Multivariate Statistics

Prof. Richard Wilkinson

Spring 2021

Contents

Introduction **5**

PART I: Prerequisites **7**

1 Statistical Preliminaries **9**
1.1 Notation 9
1.2 Exploratory data analysis (EDA) 13
1.3 Random vectors and matrices 18
1.4 Computer tasks 19
1.5 Exercises 24

2 Review of linear algebra **27**
2.1 Basics 27
2.2 Vector spaces 31
2.3 Inner product spaces 36
2.4 The Centering Matrix 42
2.5 Computer tasks 43
2.6 Exercises 46

3 Matrix decompositions **49**
3.1 Matrix-matrix products 49
3.2 Spectral/eigen decomposition 50
3.3 Singular Value Decomposition (SVD) 53
3.4 SVD optimization results 58
3.5 Low-rank approximation 60
3.6 Computer tasks 65
3.7 Exercises 67

PART II: Dimension reduction methods **71**

4 Principal Component Analysis (PCA) **75**
4.1 PCA: an informal introduction 78
4.2 PCA: a formal description with proofs 91
4.3 An alternative view of PCA 103

4.4 Computer tasks 114
4.5 Exercises 115

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 **??**), whose purpose is to describe the association between two sets of variables; and Multidimensional Scaling (Chapter **??**), 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

$$\mathbf{X} = \begin{bmatrix} - & \mathbf{x}_1^\top & - \\ - & \mathbf{x}_2^\top & - \\ - & .. & - \\ - & \mathbf{x}_n^\top & - \end{bmatrix}$$

The vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ are the observation vectors for each of the n subjects.

- The n *rows* of $\mathbf{X}$ are $\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top$ - 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

$$x_1^\top = (1, 2, 4, 3)$$

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.

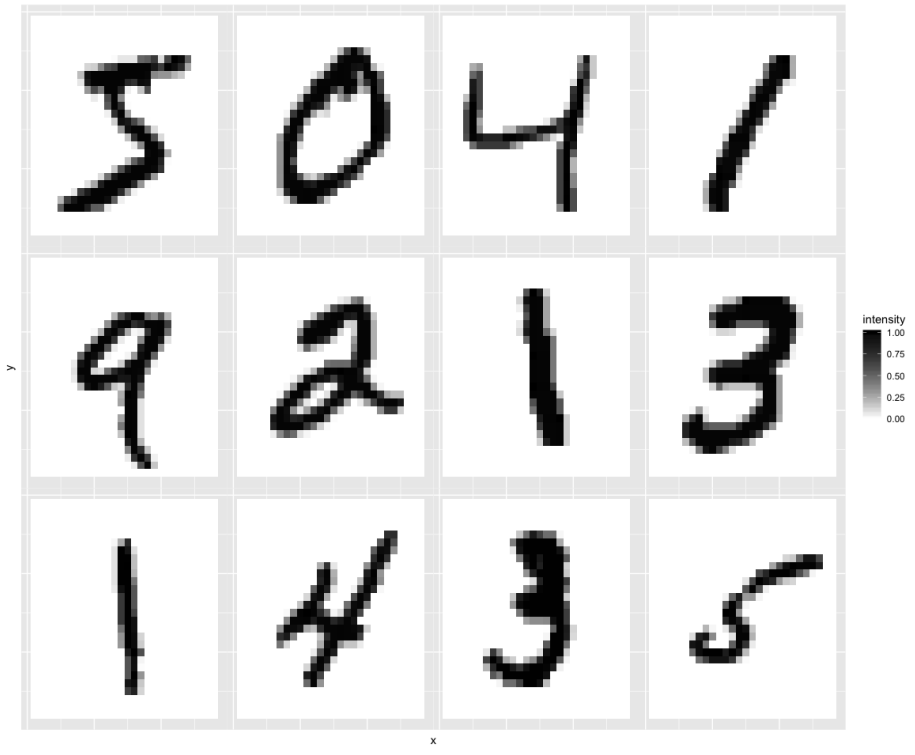
Student	P	S
1	x_{11}	x_{12}
2	x_{21}	x_{22}
$\vdots$	$\vdots$	$\vdots$
n	x_{n1}	x_{n2}

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 $n = 150$.

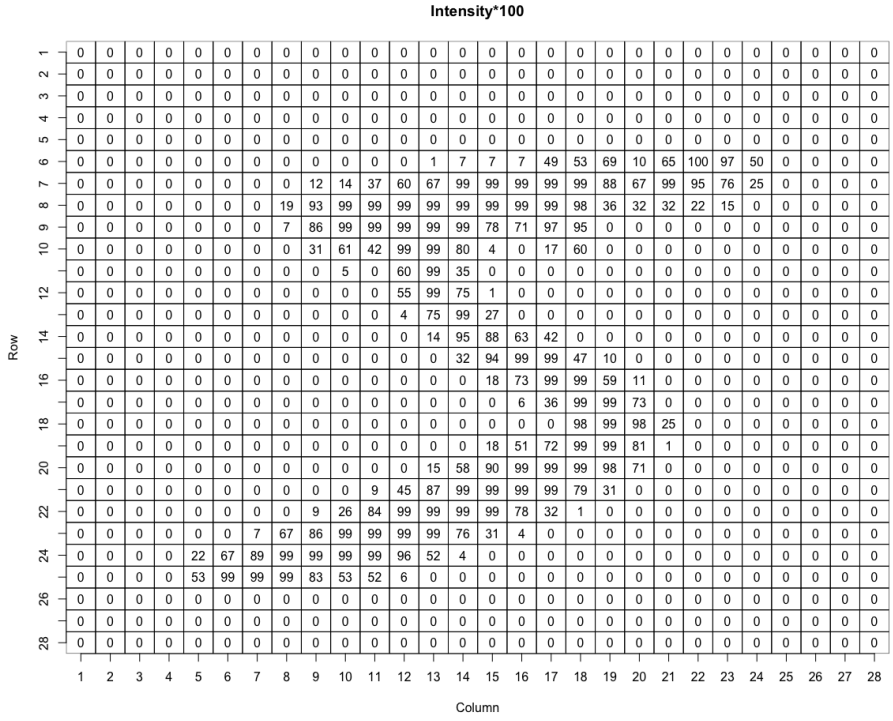
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica

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:



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

1.2 Exploratory data analysis (EDA)

A picture is worth a thousand words

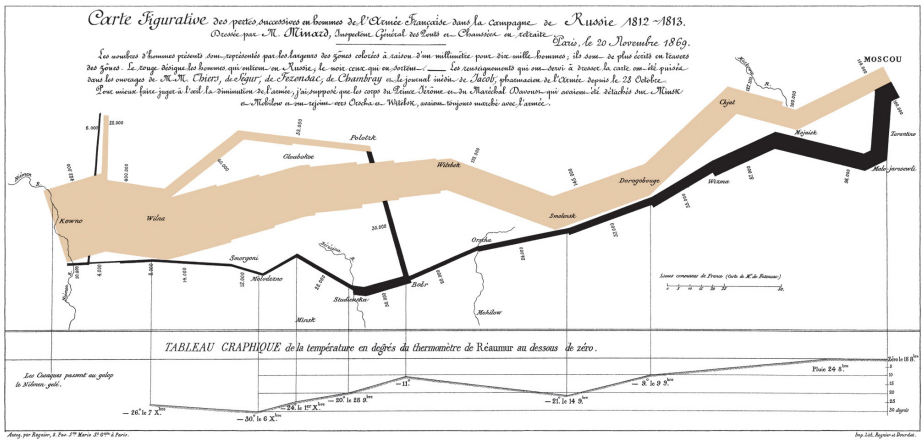


Figure 1.1: Charles Joseph Minard's famous map of Napoleon's 1812 invasion of Russian. It displays six types of data in two dimensions.

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.

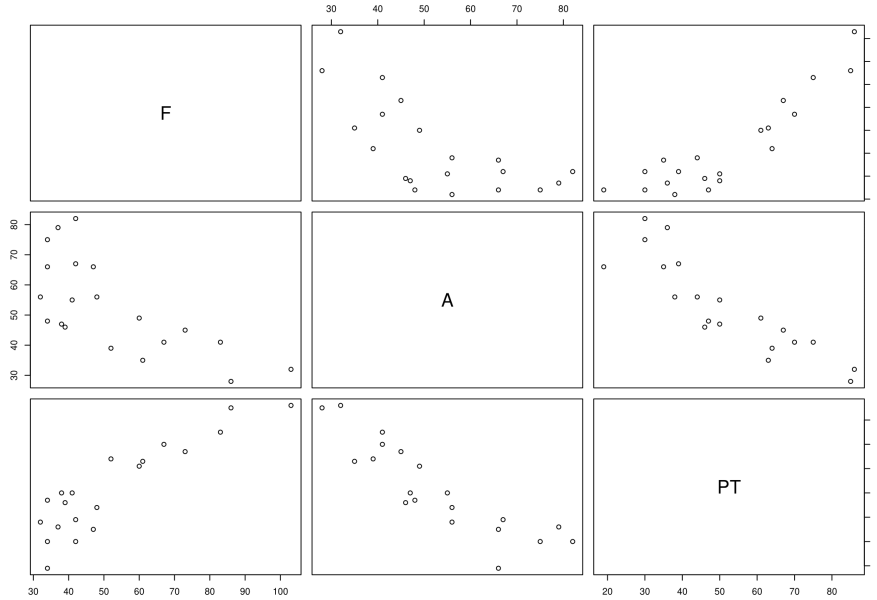


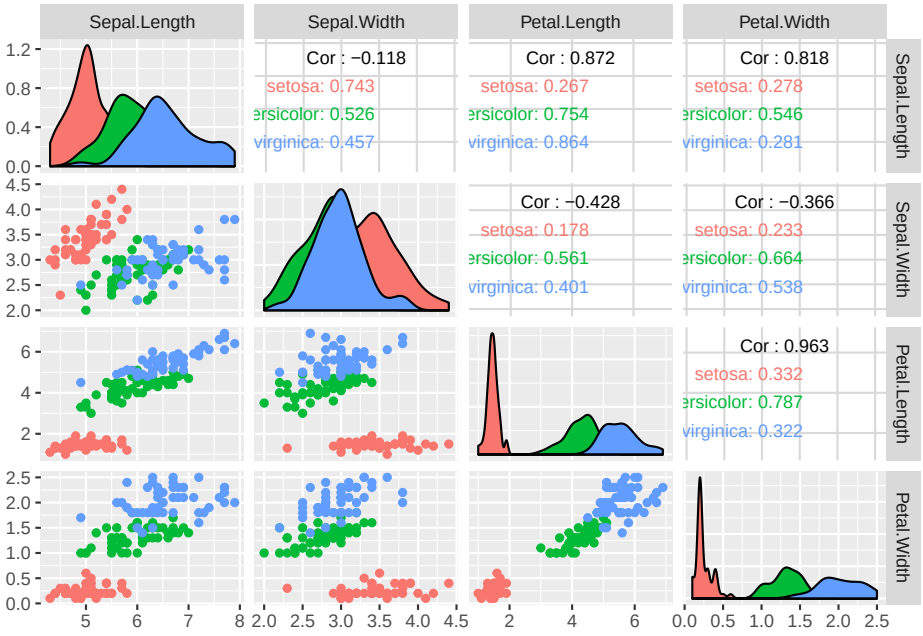
Figure 1.2: Scatter plots of goals for (F), goals against (A) and points (PT) for a recent Premier League Season

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

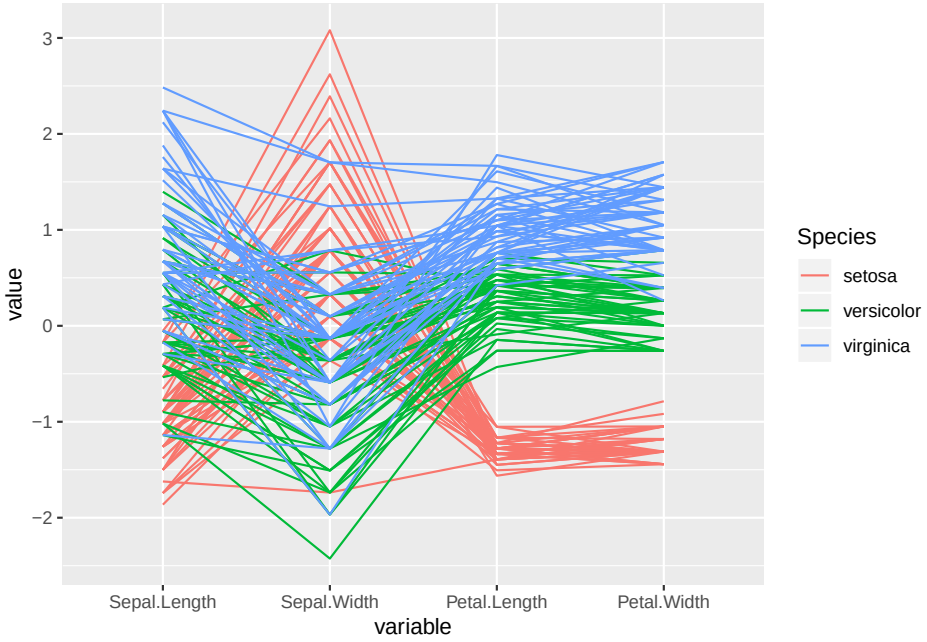
```
data(iris)
library(ggplot2)
library(GGally)
# pairs(iris) # - try the pairs command for comparison
ggpairs(iris, columns=1:4,  mapping=ggplot2::aes(colour = Species),
        upper = list(continuous = wrap("cor", size = 3)))  # fix the font size
```



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, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, the **sample mean** and **sample covariance matrix** are

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

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

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

Note that

- $\bar{\mathbf{x}} \in \mathbb{R}^p$. The jth entry in $\bar{\mathbf{x}}$ is simply the (univariate) sample mean of the jth variable.
- $\mathbf{S} \in \mathbb{R}^{p \times p}$. Note that the ij^{th} entry of $\mathbf{S}$ 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.

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

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

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

$$\mathbf{R} = \mathbf{D}^{-1}\mathbf{S}\mathbf{D}^{-1}$$

- $\mathbf{R}$ is symmetric
- the diagonal entries of $\mathbf{R}$ 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 $\mathbf{x}_i$'s then $\mathbf{S}$ will change but $\mathbf{R}$ will not.

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

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

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 $\mathbf{x}$ is

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

The **population covariance matrix** of $\mathbf{x}$ is

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

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

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

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

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

The following properties follow:

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

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

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

1.3.1 Estimators

The population mean vector $\boldsymbol{\mu}$ and population covariance matrix $\boldsymbol{\Sigma}$ will usually be unknown. We can use data to **estimate** these quantities.

- The sample mean $\bar{\mathbf{x}}$ is often used as an estimator of $\boldsymbol{\mu}$.

- The sample covariance matrix $\mathbf{S}$ is often used as an estimator of $\mathbf{\Sigma}$.

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

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

and covariance matrix

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

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 $\mathbf{\Sigma}$ are equivalent, i.e. show that
$$\mathbf{\Sigma} = \mathbb{E}[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top] = \mathbb{E}[\mathbf{x}\mathbf{x}^\top] - \boldsymbol{\mu}\boldsymbol{\mu}^\top.$$
2. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a p-dimensional sample with mean $\bar{\mathbf{x}}$ and sample covariance matrix $\mathbf{S}$. Consider the transformation $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{c}$ where $\mathbf{A}$ is a fixed $q \times p$ matrix and $\mathbf{c}$ is a fixed q-dimensional vector. Let $\mathbf{T}$ be the sample covariance matrix of $\mathbf{y}_1, \ldots, \mathbf{y}_n$. Show

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

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

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

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

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

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

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

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

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:

$$\begin{aligned}
\mathbf{A} = \overset{n\times p}{\mathbf{A}} &= \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{np} \end{pmatrix} \\
&= [a_{ij} : i = 1, \dots, n; j = 1, \dots, p] \\
&= (a_{ij}) \\
&= \begin{pmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_n^\top \end{pmatrix}
\end{aligned}$$

where the a_{ij} are the individual entries, and $\mathbf{a}_i^\top = (a_{i1}, a_{i2}, \dots, a_{ip})$ 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 $\mathbf{AB} = \overset{n\times q}{\mathbf{C}} = [c_{ij}]$ where
$$c_{ij} = \sum_{k=1}^{p} a_{ik}b_{kj}, \qquad i = 1, \ldots, n, \qquad j = 1, \ldots, q.$$

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

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 $\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$. 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* $\mathbf{A}$ *(*$n \times m$*) and* $\mathbf{B}$ *(*$m \times n$*),*
$$tr(\mathbf{AB}) = tr(\mathbf{BA}).$$

7. The *determinant* of a square matrix $\overset{n\times n}{\mathbf{A}}$ is defined as
$$\det(\mathbf{A}) = \sum (-1)^{|\tau|} a_{1\tau(1)} \ldots a_{n\tau(n)}$$
where the summation is taken over all permutations τ of $\{1, 2, \ldots, n\}$, 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* $\det(\mathbf{A}) \neq 0$*. If* $\mathbf{A}^{-1}$ *exists then*
$$\det(\mathbf{A}) = \frac{1}{\det(\mathbf{A}^{-1})}$$

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 $\mathbf{A}$ is symmetric if

$$\mathbf{A} = \mathbf{A}^\top.$$

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

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

and is **positive semi-definite** if

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

$\mathbf{A}$ is **idempotent** if $\mathbf{A}^2 = \mathbf{A}$.

