im>1]<-1 # set an intensities bigger than 1 to 1.


  if(is.vector(im)){ # a single image

    A<-matrix(im, nr=28, byrow=F)
    C<- melt(A, varnames = c("x", "y"), value.name = "intensity")
    p<-ggplot(C, aes(x = x, y = y, fill = intensity))+
      geom_tile(aes(fill=intensity))+
      scale_fill_gradient(low='white', high='black')+
      scale_y_reverse()+theme(
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        panel.spacing = unit(0, "lines"),
        axis.text = element_blank(),
        axis.ticks = element_blank()
      )
  }
  else{
```

```
    if (dim(im)[2]!=784){
      im = t(im)
    }
    n <- dim(im)[1]
    As <- array(im, dim = c(n, 28, 28))

    Cs<- melt(As, varnames = c("image","x", "y"), value.name = "intensity")
    p<-ggplot(Cs, aes(x = x, y = y, fill = intensity))+
      geom_tile(aes(fill=intensity))+
      scale_fill_gradient(low='white', high='black')+
      facet_wrap(~ image, nrow = floor(sqrt(n))+1, ncol = floor(sqrt(n))+1)+
      scale_y_reverse()+theme(
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        panel.spacing = unit(0, "lines"),
        axis.text = element_blank(),
        axis.ticks = element_blank()
      )

  }
  return(p)
}
```

- Use this command to plot the first 10 images from the MNIST training set.
- Select all the 5s from the MNIST training set. Plot a selection of these digits.

1.5 Exercises

1. Show that the two formulae for the population covariance matrix Σ are equivalent, i.e. show that

$$\Sigma = \mathbb{E}[(x-\mu)(x-\mu)^\top] = \mathbb{E}[xx^\top] - \mu\mu^\top.$$

2. Let $x_1, \ldots, x_n$ be a p-dimensional sample with mean $\bar{x}$ and sample covariance matrix S. Consider the transformation $y_i = Ax_i + c$ where A is a fixed $q \times p$ matrix and c is a fixed q-dimensional vector. Let T be the sample covariance matrix of $y_1, \ldots, y_n$. Show

- $\bar{y} = A\bar{x} + c$,
- $T = ASA^\top$.

Assuming now that x is a random vector with $\mathbb{E}(x) = \mu$, $\mathbb{V}\mathrm{ar}(x) = \Sigma$, $y = Ax + c$ with A and c as before, $\mathbb{E}(y) = \boldsymbol{\phi}$ and $\mathbb{V}\mathrm{ar}(y) = \boldsymbol{\Omega}$, what are

the population analogues of the results above?

3. A sample of size $n = 144$ produced the following summary statistics

$$\sum_{i=1}^{n} x_i = \begin{pmatrix} 392.2 \\ 1530.8 \end{pmatrix} \qquad \sum_{i=1}^{n} x_i x_i^\top = \begin{pmatrix} 1101.88 & 4305.17 \\ 4305.17 & 17120.88 \end{pmatrix}.$$

Calculate the sample mean, the sample covariance matrix and the sample correlation coefficient.

4. Let x and y be independent random p-dimensional vectors. Assuming that all relevant moments exist, show that for any real scalars α and β,

$$\mathbb{V}\,\mathrm{ar}(\alpha x + \beta y) = \alpha^2 \mathbb{V}\,\mathrm{ar}(x) + \beta^2 \mathbb{V}\,\mathrm{ar}(y).$$

What is the corresponding formula when x and y are not independent? Express your answer in terms of $\mathbb{V}\,\mathrm{ar}(x)$, $\mathbb{V}\,\mathrm{ar}(y)$ and $\mathbb{C}\,\mathrm{ov}(x, y)$.

Chapter 2

Review of linear algebra

Modern statistics and machine learning rely heavily upon linear algebra, nowhere more so than in multivariate statistics. In the first part of this chapter (sections 2.1 and 2.2) we review some concepts from linear algebra that will be needed throughout the module, including vector spaces, row and column spaces, the rank of a matrix, etc. Hopefully most of this will be familiar to you.

We then cover some basic details on inner-product or normed spaces in 2.3, which are vector spaces equipped with a concept of distance and angle. Finally, in Section 2.4 we will describe the centering matrix. Further details and proofs for this section will be tackled in the exercises in Section 2.6.

I do not provide proofs for many of the results stated in this chapter, but instead prove a small selection which I think it is useful to see. For a complete treatment of the linear algebra needed for this module, see the excellent book "Linear algebra and learning from data" by Gilbert Strang.

I have recorded videos on some (but not all) of the topics in these notes:

- Vector spaces
- Matrices
- Inner product spaces
- Orthogonal matrices
- Projection matrices

2.1 Basics

In this section, we recap some basic definitions and notation. Hopefully this material will largely be familiar to you.

2.1.1 Notation

The matrix $\mathbf{A}$ will be referred to in the following equivalent ways:

$$\begin{aligned}
\overset{n\times p}{\mathbf{A}} = \mathbf{A} &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{np} \end{bmatrix} \\
&= [a_{ij} : i = 1, \dots, m; j = 1, \dots, n] \\
&= (a_{ij}) \\
&= \begin{bmatrix} a_1^\top \\ \vdots \\ a_n^\top \end{bmatrix}
\end{aligned}$$

where the a_{ij} are the individual entries, and $a_i^\top = (a_{i1}, a_{i2}, \dots, a_{ip})$ is the i^{th} row.

A matrix of order 1×1 is called a *scalar*.

A matrix of order $n \times 1$ is called a *(column) vector*.

A matrix of order $1 \times p$ is called a *(row) vector*.

e.g. $\overset{n\times 1}{\mathbf{a}} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$ is a column vector.

The $n \times n$ *identity matrix* $\mathbf{I}_n$ has diagonal elements equal to 1 and off-diagonal elements equal to zero.

A *diagonal* matrix is an $n \times n$ matrix whose off-diagonal elements are zero. Sometimes we denote a diagonal matrix by $\text{diag}\{a_1, \dots, a_n\}$.

$$\mathbf{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{diag}\{1, 2, 3\} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

2.1.2 Elementary matrix operations

1. *Addition/Subtraction.* If $\overset{n\times p}{\mathbf{A}} = [a_{ij}]$ and $\overset{n\times p}{\mathbf{B}} = [b_{ij}]$ are given matrices then

$$\mathbf{A} + \mathbf{B} = [a_{ij} + b_{ij}] \qquad \text{and} \qquad \mathbf{A} - \mathbf{B} = [a_{ij} - b_{ij}].$$

2. *Scalar Multiplication.* If λ is a scalar and $\mathbf{A} = [a_{ij}]$ then

$$\lambda\mathbf{A} = [\lambda a_{ij}].$$

3. *Matrix Multiplication.* If $\overset{n\times p}{\mathbf{A}}$ and $\overset{p\times q}{\mathbf{B}}$ are matrices then $AB = \overset{n\times q}{\mathbf{C}} = [c_{ij}]$ where
$$c_{ij} = \sum_{k=1}^{p} a_{ik}b_{kj}, \qquad i = 1, \dots, n, \qquad j = 1, \dots, q.$$

4. *Matrix Transpose.* If $\overset{m\times n}{A} = [a_{ij} : i = 1, \dots, m; j = 1, \dots, n]$, then the transpose of A, written $A^\top$, is given by the $n \times m$ matrix
$$A^\top = [a_{ji} : j = 1, \dots, n; i = 1, \dots, m].$$
Note from the definitions that $(AB)^\top = \mathbf{B}^\top\mathbf{A}^\top$.

5. *Matrix Inverse.* The inverse of a matrix $\overset{n\times n}{\mathbf{A}}$ (if it exists) is a matrix $\overset{n\times n}{\mathbf{B}}$ such that $\mathbf{A}B = BA = \mathbf{I}_n$. We denote the inverse by $\mathbf{A}^{-1}$. Note that if $\mathbf{A}_1$ and $\mathbf{A}_2$ are both invertible, then $(\mathbf{A}_1\mathbf{A}_2)^{-1} = \mathbf{A}_2^{-1}\mathbf{A}_1^{-1}$.

6. *Trace.* The trace of a matrix $\overset{n\times n}{\mathbf{A}}$ is given by
$$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}.$$

Lemma 2.1. *For any matrices A $(n \times m)$ and B $(m \times n)$,*
$$tr(AB) = tr(BA).$$

7. The *determinant* of a square matrix $\overset{n\times n}{\mathbf{A}}$ is defined as
$$\det(\mathbf{A}) = \sum (-1)^{|\tau|} a_{1\tau(1)} \dots a_{n\tau(n)}$$
where the summation is taken over all permutations τ of $\{1, 2, \dots, n\}$, and we define $|\tau| = 0$ or 1 depending on whether τ can be written as an even or odd number of transpositions.

E.g. If $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, then $\det(\mathbf{A}) = a_{11}a_{22} - a_{12}a_{21}$.

Proposition 2.1. *Matrix $\overset{n\times n}{\mathbf{A}}$ is invertible if and only if* $\det(A) \neq 0$. *If A^{-1} exists then*
$$\det(A) = \frac{1}{\det(A^{-1})}$$

Proposition 2.2. *For any matrices $\overset{n\times n}{\mathbf{A}}$, $\overset{n\times n}{\mathbf{B}}$, $\overset{n\times n}{\mathbf{C}}$ such that $\mathbf{C} = \mathbf{AB}$,*
$$det(\mathbf{C}) = det(\mathbf{A}) \cdot det(\mathbf{B}).$$

2.1.3 Special matrices

Definition 2.1. An $n \times n$ matrix A is symmetric if

$$A = A^\top.$$

An $n \times n$ symmetric matrix A is **positive-definite** if

$$x^\top Ax > 0 \text{ for all } x \in \mathbb{R}^n, x \neq 0$$

and is **positive semi-definite** if

$$x^\top Ax \geq 0 \text{ for all } x \in \mathbb{R}^n.$$

A is **idempotent** if $A^2 = A$.

2.1.4 Vector Differentiation

Consider a real-valued function $f : \mathbb{R}^p \to \mathbb{R}$ of a vector variable $x = (x_1, \ldots, x_p)^\top$. Sometimes we will want to differentiate f. We define the partial derivative of $f(x)$ with respect to x to be the vector of partial derivatives, i.e.

$$\frac{\partial f}{\partial x}(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ .. \\ .. \\ .. \\ \frac{\partial f}{\partial x_p}(x) \end{bmatrix} \tag{2.1}$$

The following examples can be worked out directly from the definition (2.1), using the chain rule in some cases.

Example 2.1. If $f(x) = a^\top x$ where $a \in \mathbb{R}^p$ is a constant vector, then

$$\frac{\partial f}{\partial x}(x) = a.$$

Example 2.2. If $f(x) = (x-a)^\top A(x-a)$ for a fixed vector $a \in \mathbb{R}^p$ and A is a symmetric constant $p \times p$ matrix, then

$$\frac{\partial f}{\partial x}(x) = 2A(x-a).$$

Example 2.3. Suppose that $g : \mathbb{R} \to \mathbb{R}$ is a differentiable function with derivative g'. Then, using the chain rule for partial derivatives,

$$\frac{\partial g(a^\top x)}{\partial x} = g'(a^\top x)\frac{\partial}{\partial x}\left\{a^\top x\right\} = g'(a^\top x)a.$$

Example 2.4. If f is defined as in Example 2.2 and g is as in Example 2.3 then, using the chain rule again,

$$\frac{\partial}{\partial x} g\{f(x)\} = g'\{f(x)\}\frac{\partial f}{\partial x}(x) = 2g'\{(x-a)^\top A(x-a)\}A(x-a).$$

If we wish to find a maximum or minimum of $f(x)$ we should search for stationary points of f, i.e. solutions to the system of equations

$$\frac{\partial f}{\partial x}(x) \equiv \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ .. \\ .. \\ .. \\ \frac{\partial f}{\partial x_p}(x) \end{bmatrix} = \mathbf{0}_p.$$

Definition 2.2. The **Hessian** matrix of f is the $p \times p$ matrix of second derivatives.

$$\frac{\partial^2 f}{\partial x \partial x^\top}(x) = \left\{ \frac{\partial^2 f(x)}{\partial x_j \partial x_k} \right\}_{j,k=1}^{p}.$$

The nature of a stationary point is determined by the Hessian

If the Hessian is positive (negative) definite at a stationary point x, then the stationary point is a minimum (maximum).

If the Hessian has both positive and negative eigenvalues at x then the stationary point will be a *saddle point*.

2.2 Vector spaces

It will be useful to talk about **vector spaces**. These are sets of vectors that can be added together, or multiplied by a scalar. You should be familiar with these from your undergraduate degree. We don't provide a formal definition here, but you can think of a real vector space V as a set of vectors such that for any $v_1, v_2 \in V$ and $\alpha_1, \alpha_2 \in \mathbb{R}$, we have

$$\alpha_1 v_1 + \alpha_2 v_2 \in V$$

i.e., vector spaces are closed under addition and scalar multiplication.

Example 2.5. Euclidean space in p dimensions, $\mathbb{R}^p$, is a vector space. If we add any two vectors in $\mathbb{R}^p$, or multiply a vector by a real scalar, then the resulting vector also lies in $\mathbb{R}^p$.

A subset $U \subset V$ of a vector space V is called a vector **subspace** if U is also a vector space.

Example 2.6. Let $V = \mathbb{R}^2$. Then the sets

$$U_1 = \left\{ \begin{pmatrix} a \\ 0 \end{pmatrix} : a \in \mathbb{R} \right\}, \text{ and } \quad U_2 = \left\{ a \begin{pmatrix} 1 \\ 1 \end{pmatrix} : a \in \mathbb{R} \right\}$$

are both subspaces of V.

2.2.1 Linear independence

Definition 2.3. Vectors $\overset{n\times 1}{\mathbf{x}}_1, \ldots, \overset{n\times 1}{\mathbf{x}}_p$ are said to be **linearly dependent** if there exist scalars $\lambda_1, \ldots, \lambda_p$ not all zero such that

$$\lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \cdots + \lambda_p \mathbf{x}_p = \mathbf{0}.$$

Otherwise, these vectors are said to be **linearly independent**.

Definition 2.4. Given a set of vectors $S = \{s_1, \ldots, s_n\}$, the **span** of S is the smallest vector space containing S or equivalently, is the set of all linear combinations of vectors from S

$$\text{span}(S) = \left\{ \sum_{i=1}^{k} \alpha_i s_i \mid k \in \mathbb{N}, \alpha_i \in \mathbb{R}, s_i \in S \right\}$$

Definition 2.5. A **basis** of a vector space V is a set of linearly independent vectors in V that span V.

Example 2.7. Consider $V = \mathbb{R}^2$. Then the following are both bases for V:

$$B_1 = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

$$B_2 = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right\}$$

Definition 2.6. The **dimension** of a vector space is the number of vectors in its basis.

2.2.2 Row and column spaces

We can think about the matrix-vector multiplication Ax in two ways. The usual way is as the inner product between the rows of A and x.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 + 2x_2 \\ 3x_1 + 4x_2 \\ 5x_1 + 6x_2 \end{pmatrix}$$

But a better way to think of Ax is as a linear combination of the columns of A.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1 \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} + x_2 \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$$

Definition 2.7. The **column space** of a $n \times p$ matrix A is the set of all linear combinations of the columns of A:

$$\mathcal{C}(A) = \{Ax : x \in \mathbb{R}^p\} \subset \mathbb{R}^n$$

For

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

we can see that the column space is a 2-dimensional plane in $\mathbb{R}^3$. The matrix B has the same column space as A

$$B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 7 & 10 \\ 5 & 6 & 11 & 16 \end{pmatrix}$$

The number of linearly independent columns of A is called the **column rank** of A, and is equal to the dimension of the column space of $\mathcal{C}(A)$. The **column rank** of A and B is 2.

The **row space** of A is defined to be the column space of $A^\top$, and the **row rank** is the number of linearly independent rows of A.

Theorem 2.1. *The **row rank** of of a matrix equals the **column rank**.*

Thus we can simply refer to the **rank** of the matrix.

Proof. The proof of this theorem is very simple. Let C be an $n \times r$ matrix (where $r = \text{rank}(A)$) with columns chosen to be a set of r linearly independent columns from A. Then we know each column of A can be written as a linear combination of the columns of C, i.e.

$$A = CR.$$

The dimension of R must be $r \times p$. But now we can see that the rows of A are formed by a linear combination of the rows of R. Thus the row rank of A is at most r (=the column rank of A). This holds for any matrix, so is true for $A^\top$: namely row-rank$(A^\top) \leq$ column-rank$(A^\top)$. But the row space of $A^\top$ equals $\mathcal{C}(A)$, thus proving the theorem! □

Corollary 2.1. *The rank of an $n \times p$ matrix is at most* $\min(n, p)$.

Example 2.8.

$$B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Example 2.9.

$$D = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

So the rank of D is 1.

2.2.3 Linear transformations

We can view an $n \times p$ matrix A as a linear map between two vector spaces:

$$\begin{aligned} A : \ \mathbb{R}^p &\to \mathbb{R}^n \\ x &\mapsto Ax \end{aligned}$$

The **image** of A is precisely the column space of A:

$$\mathrm{Im}(A) = \{Ax : x \in \mathbb{R}^p\} = \mathcal{C}(A) \subset \mathbb{R}^n$$

The **kernel** of A is the set of vectors mapped to zero:

$$\mathrm{Ker}(A) = \{x : Ax = 0\} \subset \mathbb{R}^p$$

and is sometimes called the **null-space** of A and denoted $\mathcal{N}(A)$.

Theorem 2.2. *The **rank-nullity** theorem says if V and W are vector spaces, and $A : V \to W$ is a linear map, then*

$$\dim \mathrm{Im}(A) + \dim \mathrm{Ker}(A) = \dim V$$

If we're thinking about matrices, then $\dim \mathcal{C}(A) + \dim \mathcal{N}(A) = p$, or equivalently that
$\mathrm{rank}(A) + \dim \mathcal{N}(A) = p$.

We've already said that the row space of A is $\mathcal{C}(A^\top)$. The left-null space is $\{x \in \mathbb{R}^n : x^\top A = 0\}$ or equivalently $\{x \in \mathbb{R}^n : A^\top x = 0\} = \mathcal{N}(A^\top)$. And so by the rank-nullity theorem we must have

$$n = \dim \mathcal{C}(A^\top) + \dim \mathcal{N}(A^\top) = \mathrm{rank}(A) + \dim \mathrm{Ker}(A^\top).$$

Example 2.10. Consider again the matrix $D : \mathbb{R}^3 \to \mathbb{R}^2$

$$D = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

We have already seen that

$$\mathcal{C}(D) = \text{span}\left\{\begin{pmatrix} 1 \\ 2 \end{pmatrix}\right\}$$

and so $\dim \mathcal{C}(D) = \text{rank}(D) = 1$. The kernel, or null-space, of D is the set of vectors for which $Dx = 0$, i.e.,

$$x_1 + 2x_2 + 3x_3 = 0$$

This is a single equation with three unknowns, and so there must be a plane of solutions. We need two linearly independent vectors in this plane to describe it. Convince yourself that

$$\mathcal{N}(D) = \text{span}\left\{\begin{pmatrix} 0 \\ 3 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}\right\}$$

So we have

$$\dim \mathcal{C}(D) + \dim \mathcal{N}(D) = 1 + 2 = 3$$

as required by the rank-nullity theorem.

If we consider $D^\top$, we already know $\dim \mathcal{C}(D) = 1$ (as row-rank=column rank), and the rank-nullity theorem tells us that the dimension of the null space of $D^\top$ must be $2 - 1 = 1$. This is easy to confirm as $D^\top x = 0$ implies

$$x_1 + 2x_2 = 0$$

which is a line in $\mathbb{R}^2$

$$\mathcal{N}(D^\top) = \text{span}\left\{\begin{pmatrix} -2 \\ 1 \end{pmatrix}\right\}$$

Question: When does a square matrix A have an inverse?

- Precisely when the kernel of A contains only the zero vector, i.e., has dimension 0. In this case the column space of A is the original space, and A is surjective and so must have an inverse. A simpler way to determine if A has an inverse is to consider its determinant.

Question: Suppose we are given a $n \times p$ matrix A, and a n-vector y. When does

$$Ax = y$$

have a solution?

- When y is in the column space of A,

$$y \in \mathcal{C}(A)$$

Question: When is the answer unique?

- Suppose x and x' are both solutions with $x \neq x'$. We can write $x' = x + u$ for some vector u and note that

$$y = Ax' = Ax + Au = y + Au$$

and so $Au = 0$, i.e., $u \in \mathcal{N}(A)$. So there are multiple solutions when the null-space of A contains more than the zero vector. If the dimension of $\mathcal{N}(A)$ is one, there is a line of solutions. If the dimension is two, there is a plane of solutions, etc.

2.3 Inner product spaces

2.3.1 Distances, and angles

Vector spaces are not particularly interesting from a statistical point of view until we equip them with a sense of geometry, i.e. distance and angle.

Definition 2.8. A real **inner product space** $(V, \langle \cdot, \cdot \rangle)$ is a real vector space V equipped with a map

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$$

such that

1. $\langle \cdot, \cdot \rangle$ is a linear map in both arguments:

$$\langle \alpha v_1 + \beta v_2, u \rangle = \alpha \langle v_1, u \rangle + \beta \langle v_2, u \rangle$$

for all $v_1, v_2, u \in V$ and $\alpha, \beta \in \mathbb{R}$. 2. $\langle \cdot, \cdot \rangle$ is symmetric in its arguments: $\langle v, u \rangle = \langle u, v \rangle$ for all $u, v \in V$ 3. $\langle \cdot, \cdot \rangle$ is positive definite: $\langle v, v \rangle \geq 0$ for all $v \in V$ with equality if and only if $v = \mathbf{0}$.

An inner product provides a vector space with the concepts of

- **distance**: for all $v \in V$ define the **norm** of v to be

$$||v|| = \langle v, v \rangle^{\frac{1}{2}}$$

Thus any inner-product space $(V, \langle \cdot, \cdot \rangle)$ is also a normed space $(V, || \cdot ||)$, and a metric space $(V, d(x, y) = ||x - y||)$.

- **angle**: for $u, v \in V$ we define the angle between u and v to be θ where

$$\langle u, v \rangle = ||u||.||v|| \cos\theta$$
$$\implies \theta = \cos^{-1}\left(\frac{\langle u, v \rangle}{||u||\ ||v||}\right)$$

We will primarily be interested in the concept of **orthogonality**. We say $u, v \in V$ are orthogonal if

$$\langle u, v \rangle = 0$$

i.e., the *angle* between them is $\frac{\pi}{2}$.

If you have done any functional analysis, you may recall that a Hilbert space is a *complete* inner-product space, and a Banach space is a complete normed space. This is an applied module, so we will skirt much of the technical detail, but note that some of the proofs formally require us to be working in a Banach or Hilbert space. We will not concern ourselves with such detail.

Example 2.11. We will mostly be working with the Euclidean vector spaces $V = \mathbb{R}^n$, in which we use the *Euclidean* inner product

$$\langle u, v \rangle = u^\top v$$

sometimes called the **scalar** or **dot product** of u and v. Sometimes this gets weighted by a matrix so that

$$\langle u, v \rangle_Q = u^\top Q v.$$

The norm associated with the dot product is the square root of the sum of squared errors, denoted by $||\cdot||_2$. The **length** of u is then

$$||u||_2 = \sqrt{u^\top u} = \left(\sum_{i=1}^{n} u_i^2 \right)^{\frac{1}{2}} \geq 0.$$

Note that $||u||_2 = 0$ if and only if $u = \mathbf{0}_n$ where $\overset{n\times 1}{\mathbf{0}}_n = (0, 0, \ldots, 0)^\top$.

We say u is orthogonal to v if $u^\top v = 0$. For example, if

$$u = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \text{ and } v = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$$

then

$$||u||_2 = \sqrt{5} \text{ and } u^\top v = 0.$$

We will write $u \perp v$ if u is orthogonal to v.

Definition 2.9. p-norm: The subscript 2 hints at a wider family of norms. We define the L_p norm to be

$$||v||_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{\frac{1}{p}}.$$

2.3.2 Orthogonal matrices

Definition 2.10. A **unit vector v** is a vector satisfying $||\mathbf{v}|| = 1$, i.e., it is a vector of length 1. Vectors u and v are orthonormal if

$$||u|| = ||v|| = 1 \text{ and } \langle u, v \rangle = 0.$$

An $n \times n$ matrix $\mathbf{Q}$ is an **orthogonal matrix** if

$$\mathbf{Q}Q^\top = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_n.$$

Equivalently, a matrix $\mathbf{Q}$ is orthogonal if $\mathbf{Q}^{-1} = \mathbf{Q}^\top$.

If $\mathbf{Q} = [q_1, \dots, q_n]$ is an orthogonal matrix, then the columns $q_1, \dots, q_n$ are mutually **orthonormal** vectors, i.e.

$$q_j^\top q_k = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k. \end{cases}$$

Lemma 2.2. *Let Q be a $n \times p$ matrix and suppose $Q^\top Q = \mathbf{I}_p$, where $\mathbf{I}_p$ is the $p \times p$ identity matrix. If Q is a square matrix ($n = p$), then $QQ^\top = \mathbf{I}_p$. If Q is not square ($n \neq p$), then $QQ^\top \neq I_n$.*

Proof. Suppose $n = p$, and think of Q as a linear map""

$$\begin{aligned} Q :& \mathbb{R}^n \to \mathbb{R}^n \\ & v \mapsto Qv \end{aligned}$$

By the rank-nullity theorem,

$$\dim \operatorname{Ker}(Q) + \dim \operatorname{Im}(Q) = n$$

and because Q has a left-inverse, we must have $\dim \operatorname{Ker}(Q) = 0$, as otherwise $Q^\top$ would have to map from a vector space of dimension less than n to $\mathbb{R}^n$. So Q is of full rank, and thus must also have a right inverse, B say, with $QB = \mathbf{I}_n$. If we left multiply by $Q^\top$ we get

$$\begin{aligned} QB &= \mathbf{I}_n \\ Q^\top QB &= Q^\top \\ \mathbf{I}_n B &= Q^\top \\ B &= Q^\top \end{aligned}$$

and so we have that $Q^{-1} = Q^\top$.

Now suppose Q is $n \times p$ with $n \neq p$. Then as $Q^\top Q = \mathbf{I}_{p \times p}$, we must have $\operatorname{tr}(Q^\top Q) = p$. This implies that

$$\operatorname{tr}(QQ^\top) = \operatorname{tr}(Q^\top Q) = m$$

and so we cannot have $QQ^\top = \mathbf{I}_n$ as $\operatorname{tr} \mathbf{I}_n = n$. □

Corollary 2.2. *If $q_1, \ldots, q_n$ are mutually orthogonal $n \times 1$ unit vectors then*

$$\sum_{i=1}^{n} q_i q_i^\top = \mathbf{I}_n.$$

Proof. Let Q be the matrix with i^{th} column q_i

$$Q = \begin{pmatrix} | & & | \\ q_1 & \ldots & q_n \\ | & & | \end{pmatrix}.$$

Then $Q^\top Q = \mathbf{I}_n$, and Q is $n \times n$. Thus by Lemma 2.2, we must also have $QQ^\top = \mathbf{I}_n$ and if we think about matrix-matrix multiplication as columns times rows (c.f. section 3.1), we get

$$\mathbf{I}_n = QQ^\top = \begin{pmatrix} | & & | \\ q_1 & \ldots & q_n \\ | & & | \end{pmatrix} \begin{pmatrix} - & q_1^\top & - \\ & \vdots & \\ - & q_n^\top & - \end{pmatrix} = \sum_{i=1}^{n} q_i q_i^\top$$

as required.

□

2.3.3 Projections

Definition 2.11. $\overset{n\times n}{P}$ is a *projection* matrix if

$$P^2 = P$$

i.e., if it is idempotent.

View P as a map from a vector space W to itself. Let $U = \mathrm{Im}(P)$ and $V = \mathrm{Ker}(P)$ be the image and kernel of P.

Proposition 2.3. *We can write $w \in W$ as the sum of $u \in U$ and $v \in V$.*

Proof. Let $w \in W$. Then

$$w = \mathbf{I}_n w = (\mathbf{I} - P)w + Pw$$

Now $Pw \in \mathrm{Im}(P)$ and $(\mathbf{I} - P)w \in \mathrm{Ker}(P)$ as

$$P(\mathbf{I} - P)w = (P - P^2)w = 0.$$

□

Proposition 2.4. *If $\overset{n\times n}{P}$ is a projection matrix then $\mathbf{I}_n - P$ is also a projection matrix.*

The kernel and image of $\mathbf{I} - P$ are the image and kernel (respectively) of P:

$$\mathrm{Ker}(\mathbf{I} - P) = U = \mathrm{Im}(P)$$
$$\mathrm{Im}(\mathbf{I} - P) = V = \mathrm{Ker}(P).$$

2.3.3.1 Orthogonal projection

We are mostly interested in **orthogonal** projections.

Definition 2.12. If W is an inner product space, and U is a subspace of W, then the orthogonal projection of $w \in W$ onto U is the unique element $u \in U$ that minimizes

$$||w - u||.$$

In other words, the orthogonal projection of w onto U is the *best possible approximation* of w in U.

As above, we can split W into U and its orthogonal compliment

$$U^{\perp} = \{x \in W : \langle x, u \rangle = 0\}$$

i.e., $W = U \oplus U^{\perp}$ so that any $w \in W$ can be written as $w = u + v$ with $u \in U$ and $v \in U^{\perp}$.

Proposition 2.5. *If $\{u_1, \dots, u_k\}$ is a basis for U, then the orthogonal projection matrix (i.e., the matrix that projects $w \in W$ onto U) is*

$$P_U = A(A^{\top}A)^{-1}A^{\top}$$

where $A = [u_1 \ \dots \ u_k]$ is the matrix with columns given by the basis vectors.

Proof. We need to find $u = \sum \lambda_i u_i = A\lambda$ that minimizes $||w - u||$.

$$\begin{aligned} ||w-u||^2 &= \langle w-u, w-u \rangle \\ &= w^{\top}w - 2u^{\top}w + u^{\top}u \\ &= w^{\top}w - 2\lambda^{\top}A^{\top}w + \lambda^{\top}A^{\top}A\lambda. \end{aligned}$$

Differentiating with respect to λ and setting equal to zero gives

$$0 = -2A^{\top}w + 2A^{\top}A\lambda$$

and hence

$$\lambda = (A^{\top}A)^{-1}A^{\top}w.$$

The orthogonal projection of w is hence

$$A\lambda = A(A^{\top}A)^{-1}A^{\top}w$$

and the projection matrix is

$$P_U = A(A^{\top}A)^{-1}A^{\top}.$$

□

Notes:

1. If $\{u_1, \ldots, u_k\}$ is an orthonormal basis for U then $A^\top A = \mathbf{I}$ and $P_U = AA^\top$. We can then write
$$P_U w = \sum_i (u_i^\top w) u_i$$
and
$$P_U = \sum_{i=1}^{k} u_i u_i^\top.$$
Note that if $U = W$ (so that P_U is a projection from W onto W, i.e., the identity), then A is a square matrix ($n \times n$) and thus $A^\top A = \mathbf{I}_n \implies AA^\top$ and thus $P_U = \mathbf{I}_n$ as required. The coordinates (with respect to the orthonormal basis $\{u_1, \ldots, u_k\}$) of a point w projected onto U are $A^\top w$.
2. $P_U^2 = P_U$, so P_U is a projection matrix in the sense of definition 2.11.
3. P_U is symmetric ($P_U^\top = P_U$). This is true for orthogonal projection matrices, but not in general for projection matrices.

Example 2.12. Consider the vector space $\mathbb{R}^2$ and let $u = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. The projection of $v \in \mathbb{R}^2$ onto u is given by $(v^\top u)u$. So for example, if $v = (2,\ 1)^\top$, then its projection onto u is
$$P_U v = \frac{3}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$
Alternatively, if we treat u as a basis for U, then the coordinate of $P_U v$ with respect to the basis is 3. To check this, draw a picture!

2.3.3.2 Geometric interpretation of linear regresssion

Consider the linear regression model
$$y = X\beta + e$$
where $y \in \mathbb{R}^n$ is the vector of observations, X is the $n \times p$ design matrix, β is the $p \times 1$ vector of parameters that we wish to estimate, and e is a $n \times 1$ vector of zero-mean errors.

Least-squares regression tries to find the value of $\beta \in \mathbb{R}^p$ that minimizes the sum of squared errors, i.e., we try to find β to minimize
$$||y - X\beta||_2$$

We know that $X\beta$ is in the column space of X, and so we can see that linear regression aims to find the *orthogonal projection* onto $\mathcal{C}(X)$.
$$P_U y = \arg \min_{y' : y' \in \mathcal{C}(X)} ||y - y'||_2.$$

By Proposition 2.5 this is

$$P_U y = X(X^\top X)^{-1}X^\top y = \hat{y}$$

which equals the usual prediction obtained in linear regression ($\hat{y}$ are often called the fitted values). We can also see that the choice of β that specifies this point in $\mathcal{C}(X)$ is

$$\hat{\beta} = (X^\top X)^{-1}X^\top y$$

which is the usual least-squares estimator.

2.4 The Centering Matrix

The **centering matrix** will be play an important role in this module, as we will use it to remove the column means from a matrix (so that each column has mean zero), *centering* the matrix.

Definition 2.13. The **centering matrix** is

$$H = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top. \tag{2.2}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix, and $\mathbf{1}_n$ is an $n \times 1$ column vector of ones.

You will be asked to prove the following results about H in the exercises:

1. The matrix H is a projection matrix, i.e. $H^\top = H$ and $H^2 = H$.
2. Writing $\mathbf{0}_n$ for the $n\times 1$ vector of zeros, we have $H\mathbf{1}_n = \mathbf{0}_n$ and $\mathbf{1}_n^\top H = \mathbf{0}_n^\top$. In words: the sum of each row and each column of H is 0.
3. If $x = (x_1, \dots, x_n)^\top$, then $Hx = x - \bar{x}\mathbf{1}_n$ where $\bar{x} = n^{-1}\sum_{i=1}^n x_i$. I.e., H subtracts the mean $\bar{x}$ from x.
4. With x as in 3., we have

   $$x^\top Hx = \sum_{i=1}^n (x_i - \bar{x})^2,$$

   and so

   $$\frac{1}{n}x^\top Hx = \frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})^2 = \hat{\sigma}^2,$$

   where $\hat{\sigma}^2$ is the sample variance.
5. If

   $$X = \begin{bmatrix} - & x_1^\top & - \\ & \vdots & \\ - & x_n^\top & - \end{bmatrix} = [x_1, \dots, x_n]^\top$$

   is an $n \times p$ data matrix containing data points $x_1, \dots, x_n \in \mathbb{R}^p$, then

   $$HX = \begin{bmatrix} - & (x_1 - \bar{x})^\top & - \\ - & (x_2 - \bar{x})^\top & - \\ & \vdots & \\ - & (x_n - \bar{x})^\top & - \end{bmatrix} = [x_1 - \bar{x}, \dots, x_n - \bar{x}]^\top$$

where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \in \mathbb{R}^p$$

is the p-dimensional sample mean of $x_1, \ldots, x_n \in \mathbb{R}^p$. In words, H has subtracted the column mean from each column of X.

6. With X as in 5.

$$\frac{1}{n}X^\top HX = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top = S,$$

where S is the sample covariance matrix.

7. If $A = (a_{ij})_{i,j=1}^n$ is a symmetric $n \times n$ matrix, then

$$B = HAH = A - \mathbf{1}_n\bar{a}_+^\top - \bar{a}_+\mathbf{1}_n^\top + \bar{a}_{++}\mathbf{1}_n\mathbf{1}_n^\top,$$

or, equivalently,

$$b_{ij} = a_{ij} - \bar{a}_{i+} - \bar{a}_{+j} + \bar{a}_{++}, \qquad i, j = 1, \ldots, n,$$

where

$$\bar{a}_+ \equiv (\bar{a}_{1+}, \ldots, \bar{a}_{n+})^\top = \frac{1}{n}A\mathbf{1}_n,$$

$\bar{a}_{+j} = \bar{a}_{j+}$, for , $j = 1, \ldots, n$,, and , $\bar{a}_{++} = n^{-2}\sum_{i,j=1}^n a_{ij}$.

Note that Property 3. is a special case of Property 5., and Property 4. is a special case of Property 6. However, it is useful to see these results in the simpler scalar case before moving onto the general matrix case.

2.5 Computer tasks

This Chapter's computer tasks are short and sweet, as the focus has primarily been on the mathematics. Tasks for later chapters will be more challenging.

0. Let's consider some basic matrix computations in R. First, we show how to do matrix multiplication and addition

```
a=c(3,1,1,6)                        # define a column  vector a
b=c(5,6,2,8)                        # define a  vector b
A=matrix(a,nrow=2,byrow=TRUE)       # use a to define a matrix A
# Note that by default R fills a matrix by column. You have to explictly
# ask for it to be filled by row.
A
```

```
##      [,1] [,2]
## [1,]    3    1
## [2,]    1    6
```

```
B=matrix(b,nrow=2,byrow=TRUE)     # use b to define a matrix B
B
```

```
##      [,1] [,2]
## [1,]    5    6
## [2,]    2    8
```

```
A%*%B                             # use %*% to multiply two matrices
```

```
##      [,1] [,2]
## [1,]   17   26
## [2,]   17   54
```

```
                                  # together in the usual sense
A+B                               # add
```

```
##      [,1] [,2]
## [1,]    8    7
## [2,]    3   14
```

```
dim(A)                            # prints the dimension of a matrix.
```

```
## [1] 2 2
```

Multiplication of a matrix by a scalar is easy - but be careful if you use the * for two square matrices, as R will do element-wise multiplication

```
3*A
```

```
##      [,1] [,2]
## [1,]    9    3
## [2,]    3   18
```

```
A*B # compare with A%*%B
```

```
##      [,1] [,2]
## [1,]   15    6
## [2,]    2   48
```

Note that R won't let you multiply matrices that are not conformable (i.e. not the right shape).

The usual Euclidean inner product is just matrix multiplication

```
t(a) %*% b  # t() transposes a matrix
```

```
##      [,1]
## [1,]   71
```

The inverse, determinant, and trace of a matrix are computed as follows:

```
solve(A) # the inverse
```

```
##              [,1]        [,2]
## [1,]  0.35294118 -0.05882353
## [2,] -0.05882353  0.17647059
```

```
det(A)
```

```
## [1] 17
```

```
sum(diag(A)) # the trace is the sum of the diagonal elements of a matrix.
```

```
## [1] 9
```

Note that numerical errors will start to appear quite quickly. For example, the following should return the identity matrix. The result is very close to the identity, but not exactly equal to it. With larger matrices, numerical errors can be worse and appear alarmingly quickly.

```
A%*%solve(A)
```

```
##               [,1] [,2]
## [1,] 1.000000e+00    0
## [2,] 5.551115e-17    1
```

1. Solve the linear system for x using R.

$$\begin{pmatrix} 3 & 2 & 1 \\ 2 & 1 & 3 \\ 1 & 3 & 2 \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

3. Consider the `iris` dataset. Let X be the 4 numerical variables

```
X = as.matrix(iris[,1:4])
```

- Compute the sample mean vector, the sample covariance matrix, and the sample correlation matrix for the four numerical variables using the in built R commands `colMeans`, 'cov, and `cor`.
- Compute the centering matrix for $\mathbf{H} = \mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top$ using $n = 150$ (the number of data points in the iris dataset), and check compute the column means of HX are all zero (or close - there will be numerical error). Compute the sample covariance and correlation matrices using H.

```
n=150
H=diag(rep(1,n))-rep(1,n)%*%t(rep(1,n))/n    # calculate the centering matrix H
```

- Check the properties of the centering matrix (you can ignore 7.) given in Section 2.4
- What does the following command do?

