Dianne Cook and Ursula Laa

Interactively exploring high-dimensional data and models in R

Table of contents

References **65**

Preface

It is important to visualise your data because you might discover things that you could never have anticipated. Although there are many resources available for data visualisation, there are few comprehensive resources on high-dimensional data visualisation. High-dimensional (or multivariate) data arises when many different things are measured for each observation. While we can learn many things from plotting with 1D and 2D or 3D methods there are likely more structures hidden in the higher dimensions. This book provides guidance on visualising high-dimensional data and models using linear projections, with R.

High-dimensional data spaces are fascinating places. You may think that there's a lot of ways to plot one or two variables, and a lot of types of patterns that can be found. You might use a density plot and see skewness or a dot plot to find outliers. A scatterplot of two variables might reveal a non-linear relationship or a barrier beyond which no observations exist. We don't as yet have so many different choices of plot types for high-dimensions, but these types of patterns are also what we seek in scatterplots of high-dimensional data. The additional dimensions can clarify these patterns, that clusters are likely to be more distinct. Observations that did not appear to be very different can be seen to be lonely anomalies in high-dimensions, that no other observations have quite the same combination of values.

What's in this book?

The book is divided into these parts:

- **Introduction**: Here we introduce you to high-dimensional spaces, how they can be visualised, and notation that is useful for describing methods in later chapters.

- **Dimension reduction**: This part covers linear and non-linear dimension reduction. It includes ways to help decide on the number of dimensions needed to summarise the high dimensional data, whether linear dimension

reduction is appropriate, detecting problems that might affect the dimension reduction, and examining how well or badly a non-linear dimension reduction is representing the data.
- **Cluster analysis**: This part described methods for finding groups in data. Although it includes an explanation of a purely graphical approach, it is mostly on using graphics in association with numerical clustering algorithms. There are explanations of assessing the suitability of different numerical techniques for extracting clusters, based on the data shapes, evaluating the clustering result, and showing the solutions in high dimensions.
- **Classification**: This part describes methods for exploring known groups in the data. You'll learn how to check model assumptions, to help decide if a method is suited to the data, examine classification boundaries and explore where errors arise.
- **Miscellaneous**: The material in this part focuses on examining data from different contexts. This includes multiple time series, longitudinal data. A key pre-processing step is to convert the data into Euclidean space.

In each of these parts an emphasis is also showing your model with your data in the high dimensional space.

Our hopes are that you will come away with understanding the importance of plotting your high dimensional data as a regular step in your statistical or machine learning analyses. There are many examples of what you might miss if you don't plot the data. Effective use of graphics goes hand-in-hand with analytical techniques. With high dimensions visualisation is a challenge but it is fascinating, and leads to many surprising moments.

Audience

High-dimensional data arises in many fields such as biology, social sciences, finance, and more. Anyone who is doing exploratory data analysis and model fitting for more than two variables will benefit from learning how to effectively visualise high-dimensions. This book will be useful for students and teachers of multivariate data analysis and machine learning, and researchers, data analysts, and industry professionals who work in these areas.

How to use the book?

The book is written with explanations accompanied by examples with R code. The chapters are organised by types of analysis and focus on how to use the high-dimensional visualisation to complement the commonly used analytical methods. The toolbox chapter in the Appendix provides an overview of the primary high-dimensional visualisation methods discussed in the book and how to get started.

What should I know before reading this book?

The examples assume that you already use R, and have a working knowledge of base R and tidyverse way of thinking about data analysis. It also assumes that you have some knowledge of statistical methods, and some experience with machine learning methods.

If you feel like you need build up your skills in these areas in preparation for working through this book, these are our recommended resources:

- R for Data Science by Wickham and Grolemund for learning about data wrangling and visualisation.
- Introduction to Modern Statistics by Çetinkaya-Rundel and Hardin to learn about introductory statistics.
- Hands-On Machine Learning with R by Boehmke and Greenwell to learn about machine learning.

We will assume you know how to plot your data and models in 2D. Our material starts from 2D and beyond.

Setting up your workflow

To get started set up your computer with the current versions of R and ideally also with Rstudio Desktop.

In addition, we have made an R package to share the data and functions used in this book, called `mulgar`.[1][2]

```
install.packages("mulgar", dependencies=TRUE)
# or the development version
devtools::install_github("dicook/mulgar")
```

To get a copy of the code and data used and an RStudio project to get started, you can download with this code:

```
book_url <- "https://dicook.github.io/mulgar_book/code_and_data.zip"
usethis::use_zip(url=book_url)
```

You will be able to click on the `mulgar_book.Rproj` to get started with the code.

Suggestion, feedback or error?

We welcome suggestions, feedback or details of errors. You can report them as an issue at the Github repo for this book.

Please make a small reproducible example and report the error encountered. Reproducible examples have these components:

- a small amount of data
- small amount of code that generates the error
- copy of the error message that was generated

[1] Mulga is a type of Australian habitat composed of woodland or open forest dominated by the mulga tree. Massive clearing of mulga led to the vast wheat fields of Western Australia. Here **mulgar** is an acronym for **MUL**tivariate **G**raphical **A**nalysis with **R**.

[2] Photo of mulga tree taken by L. G. Cook.

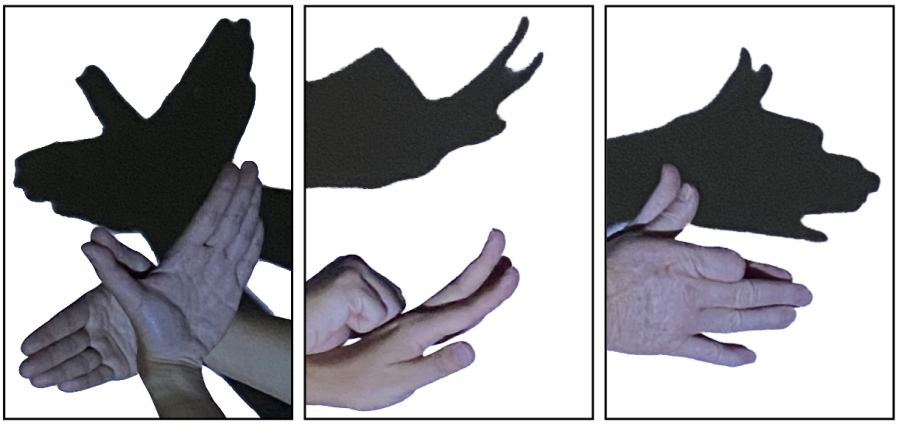
Part I

Introduction

1

Picturing high dimensions

High-dimensional data means that we have a large number of numeric features or variables, which can be considered as dimensions in a mathematical space. The variables can be different types, such as categorical or temporal, but the handling of these variables involves different techniques.



1.1 Getting familiar with tours

Figure 1.1 illustrates a tour for 2D data and 1D projections. The (grand) tour will generate all possible 1D projections of the data, and display with a univariate plot like a histogram or density plot. For this data, the `simple_clusters` data, depending on the projection, the distribution might be clustered into two groups (bimodal), or there might be no clusters (unimodal). In this example, all projections are generated by rotating a line around the centre of the plot. Clustering can be seen in many of the projections, with the strongest being when the contribution of both variables is equal, and the projection is `(0.707, 0.707)` or `(-0.707, -0.707)`. (If you are curious about the number `0.707`, read the last section of this chapter.)

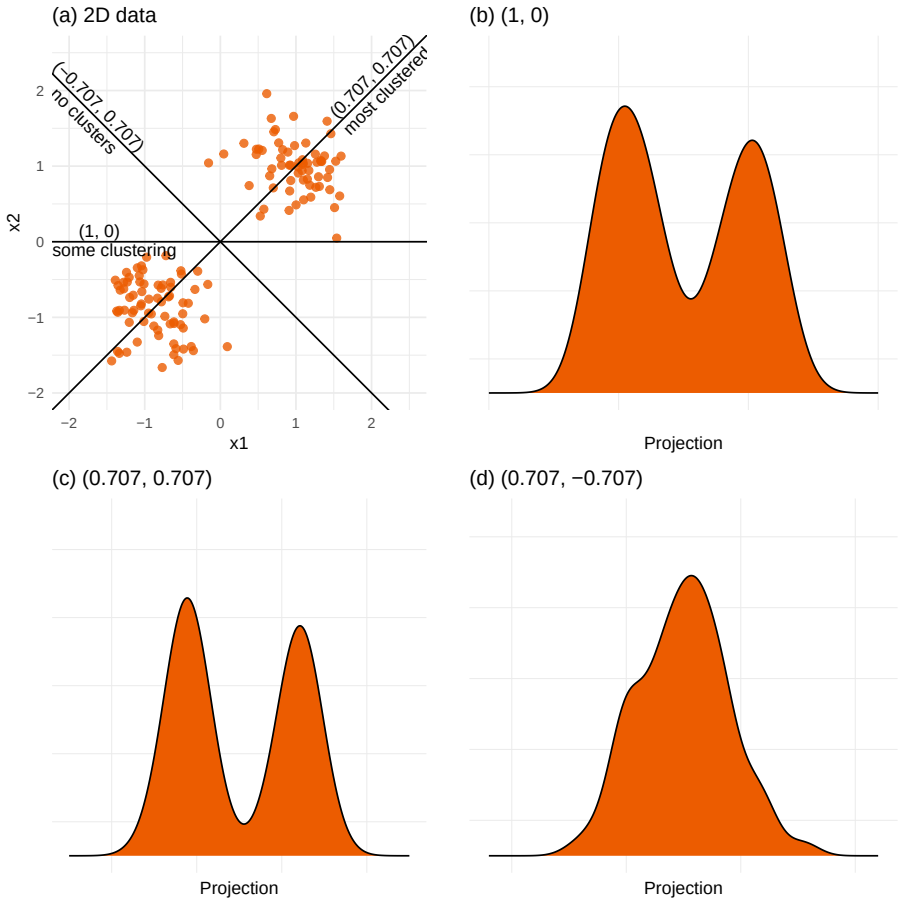


Figure 1.1: How a tour can be used to explore high-dimensional data illustrated using (a) 2D data with two clusters and (b,c,d) 1D projections from a tour shown as a density plot. Imagine spinning a line around the centre of the data plot, with points projected orthogonally onto the line. With this data, when the line is at `x1=x2 (0.707, 0.707)` or `(-0.707, -0.707)` the clustering is the strongest. When it is at `x1=-x2 (0.707, -0.707)` there is no clustering.