2.1.4 Vector Differentiation

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

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

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

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

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

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

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

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,

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

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,

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

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

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

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

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

The nature of a stationary point is determined by the Hessian

If the Hessian is positive (negative) definite at a stationary point $\mathbf{x}$, then the stationary point is a minimum (maximum).

If the Hessian has both positive and negative eigenvalues at $\mathbf{x}$ 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 $\mathbf{v}_1, \mathbf{v}_2 \in V$ and $\alpha_1, \alpha_2 \in \mathbb{R}$, we have

$$\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 \in V$$

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 $\mathbf{Ax}$ 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 $\mathbf{Ax}$ 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 $\mathbf{A}$ is the set of all linear combinations of the columns of $\mathbf{A}$:

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

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 $\mathbf{B}$ has the same column space as $\mathbf{A}$

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

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

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

Theorem 2.1. *The **row rank** 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 $\mathbf{C}$ be an $n \times r$ matrix (where $r = \text{rank}(\mathbf{A})$) with columns chosen to be a set of r linearly independent columns from A. Then we know each column of $\mathbf{A}$ can be written as a linear combination of the columns of $\mathbf{C}$, i.e.

$$\mathbf{A} = \mathbf{CR}.$$

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

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

The **image** of $\mathbf{A}$ is precisely the column space of $\mathbf{A}$:

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

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

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

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

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 $\dim \mathcal{C}(\mathbf{A}) + \dim \mathcal{N}(\mathbf{A}) = p$, or equivalently that
$\mathrm{rank}(\mathbf{A}) + \dim \mathcal{N}(\mathbf{A}) = p$.

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

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

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 $\mathbf{D}$ is the set of vectors for which $\mathbf{Dx} = \mathbf{0}$, 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 $\dim \mathcal{C}(D^\top) = 1$ (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 $\mathbf{A}$ have an inverse?

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

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

$$\mathbf{Ax} = \mathbf{y}$$

have a solution?

- When $\mathbf{y}$ is in the column space of $\mathbf{A}$,

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

Question: When is the answer unique?

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

$$\mathbf{y} = \mathbf{A}\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{A}\mathbf{u} = \mathbf{y} + \mathbf{A}\mathbf{u}$$

  and so $\mathbf{A}\mathbf{u} = \mathbf{0}$, i.e., $\mathbf{u} \in \mathcal{N}(A)$. So there are multiple solutions when the null-space of $\mathbf{A}$ 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:

$$\langle \alpha \mathbf{v}_1 + \beta \mathbf{v}_2, \mathbf{u} \rangle = \alpha \langle \mathbf{v}_1, \mathbf{u} \rangle + \beta \langle \mathbf{v}_2, \mathbf{u} \rangle$$

for all $\mathbf{v}_1, \mathbf{v}_2, \mathbf{u} \in V$ and $\alpha, \beta \in \mathbb{R}$. 2. $\langle \cdot, \cdot \rangle$ is symmetric in its arguments: $\langle \mathbf{v}, \mathbf{u} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle$ for all $\mathbf{u}, \mathbf{v} \in V$ 3. $\langle \cdot, \cdot \rangle$ is positive definite: $\langle \mathbf{v}, \mathbf{v} \rangle \geq 0$ for all $\mathbf{v} \in V$ with equality if and only if $\mathbf{v} = \mathbf{0}$.

An inner product provides a vector space with the concepts of

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

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

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

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

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

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

$$\langle \mathbf{u}, \mathbf{v} \rangle = 0$$

  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

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^\top \mathbf{v}$$

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

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

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

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

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

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

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

then

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

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

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

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

2.3.2 Orthogonal matrices

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

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

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

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

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

If $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_n]$ is an orthogonal matrix, then the columns $\mathbf{q}_1, \dots, \mathbf{q}_n$ are mutually **orthonormal** vectors, i.e.

$$\mathbf{q}_j^\top \mathbf{q}_k = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k. \end{cases}$$

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

Proof. Suppose $n = p$, and think of $\mathbf{Q}$ as a linear map

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

By the rank-nullity theorem,

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

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

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

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

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

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

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

Corollary 2.2. *If $\mathbf{q}_1, \ldots, \mathbf{q}_n$ are mutually orthogonal $n \times 1$ unit vectors then*

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

Proof. Let $\mathbf{Q}$ be the matrix with i^{th} column $\mathbf{q}_i$

$$\mathbf{Q} = \begin{pmatrix} | & & | \\ \mathbf{q}_1 & \cdots & \mathbf{q}_n \\ | & & | \end{pmatrix}.$$

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

$$\mathbf{I}_n = \mathbf{Q}\mathbf{Q}^\top = \begin{pmatrix} | & & | \\ \mathbf{q}_1 & \cdots & \mathbf{q}_n \\ | & & | \end{pmatrix} \begin{pmatrix} - & \mathbf{q}_1^\top & - \\ & \vdots & \\ - & \mathbf{q}_n^\top & - \end{pmatrix} = \sum_{i=1}^{n} \mathbf{q}_i \mathbf{q}_i^\top$$

as required.

□

2.3.3 Projections

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

$$\mathbf{P}^2 = \mathbf{P}$$

i.e., if it is idempotent.

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

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

Proof. Let $\mathbf{w} \in W$. Then

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

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

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

□

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

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

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

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 $\mathbf{w} \in W$ onto U is the unique element $\mathbf{u} \in U$ that minimizes

$$||\mathbf{w} - \mathbf{u}||.$$

In other words, the orthogonal projection of $\mathbf{w}$ onto U is the *best possible approximation* of $\mathbf{w}$ in U.

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

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

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

Proposition 2.5. *If $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ is a basis for U, then the orthogonal projection matrix (i.e., the matrix that projects $\mathbf{w} \in W$ onto U) is*

$$\mathbf{P}_U = \mathbf{A}(\mathbf{A}^{\top}\mathbf{A})^{-1}\mathbf{A}^{\top}$$

where $\mathbf{A} = [\mathbf{u}_1 \ \ldots \ \mathbf{u}_k]$ is the matrix with columns given by the basis vectors.

Proof. We need to find $\mathbf{u} = \sum \lambda_i \mathbf{u}_i = \mathbf{A}\boldsymbol{\lambda}$ that minimizes $||\mathbf{w} - \mathbf{u}||$.

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

Differentiating with respect to $\boldsymbol{\lambda}$ and setting equal to zero gives

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

and hence

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

The orthogonal projection of $\mathbf{w}$ is hence

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

and the projection matrix is

$$\mathbf{P}_U = \mathbf{A}(\mathbf{A}^{\top}\mathbf{A})^{-1}\mathbf{A}^{\top}.$$

□

Notes:

1. If $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ is an orthonormal basis for U then $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$ and $\mathbf{P}_U = \mathbf{A}\mathbf{A}^\top$. We can then write

$$\mathbf{P}_U\mathbf{w} = \sum_i (\mathbf{u}_i^\top \mathbf{w})\mathbf{u}_i$$

and

$$\mathbf{P}_U = \sum_{i=1}^{k} \mathbf{u}_i\mathbf{u}_i^\top.$$

Note that if $U = W$ (so that $\mathbf{P}_U$ is a projection from W onto W, i.e., the identity), then $\mathbf{A}$ is a square matrix ($n \times n$) and thus $\mathbf{A}^\top\mathbf{A} = \mathbf{I}_n \implies \mathbf{A}\mathbf{A}^\top$ and thus $\mathbf{P}_U = \mathbf{I}_n$ as required. The coordinates (with respect to the orthonormal basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$) of a point $\mathbf{w}$ projected onto U are $\mathbf{A}^\top\mathbf{w}$.

2. $\mathbf{P}_U^2 = \mathbf{P}_U$, so $\mathbf{P}_U$ is a projection matrix in the sense of definition 2.11.

3. $\mathbf{P}_U$ is symmetric ($\mathbf{P}_U^\top = \mathbf{P}_U$). 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 $\mathbf{u} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. The projection of $\mathbf{v} \in \mathbb{R}^2$ onto $\mathbf{u}$ is given by $(\mathbf{v}^\top\mathbf{u})\mathbf{u}$. So for example, if $\mathbf{v} = (2,\ 1)^\top$, then its projection onto $\mathbf{u}$ is

$$\mathbf{P}_U\mathbf{v} = \frac{3}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Alternatively, if we treat $\mathbf{u}$ as a basis for U, then the coordinate of $\mathbf{P}_U\mathbf{v}$ 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

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$

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

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

$$||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2$$

We know that $\mathbf{X}\boldsymbol{\beta}$ is in the column space of $\mathbf{X}$, and so we can see that linear regression aims to find the *orthogonal projection* onto $\mathcal{C}(X)$.

$$\mathbf{P}_U\mathbf{y} = \arg \min_{\mathbf{y}' : \mathbf{y}' \in \mathcal{C}(X)} ||\mathbf{y} - \mathbf{y}'||_2.$$

By Proposition 2.5 this is

$$\mathbf{P}_U\mathbf{y} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \hat{\mathbf{y}}$$

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

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

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

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

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 $\mathbf{H}$ in the exercises:

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

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

   and so

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

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

   $$\mathbf{X} = \begin{bmatrix} - & \mathbf{x}_1^\top & - \\ & \vdots & \\ - & \mathbf{x}_n^\top & - \end{bmatrix} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$$

   is an $n \times p$ data matrix containing data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, then

   $$\mathbf{H}\mathbf{X} = \begin{bmatrix} - & (\mathbf{x}_1 - \bar{\mathbf{x}})^\top & - \\ - & (\mathbf{x}_2 - \bar{\mathbf{x}})^\top & - \\ & \vdots & \\ - & (\mathbf{x}_n - \bar{\mathbf{x}})^\top & - \end{bmatrix} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots, \mathbf{x}_n - \bar{\mathbf{x}}]^\top$$

where

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

is the p-dimensional sample mean of $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$. In words, $\mathbf{H}$ has subtracted the column mean from each column of $\mathbf{X}$.

6. With $\mathbf{X}$ as in 5.

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

where $\mathbf{S}$ is the sample covariance matrix.

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

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

or, equivalently,

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

where

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

$\bar{a}_{+j} = \bar{a}_{j+}$, 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 $\mathbf{x}$ using R.