```
sweep(X, 2, colMeans(X))
```

Thus you'll see that it usually isn't worth computing the centering matrix when doing things in practice. We use H in the description of the methods as it makes the mathematics easier to write down.

- Compute the covariance matrix of X directly (ie, don't use the `cov` command - but do check your answer with `cov`).

2.6 Exercises

1. Are the vectors $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, and $\begin{pmatrix} -1 \\ 2 \end{pmatrix}$ linearly independent?
   - Give two different bases for $\mathbb{R}^2$
   - Describe three different subspaces of $\mathbb{R}^3$
2. Let
$$A = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 2 & -4 \\ 1 & 1 & 1 \end{pmatrix}$$
   - What is rank(A)?
   - Write the product Ax where $x^\top = (x_1, x_2, x_3)$ as both the inner product of the rows of A and x, and as a linear combination of the columns of A (see section 2.2.2)
   - Describe the column space of A. What is its dimension?
   - Find a vector in the kernel of A.
   - Describe the kernel of A as a vector space and give a basis for the space.
   - Is A invertible? What is det(A)?
3. Let Σ be an arbitrary covariance matrix.
   - Show Σ is symmetric and non-negative definite.
   - Give examples of both singular and non-singular covariance matrices.
   - What condition must the eigenvalues of a non-singular covariance matrix satisfy?
4. Let's consider the inner product space $\mathbb{R}^3$ with the Euclidean inner product
$$\langle x, y \rangle = x^\top y$$
   Let
$$x_1 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \quad x_2 = \begin{pmatrix} -1 \\ 0 \\ -2 \end{pmatrix}. \quad x_3 = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$$
   - What is the angle between x_1 and x_2? Which pairs of vectors are orthogonal to each other?

- What is the norm associated with this inner-product space, and compute the norm of $x_1, \ldots, x_3$. What is the geometric interpretation of the norm?

5. Prove the following statements:
   - The determinant of an orthogonal matrix must be either 1 or -1.
   - If A and B are orthogonal matrices, then AB must also be orthogonal.
   - Let A be an $n \times n$ matrix of the form

     $$A = QBQ^\top.$$

     where Q is an $n \times n$ orthogonal matrix, and B is an $n \times n$ diagonal matrix. Prove that A is symmetric.

6. Consider the matrix

   $$P = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

   - Show that P is a projection matrix.
   - Describe the subspace P projects onto.
   - Describe the image and kernel of P.
   - Repeat the above questions using $\mathbf{I} - P$ and check proposition 2.4.

7. Let $W = \mathbb{R}^3$ with the usual inner product. Consider the orthogonal projection from W onto the subspace U defined by

   $$U = \text{span}\left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}.$$

   - What is projection of the vector

     $$v = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

     onto U? Show that this vector does minimize $||v - u||$ for $u \in U$.
   - Write down the orthogonal projection matrix for the projection W onto U and check it is a projection matrix. Check your answer to the previous part of the question.

8. The centering matrix will be play an important role in this module, as we will use it to remove the column means from a matrix (so that each column has mean zero), *centering* the matrix.

   Define

   $$H = \mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top$$

   to be the $n \times n$ centering matrix (see 2.4).
   Let $x = (x_1, \ldots, x_n)^\top$ denote a vector and let $X = [x_1, \ldots, x_n]^\top$ denote an $n \times p$ data matrix.

Define the scalar sample mean $\bar{x} = n^{-1}\sum_{i=1}^{n} x_i$ and the sample mean vector $\bar{x} = n^{-1}\sum_{i=1}^{n} x_i$.

i. Show by direct calculation that H is a projection matrix, i.e. $H^\top = H$ and $H^2 = H$.

ii. Show that $\mathbf{1}_n$ is an eigenvector of H. What is the corresponding eigenvalue? What are the remaining eigenvalues equal to?

iii. Show that

$$Hx = x - \bar{x}\mathbf{1}_n = (x_1 - \bar{x}, \ldots, x_n - \bar{x})^\top.$$

Hint: first show that $n^{-1}\mathbf{1}_n^\top x = \bar{x}$.

iv. Show that

$$x^\top Hx = \sum_{i=1}^{n}(x_i - \bar{x})^2.$$

Hint: use the fact that H is a projection matrix and hence express $x^\top Hx$ as a scalar product of Hx with itself.

v. Assuming X is an $n \times p$ matrix, show that

$$HX = [x_1 - \bar{x}, \ldots, x_n - \bar{x}]^\top.$$

Hint: first show that $n^{-1}\mathbf{1}_n^\top X = \bar{x}^\top$.

vi. Using S to denote the sample covariance matrix, show that

$$X^\top HX = \sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top = nS, \tag{2.3}$$

Hint: using the fact that H is a projection matrix, show that $X^\top HX = (HX)^\top(HX)$.

Comment: Equation (2.3) provides a convenient way to calculate the sample covariance matrix directly in R, given the data matrix X.

Chapter 3

Matrix decompositions

This chapter focusses on two ways to decompose a matrix into smaller parts. We can then think about which are the most important parts of the matrix, and that will be useful when we think about dimension reduction. The highlight of the chapter is the singular value decomposition (SVD), which is one of the most useful mathematical concepts from the past century, and is relied upon throughout statistics and machine learning. The SVD extends the idea of the eigen (or spectral) decomposition of symmetric square matrices to any matrix.

- Matrix-matrix products
- Eigenvalues and the spectral decomposition
- Introduction to the singular value decomposition
- SVD optimization results
- Low-rank approximation

3.1 Matrix-matrix products

Before we can introduce the SVD, we first need to recap some basic material on matrix multiplication and eigenvalues. We saw in section 2.2.2 that we can think about matrix-vector products in two ways: Ax is rows of A times x; or as a linear combination of the columns of A. We can similarly think about matrix-matrix products in two ways.

The usual way to think about the matrix product AB is as the rows of A times the columns of B:

$$\begin{bmatrix} \cdot & \cdot & \cdot \\ a_{21} & a_{22} & a_{23} \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & b_{12} & \cdot \\ \cdot & b_{22} & \cdot \\ \cdot & b_{32} & \cdot \end{bmatrix}$$

A better way (for this module) to think of AB is as the columns of A times the rows of B. If we let a_i denote the columns of A, and b_i^* the rows of B then

$$\begin{bmatrix} | & | & | \\ a_1 & a_2 & a_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} - & b_1^* & - \\ - & b_2^* & - \\ - & b_3^* & - \end{bmatrix} = \sum_{i=1}^{3} a_i b_i^*$$

i.e., AB is a sum of the columns of A times the rows of B.

Note that if a is a vector of length n and b is a vector of length p then $ab^\top$ is an $n \times p$ matrix.

Example 3.1.

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} (2\ 3\ 1) = \begin{pmatrix} 2 & 3 & 1 \\ 4 & 6 & 2 \end{pmatrix}.$$

Note that $ab^\top$ is a rank-1 matrix as its columns are all multiples of a, or in other words, its column space is just multiples of a.

$$\mathcal{C}(ab^\top) = \{\lambda a : \lambda \in \mathbb{R}\}.$$

We sometimes call $ab^\top$ the **outer product** of a with b.

By thinking of matrix-matrix multiplication in this way

$$AB = \sum_{i=1}^{k} a_i b_i^*$$

(where k is the number of columns of A and the number of rows of B) we can see that the product is a sum of rank-1 matrices. We can think of rank-1 matrices as the building blocks of matrices.

This chapter is about ways of decomposing matrices into their most important parts, and we will do this by thinking about the most important rank-1 building blocks.

Firstly though, we need a recap on eigenvectors.

3.2 Spectral/eigen decomposition

3.2.1 Eigenvalues and eigenvectors

Consider the $n \times n$ matrix A. We say that vector $x \in \mathbb{R}^n$ is an **eigenvector** corresponding to **eigenvalue** λ of A if

$$\mathbf{Ax} = \lambda \mathbf{x}.$$

To find the eigenvalues of a matrix, we note that if λ is an eigenvalue, then $(A - \lambda \mathbf{I}_n)x = 0$, i.e., the kernel of $A - \lambda \mathbf{I}_n$ has dimension at least 1, so $A - \lambda \mathbf{I}_n$ is not invertible, and so we must have $\det(\mathbf{A} - \lambda \mathbf{I}_n) = 0$.

Let $R(\lambda) = \det(\mathbf{A} - \lambda \mathbf{I}_n)$, which is an n^{th} order polynomial in λ. To find the eigenvalues of A we find the n roots $\lambda_1, \dots, \lambda_n$ of $R(\lambda)$. We will always consider ordered eigenvalues so that $\lambda_1 \geq \cdots \geq \lambda_n$

Proposition 3.1. *If* $\mathbf{A}$ *is symmetric (i.e.* $\mathbf{A}^\top = \mathbf{A}$*) then the eigenvalues and eigenvectors of* $\mathbf{A}$ *are* real *(in* $\mathbb{R}$*).*

Proposition 3.2. *If* $\overset{n\times n}{\mathbf{A}}$ *is a symmetric matrix then its determinant is the product of its eigenvalues, i.e.* $det(\mathbf{A}) = \lambda_1 \dots \lambda_n$.

Thus,

$$A \text{ is invertible } \iff \det(A) \neq 0 \iff \lambda_i \neq 0 \forall i \iff A \text{ is of full rank}$$

3.2.2 Spectral decomposition

The key to much of dimension reduction is finding matrix decompositions. The first decomposition we will consider is the **spectral decomposition** (also called an **eigen-decomposition**).

Proposition 3.3. ***(Spectral decomposition)****. Any symmetric matrix* $\overset{n\times n}{\mathbf{A}}$ *can be written as*

$$\mathbf{A} = \mathbf{Q}\, Q^\top = \sum_{i=1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top,$$

where $\overset{n\times n}{} = diag\{\lambda_1, \dots, \lambda_n\}$ *is a diagonal matrix consisting of the eigenvalues of* $\mathbf{A}$ *and* $\overset{n\times n}{\mathbf{Q}}$ *is an orthogonal matrix (*$QQ^\top = Q^\top Q = \mathbf{I}_n$*) whose columns are unit eigenvectors* $\mathbf{q}_1, \dots, \mathbf{q}_n$ *of* $\mathbf{A}$*.*

Because Λ is a diagonal matrix, we sometimes refer to the spectral decomposition as **diagonalizing** the matrix A as $Q^\top AQ = \Lambda$ is a diagonal matrix.

This will be useful at various points throughout the module. Note that it relies upon the fact that the eigenvectors of A can be chosen to be mutually orthogonal, and as there are n of them, they form an orthonormal basis for $\mathbb{R}^n$.

Corollary 3.1. *The rank of a symmetric matrix is equal to the number of non-zero eigenvalues (counting according to their multiplicities).*

Proof. If r is the number of non-zero eigenvalues of A, then we have (after possibly reordering the λ_i)

$$\mathbf{A} = \sum_{i=1}^{r} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top.$$

Each $\mathbf{q}_i\mathbf{q}_i^\top$ is a rank 1 matrix, with column space equal to the span of q_i. As the q_i are orthogonal, the columns spaces $\mathcal{C}(q_i q_i^\top)$ are orthogonal, and their union is a vector space of dimension r. Hence the rank of A is r. □

Lemma 3.1. *Let* $\overset{n\times n}{\mathbf{A}}$ *be a symmetric matrix with (necessarily real) eigenvalues* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. *Then* $\mathbf{A}$ *is* positive definite *if and only if* $\lambda_n > 0$. *It is positive semi-definite if and only if* $\lambda_n \geq 0$.

Proof. If A is positive definite, and if x is a unit-eigenvalue of A corresponding to λ_n, then

$$0 \leq x^\top Ax = \lambda_n x^\top x = \lambda_n.$$

Conversely, suppose A has positive eigenvalues. Because A is real and symmetric, we can write it as $A = Q\,Q^\top$. Now if x is a non-zero vector, then $y = Q^\top x \neq 0$, (as $Q^\top$ has inverse Q and hence $\dim \mathrm{Ker}(Q) = 0$). Thus

$$x^\top Ax = y^\top\, y = \sum_{i=1}^{n} \lambda_i y_i^2 > 0$$

and thus A is positive definite. □

Note: A covariance matrix Σ is always positive semi- definite (and thus always has non-negative eigenvalues). To see this, recall that if x is a random vector with $\mathbb{V}\mathrm{ar}(x) = \Sigma$, then for any constant vector a, the random variable $a^\top x$ has variance $\mathbb{V}\mathrm{ar}(a^\top x) = a^\top \Sigma a$. Because variances are positive, we must have

$$a^\top \Sigma a \geq 0 \;\forall\; a.$$

Moreover, if Σ is positive definite (so that its eigenvalues are positive), then its determinant will be positive (so that Σ is **non-singular**) and we can find an inverse Σ^{-1} matrix, which is called the **precision** matrix.

Proposition 3.4. *The eigenvalues of a projection matrix* P *are all* 0 *or* 1.

3.2.3 Matrix square roots

From the spectral decomposition theorem, we can see that if A is a symmetric positive semi-definite matrix, then for any integer p

$$A^p = Q\,{}^pQ^\top.$$

If in addition A is positive definite (rather than just semi-definite), then

$$A^{-1} = Q\,{}^{-1}Q^\top$$

where ${}^{-1} = \mathrm{diag}\{\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_n}\}$.

The spectral decomposition also gives us a way to define a matrix square root. If we assume A is positive semi-definite, then its eigenvalues are non-negative, and the diagonal elements of are all non-negative.

We then define $A^{1/2}$, a matrix square root of A, to be $A^{1/2} = Q\Lambda^{1/2}Q^\top$ where $\Lambda^{1/2} = \text{diag}\{\lambda_1^{1/2}, \dots, \lambda_n^{1/2}\}$. This definition makes sense because

$$\begin{aligned} A^{1/2}A^{1/2} &= Q\Lambda^{1/2}Q^\top Q\Lambda^{1/2}Q^\top \\ &= Q\Lambda^{1/2}\Lambda^{1/2}Q^\top \\ &= Q\Lambda Q^\top \\ &= A, \end{aligned}$$

where $Q^\top Q = \mathbf{I}_n$ and $\Lambda^{1/2}\Lambda^{1/2} = \Lambda$. The matrix $A^{1/2}$ is not the only matrix square root of A, but it *is* the only symmetric, positive semi-definite square root of A.

If A is positive definite (as opposed to just positive semi-definite), then all the λ_i are positive and so we can also define $A^{-1/2} = Q\Lambda^{-1/2}Q^\top$ where $\Lambda^{-1/2} = \text{diag}\{\lambda_1^{-1/2}, \dots, \lambda_n^{-1/2}\}$. Note that

$$A^{-1/2}A^{-1/2} = Q\Lambda^{-1/2}Q^\top Q\Lambda^{-1/2}Q^\top = Q\Lambda^{-1}Q^\top = A^{-1},$$

so that, as defined above, $A^{-1/2}$ is the matrix square root of A^{-1}. Furthermore, similar calculations show that

$$A^{1/2}A^{-1/2} = A^{-1/2}A^{1/2} = \mathbf{I}_n,$$

so that $A^{-1/2}$ is the matrix inverse of $A^{1/2}$.

3.3 Singular Value Decomposition (SVD)

The spectral decomposition theorem (Proposition 3.3) gives a decomposition of any symmetric matrix. We now give a generalisation of this result which applies to *all* matrices.

If matrix A is not a square matrix, then it cannot have eigenvectors. Instead, it has **singular vectors** corresponding to **singular values**. Suppose is a $n \times p$ matrix. Then we say σ is a **singular value** with corresponding **left** and **right** singular vectors u and v (respectively) if

$$Av = \sigma u \quad \text{and} \quad A^\top u = \sigma v$$

If A if a symmetric matrix then $u = v$ is a eigenvector and σ is an eigenvalue.

The singular value decomposition (SVD) **diagonalizes** A into a product of a matrix of left singular vectors U, a diagonal matrix of singular values Σ, and a matrix of right singular vectors V.

$$A = U\Sigma V^\top.$$

Proposition 3.5. ***(Singular value decomposition)***. *Let A be a $n \times p$ matrix of rank r, where $1 \leq r \leq \min(n, p)$. Then there exists a $n \times r$ matrix $U = [u_1, \dots, u_r]$, a $p \times r$ matrix $V = [\mathbf{v}_1, \dots, \mathbf{v}_r]$, and a $r \times r$ diagonal matrix $\Sigma = diag\{\sigma_1, \dots, \sigma_r\}$ such that*

$$A = U\, V^\top = \sum_{i=1}^{r} \sigma_i u_i \mathbf{v}_i^\top,$$

where $U^\top U = \mathbf{I}_r = V^\top V$ and the $\sigma_1 \geq \dots \geq \sigma_r > 0$.

Note that the u_i and the $\mathbf{v}_i$ are necessarily unit vectors, and that we have ordered the singular values from largest to smallest. The scalars $\sigma_1, \dots, \sigma_r$ are called the **singular values** of A, the columns of U are the **left singular vectors**, and the columns of V are the **right singular vectors**.

The form of the SVD given above is called the **compact singular value decomposition**. Sometimes we write it in a non-compact form

$$A = U\Sigma V^\top$$

where U is a $n \times n$ orthogonal matrix ($U^\top U = UU^\top = \mathbf{I}_n$), V is a $p \times p$ orthogonal matrix ($V^\top V = VV^\top = \mathbf{I}_p$), and Σ is a $n \times p$ diagonal matrix

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & & & 0 \\ 0 & \sigma_2 & 0 & \dots & & \\ \vdots & & & & & \\ 0 & 0 & & \dots & \sigma_r & \\ 0 & 0 & & \dots & & 0 \quad \dots \\ \vdots & & & & & \\ 0 & 0 & & \dots & & 0 \end{pmatrix}. \tag{3.1}$$

The columns of U and V form an orthonormal basis for $\mathbb{R}^n$ and $\mathbb{R}^p$ respectively. We can see that we recover the compact form of the SVD by only using the first r columns of U and V, and truncating Σ to a $r \times r$ matrix with non-zero diagonal elements.

When A is symmetric, we take $\mathbf{U} = V$, and the spectral decomposition theorem is recovered, and in this case (but not in general) the singular values of A are eigenvalues of A.

Proof. $A^\top A$ is a $p \times p$ symmetric matrix, and so by the spectral decomposition theorem we can write it as

$$A^\top A = V\Lambda V^\top$$

where V is a $p \times p$ orthogonal matrix containing the orthonormal eigenvectors of $A^\top A$, and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_r)$ is a diagonal matrix of eigenvalues with $\lambda_1 \geq \ldots \geq \lambda_r > 0$ (by Corollary 3.1).

For $i = 1, \ldots, r$, let $\sigma_i = \sqrt{\lambda_i}$ and let $u_i = \frac{1}{\sigma_i} A v_i$. Then the vectors u_i are orthonormal:

$$\begin{aligned} u_i^\top u_j &= \frac{1}{\sigma_i \sigma_j} v_i^\top A^\top A v_j \\ &= \frac{\sigma_j^2}{\sigma_i \sigma_j} v_i^\top v_j \quad \text{as } v_j \text{ is an eigenvector of } A^\top A \\ &= \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad \text{as the } v_i \text{ are orthonormal vectors.} \end{aligned}$$

In addition

$$A^\top u_i = \frac{1}{\sigma_i} A^\top A v_i = \frac{\sigma_i^2}{\sigma_i} v_i = \sigma_i v_i$$

and so u_i and v_i are left and right singular vectors.

Let $U = [u_1 \ u_2 \ \ldots \ u_r \ \ldots \ u_n]$, where $u_{r+1}, \ldots, u_n$ are chosen to complete the orthonormal basis for $\mathbb{R}^n$ given $u_1, \ldots, u_r$, and let Σ be the $n \times p$ diagonal matrix in Equation (3.1).

Then we have shown that

$$U = AV\Sigma^{-1}$$

Thus

$$\begin{aligned} U &= AV\Sigma^{-1} \\ U\Sigma &= AV \\ U\Sigma V^\top &= A. \end{aligned}$$

□

Note that by construction we've shown that $A^\top A$ has eigenvalues σ_i^2 with corresponding eigenvectors v_i. We also can also show that $AA^\top$ has eigenvalues σ_i^2, but with corresponding eigenvectors u_i.

$$AA^\top u_i = \sigma_i A v_i = \sigma_i^2 u_i$$

Proposition 3.6. *Let A be any matrix of rank r. Then the non-zero eigenvalues of both $AA^\top$ and $A^\top A$ are $\sigma_1^2, \ldots, \sigma_r^2$. The corresponding unit eigenvectors of $AA^\top$ are given by the columns of U, and the corresponding unit eigenvectors of $A^\top A$ are given by the columns of $\mathbf{V}$.*

Notes:

1. The SVD expresses a matrix as a sum of rank-1 matrices

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^\top.$$

We can think of these as a list of the building blocks of A ordered by their importance ($\sigma_1 \geq \sigma_2 \geq ...$).

2. The singular value decomposition theorem shows that every matrix is diagonal, provided one uses the proper bases for the domain and range spaces. We can **diagonalize** A by

$$U^\top A V = \Sigma.$$

3. The SVD reveals a great deal about a matrix. Firstly, the rank of A is the number of non-zero singular values. The left singular vectors $u_1, \ldots, u_r$ are an orthonormal basis for the columns space of A, $\mathcal{C}(A)$, and the right singular vectors $v_1, \ldots, v_r$ are an orthonormal basis for $\mathcal{C}(A^\top)$, the row space of A. The vectors $v_{r+1}, \ldots, v_p$ from the non-compact SVD are a basis for the kernel of A (sometimes called the null space $\mathcal{N}(A)$), and $u_{r+1}, \ldots, u_n$ are a basis for $\mathcal{N}(A^\top)$.

4. The SVD has many uses in mathematics. One is as a generalized inverse of a matrix. If A is $n \times p$ with $n \neq p$, or if it is square but not of full rank, then A cannot have an inverse. However, we say A^+ is a generalized inverse if $AA^+A = A$. One such generalized inverse can be obtained from the SVD by $A^+ = V\Sigma^{-1}U^\top$ - this is known as the Moore-Penrose pseudo-inverse.

3.3.1 Examples

In practice, we don't compute SVDs of a matrix by hand: in R you can use the command `SVD(A)` to compute the SVD of matrix `A`. However, it is informative to do the calculation yourself a few times to help fix the ideas.

Example 3.2. Consider the matrix $A = xy^\top$. We can see this is a rank-1 matrix, so it only has one non-zero singular value which is $\sigma_1 = ||x||.||y||$. Its SVD is given by

$$U = \frac{1}{||x||}x, \quad V = \frac{1}{||y||}y, \quad \text{and } \Sigma = ||x||.||y||.$$

Example 3.3. Let

$$A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}.$$

Let's try to find the SVD of A.

We know the singular values are the square roots of the eigenvalues of $AA^\top$ and $A^\top A$. We'll work with the former as it is only 2×2.

$$AA^\top = \begin{pmatrix} 17 & 8 \\ 8 & 17 \end{pmatrix} \quad \text{and so } \det(AA^\top - \lambda \mathbf{I}) = (17-\lambda)^2 - 64$$

Solving $\det(AA^\top - \lambda \mathbf{I}) = 0$ gives the eigenvalues to be $\lambda = 25$ or 9. Thus the singular values of A are $\sigma_1 = 5$ and $\sigma_2 = 3$, and

$$\Sigma = \begin{pmatrix} 5 & 0 \\ 0 & 3 \end{pmatrix}.$$

The columns of U are the *unit* eigenvectors of $AA^\top$ which we can find by solving

$$(A - 25\mathbf{I}_2)u = \begin{pmatrix} -8 & 8 \\ 8 & -8 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and}$$
$$(A - 9\mathbf{I}_2)u = \begin{pmatrix} 8 & 8 \\ 8 & 8 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

And so, remembering that the eigenvectors used to form V need to be *unit* vectors, we can see that

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Finally, to compute V recall that $\sigma_i v_i = A^\top u_i$ and so

$$V = A^\top U \Sigma^{-1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \frac{1}{3} \\ 1 & \frac{-1}{3} \\ 0 & \frac{4}{3} \end{pmatrix}.$$

This completes the calculation, and we can see that we can express A as

$$A = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 5 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{3\sqrt{2}} & \frac{-1}{3\sqrt{2}} & \frac{4}{3\sqrt{2}} \end{pmatrix}$$

or as the sum of rank-1 matrices:

$$A = 5 \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{3\sqrt{2}} \end{pmatrix} + 3 \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{3\sqrt{2}} & \frac{-1}{3\sqrt{2}} & \frac{4}{3\sqrt{2}} \end{pmatrix}$$

This is the compact form of the SVD. To find the non-compact form we need V to be a 3×3 matrix, which requires us to find a 3rd column that is orthogonal to the first two columns (thus completing an orthonormal basis for $\mathbb{R}^3$). We can do that with the vector $v_3 = \frac{1}{\sqrt{17}}(2 \ -2 \ -3)$ giving the non-compact SVD for A.

$$A = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{3\sqrt{2}} & \frac{2}{\sqrt{17}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{3\sqrt{2}} & \frac{-2}{\sqrt{17}} \\ 0 & \frac{4}{3\sqrt{2}} & \frac{-3}{\sqrt{17}} \end{pmatrix}^\top$$

Let's check our answer in R.

```
A<- matrix(c(3,2,2,2,3,-2), nr=2, byrow=T)
svd(A)
```

```
## $d
## [1] 5 3
##
## $u
##            [,1]       [,2]
## [1,] -0.7071068 -0.7071068
## [2,] -0.7071068  0.7071068
##
## $v
##                [,1]       [,2]
## [1,] -7.071068e-01 -0.2357023
## [2,] -7.071068e-01  0.2357023
## [3,] -5.551115e-17 -0.9428090
```

The eigenvectors are only defined upto multiplication by -1 and so we can multiply any pair of left and right singular vectors by -1 and it is still a valid SVD.

Note: In practice this is a terrible way to compute the SVD as it is prone to numerical error. In practice an efficient iterative method is used in most software implementations (including R).

3.4 SVD optimization results

Why are eigenvalues and singular values useful in statistics? It is because they appear as the result of some important optimization problems. We'll see more about this in later chapters, but we'll prove a few preliminary results here.

For example, suppose $x \in \mathbb{R}^n$ is a random variable with $\mathbb{C}\text{ov}(x) = \Sigma$ (an $n \times n$ matrix), then can we find a projection of x that has either maximum or minimum variance? I.e., can we find a such that

$$\mathbb{V}\text{ar}(a^\top x) = a^\top \Sigma a$$

is maximized or minimized? To make the question interesting we need to constrain the length of a so lets assume that $||a||_2 = \sqrt{a^\top a} = 1$, otherwise we could just take $a = 0$ to obtain a projection with variance zero. So we want to solve the optimization problems involving the quadratic form $a^\top \Sigma a$:

$$\max_{a:\, a^\top a=1} \mathbf{a}^\top \mathbf{a}, \quad \text{and} \quad \min_{a:\, a^\top a=1} \mathbf{a}^\top \mathbf{a}. \tag{3.2}$$

Given that Σ is symmetric, we can write it as

$$\Sigma = V \Lambda V^\top$$

where Λ is the diagonal matrix of eigenvalues of Σ, and V is an orthogonal matrix of eigenvectors. If we let $b = V^\top a$ then

$$a^\top \Sigma a = b^\top \Lambda b = \sum_{i=1}^{n} \lambda_i b_i^2$$

and given that the eigenvalues are ordered $\lambda_1 \geq \lambda_2 \geq ...$ and that

$$\sum_{i=1}^{n} b_i^2 = b^\top b = a^\top V V^\top a = a^\top a = 1,$$

we can see that the maximumn is λ_1 obtained by setting $b = (1\ 0\ 0\ ...)^\top$. Then

$$\begin{aligned} V^\top a &= b \\ V V^\top a &= V b \\ a &= v_1 \end{aligned}$$

so we can see that the maximum is obtained when $a = v_1$, the eigenvector of Σ corresponding to the largest eigenvalue λ_1.

Similarly, the minimum is λ_n, which obtained by setting $b = (\ 0\ 0\ ...\ 0\ 1)^\top$ which corresponds to $a = v_n$.

Proposition 3.7. *For any symmetric $n \times n$ matrix Σ,*

$$\max_{a:a^\top a=1} a^\top \Sigma a = \lambda_1,$$

where the maximum occurs at $a = \pm v_1$, and

$$\min_{a:a^\top a=1} a^\top \Sigma a = \lambda_n$$

where the minimum occurs at $a = \pm v_n$, where λ_i, v_i are the ordered eigenpairs of Σ.

Note that

$$\frac{a^\top \Sigma a}{a^\top a} = \frac{a^\top \Sigma a}{||a||^\top} = (\frac{a}{||a||})^\top \Sigma (\frac{a}{||a||})$$

and so another way to write the maximization problems (3.2) is as unconstrained optimization problems:

$$\max_a \frac{a^\top \Sigma a}{a^\top a} \quad \text{and} \quad \min_a \frac{a^\top \Sigma a}{a^\top a}.$$

We obtain a similar result for non-square matrices using the singular value decomposition.

Proposition 3.8. *For any matrix A*

$$\max_{x:||x||_2=1} ||Ax||_2 = \max_x \frac{||Ax||_2}{||x||_2} = \sigma_1$$

the first singular value of A, with the maximum achieved at $x = v_1$ (the first right singular vector).

Proof. This is follows from 3.7 as

$$||Ax||_2^2 = x^\top A^\top Ax.$$

□

Finally, we will need the following result when we study canonical correlation analysis:

Proposition 3.9. *For any matrix A, we have*

$$\max_{a,b:\,||a||=||b||=1} a^\top Ab = \sigma_1.$$

with the maximum obtained at $a = u_1$ and $b = v_1$, the first left and right singular vectors of A.

Proof. See chapter 5 □

We'll see much more of this kind of thing in Chapters 4 and 5.

3.5 Low-rank approximation

One of the reasons the SVD is so widely used is that it can be used to find the best low rank approximation to a matrix. Before we discuss this, we need to define what it means for some matrix B to be a good approximation to A. To do that, we need the concept of a matrix norm.

3.5.1 Matrix norms

In Section 2.3.1 we described norms on vectors. Here will extend this idea to include norms on matrices, so that we can discuss the size of a matrix $||A||$, and the distance between two matrices $||A - B||$. There are two particular norms we will focus on. The first is called the Frobenious norm (or sometimes the Hilbert-Schmidt norm).

Definition 3.1. Let $A \in \mathbb{R}^{n\times p}$. The **Frobenius norm** of A is

$$||A||_F = \left(\sum_{i=1}^{n}\sum_{j=1}^{p} |a_{ij}|^2\right)^{\frac{1}{2}} = (\operatorname{tr} A^\top A)^{\frac{1}{2}}$$

where a_{ij} are the individual entries of A.

Note that the Frobenius norm is invariant to rotation by an orthogonal matrix U:

$$\begin{aligned} ||AU||_F^2 &= \operatorname{tr}(U^\top A^\top AU) \\ &= \operatorname{tr}(UU^\top A^\top A) \\ &= \operatorname{tr}(A^\top A) \\ &= ||A||_F^2. \end{aligned}$$

Proposition 3.10.

$$||A||_F = \left(\sum_{i=1}^{r} \sigma_i^2\right)^{\frac{1}{2}}$$

where σ_i are the singular values of A, and $r = \operatorname{rank}(A)$.

Proof. Using the (non-compact) SVD $A = U\Sigma V^\top$ we have

$$||A||_F = ||U^\top A||_F = ||U^\top AV||_F = ||\Sigma||_F = \operatorname{tr}(\Sigma^\top\Sigma)^{\frac{1}{2}} = \left(\sum \sigma_i^2\right)^{\frac{1}{2}}.$$

□

We previously defined the p-norms for vectors in $\mathbb{R}^p$ to be

$$||x||_p = \left(\sum |x_i|^p\right)^{\frac{1}{p}}.$$

These vector norms *induce* matrix norms, sometimes also called operator norms:

Definition 3.2. The p-norms for matrices are defined by

$$||A||_p = \sup_{x\neq 0} \frac{||Ax||_p}{||x||_p} = \sup_{x:||x||_p=1} ||Ax||_p$$

Proposition 3.11.

$$||A||_2 = \sigma_1$$

where σ_1 is the first singular value of A.

Proof. By Proposition 3.8. □

3.5.2 Eckart-Young-Mirsky Theorem

Now that we have defined a norm (i.e., a distance) on matrices, we can think about approximating a matrix A by a matrix that is easier to work with. We have shown that any matrix can be split into the sum of rank-1 component matrices

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^\top$$

We'll now consider a family of approximations of the form

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^\top \tag{3.3}$$

where $k <= r = \text{rank}(A)$. This is a rank-k matrix, and as we'll now show, it is the best possible rank-k approximation to A.

Theorem 3.1. ***(Eckart-Young-Mirsky)*** *For either the 2-norm $||\cdot||_2$ or the Frobenious norm $||\cdot||_F$*

$$||A - A_k|| \leq ||A - B|| \text{ for all rank-k matrices } B.$$

Moreover,

$$||A - A_k|| = \begin{cases} \sigma_{k+1} & \text{for the } ||\cdot||_2 \text{ norm} \\ \left(\sum_{i=k+1}^{r} \sigma_i^2\right)^{\frac{1}{2}} & \text{for the } ||\cdot||_F \text{ norm.} \end{cases}$$

Proof. The last part follows from Propositions 3.11 and 3.10.

Non-examinable: this is quite a tricky proof, but I've included it as its interesting to see. We'll just prove it for the 2-norm. Let B be an $n \times p$ matrix of rank k. The null space $\mathcal{N}(B) \subset \mathbb{R}^p$ must be of dimension $p - k$ by the rank nullity theorem.

Consider the $p \times (k+1)$ matrix $V_{k+1} = [v_1 \ ... \ v_{k+1}]$. This has rank $k+1$, and has column space $\mathcal{C}(V_{k+1}) \subset \mathbb{R}^p$. Because

$$\dim \mathcal{N}(B) + \dim \mathcal{C}(V_{k+1}) = p - k + k + 1 = p + 1$$

we can see that $\mathcal{N}(B)$ and $\mathcal{C}(V_{k+1})$ cannot be disjoint spaces (as they are both subsets of the p-dimensional space $\mathbb{R}^p$). Thus we can find $w \in \mathcal{N}(B) \cap \mathcal{C}(V_{k+1})$, and moreover we can choose w so that $||w||_2 = 1$.

Because $w \in \mathcal{C}(V_{k+1})$ we can write $w = \sum_{i=1}^{k+1} w_i v_i$ with $\sum_{i=1}^{k+1} w_i^2 = 1$.

Then

$$
\begin{aligned}
||A - B||_2^2 &\geq ||(A - B)w||_2^2 \quad \text{by definition of the matrix 2-norm} \\
&= ||Aw||_2^2 \quad \text{as } w \in \mathcal{N}(B) \\
&= w^\top V \Sigma^2 V^\top w \quad \text{using the SVD} A = U \Sigma V^\top \\
&= \sum_{i=1}^{k+1} \sigma_i^2 w_i^2 \quad \text{by substituting } w = \sum_{i=1}^{k+1} w_i v_i \\
&\geq \sigma_{k+1}^2 \sum_{i=1}^{k+1} w_i^2 \quad \text{as } \sigma_1 \geq \sigma_2 \geq ... \\
&= \sigma_{k+1}^2 \quad \text{as } \sum_{i=1}^{k+1} w_i^2 = 1 \\
&= ||A - A_k||_2^2
\end{aligned}
$$

as required □

This best-approximation property is what makes the SVD so useful in applications.

3.5.3 Example: image compression

As an example, lets consider the image of some peppers from the USC-SIPI image database.

```
library(tiff)
library(rasterImage)
peppers<-readTIFF("figs/Peppers.tiff")
plot(as.raster(peppers))
```

This is a 512×512 colour image, meaning that there are three matrices R, B, G of dimension 512×512) giving the intensity of red, green, and blue for each pixel. Naively storing this matrix requires 5.7Mb.

We can compute the SVD of the three colour intensity matrices, and the view the image that results from using reduced rank versions B_k, G_k, R_k instead (as in Equation (3.3)). The image below is formed using $k = 5, 30, 100$, and 300 basis vectors.

```
svd_image <- function(im,k){
  s <- svd(im)
  Sigma_k <- diag(s$d[1:k])
  U_k <- s$u[,1:k]
  V_k <- s$v[,1:k]
  im_k <- U_k %*% Sigma_k %*% t(V_k)
   ## the reduced rank SVD produces some intensities <0 and >1.
  # Let's truncate these
  im_k[im_k>1]=1
  im_k[im_k<0]=0
  return(im_k)
}

par(mfrow=c(2,2), mar=c(1,1,1,1))

pepprssvd<- peppers
for(k in c(4,30,100,300)){
  svds<-list()
  for(ii in 1:3) {
    pepprssvd[,,ii]<-svd_image(peppers[,,ii],k)
  }
```

```
  plot(as.raster(pepprssvd))
}
```

You can see that for $k = 30$ we have a reasonable approximation, but with some errors. With $k = 100$ it is hard to spot the difference with the original. The size of the four compressed images is 45Kb, 345Kb, 1.1Mb and 3.4Mb.

You can see further demonstrations of image compression with the SVD here.

We will see much more of the SVD in later chapters.

3.6 Computer tasks

1. Finding the eigenvalues and eigenvectors of a matrix is easy in R.

```
A=matrix(c(3,1,1,6),nrow=2,byrow=TRUE)    # use a to define a matrix A
Eig=eigen(A)                             # the eigenvalues and eigenvectors of A
                                           # are stored in the list Eig
lambda=Eig$values                        # extract the eigenvalues from Eig and
                                           # store in the vector e
lambda                                   # you should see the eigenvalues in
```

```
## [1] 6.302776 2.697224
```

```
                                    # descending order
Q=Eig$vectors                      # extract the eigenvectors from Eig and
                                    # store then in the columns of Q
```

The spectral decomposition of A is

$$A = Q\Lambda Q^\top$$

Let's check this in R (noting as always that there may be some numerical errors)

```
Q%*%diag(lambda)%*%t(Q)            # reconstruct A,
```

```
##      [,1] [,2]
## [1,]    3    1
## [2,]    1    6
```

```
                                    # where t(Q) gives the transpose of Q
```

Since A is positive definite, we can calculate the symmetric, positive definite square root of A.