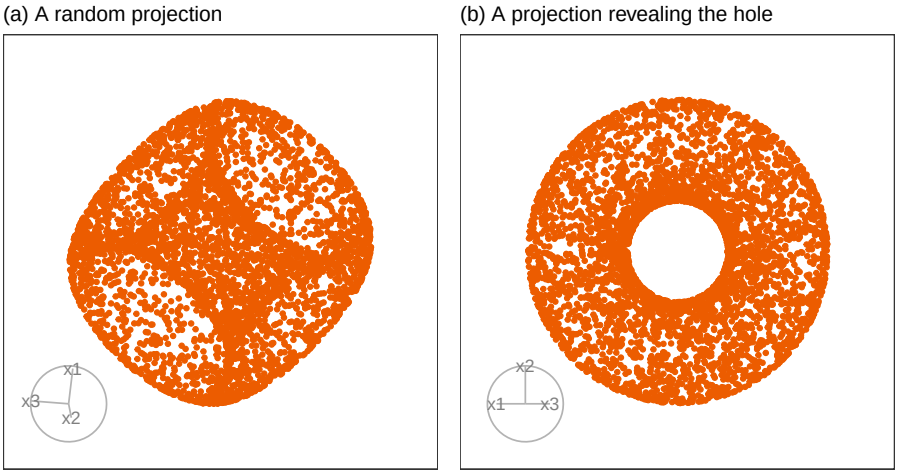


Figure 1.2: How a tour can be used to explore high-dimensional data illustrated by showing a sequence of random 2D projections of 3D data (a). The data has a donut shape with the hole revealed in a single 2D projection (b). Data usually arrives with a given number of observations, and when we plot it like this using a scatterplot, it is like shadows of a transparent object.

Figure 1.2 illustrates a tour for 3D data using 2D projections. The data are points on the surface of a donut shape. By showing the projections using a scatterplot the donut looks transparent and we can see through the data. The donut shape can be inferred from watching many 2D projections but some are more revealing that others. The projection shown in (b) is where the hole in the donut is clearly visible.

1.2 What's different about space beyond 2D?

The term "high-dimensional" in this book refers to the dimensionality of the Euclidean space. Figure 1.3 shows a way to imagine this. It shows a sequence of cube wireframes, ranging from one-dimensional (1D) through to five-dimensional (5D), where beyond 2D is a linear projection of the cube. As the dimension increases, a new orthogonal axis is added. For cubes, this is achieved by doubling the cube: a 2D cube consists of two 1D cubes, a 3D cube consists of two 2D cubes, and so forth. This is a great way to think about the space being examined by the visual methods, and also all of the machine learning methods mentioned, in this book.

Interestingly, the struggle with imagining high-dimensions this way is described

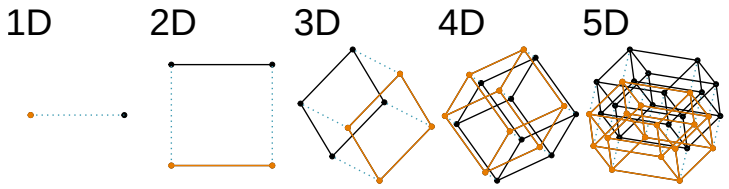


Figure 1.3: Space can be considered to be a high-dimensional cube. Here we have pictured a sequence of increasing dimension cubes, from 1D to 5D, as wireframes, it can be seen that as the dimension increase by one, the cube doubles.

in a novel published in 1884 (Abbott, 1884) [1]. Yes, more than 100 years ago! This is a story about characters living in a 2D world, being visited by an alien 3D character. It also is a social satire, serving the reader strong messages about gender inequity, although this provides the means to explain more intricacies in perceiving dimensions. There have been several movies made based on the book in recent decades (e.g. Martin (1965), D. Johnson & Travis (2007)). Although purchasing the movies may be prohibitive, watching the trailers available for free online is sufficient to gain enough geometric intuition on the nature of understanding high-dimensional spaces while living in a low-dimensional world.

When we look at high-dimensional spaces from a low-dimensional space, we meet the "curse of dimensionality", a term introduced by Bellman (1961) to express the difficulty of doing optimization in high dimensions because of the exponential growth in space as dimension increases. A way to imagine this is look at the cubes in Figure 1.3: As you go from 1D to 2D, 2D to 3D, the space expands a lot, and imagine how vast space might get as more dimensions are added[2]. The volume of the space grows exponentially with dimension, which makes it infeasible to sample enough points – any sample will be less densely covering the space as dimension increases. The effect is that most points will be far from the sample mean, on the edge of the sample space.

For visualisation, the curse manifests in an opposite manner. Projecting from high to low dimensions creates a crowding or piling of points near the center of the distribution. This was noted by Diaconis & Freedman (1984a). Figure 1.4 illustrates this phenomenon. As dimension increases, the points crowd the centre, even with as few as ten dimensions. This is something that we may need to correct for when exploring high dimensions with low-dimensional projections.

Figure 1.5 shows 2D tours of two different 5D data sets. One has clusters

[1] Thanks to Barret Schloerke for directing co-author Cook to this history when he was an undergraduate student and we were starting the geozoo project.

[2] "Space is big. Really big. You might think it's a long way to the pharmacy, but that's peanuts to space." from Douglas Adams' Hitchhiker's Guide to the Galaxy always springs to mind when thinking about high dimensions!

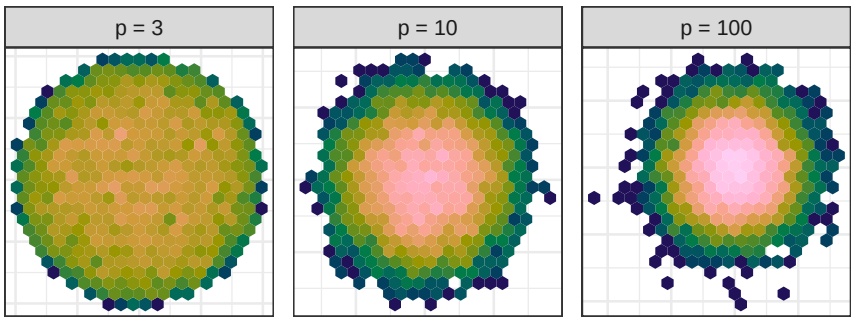
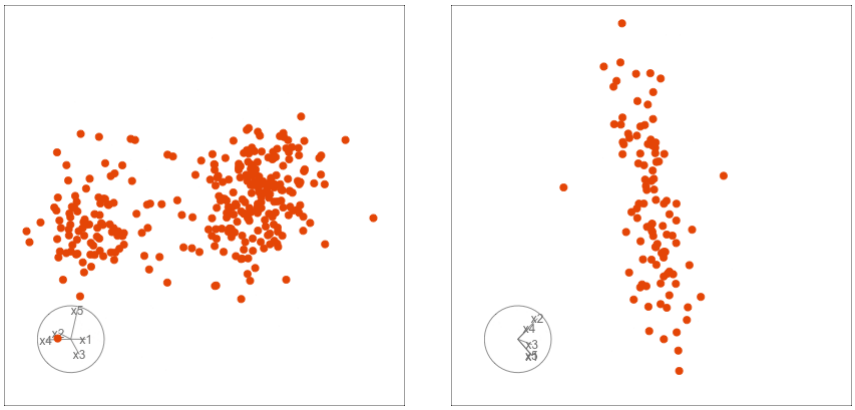


Figure 1.4: Illustration of data crowding in the low-dimensional projection as dimension increases, here from 3, 10, 100. Colour shows the number of points in each hexagon bin (pink is large, navy is small). As dimension increases the points concentrate near the centre.

(a) and the other has two outliers and a plane (b). Can you see these? One difference in the viewing of data with more than three dimensions with 2D projections is that the points seem to shrink towards the centre, and then expand out again. This the effect of dimensionality, with different variance or spread in some directions.



(a) Clusters (b) Outliers

Figure 1.5: Frames from 2D tours on two 5D datasets, with clusters of points in (a) and two outliers with a plane in (b). This figure is best viewed in the HTML version of the book.

1.3 What can you learn?

There are two ways of detecting structure in tours:

- patterns in a single low-dimensional projection
- movement patterns

with the latter being especially useful when displaying the projected data as a scatterplot. Figure 1.6 shows examples of patterns we typically look for when making a scatterplot of data. These include clustering, linear and non-linear association, outliers, barriers where there is a sharp edge beyond which no observations are seen. Not shown, but it also might be possible to observe multiple modes, or density of observations, L-shapes, discreteness or uneven spread of points. The tour is especially useful if these patterns are only visible in combinations of variables.