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

2. Consider the `iris` dataset. Let $\mathbf{X}$ 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 $\mathbf{HX}$ are all zero (or close - there will be numerical error). Compute the sample covariance and correlation matrices using $\mathbf{H}$.

```
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 $\mathbf{H}$ in the description of the methods as it makes the mathematics easier to write down.

- Compute the covariance matrix of $\mathbf{X}$ 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
$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 2 & -4 \\ 1 & 1 & 1 \end{pmatrix}$$
   - What is rank($\mathbf{A}$)?
   - Write the product $\mathbf{Ax}$ where $\mathbf{x}^\top = (x_1, x_2, x_3)$ as both the inner product of the rows of $\mathbf{A}$ and $\mathbf{x}$, and as a linear combination of the columns of $\mathbf{A}$ (see section 2.2.2)
   - Describe the column space of $\mathbf{A}$. What is its dimension?
   - Find a vector in the kernel of $\mathbf{A}$.
   - Describe the kernel of $\mathbf{A}$ as a vector space and give a basis for the space.
   - Is $\mathbf{A}$ invertible? What is det($\mathbf{A}$)?
3. Let's consider the inner product space $\mathbb{R}^3$ with the Euclidean inner product
$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y}$$
   Let
$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} -1 \\ 0 \\ -2 \end{pmatrix}. \quad \mathbf{x}_3 = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$$
   - What is the angle between $\mathbf{x}_1$ and $\mathbf{x}_2$? Which pairs of vectors are orthogonal to each other?
   - What is the norm associated with this inner-product space, and compute the norm of $\mathbf{x}_1, \ldots, \mathbf{x}_3$. What is the geometric interpretation of the norm?
4. Prove the following statements:
   - The determinant of an orthogonal matrix must be either 1 or -1.
   - If $\mathbf{A}$ and $\mathbf{B}$ are orthogonal matrices, then $\mathbf{AB}$ must also be orthogonal.

- Let $\mathbf{A}$ be an $n \times n$ matrix of the form

$$\mathbf{A} = \mathbf{Q}\mathbf{B}\mathbf{Q}^\top.$$

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

5. Consider the matrix

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

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

6. 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

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

onto U? Show that this vector does minimize $||\mathbf{v} - \mathbf{u}||$ for $\mathbf{u} \in U$.
- 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.

7. 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

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

to be the $n \times n$ centering matrix (see 2.4).
Let $\mathbf{x} = (x_1, \ldots, x_n)^\top$ denote a vector and let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ 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 $\bar{\mathbf{x}} = n^{-1}\sum_{i=1}^n \mathbf{x}_i$.

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

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

iii. Show that

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

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

iv. Show that

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

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

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

$$\mathbf{HX} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots, \mathbf{x}_n - \bar{\mathbf{x}}]^\top.$$

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

vi. Using $\mathbf{S}$ to denote the sample covariance matrix, show that

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

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

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

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: $\mathbf{Ax}$ is rows of $\mathbf{A}$ times $\mathbf{x}$; or as a linear combination of the columns of $\mathbf{A}$. We can similarly think about matrix-matrix products in two ways.

The usual way to think about the matrix product $\mathbf{AB}$ is as the rows of $\mathbf{A}$ times the columns of $\mathbf{B}$:

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

A better way (for this module) to think of $\mathbf{AB}$ is as the columns of $\mathbf{A}$ times the rows of $\mathbf{B}$. If we let $\mathbf{a}_i$ denote the columns of $\mathbf{A}$, and $\mathbf{b}_i^*$ the rows of $\mathbf{B}$ then

$$\begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} - & \mathbf{b}_1^* & - \\ - & \mathbf{b}_2^* & - \\ - & \mathbf{b}_3^* & - \end{bmatrix} = \sum_{i=1}^{3} \mathbf{a}_i \mathbf{b}_i^*$$

i.e., $\mathbf{AB}$ is a sum of the columns of $\mathbf{A}$ times the rows of $\mathbf{B}$.

Note that if $\mathbf{a}$ is a vector of length n and $\mathbf{b}$ is a vector of length p then $\mathbf{ab}^\top$ 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 $\mathbf{ab}^\top$ is a rank-1 matrix as its columns are all multiples of $\mathbf{a}$, or in other words, its column space is just multiples of $\mathbf{a}$.

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

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

By thinking of matrix-matrix multiplication in this way

$$\mathbf{AB} = \sum_{i=1}^{k} \mathbf{a}_i \mathbf{b}_i^*$$

(where k is the number of columns of $\mathbf{A}$ and the number of rows of $\mathbf{B}$) 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 $\mathbf{A}$. We say that vector $\mathbf{x} \in \mathbb{R}^n$ is an **eigenvector** corresponding to **eigenvalue** λ of $\mathbf{A}$ if

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

To find the eigenvalues of a matrix, we note that if λ is an eigenvalue, then $(\mathbf{A} - \lambda\mathbf{I}_n)\mathbf{x} = \mathbf{0}$, i.e., the kernel of $\mathbf{A} - \lambda\mathbf{I}_n$ has dimension at least 1, so $\mathbf{A} - \lambda\mathbf{I}_n$ 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 $\mathbf{A}$ we find the n roots $\lambda_1, \ldots, \lambda_n$ 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 $\mathbf{A}$ is an $n \times n$ symmetric matrix then its determinant is the product of its eigenvalues, i.e. det($\mathbf{A}$) $= \lambda_1 \ldots \lambda_n$.*

Thus,

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

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 $n \times n$ symmetric matrix $\mathbf{A}$ can be written as*

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

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

Because $\mathbf{\Lambda}$ is a diagonal matrix, we sometimes refer to the spectral decomposition as **diagonalizing** the matrix $\mathbf{A}$ as $\mathbf{Q}^\top\mathbf{A}\mathbf{Q} = \mathbf{\Lambda}$ 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 $\mathbf{A}$ 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 $\mathbf{A}$, 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 $\mathbf{q}_i$. As the $\mathbf{q}_i$ are orthogonal, the columns spaces $\mathcal{C}(\mathbf{q}_i\mathbf{q}_i^\top)$ are orthogonal, and their union is a vector space of dimension r. Hence the rank of $\mathbf{A}$ is r. □

Lemma 3.1. *Let* $\mathbf{A}$ *be an* $n \times n$ *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 $\mathbf{A}$ is positive definite, and if $\mathbf{x}$ is a unit-eigenvalue of $\mathbf{A}$ corresponding to λ_n, then

$$0 \leq \mathbf{x}^\top \mathbf{A}\mathbf{x} = \lambda_n \mathbf{x}^\top \mathbf{x} = \lambda_n.$$

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

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \mathbf{y}^\top \mathbf{\Lambda}\mathbf{y} = \sum_{i=1}^{n} \lambda_i y_i^2 > 0$$

and thus $\mathbf{A}$ is positive definite. □

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

$$\mathbf{a}^\top \mathbf{\Sigma}\mathbf{a} \geq 0 \; \forall \; \mathbf{a}.$$

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

Proposition 3.4. *The eigenvalues of a projection matrix* $\mathbf{P}$ *are all* 0 *or* 1.

3.2.3 Matrix square roots

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

$$\mathbf{A}^p = \mathbf{Q}\mathbf{\Lambda}^p \mathbf{Q}^\top.$$

If in addition $\mathbf{A}$ is positive definite (rather than just semi-definite), then

$$\mathbf{A}^{-1} = \mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^\top$$

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

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

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

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

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

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

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

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

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

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

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 $\mathbf{A}$ 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 $\mathbf{u}$ and $\mathbf{v}$ (respectively) if

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

If $\mathbf{A}$ if a symmetric matrix then $\mathbf{u} = \mathbf{v}$ 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 $\mathbf{U}$, a diagonal matrix of singular values $\mathbf{\Sigma}$, and a matrix of right singular vectors $\mathbf{V}$.

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top.$$

Proposition 3.5. ***(Singular value decomposition)***. *Let* $\mathbf{A}$ *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* $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_r]$*, a* $p \times r$ *matrix* $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_r]$*, and a* $r \times r$ *diagonal matrix* $\mathbf{\Sigma} = diag\{\sigma_1, \ldots, \sigma_r\}$ *such that*

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

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

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

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

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

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

The columns of $\mathbf{U}$ and $\mathbf{V}$ 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 $\mathbf{U}$ and $\mathbf{V}$, and truncating $\mathbf{\Sigma}$ to a $r \times r$ matrix with non-zero diagonal elements.

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

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

$$\mathbf{A}^\top\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$$

where $\mathbf{V}$ is a $p \times p$ orthogonal matrix containing the orthonormal eigenvectors of $\mathbf{A}^\top\mathbf{A}$, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_r)$ 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 $\mathbf{u}_i = \frac{1}{\sigma_i}\mathbf{A}\mathbf{v}_i$. Then the vectors $\mathbf{u}_i$ are orthonormal:

$$\begin{aligned}
\mathbf{u}_i^\top \mathbf{u}_j &= \frac{1}{\sigma_i \sigma_j} \mathbf{v}_i^\top \mathbf{A}^\top \mathbf{A} \mathbf{v}_j \\
&= \frac{\sigma_j^2}{\sigma_i \sigma_j} \mathbf{v}_i^\top \mathbf{v}_j \quad \text{as } \mathbf{v}_j \text{ is an eigenvector of } \mathbf{A}^\top \mathbf{A} \\
&= \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad \text{as the } \mathbf{v}_i \text{ are orthonormal vectors.}
\end{aligned}$$

In addition

$$\mathbf{A}^\top \mathbf{u}_i = \frac{1}{\sigma_i} \mathbf{A}^\top \mathbf{A} \mathbf{v}_i = \frac{\sigma_i^2}{\sigma_i} \mathbf{v}_i = \sigma_i \mathbf{v}_i$$

and so $\mathbf{u}_i$ and $\mathbf{v}_i$ are left and right singular vectors.

Let $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \ldots \ \mathbf{u}_r \ \ldots \ \mathbf{u}_n]$, where $\mathbf{u}_{r+1}, \ldots, \mathbf{u}_n$ are chosen to complete the orthonormal basis for $\mathbb{R}^n$ given $\mathbf{u}_1, \ldots, \mathbf{u}_r$, and let Σ be the $n \times p$ diagonal matrix in Equation (3.1).

Then we have shown that

$$\mathbf{U} = \mathbf{A}\mathbf{V}\mathbf{\Sigma}^{-1}$$

Thus

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

□

Note that by construction we've shown that $\mathbf{A}^\top \mathbf{A}$ has eigenvalues σ_i^2 with corresponding eigenvectors $\mathbf{v}_i$. We also can also show that $\mathbf{A}\mathbf{A}^\top$ has eigenvalues σ_i^2, but with corresponding eigenvectors $\mathbf{u}_i$.

$$\mathbf{A}\mathbf{A}^\top \mathbf{u}_i = \sigma_i \mathbf{A}\mathbf{v}_i = \sigma_i^2 \mathbf{u}_i$$

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

Notes:

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

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top.$$

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

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

$$\mathbf{U}^\top \mathbf{A}\mathbf{V} = \mathbf{\Sigma}.$$

3. The SVD reveals a great deal about a matrix. Firstly, the rank of $\mathbf{A}$ is the number of non-zero singular values. The left singular vectors $\mathbf{u}_1, \ldots, \mathbf{u}_r$ are an orthonormal basis for the columns space of $\mathbf{A}$, $\mathcal{C}(\mathbf{A})$, and the right singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ are an orthonormal basis for $\mathcal{C}(\mathbf{A}^\top)$, the row space of $\mathbf{A}$. The vectors $\mathbf{v}_{r+1}, \ldots, \mathbf{v}_p$ from the non-compact SVD are a basis for the kernel of $\mathbf{A}$ (sometimes called the null space $\mathcal{N}(\mathbf{A})$), and $\mathbf{u}_{r+1}, \ldots, \mathbf{u}_n$ are a basis for $\mathcal{N}(\mathbf{A}^\top)$.

4. The SVD has many uses in mathematics. One is as a generalized inverse of a matrix. If $\mathbf{A}$ is $n \times p$ with $n \neq p$, or if it is square but not of full rank, then $\mathbf{A}$ cannot have an inverse. However, we say $\mathbf{A}^+$ is a generalized inverse if $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$. One such generalized inverse can be obtained from the SVD by $\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top$ - 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 $\mathbf{A} = \mathbf{x}\mathbf{y}^\top$. We can see this is a rank-1 matrix, so it only has one non-zero singular value which is $\sigma_1 = ||\mathbf{x}||.||\mathbf{y}||$. Its SVD is given by

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

Example 3.3. Let

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

Let's try to find the SVD of $\mathbf{A}$.

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

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

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

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

The columns of $\mathbf{U}$ are the *unit* eigenvectors of $\mathbf{A}\mathbf{A}^\top$ which we can find by solving

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

And so, remembering that the eigenvectors used to form $\mathbf{V}$ need to be *unit* vectors, we can see that

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

Finally, to compute $\mathbf{V}$ recall that $\sigma_i \mathbf{v}_i = \mathbf{A}^\top \mathbf{u}_i$ and so

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

This completes the calculation, and we can see that we can express $\mathbf{A}$ as

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

or as the sum of rank-1 matrices:

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

This is the compact form of the SVD. To find the non-compact form we need $\mathbf{V}$ 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 $\mathbf{v}_3 = \frac{1}{\sqrt{17}}(2 \ -2 \ -3)$ giving the non-compact SVD for $\mathbf{A}$.

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

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

$$\mathbb{V}\,\mathrm{ar}(\mathbf{a}^\top \mathbf{x}) = \mathbf{a}^\top \mathbf{\Sigma} \mathbf{a}$$

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

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

Given that $\mathbf{\Sigma}$ is symmetric, we can write it as

$$\mathbf{\Sigma} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^\top$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{\Sigma}$, and $\mathbf{V}$ is an orthogonal matrix of eigenvectors. If we let $\mathbf{b} = \mathbf{V}^\top \mathbf{a}$ then

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

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

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

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

$$\begin{aligned} \mathbf{V}^\top \mathbf{a} &= \mathbf{b} \\ \mathbf{V}\mathbf{V}^\top \mathbf{a} &= \mathbf{V}\mathbf{b} \\ \mathbf{a} &= \mathbf{v}_1 \end{aligned}$$

so we can see that the maximum is obtained when $\mathbf{a} = \mathbf{v}_1$, the eigenvector of $\boldsymbol{\Sigma}$ corresponding to the largest eigenvalue λ_1.

Similarly, the minimum is λ_n, which obtained by setting $\mathbf{b} = (\ 0\ 0 \ldots\ 0\ 1)^\top$ which corresponds to $\mathbf{a} = \mathbf{v}_n$.

Proposition 3.7. *For any symmetric $n \times n$ matrix $\boldsymbol{\Sigma}$,*

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

where the maximum occurs at $\mathbf{a} = \pm \mathbf{v}_1$, and

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

where the minimum occurs at $\mathbf{a} = \pm \mathbf{v}_n$, where $\lambda_i, \mathbf{v}_i$ are the ordered eigenpairs of $\boldsymbol{\Sigma}$.

Note that

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

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

$$\max_{\mathbf{a}} \frac{\mathbf{a}^\top \boldsymbol{\Sigma} \mathbf{a}}{\mathbf{a}^\top \mathbf{a}} \quad \text{and} \quad \min_{\mathbf{a}} \frac{\mathbf{a}^\top \boldsymbol{\Sigma} \mathbf{a}}{\mathbf{a}^\top \mathbf{a}}.$$

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

Proposition 3.8. *For any matrix $\mathbf{A}$*

$$\max_{\mathbf{x}:||\mathbf{x}||_2=1} ||\mathbf{A}\mathbf{x}||_2 = \max_{\mathbf{x}} \frac{||\mathbf{A}\mathbf{x}||_2}{||\mathbf{x}||_2} = \sigma_1$$

the first singular value of $\mathbf{A}$, with the maximum achieved at $\mathbf{x} = \mathbf{v}_1$ (the first right singular vector).

Proof. This is follows from 3.7 as

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

□

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

Proposition 3.9. *For any matrix* $\mathbf{A}$*, we have*

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

with the maximum obtained at $\mathbf{a} = \mathbf{u}_1$ *and* $\mathbf{b} = \mathbf{v}_1$*, the first left and right singular vectors of* $\mathbf{A}$*.*

Proof. See Section **??** □

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

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 $\mathbf{B}$ to be a good approximation to $\mathbf{A}$. 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 $||\mathbf{A}||$, and the distance between two matrices $||\mathbf{A} - \mathbf{B}||$. 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 $\mathbf{A} \in \mathbb{R}^{n \times p}$. The **Frobenius norm** of $\mathbf{A}$ is

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

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

Note that the Frobenius norm is invariant to rotation by an orthogonal matrix $\mathbf{U}$:

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

Proposition 3.10.

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

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

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

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

□

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

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

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

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

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

Proposition 3.11.

$$||\mathbf{A}||_2 = \sigma_1$$

where σ_1 is the first singular value of $\mathbf{A}$.

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 $\mathbf{A}$ 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

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

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

$$\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top \tag{3.3}$$

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

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

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

Moreover,

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

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 $\mathbf{B}$ be an $n \times p$ matrix of rank k. The null space $\mathcal{N}(\mathbf{B}) \subset \mathbb{R}^p$ must be of dimension $p - k$ by the rank nullity theorem.

Consider the $p \times (k+1)$ matrix $\mathbf{V}_{k+1} = [\mathbf{v}_1 \ \ldots \ \mathbf{v}_{k+1}]$. This has rank $k+1$, and has column space $\mathcal{C}(\mathbf{V}_{k+1}) \subset \mathbb{R}^p$. Because

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

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 $\mathbf{w} \in \mathcal{N}(B) \cap \mathcal{C}(V_{k+1})$, and moreover we can choose $\mathbf{w}$ so that $||\mathbf{w}||_2 = 1$.

Because $\mathbf{w} \in \mathcal{C}(\mathbf{V}_{k+1})$ we can write $\mathbf{w} = \sum_{i=1}^{k+1} w_i \mathbf{v}_i$ with $\sum_{i=1}^{k+1} w_i^2 = 1$.

Then

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

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))
```

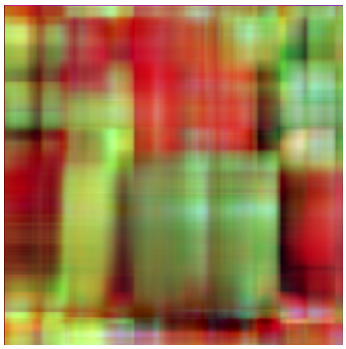
This is a 512×512 colour image, meaning that there are three matrices $\mathbf{R}, \mathbf{B}, \mathbf{G}$ of dimension 512×512) giving the intensity of red, green, and blue for each pixel. Naively storing this matrix requires 5.7Mb.

We can compute the SVD of the three colour intensity matrices, and the view the image that results from using reduced rank versions $\mathbf{B}_k, \mathbf{G}_k, \mathbf{R}_k$ 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 $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$$

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 $\mathbf{A}$, try truncating it and using just a single eigenvalue and eigenvector, i.e., compute

$$\mathbf{A}' = \lambda_1 \mathbf{q}_1 \mathbf{q}_1^\top$$

- Compute the difference between $\mathbf{A}$ and $\mathbf{A}'$ using the 2-norm and the Frobenius norm.

2. The singular value decomposition can be computed in R using the command `svd`. Let $\mathbf{X}$ 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 $\mathbf{X}$ in R and report its singular values.
- Does R report the full or compact SVD?
- Check that $\mathbf{X}\mathbf{v} = \sigma\mathbf{u}$.
- Compute the best rank-1, rank-2, and rank-3 approximations to $\mathbf{X}$, and report the 2-norm and Frobenious norm for these approximations

- Compute the eigenvalues of $\mathbf{X}^\top\mathbf{X}$. How do these relate to the singular values? How does $\mathbf{X}^\top\mathbf{X}$ relate to the sample covariance matrix of the iris data? How do the singular values relate to the eigenvalues of the covariance matrix?
- Let $\mathbf{S}$ be the sample covariance matrix of the iris dataset. What vector maximizes $\mathbf{x}^\top\mathbf{S}\mathbf{x}$?

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. Let $\mathbf{\Sigma}$ be an arbitrary covariance matrix.
   - Show $\mathbf{\Sigma}$ 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?
2. Compute, by hand (but check your answer in R), the singular value decomposition (full and compact) of the following matrices.
   - $\begin{pmatrix} 2 & 0 \\ 0 & -1 \end{pmatrix}$
   - $\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$
3. Let
$$\mathbf{X} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$
The eigen-decomposition of $\mathbf{X}^\top\mathbf{X}$ is
$$\mathbf{X}^\top\mathbf{X} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}^\top$$
Use this fact to compute answer the following questions:
   - What are the singular values of $\mathbf{X}$?
   - What are the right singular vectors of $\mathbf{X}$?
   - What are the left singular vectors of $\mathbf{X}$?
   - Give the compact SVD of $\mathbf{X}$. Check your answer, noting that the singular vectors are only specified up to multiplication by -1
   - Can you compute the full SVD of $\mathbf{X}$?
   - What is the eigen-decomposition of $\mathbf{X}\mathbf{X}^\top$?
   - Find a generalised inverse of matrix $\mathbf{X}$.

4. The SVD can be used to solve linear systems of the form

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

where $\mathbf{A}$ is a $n \times p$ matrix, with compact SVD $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.

- If $\mathbf{A}$ is a square invertible matrix, show that

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

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

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

Note that you computed the svd for $\mathbf{A}$ in Q2.