```
Asqrt=Q%*%diag(lambda**0.5)%*%t(Q) # lambda**0.5 contains the square roots
Asqrt%*%Asqrt                       # it is seen that A is recovered
```

```
##      [,1] [,2]
## [1,]    3    1
## [2,]    1    6
```

- Instead of using the full eigendecomposition for A, try truncating it and using just a single eigenvalue and eigenvector, i.e., compute

$$A' = \lambda_1 q_1 q_1^\top$$

- Compute the difference between A and A' using the 2-norm and the Frobenius norm.

2. The singular value decomposition can be computed in R using the command `svd`. Let X be the four numerical variables in the `iris` dataset with the column mean removed

```
n=150
H=diag(rep(1,n))-rep(1,n)%*%t(rep(1,n))/n   # calculate the centering matrix H
X=H%*% as.matrix(iris[,1:4])
# This can also be done using the command
# sweep(iris[,1:4], 2, colMeans(iris[,1:4]))  # do you understand why?
```

- Compute the SVD of X in R and report its singular values.
- Does R report the full or compact SVD?
- Check that $Xv = \sigma u$.

- Compute the best rank-1, rank-2, and rank-3 approximations to X, and report the 2-norm and Frobenious norm for these approximations
- Compute the eigenvalues of $X^\top X$. How do these relate to the singular values? How does $X^\top X$ relate to the sample covariance matrix of the iris data? How do the singular values relate to the eigenvalues of the covariance matrix?
- Let S be the sample covariance matrix of the iris dataset. What vector maximizes $x^\top Sx$?

3. Choose an few images from the USC-SIPI Image Database and repeat the image compression example from the notes. Which type of images compress well do you think?

3.7 Exercises

1. Compute, by hand (but check your answer in R), the singular value decomposition (full and compact) of the following matrices.
   - $\begin{pmatrix} 2 & 0 \\ 0 & -1 \end{pmatrix}$
   - $\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$
2. Let
$$X = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$
The eigen-decomposition of $X^\top X$ is
$$X^\top X = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}^\top$$
Use this fact to compute answer the following questions:
   - What are the singular values of X?
   - What are the right singular vectors of X?
   - What are the left singular vectors of X?
   - Give the compact SVD of X. Check your answer, noting that the singular vectors are only specified up to multiplication by -1
   - Can you compute the full SVD of X?
   - What is the eigen-decomposition of $XX^\top$?
   - Find a generalised inverse of matrix X.
3. The SVD can be used to solve linear systems of the form
$$Ax = y$$
where A is a $n \times p$ matrix, with compact SVD $A = U\Sigma V^\top$.

- If A is a square invertible matrix, show that

$$\tilde{x} = V\Sigma^{-1}U^\top y$$

is the unique solution to $Ax = y$, i.e., show that $A^{-1} = V\Sigma^{-1}U^\top$.

- If A is not a square matrix, then $A^+ = V\Sigma^{-1}U^\top$ is a generalized inverse (not a true inverse) matrix, and $\tilde{x} = A^+ y$ is still a useful quantity to consider as we shall now see. Let $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $y = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$. Then $Ax = y$ is an over-determined system in that there are 3 equations in 2 unknowns. Compute $\tilde{x} = A^+ y$. Is this a solution to the equation?

Note that you computed the svd for A in Q2.

- Now suppose $y = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$. There is no solution to $Ax = y$ in this case as y is not in the column space of A. Prove that $\tilde{x} = A^+ y$ solves the least squares problem

$$\tilde{x} = \arg\min_x ||y - Ax||_2.$$

Hint: You can either do this directly for this problem, or you can show that the least squares solution $(A^\top A)^{-1}A^\top y = \tilde{x}$.

4. Consider the system

$$Bx = y \text{ with } B = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, y = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

This is an underdetermined system, as there are 2 equations in 3 unknowns, and so there are an infinite number of solutions for x in this case.

- Find the full SVD for $B = U\Sigma V^\top$ (noting that $B = A^\top$ for A from the previous question).
- Compute $\tilde{x} = B^+ y$, check it is a solution to the equation, and explain why

$$\tilde{x} = \sum_{i=1}^{r} v_i \frac{u_i^\top y}{\sigma_i}$$

in general, where $r \leq max(n, p)$ is the rank of B, and write out $\tilde{x}$ explicitly in this form for the given B.

- Consider x of the form

$$x = \tilde{x} + \sum_{i=r+1}^{n} \alpha_i v_i$$

  and explain why any x of this form is also a solution to $Bx = y$. Thus write out all possible solutions of the equation.
- Prove that $\tilde{x}$ is the solution with minimum norm, i.e., $||\tilde{x}||_2 \leq ||x||_2$. **Hint** $v_1, \ldots, v_p$ form a complete orthonormal basis for $\mathbb{R}^p$.

5. Prove proposition 3.4.

PART II: Dimension reduction methods

In many applications, a large number of variables are recorded for each experimental unit under study. For example, if we think of individual people as the *experimental units*, then in a health check-up we might collect data on age, blood pressure, cholesterol level, blood test results, lung function, weight, height, BMI, etc. If you use websites such as Amazon, Facebook, and Google, they store thousands (possibly millions) of pieces of information about you (this article shows you how to download the information Google stores about you, including all the locations you've visited, every search, youtube video, or app you've used and more). They process this data to create an individual profile for each user, which they can then use to create targetted adverts.

When analysing data of moderate or high dimension, it is often desirable to seeks ways to restructure the data and reduce its dimension whilst **retaining the most important information** within the data or **preserving some feature of interest** in the data. There a variety of reasons we might want to do this.

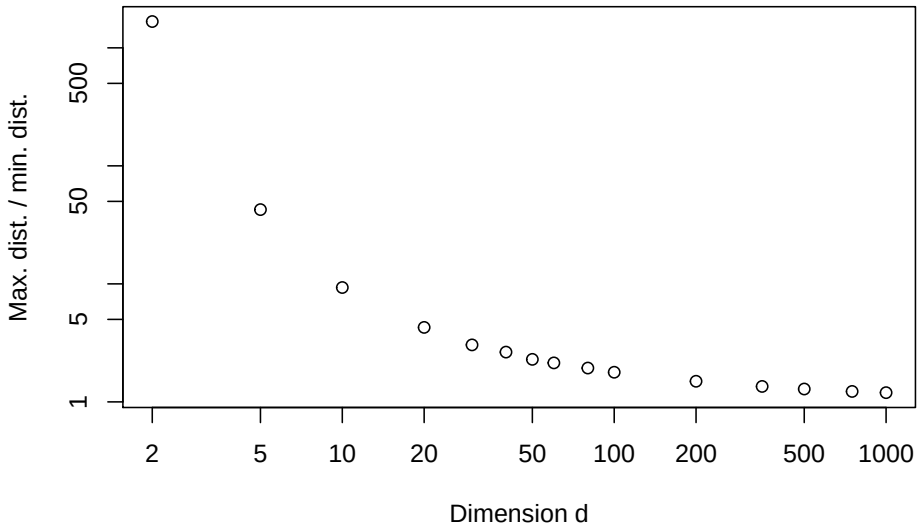
- In reduced dimensions, it is often much easier to understand and appreciate the most important features of a dataset.
- If there is a lot of reduncancy in the data, we might want to reduce the dimension to lower the memory requirements in storing it (e.g. with sound and image compression).
- In high dimensions, it can be difficult to analyse data (e.g. with statistical methods), and so reducing the dimension can be a way to make a dataset amenable to analysis.

In this part of the module we investigate three different methods for dimension reduction: Principal Component Analysis (PCA) in Chapter 4; Canonical Correlation Analysis (CCA) in Chapter 5; and Multidimensional Scaling (MDS) in Chapter 6. Matrix algebra (Chapters 2 and 3) plays a key role in all three of these techniques.

A warning

Beware that high-dimensional data can behave qualitatively differently to low-dimensional data. As an example, lets consider 1000 points uniformly distributed in $[0,1]^d$, and think about how close together or spread out the points are. A simple way to do this is to consider the ratio of the maximum and minimum distance between any two points in our sample.

```
N<-1000
averatio <-c()
ii<-1
for(d in c(2,5,10,20,30,40,50,60,80,100, 200, 350, 500, 750, 1000)){
  averatio[ii] <- mean(replicate(10, {
  X<-matrix(runif(N*d), nc=d)
  d <- as.matrix(dist(X))
  # this gives a N x N matrix of the Euclidean distances between the data points.
  maxdist <- max(d)
  mindist <- min(d+diag(10^5, nrow=N))
  # The diagonal elements of the distance matrix are zero,
  # so I've added a big number to the diagonal
  # so that we get the minimum distance between different points
  maxdist/mindist}))
  ii <- ii+1
}
plot(c(2,5,10,20,30,40,50,60,80,100, 200, 350, 500, 750, 1000),
     averatio, ylab='Max. dist. / min. dist.', xlab='Dimension d', log='xy')
```



So we can see that as the dimension increases, the ratio of the maximum and minimum distance between any two random points in our sample tends to 1. In other words, all points are the same distance apart!

Chapter 4

Principal component analysis (PCA)

With multivariate data, it is common to want to reduce the dimension of the data *in a sensible way*. For example

- exam marks across different modules are averaged to produce a single overall mark for each student
- a football league table converts the numbers of wins, draws and losses to a single measure of points.

Mathematically, these summaries are both linear combinations of the original variables of the form

$$y = u^\top x.$$

for some choice of u.

For the exam marks example, suppose each student sits $p = 4$ modules with marks, x_1, x_2, x_3, x_4. Then, writing $x = (x_1, x_2, x_3, x_4)^\top$ and choosing $u = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^\top$ gives an overall average,

$$y = u^\top x = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \frac{x_1}{4} + \frac{x_2}{4} + \frac{x_3}{4} + \frac{x_4}{4}.$$

For the football league table, if w is the number of wins, d is the number of draws and l is the number of losses then, writing $\mathbf{r} = (w, d, l)^\top$, we choose $u = (3, 1, 0)^\top$

to get the points score

$$y = u^\top \mathbf{r} = \begin{pmatrix} 3 & 1 & 0 \end{pmatrix} \begin{pmatrix} w \\ d \\ l \end{pmatrix} = 3w + 1d + 0l = 3w + d.$$

Geometric interpretation

In the two examples above, we used the vector u to convert our original variables, x, to a new variable, y, by projecting x onto u. We can think of this as a projection onto the subspace defined by u

$$U = \text{span}\{u\} = \{\lambda u : \lambda \in \mathbb{R}\} \subset \mathbb{R}^p,$$

For the exam data, each data point $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$ is a vector in $\mathbb{R}^4$, and we've expressed x in terms of its coordinates with respsect to the standard basis, $e_1^\top = (1\ 0\ 0\ 0)$ etc:

$$x = x_1 e_1 + x_2 e_2 + x_3 e_3 + x_4 e_4.$$

The vector subspace U is a line in $\mathbb{R}^4$ along the direction $u = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}^\top$.

How do we project onto subspace U?

- If $||u||_2 = 1$ then the orthogonal projection of x onto U is

  $$uu^\top x.$$

  Or in other words, the projection of x onto subspace U has coordinate $u^\top x$ with respect to basis $\{u\}$.

If you prefer to think in terms of projection matrices (see Chapter 2.3.3.1), then the matrix for projecting onto U is

$$P_U = u(u^\top u)^{-1} u^\top$$

which simplifies to

$$P_U = uu^\top$$

when $||u|| = \sqrt{u^\top u} = 1$ so that we again see the projection of x onto U is $y = P_u x = uu^\top x$.

How should we choose u?

The answer to that question depends upon the goal of the analysis. For the exam and football league examples, the choice of u is an arbitrary decision taken in order to reduce a multidimensional dataset to a single variable (average mark, or points).

A single u gives a **snapshot** or summary of the data. If u is chosen well that snapshot may tell us much of what we want to know about the data, e.g.,

- Liverpool won the league,
- student X's exam performance was first class etc.

In many cases we will want to use multiple snapshots: instead of using a single u, we will use a collection $u_1, u_2, \ldots, u_r$ and consider the derived variables

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \end{pmatrix} = \begin{pmatrix} u_1^\top x \\ u_2^\top x \\ \vdots \\ u_r^\top x \end{pmatrix}$$

In matrix notation, if we set

$$U = \begin{pmatrix} | & & | \\ u_1 & \ldots & u_r \\ | & & | \end{pmatrix}$$

then the new derived variable is

$$y = U^\top x.$$

If $\dim(y) = r < p = \dim(x)$ then we have reduced the dimension of the data. If y tells us all we need to know about the data, then we can work (plot, analyse, model) with y instead of x. If $r \ll p$ this can make working with the data significantly easier, as we can more easily visulise and understand low dimensional problems.

We will study a variety of methods for choosing U. The methods can all be expressed as constrained optimization problems:

$$\text{minimize} f_X(U) \tag{4.1}$$
$$\text{subject to } U \in \mathcal{U} \tag{4.2}$$

The objective $f_X(U)$ varies between methods: principal component analysis (PCA) maximizes variance or minimizes reconstruction error; canonical correlation analysis (CCA) maximizes correlation; multidimensional scaling (MDS) maximizes spread etc.

The contstraint on the search space $\mathcal{U}$, is usually that U must be (partially) orthogonal, but in other methods other constraints are used

4.1 PCA: an informal introduction

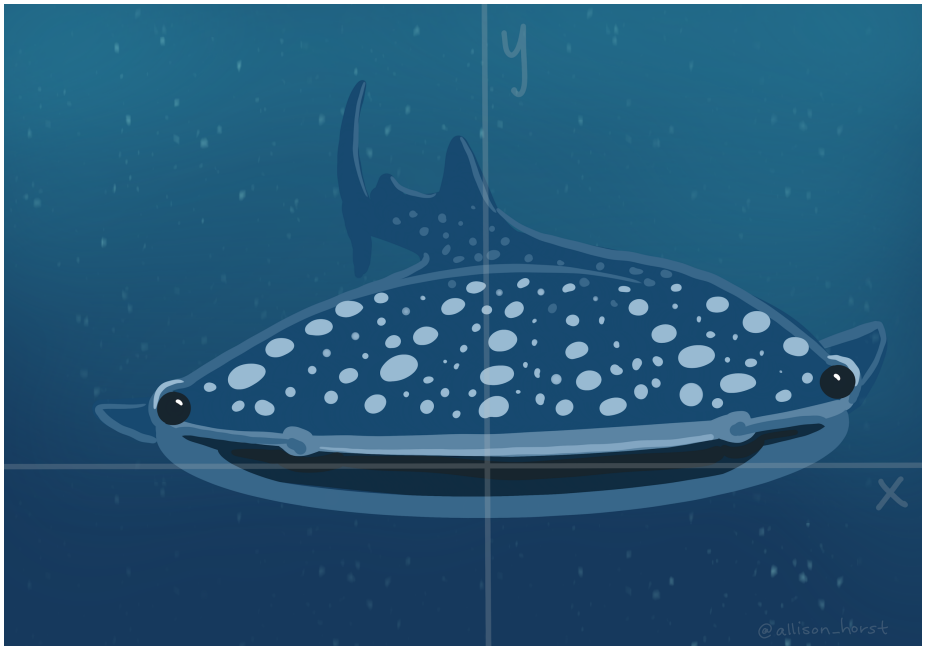
There are two different ways of motivating principal component analysis (PCA), which may in part explain why PCA is so widely used.

The first motivation, and the topic of this section, is to introduce PCA as method for maximizing the variance of the transformed variables y. We start by choosing u_1 so that $y_1 = u_1^\top x$ has maximum variance. We then choose u_2 so that $y_2 = u_2^\top x$ has maximum variance subject to being uncorrelated with y_1, and so on.

The idea is to produce a set of variables $y_1, y_2, \ldots, y_r$ that are uncorrelated, but which are most informative about the data. The thinking is that if a variable has large variance it must be informative/important.

The name **principal component analysis** comes from thinking of this as splitting the data X into its most important parts. It therefore won't surprise you to find that this involves the matrix decompositions we studied in Chapter 3.

Allison Horst (@allison_horst) gave a great illustration of how to think about PCA on Twitter. Imagine you are a whale shark with a wide mouth



and that you're swimming towards a delicious swarm of krill.

What way should you tilt your shark head in order to eat as many krill as possible? The answer is given by the first principal component of the data!

4.1.1 Notation recap

As before, let $x_1, \ldots, x_n$ be $p \times 1$ vectors of measurements on n experimental units and write

$$X = \begin{pmatrix} - & x_1^\top & - \\ - & x_2^\top & - \\ - & .. & - \\ - & x_n^\top & - \end{pmatrix}$$

IMPORTANT NOTE: In this section we will assume that X has been column centered so that the mean of each column is 0 (i.e., the sample mean of $x_1, \ldots, x_n$ is the zero vector $0 \in \mathbb{R}^p$). If X has not been column centered, replace X by

$$HX$$

where H is the centering matrix (see 2.4), or equivalently, replace x_i by $x_i - \bar{x}$. It is possible to write out the details of PCA replacing X by HX throughout, but this gets messy and obscures the important detail. Most software implementations (and in particular `prcomp` in R), automatically centre your data for you, and so in practice you don't need to worry about doing this when using a software package.

The sample covariance matrix for X (assuming it has been column centered) is

$$S = \frac{1}{n} X^\top X = \frac{1}{n} \sum x_i x_i^\top$$

Given some vector u, the transformed variables

$$y_i = u^\top x_i$$

have

- **mean** 0:
$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i = \frac{1}{n} \sum_{i=1}^{n} u^\top x_i = \frac{1}{n} u^\top \sum_{i=1}^{n} x_i = 0$$
as the mean of the x_i is 0.
- **sample covariance matrix**
$$u^\top S u$$
as
$$\frac{1}{n} \sum_{i=1}^{n} y_i^2 = \frac{1}{n} \sum_{i=1}^{n} u^\top x_i x_i^\top u = \frac{1}{n} u^\top \sum_{i=1}^{n} x_i x_i^\top u = u^\top S u$$

4.1.2 First principal component

We would like to find the u which maximises the sample variance, $u^\top S u$ over unit vectors u, i.e., vectors with $||u|| = 1$. Why do we focus on unit vectors? If we don't, we could make the variance as large as we like, e.g., if we replace u by $10u$ it would increase the variance by a factor of 100. Thus, we constrain the problem and only consider unit vectors for u.

We know from Proposition 3.7 in Section 3.4 that v_1, the first eigenvector of S (also the first right singular vector of X), maximizes $u^\top S u$ with

$$\max_{u:||u||=1} u^\top S u = v_1 S v_1 = \lambda_1$$

where λ_1 is the largest eigenvalue of S.

So the first principal component of X is v_1, and the first transformed variable (sometimes called a principal component score) is $y_1 = v_1^\top x$. Applying this to each data point we get n instances of this new variable

$$y_{i1} = v_1^\top x_i.$$

A note on singular values: We know $S = \frac{1}{n} X^\top X$ and so the eigenvalues of S are the same as the squared singular values of $\frac{1}{\sqrt{n}} X$:

$$\sqrt{\lambda_1} = \sigma_1 \left(\frac{1}{\sqrt{n}} X \right)$$

If we scale X by a factor c, then the singular values are scaled by the same amount, i.e.,

$$\sigma_i(cX) = c\sigma_i(X)$$

and in particular

$$\sigma_i\left(\frac{1}{\sqrt{n}}X\right) = \frac{1}{\sqrt{n}}\sigma_i(X)$$

We will need to remember this scaling if we use the SVD of X to do PCA. Note that scaling X does not change the singular vectors/principal components.

4.1.3 Second principal component

y_1 is the transformed variable that has maximum variance. What should we choose to be our next transformed variable, i.e., what u_2 should we choose for $y_2 = u_2^\top x$? It makes sense to choose y_2 to be uncorrelated with y_1, as otherwise it contains some of the same information given by y_1. The sample covariance between y_1 and $u_2^\top x$ is

$$\begin{aligned} s_{y_2 y_1} &= \frac{1}{n}\sum_{i=1}^{n} u_2^\top x_i x_i^\top v_1 \\ &= u_2^\top S v_1 \\ &= \lambda_1 u_2^\top v_1 \text{ as } v_1 \text{ is an eigenvector of } S \end{aligned}$$

So to make y_2 uncorrelated with y_1 we have to choose u_2 to be orthogonal to v_1, i.e., $u_2^\top v_1 = 0$. So we choose u_2 to be the solution to the optimization problem

$$\max_u u^\top S u \text{ subject to } u^\top v_1 = 0.$$

The solution to this problem is to take $u_2 = v_2$, i.e., the second eigenvector of S (or second right singular vector of X), and then

$$v_2^\top S v_2 = \lambda_2.$$

We'll prove this result in the next section.

Later principal components

Our first transformed variable is

$$y_{i1} = v_1^\top x_i$$

and our second transformed variable is

$$y_{i2} = v_2^\top x_i.$$

At this point, you can probably guess that the j^{th} transformed variable is going to be

$$y_{ij} = v_j^\top x_i.$$

where v_j is the j^{th} eigenvector of S.

- The transformed variables y_i are the **principal component scores**. y_1 is the first score etc.
- The eigenvectors/right singular vectors are sometimes refered to as the **loadings** or simply as the **principal components**.

4.1.4 Geometric interpretation

We think of PCA as projecting the data points x onto a subspace V. The basis vectors for this subspace are the eigenvectors of S, which are the same as the right singular vectors of X (the loadings):

$$V = \text{span}\{v_1, ..., v_r\}.$$

The orthogonal projection matrix (see Section 2.3.3.1) for projecting onto V is

$$P_V = VV^\top$$

as $V^\top V = \mathbf{I}$.
The coordinates of the data points projected onto V (with respect to the basis for V) are the **principal component scores**:

$$y_i = \begin{pmatrix} y_{i1} \\ \vdots \\ y_{ir} \end{pmatrix} = V^\top x_i$$

where

$$V = \begin{pmatrix} | & & | \\ v_1 & ... & v_r \\ | & & | \end{pmatrix}$$

is the matrix of right singular vectors from the SVD of X. The transformed variables are

$$Y = \begin{pmatrix} - & y_1^\top & - \\ - & .. & - \\ - & y_n^\top & - \end{pmatrix} = XV.$$

Substituting the SVD for $X = U\Sigma V^\top$ we can see the transformed variable matrix/principal component scores are

$$Y = U\Sigma.$$

Y is a $n \times r$ matrix, and so if $r < p$ we have reduced the dimension of X, keeping the most important parts of the data

4.1.5 Example

We consider the marks of $n = 10$ students who studied G11PRB and G11STA.

student	PRB	STA
1	81	75
2	79	73
3	66	79
4	53	55
5	43	53
6	59	49
7	62	72
8	79	92
9	49	58
10	55	56

These data haven't been column centered, so let's do that in R. You can do it using the centering matrix as previously, but here is a different approach:

```
secondyr <- data.frame(
  student = 1:10,
PRB=c(81 , 79 , 66 , 53 , 43 , 59 , 62 , 79 , 49 , 55),
STA =c(75 , 73 , 79 , 55 , 53 , 49 , 72 , 92 , 58 , 56)
        )
xbar <- colMeans(secondyr[,2:3]) #only columns 2 and 3 are data
X <- as.matrix(sweep(secondyr[,2:3], 2, xbar) )
```

PRB	STA
18.4	8.8
16.4	6.8
3.4	12.8
-9.6	-11.2
-19.6	-13.2
-3.6	-17.2
-0.6	5.8
16.4	25.8
-13.6	-8.2
-7.6	-10.2

The sample covariance matrix can be computed in two ways:

```
1/10* t(X)%*%X
```

```
##         PRB    STA
```

```
## PRB 162.04 135.38
## STA 135.38 175.36
```

```
cov(X)*9/10
```

```
##        PRB    STA
## PRB 162.04 135.38
## STA 135.38 175.36
```

```
# Remember R uses the unbiased factor 1/(n-1),
# so the 9/10=(n-1)/n changes this to 1/n
# to match the notes
```

We can find the singular value decomposition of X using R

```
(X_svd = svd(X))
```

```
## $d
## [1] 55.15829 18.20887
##
## $u
##               [,1]        [,2]
##  [1,] -0.34556317 -0.39864295
##  [2,] -0.29430029 -0.39482564
##  [3,] -0.21057607  0.34946080
##  [4,]  0.26707104 -0.04226416
##  [5,]  0.41833934  0.27975879
##  [6,]  0.27085156 -0.50812066
##  [7,] -0.06865802  0.24349429
##  [8,] -0.54378479  0.32464825
##  [9,]  0.27768146  0.23043980
## [10,]  0.22893893 -0.08394852
##
## $v
##            [,1]       [,2]
## [1,] -0.6895160 -0.7242705
## [2,] -0.7242705  0.6895160
```

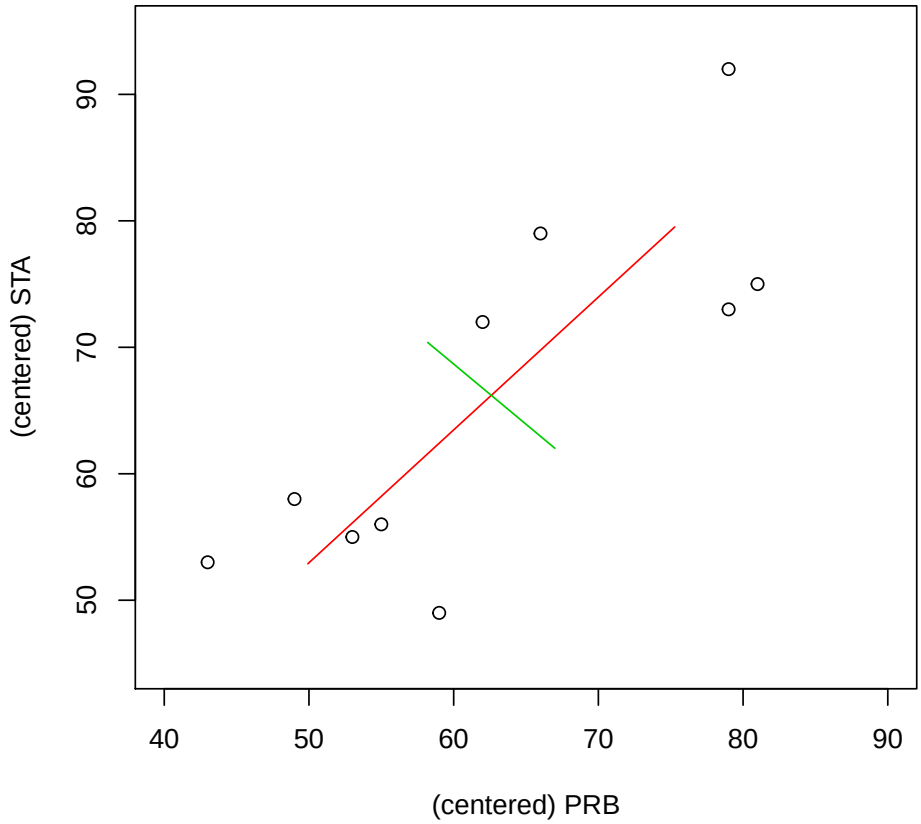
So we can see that the eigenvectors/right singular vectors/loadings are

$$v_1 = \begin{pmatrix} -0.69 \\ -0.724 \end{pmatrix}, \qquad v_2 = \begin{pmatrix} -0.724 \\ 0.69 \end{pmatrix}$$

Sometimes the new variables have an obvious interpretation. In this case the first PC gives approximately equal weight to PRB and STA and thus represents some form of negative ''average" mark. Note that the singular vectors are only determined upto multiplication by ± 1. In this case, R has chosen v_1 to have negative entries, but we could multiply v_1 by -1 so that the first PC was more

like the avearge. As it is, a student that has a high mark on PRB and STA will have a low negative value for y_1. The second PC, meanwhile, represents a contrast between PRB and STA. For example, a large positive value for y_2 implies the student did much better on STA than PRB, and a large negative value implies the opposite.

If we plot the data along with the principal components. The two lines, centred on $\bar{x}$, are in the direction of the principal components/eigenvectors, and their lengths are $2\sqrt{\lambda_j}$, $j = 1, 2$. We can see that the first PC is in the direction of greatest variation (shown in red), and that the second PC (shown in green) is orthogonal to the first PC.



We can find the transformed variables by computing either XV or $U\Sigma$

```
X %*% X_svd$v
```

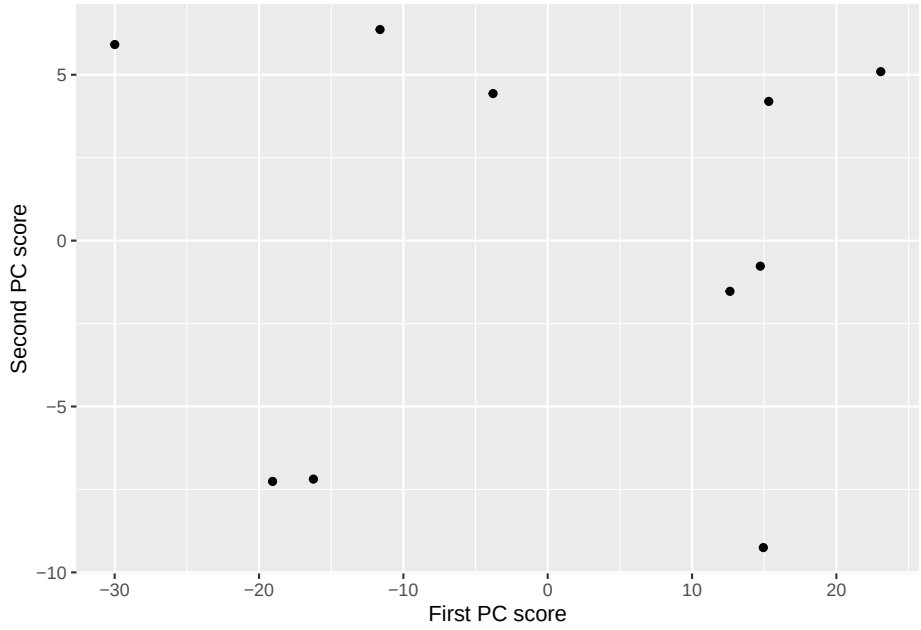
```
##              [,1]       [,2]
##  [1,] -19.060674 -7.2588361
##  [2,] -16.233101 -7.1893271
##  [3,] -11.615016  6.3632849
##  [4,]  14.731183 -0.7695824
```

```
##  [5,]  23.074883  5.0940904
##  [6,]  14.939710 -9.2523011
##  [7,]  -3.787059  4.4337549
##  [8,] -29.994240  5.9114764
##  [9,]  15.316435  4.1960474
## [10,]  12.627880 -1.5286074
```

```
X_svd$u %*% diag(X_svd$d)
```

```
##             [,1]       [,2]
##  [1,] -19.060674 -7.2588361
##  [2,] -16.233101 -7.1893271
##  [3,] -11.615016  6.3632849
##  [4,]  14.731183 -0.7695824
##  [5,]  23.074883  5.0940904
##  [6,]  14.939710 -9.2523011
##  [7,]  -3.787059  4.4337549
##  [8,] -29.994240  5.9114764
##  [9,]  15.316435  4.1960474
## [10,]  12.627880 -1.5286074
```

If we plot the PC scores we can see that the variation is now in line with the new coordinate axes:



R also has a built-in function for doing PCA.

```
pca <- prcomp(secondyr[,2:3]) # prcomp will automatocally remove the column mean
pca$rotation # the loadings
```

```
##            PC1        PC2
## PRB -0.6895160 -0.7242705
## STA -0.7242705  0.6895160
```

```
pca$x # the scores
```

```
##               PC1        PC2
##  [1,] -19.060674 -7.2588361
##  [2,] -16.233101 -7.1893271
##  [3,] -11.615016  6.3632849
##  [4,]  14.731183 -0.7695824
##  [5,]  23.074883  5.0940904
##  [6,]  14.939710 -9.2523011
##  [7,]  -3.787059  4.4337549
##  [8,] -29.994240  5.9114764
##  [9,]  15.316435  4.1960474
## [10,]  12.627880 -1.5286074
```

“‘

Note that the new variables have sample mean $\bar{y} = 0$. The sample covariance matrix is a diagonal with entries given by the eigenvalues (see part 4. of Proposition 4.1). Note that there is always some numerical error (so quantities are never 0, and instead are just very small numnbers).

$$\Lambda = \text{diag}(\lambda_1, \lambda_2) = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

```
colMeans(pca$x)
```

```
##           PC1           PC2
##  2.842171e-15 -9.769963e-16
```

```
cov(pca$x)*9/10 # to convert to using 1/n as the denominator
```

```
##              PC1          PC2
## PC1 3.042437e+02 1.974167e-14
## PC2 1.974167e-14 3.315628e+01
```

Finally, note that we did the singular value decomposition for X above not $\frac{1}{\sqrt{10}}X$, and so we'd need to square and scale the singular values to find the eigenvalues. Let's check:

```
X_svd$d^2/10 # square and scale the singular values
```

```
## [1] 304.24372  33.15628
```

```
eigen(t(X) %*% X/10)$values  # compute the eigenvalues of the covariance matrix
```

```
## [1] 304.24372  33.15628
```

```
svd(X/sqrt(10))$d^2 # compute the singular values of X/sqrt(10) and square
```

```
## [1] 304.24372  33.15628
```

4.1.6 Example: Iris

In general when using R to do PCA, we don't need to compute the SVD and then do the projections, as there is an R command `prcomp` that will do it all for us. The `princomp` will also do PCA, but is less stable than `prcomp`, and it is recommended that you use `prcomp` in preference.

Let's do PCA on the iris dataset discussed in Chapter 1. The `prcomp` returns the square root of the eigenvalues (the standard devaiation of the PC scores), and the PC scores.

```
iris.pca = prcomp(iris[,1:4])
iris.pca$sdev # the square root of the eigenvalues
```

```
## [1] 2.0562689 0.4926162 0.2796596 0.1543862
```

```
head(iris.pca$x)  #the PC scores
```

```
##             PC1        PC2         PC3          PC4
## [1,] -2.684126 -0.3193972  0.02791483  0.002262437
## [2,] -2.714142  0.1770012  0.21046427  0.099026550
## [3,] -2.888991  0.1449494 -0.01790026  0.019968390
## [4,] -2.745343  0.3182990 -0.03155937 -0.075575817
## [5,] -2.728717 -0.3267545 -0.09007924 -0.061258593
## [6,] -2.280860 -0.7413304 -0.16867766 -0.024200858
```

The PC loadings/eigenvectors can also be accessed, as can the sample mean

```
iris.pca$rotation #the eigenvecstors
```

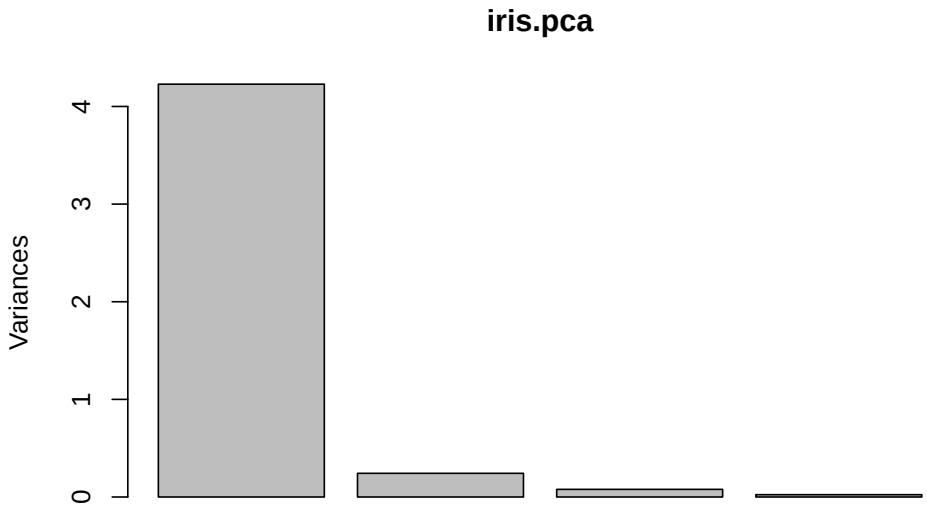
```
##                     PC1         PC2         PC3        PC4
## Sepal.Length  0.36138659 -0.65658877  0.58202985  0.3154872
## Sepal.Width  -0.08452251 -0.73016143 -0.59791083 -0.3197231
## Petal.Length  0.85667061  0.17337266 -0.07623608 -0.4798390
## Petal.Width   0.35828920  0.07548102 -0.54583143  0.7536574
```

```
iris.pca$center # the sample mean of the data
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

A scree plot can be obtained simply by using the `plot` command. The summary command also gives useful information about the importance of each PC.

```
plot(iris.pca)
```

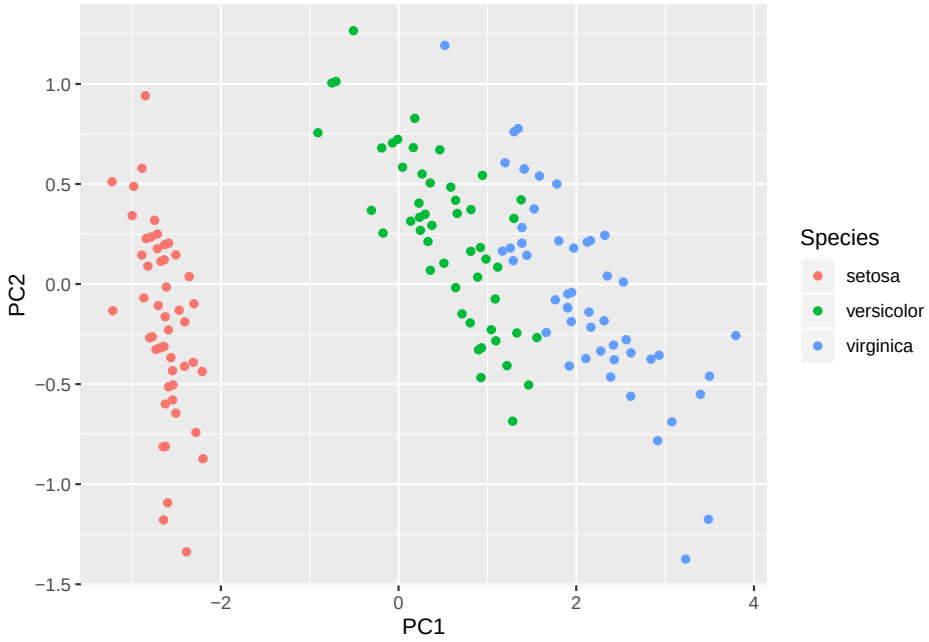



```
summary(iris.pca)
```

```
## Importance of components:
##                           PC1     PC2    PC3     PC4
## Standard deviation     2.0563 0.49262 0.2797 0.15439
## Proportion of Variance 0.9246 0.05307 0.0171 0.00521
## Cumulative Proportion  0.9246 0.97769 0.9948 1.00000
```

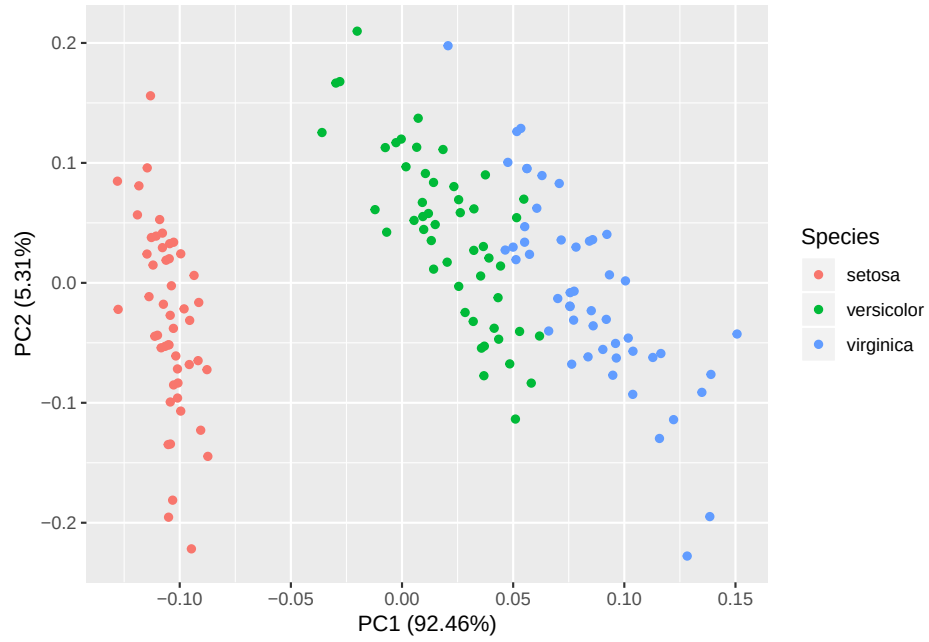
To plot the PC scores, you can either manually create a plot or use the `ggfortify` package. For example, here is a plot of the first two PC scores coloured according to the species of iris.

```
iris$PC1=iris.pca$x[,1]
iris$PC2=iris.pca$x[,2]
qplot(PC1, PC2, colour=Species, data=iris)
```



The `ggfortify` package provides a nice wrapper for some of this functionality.

```
library(ggfortify)
autoplot(iris.pca, data = iris, colour = 'Species')
```



4.2 PCA: a formal description with proofs

Let's now summarize what we've said so far and prove some results about principal component analysis.

Let $x_1, \dots, x_n$ denote a sample of vectors in $\mathbb{R}^p$ with sample mean vector $\bar{x}$ and sample covariance matrix S. Suppose $S = X^\top HX$ has spectral decomposition (see Proposition 3.3)

$$S = V\Lambda V^\top = \sum_{j=1}^{p} \lambda_j v_j v_j^\top, \tag{4.3}$$

where the eigenvalues are $\lambda_1 \geq \lambda_2 \geq \lambda_p \geq 0$ with $\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_p\}$, and V contains the eigenvectors of S.

The principal components of X are defined sequentially. The j^{th} principal component is the solution to the following optimization problem:

$$\max_{u:\, ||u||=1} u^\top Su \tag{4.4}$$

subject to

$$v_k^\top u = 0, \qquad k = 1, \dots, j-1. \tag{4.5}$$

(for $j = 1$ there is no orthogonality constraint).

Proposition 4.1. *The maximum of Equation* (4.4) *subject to Equation* (4.5) *is equal to* λ_j *and is obtained when* $u = v_j$.

Proof. We can prove this using the method of Lagrange multipliers. For $j = 1$ our objective is

$$\mathcal{L} = u^\top Su + \lambda(1 - u^\top u)$$

Differentiating (see 2.1.4) with respect to u and setting the derivative equal to zero gives

$$2Su - 2\lambda u = 0$$

Rearranging we see that u must satify

$$Su = \lambda u \text{ with } u^\top u = 1$$

i.e., u is a unit eigenvector of S. Substituting this back in to the objective we see

$$uSu = \lambda$$

and so we must choose $u = v_1$, the eigenvector corresponding to the largest eigenvalue of S.