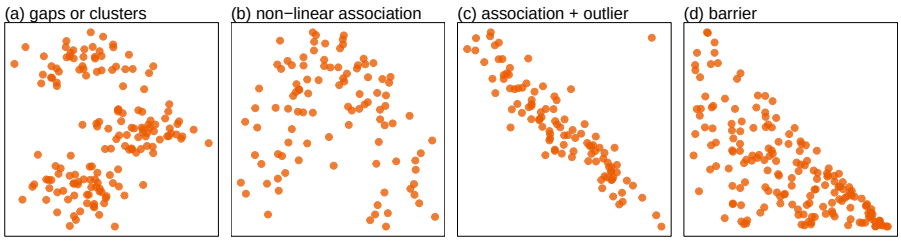
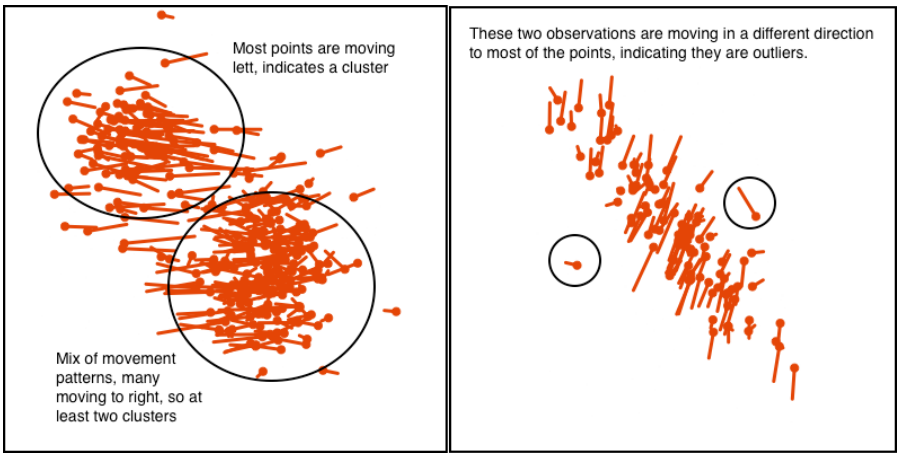


Figure 1.6: Example structures that might be visible in a 2D projection that imply presence of structure in high dimensions. These include clusters, linear and non-linear association, outliers and barriers.

Figure 1.7 illustrates how movement patterns of points when using scatterplots to display 2D projections indicate clustering (a, b) and outliers (c, d).

This type of visualisation is useful for many activities in dealing with high-dimensional data, including:

- exploring high-dimensional data.
- detecting if the data lives in a lower dimensional space than the number of variables.
- checking assumptions required for multivariate models to be applicable.
- check for potential problems in modeling such as multicollinearity among predictors.
- checking assumptions required for probabilties calculated for statistical hypothesis testing to be valid.



(a) Clustering (b) Outliers

Figure 1.7: The movement of points give further clues about the structure of the data in high-dimensions. In the data with clustering, often we can see a group of points moving differently from the others. Because there are three clusters, you should see three distinct movement patterns. It is similar with outliers, except these may be individual points moving alone, and different from all others. This can be seen in the static plot, one point (top left) has a movement pattern upwards whereas most of the other observations near it are moving down towards the right.

- diagnosing the fit of multivariate models.

With a tour we slowly rotate the viewing direction, this allows us to see many individual projections and to track movement patterns. Look for interesting structures such as clusters or outlying points.

1.4 A little history

Viewing high-dimensional data based on low-dimensional projections can probably be traced back to the early work on principal component analysis by Pearson (1901) and Hotelling (1933), which was extended to known classes as part of discriminant analysis by Fisher (1936a).

With computer graphics, the capability of animating plots to show more than a single best projection became possible. The video library (ASA Statistical Graphics Section, 2023) is the best place to experience the earliest work. Kruskal's 1962 animation of multidimensional scaling showed the process of finding a good 2D representation of high dimensional data, although the views are not projections. Chang's 1970 video shows her rotating a high dimensional point cloud along coordinate axes to find a special projection where all the numbers align. The classic video that must be watched is PRIM9 (Fisherkeller et al., 1973) where a variety of interactive and dynamic tools are used together to explore high dimensional physics data, documented in Fisherkeller et al. (1974).

The methods in this book primarily emerge from Asimov (1985)'s grand tour method. The algorithm provided the first smooth and continuous sequence of low dimensional projections, and guaranteed that all possible low dimensional projections were likely to be shown. The algorithm was refined in Buja & Asimov (1986) (and documented in detail in Buja et al. (2005)) to make it *efficiently* show all possible projections. Since then there have been numerous varieties of tour algorithms developed to focus on specific tasks in exploring high dimensional data, and these are documented in S. Lee et al. (2022).

This book is an evolution from Cook & Swayne (2007). One of the difficulties in working on interactive and dynamic graphics research has been the rapid change in technology. Programming languages have changed a little (fortran to C to java to python) but graphics toolkits and display devices have changed a lot! The tour software used in this book evolved from XGobi, which was written in C and used the X Window System, which was then rewritten in GGobi using gtk. The video library has engaging videos of these software

systems There have been several other short-lived implementations, including orca (Sutherland et al., 2000a), written in java, and cranvas (Xie et al., 2014), written in R with a back-end provided by wrapper functions to qt libraries.

Although attempts were made with these ancestor systems to connect the data plots to a statistical analysis system, these were always limited. With the emergence of R, having graphics in the data analysis workflow has been much easier, albeit at the cost of the interactivity with graphics that matches the old systems. We are mostly using the R package, `tourr` (Wickham et al., 2011a) for examples in this book. It provides the machinery for running a tour, and has the flexibility that it can be ported, modified, and used as a regular element of data analysis.

Exercises

1. Randomly generate data points that are uniformly distributed in a hyper-cube of 3, 5 and 10 dimensions, with 500 points in each sample, using the `cube.solid.random` function of the `geozoo` package. What differences do we expect to see? Now visualise each set in a grand tour and describe how they differ, and whether this matched your expectations?
2. Use the `geozoo` package to generate samples from different shapes and use them to get a better understanding of how shapes appear in a grand tour. You can start with exploring the conic spiral in 3D, a torus in 4D and points along the wire frame of a cube in 5D.
3. For each of the challenge data sets, `c1`, ..., `c7` from the `mulgar` package, use the grand tour to view and try to identify structure (outliers, clusters, non-linear relationships).

2

Notation conventions and R objects

The data can be considered to be a matrix of numbers with the columns corresponding to variables, and the rows correspond to observations. It can be helpful to write this in mathematical notation, like:

$$X_{n\times p} = [X_1 \ X_2 \ \ldots \ X_p]_{n\times p} = \begin{bmatrix} X_{11} & X_{12} & \ldots & X_{1p} \\ X_{21} & X_{22} & \ldots & X_{2p} \\ \vdots & \vdots & & \vdots \\ X_{n1} & X_{n2} & \ldots & X_{np} \end{bmatrix}_{n\times p}$$

where X indicates the the $n\times p$ data matrix, X_j indicates variable $j, j = 1, \ldots, p$ and X_{ij} indicates the value j^{th} variable of the i^{th} observation. (It can be confusing to distinguish whether one is referring to the observation or a variable, because X_i is used to indicate observation also. When this is done it is usually accompanied by qualifying words such as **observation** X_3, or **variable** X_3.)

Having notation is helpful for concise explanations of different methods, to explain how data is scaled, processed and projected for various tasks, and how different quantities are calculated from the data.

When there is a response variable(s), it is common to consider X to be the predictors, and use Y to indicate the response variable(s). Y could be a matrix, also, and would be $n \times q$, where commonly $q = 1$. Y could be numeric or categorical, and this would change how it is handled with visualisation.

To make a low-dimensional projection (shadow) of the data, we need a projection matrix:

$$A_{p\times d} = \begin{bmatrix} A_{11} & A_{12} & \ldots & A_{1d} \\ A_{21} & A_{22} & \ldots & A_{2d} \\ \vdots & \vdots & & \vdots \\ A_{p1} & A_{p2} & \ldots & A_{pd} \end{bmatrix}_{p\times d}$$

A should be an orthonormal matrix, which means that the $\sum_{j=1}^{p} A_{jk}^2 = 1, k =$

$1, \ldots, d$ (columns represent vectors of length 1) and $\sum_{j=1}^{p} A_{jk}A_{jl} = 0, k, l = 1, \ldots, d; k \neq l$ (columns represent vectors that are orthogonal to each other). In matrix notation, this can be written as $A^\top A = I_d$.

Then the projected data is written as:

$$Y_{n\times d} = XA = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1d} \\ y_{21} & y_{22} & \cdots & y_{2d} \\ \vdots & \vdots & & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nd} \end{bmatrix}_{n\times d}$$

where $y_{ij} = \sum_{k=1}^{p} X_{ik}A_{kj}$. Note that we are using Y as the projected data here, as well as it possibly being used for a response variable. Where necessary, this will be clarified with words in the text, when notation is used in explanations later.