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

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2.$$

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

5. Consider the system

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

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

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

$$\tilde{\mathbf{x}} = \sum_{i=1}^{r} \mathbf{v}_i \frac{\mathbf{u}_i^\top\mathbf{y}}{\sigma_i}$$

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

- Consider $\mathbf{x}$ of the form

$$\mathbf{x} = \tilde{\mathbf{x}} + \sum_{i=r+1}^{n} \alpha_i \mathbf{v}_i$$

  and explain why any $\mathbf{x}$ of this form is also a solution to $\mathbf{Bx} = \mathbf{y}$. Thus write out all possible solutions of the equation.
- Prove that $\tilde{\mathbf{x}}$ is the solution with minimum norm, i.e., $||\tilde{\mathbf{x}}||_2 \leq ||\mathbf{x}||_2$. **Hint** $\mathbf{v}_1, \ldots, \mathbf{v}_p$ form a complete orthonormal basis for $\mathbb{R}^p$.

6. 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.

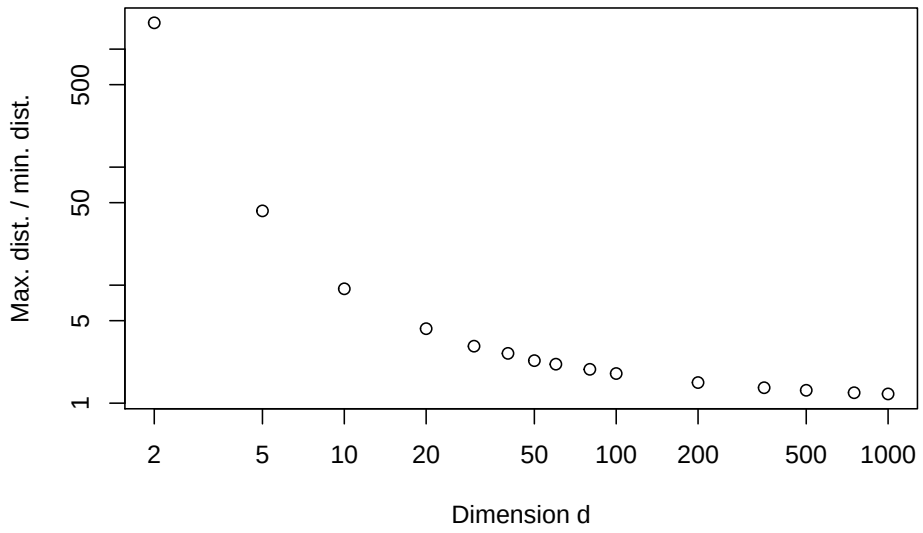
- In reduced dimensions, it is often much easier to understand and appreciate the most important features of a dataset.
- If there is a lot of reduncancy in the data, we might want to reduce the dimension to lower the memory requirements in storing it (e.g. with sound and image compression).
- In high 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 **??**; and Multidimensional Scaling (MDS) in Chapter **??**. 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!

Video

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

$$y = \mathbf{u}^\top \mathbf{x}.$$

for some choice of $\mathbf{u}$.

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

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

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 $\mathbf{u} = (3, 1, 0)^\top$

to get the points score

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

Geometric interpretation

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

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

For the exam data, each data point $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$ is a vector in $\mathbb{R}^4$, and we've expressed $\mathbf{x}$ in terms of its coordinates with respsect to the standard basis, $\mathbf{e}_1^\top = (1\ 0\ 0\ 0)$ etc:

$$\mathbf{x} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3 + x_4\mathbf{e}_4.$$

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

How do we project onto subspace U?

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

  $$\mathbf{u}\mathbf{u}^\top \mathbf{x}.$$

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

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

$$\mathbf{P}_U = \mathbf{u}(\mathbf{u}^\top \mathbf{u})^{-1}\mathbf{u}^\top$$

which simplifies to

$$\mathbf{P}_U = \mathbf{u}\mathbf{u}^\top$$

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

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 $\mathbf{u}$ is an arbitrary decision taken in order to reduce a multidimensional dataset to a single variable (average mark, or points).

A single $\mathbf{u}$ gives a **snapshot** or summary of the data. If $\mathbf{u}$ 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 $\mathbf{u}$, we will use a collection $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r$ and consider the derived variables

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1^\top \mathbf{x} \\ \mathbf{u}_2^\top \mathbf{x} \\ \vdots \\ \mathbf{u}_r^\top \mathbf{x} \end{pmatrix}$$

In matrix notation, if we set

$$\mathbf{U} = \begin{pmatrix} | & & | \\ \mathbf{u}_1 & \ldots & \mathbf{u}_r \\ | & & | \end{pmatrix}$$

then the new derived variable is

$$\mathbf{y} = \mathbf{U}^\top \mathbf{x}.$$

If $\dim(\mathbf{y}) = r < p = \dim(\mathbf{x})$ then we have reduced the dimension of the data. If $\mathbf{y}$ tells us all we need to know about the data, then we can work (plot, analyse, model) with $\mathbf{y}$ instead of $\mathbf{x}$. 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 $\mathbf{U}$. The methods can all be expressed as constrained optimization problems:

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

The objective $f_{\mathbf{X}}(\mathbf{U})$ 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 $\mathbf{U}$ must be (partially) orthogonal, but in other methods other constraints are used

4.1 PCA: an informal introduction

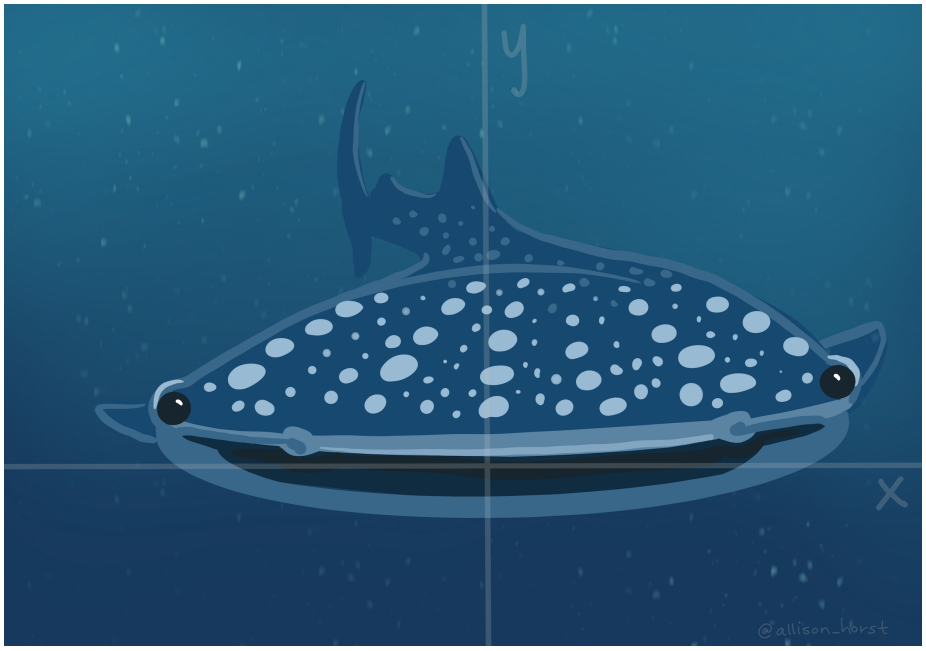
There are two different ways of motivating principal component analysis (PCA), which may in part explain why PCA is so widely used.

The first motivation, and the topic of this section, is to introduce PCA as method for maximizing the variance of the transformed variables $\mathbf{y}$. We start by choosing $\mathbf{u}_1$ so that $y_1 = \mathbf{u}_1^\top \mathbf{x}$ has maximum variance. We then choose $\mathbf{u}_2$ so that $y_2 = \mathbf{u}_2^\top \mathbf{x}$ 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 $\mathbf{X}$ 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 $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be $p \times 1$ vectors of measurements on n experimental units and write

$$\mathbf{X} = \begin{pmatrix} - & \mathbf{x}_1^\top & - \\ - & \mathbf{x}_2^\top & - \\ - & .. & - \\ - & \mathbf{x}_n^\top & - \end{pmatrix}$$

IMPORTANT NOTE: In this section we will assume that $\mathbf{X}$ has been column centered so that the mean of each column is 0 (i.e., the sample mean of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is the zero vector $\mathbf{0} \in \mathbb{R}^p$). If $\mathbf{X}$ has not been column centered, replace $\mathbf{X}$ by

$$\mathbf{HX}$$

where $\mathbf{H}$ is the centering matrix (see 2.4), or equivalently, replace $\mathbf{x}_i$ by $\mathbf{x}_i - \bar{\mathbf{x}}$. It is possible to write out the details of PCA replacing $\mathbf{X}$ by $\mathbf{HX}$ 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 $\mathbf{X}$ (assuming it has been column centered) is

$$\mathbf{S} = \frac{1}{n}\mathbf{X}^\top\mathbf{X} = \frac{1}{n}\sum \mathbf{x}_i\mathbf{x}_i^\top$$

Given some vector $\mathbf{u}$, the transformed variables

$$y_i = \mathbf{u}^\top\mathbf{x}_i$$

have

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

4.1.2 First principal component

We would like to find the $\mathbf{u}$ which maximises the sample variance, $\mathbf{u}^\top\mathbf{S}\mathbf{u}$ over unit vectors $\mathbf{u}$, i.e., vectors with $||\mathbf{u}|| = 1$. 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 $\mathbf{u}$ by $10\mathbf{u}$ it would increase the variance by a factor of 100. Thus, we constrain the problem and only consider unit vectors for $\mathbf{u}$.

We know from Proposition 3.7 in Section 3.4 that $\mathbf{v}_1$, the first eigenvector of $\mathbf{S}$ (also the first right singular vector of $\mathbf{X}$), maximizes $\mathbf{u}^\top\mathbf{S}\mathbf{u}$ with

$$\max_{\mathbf{u}:||\mathbf{u}||=1} \mathbf{u}^\top\mathbf{S}\mathbf{u} = \mathbf{v}_1\mathbf{S}\mathbf{v}_1 = \lambda_1$$

where λ_1 is the largest eigenvalue of $\mathbf{S}$.

So the first principal component of $\mathbf{X}$ is $\mathbf{v}_1$, and the first transformed variable (sometimes called a principal component score) is $y_1 = \mathbf{v}_1^\top\mathbf{x}$. Applying this to each data point we get n instances of this new variable

$$y_{i1} = \mathbf{v}_1^\top\mathbf{x}_i.$$

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

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

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

$$\sigma_i(c\mathbf{X}) = c\sigma_i(\mathbf{X})$$

and in particular

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

We will need to remember this scaling if we use the SVD of $\mathbf{X}$ to do PCA. Note that scaling $\mathbf{X}$ 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 $\mathbf{u}_2$ should we choose for $y_2 = \mathbf{u}_2^\top \mathbf{x}$? 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 $\mathbf{u}_2^\top \mathbf{x}$ is

$$\begin{aligned} s_{y_2 y_1} &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{u}_2^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{v}_1 \\ &= \mathbf{u}_2^\top \mathbf{S}\mathbf{v}_1 \\ &= \lambda_1 \mathbf{u}_2^\top \mathbf{v}_1 \text{ as } \mathbf{v}_1 \text{ is an eigenvector of } S \end{aligned}$$

So to make y_2 uncorrelated with y_1 we have to choose $\mathbf{u}_2$ to be orthogonal to $\mathbf{v}_1$, i.e., $\mathbf{u}_2^\top \mathbf{v}_1 = 0$. So we choose $\mathbf{u}_2$ to be the solution to the optimization problem

$$\max_{\mathbf{u}} \mathbf{u}^\top \mathbf{S}\mathbf{u} \text{ subject to } \mathbf{u}^\top \mathbf{v}_1 = 0.$$

The solution to this problem is to take $\mathbf{u}_2 = \mathbf{v}_2$, i.e., the second eigenvector of $\mathbf{S}$ (or second right singular vector of $\mathbf{X}$), and then

$$\mathbf{v}_2^\top \mathbf{S}\mathbf{v}_2 = \lambda_2.$$

We'll prove this result in the next section.

Later principal components

Our first transformed variable is

$$y_{i1} = \mathbf{v}_1^\top \mathbf{x}_i$$

and our second transformed variable is

$$y_{i2} = \mathbf{v}_2^\top \mathbf{x}_i.$$

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

$$y_{ij} = \mathbf{v}_j^\top \mathbf{x}_i.$$

where $\mathbf{v}_j$ is the j^{th} eigenvector of $\mathbf{S}$.

- 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 $\mathbf{x}$ onto a subspace V. The basis vectors for this subspace are the eigenvectors of $\mathbf{S}$, which are the same as the right singular vectors of $\mathbf{X}$ (the loadings):

$$V = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}.$$

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

$$\mathbf{P}_V = \mathbf{V}\mathbf{V}^\top$$

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

$$\mathbf{y}_i = \begin{pmatrix} y_{i1} \\ \vdots \\ y_{ir} \end{pmatrix} = \mathbf{V}^\top \mathbf{x}_i$$

where

$$\mathbf{V} = \begin{pmatrix} | & & | \\ \mathbf{v}_1 & \ldots & \mathbf{v}_r \\ | & & | \end{pmatrix}$$

is the matrix of right singular vectors from the SVD of $\mathbf{X}$. The transformed variables are

$$\mathbf{Y} = \begin{pmatrix} - & \mathbf{y}_1^\top & - \\ - & .. & - \\ - & \mathbf{y}_n^\top & - \end{pmatrix} = \mathbf{X}\mathbf{V}.$$

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

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}.$$

$\mathbf{Y}$ is a $n \times r$ matrix, and so if $r < p$ we have reduced the dimension of $\mathbf{X}$, 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 $\mathbf{X}$ 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
```

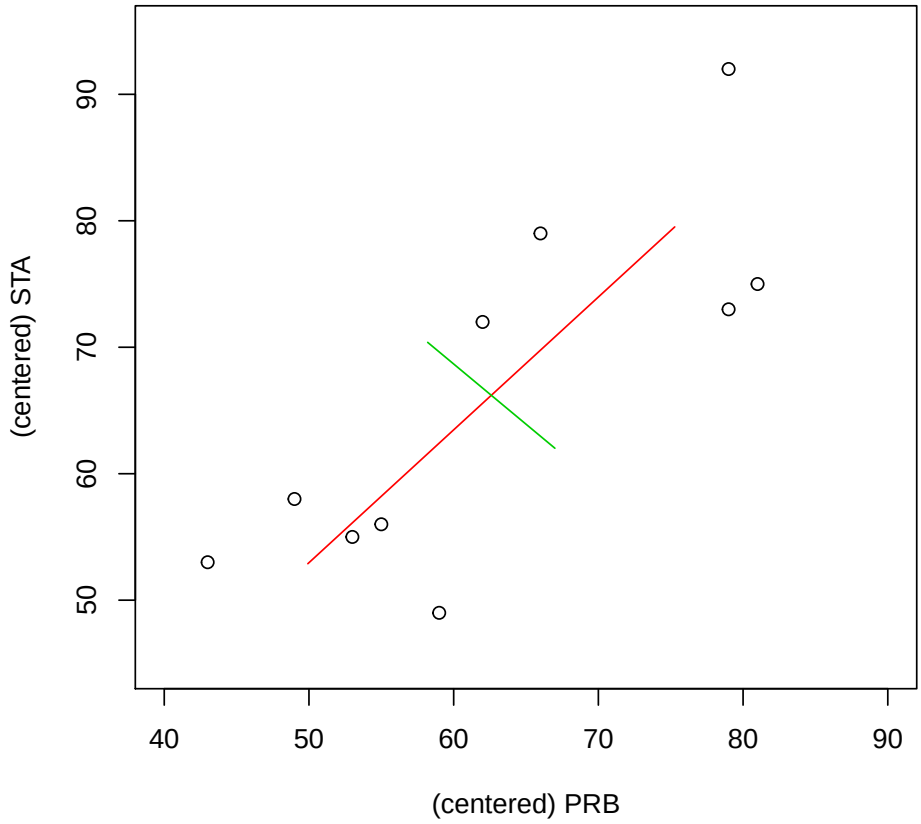
So we can see that the eigenvectors/right singular vectors/loadings are

$$\mathbf{v}_1 = \begin{pmatrix} -0.69 \\ -0.724 \end{pmatrix}, \qquad \mathbf{v}_2 = \begin{pmatrix} -0.724 \\ 0.69 \end{pmatrix}$$

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 $\mathbf{v}_1$ to have negative entries, but we could multiply $\mathbf{v}_1$ 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 $\bar{\mathbf{x}}$, 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 $\mathbf{XV}$ or $\mathbf{U\Sigma}$

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

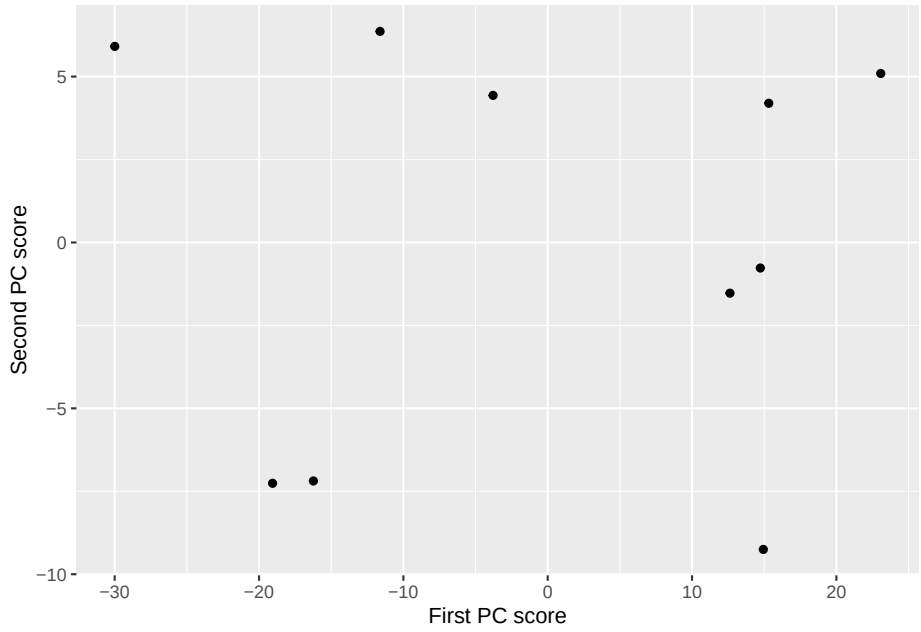
```
##             [,1]       [,2]
##  [1,] -19.060674 -7.2588361
##  [2,] -16.233101 -7.1893271
##  [3,] -11.615016  6.3632849
##  [4,]  14.731183 -0.7695824
```

```
##  [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 $\bar{\mathbf{y}} = \mathbf{0}$. 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).