We now proceed inductively and assume the result is true for $k = 1, \dots, j-1$. The Lagrangian for the j^{th} optimization problem is

$$\mathcal{L} = u^\top Su + \lambda(1 - u^\top u) + \sum_{k=1}^{j-1} \mu_k (1 - u^\top v_k)$$

where we now have j Lagrange multipliers $\lambda, \mu_1, \ldots, \mu_{j-1}$ - one for each constraint. Differentiating with respect to u and setting equal to zero gives

$$0 = 2Su - 2\lambda u - \sum_{k=1}^{j-1} \mu_k v_k = 0$$

If we left multiply by v_l we get

$$2v_l^\top Su - \lambda v_l u - \sum \mu_k v_l^\top v_k = 0$$

We know v_l is an eigenvector of S and so $Sv_l = \lambda_l v_l$ and hence $v_k Su = 0$ as $v_l^\top u = 0$. Also

$$v_l^\top v_k = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{otherwise,} \end{cases}$$

and thus we've shown that $\mu_l = 0$ for $l = 1, \ldots, j-1$. So again we have that

$$Su = \lambda u$$

i.e., u must be a unit eigenvector of S. It only remains to show *which* eigenvector it is. Because u must be orthogonal to $v_1, \ldots, v_{j-1}$, and as $v_l^\top Sv_l = \lambda_l$, we must choose $u = v_j$, the eigenvector corresponding to the j^{th} largest eigenvalue. □

4.2.1 Properties of principal components

For $j = 1, \ldots, p$, the scores of the j^{th} principal component (PC) are given by

$$y_{ij} = v_j^\top (x_i - \bar{x}), \qquad i = 1, \ldots, n.$$

The j^{th} eigenvector v_j is sometimes referred to as the vector of **loadings** for the j^{th} PC.

In vector notation

$$y_i = (y_{i1}, y_{i2}, \ldots, y_{ip})^\top = V^\top (x_i - \bar{x}), \qquad i = 1, \ldots, n.$$

In matrix form, the full set of PC scores is given by

$$Y = [y_1, \ldots, y_n]^\top = HXV.$$

If $\tilde{X} = HX$ is the column centered data matrix, with singular value decomposition $\tilde{X} = U\Sigma V^\top$ with V as in Equation (4.3), then

$$Y = \tilde{X}V = U\Sigma.$$

The transformed variables $y = HXV$ have some important properties which we collect together in the following proposition.

Proposition 4.2. *The following results hold:*

1. *The sample mean vector of* $y_1, \dots, y_n$ *is the zero vector:* $\bar{y} = \mathbf{0}_p$
2. *The sample covariance matrix of* $y_1, \dots, y_n$ *is*

$$\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_p)$$

   i.e., for each fixed j*, the sample variance of* y_{ij} *is* λ_j*, and* y_{ij} *is uncorrelated with with* y_{ik} *for* $j \neq k$*.*

3. *For* $j \leq k$ *the sample variance of* $\{y_{ij}\}_{i=1,\dots,n}$ *is greater than or equal to the sample variance of* $\{y_{ik}\}_{i=1,\dots,n}$*.*

$$q_1^\top S q_1 \geq q_2^\top S q_2 \geq \dots \geq q_p^\top S q_p \geq 0$$

4. *The sum of the sample variances is equal to the trace of* S

$$\sum_{j=1}^{p} q_j^\top S q_j = \sum_{j=1}^{p} \lambda_j = tr(S)$$

5. *The product of the sample variances is equal to the determinant of* S

$$\prod_{j=1}^{p} q_j^\top S q_j = \prod_{j=1}^{p} \lambda_j = |S|.$$

Proof. For i.

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} V^\top (x_i - \bar{x}) = \frac{1}{n} V^\top \sum_{i=1}^{n} (x_i - \bar{x}) = 0.$$

For 2. the sample covariance matrix of $y_1, \dots, y_n$ is

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^{n} y_i y_i^\top &= \frac{1}{n} \sum V^\top (x_i - \bar{x})(x_i - x)^\top V \\ &= V^\top S V \\ &= V^\top V \Lambda V^\top V \text{ substiting the spectral decomposition for } S \\ &= \Lambda \end{aligned}$$

3. is a consequence 2. and of ordering the eigenvalues in decreasing magnitude.
4. follows from lemma 2.1 and the spectral decomposition of S:

$$\operatorname{tr}(S) = \operatorname{tr}(V \Lambda V^\top) = \operatorname{tr}(V^\top V \Lambda) = \operatorname{tr}(\Lambda) = \sum \lambda_i$$

5. follows from 3.2.

□

From these properties we say that a proportion

$$\frac{\lambda_j}{\lambda_1 + ... + \lambda_p}$$

of the variability in the sample is 'explained' by the j^{th} PC.

One tool for looking at the contributions of each PC is to look at the **scree plot** which plots the percentage of variance explained by PC j against j. We'll see examples of scree plots below.

4.2.2 Example: Football

We can apply PCA to a football league table where W, D, L are the number of matches won, drawn and lost and G and GA are the goals scored for and against, and GD is the goal difference $(G - GA)$. An extract of the table for a recent Premiership season is:

Team	W	D	L	G	GA	GD
Liverpool	32	3	3	85	33	52
Manchester City	26	3	9	102	35	67
Manchester United	18	12	8	66	36	30
Chelsea	20	6	12	69	54	15
Leicester City	18	8	12	67	41	26
Tottenham Hotspur	16	11	11	61	47	14
Wolverhampton	15	14	9	51	40	11
Arsenal	14	14	10	56	48	8
Sheffield United	14	12	12	39	39	0
Burnley	15	9	14	43	50	-7

The sample mean vector is

$$\bar{x} = \begin{pmatrix} 14.4 \\ 9.2 \\ 14.4 \\ 51.7 \\ 51.7 \\ 0 \end{pmatrix}.$$

Note that the total goals scored must equal the total goals conceded, and that the sum of the goal differences must be 0. The sample covariance matrix is

$$S = \begin{pmatrix} 38.3 & -9.18 & -29.2 & 103 & -57 & 160 \\ -9.18 & 10.2 & -0.98 & -27.5 & -2.24 & -25.2 \\ -29.2 & -0.98 & 30.1 & -75.3 & 59.3 & -135 \\ 103 & -27.5 & -75.3 & 336 & -147 & 483 \\ -57 & -2.24 & 59.3 & -147 & 134 & -281 \\ 160 & -25.2 & -135 & 483 & -281 & 764 \end{pmatrix} \tag{4.6}$$

The eigenvalues of S are

$$\Lambda = \text{diag}\begin{pmatrix} 1300 & 71.9 & 8.05 & 4.62 & -2.65e-14 & -3.73e-14 \end{pmatrix}$$

Note that we have two zero eigenvalues (which won't be computed as exactly zero because of numerical rounding errors) because two of our variables are a linear combinations of the other variables, $W + D + L = 38$ and $GD = G - GA$. The corresponding eigenvectors are

$$V = [v_1 \dots v_6] = \begin{pmatrix} 0.166 & -0.0262 & 0.707 & 0.373 & -0.577 & 0 \\ -0.0282 & 0.275 & -0.661 & 0.391 & -0.577 & 1.06e-14 \\ -0.138 & -0.249 & -0.0455 & -0.764 & -0.577 & -2.04e-14 \\ 0.502 & -0.6 & -0.202 & 0.117 & 3.22e-15 & -0.577 \\ -0.285 & -0.701 & -0.11 & 0.286 & -6.11e-15 & 0.577 \\ 0.787 & 0.101 & -0.0915 & -0.169 & -3.33e-16 & 0.577 \end{pmatrix}$$

The proportion of variability explained by each of the PCs is:

$$\begin{pmatrix} 0.939 & 0.052 & 0.00583 & 0.00334 & -1.92e-17 & -2.7e-17 \end{pmatrix}$$

There is no point computing the scores for PC 5 and 6, because these do not explain any of the variability in the data. Similarly, there is little value in computing the scores for PCs 3 & 4 because they account for less than 1% of the variability in the data.

We can, therefore, choose to compute only the first two PC scores. We are reducing the dimension of our data set from $p = 5$ to $p = 2$ while still retaining 99% of the variability. The first PC score/transformed variable is given by:

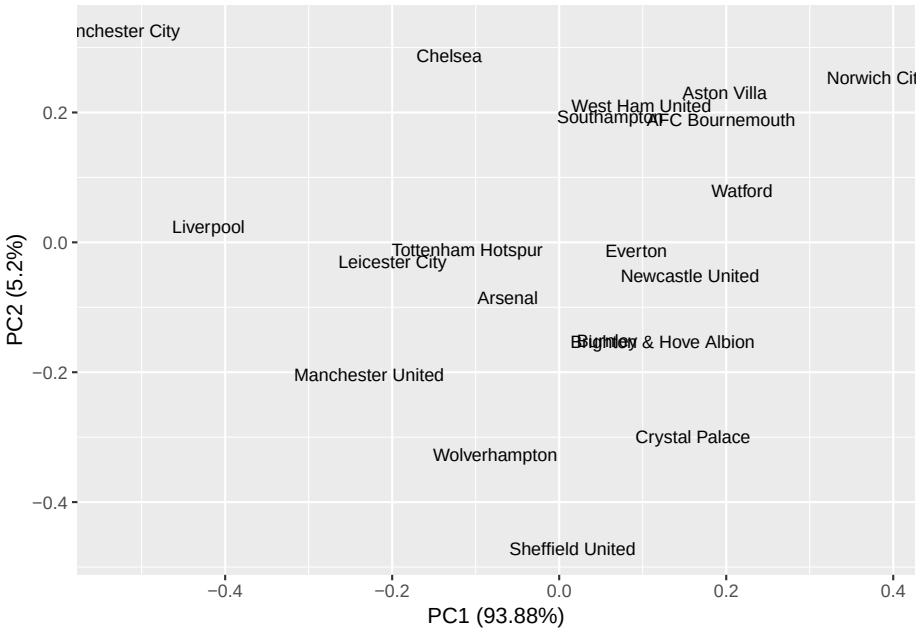
$$\begin{aligned} y_{i1} = {} & 0.17(W_i - \bar{W}) + -0.03(D_i - \bar{D}) + -0.14(L_i - \bar{L}) \\ & + 0.5(G_i - \bar{G}) + -0.28(GA_i - \bar{GA}) + 0.79(GD_i - \bar{GD}), \end{aligned}$$

and similarly for PC 2.

The first five rows of our revised "league table" are now

Team	PC1	PC2
Liverpool	-67.6	0.9
Manchester City	-85.6	12.3
Manchester United	-36.7	-7.7
Chelsea	-21.2	10.9
Leicester City	-32.2	-1.1

Now that we have reduced the dimension to $p = 2$, we can visualise the differences between the teams.



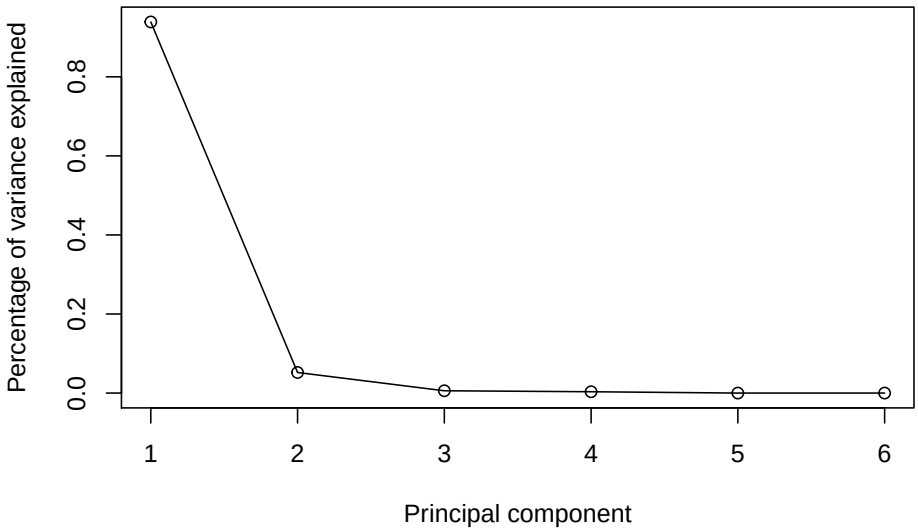
We might interpret the PCs as follows. The first PC seems to measure the difference in goals scored and conceded between teams. It rewards teams with 0 for positive goal difference, and 0.37 for each goal scored, whilst penalising them by -0.58 for every goal they concede. So a team with a large positive PC1 score tends to score lots of goals and concede few. If we rank teams by their PC1 score, and compare this with the rankings using 3 points for a win and 1 point for a draw we get a different ranking of the teams.

	PC1	PC2
Liverpool	-67.637127	0.9306615
Manchester City	-85.593967	12.3492387
Manchester United	-36.661797	-7.7344979
Chelsea	-21.190382	10.9021516
Leicester City	-32.155292	-1.1285032
Tottenham Hotspur	-17.710519	-0.4325749
Wolverhampton	-12.342929	-12.3850152
Arsenal	-9.913433	-3.2498502
Sheffield United	2.580462	-17.9013025
Burnley	9.235955	-5.7314925

The second PC has a strong negative loading for both goals for and against. A team with a large negative PC 2 score was, therefore, involved in matches with lots of goals. We could, therefore, interpret PC 2 as an ''entertainment" measure, ranking teams according to their involvement in high-scoring games.

The above example raises the question of how many PCs should we use in practice. If we reduce the dimension to $p = 1$ then we can rank observations and analyse our new variable with univariate statistics. If we reduce the dimension to $p = 2$ then it is still easy to visualise the data. However, reducing the dimension to $p = 1$ or $p = 2$ may involve losing lots of information and a sensible answer should depend on the objectives of the analysis and the data itself.

The scree graph for the football example is:



There are many possible methods for choosing the number of PCs to retain for analysis, including:

- retaining enough PCs to explain, say, 90% of the total variation;

- retaining PCs where the eigenvalue is above the average.

To retain enough PCs to explain 90% of the total variance, would require us to keep just a single PCs in this case.

4.2.3 PCA based on R versus PCA based on S

Recall the distinction between the sample covariance matrix S and the sample correlation matrix R. Note that all correlation matrices are also covariance matrices, but not all covariance matrices are correlation matrices. Before doing PCA we must decide whether to do PCA based on S or R? As we will see later

- PCA based on R (but not S) is scale invariant, whereas
- PCA based on S is invariant under orthogonal rotation.

If the original p variables represent very different types of quantity or show marked differences in variances, then it will usually be better to use R rather than S. However, in some circumstances, we may wish to use S, such as when the p variables are measuring similar entities and the sample variances are not too different.

Given that the required numerical calculations are easy to perform in R, we might wish to do it both ways and see if it makes much difference. To use the correlation matrix R, we just add the option `scale=TRUE` when using the `prcomp` command.

4.2.3.1 Football example continued

If we repeat the analysis of the football data using R instead of S, we get find principal components:

$$\Lambda = \text{diag}\begin{pmatrix} 4.51 & 1.25 & 0.156 & 0.0863 & 3.68e-32 & 2.48e-33 \end{pmatrix}$$

$$V = [v_1 \dots v_6] = \begin{pmatrix} -0.456 & 0.149 & -0.342 & -0.406 & 0.466 & 0.52 \\ 0.143 & -0.844 & 0.344 & -0.143 & 0.24 & 0.268 \\ 0.432 & 0.321 & 0.186 & 0.541 & 0.413 & 0.461 \\ -0.438 & 0.214 & 0.7 & -0.0181 & 0.389 & -0.348 \\ 0.419 & 0.342 & 0.386 & -0.671 & -0.245 & 0.22 \\ -0.466 & -0.00136 & 0.302 & 0.269 & -0.586 & 0.525 \end{pmatrix}$$

The effect of using R is to standardize each of the original variables to have variance 1. The first PC now has loadings which are more evenly balanced across the 6 original variables.

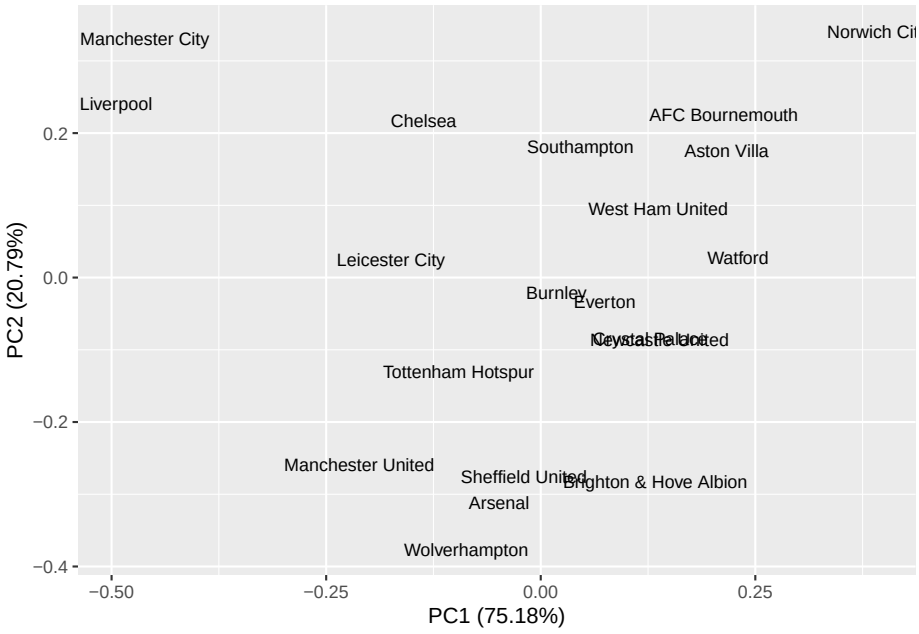
Teams will have a small value of PC1 score if they won lots, lost rarely, scored a lot, and conceded rarely. In other words, PC1 is a complete measure of overall

performance. If we look at the league table based on ordering according to PC1 we get a table that looks more like the original table.

	PC1	PC2
Liverpool	-4.6996438	1.2003675
Manchester City	-4.3806921	1.6517491
Manchester United	-2.0067554	-1.2938783
Chelsea	-1.2946867	1.0826790
Leicester City	-1.6563468	0.1216950
Tottenham Hotspur	-0.9093467	-0.6509021
Wolverhampton	-0.8242372	-1.8777265
Arsenal	-0.4606630	-1.5565564
Sheffield United	-0.1849220	-1.3791244
Burnley	0.1752204	-0.1047675

Overall for these data, doing PCA with R instead of S better summarizes the data (although this is just my subjective opinion - you may feel differently).

```
library(ggfortify)
autoplot(prem.pca, data = table,  label = TRUE,  label.size = 3, shape=FALSE)
```



4.2.4 Population PCA

So far we have considered sample PCA based on the sample covariance matrix or sample correlation matrix:

$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top.$$

We note now that there is a *population* analogue of PCA based on the population covariance matrix Σ. Although the population version of PCA is not of as much direct practical relevance as sample PCA, it is nevertheless of conceptual importance.

Let x denote a $p \times 1$ random vector with $\mathbb{E}(x) = \boldsymbol{\mu}$ and $\mathbb{V}\mathrm{ar}(x) = \boldsymbol{\Sigma}$. As defined, $\boldsymbol{\mu}$ is the population mean vector and $\boldsymbol{\Sigma}$ is the population covariance matrix.

Since $\boldsymbol{\Sigma}$ is symmetric, the spectral decomposition theorem tells us that

$$\boldsymbol{\Sigma} = \sum_{j=1}^{p} \check{\lambda}_j \check{v}_j \check{v}_j^\top = \check{V}\check{\Lambda}\check{V}^\top$$

where the 'check' symbol $\check{}$ is used to distinguish population quantities from their sample analogues.

Then:

- the first population PC is defined by $Y_1 = \check{v}_1^\top(x - \boldsymbol{\mu})$;
- the second population PC is defined by $Y_2 = \check{v}_2^\top(x - \boldsymbol{\mu})$;
- ...
- the pth population PC is defined by $Y_p = \check{v}_p^\top(x - \boldsymbol{\mu})$.

The $Y_1, \ldots, Y_p$ are random variables, unlike the sample PCA case, where the y_{ij} are observed quantities. In the sample PCA case, the y_{ij} can often be regarded as the observed values of random variables.

In matrix form, the above definitions can be summarised by writing

$$y = \begin{pmatrix} Y_1 \\ Y_2 \\ \ldots \\ \ldots \\ Y_p \end{pmatrix} = \check{V}^\top(x - \boldsymbol{\mu}).$$

The population PCA analogues of the sample PCA properties listed in Proposition 4.2 are now given. Note that the Y_j's are random variables as opposed to observed values of random variables.

Proposition 4.3. *The following results hold for the random variables $Y_1, \ldots, Y_p$ defined above.*

1. $\mathbb{E}(Y_j) = 0$ *for* $j = 1, \ldots, p$;
2. $\mathbb{V}\mathrm{ar}(Y_j) = \check{\lambda}_j$ *for* $j = 1, \ldots, p$;
3. $\mathbb{C}\mathrm{ov}(Y_j, Y_k) = 0$ *if* $j \neq k$;
4. $\mathbb{V}\mathrm{ar}(Y_1) \geq \mathbb{V}\mathrm{ar}(Y_2) \geq \cdots \geq \mathbb{V}\mathrm{ar}(Y_p) \geq 0$;
5. $\sum_{j=1}^{p} \mathbb{V}\mathrm{ar}(Y_j) = \sum_{j=1}^{p} \check{\lambda}_j = tr(\Sigma)$;
6. $\prod_{j=1}^{p} Var(Y_j) = \prod_{j=1}^{p} \check{\lambda}_j = |\Sigma|$.

Note that, defining $y = (Y_1, \ldots, Y_p)^\top$ as before, part 1. implies that $\mathbb{E}(y) = \mathbf{0}_p$ and parts 2. and 3. together imply that

$$\mathrm{Var}(y) = \Lambda \equiv \mathrm{diag}(\check{\lambda}_1, \ldots, \check{\lambda}_p).$$

Consider now a repeated sampling framework in which we assume that $x_1, \ldots, x_n$ are IID random vectors from a population with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ.

What is the relationship between the sample PCA based on the sample of observed vectors $x_1, \ldots, x_n$, and the population PCA based on the unobserved random vector x, from the same population?

If the elements of Σ are all finite, then as n increases, the elements of the sample covariance matrix S will converge to the corresponding elements of the population covariance matrix Σ. Consequently, we expect the principal components from sample PCA to converge to the population PCA values as n grows large. Justification of this statement comes from the weak law of large numbers applied to the components of Σ, but the details are beyond the scope of this module.

4.2.5 PCA under transformations of variables

We'll now consider what happens to PCA when the data are transformed in various ways.

Addition transformation

Firstly, consider the transformation of addition where, for example, we add a fixed amount to each variable. We can write this transformation as $z_i = x_i + c$, where c is a fixed vector. Under this transformation the sample mean changes, $\bar{z} = \bar{x} + c$, but the sample variance remains S. Consequently, the eigenvalues and eigenvectors remain the same and, therefore, so do the principal component scores/transformed vasriables,

$$y_i = V^\top (z_i - \bar{z}) = V^\top (x_i + c - (\bar{x} + c)) = V^\top (x_i - \bar{x}).$$

We say that the principal components are **invariant** under the addition transformation. An important special case is to choose $c = -\bar{x}$ so that the PC scores are simply $y_i = V^\top z_i$.

Scale transformation

Secondly, we consider the scale transformation where each variable is multiplied by a fixed amount. A scale transformation occurs more naturally when we convert units of measurement from, say, metres to kilometres. We can write this transformation as $z_i = Dx_i$, where D is a diagonal matrix with positive elements. Under this transformation the sample mean changes from $\bar{x}$ to $\bar{z} = D\bar{x}$, and the sample covariance matrix changes from S to DSD. Consequently, the principal components also change.

This lack of scale-invariance is undesirable. For example, if we analysed data that included some information on distances, we don't want the answer to depend upon whether we use km, metres, or miles as the measure of distance. One solution is to scale the data using

$$D = \text{diag}(s_{11}^{-1/2}, \ldots, s_{pp}^{-1/2}),$$

where s_{ii} is the ith diagonal element of S. In effect, we have standardised all the new variables to have variance 1. In this case the sample covariance matrix of the z_i's is simply the sample correlation matrix R of the original variables, x_i. Therefore, we can carry out PCA on the sample correlation matrix, R, which is invariant to changes of scale.

In summary: R is scale-invariant while S is not. To do PCA on R in R we use the option `scale=TRUE` in the `prcomp` command.

We saw an example of this in section 4.2.3 with the football data. Because the sample variances of G and GA are much larger than the sample variances of W, D and L, doing PCA with R instead of S completely changed the analysis.

Orthogonal transformations

Thirdly, we consider a transformation by an orthogonal matrix, $\underset{p\times p}{A}$, such that $AA^\top = A^\top A = \mathbf{I}_p$, and write $z_i = Ax_i$. This is equivalent to rotating and/or reflecting the original data.

Let S be the sample covariance matrix of the x_i and let T be the sample covariance matrix of the z_i. Under this transformation the sample mean changes from $\bar{x}$ to $\bar{z} = A\bar{x}$, and the sample covariance matrix S changes from S to $T = ASA^\top$.

However, if we write S in terms of its spectral decomposition $S = V\Lambda V^\top$, then $T = AV\Lambda V^\top A^\top = B\Lambda B^\top$ where $B = AV$ is also orthogonal. It is therefore apparent that the eigenvalues of T are the same as those of S; and the eigenvectors of T are given by b_j where $b_j = Av_j$, $j = 1, \ldots, p$. The PC scores of the rotated variables are

$$y_i = B^\top(z_i - \bar{z}) = V^\top A^\top A(x_i - \bar{x}) = V_1^\top(x_i - \bar{x}),$$

and so they are identical to the PC scores of the original variables.

Therefore, under an orthogonal transformation the eigenvalues and PC scores are unchanged; the PCs are orthogonal transformations of the original PCs. We say that the principal components are **equivariant** with respect to orthogonal transformations.

4.3 An alternative view of PCA

Consider a sample $x_1, \dots, x_n \in \mathbb{R}^p$ with zero mean (replace x_i by $x_i - \bar{x}$ if the mean is not zero). In order to find the r leading principal components, we solve the optimization problem

$$\text{For } k = 1, \dots, r \text{ maximize } u_k^\top S u_k$$
$$\text{subject to } u_k^\top u_j = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise.} \end{cases}$$

We can write this in the form given in the introduction to this chapter (Equation (4.1)) as

$$\begin{aligned} &\text{Maximize } \operatorname{tr}(U^\top S U) \\ &\text{subject to } U^\top U = \mathbf{I}_r, \end{aligned}$$

as $\operatorname{tr}(U^\top S U) = \sum_{k=1}^r u_k^\top S u_k$ if U has columns $u_1, \dots, u_r$.

An equivalent problem

There is another optimization problem that we sometimes wish to solve, that turns out to be equivalent to the above, thus providing another reason why PCA is so widely used.

Suppose we want to find the best rank-r linear approximation to the dataset $\{x_1, \dots, x_n\}$. One way to think about this is seek a $p \times r$ matrix U for which the rank r linear model

$$f(y) = Uy$$

can be used to represent the data.

Let's choose $y_i \in \mathbb{R}^r$ and U to minimize the sum of squared errors

$$\sum_{i=1}^n ||x_i - U y_i||_2^2.$$

If we write

$$Y^\top = \begin{pmatrix} | & & | \\ y_1 & \dots & y_n \\ | & & | \end{pmatrix}$$

then

$$\begin{aligned}\sum_{i=1}^{n} ||x_i - Uy_i||_2^2 &= \text{tr}((X^\top - UY^\top)^\top (X^\top - UY^\top)) \\ &= ||X^\top - UY^\top||_F^2\end{aligned}$$

i.e., we're looking for the rank-r matrix X_r that minimizes $||X - X_r||_F = ||X^\top - X_r^\top||_F$, noting that we can write an arbitrary rank-r matrix as $X_r^\top = UY^\top$ for some $p \times r$ matrix U and a $n \times r$ matrix Y.

It makes sense to restrict the columns of U to be orthonormal so that $U^\top U = \mathbf{I}_r$ as non-orthonormal coordinates systems are confusing. We know that the $u \in \mathcal{C}(U)$ (where $\mathcal{C}(U)$ is the column space of U) that minimizes

$$||x - u||_2$$

is the orthogonal projection of x onto $\mathcal{C}(U)$, which given the columns of U are orthonormal is $u = UU^\top x$ (see Section 2.3.3.1). So we must have $X_r^\top = UU^\top X^\top$ and $Y^\top = U^\top X^\top$.

So it remains to find the optimal choice for U by minimizing

$$\begin{aligned}||X^\top - UU^\top X^\top||_F &= ||X - XUU^\top||_F \\ &= \text{tr}((X - XUU^\top)^\top (X - XUU^\top)) \\ &= \text{tr}(X^\top X) - 2\,\text{tr}(UU^\top X^\top X) + \text{tr}(UU^\top X^\top XUU^\top) \\ &= \text{tr}(X^\top X) - \text{tr}(U^\top X^\top XU)\end{aligned}$$

where we've used the fact $\text{tr}(AB) = \text{tr}(BA)$ and that $U^\top U = \mathbf{I}_r$.

Minimizing the equation above with respect to U is equivalent to maximizing

$$\text{tr}(U^\top SU)$$

which is the maximum variance objective we used to introduce PCA.

So to summarize, the optimization problem

$$\begin{aligned}&\text{Minimize } ||X^\top - UU^\top X^\top||_F \\ &\text{subject to } U^\top U = \mathbf{I}_r,\end{aligned}$$

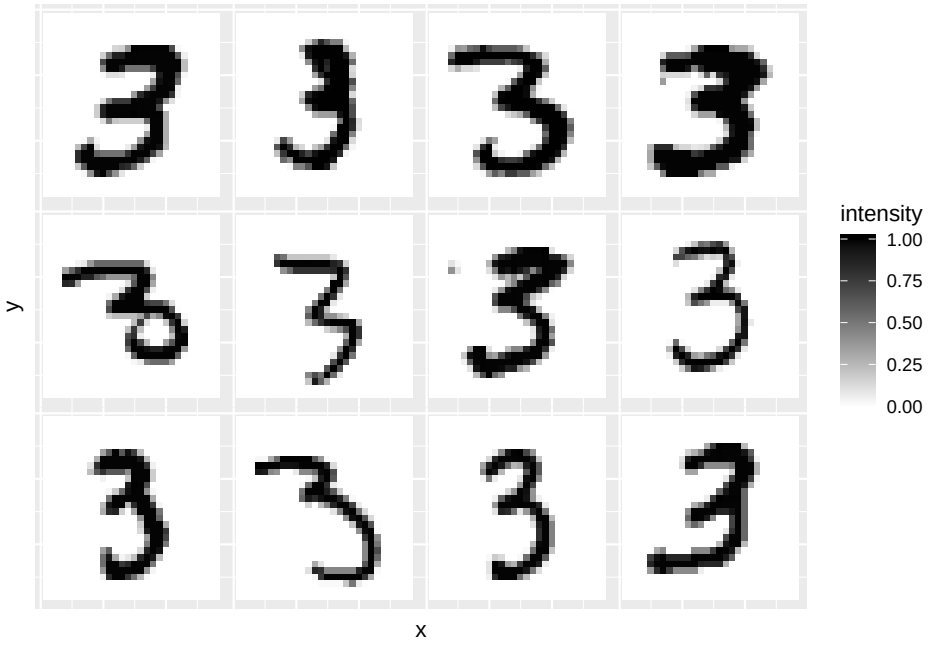
is equivalent to (and has the same as) the PCA optimization problem.

4.3.1 Example: MNIST handwritten digits

Let's consider the MNIST dataset of handwritten digits discussed in Chapter 1. Recall this is a collection of 60,000 digits, each of which has been converted to a 28×28 pixel greyscale image (so $p = 784$). I've made a clean version of

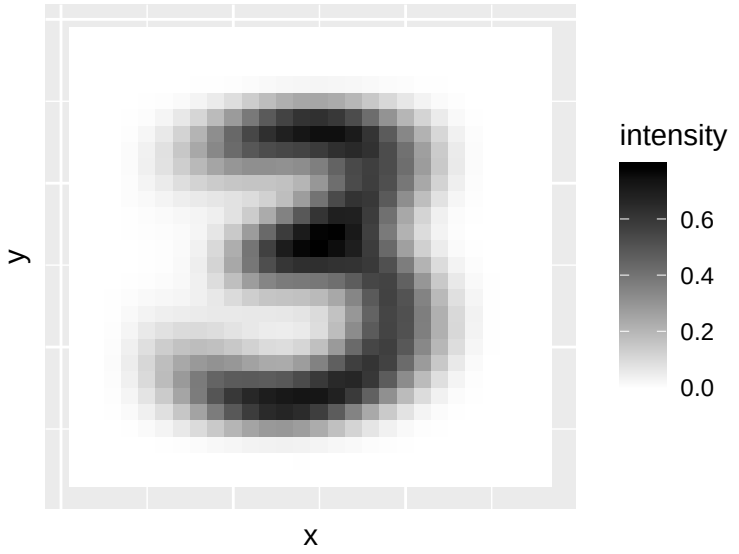
the dataset available on Moodle, so you can try this analysis for yourself. Let's look at just the 3s. I've created a plotting function `plot.mnist`, which is in the code file on Moodle.

```
load(file="mnist.rda")
source('mnisttools.R')
mnist3 = mnist$train$x[mnist$train$y==3,] # select just the 3s
plot.mnist(mnist3[1:12,]) # plot the first 12 images
```



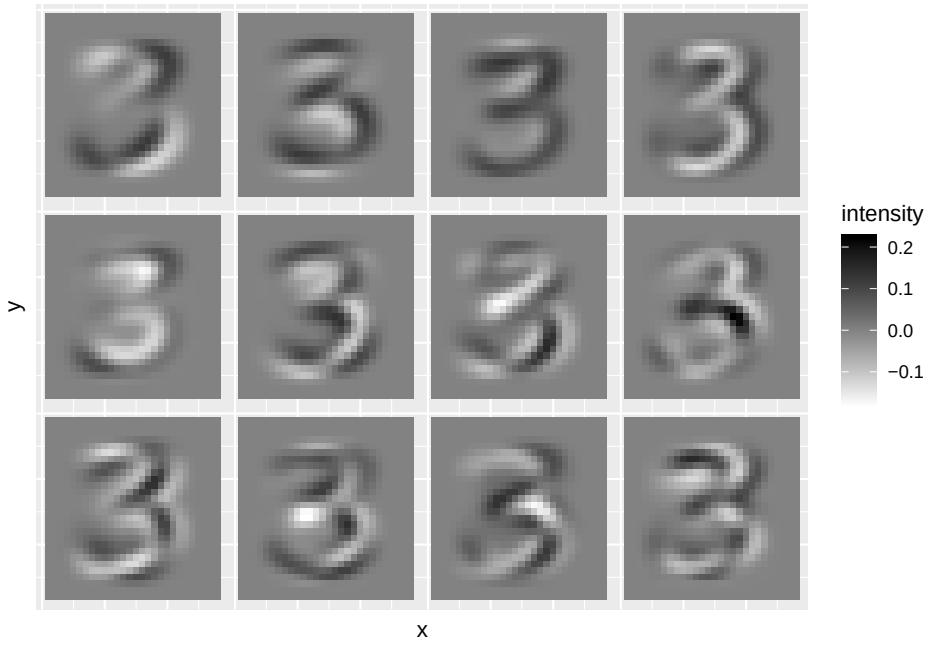
We can see there is quite a bit of variation between them. Now lets look at $\bar{x}$, the average 3.

```
xbar=colMeans(mnist3)
plot.mnist(xbar)
```



We can use the `prcomp` command to find the principal components. Note that we can't use the `scale=TRUE` option as some of the columns are all 0, and so R throws an error as it cannot rescale these to have variance 1. Let's plot the first few principal components/eigenvectors/loading vectors.

```
mnist3.pca <- prcomp(mnist3)
plot.mnist(mnist3.pca$rotation[,1:12])
```

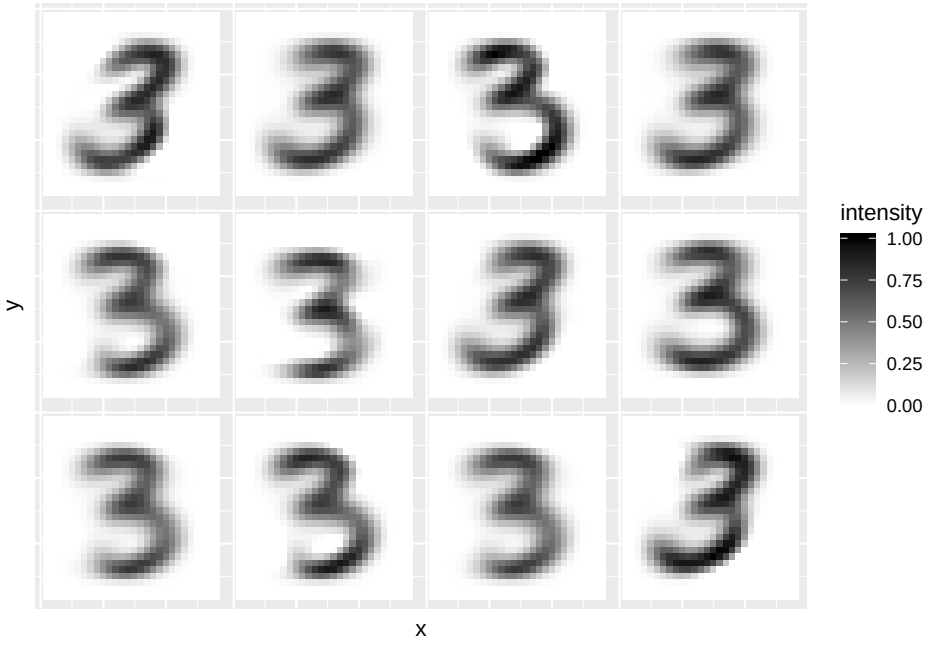


These show the main mode of variability in the 3s. Focusing on the first PC, we can see that this is a form of rotation and causes the 3 to slant either forward or backward. If we wanted a rank-2 approximation to the data we would use

$$f(y) = \bar{x} + y_1 v_1 + y_2 v_2$$

Let's try reconstructing the data with $r = 2$.

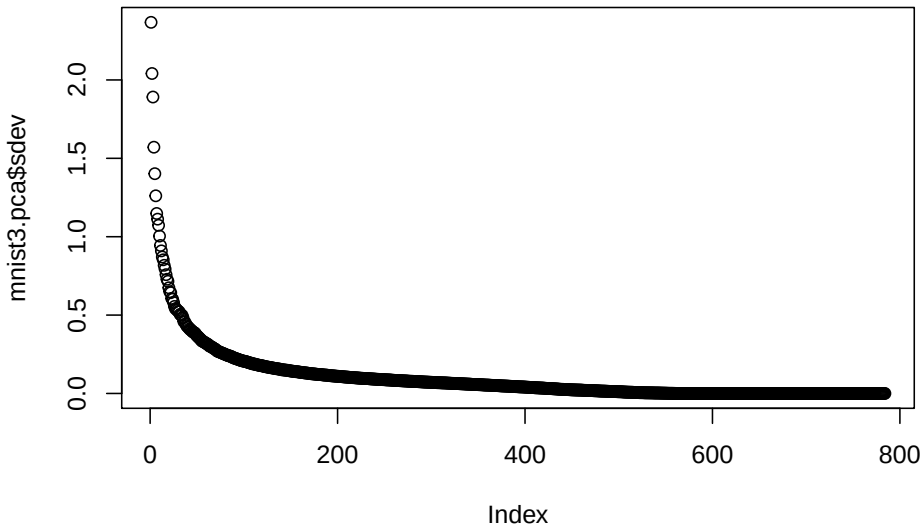
```
r=2
recon =  mnist3.pca$x[,1:r] %*% t(mnist3.pca$rotation[,1:r])
plot.mnist2(matrix(rep(xbar,12), byrow=T, nr=12)+recon[1:12,])
```



We can see that all of these 3s still look a lot like the average 3, but that they vary in their slant, and the heaviness of the line.