When using R, if we only have the data corresponding to X it makes sense to use a `matrix` object. However, if the response variable is included and it is categorical, then we might use a `data.frame` or a `tibble` which can accommodate non-numerical values. Then to work with the data, we can use the base R methods:

```
X <- matrix(c(1.1, 1.3, 1.4, 1.2,
              2.7, 2.6, 2.4, 2.5,
              3.5, 3.4, 3.2, 3.6),
            ncol=4, byrow=TRUE)
X
```

```
     [,1] [,2] [,3] [,4]
[1,]  1.1  1.3  1.4  1.2
[2,]  2.7  2.6  2.4  2.5
[3,]  3.5  3.4  3.2  3.6
```

which is a data matrix with $n = 3, p = 4$ and to extract a column (variable):

```
X[,2]
```

```
[1] 1.3 2.6 3.4
```

or a row (observation):

```
X[2,]
```

```
[1] 2.7 2.6 2.4 2.5
```

or an individual cell (value):

```
X[3,2]
```

```
[1] 3.4
```

To make a projection we need an orthonormal matrix:

```
A <- matrix(c(0.707,0.707,0,0,0,0,0.707,0.707), ncol=2, byrow=FALSE)
A
```

```
      [,1]  [,2]
[1,] 0.707 0.000
[2,] 0.707 0.000
[3,] 0.000 0.707
[4,] 0.000 0.707
```

You can check that it is orthonormal by

```
sum(A[,1]^2)
```

```
[1] 0.999698
```

```
sum(A[,1]*A[,2])
```

```
[1] 0
```

and make a projection using matrix multiplication:

```
X %*% A
```

```
       [,1]   [,2]
[1,] 1.6968 1.8382
[2,] 3.7471 3.4643
[3,] 4.8783 4.8076
```

The seemingly magical number `0.707` used above and to create the projection in Figure 1.1 arises from normalising a vector with equal contributions from each variable, `(1, 1)`. Dividing by `sqrt(2)` gives `(0.707, 0.707)`.

The notation convention used throughout the book is:

`n` = number of observations `p` = number of variables, dimension of data `d` = dimension of the projection `g` = number of groups, in classification `X` = data matrix

Exercises

1. Generate a matrix A with $p = 5$ (rows) and $d = 2$ (columns), where each value is randomly drawn from a standard normal distribution. Extract the element at row 3 and column 1.
2. We will interpret A as a projection matrix and therefore it needs to be orthonormalised. Use the function `tourr::orthonormalise` to do this, and explicitly check that each column is normalised and that the two columns are orthogonal now. Which dimensions contribute most to the projection for your A?
3. Use matrix multiplication to calculate the projection of the `mulgar::clusters` data onto the 2D plane defined by A. Make a scatterplot of the projected data. Can you identify clustering in this view?

Part II

Dimension reduction

3

Overview

This chapter will focus on methods for reducing dimension, and how the tour can be used to assist with the common methods such as principal component analysis (PCA), multidimensional scaling (MDS), t-stochastic neighbour embedding (t-SNE), and factor analysis.

Dimension is perceived in a tour using the spread of points. When the points are spread far apart, then the data is filling the space. Conversely when the points "collapse" into a sub-region then the data is only partially filling the space, and some dimension reduction to reduce to this smaller dimensional space may be worthwhile.

When points do not fill the plotting canvas fully, it means that it lives in a lower dimension. This low-dimensional space might be linear or non-linear, with the latter being much harder to define and capture.

Let's start with some 2D examples. You need at least two variables to be able to talk about association between variables. Figure 3.1 shows three plots of two variables. Plot (a) shows two variables that are strongly linearly associated[1], because when `x1` is low, `x2` is low also, and conversely when `x1` is high, `x2` is also high. This can also be seen by the reducton in spread of points (or "collapse") in one direction making the data fill less than the full square of the plot. *So from this we can conclude that the data is not fully 2D.* The second step is to infer which variables contribute to this reduction in dimension. The axes for `x1` and `x2` are drawn extending from $(0, 0)$ and because they both extend out of the cloud of points, in the direction away from the collapse of points we can say that they are jointly responsible for the dimension reduction.

Figure 3.1 (b) shows a pair of variables that are **not** linearly associated. Variable `x1` is more varied than `x3` but knowing the value on `x1` tells us nothing about possible values on `x3`. Before running a tour all variables are typically scaled to have equal spread. The purpose of the tour is to capture association and

[1]It is generally better to use *associated* than *correlated*. Correlation is a statistical quantity, measuring linear association. The term *associated* can be prefaced with the type of association, such as *linear* or *non-linear*.

relationships between the variables, so any univariate differences should be removed ahead of time. Figure 3.1 (c) shows what this would look like when `x3` is scaled - the points are fully spread in the full square of the plot.

```
library(tibble)
set.seed(6045)
x1 <- runif(123)
x2 <- x1 + rnorm(123, sd=0.1)
x3 <- rnorm(123, sd=0.2)
df <- tibble(x1 = (x1-mean(x1))/sd(x1),
             x2 = (x2-mean(x2))/sd(x2),
             x3,
             x3scaled = (x3-mean(x3))/sd(x3))
```

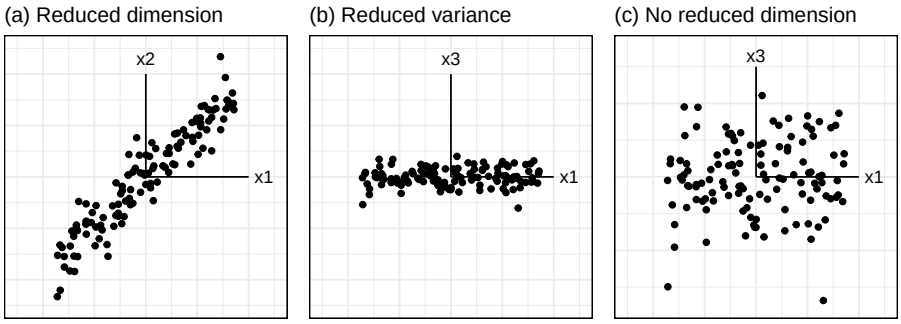


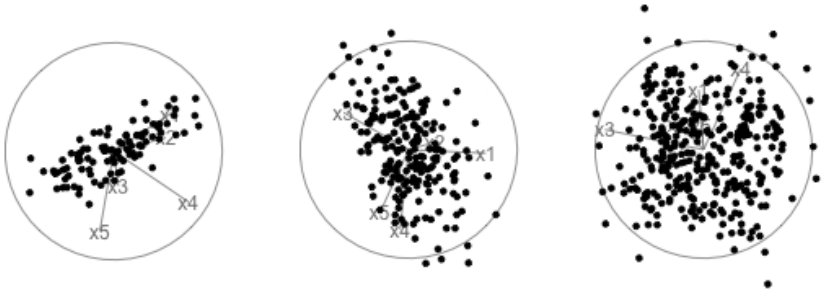
Figure 3.1: Explanation of how dimension reduction is perceived in 2D, relative to variables: (a) Two variables with strong linear association. Both variables contribute to the association, as indicated by their axes extending out from the 'collapsed' direction of the points; (b) Two variables with no linear association. But x3 has less variation, so points collapse in this direction; (c) The situation in plot (b) does not arise in a tour because all variables are (usually) scaled. When an axes extends out of a direction where the points are collapsed, it means that this variable is partially responsible for the reduced dimension.

Now let's think about what this looks like with five variables. Figure 3.2 shows a grand tour on five variables, with (a) data that is primarily 2D, (b) data that is primarily 3D and (c) fully 5D data. You can see that both (a) and (b) the spread of points collapse in some projections, with it happening more in (a). In (c) the data is always spread out in the square, although it does seem to concentrate or pile in the centre. This piling is typical when projecting from high dimensions to low dimensions. The sage tour (Laa et al., 2020a) makes a correction for this.

```
library(mulgar)
data(plane)
data(box)
render_gif(plane,
           grand_tour(),
           display_xy(),
           gif_file="gifs/plane.gif",
           frames=500,
           width=200,
           height=200)
render_gif(box,
           grand_tour(),
           display_xy(),
           gif_file="gifs/box.gif",
           frames=500,
           width=200,
           height=200)
# Simulate full cube
library(geozoo)
cube5d <- data.frame(cube.solid.random(p=5, n=300)$points)
colnames(cube5d) <- paste0("x", 1:5)
cube5d <- data.frame(apply(cube5d, 2, function(x) (x-mean(x))/sd(x)))
render_gif(cube5d,
           grand_tour(),
           display_xy(),
           gif_file="gifs/cube5d.gif",
           frames=500,
           width=200,
           height=200)
```

The next step is to determine which variables contribute. In the examples just provided, all variables are linearly associated in the 2D and 3D data. You can check this by making a scatterplot matrix, Figure 3.3.

```
library(GGally)
library(mulgar)
data(plane)
ggscatmat(plane) +
  theme(panel.background =
          element_rect(colour="black", fill=NA),
    axis.text = element_blank(),
    axis.ticks = element_blank())
```



(a) 2D plane in 5D (b) 3D plane in 5D (c) 5D plane in 5D

Figure 3.2: Single frames from different dimensional planes - 2D, 3D, 5D - displayed in a grand tour projecting into 2D. Notice that the 5D in 5D always fills out the box (although it does concentrate some in the middle which is typical when projecting from high to low dimensions). Also you can see that the 2D in 5D, concentrates into a line more than the 3D in 5D. This suggests that it is lower dimensional. (Animations can be viewed here.)