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

```
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 $\mathbf{X}$ above not $\frac{1}{\sqrt{10}}\mathbf{X}$, 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
```

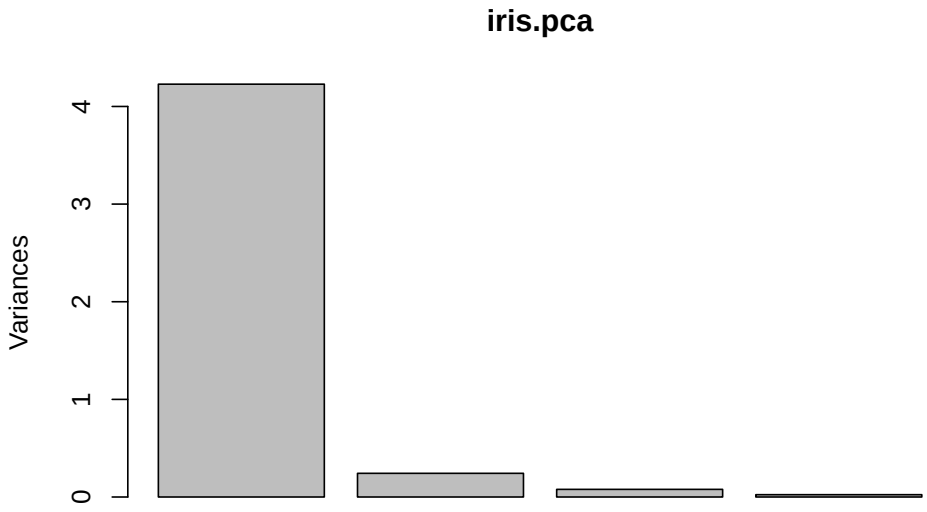
```
##                     PC1         PC2         PC3        PC4
## Sepal.Length  0.36138659 -0.65658877  0.58202985  0.3154872
## Sepal.Width  -0.08452251 -0.73016143 -0.59791083 -0.3197231
## Petal.Length  0.85667061  0.17337266 -0.07623608 -0.4798390
## Petal.Width   0.35828920  0.07548102 -0.54583143  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)
```

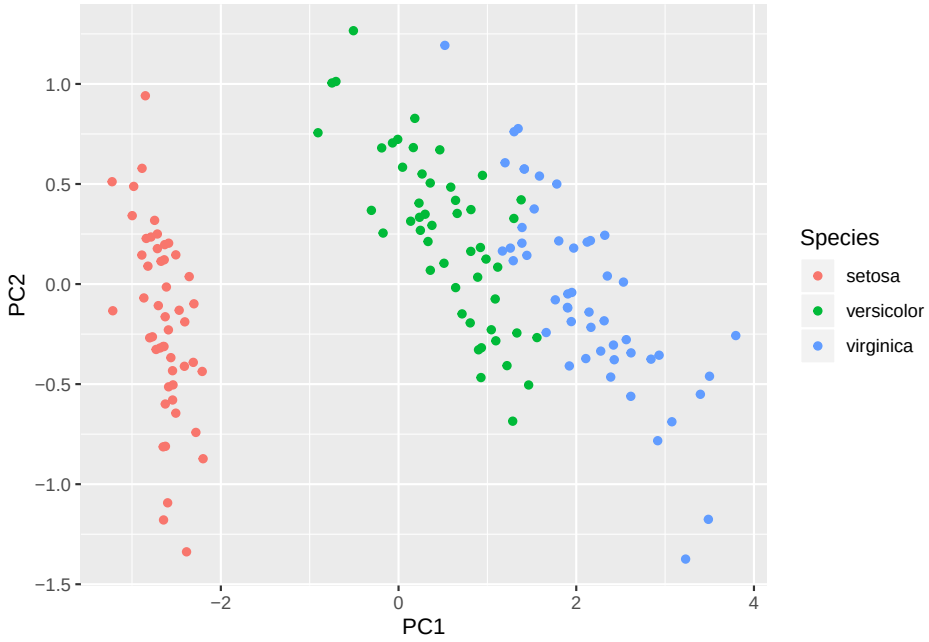



```
summary(iris.pca)
```

```
## 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
```