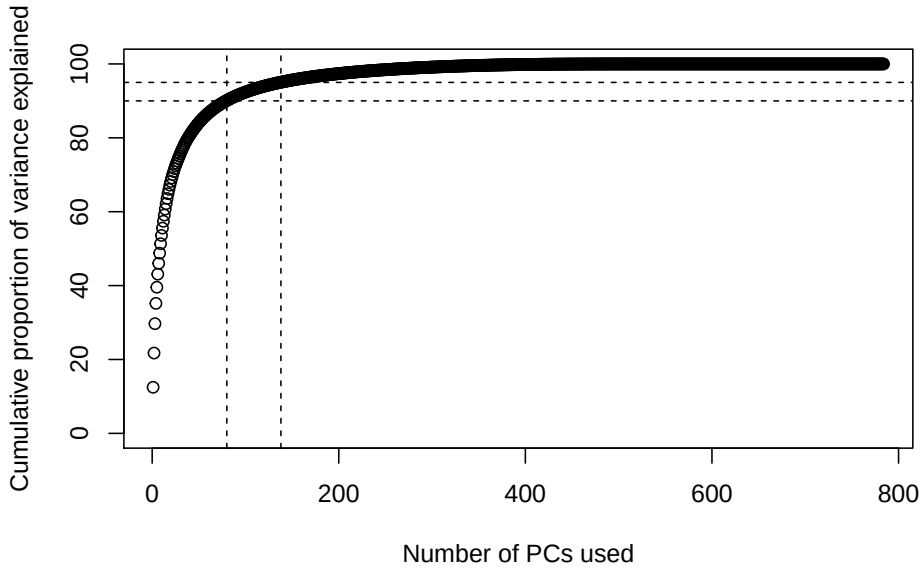
The scree plot shows a sharp decrease in the eigenvalues until about the 100th component, at which point they level off.

```
plot(mnist3.pca$sdev) # scree plot
```



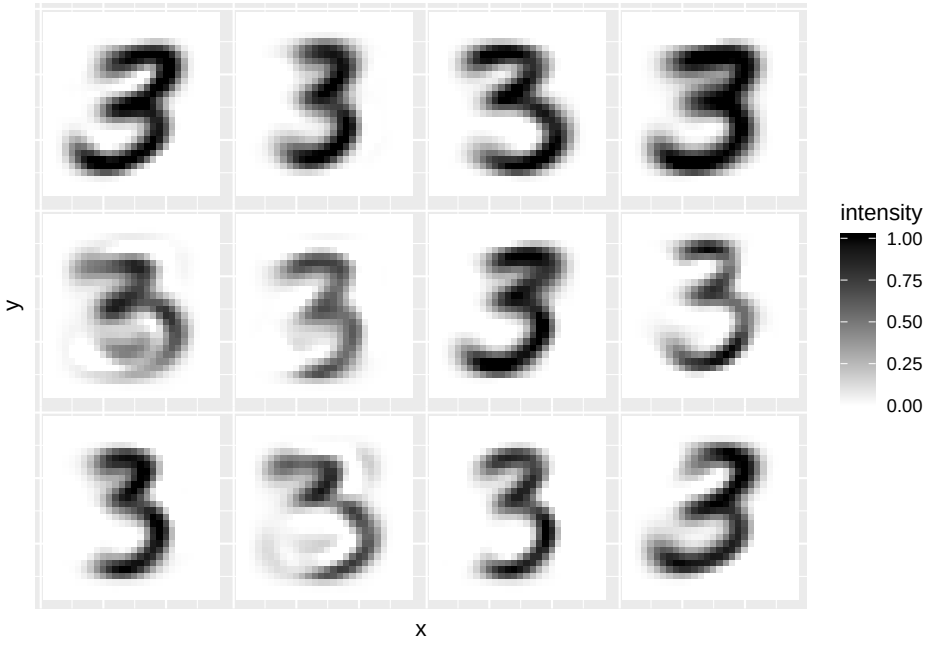
It can also be useful to plot the cumulative sum of the total proportion of variance explained by a given number of principal components. I've drawn on horizontal lines at 90% and 95% of variance explained, to help identify when we cross these thresholds. We need 80 components to explain 90% of the variance, and 138 components to explain 95% of the variance.

```
cumvar = 100*cumsum(mnist3.pca$sdev^2) / sum(mnist3.pca$sdev^2)
plot(cumvar, ylab="Cumulative proportion of variance explained", xlab="Number of PCs us
abline(h=90, lty=2)
abline(v=min(which(cumvar>90)), lty=2)
abline(h=95, lty=2)
abline(v=min(which(cumvar>95)), lty=2)
```

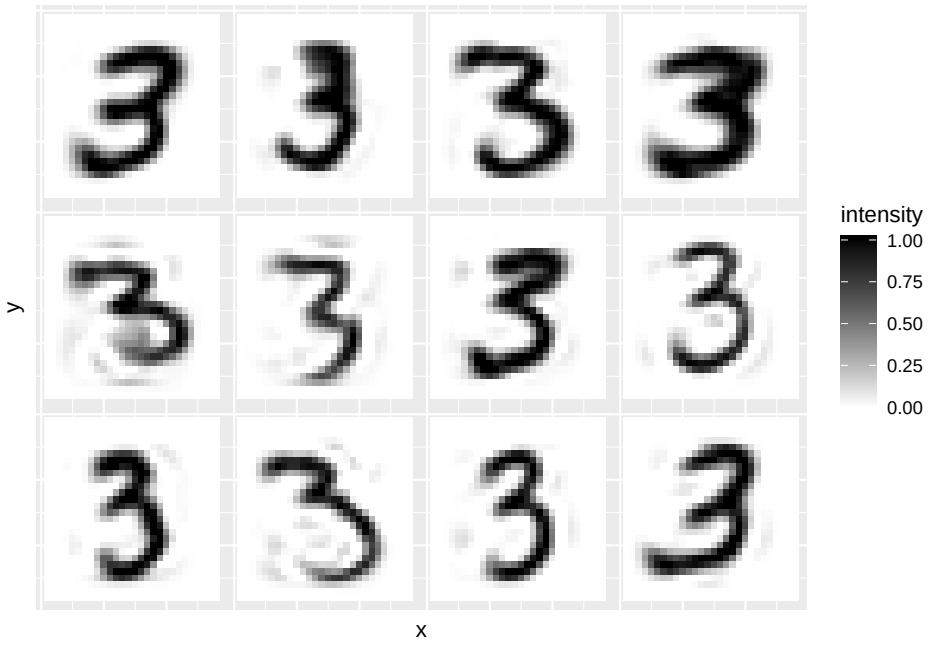


Let's now look at the reconstruction using $r = 10,\ 50,\ 100$ and 500 components to see how the accuracy changes.

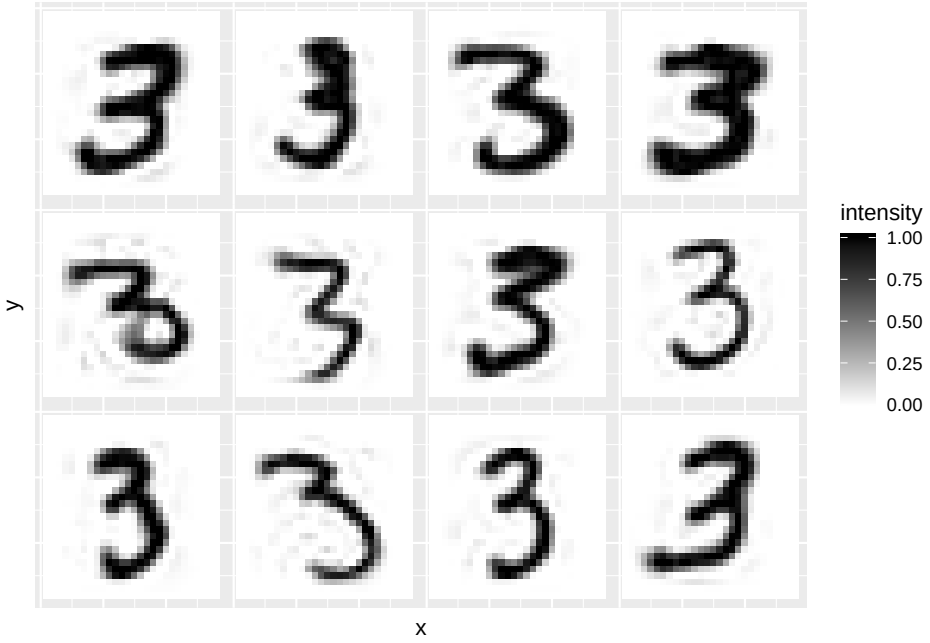
```
r=10
recon =  mnist3.pca$x[,1:r] %*% t(mnist3.pca$rotation[,1:r])
plot.mnist2(matrix(rep(xbar,12), byrow=T, nr=12)+recon[1:12,])
```



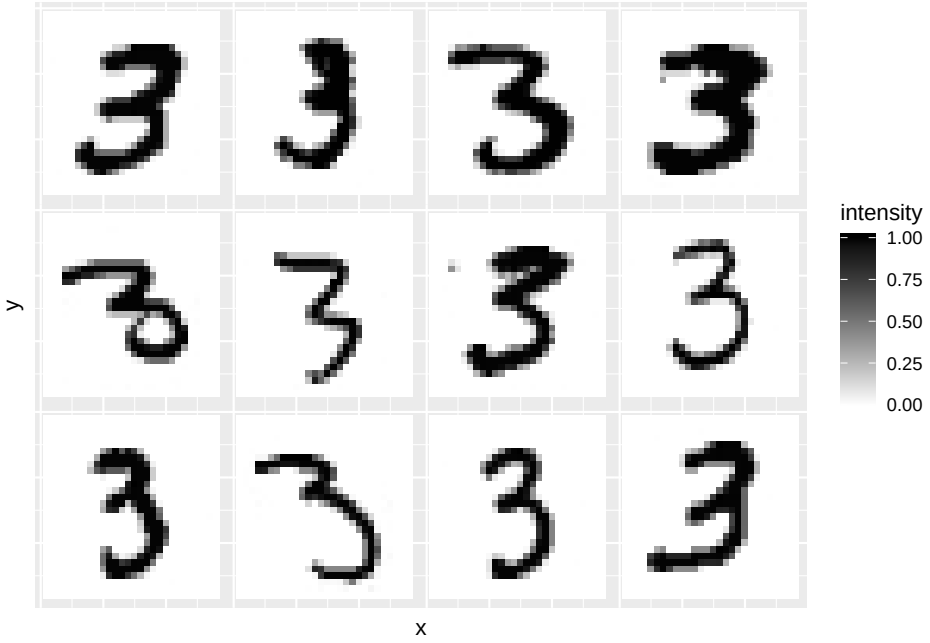
```
r=50
recon =  mnist3.pca$x[,1:r] %*% t(mnist3.pca$rotation[,1:r])
plot.mnist2(matrix(rep(xbar,12), byrow=T, nr=12)+recon[1:12,])
```




```
r=100
recon =  mnist3.pca$x[,1:r] %*% t(mnist3.pca$rotation[,1:r])
plot.mnist2(matrix(rep(xbar,12), byrow=T, nr=12)+recon[1:12,])
```




```
r=500
recon =  mnist3.pca$x[,1:r] %*% t(mnist3.pca$rotation[,1:r])
plot.mnist2(matrix(rep(xbar,12), byrow=T, nr=12)+recon[1:12,])
```

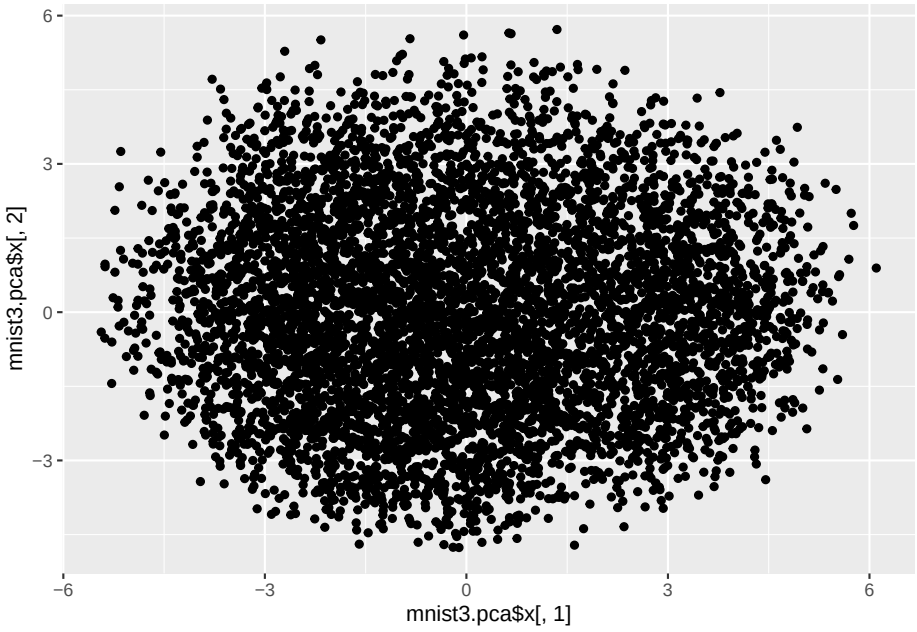


We can see that as the number of components increases the reconstructions start

to look more like the original 12 images.

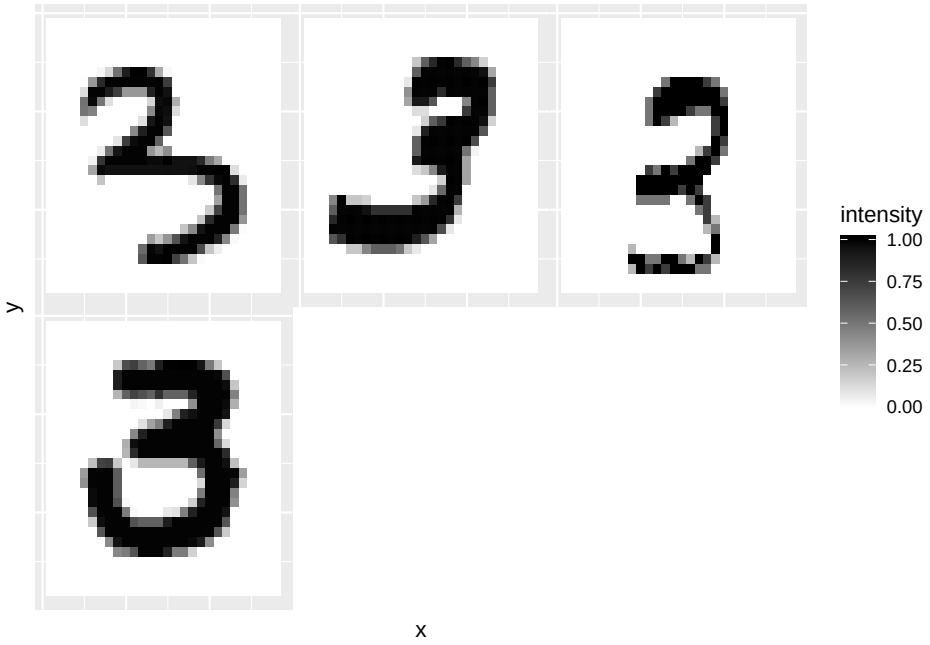
We can visualise the range of 3s by looking at a scatter plot of the first two principal components.

```
library(ggplot2)
qplot(mnist3.pca$x[,1], mnist3.pca$x[,2])
```



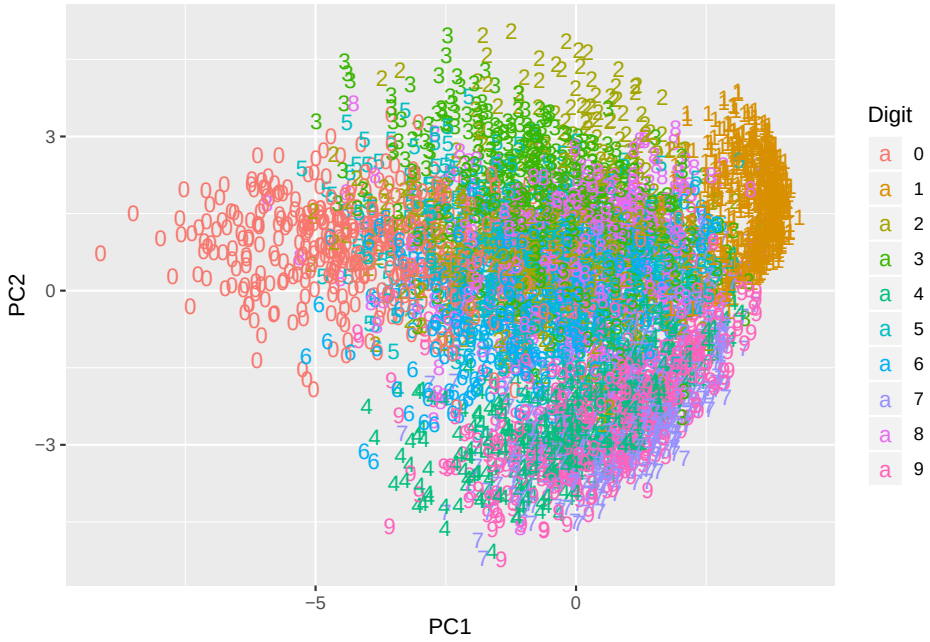
We can then finding images that differ according to these two PC scores. The first plot below is the 3 with the smallest PC1 score, and the second has the largest PC1 score. The third plot has the smallest PC2 score, and the fourth plot the largest PC2 score. These four different 3s differ in more than just the first two principal components, but you can see the effect of the PC1 score is to slant the image forward or backward, whereas PC2 changes the thickness of the line.

```
image_list <- c(which.min(mnist3.pca$x[,1]), which.max(mnist3.pca$x[,1]), which.min(mni
plot.mnist(mnist3[image_list,]) # plot the first 12 images
```



Finally, let's do PCA on a selection of the 60,000 images (not just the 3s). You can compute the SVD (which is what `prcomp` uses to do PCA) on a $60,000 \times 784$ matrix, but it takes a long time on most computers, so here I've just computed the first two components on a random selection of 5,000 images using the option `rank=2` which significantly speeds up the computation time.

```
# Note this is slow to compute!
image_index <- sample(1:60000, size=5000) # select a random sample of images
mnist.pca <- prcomp(mnist$train$x[image_index,], rank=2)
Digit = as.factor(mnist$train$y[image_index])
ggplot(as.data.frame(mnist.pca$x), aes(x=PC1, y=PC2, colour=Digit, label=Digit))+
geom_text(aes(label=Digit))
```



We can see from this scatter plot that the first two principal components do a surprisingly good job of separating and clustering the digits.

4.4 Computer tasks

1. Using the `iris` dataset, familiarize yourself with the `prcomp` command and its output.

   Now, instead of using `prcomp` we will do the analysis ourselves using the `eigen` command.

- Start by computing the sample mean and sample variance of the dataset (use $n-1$ as the denominator when you compute the sample variance to get the same answer as provided by `prcomp`).
- Now compute the eigenvalues and eigenvectors of the covariance matrix using `eigen`. Check that these agree with those computed by `prcomp` (noting that `prcomp` returns the standard deviation which is the square root of the eigenvalues).
- Now compute the principal component scores by multiplying X by the matrix of eigenvectors V. Check your answer agrees with the scores provided by `prcomp`.

Now we will do the same thing again, but using the `svd` command.

- Compute the column centred data matrix $\frac{1}{\sqrt{n-1}}HX$

- Compute the SVD of this matrix. Check the singular values match the square root of the eigenvalues computed previously.
- Compute the SVD scores by doing both XV and $U\Sigma$.

2. We first look at the crabs data, which is a dataset in the MASS library. First, we obtain the data. Then we focus on 5 continuous variables, all measured in mm: FL = frontal lobe size; RW = rear width; CL = carapace length; CW = carapace width; and BD = body depth. The sample size is 200.

```
library(MASS)
?crabs            # read the help page to find out about the dataset
X=crabs[4:8]      # construct data matrix X with columns FL, RW, CL, CW, BD
```

Carry out PCA on the data in X, including obtaining a scree plot and plotting the PC scores.

```
pca <- prcomp(X, scale=FALSE)   #carry out PCA on S
pca
lambda <- pca$sdev**2    #eigenvalues of S
plot( lambda , ylim=c( 0, max(lambda) )
lines(lambda)
```

Some questions:

- Do you have any suggestions for an interpretation for the 1st PC?
- Are you able to come up with an interpretation for the 2nd PC?
- Do you think an analysis based on the sample covariance matrix $\mathbf{S}$ or the correlation matrix $\mathbf{R}$ is preferable with this dataset? Note that you can use `{scale=TRUE} in {prcomp} to carry out PCA on` $\mathbf{R}$`. Does it make much difference which is used?`

4.5 Exercises

1. Consider the following data in $\mathbb{R}^2$

$$x_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \; x_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \; x_3 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \; x_3 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$$

- What is the orthogonal projection of these points onto

$$u_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and onto

$$u_2 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \end{pmatrix} ?$$

- Compute the sample variance matrix of the three data points, and compute its spectral decomposition.
- What vector u would maximize the variance of these projection?
- What vector u would minimize

$$\sum_{i=1}^{4} ||x_i - uu^\top x_i||_2^2?$$

This is the sum of squared errors from a rank 1 approximation to the data.

- Plot the data points and convince yourself that your answers make intuitive sense.

2. Consider a population covariance matrix Σ of the form

$$\Sigma = \gamma \mathbf{I}_p + aa^\top$$

where $\gamma > 0$ is a scalar, $\mathbf{I}_p$ is the $p \times p$ identity matrix and a is a vector of dimension p.
   - Show that a is an eigenvector of Σ.
   - Show that if b is any vector such that $a^\top b = 0$, then b is also an eigenvector of Σ.
   - Obtain all the eigenvalues of Σ.
   - Determine expressions for the proportion of (population) variability "explained" by: - the largest (population) principal component of Σ; - the r largest (population) principal components of Σ, where $1 < r \leq p$.

3. A covariance matrix has the following eigenvalues:

```
##  [1] 4.22 2.38 1.88 1.11 0.91 0.82 0.58 0.44 0.35 0.19 0.05 0.04 0.04
```

   - Sketch a scree plot.
   - Determine the minimum number of principal components needed to explain 90% of the total variation.
   - Determine the number of principal components whose eigenvalues are above average.

4. Measurements are taken on $p = 3$ variables x_1, x_2 and x_3, with sample correlation matrix

$$R = \begin{pmatrix} 1 & 0.5792 & 0.2414 \\ 0.5792 & 1 & 0.5816 \\ 0.2414 & 0.5816 & 1 \end{pmatrix}.$$

The variable z_j is the standardised versions of x_j, $j = 1, 2, 3$, i.e. each z_j has sample mean 0 and variance 1. One observation has $z_1 = z_2 = z_3 = 0$ and a second observation has $z_1 = z_2 = z_3 = 1$. Calculate the three principal component scores for each of these observations.

Chapter 5

Canonical Correlation Analysis

Suppose we observe a random sample of n bivariate observations

$$z_1 = (x_1, y_1)^\top, \ldots, z_n = (x_n, y_n)^\top.$$

If we are interested in exploring possible dependence between the x_i's and y_i's then among the first things we would do would be to obtain a scatterplot of the x_i's against the y_i's and calculate the correlation coefficient. Recall that the sample correlation coefficient is defined by

$$r = r(x, y) = \frac{S_{xy}}{\sqrt{S_{xx}}\sqrt{S_{yy}}} \tag{5.1}$$

$$= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left(\sum_{i=1}^n (x_i - \bar{x})^2\right)^{1/2} \left(\sum_{i=1}^n (y_i - \bar{y})^2\right)^{1/2}} \tag{5.2}$$

where $\bar{x} = n^{-1} \sum_{i=1}^n x_i$ and $\bar{y} = n^{-1} \sum_{i=1}^n y_i$ are the sample means.

Recall that the sample correlation is a **scale-free measure** of the strength of **linear dependence** between the x_i's and the y_i's.

In this chapter we investigate the multivariate analogue of this question. Instead of our bivariate observations being a pair of scalars, suppose instead that we are given two different random vectors x and y. In otherwords, for each subject/case i we have observations $z_i = (x_i^\top, y_i^\top)^\top, \qquad i = 1, \ldots, n.$

Multivariate data structures can be understood better if we look at low-dimensional projections of the data. The question is, given a sample $\{x_i,\ y_i\}_{i=1,\ldots,n}$, what is a sensible way to assess and describe the strength of the linear dependence between the two vectors?

Canonical correlation analysis (CCA) gives an answer to this question in terms of the best low-dimensional linear projections of the x and y random variables. In a comparable way to PCA, 'best' is defined in terms of maximizing correlations etc. A key role is played by the singular valued decomposition (SVD) introduced in Chapter 3.

5.1 The first pair of canonical variables

Some notation

Assume we are given a random sample of vectors x_i, y_i, and that we stack these into a vector z_i

$$z_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} = (x_i^\top, y_i^\top)^\top : i = 1, \ldots, n,$$

where the x_i are $p \times 1$, the y_i are $q \times 1$ and, consequently, the z_i are $(p+q) \times 1$. We are interested in determining the strength of linear association between the x_i vectors and the y_i vectors.

We formulate this task as an optimisation problem (cf. PCA). First, we introduce some notation.

- Let $\bar{x}$, $\bar{y}$, and $\bar{z}$ denote the sample mean vectors of the z_i, x_i and y_i respectively.
- Let S_{zz} denote the sample covariance matrix of the z_i, $i = 1, \ldots, n$. Then S_{zz} can be written in block matrix form

$$S_{zz} = \begin{bmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{bmatrix},$$

  where S_{xx} $(p \times p)$ is the sample covariance matrix of the x_i, S_{yy} $(q \times q)$ is the sample covariance of the y_i, and the cross-covariance matrices are given by

$$\overset{p\times q}{S}_{xy} = n^{-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})^\top \qquad \text{and} \qquad \overset{q\times p}{S}_{yx} = S_{xy}^\top.$$

Defining the optimization objective

We want to find the linear combination of the x-variables and the linear combination of the y-variables which is most highly correlated.

One version of the optimisation problem we want to solve is: find non-zero vectors $\overset{p\times 1}{a}$ and $\overset{q\times 1}{b}$ which maximise the correlation coefficient

$$\mathbb{C}\,\mathrm{or}(a^\top x, b^\top y) = r(a^\top x, b^\top y) = \frac{a^\top S_{xy} b}{(a^\top S_{xx} a)^{1/2} (b^\top S_{yy} b)^{1/2}}.$$

In other words:

$$\begin{aligned} &\text{Maximise} \qquad r(a^\top x, b^\top y), \\ &\text{for non-zero vectors} \quad a \ \ (p \times 1) \text{ and } b \ \ (q \times 1) \end{aligned} \tag{5.3}$$

where $r(\cdot, \cdot)$ is defined in (5.2). Intuitively, this makes sense, because we want to find the linear combination of the x-variables and the linear combination of the y-variables which are most highly correlated.

However, note that for any $\gamma > 0$ and $\delta > 0$,

$$r(\gamma a^\top x, \delta b^\top y) = \frac{\gamma\delta}{\sqrt{\gamma^2\delta^2}} r(a^\top x, b^\top y) \tag{5.4}$$

$$= r(a^\top x, b^\top y), \tag{5.5}$$

i.e. $r(a^\top x, b^\top y)$ is invariant to (i.e. unchanged by) multiplication of a and b by positive scalars. Consequently there will be an infinite number of solutions to this optimisation problem, because if a and b are solutions, then so are γa and δb, for any $\gamma > 0$ and $\delta > 0$.

A more useful way to formulate this optimisation problem is the following:

$$\begin{aligned} &\text{Maximize} \quad a^\top S_{xy} b \\ &\text{subject to} \qquad a^\top S_{xx} a = b^\top S_{yy} b = 1. \end{aligned} \tag{5.6}$$

Thankfully, there is a link between the solutions of the two optimization problems (5.3) and (5.6). Firstly, the invariance of $r(a^\top x, b^\top y)$ when we change a to γa and change b to δb, where $\gamma > 0$ and $\delta > 0$ are scalars, means that if $\breve{a}$ and $\breve{b}$ are a solution to problem (5.6), then for any $\gamma > 0$ and $\delta > 0$, $a = \gamma\breve{a}$ and $b = \delta\breve{b}$ are a solution to (5.3).

Conversely, we can convert any solution to optimization problem (5.3) to be a solution to the problem (5.6):

Proposition 5.1. *If a and b maximise* (5.3)*, then*

$$\breve{a} = \frac{a}{(a^\top S_{xx} a)^{1/2}} \qquad \textit{and} \qquad \breve{b} = \frac{b}{(b^\top S_{yy} b)^{1/2}}$$

are a solution to the constrained maximization problem (5.6)*.*

Proof. Suppose a and b are solutions to optimization problem (5.3). Then invariance with respect to rescaling implies that $\breve{a} = a/(a^\top S_{xx} a)^{1/2}$ and $\breve{b} =$

$b/(b^\top S_{yy} b)^{1/2}$ also achieve the optima. But $\breve{a}$ and $\breve{b}$ satisfy the constraints $a^\top S_{xx} a = b^\top S_{yy} b = 1$ because

$$\breve{a}^\top S_{xx} \breve{a} = \frac{a^\top S_{xx} a}{\left\{(a^\top S_{xx} a)^{1/2}\right\}^2} = \frac{a^\top S_{xx} a}{a^\top S_{xx} a} = 1$$

and similarly for $\breve{b}$. So $\breve{a}$ and $\breve{b}$ maximises (5.6) subject to the constraints. □

The first canonical components

As in the chapter on PCA, the optimal solution for CCA can be computed using the singular value decomposition. Before we describe the result, let's prove the following proposition from Chapter 3

Proposition 5.2. *For any matrix Q, we have*

$$\max_{a,b:\, ||a||=||b||=1} a^\top Q b = \sigma_1.$$

with the maximum obtained at $a = u_1$ and $b = v_1$, the first left and right singular vectors of Q.

Proof. We'll use the method of Lagrange multipliers to prove this result. Consider the objective

$$\mathcal{L} = a^\top Q b + \frac{\lambda_1}{2}(1 - a^\top a) + \frac{\lambda_2}{2}(1 - b^\top b).$$

The factor of $1/2$ is there to simplify the maths once we differentiate. Differentiating with respect to a and b and setting the derivative equal to zero gives

$$0 = Qb - \lambda_1 a \tag{5.7}$$

$$0 = Q^\top a - \lambda_2 b \tag{5.8}$$

where for the second equation we've noted that $a^\top Q b = b^\top Q^\top a$. Left multiplying the first equation by $a^\top$ and the second by $b^\top$, and recalling that $a^\top a = b^\top b = 1$, shows that the two Lagrange multipliers are the same $\lambda_1 = \lambda_2 =: \lambda$ say.

Substituting $a = Qb/\lambda$ into (5.7) gives

$$Q^\top Q b = \lambda^2 b,$$

and so we can see that b is an eigenvalue of $Q^\top Q$, and thus we must have $b = v_i$ for some i, i.e., b is one of the right singular vectors of Q. Similarly, substituting $b = Q^\top a/\lambda$ into (5.7) gives

$$QQ^\top a = \lambda^2 a.$$

So $a = u_j$ for some j, i.e., a is one of the left singular vectors of Q.

Finally, consider the original objective with $a = u_j$ and $b = v_i$:

$$u_j^\top Q v_i = \sigma_i u_j^\top u_i = \begin{cases} \sigma_i \text{ if } i = j \\ 0 \text{ otherwise.} \end{cases}$$

Hence we minimize the objective by taking $a = u_1$ and $b = v_1$, and then we find

$$\max_{a,b:\, ||a||=||b||=1} a^\top Q b = \sigma_1.$$

□

We're now in a position to describe the main result giving the first pair of canonical variables

Proposition 5.3. *Assume that S_{xx} and S_{yy} are both non-singular, and consider the singular value decomposition of the matrix $Q = S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2}$*

$$Q = S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sum_{j=1}^{t} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top, \tag{5.9}$$

where $t = \text{rank}(Q)$ and $\sigma_1 \geq \cdots \geq \sigma_t > 0$.

Then the solution to the constrained optimization problem (5.6) *is*

$$a = S_{xx}^{-1/2} \mathbf{u}_1 \quad \textit{and} \quad b = S_{yy}^{-1/2} \mathbf{v}_1.$$

The maximum value of the correlation coefficient is given by the largest singular value σ_1:

$$\max_{a,b} r(a^\top x, b^\top b) = \sigma_1.$$

Proof. If we let

$$\tilde{a} = S_{xx}^{1/2} a \qquad \text{and} \qquad \tilde{b} = S_{yy}^{1/2} b,$$

we may write the constraints $a^\top S_{xx} a = b^\top S_{yy} b = 1$ as

$$\tilde{a}^\top \tilde{a} = 1 \qquad \text{and} \qquad \tilde{b}^\top \tilde{b} = 1.$$

Because S_{xx} and S_{yy} are non-singular, $S_{xx}^{1/2}$ and $S_{yy}^{1/2}$ must also be non-singular, and so we can compute $S_{xx}^{-1/2}$ and $S_{yy}^{-1/2}$, using the matrix square roots defined in Section 3.2.3.

If we write

$$a = S_{xx}^{-1/2} \tilde{a} \qquad \text{and} \qquad b = S_{yy}^{-1/2} \tilde{b},$$

then the optimisation problem (5.6) becomes

$$\max_{\tilde{a},\tilde{b}} \tilde{a}^\top S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2} \tilde{b}$$

subject to

$$||\tilde{a}|| = 1 \qquad \text{and} \qquad ||\tilde{b}|| = 1.$$

Then by proposition 5.2 we can see that

$$\tilde{a} = u_1 \quad \text{and} \quad \tilde{b} = v_1$$

and the result follows.

□

We will label the solution found as

$$a_1 := S_{xx}^{-\frac{1}{2}} u_1 \qquad \text{and} \qquad b_1 := S_{yy}^{-\frac{1}{2}} v_1$$

to stress that a_1 and b_1 are the **first** pair of **canonical correlation (CC) vectors**. The variables $\eta_1 = a_1^\top x$ and $\psi_1 = b_1^\top y$ are called the first pair of **canonical correlation variables**, and σ_1 is the **first canonical correlation**.

5.1.1 Example: Premier league football

Lets again return to the Premier League from the previous chapter.

```
library(dplyr)
prem1920 <- read.csv('data/2019_2020EPL.csv')
# the data can be downloaded from https://www.rotowire.com/soccer/league-table.php
table <- prem1920 %>% dplyr::select(Team, W, D, L, G, GA, GD)
knitr::kable(head(table,5), booktabs = TRUE, escape=FALSE)
```

Team	W	D	L	G	GA	GD
Liverpool	32	3	3	85	33	52
Manchester City	26	3	9	102	35	67
Manchester United	18	12	8	66	36	30
Chelsea	20	6	12	69	54	15
Leicester City	18	8	12	67	41	26

We shall treat W and D, the number of wins and draws, respectively, as the x-variables. The number of goals for and against, G and GA, will be treated as the y-variables. The number of losses and the goal difference, L and GD, are omitted as they provide no additional information when we know W and D (and we've assumed S_{xx} is invertible).

We shall consider the questions

- how strongly associated are the match outcome variables, W and D, with the goals for and against variables, G and GA?
- what linear combination of W and D, and of G and GA are most strongly correlated?

Firstly, we need to compute the three covariance matrices needed for CCA. These are easily computed in R:

```
X <- table[2:3] # W and D
Y <- table[5:6] # G and GA
S_xx <- cov(X)
S_yy <- cov(Y)
S_xy <- cov(X,Y)
```

giving

$$S_{xx} = \begin{pmatrix} 40.4 & -9.66 \\ -9.66 & 10.7 \end{pmatrix}, \quad S_{yy} = \begin{pmatrix} 354 & -155 \\ -155 & 141 \end{pmatrix}, \quad S_{xy} = S_{yx}^{\top} = \begin{pmatrix} 108 & -60 \\ -28.9 & -2.36 \end{pmatrix}$$

We now want to calculate the matrix Q in (5.9) and then find its singular valued decomposition. We first need to find $S_{xx}^{-1/2}$ and $S_{yy}^{-1/2}$. Using R to do the computations we obtain the spectral decompositions

```
eigen_xx <- eigen(S_xx)
S_xx_invsqrt <- eigen_xx$vectors%*% diag(1/sqrt(eigen_xx$values)) %*% t(eigen_xx$vectors)
# check S_xx %*% S_xx_invsqrt %*% S_xx_invsqrt is the identity matrix
```

$$S_{xx} = Q\Lambda Q^{\top} = \begin{pmatrix} -0.959 & -0.285 \\ 0.285 & -0.959 \end{pmatrix} \begin{pmatrix} 43.2 & 0 \\ 0 & 7.82 \end{pmatrix} \begin{pmatrix} -0.959 & 0.285 \\ -0.285 & -0.959 \end{pmatrix}$$

and so

$$\begin{aligned} S_{xx}^{-1/2} &= Q\Lambda^{-\frac{1}{2}}Q^{\top} \\ &= \begin{pmatrix} -0.959 & -0.285 \\ 0.285 & -0.959 \end{pmatrix} \begin{pmatrix} 0.152 & 0 \\ 0 & 0.357 \end{pmatrix} \begin{pmatrix} -0.959 & 0.285 \\ -0.285 & -0.959 \end{pmatrix} \\ &= \begin{pmatrix} 0.169 & 0.0561 \\ 0.0561 & 0.341 \end{pmatrix}. \end{aligned}$$

Similarly, we find

```
eigen_yy <- eigen(S_yy)
S_yy_invsqrt <- eigen_yy$vectors%*% diag(1/sqrt(eigen_yy$values)) %*% t(eigen_yy$vectors)
```

$$S_{yy}^{-1/2} = \begin{pmatrix} 0.0657 & 0.0337 \\ 0.0337 & 0.112 \end{pmatrix}.$$

Consequently,

$$\begin{aligned} Q &= S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2} \\ &= \begin{pmatrix} 0.169 & 0.0561 \\ 0.0561 & 0.341 \end{pmatrix} \begin{pmatrix} 108 & -60 \\ -28.9 & -2.36 \end{pmatrix} \begin{pmatrix} 0.169 & 0.0561 \\ 0.0561 & 0.341 \end{pmatrix} \\ &= \begin{pmatrix} 0.747 & -0.588 \\ -0.39 & -0.595 \end{pmatrix} \end{aligned}$$

The SVD of Q is given by

$$\begin{aligned} Q &= U \mathbf{\Sigma} V^\top \\ &= \begin{pmatrix} -0.99 & -0.143 \\ -0.143 & 0.99 \end{pmatrix} \begin{pmatrix} 0.955 & 0 \\ 0 & 0.705 \end{pmatrix} \begin{pmatrix} -0.715 & -0.699 \\ 0.699 & -0.715 \end{pmatrix}^\top \end{aligned} \tag{5.10}$$

```
A = S_xx_invsqrt %*% S_xy %*% S_yy_invsqrt
A_svd = svd(A)
```

So the 1st CC coefficient is 0.955, which is close to its maximum value of 1. The 1st CC weight vectors are given by

```
a1 = S_xx_invsqrt%*% A_svd$u[,1]
b1 = S_yy_invsqrt%*% A_svd$v[,1]
```

$$\begin{aligned} a_1 &= S_{xx}^{-1/2} u_1 \\ &= \begin{pmatrix} 0.169 & 0.0561 \\ 0.0561 & 0.341 \end{pmatrix} \begin{pmatrix} -0.99 \\ -0.143 \end{pmatrix} \\ &= \begin{pmatrix} -0.175 \\ -0.104 \end{pmatrix} \\ b_1 &= S_{yy}^{-1/2} v_1 = \begin{pmatrix} -0.0234 \\ 0.0541 \end{pmatrix} \end{aligned}$$

In order to make interpretation easier:

- We change a_1 to $-a_1$ and b_1 to $-b_1$. [This entails changing u_1 to $-u_1$ and v_1 to $-v_1$; note that, provided we change the sign of **both** u_1 and v_1, we do not change the matrix Q.]
- We rescale a_1 and b_1 so that they are unit vectors.

```
a1n = -a1/sqrt(sum(a1^2))
b1n = -b1/sqrt(sum(b1^2))
```

This leads to the standardised 1st CC weight vectors

$$a_1 = \begin{pmatrix} 0.859 \\ 0.512 \end{pmatrix} \qquad \text{and} \qquad b_1 = \begin{pmatrix} 0.397 \\ -0.918 \end{pmatrix}$$

and the 1st CC variables, obtained by using these weights, are

$$\eta_1 = 0.859(W - \bar{W}) + 0.512(D - \bar{D})$$

and

$$\psi_1 = 0.397(G - \bar{G}) - 0.918(GA - \bar{GA}),$$

where the bars are used to denote sample means.

THIS IS THE FIRST MENTION OF REMOVING MEANS?