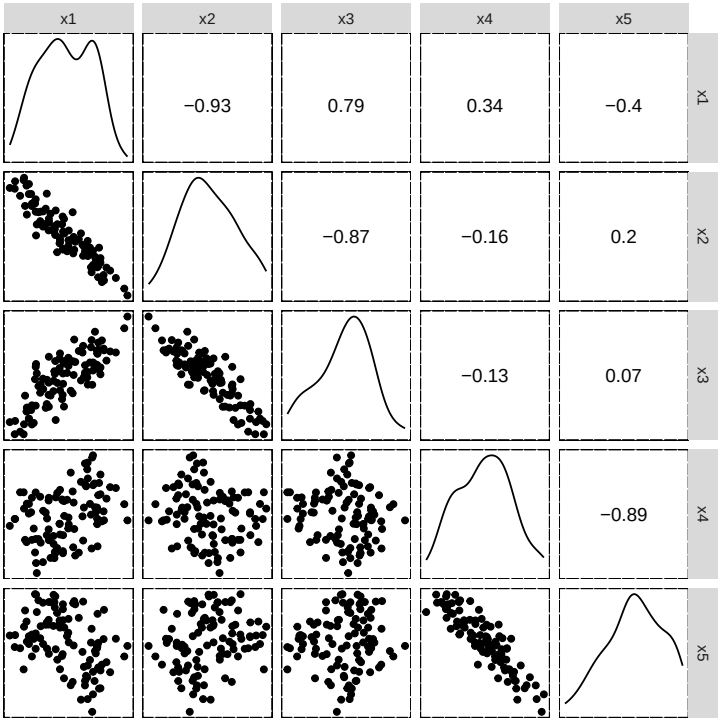


Figure 3.3: Scatterplot matrix of plane data. You can see that x1-x3 are strongly linearly associated, and also x4 and x5. When you watch the tour of this data, any time the data collapses into a line you should see only (x1, x2, x3) or (x4, x5). When combinations of x1 and x4 or x5 show, the data should be spread out.

To make an example where not all variables contribute, we have added two additional variables to the `plane` data set, which are purely noise.

```
# Add two pure noise dimensions to the plane
plane_noise <- plane
plane_noise$x6 <- rnorm(100)
plane_noise$x7 <- rnorm(100)
plane_noise <- data.frame(apply(plane_noise, 2,
    function(x) (x-mean(x))/sd(x)))
ggduo(plane_noise, columnsX = 1:5, columnsY = 6:7,
    types = list(continuous = "points")) +
  theme(aspect.ratio=1,
    panel.background =
          element_rect(colour="black", fill=NA),
    axis.text = element_blank(),
    axis.ticks = element_blank())
```

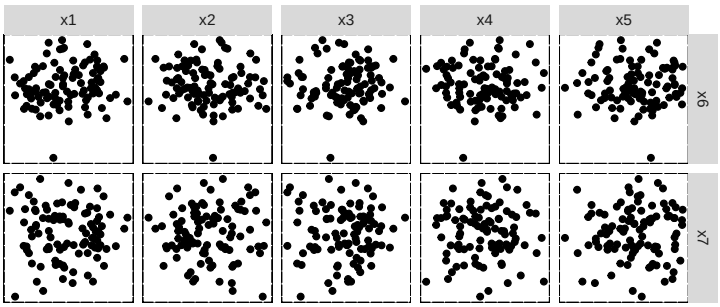


Figure 3.4: Scatterplots showing two additional noise variables that are not associated with any of the first five variables.

Now we have 2D structure in 7D, but only five of the variables contribute to the 2D structure, that is, five of the variables are linearly related with each other. The other two variables (x6, x7) are not linearly related to any of the others.

The data is viewed with a grand tour in Figure 3.5. We can still see the concentration of points along a line in some dimensions, which tells us that the data is not fully 7D. Then if you look closely at the variable axes you will see that the collapsing to a line only occurs when any of x1-x5 contribute strongly in the direction orthogonal to this. This does not happen when x6 or x7 contribute strongly to a projection - the data is always expanded to fill much of the space. That tells us that x6 and x7 don't substantially contribute to the dimension reduction, that is, they are not linearly related to the other variables.

```
library(ggplot2)
library(plotly)
library(htmlwidgets)

set.seed(78)
b <- basis_random(7, 2)
pn_t <- tourr::save_history(plane_noise,
                    tour_path = grand_tour(),
                    start = b,
                    max_bases = 8)
pn_t <- interpolate(pn_t, 0.1)
pn_anim <- render_anim(plane_noise,
                         frames=pn_t)

pn_gp <- ggplot() +
     geom_path(data=pn_anim$circle,
               aes(x=c1, y=c2,
                   frame=frame), linewidth=0.1) +
     geom_segment(data=pn_anim$axes,
                  aes(x=x1, y=y1,
                      xend=x2, yend=y2,
                      frame=frame),
                  linewidth=0.1) +
     geom_text(data=pn_anim$axes,
               aes(x=x2, y=y2,
                   frame=frame,
                   label=axis_labels),
               size=5) +
     geom_point(data=pn_anim$frames,
                aes(x=P1, y=P2,
                    frame=frame),
                alpha=0.8) +
     xlim(-1,1) + ylim(-1,1) +
     coord_equal() +
     theme_bw() +
     theme(axis.text=element_blank(),
         axis.title=element_blank(),
         axis.ticks=element_blank(),
         panel.grid=element_blank())
pn_tour <- ggplotly(pn_gp,
                        width=500,
                        height=550) %>%
       animation_button(label="Go") %>%
       animation_slider(len=0.8, x=0.5,
```

```
                        xanchor="center") %>%
        animation_opts(easing="linear",
                       transition = 0)

htmlwidgets::saveWidget(pn_tour,
           file="html/plane_noise.html",
           selfcontained = TRUE)
```

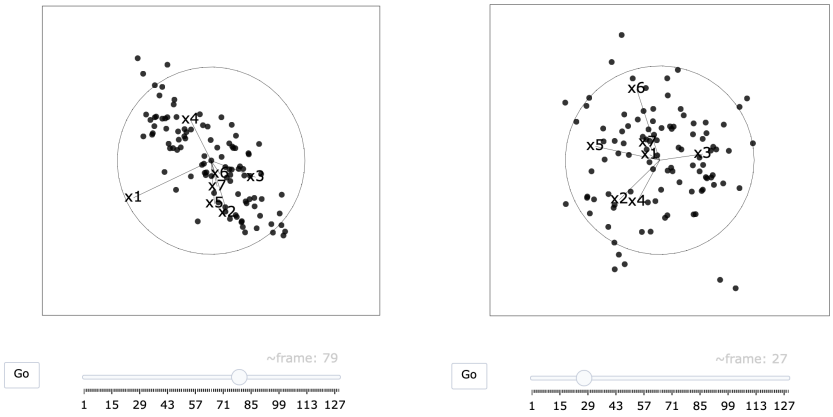


Figure 3.5: Two frames from a grand tour of the plane with two additional dimensions of pure noise (can be viewed here). The collapsing of the points indicates that this is not fully 7D. This only happens when any of x1-x5 are contributing strongly (frame 49 x4, x5; frame 79 x1; frame 115 x2, x3). If x6 or x7 are contributing strongly the data is spread out fully (frames 27, 96). This tells us that x6 and x7 are not linearly associated, but other variables are.

To determine which variables are responsible for the reduced dimension look for the axes that extend out of the point cloud. These contribute to smaller variation in the observations, and thus indicate dimension reduction.

The simulated data here is very simple, and what we have learned from the tour could also be learned from principal component analysis. However, if there are small complications, such as outliers or nonlinear relationships, that might not be visible from principal component analysis, the tour can help you to see them.

Figure 3.6 and Figure 3.7(a) show example data with an outlier and Figure 3.7(b) shows data with non-linear relationships.

```
# Add several outliers to the plane_noise data
plane_noise_outliers <- plane_noise
plane_noise_outliers[101,] <- c(2, 2, -2, 0, 0, 0, 0)
plane_noise_outliers[102,] <- c(0, 0, 0,-2, -2, 0, 0)

ggscatmat(plane_noise_outliers, columns = 1:5) +
  theme(aspect.ratio=1,
    panel.background =
          element_rect(colour="black", fill=NA),
    axis.text = element_blank(),
    axis.ticks = element_blank())
```