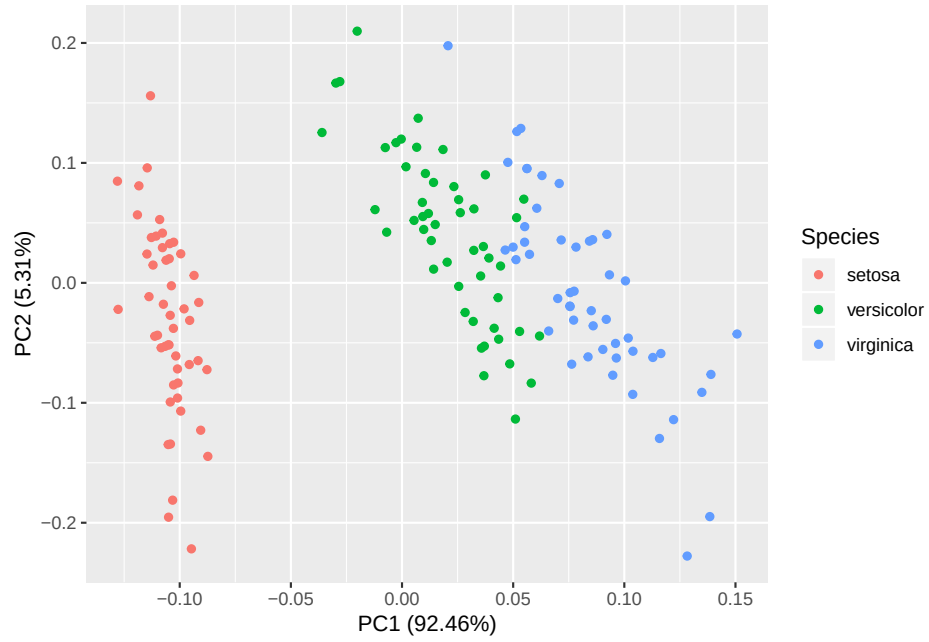
To plot the PC scores, you can either manually create a plot or use the `ggfortify` package. For example, here is a plot of the first two PC scores coloured according to the species of iris.

```
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 $\mathbf{x}_1, \ldots, \mathbf{x}_n$ denote a sample of vectors in $\mathbb{R}^p$ with sample mean vector $\bar{\mathbf{x}}$ and sample covariance matrix $\mathbf{S}$. Suppose $\mathbf{S} = \mathbf{X}^\top \mathbf{H} \mathbf{X}$ has spectral decomposition (see Proposition 3.3)

$$\mathbf{S} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^\top = \sum_{j=1}^{p} \lambda_j \mathbf{v}_j \mathbf{v}_j^\top, \tag{4.3}$$

where the eigenvalues are $\lambda_1 \geq \lambda_2 \geq \lambda_p \geq 0$ with $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_p\}$, and $\mathbf{V}$ contains the eigenvectors of $\mathbf{S}$.

The principal components of $\mathbf{X}$ are defined sequentially. The j^{th} principal component is the solution to the following optimization problem:

$$\max_{\mathbf{u}:\, ||\mathbf{u}||=1} \mathbf{u}^\top \mathbf{S} \mathbf{u} \tag{4.4}$$

subject to

$$\mathbf{v}_k^\top \mathbf{u} = 0, \qquad k = 1, \ldots, j-1. \tag{4.5}$$

(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* $\mathbf{u} = \mathbf{v}_j$.

Proof. We can prove this using the method of Lagrange multipliers. For $j = 1$ our objective is

$$\mathcal{L} = \mathbf{u}^\top \mathbf{S} \mathbf{u} + \lambda(1 - \mathbf{u}^\top \mathbf{u})$$

Differentiating (see 2.1.4) with respect to $\mathbf{u}$ and setting the derivative equal to zero gives

$$2\mathbf{S}\mathbf{u} - 2\lambda \mathbf{u} = 0$$

Rearranging we see that $\mathbf{u}$ must satify

$$\mathbf{S}\mathbf{u} = \lambda \mathbf{u} \text{ with } \mathbf{u}^\top \mathbf{u} = 1$$

i.e., $\mathbf{u}$ is a unit eigenvector of $\mathbf{S}$. Substituting this back in to the objective we see

$$\mathbf{u}\mathbf{S}\mathbf{u} = \lambda$$

and so we must choose $\mathbf{u} = \mathbf{v}_1$, the eigenvector corresponding to the largest eigenvalue of $\mathbf{S}$.

We now proceed inductively and assume the result is true for $k = 1, \ldots, j-1$. The Lagrangian for the j^{th} optimization problem is

$$\mathcal{L} = \mathbf{u}^\top \mathbf{S} \mathbf{u} + \lambda(1 - \mathbf{u}^\top \mathbf{u}) + \sum_{k=1}^{j-1} \mu_k (1 - \mathbf{u}^\top \mathbf{v}_k)$$

where we now have j Lagrange multipliers $\lambda, \mu_1, \ldots, \mu_{j-1}$ - one for each constraint. Differentiating with respect to $\mathbf{u}$ and setting equal to zero gives

$$0 = 2\mathbf{S}\mathbf{u} - 2\lambda\mathbf{u} - \sum_{k=1}^{j-1} \mu_k \mathbf{v}_k = 0$$

If we left multiply by $\mathbf{v}_l$ we get

$$2\mathbf{v}_l^\top \mathbf{S}\mathbf{u} - \lambda \mathbf{v}_l \mathbf{u} - \sum \mu_k \mathbf{v}_l^\top \mathbf{v}_k = 0$$

We know $\mathbf{v}_l$ is an eigenvector of $\mathbf{S}$ and so $\mathbf{S}\mathbf{v}_l = \lambda_l \mathbf{v}_l$ and hence $\mathbf{v}_k \mathbf{S}\mathbf{u} = 0$ as $\mathbf{v}_l^\top \mathbf{u} = 0$. Also

$$\mathbf{v}_l^\top \mathbf{v}_k = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{otherwise,} \end{cases}$$

and thus we've shown that $\mu_l = 0$ for $l = 1, \ldots, j-1$. So again we have that

$$\mathbf{S}\mathbf{u} = \lambda \mathbf{u}$$

i.e., $\mathbf{u}$ must be a unit eigenvector of $\mathbf{S}$. It only remains to show *which* eigenvector it is. Because $\mathbf{u}$ must be orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_{j-1}$, and as $\mathbf{v}_l^\top \mathbf{S}\mathbf{v}_l = \lambda_l$, we must choose $\mathbf{u} = \mathbf{v}_j$, 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

$$y_{ij} = \mathbf{v}_j^\top (\mathbf{x}_i - \bar{\mathbf{x}}), \qquad i = 1, \ldots, n.$$

The j^{th} eigenvector $\mathbf{v}_j$ is sometimes referred to as the vector of **loadings** for the j^{th} PC.

In vector notation

$$\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{ip})^\top = \mathbf{V}^\top (\mathbf{x}_i - \bar{\mathbf{x}}), \qquad i = 1, \ldots, n.$$

In matrix form, the full set of PC scores is given by

$$\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\top = \mathbf{H}\mathbf{X}\mathbf{V}.$$

If $\tilde{\mathbf{X}} = \mathbf{H}\mathbf{X}$ is the column centered data matrix, with singular value decomposition $\tilde{\mathbf{X}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ with $\mathbf{V}$ as in Equation (4.3), then

$$\mathbf{Y} = \tilde{\mathbf{X}}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}.$$

The transformed variables $\mathbf{y} = \mathbf{H}\mathbf{X}\mathbf{V}$ 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* $\mathbf{y}_1, \ldots, \mathbf{y}_n$ *is the zero vector:* $\bar{\mathbf{y}} = \mathbf{0}_p$
2. *The sample covariance matrix of* $\mathbf{y}_1, \ldots, \mathbf{y}_n$ *is*

$$\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_p)$$

   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* $\{y_{ij}\}_{i=1,\ldots,n}$ *is greater than or equal to the sample variance of* $\{y_{ik}\}_{i=1,\ldots,n}$*.*

$$\mathbf{q}_1^\top \mathbf{S}\mathbf{q}_1 \geq \mathbf{q}_2^\top \mathbf{S}\mathbf{q}_2 \geq \ldots \geq \mathbf{q}_p^\top \mathbf{S}\mathbf{q}_p \geq 0$$

4. *The sum of the sample variances is equal to the trace of* $\mathbf{S}$

$$\sum_{j=1}^{p} \mathbf{q}_j^\top \mathbf{S}\mathbf{q}_j = \sum_{j=1}^{p} \lambda_j = tr(\mathbf{S})$$

5. *The product of the sample variances is equal to the determinant of* $\mathbf{S}$

$$\prod_{j=1}^{p} \mathbf{q}_j^\top \mathbf{S}\mathbf{q}_j = \prod_{j=1}^{p} \lambda_j = |\mathbf{S}|.$$

Proof. For i.

$$\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{V}^\top(\mathbf{x}_i - \bar{\mathbf{x}}) = \frac{1}{n}\mathbf{V}^\top \sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}}) = \mathbf{0}.$$

For 2. the sample covariance matrix of $\mathbf{y}_1, \ldots, \mathbf{y}_n$ is

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i^\top &= \frac{1}{n}\sum \mathbf{V}^\top(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \mathbf{x})^\top \mathbf{V} \\
&= \mathbf{V}^\top \mathbf{S}\mathbf{V} \\
&= \mathbf{V}^\top \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top \mathbf{V} \text{ substiting the spectral decomposition for } \mathbf{S} \\
&= \mathbf{\Lambda}
\end{aligned}$$

3. is a consequence 2. and of ordering the eigenvalues in decreasing magnitude.
4. follows from lemma 2.1 and the spectral decomposition of $\mathbf{S}$:

$$\operatorname{tr}(\mathbf{S}) = \operatorname{tr}(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top) = \operatorname{tr}(\mathbf{V}^\top \mathbf{V}\mathbf{\Lambda}) = \operatorname{tr}(\mathbf{\Lambda}) = \sum \lambda_i$$

5. follows from 3.2.

□

From these properties we say that a proportion

$$\frac{\lambda_j}{\lambda_1 + \ldots + \lambda_p}$$

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

$$\bar{\mathbf{x}} = \begin{pmatrix} 14.4 \\ 9.2 \\ 14.4 \\ 51.7 \\ 51.7 \\ 0 \end{pmatrix}.$$

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

$$\mathbf{S} = \begin{pmatrix} 38.3 & -9.18 & -29.2 & 103 & -57 & 160 \\ -9.18 & 10.2 & -0.98 & -27.5 & -2.24 & -25.2 \\ -29.2 & -0.98 & 30.1 & -75.3 & 59.3 & -135 \\ 103 & -27.5 & -75.3 & 336 & -147 & 483 \\ -57 & -2.24 & 59.3 & -147 & 134 & -281 \\ 160 & -25.2 & -135 & 483 & -281 & 764 \end{pmatrix} \tag{4.6}$$

The eigenvalues of $\mathbf{S}$ are

$$\mathbf{\Lambda} = \text{diag}\begin{pmatrix} 1300 & 71.9 & 8.05 & 4.62 & -2.65e-14 & -3.73e-14 \end{pmatrix}$$

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

$$\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_6] = \begin{pmatrix} 0.166 & -0.0262 & 0.707 & 0.373 & -0.577 & 0 \\ -0.0282 & 0.275 & -0.661 & 0.391 & -0.577 & 1.06e-14 \\ -0.138 & -0.249 & -0.0455 & -0.764 & -0.577 & -2.04e-14 \\ 0.502 & -0.6 & -0.202 & 0.117 & 3.22e-15 & -0.577 \\ -0.285 & -0.701 & -0.11 & 0.286 & -6.11e-15 & 0.577 \\ 0.787 & 0.101 & -0.0915 & -0.169 & -3.33e-16 & 0.577 \end{pmatrix}$$

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:

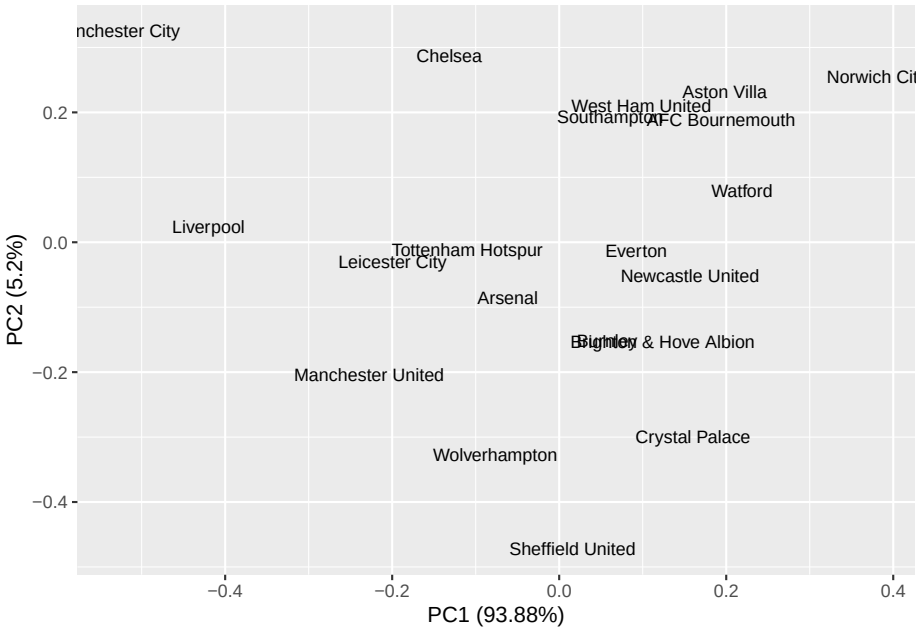
$$\begin{aligned} y_{i1} = {} & 0.17(W_i - \bar{W}) + -0.03(D_i - \bar{D}) + -0.14(L_i - \bar{L}) \\ & + 0.5(G_i - \bar{G}) + -0.28(GA_i - \bar{GA}) + 0.79(GD_i - \bar{GD}), \end{aligned}$$

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.



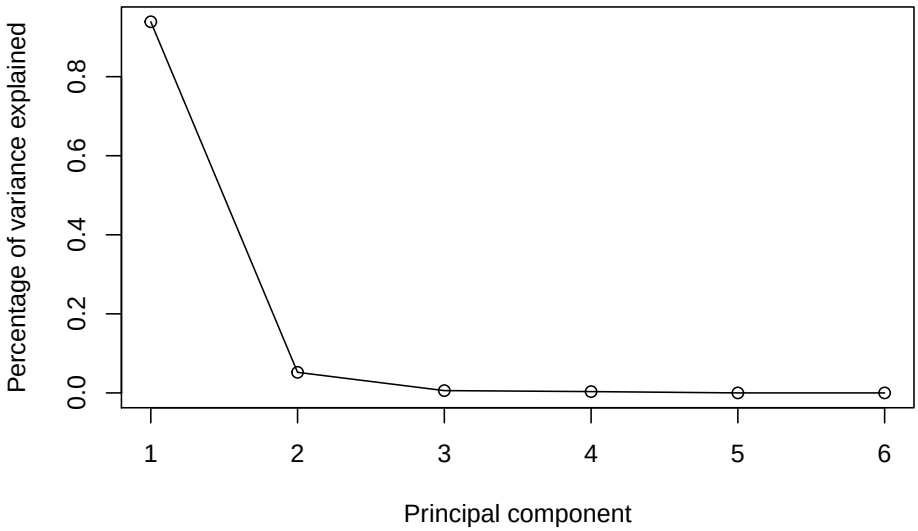
We might interpret the PCs as follows. The first PC seems to measure the difference in goals scored and conceded between teams. It rewards teams with 0 for positive goal difference, and 0.37 for each goal scored, whilst penalising them by -0.58 for every goal they concede. So a team with a large positive 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:

$$\mathbf{\Lambda} = \text{diag}\begin{pmatrix} 4.51 & 1.25 & 0.156 & 0.0863 & 3.68e-32 & 2.48e-33 \end{pmatrix}$$

$$\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_6] = \begin{pmatrix} -0.456 & 0.149 & -0.342 & -0.406 & 0.466 & 0.52 \\ 0.143 & -0.844 & 0.344 & -0.143 & 0.24 & 0.268 \\ 0.432 & 0.321 & 0.186 & 0.541 & 0.413 & 0.461 \\ -0.438 & 0.214 & 0.7 & -0.0181 & 0.389 & -0.348 \\ 0.419 & 0.342 & 0.386 & -0.671 & -0.245 & 0.22 \\ -0.466 & -0.00136 & 0.302 & 0.269 & -0.586 & 0.525 \end{pmatrix}$$

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.

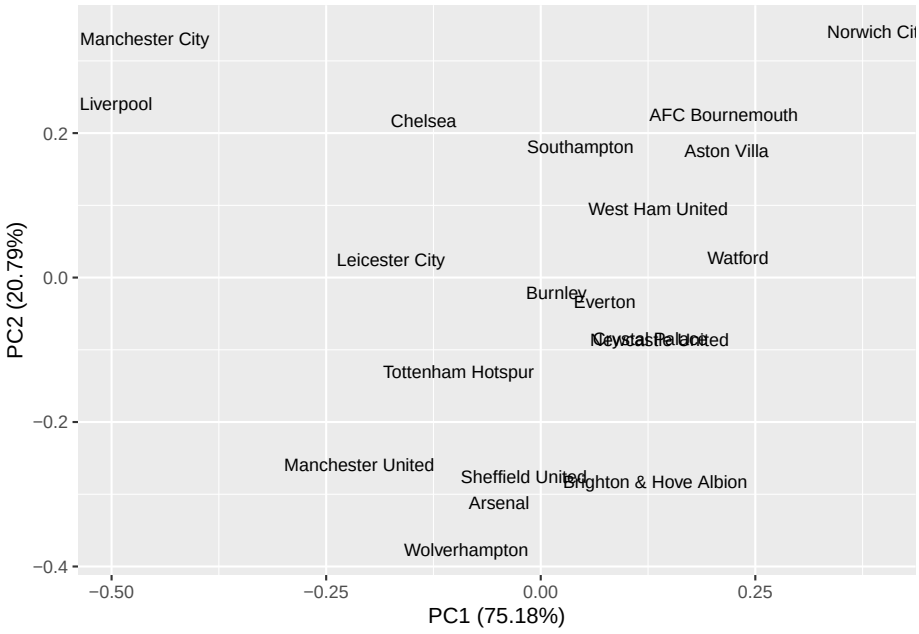
Teams will have a small value of PC1 score if they won lots, lost rarely, scored a lot, and conceded rarely. In other words, PC1 is a complete measure of overall

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:

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top.$$

We note now that there is a *population* analogue of PCA based on the population covariance matrix $\boldsymbol{\Sigma}$. Although the population version of PCA is not of as much direct practical relevance as sample PCA, it is nevertheless of conceptual importance.

Let $\mathbf{x}$ denote a $p \times 1$ random vector with $\mathbb{E}(\mathbf{x}) = \boldsymbol{\mu}$ and $\mathbb{V}\mathrm{ar}(\mathbf{x}) = \boldsymbol{\Sigma}$. 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

$$\boldsymbol{\Sigma} = \sum_{j=1}^{p} \check{\lambda}_j \check{\mathbf{v}}_j \check{\mathbf{v}}_j^\top = \check{\mathbf{V}}\check{\boldsymbol{\Lambda}}\check{\mathbf{V}}^\top$$

where the 'check' symbol $\check{}$ is used to distinguish population quantities from their sample analogues.

Then:

- the first population PC is defined by $Y_1 = \check{\mathbf{v}}_1^\top(\mathbf{x} - \boldsymbol{\mu})$;
- the second population PC is defined by $Y_2 = \check{\mathbf{v}}_2^\top(\mathbf{x} - \boldsymbol{\mu})$;
- ...
- the pth population PC is defined by $Y_p = \check{\mathbf{v}}_p^\top(\mathbf{x} - \boldsymbol{\mu})$.

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

$$\mathbf{y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \ldots \\ \ldots \\ Y_p \end{pmatrix} = \check{\mathbf{V}}^\top(\mathbf{x} - \boldsymbol{\mu}).$$

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. $\sum_{j=1}^{p} \mathbb{V}\,\mathrm{ar}(Y_j) = \sum_{j=1}^{p} \check{\lambda}_j = tr(\boldsymbol{\Sigma})$;
6. $\prod_{j=1}^{p} Var(Y_j) = \prod_{j=1}^{p} \check{\lambda}_j = |\boldsymbol{\Sigma}|$.

Note that, defining $\mathbf{y} = (Y_1, \ldots, Y_p)^\top$ as before, part 1. implies that $\mathbb{E}(\mathbf{y}) = \mathbf{0}_p$ and parts 2. and 3. together imply that

$$\mathrm{Var}(\mathbf{y}) = \boldsymbol{\Lambda} \equiv \mathrm{diag}(\check{\lambda}_1, \ldots, \check{\lambda}_p).$$

Consider now a repeated sampling framework in which we assume that $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are IID random vectors from a population with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

What is the relationship between the sample PCA based on the sample of observed vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, and the population PCA based on the unobserved random vector $\mathbf{x}$, from the same population?

If the elements of $\boldsymbol{\Sigma}$ are all finite, then as n increases, the elements of the sample covariance matrix $\mathbf{S}$ will converge to the corresponding elements of the population covariance matrix $\boldsymbol{\Sigma}$. 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 $\mathbf{z}_i = \mathbf{x}_i + \mathbf{c}$, where $\mathbf{c}$ is a fixed vector. Under this transformation the sample mean changes, $\bar{\mathbf{z}} = \bar{\mathbf{x}} + \mathbf{c}$, but the sample variance remains $\mathbf{S}$. Consequently, the eigenvalues and eigenvectors remain the same and, therefore, so do the principal component scores/transformed vasriables,

$$\mathbf{y}_i = \mathbf{V}^\top(\mathbf{z}_i - \bar{\mathbf{z}}) = \mathbf{V}^\top(\mathbf{x}_i + \mathbf{c} - (\bar{\mathbf{x}} + \mathbf{c})) = \mathbf{V}^\top(\mathbf{x}_i - \bar{\mathbf{x}}).$$

We say that the principal components are **invariant** under the addition transformation. An important special case is to choose $\mathbf{c} = -\bar{\mathbf{x}}$ so that the PC scores are simply $\mathbf{y}_i = \mathbf{V}^\top \mathbf{z}_i$.

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 $\mathbf{z}_i = \mathbf{D}\mathbf{x}_i$, where $\mathbf{D}$ is a diagonal matrix with positive elements. Under this transformation the sample mean changes from $\bar{\mathbf{x}}$ to $\bar{\mathbf{z}} = \mathbf{D}\bar{\mathbf{x}}$, and the sample covariance matrix changes from $\mathbf{S}$ to $\mathbf{DSD}$. 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

$$\mathbf{D} = \text{diag}(s_{11}^{-1/2}, \ldots, s_{pp}^{-1/2}),$$

where s_{ii} is the ith diagonal element of $\mathbf{S}$. In effect, we have standardised all the new variables to have variance 1. In this case the sample covariance matrix of the $\mathbf{z}_i$'s is simply the sample correlation matrix $\mathbf{R}$ of the original variables, $\mathbf{x}_i$. Therefore, we can carry out PCA on the sample correlation matrix, $\mathbf{R}$, which is invariant to changes of scale.

In summary: $\mathbf{R}$ is scale-invariant while $\mathbf{S}$ is not. To do PCA on $\mathbf{R}$ 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 $\mathbf{R}$ instead of $\mathbf{S}$ completely changed the analysis.

Orthogonal transformations

Thirdly, we consider a transformation by an orthogonal matrix, $\overset{p\times p}{\mathbf{A}}$, such that $\mathbf{A}\mathbf{A}^\top = \mathbf{A}^\top\mathbf{A} = \mathbf{I}_p$, and write $\mathbf{z}_i = \mathbf{A}\mathbf{x}_i$. This is equivalent to rotating and/or reflecting the original data.

Let $\mathbf{S}$ be the sample covariance matrix of the $\mathbf{x}_i$ and let $\mathbf{T}$ be the sample covariance matrix of the $\mathbf{z}_i$. Under this transformation the sample mean changes from $\bar{\mathbf{x}}$ to $\bar{\mathbf{z}} = \mathbf{A}\bar{\mathbf{x}}$, and the sample covariance matrix $\mathbf{S}$ changes from $\mathbf{S}$ to $\mathbf{T} = \mathbf{A}\mathbf{S}\mathbf{A}^\top$.

However, if we write $\mathbf{S}$ in terms of its spectral decomposition $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, then $\mathbf{T} = \mathbf{A}\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top\mathbf{A}^\top = \mathbf{B}\mathbf{\Lambda}\mathbf{B}^\top$ where $\mathbf{B} = \mathbf{A}\mathbf{V}$ is also orthogonal. It is therefore apparent that the eigenvalues of $\mathbf{T}$ are the same as those of $\mathbf{S}$; and the eigenvectors of $\mathbf{T}$ are given by $\mathbf{b}_j$ where $\mathbf{b}_j = \mathbf{A}\mathbf{v}_j$, $j = 1, \ldots, p$. The PC scores of the rotated variables are

$$\mathbf{y}_i = \mathbf{B}^\top(\mathbf{z}_i - \bar{\mathbf{z}}) = \mathbf{V}^\top\mathbf{A}^\top\mathbf{A}(\mathbf{x}_i - \bar{\mathbf{x}}) = \mathbf{V}_1^\top(\mathbf{x}_i - \bar{\mathbf{x}}),$$

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 $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ with zero mean (replace $\mathbf{x}_i$ by $\mathbf{x}_i - \bar{\mathbf{x}}$ if the mean is not zero). In order to find the r leading principal components, we solve the optimization problem

$$\text{For } k = 1, \ldots, r \text{ maximize } \mathbf{u}_k^\top \mathbf{S}\mathbf{u}_k$$
$$\text{subject to } \mathbf{u}_k^\top \mathbf{u}_j = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise.} \end{cases}$$

We can write this in the form given in the introduction to this chapter (Equation (4.1)) as

$$\begin{aligned} &\text{Maximize } \operatorname{tr}(\mathbf{U}^\top \mathbf{S}\mathbf{U}) \\ &\text{subject to } \mathbf{U}^\top \mathbf{U} = \mathbf{I}_r, \end{aligned}$$

as $\operatorname{tr}(\mathbf{U}^\top \mathbf{S}\mathbf{U}) = \sum_{k=1}^r \mathbf{u}_k^\top \mathbf{S}\mathbf{u}_k$ if $\mathbf{U}$ has columns $\mathbf{u}_1, \ldots, \mathbf{u}_r$.

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 $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. One way to think about this is seek a $p \times r$ matrix $\mathbf{U}$ for which the rank r linear model

$$f(\mathbf{y}) = \mathbf{U}\mathbf{y}$$

can be used to represent the data.

Let's choose $\mathbf{y}_i \in \mathbb{R}^r$ and $\mathbf{U}$ to minimize the sum of squared errors

$$\sum_{i=1}^n ||\mathbf{x}_i - \mathbf{U}\mathbf{y}_i||_2^2.$$

If we write

$$\mathbf{Y}^\top = \begin{pmatrix} | & & | \\ \mathbf{y}_1 & \cdots & \mathbf{y}_n \\ | & & | \end{pmatrix}$$

then

$$\begin{aligned}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{U}\mathbf{y}_i||_2^2 &= \text{tr}((\mathbf{X}^\top - \mathbf{U}\mathbf{Y}^\top)^\top(\mathbf{X}^\top - \mathbf{U}\mathbf{Y}^\top)) \\ &= ||\mathbf{X}^\top - \mathbf{U}\mathbf{Y}^\top||_F^2\end{aligned}$$

i.e., we're looking for the rank-r matrix $\mathbf{X}_r$ that minimizes $||\mathbf{X} - \mathbf{X}_r||_F = ||\mathbf{X}^\top - \mathbf{X}_r^\top||_F$, noting that we can write an arbitrary rank-r matrix as $\mathbf{X}_r^\top = \mathbf{U}\mathbf{Y}^\top$ for some $p \times r$ matrix $\mathbf{U}$ and a $n \times r$ matrix $\mathbf{Y}$.

It makes sense to restrict the columns of $\mathbf{U}$ to be orthonormal so that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_r$ as non-orthonormal coordinates systems are confusing. We know that the $\mathbf{u} \in \mathcal{C}(\mathbf{U})$ (where $\mathcal{C}(\mathbf{U})$ is the column space of $\mathbf{U}$) that minimizes

$$||\mathbf{x} - \mathbf{u}||_2$$

is the orthogonal projection of $\mathbf{x}$ onto $\mathcal{C}(\mathbf{U})$, which given the columns of $\mathbf{U}$ are orthonormal is $\mathbf{u} = \mathbf{U}\mathbf{U}^\top\mathbf{x}$ (see Section 2.3.3.1). So we must have $\mathbf{X}_r^\top = \mathbf{U}\mathbf{U}^\top\mathbf{X}^\top$ and $\mathbf{Y}^\top = \mathbf{U}^\top\mathbf{X}^\top$.

So it remains to find the optimal choice for $\mathbf{U}$ by minimizing

$$\begin{aligned}||\mathbf{X}^\top - \mathbf{U}\mathbf{U}^\top\mathbf{X}^\top||_F &= ||\mathbf{X} - \mathbf{X}\mathbf{U}\mathbf{U}^\top||_F \\ &= \text{tr}((\mathbf{X} - \mathbf{X}\mathbf{U}\mathbf{U}^\top)^\top(\mathbf{X} - \mathbf{X}\mathbf{U}\mathbf{U}^\top)) \\ &= \text{tr}(\mathbf{X}^\top\mathbf{X}) - 2\,\text{tr}(\mathbf{U}\mathbf{U}^\top\mathbf{X}^\top\mathbf{X}) + \text{tr}(\mathbf{U}\mathbf{U}^\top\mathbf{X}^\top\mathbf{X}\mathbf{U}\mathbf{U}^\top) \\ &= \text{tr}(\mathbf{X}^\top\mathbf{X}) - \text{tr}(\mathbf{U}^\top\mathbf{X}^\top\mathbf{X}\mathbf{U})\end{aligned}$$

where we've used the fact $\text{tr}(\mathbf{A}\mathbf{B}) = \text{tr}(\mathbf{B}\mathbf{A})$ and that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_r$.

Minimizing the equation above with respect to $\mathbf{U}$ is equivalent to maximizing

$$\text{tr}(\mathbf{U}^\top\mathbf{S}\mathbf{U})$$

which is the maximum variance objective we used to introduce PCA.

So to summarize, the optimization problem

$$\begin{aligned}&\text{Minimize } ||\mathbf{X}^\top - \mathbf{U}\mathbf{U}^\top\mathbf{X}^\top||_F \\ &\text{subject to } \mathbf{U}^\top\mathbf{U} = \mathbf{I}_r,\end{aligned}$$

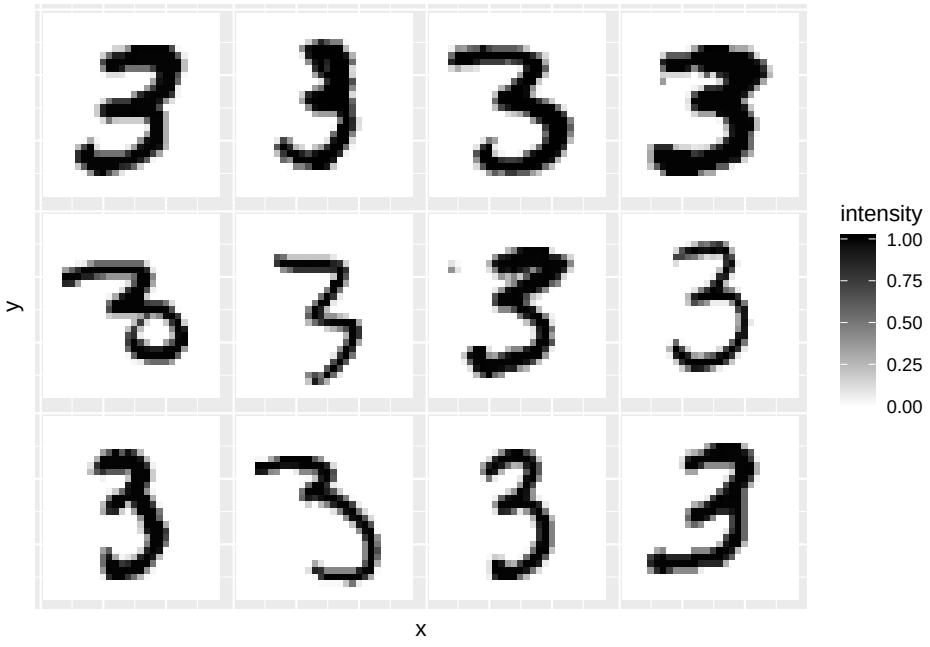
is equivalent to (and has the same as) the PCA optimization problem.

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

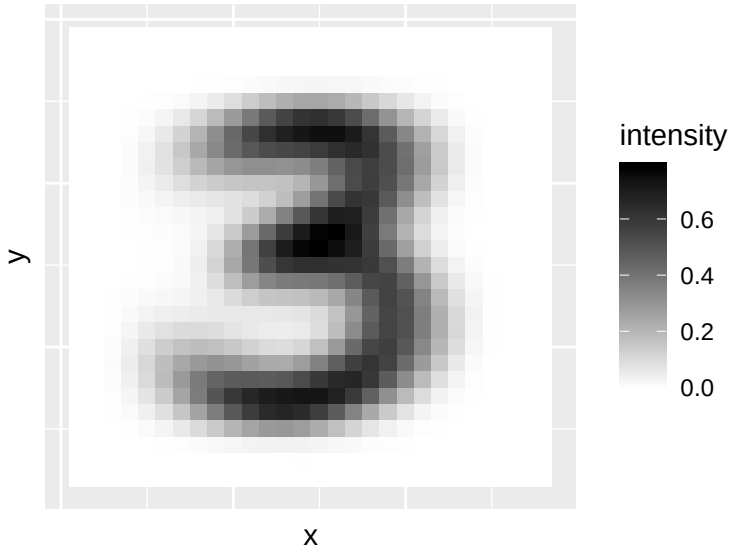
dataset available on Moodle, so you can try this analysis for yourself. Let's look at just the 3s. I've created a plotting function `plot.mnist`, which is in the code file on Moodle.