We can see that ψ_1 is measuring something similar to goal difference $G-GA$, as usually defined, but it gives higher weight to goals scored than goals conceded (0.397 versus $'r - 1 * signif(b1n[1], 3)`$).

It is also seen that η_1 is measuring something similar to number of points $3W+D$, as usually defined, but the ratio of points for a win to points for a draw is lower, at around 2:1, as opposed to the usual ratio 3:1.

The full list of the first canonical correlation variables (using the original canonical vectors) is thus

```
Xcent <- sweep(X,2, colMeans(X)) # column centre the matrix
Ycent <- sweep(Y,2, colMeans(Y)) # column centre the matrix
(eta = as.matrix(Xcent)%*%a1)
```

```
##               [,1]
##  [1,] -2.43407235
##  [2,] -1.38385898
##  [3,] -0.92211994
##  [4,] -0.64649410
##  [5,] -0.50498863
##  [6,] -0.46776599
##  [7,] -0.60557891
##  [8,] -0.43054335
##  [9,] -0.22197769
## [10,] -0.08416477
## [11,]  0.12440088
## [12,]  0.16162352
## [13,]  0.40741182
## [14,]  0.51169465
## [15,]  0.44463446
```

```
## [16,]  0.79101303
## [17,]  1.07033142
## [18,]  1.17461425
## [19,]  1.03680133
## [20,]  1.97903932
```

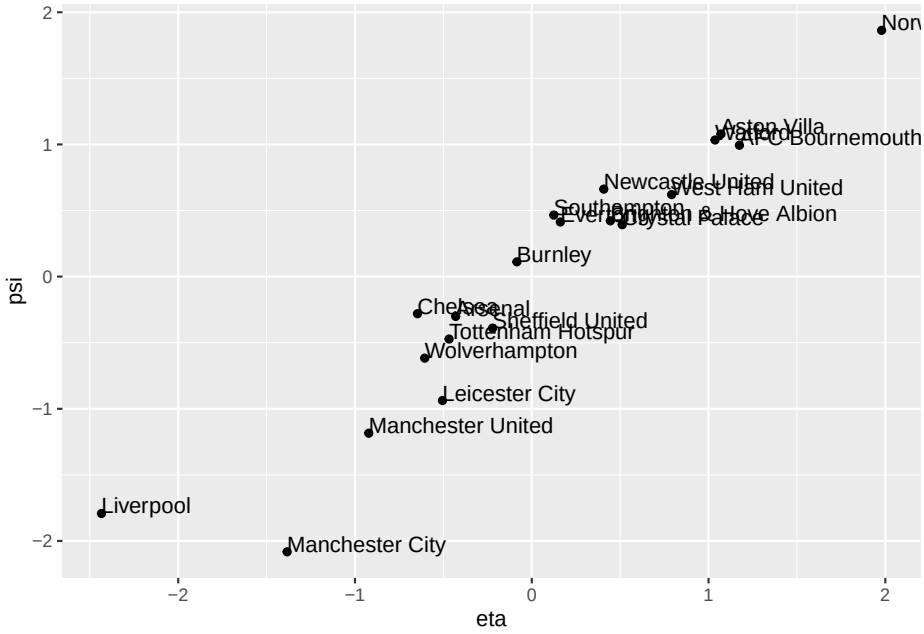
```
(psi = as.matrix(Ycent)%*%b1)
```

```
##              [,1]
##  [1,] -1.7921117
##  [2,] -2.0820965
##  [3,] -1.1846733
##  [4,] -0.2807761
##  [5,] -0.9374949
##  [6,] -0.4722204
##  [7,] -0.6168145
##  [8,] -0.3009743
##  [9,] -0.3898343
## [10,]  0.1117929
## [11,]  0.4655992
## [12,]  0.4130920
## [13,]  0.6618838
## [14,]  0.3928938
## [15,]  0.4219760
## [16,]  0.6206908
## [17,]  1.0786948
## [18,]  0.9938785
## [19,]  1.0334581
## [20,]  1.8630363
```

A scatter plotof the two canonical correlation variables shows the strong correlation between them.

ADD TEAM NAMES?

```
cca.out <- data.frame(eta=eta, psi=psi, Team=table$Team)
library(ggplot2)
ggplot(cca.out, aes(x= eta, y= psi, label=Team))+   geom_point() +
geom_text(aes(label=Team),hjust=0, vjust=0, size=4)
```



WHAT HAPPENS IF WE INCLUDE L? THen S_XX is not invertible.

PLOTS + ORTHOGONALIZATION?

Projected variable plots

Do with R package

MOVE TO AFTER DONE THE FULL SET.

We can also do this with the `CCA` package in R. See if you can find the outputs computed above in the output of the `cc` command.

```
library(CCA)
prem.cca <- cc(X,Y)
prem.cca$cor # the canonical correlations
```

```
## [1] 0.9550749 0.7054825
```

```
prem.cca$xcoef # the canonical correlation vectors for X
```

```
##         [,1]      [,2]
## W -0.1750356 0.0313256
## D -0.1042828 0.3293057
```

```
prem.cca$ycoef  # the canonical correlation vectors for Y
```

```
##            [,1]        [,2]
## G  -0.02342507 -0.07003113
## GA  0.05412069 -0.10372100
```

```
prem.cca$scores$xscores # the canonical correlation variables
```

```
##              [,1]        [,2]
##  [1,] -2.43407235 -1.49036509
##  [2,] -1.38385898 -1.67831867
##  [3,] -0.92211994  1.03482821
##  [4,] -0.64649410 -0.87835503
##  [5,] -0.50498863 -0.28239474
##  [6,] -0.46776599  0.64287128
##  [7,] -0.60557891  1.59946290
##  [8,] -0.43054335  1.56813730
##  [9,] -0.22197769  0.90952583
## [10,] -0.08416477 -0.04706579
## [11,]  0.12440088 -0.70567727
## [12,]  0.16162352  0.21958876
## [13,]  0.40741182  0.48624330
## [14,]  0.51169465  0.15693757
## [15,]  0.44463446  1.41150932
## [16,]  0.79101303 -0.20369377
## [17,]  1.07033142 -0.56432510
## [18,]  1.17461425 -0.89363084
## [19,]  1.03680133  0.06296078
## [20,]  1.97903932 -1.34823896
```

```
prem.cca$scores$yscores # the canonical correlation variables
```

```
##             [,1]        [,2]
##  [1,] -1.7921117 -0.39245373
##  [2,] -2.0820965 -1.79042487
##  [3,] -1.1846733  0.62697465
##  [4,] -0.2807761 -1.45009678
##  [5,] -0.9374949  0.03833851
##  [6,] -0.4722204 -0.16380075
##  [7,] -0.6168145  1.26255753
##  [8,] -0.3009743  0.08263387
##  [9,] -0.3898343  2.20665204
## [10,]  0.1117929  0.78559650
## [11,]  0.4655992 -0.81186254
## [12,]  0.4130920  0.09323935
## [13,]  0.6618838  0.30598410
## [14,]  0.3928938  1.62597001
## [15,]  0.4219760  0.65083699
## [16,]  0.6206908 -0.87924229
## [17,]  1.0786948 -0.83759830
## [18,]  0.9938785 -0.56012517
## [19,]  1.0334581 -0.17627967
```

```
## [20,]  1.8630363 -0.61689944
```

```
matcor(X,Y)
```

5.2 The full set of canonical correlations

Let us first recap what we did in the previous section: we found a linear combination of the x-variables and a linear combination of the y-variables which maximised the correlation, and expressed the answer in terms of quantities which arise in the SVD of Q, where

$$Q \equiv S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2} = U \mathbf{\Sigma} V^\top = \sum_{j=1}^{t} \sigma_j u_j v_j^\top.$$

We found the maximum value of the correlation $\mathbb{C}\,\mathrm{or}(a^\top x, b^\top y)$ to be σ_1, achieved using the linear combinations $\eta_1 = a_1^\top x$ and $\psi_1 = b_1^\top y$ with

$$a_1 = S_{xx}^{-1/2} u_1 \quad \text{and } b_1 = S_{yy}^{-1/2} v_1.$$

We now repeat this process to find the next most important linear combination, subject to being uncorrelated with the first linear combination, as we did with PCA. For $a^\top x$ to be uncorrelated with $\eta_1 = a_1^\top x$ we require

$$0 = \mathbb{C}\,\mathrm{ov}(a_1^\top x, a^\top x) = a_1^\top S_{xx} a,$$

and similarly we require the condition $b_1^\top S_{yy} b = 0$ for b.

Thus, to obtain the second canonical correlation coefficient, plus the associated sets of canonical correlation vectors and variables, we need to solve the following optimisation problem:

$$\max_{a,\,b} a^\top S_{xy} b \tag{5.11}$$

subject to the constraints

$$a^\top S_{xx} a = b^\top S_{yy} b = 1, \tag{5.12}$$

$$a_1^\top S_{xx} a = b_1^\top S_{yy} b = 0. \tag{5.13}$$

Note that maximising (5.11) subject to (5.12) is very similar to the optimisation problem (5.6) considered in the previous section. What is new are the constraints (5.13), which take into account that we have already found the first canonical correlation.

It will probably not surprise you to find that the solution is

$$a^\top S_{xy} b = \sigma_2 \quad \text{achieved at} \quad a = a_2 := S_{xx}^{-1/2} u_2 \text{ and } b = b_2 := S_{yy}^{-1/2} v_2$$

where σ_2 is the second largest singular value of Q, and u_2 and v_2 are the corresponding left and right singular vectors.

We now state the result in its full generality.

Proposition 5.4. *For $k = 1, \ldots, r = \operatorname{rank}(S_{xy})$, the solution to sequence of optimization problems*

$$\begin{aligned} \textit{Maximize} \quad & a^\top S_{xy} b && (5.14)\\ \textit{subject to} \quad & a^\top S_{xx} a = b^\top S_{yy} b = 1 && (5.15)\\ \textit{and} \quad & a_i^\top S_{xx} a = b_i^\top S_{yy} b = 0 \textit{ for } i = 1, \ldots, k && (5.16) \end{aligned}$$

is achieved at $a_k = S_{xx}^{-1/2} u_k$ and $b_k = S_{yy}^{-1/2} v_k$ with $a_k S_{xy} b_k = \sigma_k$.

Note that an equivalent way of writing down the problem is as

$$\begin{aligned} \text{Maximize} \quad & \operatorname{tr}(A^\top S_{xy} B) = \sum_{i=1}^{k} a_i^\top S_{xy} b_i \\ \text{subject to} \quad & A^\top S_{xx} A = \mathbf{I} \\ \text{and} \quad & B^\top S_{yy} B = \mathbf{I} \end{aligned}$$

which is in the form of the general problem given in Equation (4.1) if

$$A = \begin{pmatrix} a_1 & \ldots & a_k \end{pmatrix}, \quad B = \begin{pmatrix} b_1 & \ldots & b_k \end{pmatrix}$$

are matrices containing the canonical correlation vectors as columns.

Before we prove this result, we first give an extension of Proposition 5.2.

Proposition 5.5. *Let Q be an arbitrary matrix with SVD $Q = \sum_{k=1}^{r} \sigma_i u_i v_i^\top$. For $k = 1, \ldots, r$ the solution to the optimization problem*

$$\begin{aligned} \textit{Maximize} \quad & a^\top Q b && (5.17)\\ \textit{subject to} \quad & a^\top a = b^\top b = 1 && (5.18)\\ & a_i^\top a = b_i^\top b = 0 \textit{ for } i = 1, \ldots, k-1 && (5.19) \end{aligned}$$

is achieved at

$$a_k = u_k, \qquad b_k = v_k$$

with

$$a_k^\top Q b_k = \sigma_k.$$

We will work through the proof of this proposition in the Exercises.

Proof. To prove Proposition 5.4 given Proposition 5.5, we note as before that if we write $\tilde{a}_j = S_{xx}^{1/2} a_j$ and $\tilde{b}_j = S_{yy}^{1/2} b_j$, then the constraints become

$$\tilde{a}^\top \tilde{a} = \tilde{b}^\top \tilde{b} = 1 \text{ and } \tilde{a}_i^\top \tilde{a} = \tilde{b}_i^\top \tilde{b} = 0 \text{ for } i = 1, \ldots, k.$$

Consequently, we may view constraints (5.13) as corresponding to orthogonality constraints (cf. PCA) in modified coordinate systems.

The objective $a^\top S_{xy} b$ becomes $\tilde{a}^\top Q \tilde{b}$ with

$$Q = S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2}.$$

Thus applying Proposition 5.5 gives the desired result. □

To summarize:

- The k^{th} **canonical correlation** is σ_k, the k^{th} largest singular value of Q.
- The k^{th} **canonical correlation vectors** (sometimes called the **weights** for the x and y variables) are

$$a_k = S_{xx}^{-1/2} u_k, \qquad b_k = S_{yy}^{-1/2} v_k$$

- The k^{th} **canonical correlation variables** (or **canonical correlation scores**) are

$$\eta_{ik} = a_k^\top (x_i - \bar{x}), \qquad \psi_{ik} = b_k^\top (y_i - \bar{y}).$$

We define the CC variable/score vectors to be

$$\boldsymbol{\eta}_k = (\eta_{1k}, \ldots, \eta_{nk})^\top \text{ and } \boldsymbol{\psi}_k = (\psi_{1k}, \ldots, \psi_{nk})^\top.$$

Example ?? (continued) From (5.10), it is seen that the 2nd CC coefficient is given by $\sigma_2 = 0.508$. So the correlation between the second pair of CC variables is a lot smaller than the 1st CC coefficient, though still appreciably different from 0. We now calculate the 2nd CC weight vectors:

$$a_2 = S_{xx}^{-1/2} q_2 = \begin{pmatrix} 0.073 \\ 0.396 \end{pmatrix} \qquad \text{and} \qquad b_2 = S_{yy}^{-1/2} r_2 = -\begin{pmatrix} 0.062 \\ 0.086 \end{pmatrix},$$

with standardised version (without the sign changes this time)

$$a_2 = \begin{pmatrix} 0.181 \\ 0.984 \end{pmatrix} \qquad \text{and} \qquad b_2 = -\begin{pmatrix} 0.589 \\ 0.808 \end{pmatrix},$$

and new variables

$$\eta_2 = 0.181 * (W - \bar{W}) + 0.984 * (D - \bar{D})$$

and

$$\psi_2 = -\{0.589 * (F - \bar{F}) + 0.808 * (A - \bar{A})\}.$$

Note that, to a good approximation, η_2 is measuring something similar to the number of draws and, approximately, ψ_2 is something related to the negative of total number of goals in a team's games. So large ψ_2 means relatively few goals

in a team's games, and small (i.e. large negative) ψ_2 means a relatively large number of goals in a team's games.

Interpretation of the 2nd CC: teams that have a lot of draws tend to be in low-scoring games and/or teams that have few draws tend to be in high-scoring games.

5.3 Properties

Proposition 5.6. *Assume that S_{xx} and S_{yy} both have full rank. Then for $1 \leq k, \ell \leq t$,*

$$\mathbb{C}\,\mathrm{or}(\eta_k, \psi_\ell) = \begin{cases} \sigma_k & \text{if} \quad k = \ell \\ 0 & \text{if} \quad k \neq \ell, \end{cases}$$

and

$$\mathbb{C}\,\mathrm{or}(\eta_j, \eta_k) = \mathbb{C}\,\mathrm{or}(\psi_j, \psi_k) = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases}$$

5.3.1 Connection with linear regression when $q = 1$

Although CCA analysis is clearly a different technique to linear regression, it turns out that when either $\dim x = p = 1$ or $\dim y = q = 1$, there is a close connection between the two approaches.

Consider the case $q = 1$ and $p > 1$. Hence there is only a single y-variable but we still have $p > 1$ x-variables.

Let's make the following assumptions:

1. The x_i have been centred so that $\bar{x} = \mathbf{0}_p$, the zero vector.
2. The covariance matrix for the x-variables, S_{xx}, has full rank p.

The first assumption means that

$$S_{xx} = \frac{1}{n} X^\top X \quad \text{and} \quad S_{xy} = X^\top y,$$

and the second means that $(X^\top X)^{-1}$ exists.

Since $q = 1$, the matrix we decompose in CCA

$$Q = S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2}$$

is a $p \times 1$ vector. Consequently, its SVD is just

$$Q = \sigma_1 u_1,$$

where

$$\sigma_1 = ||Q||_F = (Q^\top Q)^{\frac{1}{2}} \qquad \text{and} \qquad u_1 = Q/||Q||_F.$$

Consequently, the first canoncial correlation vector for x is

$$\begin{aligned} a &= S_{xx}^{-1/2} u_1 = S_{xx}^{-1/2} \frac{Q}{||Q||_F} \\ &= S_{xx}^{-1/2} \frac{1}{||S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2}||_F} S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2} \\ &= \frac{1}{||S_{xx}^{-1/2} S_{xy}||_F} S_{xx}^{-1} S_{xy} \\ &= c(X^\top X)^{-1} X^\top y \end{aligned}$$

where $c = ||S_{xx}^{-1/2} S_{xy}||_F$ is a scalar.

Thus, we can see that the first canonical correlation vector a is a scalar multiple of

$$\hat{\boldsymbol{\beta}} = \left(X^\top X\right)^{-1} X^\top y,$$

the classical least squares estimator. Therefore the least squares estimator $\hat{\boldsymbol{\beta}}$ solves (5.3). However, it does not usually solve the constrained optimisation problem (5.6) because typically $\hat{\boldsymbol{\beta}}^\top S_{xx} \hat{\boldsymbol{\beta}} \neq 1$, so that the constraint in Equation (5.6) will not be satisfied.

5.3.2 Invariance/equivariance properties of CCA

Suppose we apply orthogonal transformations and translations to the x_i and the y_i of the form

$$\mathbf{h}_i = \mathbf{T} x_i + \boldsymbol{\mu} \qquad \text{and} \qquad \mathbf{k}_i = \mathbf{R} y_i + \boldsymbol{\eta}, \qquad i = 1, \ldots, n, \tag{5.20}$$

where $\mathbf{T}$ $(p \times p)$ and $\mathbf{R}$ $(q \times q)$ are orthogonal matrices, and $\boldsymbol{\mu}$ $(p \times 1)$ and $\boldsymbol{\eta}$ $(q \times 1)$ are fixed vectors.

How do these transformations affect the CC analysis?

First of all, since CCA depends only on sample covariance matrices, it follows that the translation vectors $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ have no effect on the analysis.

As seen in the previous section, the CCA in the original coordinates depends on

$$Q \equiv Q_{xy} = S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2}. \tag{5.21}$$

In the new coordinates we have

$$\tilde{S}_{hh} = \mathbf{T} S_{xx} \mathbf{T}^\top, \qquad \tilde{S}_{kk} = \mathbf{R} S_{yy} \mathbf{R}^\top,$$

$$\tilde{S}_{hk} = \mathbf{T} S_{xy} \mathbf{R}^\top = \tilde{S}_{kh}^\top,$$

where here and below, a tilde above a symbol is used to indicate that the corresponding term is defined in terms of the new h, k coordinates, rather than the old x, y coordinates. Due to the fact that $\mathbf{T}$ and $\mathbf{R}$ are orthogonal,

$$\tilde{S}_{hh}^{1/2} = \mathbf{T} S_{xx}^{1/2} \mathbf{T}^\top, \qquad \tilde{S}_{hh}^{-1/2} = \mathbf{T} S_{xx}^{-1/2} \mathbf{T}^\top$$

$$\tilde{S}_{kk}^{1/2} = \mathbf{R} S_{yy}^{1/2} \mathbf{R}^\top \qquad \text{and} \qquad \tilde{S}_{kk}^{-1/2} = \mathbf{R} S_{yy}^{-1/2} \mathbf{R}^\top.$$

The analogue of (5.21) in the new coordinates is given by

$$\begin{aligned} \tilde{Q}_{\mathbf{h}k} &= \tilde{S}_{hh}^{-1/2} \tilde{S}_{\mathbf{h}k} \tilde{S}_{kk}^{-1/2} \\ &= \mathbf{T} S_{xx}^{-1/2} \mathbf{T}^\top \mathbf{T} S_{xy} \mathbf{R}^\top \mathbf{R} S_{yy}^{-1/2} \mathbf{R}^\top \\ &= \mathbf{T} S_{xx}^{-1/2} S_{xy} S_{yy}^{-1/2} \mathbf{R}^\top \\ &= \mathbf{T} Q_{xy} \mathbf{R}^\top. \end{aligned}$$

So, again using the fact that $\mathbf{T}$ and $\mathbf{R}$ are orthogonal matrices, if Q_{xy} has SVD $\sum_{j=1}^{t} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top$, then $\tilde{Q}_{hk}$ has SVD

$$\begin{aligned} \tilde{Q}_{hk} &= \mathbf{T} Q_{xy} \mathbf{R}^\top = \mathbf{T} \left(\sum_{j=1}^{t} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top \right) \mathbf{R}^\top \\ &= \sum_{j=1}^{t} \sigma_j \mathbf{T} \mathbf{u}_j \mathbf{v}_j^\top \mathbf{R}^\top = \sum_{j=1}^{t} \sigma_j \left(\mathbf{T} \mathbf{u}_j \right) \left(\mathbf{R} \mathbf{v}_j \right)^\top = \sum_{j=1}^{t} \sigma_j \tilde{u}_j \tilde{\mathbf{v}}_j^\top, \end{aligned}$$

where, for $j = 1, \ldots, t$, the $\tilde{u}_j = \mathbf{T} u_j$ are mutually orthogonal unit vectors, and the $\tilde{\mathbf{v}}_j = \mathbf{R} \mathbf{v}_j$ are also mutually orthogonal unit vectors.

Consequently, $\tilde{Q}_{\mathbf{h}k}$ has the same singular values as Q_{xy}, namely $\sigma_1, \ldots, \sigma_t$ in both cases, and so the canonical correlation coefficients are invariant with respect to the transformations (5.20). Moreover, since the optimal linear combinations for the jth CC in the original coordinates are given by $a_j = S_{xx}^{-1/2} \mathbf{u}_j$ and $b_j = S_{yy}^{-1/2} \mathbf{v}_j$, the optimal linear combinations in the new coordinates are given by

$$\begin{aligned} \tilde{a}_j &= S_{hh}^{-1/2} \mathbf{T} \mathbf{u}_j \\ &= \mathbf{T} S_{xx}^{-1/2} \mathbf{T}^\top \mathbf{T} \mathbf{u}_j \\ &= \mathbf{T} S_{xx}^{-1/2} \mathbf{u}_j \\ &= \mathbf{T} a_j, \end{aligned}$$

and a similar argument shows that $\tilde{b}_j = \mathbf{R} b_j$. So under transformations (5.20), the optimal vectors a_j and b_j transform in an equivariant manner to $\tilde{a}_j$ and $\tilde{b}_j$, respectively, $j = 1, \ldots, t$.

If either of $\mathbf{T}$ or $\mathbf{R}$ in (5.20) is not an orthogonal matrix then the singular values are not invariant and the CC vectors do not transform in an equivariant manner.

5.4 Exercises

1. Suppose that $z = (x^\top y^\top)^\top$ is a random vector, where both x and y are sub-vectors of dimension p, so that z is $(2p) \times 1$. Define

$$\mathbb{V}\mathrm{ar}(z) = \Sigma_{zz} = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}.$$

   - Suppose that $y = Tx$ where T is a fixed matrix. Find Σ_{xy} and Σ_{yy} in terms of Σ_{xx} and T.
   - Assuming now that T is an orthogonal matrix and Σ_{xx} is of full rank, determine the singular values of the matrix $Q = \Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2}$, and hence write down the canonical correlation coefficients.
   - Suppose now that T is non-singular but not orthogonal. Comment on whether the answer to part (b) changes.

2. We will now prove Proposition 5.5 by induction. The case for $k = 1$ was proved in Section 5.1 in Proposition 5.2. Assume the result is true for k. Consider the objective

$$\mathcal{L} = a^\top Q b + \sum_{i=1}^{k} \gamma_i a^\top a_i + \sum_{i=1}^{k} \mu_i b^\top b_i + \frac{\lambda_1}{2}(1 - a^\top a) + \frac{\lambda_2}{2}(1 - b^\top b)$$

   where $\lambda_i, \mu_i, \gamma_i$ are Lagrangian multipliers.
   - By differentiating with respect to a and b and setting the derivative to zero show that

$$Qb + \sum \gamma_i a_i - \lambda_1 a = 0 \tag{5.22}$$
$$Q^\top a + \sum \mu_i b_i - \lambda_2 b = 0. \tag{5.23}$$

   - By left multiplying the equations above by $a^\top$ and $b^\top$ respectively show that

$$\lambda_1 = \lambda_2 = a^\top Q b.$$

   - By left multiplying (5.22) by $a_i^\top$ show that $\gamma_i = 0$ for $i = 1, \ldots, k$. Show similarly that $\mu_i = 0$ for $i = 1, \ldots, k$.
   - Finally, by copying the proof of Proposition 5.2, prove Proposition 5.5.

3. Show the mean of the cc variables η_k and ψ_k is zero. Prove Proposition 5.6 giving the variance of covariance the cc variables.

5.5 Computer tasks

1. Use generalized inverse to find $S_{xx}^{-1/2}$
2. What if do football example but allow L to be a variable in Y so that some of X perfectly explains Y.

3. Lets consider again the crabs dataset you looked at in the exercises in the chapter on PCA (see **??**). We now consider a canonical correlation analysis in which one set of variables, the x-set, is given by CL and CW and the other set, the y-set, is given by FL, RW and BD.

```
library(MASS)
?crabs            # read the help page to find out about the dataset
X=as.matrix(crabs[4:8])
n=200                                          # sample size
H=diag(rep(1,n))-rep(1,n)%*%t(rep(1,n))/n      # n times n centering matrix
X1=X[,3:4]                                     # store CL and CW in X1
Y1=cbind(X[,1],X[,2],X[,5])                    # store FL, RW and BD in Y1
Sxx=t(X1)%*%H%*%X1/n                           # find x-variable variance matrix
Syy=t(Y1)%*%H%*%Y1/n                           # find y-variable variance matrix
Sxy=t(X1)%*%H%*%Y1/n                           # find cross-covariance matrix
```

Now calculate $\mathbf{C_x} = \mathbf{S_{xx}^{-1/2}}$ and $\mathbf{C_y} = \mathbf{S_{yy}^{-1/2}}$, and store the results in `Cx` and `Cy`, respectively.

Then calculate

```
A=Cx%*%Sxy%*%Cy              # calculate the A matrix (see Chapter 5)
U4=svd(A)$u                  # extract the U matrix (our notation) in svd
V4=svd(A)$v                  # extract the V matrix (our notation) in svd
d4=svd(A)$d                  # extract the singular values in svd
acheck=Cx%*%U4[,1]           # calculate optimal x-weights
bcheck=Cy%*%V4[,1]           # calculate optimal y-weights
xscores1=H%*%X1%*%acheck     # calculate centred x-scores for 1st CC
yscores1=H%*%Y1%*%bcheck     # calculate centred y-scores for 1st CC
plot(xscores1,yscores1)      # plot x-scores against y-scores for 1st CC
```

What does the plot tell you (if anything)? What is the 1st CC coefficient?

Repeat the above to find the two sets of 2nd CC scores, `xscores2` and `yscores2` say, and plot these against each other and against the first CC scores. [Hint: you need to extract the second column from `Q4` and `R4` rather than the first column.] Is there any interesting structure in any of the plots? Which plots suggest random scatter?

Test the following hypotheses: (i) that both CC coefficients are zero, and (ii) that at most one of the CC coefficients is non-zero. What do you conclude from your findings?

2. Explore using CCA of the crabs data with the command `cc` and `plt.cc` in the package `CCA` which you will need to download. [Note: these commands were also used on the Boston Housing data in lectures and I have provided the html output on the Moodle page, which could be helpful].

Note that you can use the commands

```
library(CCA)
matcor(X1, Y1)
```

3. The full Premier League dataset is available on the Moodle page. Read the data into R. You will need to download the data into your file space, and then read it in, e.g. using

SORT

```
x <- read.csv(x , file="/YOURDIRECTORY/prem_league_data.txt, sep=" ", header=TRUE)
```

If you are not sure what the name of YOURDIRECTORY is where the file is located, then a useful command to find out is `file.choose()`

Check that you agree with the calculations in Chapters 3 and 4 of the lecture notes. Also, look at the PC scores as indicated in point 2 and the CC scores as indicated in point 3. Note that there will be some differences due to `prcomp` using divisor $n-1$ in the calculation of the sample covariance matrix, whereas we use n in the notes. Also, there may be some minor numerical/rounding differences.

Chapter 6

Multidimensional Scaling

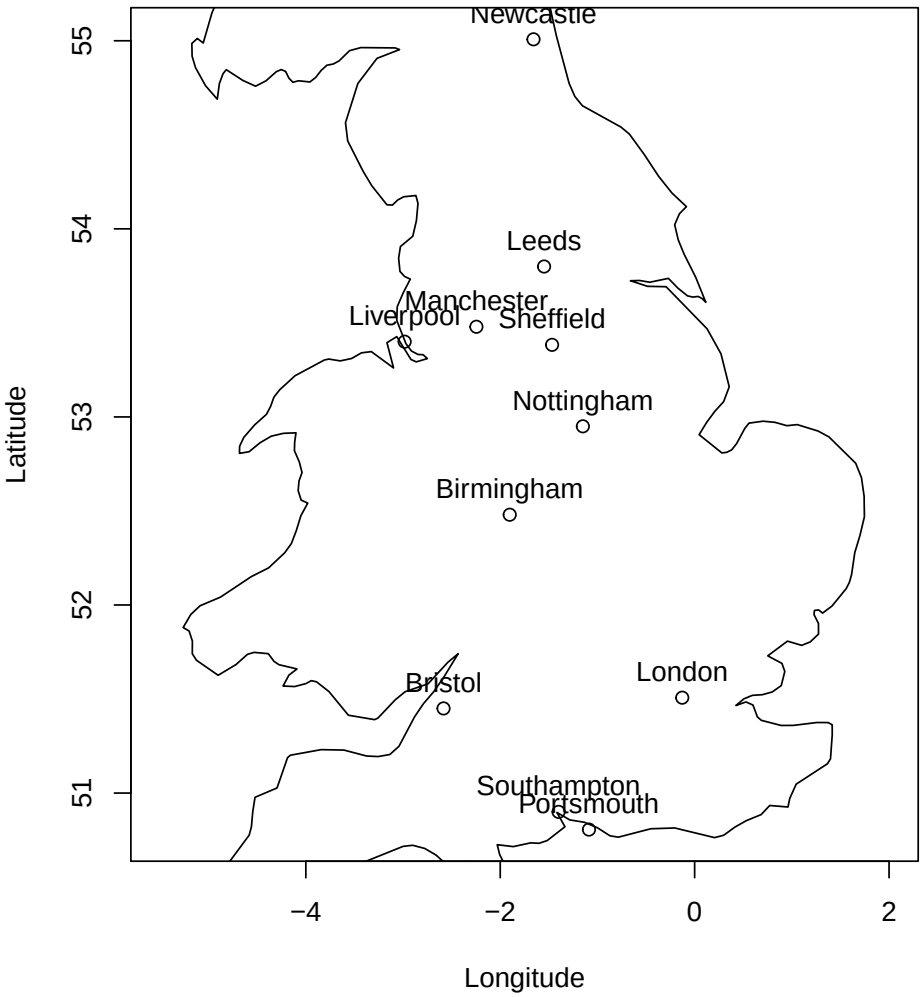
In PCA, we start with n data points $x_i \in \mathbb{R}^p$, and then try to find a low dimensional projection of these points, e.g., $y_1, \ldots, y_n \in \mathbb{R}^r$ with $r < p$, in such a way that they minimize the reconstruction error (or maximize the variance).

The focus in **Multidimensional Scaling (MDS)** is somewhat different. Instead of being given the data X, our starting point is a matrix of **distances** or **dissimilarities** between the data points, D. For example, if we have data on n different experimental units, then we would be given the distances d_{ij} between any pair of experimental units i and j. We compile these into a $n \times n$ **distance matrix** $D = (d_{ij} : i, j = 1, \ldots, n)$.

The goal of MDS is to find a set of points in a low-dimensional Euclidean space, e.g. $\mathbb{R}$ or $\mathbb{R}^2$, whose inter-point distances (or dissimilarities) are approximately equal to the d_{ij}. We may or may not have access to the original data X, and we may not even know the dimension of the original data points x_i. The aim is simply to find a set of points $y_1, \ldots, y_n$ whose distance matrix is approximately D, i.e., for which

$$\text{distance}(y_i, y_j) \approx d_{ij}.$$

As an illustrative example, consider the location of some of England's cities.



If we are told their latitude and longitude, it is easy to calculate the distances between the cities.

	London	Birmingham	Manchester	Leeds	Newcastle	Liverpool	Portsmouth	Southampton	No
London	0	163	262	273	403	286	103	112	
Birmingham	163	0	114	149	282	125	195	179	
Manchester	262	114	0	58	174	50	308	293	
Leeds	273	149	58	0	135	105	335	323	
Newcastle	403	282	174	135	0	199	469	458	
Liverpool	286	125	50	105	199	0	317	299	
Portsmouth	103	195	308	335	469	317	0	24	
Southampton	112	179	293	323	458	299	24	0	
Nottingham	175	73	94	98	231	132	239	229	
Bristol	170	124	227	271	401	219	127	103	
Sheffield	228	105	53	47	181	101	288	276	

But can we reconstruct the map from the distance matrix? This is the aim of multidimensional scaling. MDS constructs a set of points that have distances between them given by the distance matrix D. In other words, it creates a map with a set of coordinates for which the distances between points are approximately the same as in the real data.

Of course, this illustrative example is not very interesting, as the original data (the city locations) are only two-dimensional, but in problems with high dimensional data, finding a way to represent the points in a low-0dimensional space will make visualization and statistical analysis easier. As we shall see, there is a close connection between MDS and PCA.

6.1 Classical Multidimensional Scaling

We call an $n \times n$ matrix $D = (d_{ij})_{i,j=1}^n$ a **distance matrix** or, equivalently, a **dissimilarity matrix**, if the following properties are satisfied:

1. For $i = 1, \dots, n$, $d_{ii} = 0$.
2. Symmetry: $d_{ij} = d_{ji} \geq 0$ for all $i, j = 1, \dots, n$.
3. Definiteness: $d_{ij} = 0$ implies $i = j$.

A comment on our terminology. We do not require distances necessarily to satisfy the triangle inequality

$$d_{ik} \leq d_{ij} + d_{jk}. \tag{6.1}$$

A distance function which always satisfies the triangle inequality is called a **metric distance** or just a **metric**, and a distance function which does not always satisfy the triangle inequality is called **non-metric** distance.

Suppose $x_1, \dots, x_n$ are points in $\mathbb{R}^p$. If the d_{ij} are of the form

$$d_{ij} = ||x_i - x_j|| = \sqrt{(x_i - x_j)^\top (x_i - x_j)}.$$

Then each d_{ij} is called a **Euclidean distance** and, in this case, D is called a **Euclidean distance matrix**. Since Euclidean distances satisfy the triangle inequality (6.1), it follows that Euclidean distance is a metric distance.

Given a distance matrix $\mathbf{D} = \{d_{ij}\}_{i,j=1}^n$, define the matrix

$$A = \{a_{ij}\}_{i,j=1}^n, \quad \text{where} \qquad a_{ij} = -\frac{1}{2} d_{ij}^2. \tag{6.2}$$

Note that, for $i = 1, \dots, n$, $a_{ii} = -d_{ii}^2/2 = 0$.

Now define the matrix

$$\mathbf{B} = \mathbf{H} A \mathbf{H}, \tag{6.3}$$

where

$$\mathbf{H} = \mathbf{I}_n - n^{-1} \mathbf{1}_n \mathbf{1}_n^\top \tag{6.4}$$

is the $n \times n$ **centering matrix**; see §2.7. For reasons that will soon become clear, B defined by (6.3) is known as a centred inner-product matrix.

Let $x_1, \ldots, x_n$ denote n points in $\mathbb{R}^p$. Then the $n \times p$ matrix $\mathbf{X} = [x_1, \ldots, x_n]^\top$ is the data matrix, as before.

We now present the key result for classical MDS.

Proposition 6.1. *Let D denote an $n \times n$ distance matrix and suppose A, B and $\mathbf{H}$ be as defined in* (6.2), (6.3) *and* (6.4), *respectively.*

1. *The matrix D is a Euclidean distance matrix if and only if B is a non-negative definite matrix.*
2. *If D is a Euclidean distance matrix for the sample of n vectors $x_1, \ldots, x_n$, then*
$$b_{ij} = (x_i - \bar{x})^\top (x_j - \bar{x}), \qquad i, j = 1, \ldots, n, \tag{6.5}$$
where $\bar{\mathbf{x}} = n^{-1} \sum_{i=1}^n x_i$ is the sample mean vector. Equivalently, we may write
$$B = (\mathbf{HX})(\mathbf{HX})^\top,$$
where $\mathbf{X} = [x_1, \ldots, x_n]^\top$ is the data matrix, and H is the $n \times n$ centering matrix. Consequently, B is non-negative definite.
3. *Suppose B is non-negative definite with positive eigenvalues $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_k$ and spectral decomposition $B = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, where $\mathbf{\Lambda} = diag\{\lambda_1 \ldots \lambda_k\}$ and $\mathbf{Q}$ is $n \times k$ and satisfies $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_k$. Then $\mathbf{X} = [x_1, \ldots, x_n]^\top = \mathbf{Q}\mathbf{\Lambda}^{1/2}$ is an $n \times k$ data matrix for points $x_1, \ldots, x_n$ in $\mathbb{R}^k$, which have inter-point distances given by $D = (d_{ij})$. Moreover, for this data matrix $\bar{x} = \mathbf{0}_k$ and B represents the inner product matrix with elements given by* (6.5).

Proof. Part 1. is a direct consequence of parts 2. and 3. Parts 2. and 3. are proved in the example sheets. □

Important Point: Proposition 6.1 may be useful even if $\mathbf{D}$ is not a Euclidean distance matrix, in which case B has some negative eigenvalues. What we can do is to replace B by its positive part. If B has spectral decomposition $\sum_{j=1}^p \lambda_j q_j q_j^\top$, then its positive definite part is defined by

$$B_{\text{pos}} = \sum_{j:\, \lambda_j > 0} \lambda_j q_j q_j^\top.$$

In other words, we sum over those j such that λ_j is positive. Then B_{pos} is non-negative definite and so we can use Theorem 5.1(iii) to determine a Euclidean configuration which has centred inner-product matrix B_{pos}. Then, provided the negative eigenvalues are small in absolute value relative to the positive eigenvalues, the inter-point distances of the new points in Euclidean space should provide a good approximation to the original inter-point distances (d_{ij}).