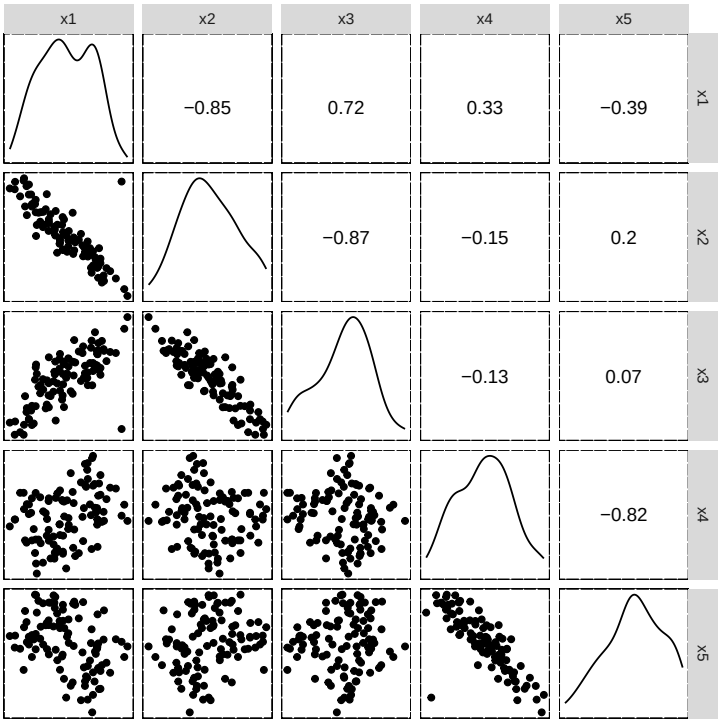
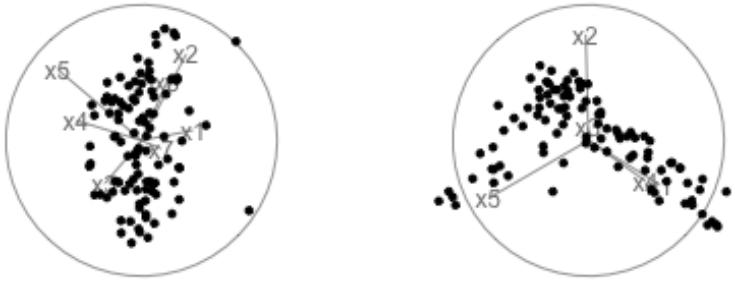


Figure 3.6: Scatterplot matrix of the plane with noise data, with two added outliers in variables with strong correlation.

```
render_gif(plane_noise_outliers,
           grand_tour(),
           display_xy(),
           gif_file="gifs/pn_outliers.gif",
```

```
            frames=500,
            width=200,
            height=200)

data(plane_nonlin)
render_gif(plane_nonlin,
            grand_tour(),
            display_xy(),
            gif_file="gifs/plane_nonlin.gif",
            frames=500,
            width=200,
            height=200)
```



(a) Outliers (b) Non-linear relationship

Figure 3.7: Two frames from tours of examples of different types of dimensionality issues: outliers (a) and non-linearity (b). In (a) you can see two points far from the others in the projection. During a tour the two can be seen with different movement patterns – moving faster and in different directions than other points. Outliers will affect detection of reduced dimension, but they can be ignored when assessing dimensionality with the tour. In (b) there is a non-linear relationship between several variables, primarily with x3. Non-linear relationships may not be easily captured by other techniques but are often visible with the tour. (The tours can be viewed here.)

Exercises

1. Multicollinearity is when the predictors for a model are strongly linearly associated. It can adversely affect the fitting of most models, because many possible models may be equally as good. Variable importance might be masked by correlated variables, and confidence intervals generated for linear models might be too wide. Check the for multicollinearity or other associations between the predictors in:

a. 2001 Australian election data
b. 2016 Australian election data

2. Examine 5D multivariate normal samples drawn from populations with a range of variance-covariance matrices. (You can use the `mvtnorm` package to do the sampling, for example.) Examine the data using a grand tour. What changes when you change the correlation from close to zero to close to 1? Can you see a difference between strong positive correlation and strong negative correlation?

4

Principal component analysis

Reducing dimensionality using principal component analysis (PCA) dates back to Pearson (1901) and Hotelling (1933), and Jolliffe & Cadima (2016) provides a current overview. The goal is to find a smaller set of variables, $q(< p)$, that contain as much information as the original as possible. The new set of variables, known as principal components (PCs), are linear combinations of the original variables. The PCs can be used to represent the data in a lower-dimensional space.

The process is essentially an optimisation procedure, although PCA has an analytical solution. It solves the problem of

$$\max_{a_k} \ \text{Var}(Xa_k),$$

where X is the $n \times p$ data matrix, $a_k (k = 1, ..., p)$ is a 1D projection vector, called an eigenvector, and the $\text{Var}(Xa_k)$ is called an eigenvalue. So PCA is a sequential process, that will find the direction in the high-dimensional space (as given by the first eigenvector) where the data is most varied, and then find the second most varied direction, and so on. The eigenvectors define the combination of the original variables, and the eigenvalues define the amount of variance explained by the reduced number of variables.

PCA is very broadly useful for summarising linear association by using combinations of variables that are highly correlated. However, high correlation can also occur when there are outliers, or clustering. PCA is commonly used to detect these patterns also.

With visualisation we want to assess whether it is appropriate to use PCA to summarise any linear association by using combinations of variables that are highly correlated. It can help to detect other patterns that might affect the PCA results such as outliers, clustering or non-linear dependence.

PCA is not very effective when the distribution of the variables is highly skewed, so it can be helpful to transform variables to make them more symmetrically distributed before conducting PCA. It is also possible to summarise different

types of structure by generalising the optimisation criteria to any function of projected data, $f(XA)$, which is called *projection pursuit* (PP). PP has a long history (Kruskal (1964a), Friedman & Tukey (1974), Diaconis & Freedman (1984b), Jones & Sibson (1987), Huber (1985)), and there are regularly new developments (e.g. E.-K. Lee & Cook (2009), Perisic & Posse (2005), Y. D. Lee et al. (2013), Loperfido (2018), Bickel et al. (2018), C. Zhang et al. (2023)).

4.1 Determining how many dimensions

We would start by examining the data using a grand tour. The goal is to check whether there might be potential issues for PCA, such as skewness, outliers or clustering, or even non-linear dependencies.

We'll start be showing PCA on the simulated data from Chapter 3. The scree plots show that PCA supports that the data are 2D, 3D and 5D respectively.

```
library(dplyr)
library(ggplot2)
library(mulgar)
data(plane)
data(box)
library(geozoo)
cube5d <- data.frame(cube.solid.random(p=5, n=300)$points)
colnames(cube5d) <- paste0("x", 1:5)
cube5d <- data.frame(apply(cube5d, 2, function(x) (x-mean(x))/sd(x)))
p_pca <- prcomp(plane)
b_pca <- prcomp(box)
c_pca <- prcomp(cube5d)
p_scree <- ggscree(p_pca, q = 5) + theme_minimal()

b_scree <- ggscree(b_pca, q = 5) + theme_minimal()
c_scree <- ggscree(c_pca, q = 5) + theme_minimal()
```

The next step is to look at the coefficients for the selected number of PCs. Table 4.1 shows the coefficients for the first two PCs of the `plane` data. All five variables contribute, with `x1`, `x2`, `x3` contributing more to `PC1`, and `x4`, `x5` contributing more to `PC2`. Table 4.2 shows the coefficients for the first three PCs. Variables `x1`, `x2`, `x3` contribute strongly to `PC1`, `PC2` has contributions from all variables except `x3` and variables `x4` and `x5` contribute strongly to `PC3`.

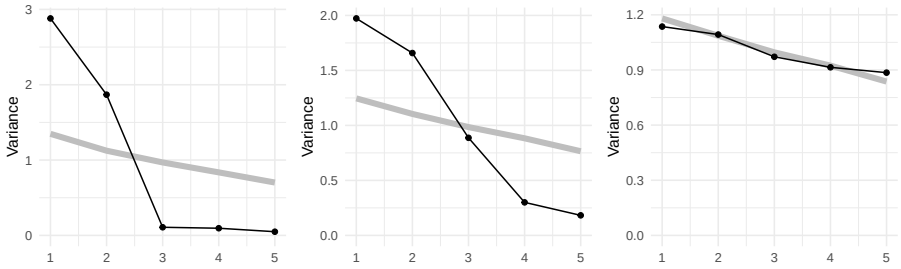


Figure 4.1: Scree plots for the three simulated data sets shown in Figure 3.2. The 2D in 5D is clearly recognised by PCA to be 2D because the variance drops substantially between 2-3 principal components. The 3D in 5D is possibly 3D because the variance drops from 3-4 principal components. The fully 5D data has no drop in variance, and all values are close to the typical value one would observe if the data was fully 5D.

```
library(gt)
p_pca$rotation[,1:2] %>%
  as_tibble(rownames="Variable") %>%
  gt() %>%
  fmt_number(columns = c(PC1, PC2),
             decimals = 2)
```

Table 4.1: Coefficients for the first two PCs for the plane data.

Variable	PC1	PC2
x1	0.58	−0.06
x2	−0.55	0.21
x3	0.47	−0.41
x4	0.25	0.64
x5	−0.29	−0.62

```
b_pca$rotation[,1:3] %>%
  as_tibble(rownames="Variable") %>%
  gt() %>%
  fmt_number(columns = c(PC1, PC2, PC3),
             decimals = 2)
```

Table 4.2: Coefficients for the first three PCs for the box data.