```
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
```



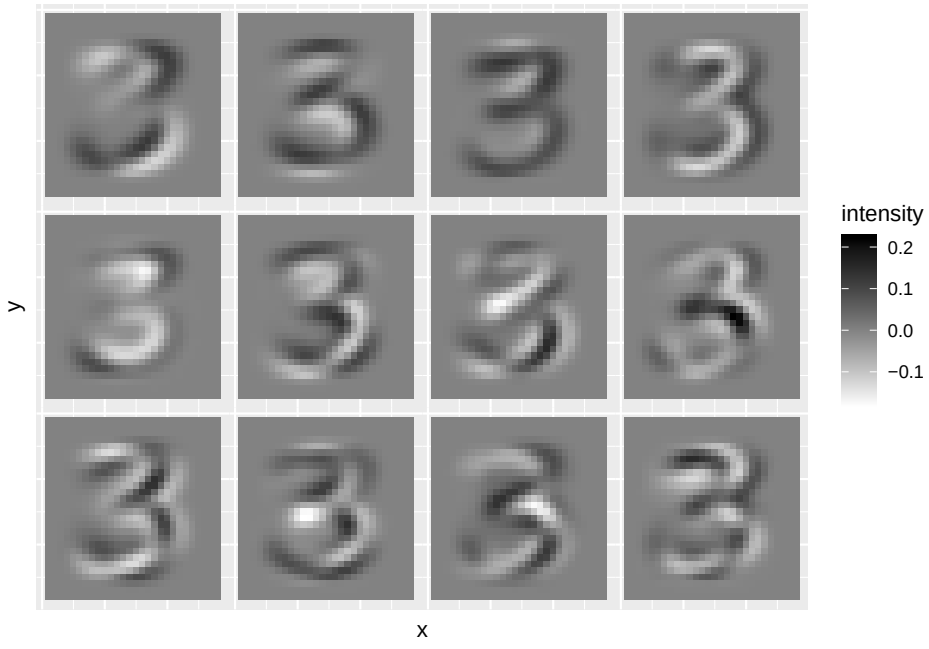
We can see there is quite a bit of variation between them. Now lets look at $\bar{\mathbf{x}}$, the average 3.

```
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])
```

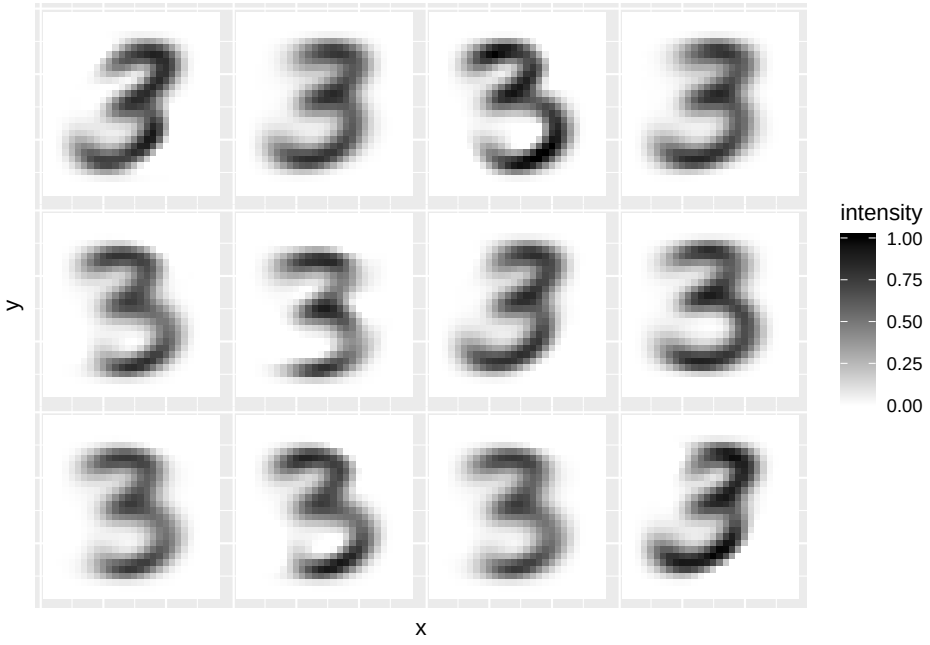


These show the main mode of variability in the 3s. Focusing on the first PC, we can see that this is a form of rotation and causes the 3 to slant either forward or backward. If we wanted a rank-2 approximation to the data we would use

$$f(\mathbf{y}) = \bar{\mathbf{x}} + y_1\mathbf{v}_1 + y_2\mathbf{v}_2$$

Let's try reconstructing the data with $r = 2$.

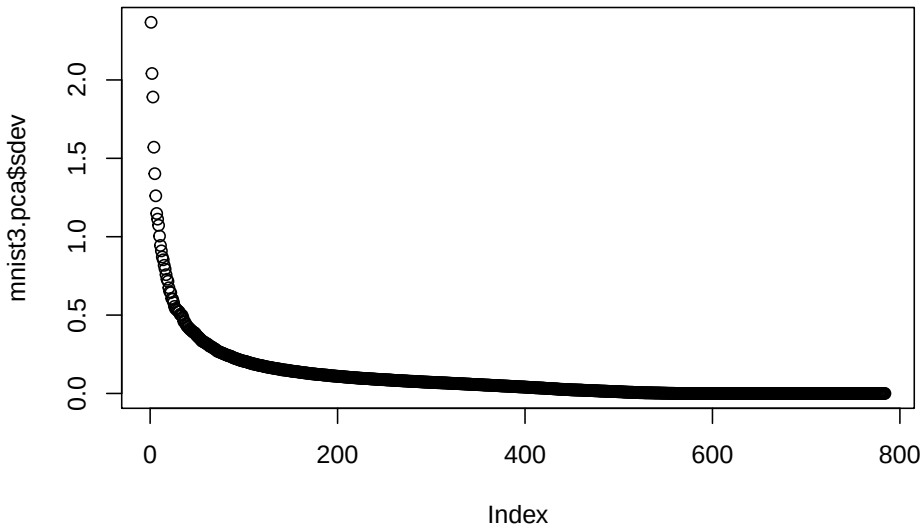
```
r=2
recon =  mnist3.pca$x[,1:r] %*% t(mnist3.pca$rotation[,1:r])
plot.mnist2(matrix(rep(xbar,12), byrow=T, nr=12)+recon[1:12,])
```



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.

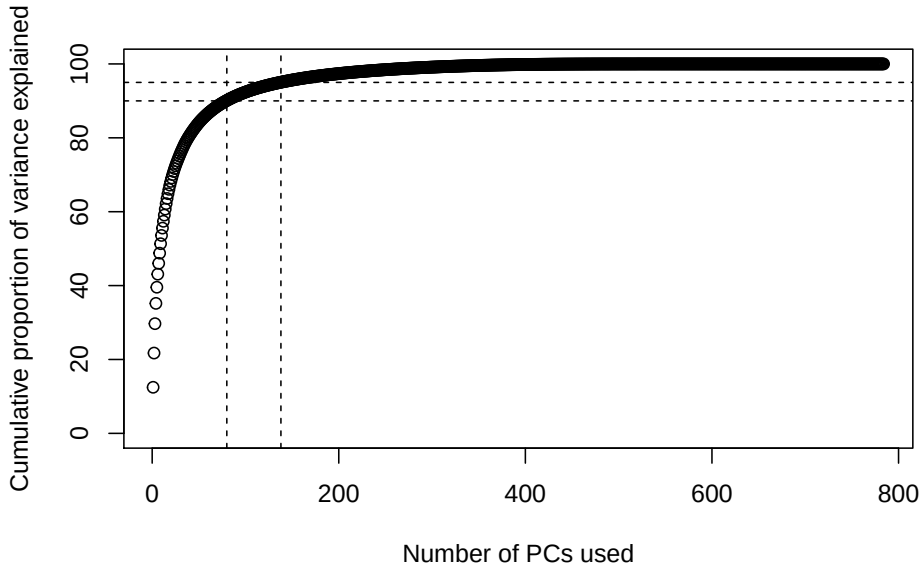
The scree plot shows a sharp decrease in the eigenvalues until about the 100th component, at which point they level off.