6.1.1 Example

Consider the five point in $\mathbb{R}^2$:

$$x_1 = (0,0)^\top, x_2 = (1,0)^\top, \quad x_3 = (0,1)^\top$$

$$x_4 = (-1,0)^\top \quad \text{and} \quad x_5 = (0,-1)^\top.$$

```
X <- matrix(c(0,0,
              1,0,
              0,1,
              -1,0,
              0,-1), nc=2, byrow=TRUE)
(D <- as.matrix(dist(X, upper=T, diag=T)))
```

```
##   1        2        3        4        5
## 1 0 1.000000 1.000000 1.000000 1.000000
## 2 1 0.000000 1.414214 2.000000 1.414214
## 3 1 1.414214 0.000000 1.414214 2.000000
## 4 1 2.000000 1.414214 0.000000 1.414214
## 5 1 1.414214 2.000000 1.414214 0.000000
```

```
(A <- -D^2/2)
```

```
##      1    2    3    4    5
## 1  0.0 -0.5 -0.5 -0.5 -0.5
## 2 -0.5  0.0 -1.0 -2.0 -1.0
## 3 -0.5 -1.0  0.0 -1.0 -2.0
## 4 -0.5 -2.0 -1.0  0.0 -1.0
## 5 -0.5 -1.0 -2.0 -1.0  0.0
```

```
H <- diag(5) - 1/5 * matrix(rep(1,5), nc=1)%*%matrix(rep(1,5), nr=1)
B <- H%*% A%*%H
print(round(B,0))
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    1    0   -1    0
## [3,]    0    0    1    0   -1
## [4,]    0   -1    0    1    0
## [5,]    0    0   -1    0    1
```

```
B.eig <-eigen(B)
Y<-sqrt(B.eig$values[1:2])*B.eig$vectors[,1:2]
dist(Y, upper=T, diag=T) # should agree with D.
```

```
##          1        2        3        4        5
## 1 0.000000 1.000000 1.000000 1.000000 1.000000
## 2 1.000000 0.000000 1.414214 2.000000 1.414214
```

```
## 3 1.000000 1.414214 0.000000 1.414214 2.000000
## 4 1.000000 2.000000 1.414214 0.000000 1.414214
## 5 1.000000 1.414214 2.000000 1.414214 0.000000
```

```
Y.mds <- cmdscale(D, eig=TRUE)
dist(Y.mds$points, upper=T, diag=T)
```

```
##          1        2        3        4        5
## 1 0.000000 1.000000 1.000000 1.000000 1.000000
## 2 1.000000 0.000000 1.414214 2.000000 1.414214
## 3 1.000000 1.414214 0.000000 1.414214 2.000000
## 4 1.000000 2.000000 1.414214 0.000000 1.414214
## 5 1.000000 1.414214 2.000000 1.414214 0.000000
```

The resulting distance matrix is

$$D = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & \sqrt{2} & 2 & \sqrt{2} \\ 1 & \sqrt{2} & 0 & \sqrt{2} & 2 \\ 1 & 2 & \sqrt{2} & 0 & \sqrt{2} \\ 1 & \sqrt{2} & 2 & \sqrt{2} & 0 \end{bmatrix}.$$

Using (6.2) first to calculate A, and then using (6.3) to calculate B, we find that

$$A = -\begin{bmatrix} 0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0 & 1 & 2 & 1 \\ 0.5 & 1 & 0 & 1 & 2 \\ 0.5 & 2 & 1 & 0 & 1 \\ 0.5 & 1 & 2 & 1 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}.$$

Further numerical calculations using R show that the eigenvalues of B are

$$\lambda_1 = \lambda_2 = 2 \qquad \text{and} \qquad \lambda_3 = \lambda_4 = \lambda_5 = 0.$$

Note that, as expected from Proposition 6.1, B is non-negative definite because it is a Euclidean distance matrix.

The following mutually orthogonal unit eigenvectors corresponding to the repeated eigenvalue 2 are produced by R:

$$q_1 = \begin{pmatrix} 0 \\ -0.439 \\ -0.554 \\ 0.439 \\ 0.554 \end{pmatrix} \qquad \text{and} \qquad q_2 = \begin{pmatrix} 0 \\ 0.554 \\ -0.439 \\ -0.554 \\ 0.439 \end{pmatrix}.$$

So the coordinates of five points in $\mathbb{R}^2$ which have the same inter-point distance matrix, D, as the original five points in $\mathbb{R}^2$, are given by the rows of the matrix

$$Q\Lambda^{1/2} = \sqrt{2}[q_1, q_2] = \begin{pmatrix} 0 & 0 \\ -0.621 & 0.784 \\ -0.784 & -0.621 \\ 0.621 & -0.784 \\ 0.784 & 0.621 \end{pmatrix}.$$

In the example sheets you asked to verify that there is an orthogonal transformation which maps the original five points onto the new five points.

6.2 Principal Coordinates

Starting with a distance matrix D, and using the matrix B, we now show how to calculate exact or approximate Euclidean coordinates for the n objects under study. We already know from Proposition 6.1 how to do this when the distance matrix D is Euclidean, but we will see now that this construction works more generally. Moreover, there is a very close connection with principal components analysis.

- **Step 1**: Given a distance matrix $\overset{n\times n}{D}$, calculate A according to (6.2).
- **Step 2**: Calculate $B = (b_{ij})_{i,j=1}^n$ in (6.3) using

$$b_{ij} = a_{ij} - \bar{a}_{i+} - \bar{a}_{+j} + \bar{a}_{++}, \qquad i, j = 1, \ldots, n,$$

  where

$$\bar{a}_{i+} = n^{-1}\sum_{j=1}^n a_{ij}, \quad \bar{a}_{+j} = n^{-1}\sum_{i=1}^n a_{ij} \quad \text{and} \quad \bar{a}_{++} = n^{-2}\sum_{i,j=1}^n a_{ij}.$$

- **Step 3**: Assume that the k largest eigenvalues of $B = (b_{ij})_{i,j=1}^n$, $\lambda_1 > \lambda_2 > \cdots > \lambda_k$ are all positive and have associated unit eigenvectors $v_1, \ldots, v_k$.
- **Step 4**: Define $V = [v_1, \ldots, v_k]$ and

$$X \equiv [x_1, \ldots, x_n]^\top = V\Lambda^{1/2} = [\sqrt{\lambda_1}v_1, \ldots, \sqrt{\lambda_k}v_k].$$

Then $x_i \in \mathbb{R}^k$, $i = 1, \ldots, n$, are the principal coordinates of the n points in k dimensions.

It turns out that there is a very close connection between principal coordinate and principal components.

Proposition 6.2. *Let X be an $n \times p$ data matrix with associated Euclidean distance matrix*

$$d_{ij}^2 = (x_i - x_j)^\top (x_i - x_j),$$

where $x_1^\top, \ldots, x_n^\top$ are the rows of X. Then the centred PC scores based on the first k principal components are principal coordinates of the n points in k dimensions based on the distance matrix D.

6.3 Similarity measures

Recap: so far in this chapter we have considered distances matrices $D = (d_{ij})_{i,j=1}^n$ with distances d_{ij}. In this setting, the larger d_{ij} is, the more distant, or dissimilar, object i is from object j.

Recall that we have distinguished between metric distances ("metrics"), which satisfy the triangle inequality (6.1), and non-metric distances, or dissimilarities, which need not satisfy (6.1).

In this section, we now consider the analysis of measures of *similarity* as opposed to measures of dissimilarity.

A *similarity* matrix is defined to be an $n \times n$ matrix $=(f_{ij})_{i,j=1}^n$ with the following properties:

1. Symmetry, i.e. $f_{ij} = f_{ji}$, $i, j = 1, \ldots, n$.
2. For all $i, j = 1, \ldots, n$, $f_{ij} \leq f_{ii}$.

Note that when working with similarities f_{ij}, the larger f_{ij} is, the more similar objects i and j are.

Condition 1. implies that object i is as similar to object j as object j is to object i (symmetry).

Condition 2. implies that an object is at least as similar to itself as it is to any other object.

One important class of problems is when the similarity between any two objects is measured by the number of common attributes. We illustrate this through two examples.

Example 6.1. Suppose there are 4 attributes we wish to consider.

1. Attribute 1: Carnivore? If yes, put $a_1 = 1$; if no, put $a_1 = 0$.
2. Attribute 2: Mammal? If yes, put $a_2 = 1$; if no, put $a_2 = 0$.
3. Attribute 3: Natural habitat in Africa? If yes, put $a_3 = 1$; if no, put $a_3 = 0$.
4. Attribute 4: Can climb trees? If yes, put $a_4 = 1$; if no, put $a_4 = 0$.

Consider a lion. Each of the attributes is present so $a_1 = a_2 = a_3 = a_4 = 1$.

A tiger? In this case, 3 of the attributes are present (1, 2 and 4) but 3 is absent. So for a tiger, $a_1 = a_2 = a_4 = 1$ and $a_3 = 0$.

How might we measure the similarity of lions and tigers based on the presence or absence of these four attributes?

First form a 2×2 table as follows.

	1	0
1	a	b
0	c	d

Here a counts the number of attributes common to both lion and tiger; b counts the number of attributes the lion has but the tiger does not have; c counts the number of attributes the tigher has that the lion does not have; and d counts the number of attributes which neither the lion nor the tiger has.

In the above, $a = 3$, $b = 1$ and $c = d = 0$.

How might we make use of the information in the 2×2 table to construct a measure of similarity?

The simplest measure of similarity is the proportion of the attributes which are shared.

$$\frac{a}{a+b+c+d},$$

which gives 0.75 in this example. A second similarity measure, which gives the same value in this example but not in general, is known as the *similarity matching coefficient* and is given by

$$\frac{a+d}{a+b+c+d}. \tag{6.6}$$

There are many other possibilities, e.g. we could consider weighted versions of the above if we wish to weight different attributes differently.

Example 6.2. Let us now consider a similar but more complex example with 6 unspecified attributes (not the same attributes as in Example 1) and 5 types of living creature, with the following data matrix, consisting of zeros and ones.

	1	2	3	4	5	6
$Lion$	1	1	0	0	1	1
$Giraffe$	1	1	1	0	0	1
Cow	1	0	0	1	0	1
$Sheep$	1	0	0	1	0	1
$Human$	0	0	0	0	1	0

Suppose we decide to use the similarity matching coefficient (6.6) to measure similarity. Then the following similarity matrix is obtained.

$F =$

	Lion	Giraffe	Cow	Sheep	Human
$Lion$	1	2/3	1/2	1/2	1/2
$Giraffe$	2/3	1	1/2	1/2	1/6
Cow	1/2	1/2	1	1	1/3
$Sheep$	1/2	1/2	1	1	1/3
$Human$	1/2	1/6	1/3	1/3	1

It is easily checked from the definition that $=(f_{ij})_{i,j=1}^{5}$ is a similarity matrix.

We now return to the general case. What should we do once we have calculated a similarity matrix? It turns out there is a nice transformation from a similarity matrix to a distance matrix $D = (d_{ij})_{i,j=1}^{n}$ defined by

$$d_{ij} = \left(f_{ii} + f_{jj} - 2f_{ij}\right)^{1/2}, \qquad i, j = 1, \ldots, n. \tag{6.7}$$

Note that, provided F is a similarity matrix, the d_{ij} are well-defined (i.e. real, not imaginary) because $f_{ii} + f_{jj} - 2f_{ij} \geq 0$ by condition 2., so the bracket is non-negative.

We have the following result.

Proposition 6.3. *Suppose that F is a similarity matrix. If, in addition, F is non-negative definite, then D defined in* (6.7) *is Euclidean with centred inner product matrix*

$$B = HFH, \tag{6.8}$$

where $H = \mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top$ is the centering matrix.

Proof. Since F is non-negative definite by assumption, and $H^\top = H$ by definition of H, it follows that HFH must also be non-negative definite. So by Result 5.1, we just need to show that (6.8) holds, where B is given by $B = HAH$ and A is defined as in (6.2), and the d_{ij} are defined by (6.7). Then

$$a_{ij} = -\frac{1}{2}d_{ij}^2 = f_{ij} - \frac{1}{2}(f_{ii} + f_{jj}).$$

Define

$$t = n^{-1}\sum_{i=1}^{n} f_{ii}.$$

Then, summing over $j = 1, \ldots, n$ for fixed i,

$$\bar{a}_{i+} = n^{-1}\sum_{j=1}^{n} a_{ij} = \bar{f}_{i+} - \frac{1}{2}(f_{ii} + t);$$

similarly,

$$\bar{a}_{+j} = n^{-1}\sum_{i=1}^{n} a_{ij} = \bar{f}_{+j} - \frac{1}{2}(f_{jj} + t),$$

and also

$$\bar{a}_{++} = n^{-2}\sum_{i,j=1}^{n} a_{ij} = \bar{f}_{++} - \frac{1}{2}(t + t).$$

So, using part (vii) of section 7 of Chapter 2 (FIX FIX),

$$\begin{aligned} b_{ij} &= a_{ij} - \bar{a}_{i+} - \bar{a}_{+j} + \bar{a}_{++} \\ &= f_{ij} - \frac{1}{2}(f_{ii} + f_{jj}) - \bar{f}_{i+} + \frac{1}{2}(f_{ii} + t) \\ &\qquad - \bar{f}_{+j} + \frac{1}{2}(f_{jj} + t) + \bar{f}_{++} - t \\ &= \qquad f_{ij} - \bar{f}_{i+} - \bar{f}_{+j} + \bar{f}_{++}. \end{aligned}$$

Consequently, $B = HFH$, using part (vii) of §2.7 again, and the result is proved. □ □

6.4 Exercises

1. In this question we will prove part (ii) of ????????????? Result 5.1 in the lecture notes.
   - Define
   $$b_{ij} = (x_i - \bar{x})^\top (x_j - \bar{x}), \quad i, j = 1, \ldots, n,$$
   and write $B = (b_{ij})_{i,j=1}^n$. Prove that
   $$B = (HX)(HX)^\top,$$
   where H is the $n \times n$ centering matrix and $X = (x_1, \ldots, x_n)^\top$ is the data matrix.
   - Show that B is non-negative definite (in which case part (ii) of Result 5.1 holds). **Hint**: explain why, for all $n \times 1$ vectors a,
   $$a^\top B a = a^\top (HX)(HX)^\top a \geq 0.$$
2. In this question we will prove part (iii) of Result 5.1.?????????????

```
- Prove Property 7. of Section \\ref{centering-matrix}.  Specifically: if $\bA$ is a  symmetric $
```

$$B = HAH$$

has elements given by

$$b_{ij} = a_{ij} - \bar{a}_{i+} - \bar{a}_{+j} + \bar{a}_{++}, \qquad i, j = 1, \ldots, n,$$

where

$$\bar{a}_{i+} = \frac{1}{n}\sum_{j=1}^n a_{ij}, \quad \bar{a}_{+j} = \frac{1}{n}\sum_{i=1}^n a_{ij}, \quad \text{and} \quad \bar{a}_{++} = \frac{1}{n^2}\sum_{i,j=1}^n a_{ij}.$$

```
- Assume that $\bB$ is non-negative definite with $k$ strictly positive eigenvalues and let
```

$$B = \sum_{j=1}^{k} \lambda_j q_j q_j^\top = Q \Lambda Q^\top,$$

where $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_k\}$ and Q is $n \times k$ and satisfies $Q^\top Q = \mathbf{I}_k$. Now define a 'new' $n \times k$ data matrix

$$X = [x_1, \ldots, x_n]^\top = Q \Lambda^{1/2}.$$

Show that $b_{ij} = x_i^\top x_j$ for all $i, j = 1, \ldots, n$.
Hint: check that for X defined as above, $XX^\top = B$.

```
- We now need to show that $\bD=(d_{ij})$ represents the set of inter-point distances i
```

$$(x_i - x_j)^\top (x_i - x_j) = b_{ii} + b_{jj} - 2b_{ij};$$

and so, using the first part of this question, show that

$$(x_i - x_j)^\top (x_i - x_j) = -2a_{ij} = d_{ij}^2.$$

Hence the new inter-point distances are the same as the original ones, and part (iii) of Result 5.1 is proved.

6.5 Computer Tasks

1. Look at this computational trick https://www.robots.ox.ac.uk/~albanie/notes/Euclidean_distance_trick.pdf for computing the distance matrix.
2. Do UK map?

```
mds <- cmdscale(distances)
plot(mds[,2], mds[,1])
text(mds[,2], mds[,1], labels=city$Name, pos=3)
```

```
cmdscale
```

3. http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/122-multidimensional-scaling-essentials-algorithms-and-r-code/ Visualizing a correlation matrix using Multidimensional Scaling

MDS can be also used to reveal a hidden pattern in a correlation matrix.

Correlation actually measures similarity, but it is easy to transform it to a measure of dissimilarity. Distance between objects can be calculated as 1 - res.cor.

```
res.cor <- cor(mtcars, method = "spearman")
mds.cor <-cmdscale((1 - res.cor) )
colnames(mds.cor) <- c("Dim.1", "Dim.2")

library(ggpubr)
ggscatter(mds.cor, x = "Dim.1", y = "Dim.2",
          label = rownames(swiss),
          size = 1,
          repel = TRUE)
```

Part III: Inference using the Multivariate Normal Distribution (MVN)

Part III of this module covers statistical inference based on the multivariate normal (MVN) distribution.

Chapter 7 focuses on classical distribution theory relating to the MVN distribution, including the Wishart distribution, which is defined on the set of symmetric positive definite matrices and is a natural generalisation of the χ^2 distribution. Another important distribution related to the MVN distribution is the Hotelling T^2 distribution, which is a multivariate analogue of the Student t-distribution. The Wishart and Hotelling T^2 distributions then allow us to conduct hypothesis tests concerning vector means in 1-sample and 2-sample settings.

Chapter **??** is concerned with the multivariate linear model, in which the responses consist of random vectors rather than single random variables. Errors in this setting take the form of random vectors.

The results in Part III turn out to be natural but non-trivial generalisations of the results in the univariate case.

Chapter 7

The Multivariate Normal Distribution

The multivariate normal distribution (MVN) generalises the univariate normal distribution from scalar to vector random variables. It is important for a number of reasons:

1. It is entirely defined by its mean vector μ and its covariance matrix Σ.
2. Zero correlation implies independence.
3. Linear functions of multivariate normal vectors are also multivariate normal vectors.
4. The multivariate version of the Central Limit Theorem means that it appears naturally throughout statistics.
5. It has simple geometric properties, and is easy to work with mathematically.

7.1 Definition and Properties of the MVN

Definition 7.1. A random vector $x = (x_1, \ldots, x_p)^\top$ has a p-dimensional MVN distribution if and only if $a^\top x$ is a univariate normal random variable for all fixed $p \times 1$ vectors a.

Notation: If x $(p \times 1)$ is MVN with mean μ and covariance matrix Σ then we write

$$x \sim N_p(\mu, \Sigma).$$

Proposition 7.1. *If* $\overset{p\times 1}{x}$ *is a multivariate normal random variable, then for each constant matrix* A $(q \times p)$ *and constant vector* c $(q \times 1)$, $y = Ax + c$ *has a* q*-dimensional MVN.*

Proof. Let b $(q \times 1)$ be a fixed vector. Then

$$b^\top y = b^\top Ax + b^\top c = a^\top x + b^\top c$$

where $a^\top = b^\top A$. Now $a^\top x$ is univariate normal for all a since x is MVN. Therefore $b^\top y$ is univariate normal for all b, so y is MVN. □

Corollary 7.1. *Any subset of the components of a MVN vector x is also MVN.*

Definition 7.2. If the population covariance matrix Σ $(p \times p)$ is positive definite (i.e. full rank), so that Σ^{-1} exists, then the **probability density function** (pdf) of the MVN distribution is given by

$$f(x) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right).$$

If $p = 1$, so that $x = x$, $\mu = \mu$ and $\Sigma = \sigma^2$, say, then the pdf simplifies to

$$\begin{aligned} f(x) &= \frac{1}{|2\pi\sigma^2|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)(\sigma^2)^{-1}(x-\mu)\right) \\ &= \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \end{aligned}$$

which is the familiar pdf of the univariate normal distribution $N(\mu, \sigma^2)$.

If $p > 1$ and $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$ then

$$\begin{aligned} f(x) &= \frac{1}{(2\pi)^{p/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) \\ &= \frac{1}{(2\pi)^{p/2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{p} \frac{(x_i-\mu_i)^2}{\sigma_i^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_1^2}(x_1-\mu_1)^2\right)\right) \\ &\qquad \times \ldots \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_p^2}(x_p-\mu_p)^2\right)\right) \end{aligned}$$

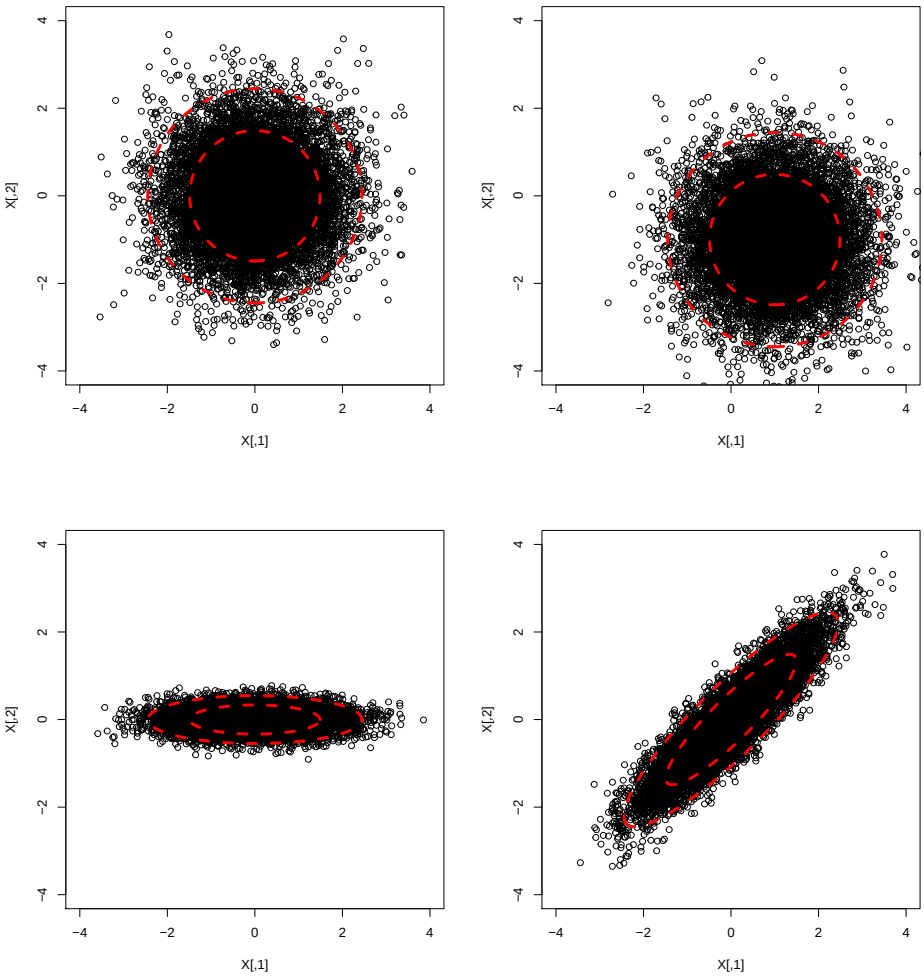
Thus, by the factorisation theorem for probability densities, the components of x have independent univariate normal distributions: $x_i \sim N(\mu_i, \sigma_i^2)$.

If $p = 2$ we can plot $f(x)$ using contour plots. Below, I've generated 1000 points from four different normal distributions using mean vectors

$$\mu_1 = \mu_3 = \mu_4 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

and covariance matrices

$$\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & 0.05 \end{pmatrix}, \quad \Sigma_4 = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}.$$



The contours on each plot are obtained by finding values of x for which $f(x) = c$. The constant c is chosen so that the the shapes enclose 66% and 95% of the data.

Ellipses

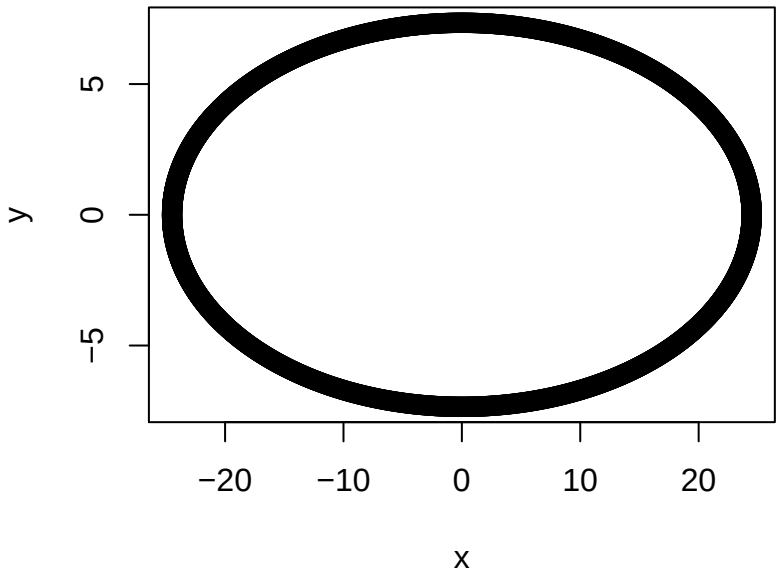
What is the shape of the contours in the plots above? They are defined by $f(x) = c$, which implies

$$(x - \mu)^\top \Sigma^{-1} (x - \mu) = c' \tag{7.1}$$

for some constant c'. This is the equation of an ellipse. To see this, note that a standard ellipse in $\mathbb{R}^2$ is given by the equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (a > b > 0). \tag{7.2}$$

and recall that a standard ellipse has axes of symmetry given by the x-axis and y-axis (if $a > b$, the x-axis is the major axis, and the y-axis the minor axis). For example, $a = 10, b = 3$ gives the ellipse:



If we define $\mathbf{A} = \begin{pmatrix} a^2 & 0 \\ 0 & b^2 \end{pmatrix}$ and write $\mathbf{x} = \binom{x}{y}$, then Equation (7.2) can be written in the form

$$x^\top \mathbf{A}^{-1} x = c'.$$

To shift the centre of the ellipse from the origin to the point μ we modify the equation to be

$$(x-\mu)^\top \mathbf{A}^{-1}(x-\mu) = c'.$$

What if instead of using a diagonal matrix A, we use a non-diagonal matrix Σ as in Equation (7.1)? If Σ has spectral decomposition $\Sigma = V\Lambda V^\top$, then

$$(x-\mu)^\top \Sigma^{-1}(x-\mu) = (x-\mu)^\top V\Lambda^{-1}V^\top(x-\mu) = y^\top \Lambda^{-1} y$$

where Λ is a diagonal matrix of eigenvalues, and $y = V^\top(x-u)$. Because V is an orthogonal matrix (a rotation), we can see that this is the equation of a standard ellipse when using the eigenvectors as the coordinate system. Or in other words, it is an ellipse with major axis given by the first eigenvector, and minor axis given by the second eigenvector, centered aroubd μ.

Analogous results for ellipsoids and quadratic forms hold in three and higher dimensions.

7.1.1 Transformations

Proposition 7.2. *If $x \sim N_p(\mu, \Sigma)$ and $y = Ax + c$, where A $(q \times p)$ and c $(q \times 1)$ are constant, then*

$$y \sim N_q(A\mu + c, A\Sigma A^\top).$$

Proof. We know y is MVN by Proposition 7.1. We also know $\mathbb{E}(y)$ and $\mathbb{V}\mathrm{ar}(y)$ from Section **??**. □

This implies that a linear transformation of a MVN random variable is also MVN. We can use this result to prove two important corollaries. The first corollary is useful for simulating data from a general MVN distribution.

Corollary 7.2. *If $x \sim N_p(0, \mathbf{I}_p)$ and $y = \Sigma^{1/2}x + \mu$ then*

$$y \sim N_p(\mu, \Sigma).$$

Proof. Apply 7.2 with $A = \Sigma^{1/2}$ and $c = \mu$. Therefore

$$\mathbb{E}(y) = \Sigma^{1/2}0_p + \mu = \mu \quad \text{and} \quad \mathbb{V}\mathrm{ar}(y) = \Sigma^{1/2}\mathbf{I}_p\Sigma^{1/2} = \Sigma.$$

□

The second corollary says that any MVN random variable can be transformed into standard form.

Corollary 7.3. *Suppose $x \sim N_p(\mu, \Sigma)$, where Σ has full rank. Then*

$$y = \Sigma^{-1/2}(x - \mu) \sim N_p(0, \mathbf{I}_p).$$

Proof. Apply Proposition 7.2 with $A = \Sigma^{-1/2}$ and $c = -\Sigma^{-1/2}\mu$.

□

7.1.2 Moment Generating Functions

Definition 7.3. The **moment generating function** of a random vector $x \in \mathbb{R}^p$ is given by

$$M(t) = \mathbb{E}[e^{t^\top x}],$$

and is defined for all $t \in \mathbb{R}^p$ for which $M(t)$ is finite.

Proposition 7.3. *The moment generating function of $x \sim N_p(\mu, \Sigma)$ is given by*

$$M(t) = \exp\left(\mu^\top t + \frac{1}{2}t^\top \Sigma t\right). \tag{7.3}$$

Proof. For fixed t, define the random variable $Y = x^\top t$. From Proposition 7.2, $Y \sim N(\mu_t, {\sigma_t}^2)$, where $\mu_t = \mu^\top t$ and $\sigma_t^2 = t^\top \Sigma t$.

If $\sigma_t = 0$ then $Y = \mu^\top t$ with probability one, and $M(t) = e^{\mu^\top t}$ which agrees with (7.3).

Assume $\sigma_t > 0$. Then

$$\begin{aligned} M(t) &= \mathbb{E}[e^{x^\top t}] \\ &= \mathbb{E}[e^Y] = \int_{-\infty}^{\infty} \exp(y) \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{1}{2}\frac{(y-\mu_t)^2}{\sigma_t^2}\right) dy. \end{aligned}$$

The integral above can be evaluated by completing the square in the exponent, using the identity

$$y - \frac{1}{2}\frac{(y-\mu_t)^2}{\sigma_t^2} = \mu_t + \frac{1}{2}\sigma_t^2 - \frac{1}{2}\frac{(y-\mu_t-\sigma_t^2)^2}{\sigma_t^2}.$$

Consequently

$$\begin{aligned} M(t) &= \int_{-\infty}^{\infty} \exp\left\{\mu_t + \frac{1}{2}\sigma_t^2\right\} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left\{-\frac{1}{2}\frac{(y-\mu_t-\sigma_t^2)^2}{\sigma_t^2}\right\} dy \\ &= \exp\left(\mu_t + \frac{1}{2}\sigma_t^2\right) \\ &= \exp\left(\mu^\top t + \frac{1}{2} t^\top \Sigma t\right), \end{aligned}$$

as required. □

Proposition 7.4. *Two vectors x $(p \times 1)$ and y $(q \times 1)$ which are jointly multivariate normal are independent if and only if they are uncorrelated, i.e. $\mathbb{C}\,\mathrm{ov}(x, y) = 0_{p,q}$.*

Proof. We prove this result using the factorisation theorem for moment generating functions (MGFs), which is now stated. Let

$$t = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

where $t_1 \in \mathbb{R}^p$, $t_2 \in \mathbb{R}^q$ and $t \in \mathbb{R}^{p+q}$. The joint MGF of two arbitrary random vectors $\overset{p\times 1}{x}$ and $\overset{q\times 1}{y}$ is

$$M(t_1, t_2) = \mathbb{E}[e^{t_1^\top x + t_2^\top y}].$$

The factorisation theorem for MGFs states that x and y are independent if and only if $M(t_1, t_2)$ factorises, i.e.,

$$M(t_1, t_2) = M_1(t_1) M_2(t_2)$$

for some functions M_1 and M_2, in which case M_1 and M_2 are the marginal MGFs of x and y.

Now suppose x and y are multivariate normal random variables with

$$\mathbb{E}[x] = \mu_x, \qquad \mathbb{E}[y] = \mu_y, \quad \mathbb{V}\,\mathrm{ar}(x) = \Sigma_{xx}, \quad \mathbb{V}\,\mathrm{ar}(y) = \Sigma_{yy}, \tag{7.4}$$

and

$$\mathbb{C}\,\mathrm{ov}(x, y) = \Sigma_{xy} = \Sigma_{yx}^\top = \mathbb{C}\,\mathrm{ov}(y, x)^\top. \tag{7.5}$$

Using Proposition 7.3 and definitions (7.4) and (7.5),

$$\begin{aligned} M(t_1, t_2) &= \exp\left(\mu^\top t + \frac{1}{2} t^\top \Sigma t\right) \\ &= \exp\left(\mu_x^\top t_1 + \mu_y^\top t_2 + \frac{1}{2} t_1^\top \Sigma_{xx} t_1 \right. \\ &\qquad \left. + \frac{1}{2} t_2^\top \Sigma_{yy} t_2 + \frac{1}{2} 2 t_1^\top \Sigma_{xy} t_2\right) \\ &= M_1(t_1) M_2(t_2) M_3(t_1, t_2), \end{aligned}$$

where $M_1(t_1)$ and $M_2(t_2)$ are the marginal MGFs of x and y respectively, and

$$M_3(t_1, t_2) = \exp\left(t_1^\top \Sigma_{xy} t_2\right).$$

Thus, by the factorisation theorem, x and y are independent if and only if $M_3(t_1, t_2)$ is constant with respect to t_1 and t_2, which is the case if and only if $\Sigma_{xy} = \mathbf{0}_{p,q}$. □

Proposition 7.4 means that zero correlation implies independence for the MVN distribution. This is not true in general for other distributions.

Note: Propositions 7.1 - 7.4 holds regardless irregardless of whether the covariance matrix Σ is invertible.

The term $(x-\mu)^\top \Sigma^{-1}(x-\mu)$ appears in the exponent of the pdf and will be important later. We now derive its distribution:

Proposition 7.5. *If $x \sim N_p(\mu, \Sigma)$ and Σ is positive definite then*

$$(x-\mu)^\top \Sigma^{-1}(x-\mu) \sim \chi_p^2.$$

Proof. Define $y = \Sigma^{-1/2}(x-\mu)$ so

$$\begin{aligned} (x-\mu)^\top \Sigma^{-1}(x-\mu) &= \left(\Sigma^{-1/2}(x-\mu)\right)^\top \left(\Sigma^{-1/2}(x-\mu)\right) \\ &= y^\top y = \sum_{i=1}^p y_i^2 \end{aligned}$$

By Corollary 7.3, $y \sim N_p(0, \mathbf{I}_p)$, and so the components of y have independent univariate normal distributions with mean 0 and variance 1. Recall from univariate statistics that if $z \sim N(0,1)$ then $z^2 \sim \chi^2_1$ and if $z_1, \ldots, z_n$ are iid $N(0,1)$ then $\sum_{i=1}^n z_i^2 \sim \chi^2_n$. It therefore follows that

$$\sum_{i=1}^{p} y_i^2 \sim \chi^2_p.$$

□

We saw earlier in this section chapter that the MVN distribution in p dimensions has constant density on ellipses or ellipsoids given by $f(x) = c$ for some constant $c > 0$, and that we can rearrange this equation to be of the form

$$U(x) = (x-\mu)^\top \Sigma^{-1}(x-\mu) = k$$

where $k = -2\log(c) - \log|2\pi\Sigma| > 0$ is a combination of the constant, c, and the normalising constant in the pdf. Proposition 7.5 means we can calculate the probability, $P(U(x) < k)$, which is the probability of x lying within a particular ellipsoid.

7.1.3 Sampling results for the MVN

In this section we present two important results which are natural generalisations of what happens in the univariate case.

Proposition 7.6. *If $x_1, \ldots, x_n$ is an IID random sample from $N_p(\mu, \Sigma)$, then the sample mean and sample variance matrix*

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top$$

are independent.

Proof. From Proposition 7.1 and Proposition 7.2 we can see that if $x_1, \ldots, x_n \sim N_p(\mu, \Sigma)$ then $\bar{x} \sim N_p(\mu, n^{-1}\Sigma)$. Then

$$\begin{aligned}
\mathbb{C}\,\mathrm{ov}(\bar{x}, y_i) &= \mathbb{C}\,\mathrm{ov}(\bar{x}, x_i - \bar{x}) \\
&= \mathbb{C}\,\mathrm{ov}(\bar{x}, x_i) - \mathbb{C}\,\mathrm{ov}(\bar{x}, \bar{x}) \\
&= n^{-1}\sum_{j=1}^{n}\left\{\mathbb{E}[(x_j - \mu)(x_i - \mu)^\top]\right\} \\
&\qquad - \mathbb{E}[(\bar{x} - \mu)(\bar{x} - \mu)^\top] \\
&= n^{-1}\Sigma - n^{-1}\Sigma \\
&= \mathbf{0}_{p,p}.
\end{aligned}$$

Thus Proposition 7.4 gives that $\bar{x}$ and y_i are independent, and therefore $\bar{x}$ and

$$S = \frac{1}{n}\sum_{i=1}^{n} y_i y_i^\top = n^{-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top$$

are independent.

□

Recall from above that if $x_1, \ldots, x_n$ is a random sample from $N_p(\mu, \Sigma)$ then

$$\bar{x} \sim N_p(\mu, \frac{1}{n}\Sigma).$$

This result is also approximately true for large samples from non-normal distributions, as is now stated in the multivariate central limit theorem.

Proposition 7.7. ***Central limit theorem*** *Let $x_1, x_2, \ldots \in \mathbb{R}^p$ be a sample of independent and identically distributed random vectors from a distribution with mean μ and finite variance matrix Σ. Then asymptotically as $n \to \infty$, $\sqrt{n}(\bar{x}-\mu)$ converges in distribution to $N_p(\mathbf{0}_p, \Sigma)$.*

Proof. Beyond the scope of this module. □

7.2 The Wishart distribution

The Wishart distribution is a multivariate generalisation of the univariate χ^2 distribution. In univariate statistics the χ^2 distribution plays an important role in inference related to the univariate normal, e.g. in the definition of Student's t distribution. An analogous role is played by the Wishart distribution in multivariate statistics. In this section we introduce the Wishart distribution and show that for MVN random variables, the sample covariance matrix S has a Wishart distribution.

Definition 7.4. Let $x_1, \ldots, x_n$ be an IID random sample from $N_p(0, \Sigma)$. Then

$$M = \sum_{i=1}^{n} x_i x_i^\top \in \mathbb{R}^{p\times p}$$

is said to have a Wishart distribution with n degrees of freedom and scale matrix Σ. We write this as

$$M \sim W_p(\Sigma, n).$$

and refer to $W_p(\mathbf{I}_p, n)$ as a standard Wishart distribution.

Note:

- $W_p(\Sigma, n)$ is a probability distribution on the set of $p \times p$ symmetric non-negative definite random matrices.

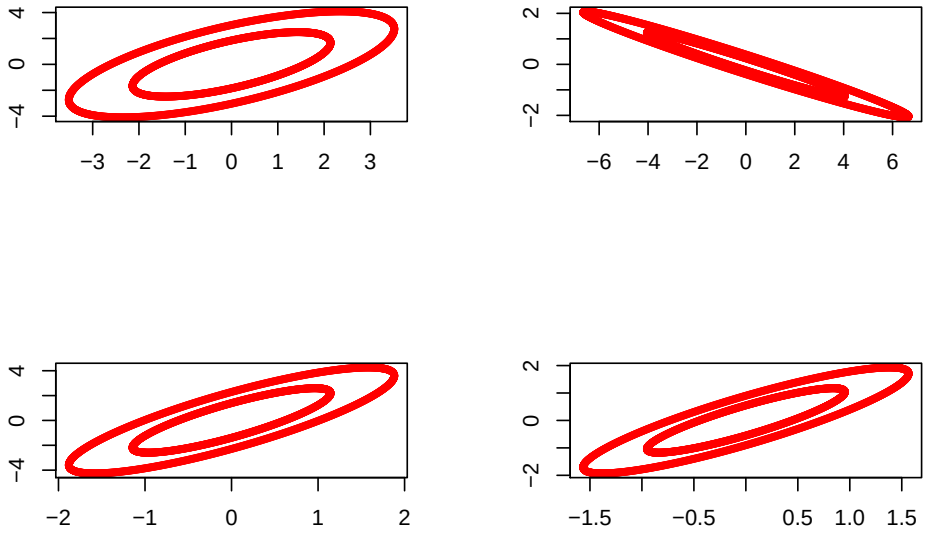
- When $p = 1$, $W_1(1, n)$ is the χ^2_n distribution and $W_1(\sigma^2, n)$ is the $\sigma^2\chi^2_n$ distribution. This claim follows from 7.10 below.
- If X is the usual $n \times p$ matrix with rows $x_i^\top$, then

$$M = X^\top X.$$

We can sample from the Wishart distribution in R using the `rWishart` command. For example, setting $\Sigma = \mathbf{I}_2$ and using 2 degrees of freedom, we can generate 4 random samples $M_1, \ldots, M_4 \sim W_2(\mathbf{I}_2, 2)$ as follows:

```
out <- rWishart(n=4, df=2, Sigma=diag(1,2))
```

Visualizing these by plotting the ellipses with $x^\top M_i x = c$ for some constant c, we can see the variability in these random matrices:



Proposition 7.8. *Let $M \sim W_p(\Sigma, n)$. Then*

$$\mathbb{E}M = n\Sigma$$

and if the ij^{th} element of Σ is σ_{ij}, and the ij^{th} element of M is m_{ij}, then

$$\mathbb{V}\mathrm{ar}(m_{ij}) = n\left(\sigma_{ij}^2 + \sigma_{ii}\sigma_{jj}\right)$$

7.2.1 Properties

We now use the definition of $W_p(\Sigma, n)$ to prove some important results.