Variable	PC1	PC2	PC3
x1	−0.51	0.46	0.11
x2	0.51	0.46	0.00
x3	−0.65	−0.09	0.23
x4	−0.22	0.36	−0.87
x5	0.02	0.66	0.43

In each of these simulated data sets, all five variables contributed to the dimension reduction. If we added two purely noise variables to the plane data, as done in Chapter 3, the scree plot would indicate that the data is now 4D, and we would get a different interpretation of the coefficients from the PCA. We see that `PC1` and `PC2` are approximately the same as before, with main variables being (`x1`, `x2`, `x3`) and (`x4`, `x5`) respectively. `PC3` and `PC4` are both `x6` and `x7`.

```
set.seed(5143)
plane_noise <- plane
plane_noise$x6 <- rnorm(100)
plane_noise$x7 <- rnorm(100)
plane_noise <- data.frame(apply(plane_noise, 2, function(x) (x-mean(x))/sd(x)))

pn_pca <- prcomp(plane_noise)
ggscree(pn_pca, q = 7) + theme_minimal()
```

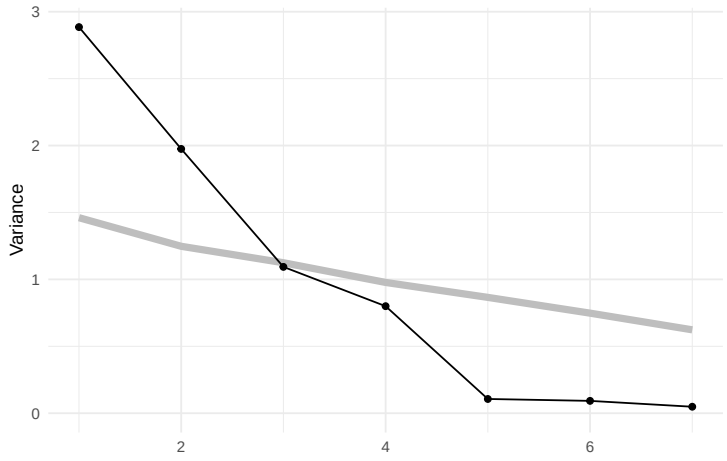


Figure 4.2: Additional noise variables expands the data to 4D.

```
pn_pca$rotation[,1:4] %>%
  as_tibble(rownames="Variable") %>%
  gt() %>%
  fmt_number(columns = c(PC1, PC2, PC3, PC4),
             decimals = 2)
```

Table 4.3: Coefficients for the first four PCs for the box data.

Variable	PC1	PC2	PC3	PC4
x1	0.58	0.04	0.01	0.00
x2	−0.55	−0.18	−0.03	0.07
x3	0.47	0.37	0.05	−0.20
x4	0.24	−0.62	−0.06	0.17
x5	−0.28	0.60	0.07	−0.14
x6	0.05	0.29	−0.58	0.76
x7	−0.02	−0.08	−0.81	−0.58

4.1.1 Example: pisa

The `pisa` data contains simulated data from math, reading and science scores, totalling 30 variables. PCA is used here to examine the association. We might expect that it is 3D, but what we see suggests it is primarily 1D. This means that a student that scores well in math, will also score well in reading and science.

```
data(pisa)
pisa_std <- pisa %>%
  filter(CNT == "Australia") %>%
  select(-CNT) %>%
  mutate_all(mulgar:::scale2)
pisa_pca <- prcomp(pisa_std)
pisa_scree <- ggscree(pisa_pca, q = 15) + theme_minimal()
```

The scree plot in **?@fig-pisa-scree** shows a big drop from one to two PCs in the amount of variance explained. A grand tour on the 30 variables can be run using `animate_xy()`:

```
animate_xy(pisa_std, half_range=1)
```

or rendered as an animated gif using `render_gif()`:

```
render_gif(pisa_std,
           grand_tour(),
           display_xy(half_range=0.9),
           gif_file="gifs/pisa_gt.gif",
           frames=500,
           width=400,
           height=400,
           loop=FALSE)
```

and we can see that the data is elliptical in most projections, sometimes shrinking to be a small circle. This pattern strongly indicates that there is one primary direction of variation in the data, with only small variation in any direction away from it. Shrinking to the small circle is analogous to to how *a pencil or cigar or water bottle in 3D looks from some angles.*

The coefficients of the first PC (first eigenvector) are roughly equal in magnitude (as shown below), which tells us that all variables roughly contribute. Interestingly, they are all negative, which is not actually meaningful. With different software these could easily have been all positive. The sign of the coefficients can be reversed, as long as all are reversed, which is the same as an arrow pointing one way, changing and pointing the other way.

```
round(pisa_pca$rotation[,1], 2)
```

```
PV1MATH  PV2MATH  PV3MATH  PV4MATH  PV5MATH  PV6MATH
  -0.18    -0.18    -0.18    -0.18    -0.18    -0.18
PV7MATH  PV8MATH  PV9MATH PV10MATH  PV1READ  PV2READ
  -0.18    -0.18    -0.18    -0.18    -0.19    -0.18
PV3READ  PV4READ  PV5READ  PV6READ  PV7READ  PV8READ
  -0.19    -0.19    -0.19    -0.19    -0.19    -0.19
PV9READ PV10READ  PV1SCIE  PV2SCIE  PV3SCIE  PV4SCIE
  -0.19    -0.19    -0.18    -0.18    -0.19    -0.18
PV5SCIE  PV6SCIE  PV7SCIE  PV8SCIE  PV9SCIE PV10SCIE
  -0.19    -0.18    -0.19    -0.18    -0.19    -0.18
```

The tour verifies that the ‘pisa‘ data is primarily 1D, indicating that a student who scores well in math, probably scores well in reading and science, too. More interestingly, the regular shape of the data strongly indicates that it is "synthetic", simulated rather than observed.

4.1.2 Example: aflw

This data has player statistics for all the matches in the 2021 season. We would be interested to know which variables contain similar information, and thus might be combined into single variables. We would expect that many statistics to group into a few small sets, such as offensive and defensive skills. We might also expect that some of the statistics are skewed, most players have low values and just a handful of players are stellar. It is also possible that there are some extreme values. These are interesting features, but they will distract from the main purpose of grouping the statistics. Thus the tour is used to check for potential problems with the data prior to conducting PCA.

```
library(tourr)
data(aflw)
aflw_std <- aflw %>%
  mutate_if(is.numeric, function(x) (x-
      mean(x, na.rm=TRUE))/
      sd(x, na.rm=TRUE))
```

To look at all of the 29 player statistics in a grand tour.

```
animate_xy(aflw_std[,7:35], half_range=0.9)
render_gif(aflw_std[,7:35],
           grand_tour(),
           display_xy(half_range=0.9),
           gif_file="gifs/aflw_gt.gif",
           frames=500,
           loop=FALSE)
```

No major surprises! There is a small amount of skewness, and there are no major outliers. Skewness indicates that most players have reasonably similar skills (bunching of points), except for some key players (the moderate outliers). The skewness could be reduced by applying a log or square root transformation to some variables prior to running the PCA. However, we elect not to do this because the moderate outliers are of interest. These correspond to talented players that we'd like to explore further with the analysis.

Below we have the conventional summary of the PCA, a scree plot showing the reduction in variance to be explained when each additional PC is considered. It is also conventional to look at a table summarising the proportions of variance explained by PCs, but with almost 30 variables it is easier to make some decision on the number of PCs needed based on the scree plot.

```
aflw_pca <- prcomp(aflw_std[,7:35],
                   scale = FALSE,
                   retx=TRUE)

ggscree(aflw_pca, q = 29) + theme_minimal()
```

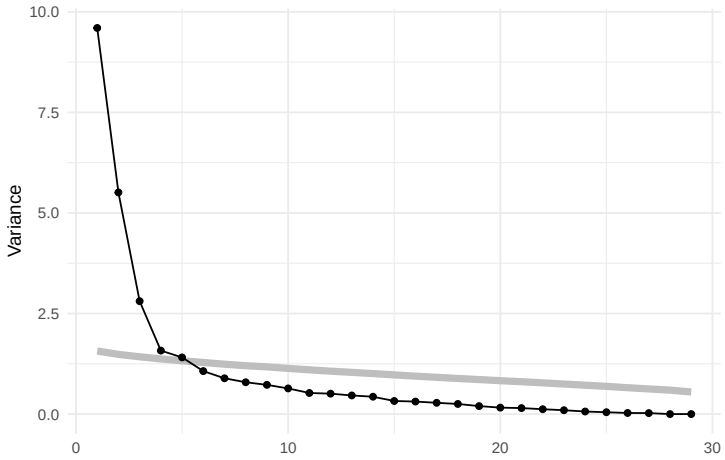


Figure 4.3: Scree plot showing decay in variance of PCs.

From the scree plot in Figure 4.3, we see a sharp drop from one to two, two to three and then smaller drops. After four PCs the variance drops again at six PCs and then gradually decays. We will choose four PCs to examine more closely. This explains 67.2% of the variance.

```
library(gt)
aflw_pca$rotation[,1:4] %>%
  as_tibble(rownames="Variable") %>%
  arrange(desc(PC1), desc(PC2), desc(PC3)) %>%
  gt() %>%
  fmt_number(columns = c(PC1, PC2, PC3, PC4),
             decimals = 2)
```

Table 4.4: Coefficients for the first four PCs.

Variable	PC1	PC2	PC3	PC4
disposals	0.31	−0.05	−0.03	0.07
possessions	0.31	−0.03	−0.07	0.09
kicks	0.29	−0.04	0.09	−0.12

metres	0.28	−0.03	0.10	−0.15
contested	0.28	0.01	−0.12	0.23
uncontested	0.28	−0.06	−0.01	−0.05
turnovers	0.27	−0.01	−0.01	−0.29
clearances	0.23	0.00	−0.29	0.19
clangers	0.23	−0.02	−0.06	−0.33
handballs	0.23	−0.04	−0.19	0.31
frees_for	0.21	0.02	−0.13	0.18
marks	0.21	0.03	0.32	0.02
tackles	0.20	0.01	−0.28	0.09
time_pct	0.16	−0.04	0.35	−0.02
intercepts	0.13	−0.28	0.24	0.03
rebounds_in50	0.13	−0.28	0.24	−0.06
frees_against	0.13	0.03	−0.16	−0.23
assists	0.09	0.23	0.00	0.05
bounces	0.09	0.03	0.02	−0.28
behinds	0.09	0.32	0.08	−0.02
shots	0.08	0.38	0.12	−0.03
tackles_in50	0.07	0.27	−0.18	0.03
marks_in50	0.06	0.34	0.18	0.04
contested_marks	0.05	0.16	0.34	0.15
goals	0.04	0.37	0.16	0.03
accuracy	0.04	0.34	0.10	0.06
one_pct	0.03	−0.21	0.33	0.08
disposal	0.02	−0.13	0.20	0.50
hitouts	−0.04	0.00	−0.03	0.32

When there are as many variables as this, it can be hard to digest the combinations of variables most contributing to each PC. Rearranging the table by sorting on a selected PC can help. Table 4.4 has been sorted according to the PC 1 coefficients.