```
plot(mnist3.pca$sdev) # scree plot
```



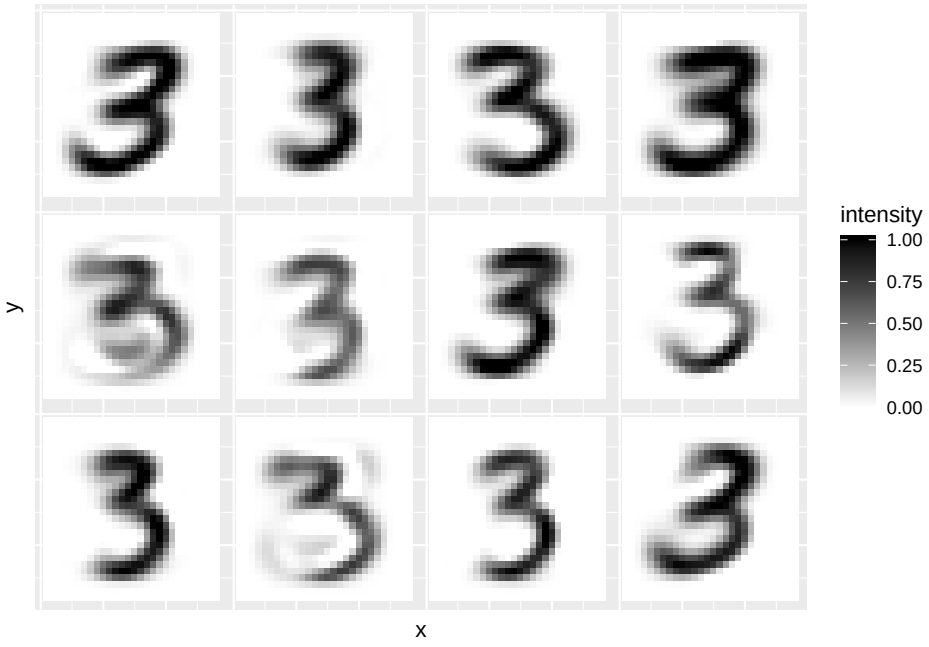
It can also be useful to plot the cumulative sum of the total proportion of variance explained by a given number of principal components. I've drawn on horizontal lines at 90% and 95% of variance explained, to help identify when we cross these thresholds. We need 80 components to explain 90% of the variance, 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)
```

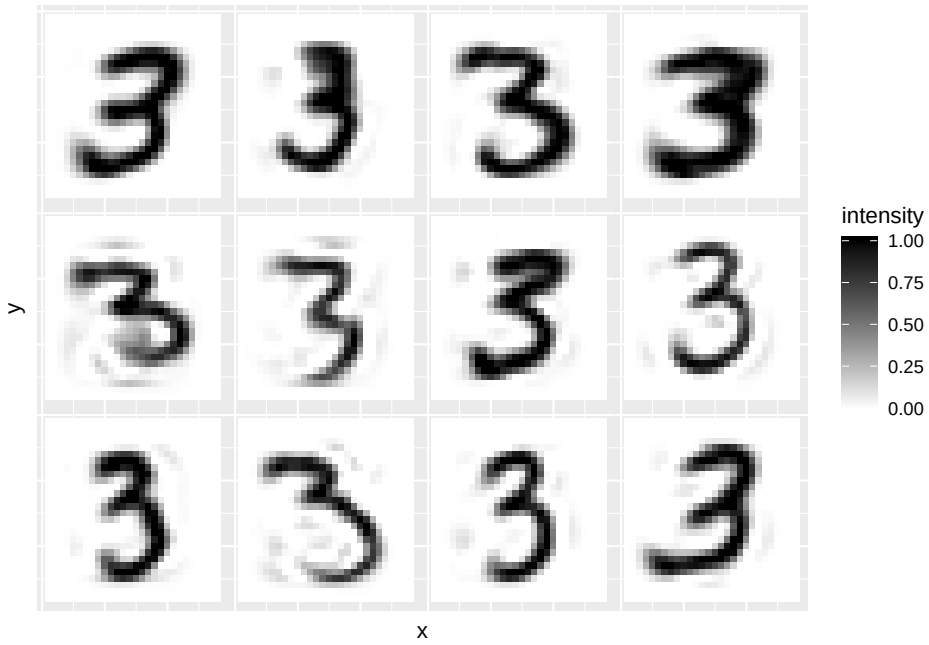


Let's now look at the reconstruction using $r = 10,\ 50,\ 100$ and 500 components to see how the accuracy changes.

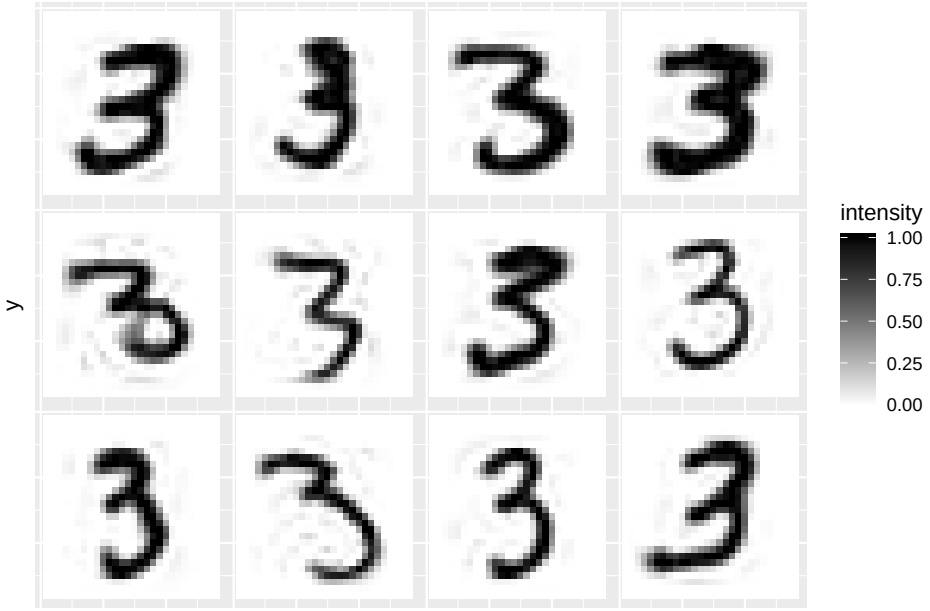
```
r=10
recon =  mnist3.pca$x[,1:r] %*% t(mnist3.pca$rotation[,1:r])
plot.mnist2(matrix(rep(xbar,12), byrow=T, nr=12)+recon[1:12,])
```



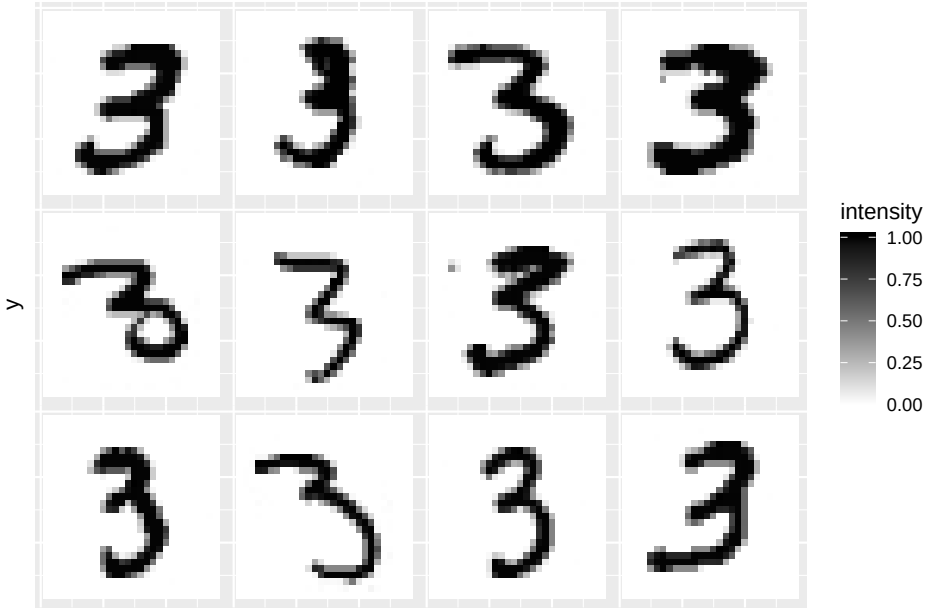
```
r=50
recon =  mnist3.pca$x[,1:r] %*% t(mnist3.pca$rotation[,1:r])
plot.mnist2(matrix(rep(xbar,12), byrow=T, nr=12)+recon[1:12,])
```




```
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,])
```

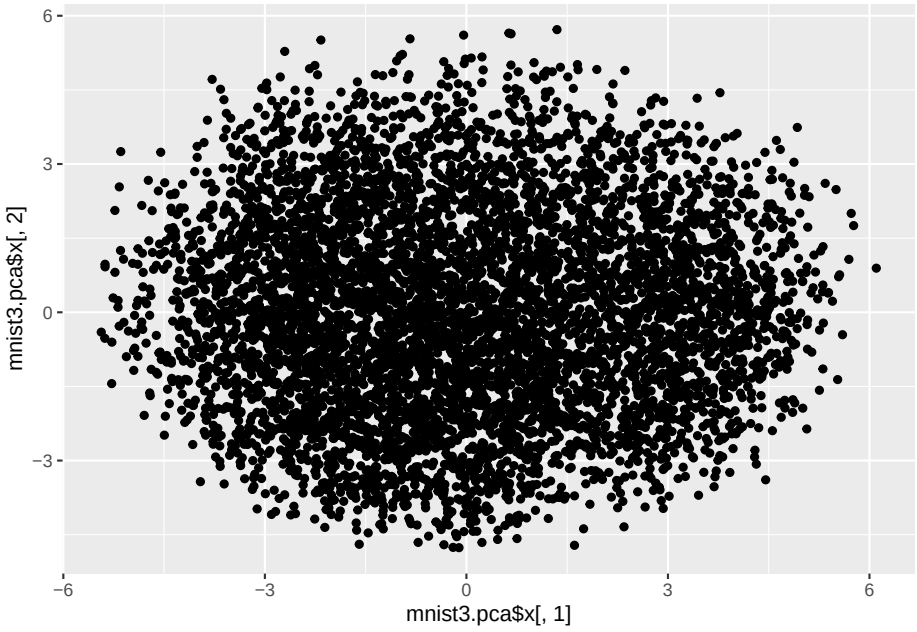


We can see that as the number of components increases the reconstructions start

to look more like the original 12 images.

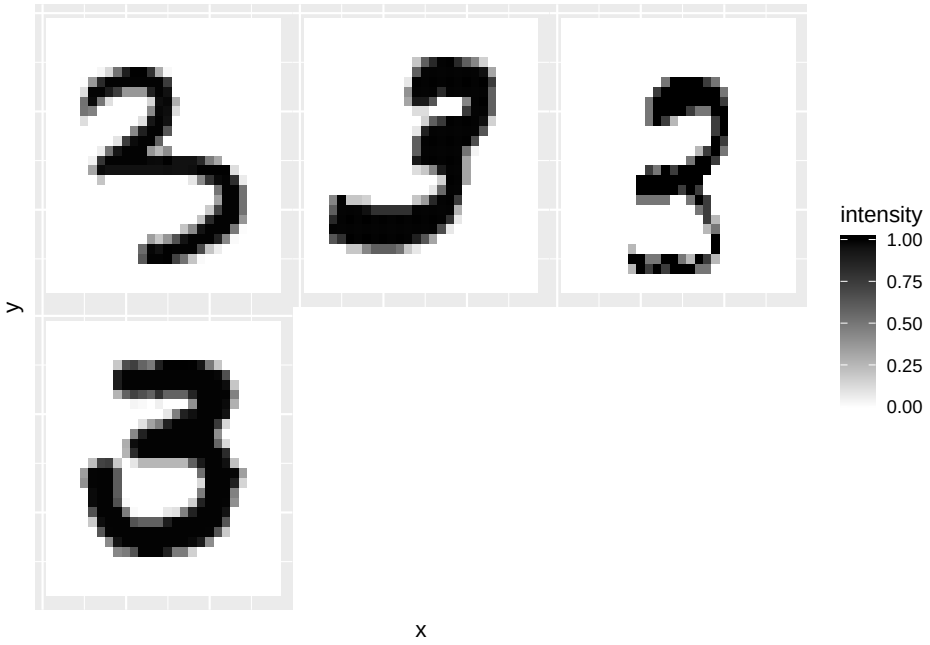
We can visualise the range of 3s by looking at a scatter plot of the first two principal components.

```
library(ggplot2)
qplot(mnist3.pca$x[,1], mnist3.pca$x[,2])
```



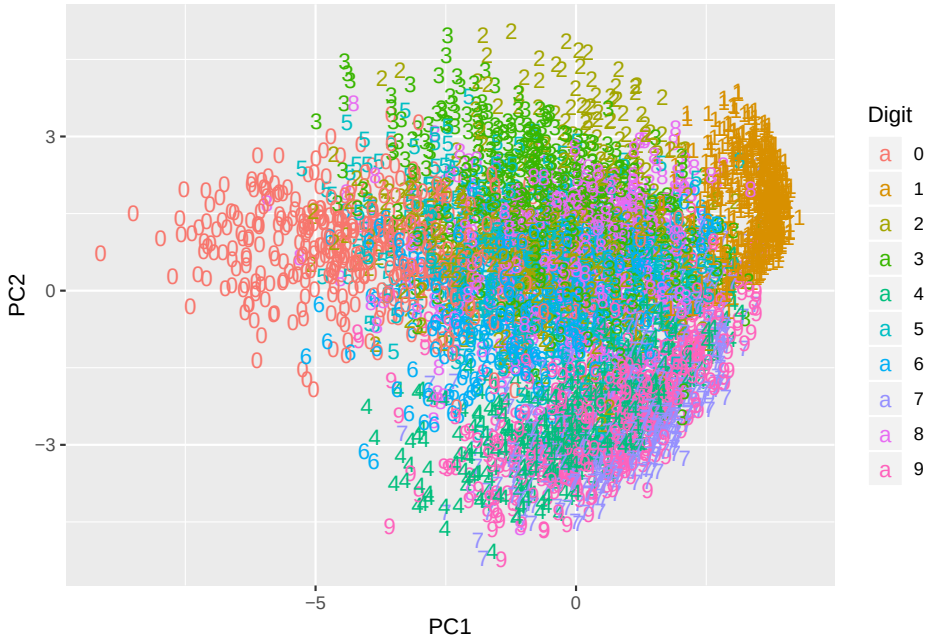
We can then finding images that differ according to these two PC scores. The first plot below is the 3 with the smallest PC1 score, and the second has the largest PC1 score. The third plot has the smallest PC2 score, and the fourth plot the largest PC2 score. These four different 3s differ in more than 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 $\mathbf{X}$ by the matrix of eigenvectors $\mathbf{V}$. 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 $\frac{1}{\sqrt{n-1}}\mathbf{HX}$

- 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 $\mathbf{XV}$ and $\mathbf{U\Sigma}$.

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 $\mathbb{R}^2$

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \ \mathbf{x}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \ \mathbf{x}_3 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \ \mathbf{x}_3 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$$

- What is the orthogonal projection of these points onto

$$\mathbf{u}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and onto

$$\mathbf{u}_2 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \end{pmatrix}?$$

- Compute the sample variance matrix of the three data points, and compute its spectral decomposition.
- What vector $\mathbf{u}$ would maximize the variance of these projection?
- What vector $\mathbf{u}$ would minimize

$$\sum_{i=1}^{4} ||\mathbf{x}_i - \mathbf{u}\mathbf{u}^\top \mathbf{x}_i||_2^2?$$

  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 $\mathbf{\Sigma}$ of the form

$$\mathbf{\Sigma} = \gamma \mathbf{I}_p + \mathbf{a}\mathbf{a}^\top$$

   where $\gamma > 0$ is a scalar, $\mathbf{I}_p$ is the $p \times p$ identity matrix and $\mathbf{a}$ is a vector of dimension p.
   - Show that $\mathbf{a}$ is an eigenvector of $\mathbf{\Sigma}$.
   - Show that if $\mathbf{b}$ is any vector such that $\mathbf{a}^\top \mathbf{b} = 0$, then $\mathbf{b}$ is also an eigenvector of $\mathbf{\Sigma}$.
   - Obtain all the eigenvalues of $\mathbf{\Sigma}$.
   - Determine expressions for the proportion of (population) variability "explained" by: - the largest (population) principal component of $\mathbf{\Sigma}$; - the r largest (population) principal components of $\mathbf{\Sigma}$, 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

$$\mathbf{R} = \begin{pmatrix} 1 & 0.5792 & 0.2414 \\ 0.5792 & 1 & 0.5816 \\ 0.2414 & 0.5816 & 1 \end{pmatrix}.$$

   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.