Proposition 7.9. *If $M \sim W_p(\Sigma, n)$ and A is a fixed $q \times p$ matrix, then*

$$AMA^\top \sim W_q\left(A\Sigma A^\top, n\right).$$

Proof. From the definition, let $M = \sum_{i=1}^{n} x_i x_i^\top$, where the x_i are IID $N_p(0, \Sigma)$. Then

$$\begin{aligned} AMA^\top &= A\left(\sum_{i=1}^{n} x_i x_i^\top\right) A^\top \\ &= \sum_{i=1}^{n} (Ax_i)(Ax_i)^\top = \sum_{i=1}^{n} y_i y_i^\top \end{aligned}$$

where $y_i = Ax_i \sim N_q(0, A\Sigma A^\top)$, by Proposition 7.2. Now we apply the definition of the Wishart distribution to $y_1, \ldots, y_n$ and, hence, $\sum_{i=1}^{n} y_i y_i^\top \sim W_q\left(A\Sigma A^\top, n\right)$. □

Proposition 7.10. *If $M \sim W_p(\Sigma, n)$ and a is a fixed $p \times 1$ vector then*

$$a^\top M a \sim \left(a^\top \Sigma a\right) \chi_n^2.$$

Proof. Apply Proposition 7.9 with $A = a^\top$ then $a^\top M a \sim W_1(a^\top \Sigma a, n)$. But $W_1(1, n)$ is equal in distribution to $\sum_{i=1}^{n} z_i^2$ where the z_i are IID $N(0, 1)$, and so has χ_n^2 distribution. Moreover, using Proposition 7.9 with $p = q = 1$ and $A = \sigma$, it is seen that $W_1(\sigma^2, n)$ is equal in distribution to $\sigma^2 \chi_n^2$. □

Note that an alternative form of the above result is

$$\frac{a^\top M a}{a^\top \Sigma a} \sim \chi_n^2.$$

Corollary 7.4. *Let m_{ii} and σ_{ii} be the ith diagonal entry for M and Σ respectively, then $m_{ii} \sim \sigma_{ii} \chi_n^2$ for $i = 1, \ldots, p$.*

Proof. Let $a = (a_1, \ldots, a_p)^\top$ where $a_j = 1$ if $j = i$ and $a_j = 0$ otherwise. Then $a^\top M a = m_{ii}$ and $a^\top \Sigma a = \sigma_{ii}$. Now apply Proposition 7.10

□

Note, however, that the m_{ii}, $i = 1, \ldots, p$, are not, in general, independent.

Proposition 7.11. *If $M_1 \sim W_p(\Sigma, n_1)$ and $M_2 \sim W_p(\Sigma, n_2)$ are independent then*

$$M_1 + M_2 \sim W_p(\Sigma, n_1 + n_2).$$

Proof. From the definition, let $M_1 = \sum_{i=1}^{n_1} x_i x_i^\top$ and let $M_2 = \sum_{i=n_1+1}^{n_1+n_2} x_i x_i^\top$, where $x_i \sim N_p(0, \Sigma)$, then $M_1 + M_2 = \sum_{i=1}^{n_1+n_2} x_i x_i^\top \sim W_p(\Sigma, n_1 + n_2)$ by the definition of the Wishart distribution. □

7.2.2 Cochran's theorem

Our next result is known as Cochran's theorem. We use Cochran's theorem to show that sample covariance matrices have a scaled Wishart distribution.

First though, recall the definition of projection matrices from Section 2.3.3. Namely, that P is a projection matrix if $P^2 = P$.

Theorem 7.1. ***(Cochran's Theorem)*** *Suppose* $\overset{n\times n}{\mathbf{P}}$ *is a projection matrix of rank* r*. Assume that* X *is an* $n \times p$ *data matrix with IID rows that have a common* $N_p(\mathbf{0}_p, \Sigma)$ *distribution, where* Σ *has full rank* p*, and note the identity*

$$X^\top X = X^\top \mathbf{P} X + X^\top (\mathbf{I}_n - \mathbf{P}) X. \tag{7.6}$$

Then

$$X^\top \mathbf{P} X \sim W_p(\Sigma, r), \qquad X^\top (\mathbf{I}_n - \mathbf{P}) X \sim W_p(\Sigma, n-r), \tag{7.7}$$

and $X^\top \mathbf{P} X$ *and* $X^\top (\mathbf{I}_n - \mathbf{P}) X$ *are independent.*

Proof. We first of all prove the result in the particular case $\Sigma = \mathbf{I}_p$ and then consider the general case. Using the Spectral Decomposition Theorem 3.3 and noting that the eigenvalues of projection matrices must be either 0 or 1, we may write

$$\mathbf{P} = \sum_{j=1}^{r} q_j q_j^\top \qquad \text{and} \qquad (\mathbf{I}_n - \mathbf{P}) = \sum_{j=r+1}^{n} q_j q_j^\top$$

where $q_1, \ldots, q_n \mathbb{R}^n$ are mutually orthogonal unit vectors. Then

$$\begin{aligned} X^\top P X &= X^\top \left(\sum_{j=1}^{r} q_j q_j^\top \right) X \\ &= \sum_{j=1}^{r} X^\top q_j q_j^\top X = \sum_{j=1}^{r} y_j y_j^\top, \end{aligned} \tag{7.8}$$

and similarly,

$$\begin{aligned} X^\top (\mathbf{I}_n - P) X &= X^\top \left(\sum_{j=r+1}^{n} q_j q_j^\top \right) X \\ &= \sum_{j=r+1}^{n} X^\top q_j q_j^\top X = \sum_{j=r+1}^{n} y_j y_j^\top, \end{aligned} \tag{7.9}$$

where $y_j = X^\top q_j$ is a $p \times 1$ vector. We shall now prove that the y_j are IID $N_p(\mathbf{0}_p, \mathbf{I}_p)$. Write $X = (x_{[1]}, \ldots, x_{[p]})$, where $x_{[u]}$ is column u of X. Then

$x_{[1]}, \ldots, x_{[p]}$ are IID $N_n(\mathbf{0}_n, \mathbf{I}_n)$. Moreover,

$$y_j = X^\top q_j = \begin{bmatrix} q_j^\top x_{[1]} \\ q_j^\top x_{[2]} \\ .. \\ .. \\ .. \\ q_j^\top x_{[p]} \end{bmatrix}.$$

But

$$\begin{aligned} \mathbb{E}[q_j^\top x_{[u]} q_k^\top x_{[v]}] &= \mathbb{E}[q_j^\top x_{[u]} x_{[v]}^\top q_k] \\ &= q_j^\top \mathbb{E}[x_{[u]} x_{[v]}^\top] q_k \\ &= q_j^\top \left(\delta_{uv} \mathbf{I}_n\right) q_k \\ &= q_j^\top q_k \delta_{uv} \\ &= \delta_{jk} \delta_{uv}, \end{aligned} \tag{7.10}$$

where δ is the Kronecker δ defined by

$$\delta_{ab} = \begin{cases} 0 & \text{if} \quad a \neq b \\ 1 & \text{if} \quad a = b \end{cases}.$$

If follows immediately from (7.10) that

$$\mathbb{V}\,\mathrm{ar}(y_j) = \mathbf{I}_p \qquad \mathbb{C}\,\mathrm{ov}(y_j, y_k) = \mathbf{0}_{pp} \quad \text{if} \quad j \neq k.$$

By Proposition 7.4, the y_j, $j = 1, \ldots, n$, are IID $N_p(\mathbf{0}_p, \mathbf{I}_p)$, and therefore it follows from the definition of the Wishart distribution that, when $\Sigma = \mathbf{I}_p$, (7.8) has a Wishart $W_p(\mathbf{I}_p, r)$ distribiton, (7.9) has a Wishart $W_p(\mathbf{I}_p, n-r)$ distrubtion. Moreover, these random Wishart matrices are independent becasue the y_j are all independent.

Finally, we consider the case of a general covariance matrix Σ. We have proved that (7.6) holds when $\Sigma = \mathbf{I}_p$, so pre-multiply both sides by the matrix square root $\Sigma^{1/2}$, and post-multiply both sides by $\Sigma^{1/2}$. This corresponds to the case where the x_i are IID $N_p(\mathbf{0}_p, \Sigma)$. Then, using Proposition 7.9,

$$\Sigma^{1/2} W_p(\mathbf{I}_p, t) \Sigma^{1/2} \stackrel{d}{=} W_p(\Sigma^{1/2} \Sigma^{1/2}, t) \stackrel{d}{=} W_p(\Sigma, t),$$

when $t = r$ and $t = n - r$. Moreover, since $\Sigma^{1/2}$ is a non-random matrix, independence is preserved when we pre- and post-multiply by $\Sigma^{1/2}$, and the result is proved. □

Proposition 7.12. *If $x_1, \ldots, x_n$ is an IID sample from $N_p(\mu, \Sigma)$, then*

$$nS = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^\top \sim W_p(\Sigma, n-1).$$

Proof. Define $P = \mathbf{H} \equiv \mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top$ where $\mathbf{1}_n$ is the $n \times 1$ vector of ones. Note that H is the $n \times n$ centering matrix and, from Property 1. of 2.4, H is a projection matrix. Clearly, $\mathbf{I}_n - P = n^{-1}\mathbf{1}_n\mathbf{1}_n^\top$ has rank 1, so H has rank $n-1$. Therefore, using Theorem 7.1 ,

$$X^\top HX \sim W_p(\Sigma, n-1).$$

But from Property 6. in Section 2.4, $X^\top HX = nS$, and consequently, $nS \sim W_p(\Sigma, n-1)$, as required. □

7.3 Hotelling's T^2 distribution

Recall that in univariate statistics, Student's t-distribution appears as the sampling distribution of $\frac{\bar{x}-\mu}{s/\sqrt{n}}$, which is used for hypothesis tests and constructing confidence intervals.

Hotelling's T^2 distribution is the multivariate analogue of Student's t-distribution. It plays an important role in multivariate hypothesis testing and confidence region construction, just as the Student t-distribution does in the univariate setting.

Definition 7.5. Suppose $x \sim N_p(0, \mathbf{I}_p)$ and $M \sim W_p(\mathbf{I}_p, n)$ are independent, then the quantity $\tau^2 = nx^\top M^{-1}x$ is said to have **Hotelling's T^2 distribution** with parameters p and n. We write this as $\tau^2 \sim T^2(p, n)$.

We can generalise the definition with the following result.

Proposition 7.13. *Suppose $x \sim N_p(\mu, \Sigma)$ and $M \sim W_p(\Sigma, n)$ are independent and Σ has full rank p. Then*

$$n(x-\mu)^\top M^{-1}(x-\mu) \sim T^2(p, n).$$

Proof. Define $y = \Sigma^{-1/2}(x-\mu)$. Then, by Corollary 7.3, $y \sim N_p(0, \mathbf{I}_p)$. Further, let $Z = \Sigma^{-1/2}M\Sigma^{-1/2}$ then $Z \sim W_p(\mathbf{I}_p, n)$ by applying 7.9 with $A = \Sigma^{-1/2}$. From the definition, $ny^\top Z^{-1}y \sim T^2(p, n)$ and

$$\begin{aligned} ny^\top Z^{-1}y &= n(x-\mu)^\top\Sigma^{-1/2}\Sigma^{1/2}M^{-1}\Sigma^{1/2}\Sigma^{-1/2}(x-\mu) \\ &= n(x-\mu)^\top M^{-1}(x-\mu) \end{aligned}$$

so the result is proved. □

This result gives rise to an important corollary used in hypothesis testing when Σ is unknown.

Corollary 7.5. *If $\bar{x}$ and S are the mean and covariance matrix based on a sample of size n from $N_p(\mu, \Sigma)$ then*

$$(n-1)(\bar{x}-\mu)^\top S^{-1}(\bar{x}-\mu) \sim T^2(p, n-1).$$

Proof. We have seen earlier that $\bar{x} \sim N_p(\mu, \frac{1}{n}\Sigma)$. Let $x^* = n^{1/2}\bar{x}$ and let $\mu^* = n^{1/2}\mu$. Then $x^* = n^{1/2}\bar{x} \sim N_p(\mu^*, \Sigma)$.

From Proposition 7.12 we know $nS \sim W_p(\Sigma, n-1)$, and from Theorem 7.6 we know $\bar{x}$ and S are independent. Applying Proposition 7.13 with $x = x^*$ and $M = nS$ we obtain

$$(n-1)(x^* - \mu^*)^\top (nS)^{-1}(x^* - \mu^*) \sim T^2(p, n-1),$$

and given $x^* - \mu^* = n^{1/2}(x - \mu)$ then

$$\begin{aligned}(n-1)&(x^* - \mu^*)^\top (nS)^{-1}(x^* - \mu^*) \\ &= (n-1)n^{1/2}(\bar{x} - \mu)^\top n^{-1} S^{-1} n^{1/2}(\bar{x} - \mu) \\ &= (n-1)(\bar{x} - \mu)^\top S^{-1}(\bar{x} - \mu).\end{aligned}$$

□

Hotelling's T^2 distribution is not often included in statistical tables but the next result tells us that Hotelling's T^2 is a scale transformation of an F distribution.

Proposition 7.14. *If* $\tau^2 \sim T^2(p, n)$ *then*

$$\gamma^2 = \frac{n-p+1}{np}\tau^2 \sim F_{p,n-p+1}.$$

Proof. Beyond the scope of the module. □

We can apply this result to the previous corollary.

Corollary 7.6. *If* $\tau^2 = (n-1)(\bar{x} - \mu)^\top S^{-1}(\bar{x} - \mu)$ *then*

$$\gamma^2 = \frac{n-p}{p}(\bar{x} - \mu)^\top S^{-1}(\bar{x} - \mu) \sim F_{p,n-p}.$$

Proof. From Corollary 7.6 we know $\tau^2 \sim T^2(p, n-1)$. Applying Proposition 7.14 we get

$$\begin{aligned}&\gamma^2 \\ &= \frac{(n-1)-p+1}{(n-1)p}(n-1)(\bar{x} - \mu)^\top S^{-1}(\bar{x} - \mu) \sim F_{p,(n-1)-p+1} \\ &= \frac{n-p}{p}(\bar{x} - \mu)^\top S^{-1}(\bar{x} - \mu) \sim F_{p,n-p}\end{aligned}$$

□

7.4 Inference based on the MVN

In univariate statistical analysis, you will have seen how to do hypothesis testing for the mean of a population.

1. In the case where you have a single sample $x_1, \ldots, x_n$ which come from a population with known variance σ^2 we use a z-test when testing hypotheses such as $H_0 : \mu = \mu_1$.
2. When the variance of the population, σ^2, is unknown, then we have to use a t-test.
3. When we have two samples, $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$, we use either a paired or an unpaired t-test.

We now develop analogous results in the multivariate case. The role of the Student t distribution will be played by Hotelling's T^2, and the role of the χ^2 is played by the Wishart distribution.

The next three subsections deal with the multivariate equivalent of the three situations listed above. Before we do, lets quickly recap how hypothesis testing works:

Recap of hypothesis testing framework

Suppose that we have a null hypothesis H_0 represented by a completely specified model and that we wish to test this hypothesis using data $x_1, \ldots, x_n$. We proceed as follows

1. Assume H_0 is true.
2. Find a test statistic $T(x_1, \ldots, x_n)$ for which large values indicate departure from H_0.
3. Calculate the theoretical sampling distribution of T under H_0.
4. The observed value $T_{obs} = T(x_1, \ldots, x_n)$ of the test statistic is compared with the distribution of T under H_0. Either
   - (Neyman-Pearson) reject H_0 if $T_{obs} > c$. Here c is chosen so that $\mathbb{P}(T \geq c|H_0) = \alpha$ where α is the **size** of the test, i.e., $\mathbb{P}(\text{reject } H_0|H_0 \text{ true}) = \alpha$.
   - (Fisherian) compute the p-value $p = \mathbb{P}(T \geq T_{obs}|H_0)$ and report it. This represents the strength of evidence against H_0.

7.4.1 Σ known

Let $x_1, \ldots, x_n$ be a random sample from $N_p(\mu, \Sigma)$ where $\mu = (\mu_1, \ldots, \mu_p)^\top$. Suppose we wish to conduct the following hypothesis test

$$H_0 : \mu = a \text{ vs } H_1 : \mu \neq a$$

where a is fixed and pre-specified. Let's first **assume that Σ is known**. This will result in the multivariate analogue of the z-test.

One approach would be to conduct p separate univariate z-tests tests with null hypotheses

$$H_0 : \mu_i = a_i \qquad \text{vs.} \qquad H_1 : \mu_i \neq a_i, \quad \text{for } i = 1, \ldots, p.$$

However, this ignores possible correlations between the variables - see the example for a situation in which this can make a difference.

A better approach is to conduct a single (multivariate) hypothesis test using the test statistic

$$\zeta^2 = n(\bar{x} - a)^\top \Sigma^{-1} (\bar{x} - a).$$

We need to compute the distribution of ζ^2 when H_0 is true. Note that

$$\zeta^2 = (n^{1/2}\bar{x} - n^{1/2}\mu)^\top \Sigma^{-1} (n^{1/2}\bar{x} - n^{1/2}\mu).$$

Recall that $\bar{x} \sim N_p(\mu, \frac{1}{n}\Sigma)$, and that therefore $n^{1/2}\bar{x} \sim N_p(n^{1/2}\mu, \Sigma)$. Applying Proposition
Proposition 7.5 we thus see that

$$\zeta^2 \sim \chi^2_p$$

when H_0 is true.

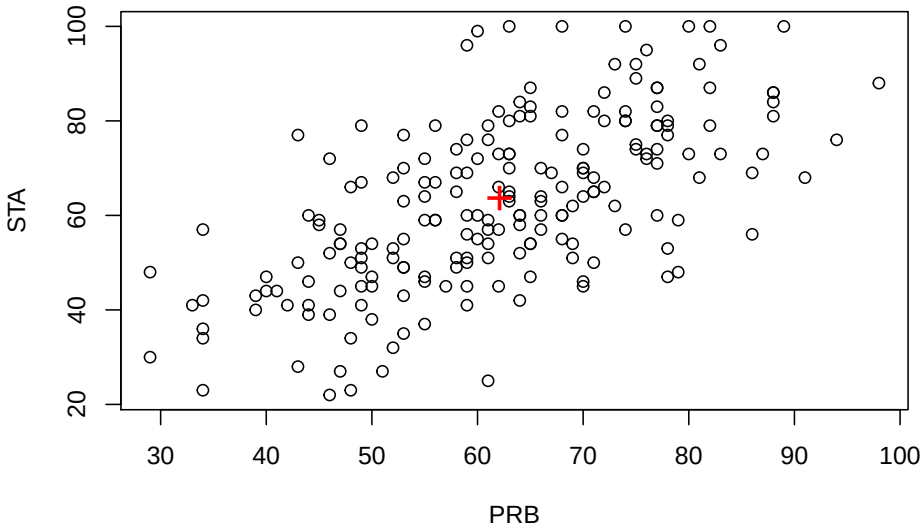
Thus to conduct the hypothesis test, we compute ζ^2 for the observed data, and if the value is large compared to a χ^2_p distribution, we reject H_0.

- The Neyman-Pearson approach is to define a critical region and reject H_0 if $\zeta^2 > \chi^2_{p,\alpha}$, where $\chi^2_{p,\alpha}$ is the upper α quantile of the χ^2_p distribution, i.e., $\mathbb{P}(\chi^2_p > \chi^2_{p,\alpha}) = \alpha$.
- The Fisherian approach is to state the result as a p-value where $p = \mathbb{P}(\chi^2_p > \zeta^2_{\text{obs}})$, and ζ^2_{obs} is the observed value of the statistic ζ^2.

The multivariate equivalent of a confidence interval is a **confidence region** and the $100(1-\alpha)\%$ confidence region for μ is $\{a : \zeta^2 \leq \chi^2_{p,\alpha}\}$. This confidence region will be the interior of an ellipse or ellipsoid.

Example

Consider again the exam marks for first year maths students in the two modules PRB and STA. The scatterplot below shows the module marks for $n = 203$ students on probability (PRB, x_1) and statistics (STA, x_2), with the sample mean vector $\begin{pmatrix} 62.1 \\ 62.7 \end{pmatrix}$ marked on as a red '+'.



The target for the module mean for a large population of students should be exactly 60 for both modules. We now conduct a hypothesis test to see if the lecturers have missed the target and made the exam too difficult. We will test the hypotheses H_0 versus H_1 at the 5% level where

$$H_0 : \mu = \begin{pmatrix} 60 \\ 60 \end{pmatrix} \qquad \text{and} \qquad H_1 : \mu \neq \begin{pmatrix} 60 \\ 60 \end{pmatrix}.$$

Let's assume to begin with that observations $x_1, \ldots, x_{203}$ are a random sample from $N_2(\mu, \Sigma)$ where

$$\Sigma = \begin{pmatrix} 200 & 150 \\ 150 & 300 \end{pmatrix}$$

is assumed known.

The test statistic is

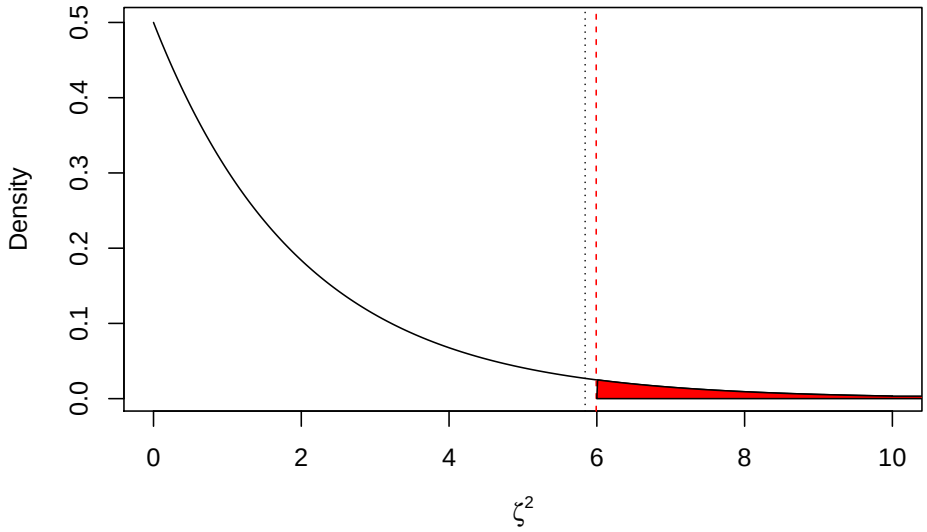
$$\begin{aligned} \zeta^2 &= 203 \begin{pmatrix} 62.1 - 60 \\ 62.7 - 60 \end{pmatrix}^\top \begin{pmatrix} 200 & 150 \\ 150 & 300 \end{pmatrix}^{-1} \begin{pmatrix} 62.1 - 60 \\ 62.7 - 60 \end{pmatrix} \\ &= 5.84 \end{aligned}$$

Under H_0, $\zeta^2 \sim \chi^2_2$. and so the critical value is $\chi^2_{2,0.05} = 5.991$.

```
qchisq(0.95, 2)
```

```
## [1] 5.991465
```

The plot below shows the density of a χ^2_2 random variable. The vertical red line shows the critical value, the vertical black line the observed value, and the shaded region shows the critical region. As $\zeta^2 < \chi^2_{p,0.05}$ we can see that we do not reject the null hypothesis at the 5% level.



The p-value is

```
1-pchisq(zeta2, 2)
```

```
##             [,1]
## [1,] 0.05389049
```

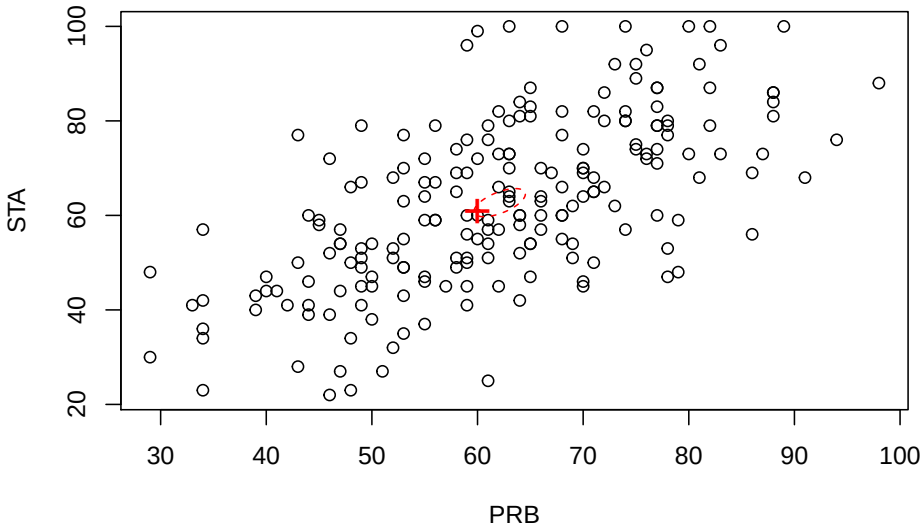
which is area of the red shaded region.

Note that if we had conducted separate univariate hypothesis tests of $H_0 : \mu_1 = 60$ and $H_0 : \mu_2 = 60$ then the test statistics would have been:

$$\begin{aligned} z_1 &= \frac{\bar{x}_1 - \mu_1}{\sqrt{\sigma_1^2/n}} = \frac{62-60}{\sqrt{200/203}} = 2.11 \\ z_2 &= \frac{\bar{x}_2 - \mu_2}{\sqrt{\sigma_2^2/n}} = \frac{62-60}{\sqrt{300/203}} = 2.22. \end{aligned}$$

The critical value would have been $Z_{0.025} = 1.960$ and both null hypotheses would have been rejected. Therefore we see that a multivariate hypothesis can be accepted when each of univariate components is rejected and vice-versa.

The 95% confidence region is the interior of an ellipse, centred on $\bar{x}$, with the angle of the major-axis governed by Σ (given by the eigenvectors of Σ). We can see from the plot below that $(60, 60)^\top$, marked with a cross, lies just inside the confidence region.‘



7.4.2 Σ unknown: 1 sample

In the previous section we considered a hypothesis test of

$$H_0 : \mu = a \text{ vs } H_1 : \mu \neq a$$

based on an IID sample from $N_p(\mu, \Sigma)$ when Σ was **known**. In reality, we rarely know Σ, so we **replace it with the sample covariance matrix**, S. Corollary 7.6 tells us that the distribution is then $F_{p,n-p}$ rather than χ^2_p as was the case when Σ was known.

More specifically, we use the test statistic:

$$\gamma^2 = \frac{n-p}{p}(\bar{x} - a)^\top S^{-1}(\bar{x} - a),$$

Corollary 7.6 tellsus that when H_0 is true,

$$\gamma^2 \sim F_{p,n-p}.$$

As before, depending upon our approach we either

- (Neyman-Pearson approach) reject H_0 if $\gamma^2 > F_{p,n-p,\alpha}$, where α is the significance level.
- (Fisherian approach) compute the p-value $p = \mathbb{P}(F_{p,n-p} > \gamma^2_{obs})$.

The $100(1-\alpha)\%$ confidence region for μ is $\{a : \gamma^2 \leq F_{p,n-p,\alpha}\}$, which will again be the interior of an ellipse or ellipsoid, but the confidence region is now determined by S rather than Σ.

Example continued

We return to the example with the module marks for $n = 203$ students on probability (PRB, x_1) and statistics (STA, x_2), but now we assume that Σ is unknown.

The sample mean and sample covariance matrix are

$$\bar{x} = \begin{pmatrix} 62.1 \\ 62.7 \end{pmatrix} \qquad S = \begin{pmatrix} 191 & 155.6 \\ 155.6 & 313.5 \end{pmatrix}$$

We conduct a hypothesis test at the 5% level of:

$$H_0 : \mu = \begin{pmatrix} 60 \\ 60 \end{pmatrix} \qquad \text{vs.} \qquad H_1 : \mu \neq \begin{pmatrix} 60 \\ 60 \end{pmatrix}.$$

The test statistic is

$$\begin{aligned} \gamma^2 &= \frac{203-2}{2} \begin{pmatrix} 62.1-60 \\ 62.7-60 \end{pmatrix}^\top \begin{pmatrix} 191 & 155.6 \\ 155.6 & 313.5 \end{pmatrix}^{-1} \begin{pmatrix} 62.1-60 \\ 62.7-60 \end{pmatrix} \\ &= 2.84. \end{aligned}$$

The critical value is $F_{2,201,0.05}$

```
qf(0.95, 2,201)
```

```
## [1] 3.040828
```

so $\gamma^2 < F_{p,n-p,0.05}$ and we do not reject the null hypothesis at the 5% level.

The p-value is

```
1-pf(gamma2, 2, 201)
```

```
##             [,1]
## [1,] 0.06051421
```

Thankfully, we don't need to do all the computation ourselves whenever we want to do a test, as there are several R packages that will do the work for us:

```
library(ICSNP) # you'll need to install this package the first time
HotellingsT2(X,  mu = mu)
```

```
##
##  Hotelling's one sample T2-test
##
## data:  X
## T.2 = 2.8444, df1 = 2, df2 = 201, p-value = 0.06051
## alternative hypothesis: true location is not equal to c(60,60)
```

Notice again the difference to the two univariate tests

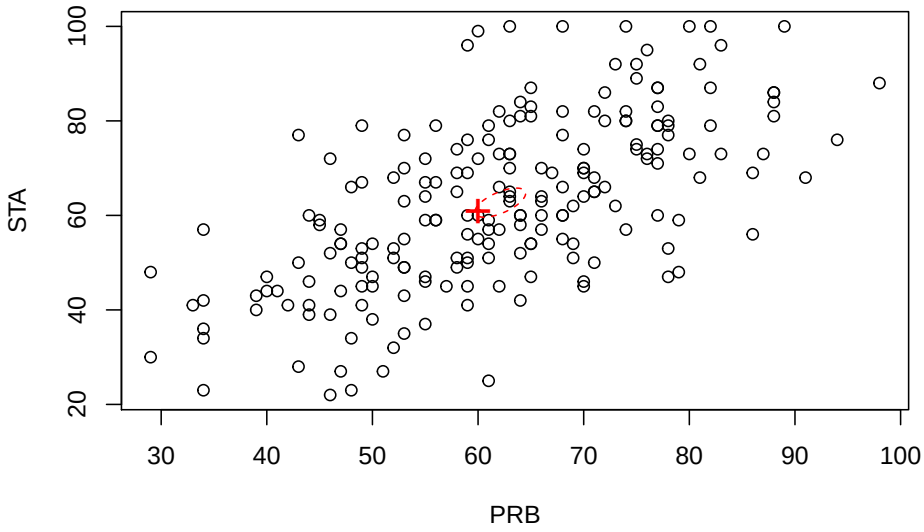
```
t.test(X[,1], mu=60)
```

```
##
##  One Sample t-test
##
## data:  X[, 1]
## t = 2.1581, df = 202, p-value = 0.0321
## alternative hypothesis: true mean is not equal to 60
## 95 percent confidence interval:
##  60.18121 64.01584
## sample estimates:
## mean of x
##  62.09852
```

```
t.test(X[,2], mu=60)
```

```
##
##  One Sample t-test
##
## data:  X[, 2]
## t = 2.1669, df = 202, p-value = 0.03141
## alternative hypothesis: true mean is not equal to 60
## 95 percent confidence interval:
##  60.24305 65.15596
## sample estimates:
## mean of x
##  62.69951
```

The 95% confidence region is the interior of an ellipse, centred on $\bar{x}$, with the angle of the major-axis governed by S. The confidence region is very slightly larger than when Σ was known.



7.4.3 Σ unknown: 2 samples

Suppose now we have data from two populations $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$, and that we wish to test the difference between the two population means. As with the univariate case, there are two cases to consider:

Paired case

If $m = n$ and there exists some experimental link between x_i and y_i then we can look at the differences $z_i = y_i - x_i$ for $i = 1, \ldots, n$. For example, x_i and y_i could be vectors of pre-treatment and post-treatment measurements, respectively, of the same variables. The crucial assumption is that the differences z_i are IID $N_p(\mu, \Sigma)$. To examine the null hypothesis of no difference between the means we would test

$$H_0 : \mu = \mathbf{0}_p \text{ vs } H_1 : \mu \neq \mathbf{0}_p.$$

We then base our inference on $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i = \bar{y} - \bar{x}$, and proceed exactly as in the 1 sample case, using the test in Secion 7.4.1 if Σ is known, or else the test in Section 7.4.2 if with $S = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})(z_i - \bar{z})^\top$.

Unpaired case

The unpaired case is where x_i and y_i are independent and not connected to each other. For example, in a clinical trial we may have two separate groups of patients, where one group receives a placebo and the other group receives an active treatment. Let $x_1, \ldots, x_n$ be an IID sample from $N_p(\mu_1, \Sigma)$ and let $y_1, \ldots, y_m$ be an IID sample from $N_p(\mu_2, \Sigma)$. In this case, we can base our inference on the following result.

Proposition 7.15. *Suppose*

$$x_1, \ldots, x_n \sim N_p(\mu_1, \Sigma_1)$$
$$y_1, \ldots, y_m \sim N_p(\mu_2, \Sigma_2).$$

Then when $\mu_1 = \mu_2$ and $\Sigma_1 = \Sigma_2$ (i.e. when the two populations have the same distribution),

$$\frac{nm}{n+m}(\bar{y} - \bar{x})^\top S_u^{-1}(\bar{y} - \bar{x}) \sim T^2(p, n+m-2),$$

where

$$S_u = \frac{nS_1 + mS_2}{n+m-2}$$

is the pooled unbiased variance matrix estimator and S_j is the sample covariance matrix for group j.

Proof. We know that $\bar{x} \sim N_p\left(\mu_1, n^{-1}\Sigma_1\right)$ and $\bar{y} \sim N_p\left(\mu_2, m^{-1}\Sigma_2\right)$, and $\bar{x}$ and $\bar{y}$ are independent, so

$$\bar{y} - \bar{x} \sim N_p\left(\mu_2 - \mu_1, \frac{1}{n}\Sigma_1 + \frac{1}{m}\Sigma_2\right).$$

If $\mu_1 = \mu_2$ and $\Sigma_1 = \Sigma_2 = \Sigma$, then $\bar{y} - \bar{x} \sim N_p\left(0_p, \left(\frac{1}{n} + \frac{1}{m}\right)\Sigma\right)$ and

$$z = \left(\frac{1}{n} + \frac{1}{m}\right)^{-1/2}(\bar{y} - \bar{x}) \sim N_p(0_p, \Sigma).$$

From Proposition 7.12 we know that $nS_1 \sim W_p(\Sigma_1, n-1)$ and $mS_2 \sim W_p(\Sigma_2, m-1)$. Therefore when $\Sigma_1 = \Sigma_2 = \Sigma$,

$$\begin{aligned} M = (n+m-2)S_u &= (n+m-2)\frac{nS_1 + mS_2}{n+m-2} \\ &= nS_1 + mS_2 \sim W_p(\Sigma, n+m-2) \end{aligned}$$

by Proposition 7.11, using the fact that S_1 and S_2 are independent.

Now z is independent of M, since $\bar{x}$ and $\bar{y}$ are independent of S_1 and S_2, respectively, by Proposition 7.6. Therefore, applying Proposition 7.13 with $x = z$ and $M = (n+m-2)S_u$, we have

$$\begin{aligned} (n+m-2)z^\top((n+m-2)S_u)^{-1}z = z^\top S_u^{-1} z \\ \sim T^2(p, n+m-2) \end{aligned}$$

and

$$\begin{aligned} z^\top S_u^{-1} z &= \left(\frac{1}{n} + \frac{1}{m}\right)^{-1/2}(\bar{y} - \bar{x})^\top S_u^{-1}\left(\frac{1}{n} + \frac{1}{m}\right)^{-1/2}(\bar{y} - \bar{x}) \\ &= \left(\frac{1}{n} + \frac{1}{m}\right)^{-1}(\bar{y} - \bar{x})^\top S_u^{-1}(\bar{y} - \bar{x}). \end{aligned}$$

Finally,

$$\left(\frac{1}{n}+\frac{1}{m}\right)^{-1}=\left(\frac{m}{nm}+\frac{n}{nm}\right)^{-1}=\left(\frac{n+m}{nm}\right)^{-1}=\frac{nm}{n+m},$$

so Proposition 7.15 is proved.

□

As in the one sample case, we can convert Hotelling's two-sample T^2 statistic to the F distribution using Proposition 7.14.

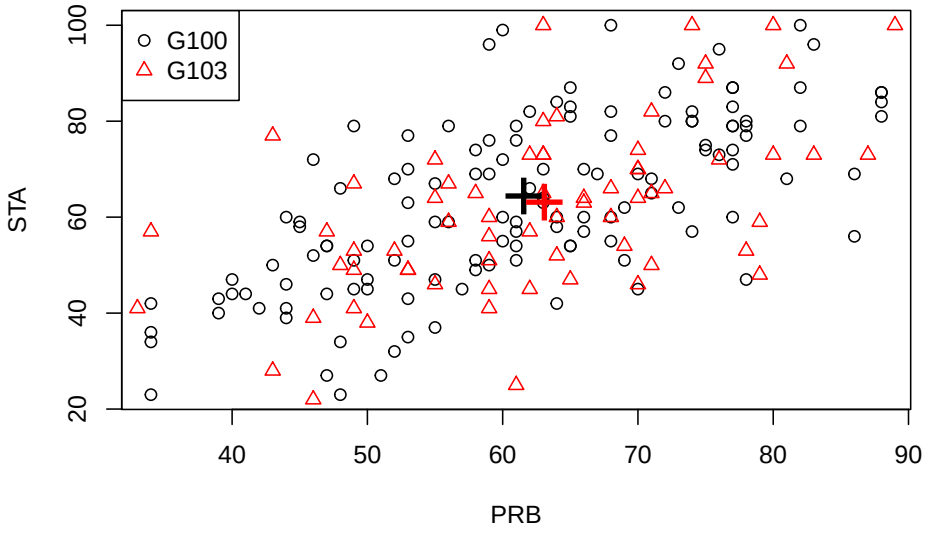
Corollary 7.7. *Using the notation of Proposition 7.15, it follows that*

$$\delta^2=\frac{(n+m-p-1)}{(n+m-2)p}\frac{nm}{(n+m)}(\bar{y}-\bar{x})^\top S_u^{-1}(\bar{y}-\bar{x})\sim F_{p,n+m-p-1}.$$

Proof. Simply apply Proposition 7.14 to the statistic in Proposition 7.15 (replace n with $n+m-2$). □

Example continued

There are two different maths undergraduate programmes at the University of Nottingham: a 3-year (G100) and a 4-year (G103) programme. For the exam marks example, is there a significant difference between students registered on the two different programmes? Let $x_1,\ldots,x_{131}$ be the exam marks of the G100 students and let $y_1,\ldots,y_{72}$ be the exam marks of the G103 students. The data is shown below, with the sample means marked as large '+' signs.



Let μ_1 and μ_2 be the population means for G100 and G103 respectively. Our hypotheses are

$$H_0:\mu_1=\mu_2 \qquad \text{vs.} \qquad H_1:\mu_1\neq\mu_2.$$

We will assume that

$$x_n, \dots, x_{131} \sim N_2(\mu_1, \Sigma_1)$$
$$y_1, \dots, y_m \sim N_2(\mu_2, \Sigma_2).$$

The sample summary statistics are:

$$n = 131 \qquad m = 72$$
$$\bar{x} = \begin{pmatrix} 61.6 \\ 63.2 \end{pmatrix} \qquad \bar{y} = \begin{pmatrix} 63.1 \\ 61.9 \end{pmatrix}$$
$$S_1 = \begin{pmatrix} 180.2 & 158.5 \\ 158.5 & 312.8 \end{pmatrix} \qquad S_2 = \begin{pmatrix} 179.1 & 157.5 \\ 157.5 & 310.8 \end{pmatrix}$$

The assumption $\Sigma = \Sigma_1 = \Sigma_2$ does not look unreasonable given the sample covariance matrices. Using the `HotellingT2` command from the `ICSNP` package, we find

```
library(ICSNP) # you'll need to install this
HotellingsT2(G100, G103)
```

```
##
##  Hotelling's two sample T2-test
##
## data:  G100 and G103
## T.2 = 1.0696, df1 = 2, df2 = 200, p-value = 0.3451
## alternative hypothesis: true location difference is not equal to c(0,0)
```

So the test statistic was computed to be $\delta^2 = 1.06962$ and the p-value is $p = 0.345$.

The critical value for $\alpha = 0.05$ is

$$F_{2,n+m-2-1,\alpha} = F_{2,200,0.05} = 3.041.$$

```
qf(0.95,2, 200)
```

```
## [1] 3.041056
```

Therefore $\delta^2 < F_{p,n+m-p-1}$, so we do not reject the null hypothesis at the 5% level.