PC 1 is primarily composed of `disposals`, `possessions`, `kicks`, `metres`, `uncontested`, `contested`, ... Actually almost all variables positively contribute, albeit in different amounts! It is quite common in PCA for the first PC to be a combination of all variables, although it might commonly be a closer to equal contribution, and it tells us that there is one main direction of variation in the data. For PC 1 in the `aflw` data, PCA is telling us that the primary variation is through a combination of skills, and this maps to basic football playing skills, where some skills (e.g. disposals, possessions, kicks, ...) are more important.

Thus the second PC might be the more interesting. PC 2 is primarily a combination of `shots`, `goals`, `marks_in50`, `accuracy`, and `behinds` contrasted against `rebounds_in50` and `intercepts`. The negative coefficients are primary

offensive skills and the positive coefficients are defensive skills. This PC is reasonable measure of the offensive vs defensive skills of a player.

We would continue to interpret each PC by examining large coefficients to help decide how many PCs are a suitable summary of the information in the data. Briefly, PC 3 is a measure of worth of the player because `time_pct` has a large coefficient, so players that are on the field longer will contribute strongly to this new variable. It also has large (and opposite) contributions from `clearances`, `tackles`, `contested_marks`. PC 4 appears to be related to aggressive play with `clangers`, `turnovers`, `bounces` and `frees_against` featuring. So all four PCs have useful information. (Note, if we had continued to examine large coefficients on PC 5 we would find that all variables already have had reasonably large coefficients on PC 1-4, which supports restricting attention to the first four.)

Ideally, when we tour the four PCs, we'd like to be able to stop and identify players. This involves creating a pre-computed animation, with additional mouse-over. This is only feasible with a small number of observations, like the `aflw` data, because all of the animation frames are constructed in a single object and passed to `plotly`. This object gets large very quickly!

```
library(plotly)
library(htmlwidgets)
set.seed(20)
b <- basis_random(4, 2)
aflw_pct <- tourr::save_history(aflw_pca$x[,1:4],
                    tour_path = grand_tour(),
                    start = b,
                    max_bases = 5)
# To reconstruct projected data plots, later
save(aflw_pct, file="data/aflw_pct.rda")
aflw_pcti <- interpolate(aflw_pct, 0.1)
aflw_anim <- render_anim(aflw_pca$x[,1:4],
                         frames=aflw_pcti,
             obs_labels=paste0(aflw$surname,
                               aflw$given_name))

aflw_gp <- ggplot() +
     geom_path(data=aflw_anim$circle,
               aes(x=c1, y=c2,
                   frame=frame), linewidth=0.1) +
     geom_segment(data=aflw_anim$axes,
                  aes(x=x1, y=y1,
                      xend=x2, yend=y2,
                      frame=frame),
```

```
                  linewidth=0.1) +
      geom_text(data=aflw_anim$axes,
                aes(x=x2, y=y2,
                    frame=frame,
                    label=axis_labels),
                size=5) +
      geom_point(data=aflw_anim$frames,
                 aes(x=P1, y=P2,
                     frame=frame,
                     label=obs_labels),
                 alpha=0.8) +
      xlim(-1,1) + ylim(-1,1) +
      coord_equal() +
      theme_bw() +
      theme(axis.text=element_blank(),
          axis.title=element_blank(),
          axis.ticks=element_blank(),
          panel.grid=element_blank())
aflw_pctour <- ggplotly(aflw_gp,
                        width=500,
                        height=550) %>%
        animation_button(label="Go") %>%
        animation_slider(len=0.8, x=0.5,
                         xanchor="center") %>%
        animation_opts(easing="linear", transition = 0)

htmlwidgets::saveWidget(aflw_pctour,
           file="html/aflw_pca.html",
           selfcontained = TRUE)
```

From **?@fig-aflw-pcatour** the shape of the four PCs is similar to that of all the variables, bunching of points in the centre with a lot of moderate outliers.

```
library(plotly)
load("data/aflw_pct.rda")
aflw_pcti <- interpolate(aflw_pct, 0.1)
f18 <- matrix(aflw_pcti[,,18], ncol=2)
p18 <- render_proj(aflw_pca$x[,1:4], f18,
                   obs_labels=paste0(aflw$surname,
                                     aflw$given_name))
pg18 <- ggplot() +
  geom_path(data=p18$circle, aes(x=c1, y=c2)) +
  geom_segment(data=p18$axes, aes(x=x1, y=y1, xend=x2, yend=y2)) +
```

```
  geom_text(data=p18$axes, aes(x=x2, y=y2, label=rownames(p18$axes))) +
  geom_point(data=p18$data_prj, aes(x=P1, y=P2, label=obs_labels)) +
  xlim(-1,1) + ylim(-1, 1) +
  #ggtitle("Frame 18") +
  theme_bw() +
  theme(
    axis.text=element_blank(),
    axis.title=element_blank(),
    axis.ticks=element_blank(),
    panel.grid=element_blank())
ggplotly(pg18, width=500, height=500)
```

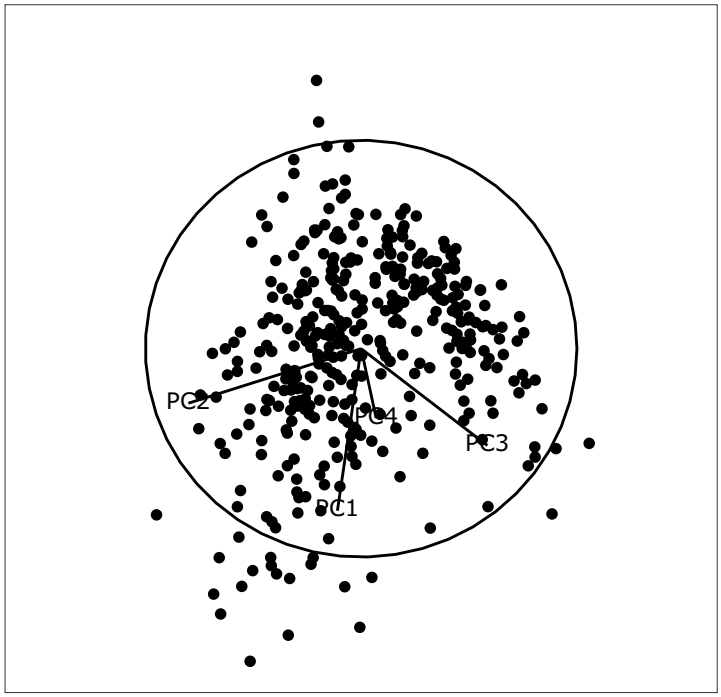


Figure 4.4: Frame 18 re-plotted so that players can be identified on mouse-over.

For any particular frame, like 18 re-plotted in Figure 4.4, we can investigate further. Here there is a branching pattern, where the branch points in the direction of PC 1. Mouse-over the players at the tip of this branch and we find players like Alyce Parker, Brittany Bonnici, Dana Hooker, Kiara Bowers. If you look up the bios of these players you'll find they all have generally good player descriptions like "elite disposals", "powerful left foot", "hard-running midfielder", "best and fairest".

In the direction of PC 2, you'll find players like Lauren Ahrens, Stacey Livingstone who are star defenders. Players in this end of PC 1, have high scores on `intercepts` and `rebounds_in50`.

Another interesting frame for inspecting PC 2 is 59. PC 2 at one end has players with high goal scoring skills, and the other good defending skills. So mousing over the other end of PC 2 finds players like Gemma Houghton and Katie Brennan who are known for their goal scoring. The branch pattern is an interesting one, because it tells us there is some combination of skills that are lacking among all players, primarily this appears to be there some distinction between defenders skills and general playing skills. It's not as simple as this because the branching is only visible when PC 1 and PC 2 are examined with PC 3.

PCA is useful for getting a sense of the variation in a high-dimensional data set. Interpreting the principal components is often useful, but it can be discombobulating. For the `aflw` data it would be good to think about it as a guide to the main directions of variation and to follow with a more direct engineering of variables into interesting player characteristics. For example, calculate offensive skill as an equal combination of goals, accuracy, shots, behinds. A set of new variables specifically computed to measure particular skills would make explaining an analysis easier.

4.2 Examining the PCA model in the data space

When you choose a smaller number of PCs (k) than the number of original variables, this is essentially producing a model for the data. The model is the lower dimensional k-D space. It is analogous to a linear regression model, except that the residuals from the model are $(p-k)$-D.

It is common to show the model, that is the data projected into the k-D model space. When $k = 2$ this is called a "biplot". For the `plane` and `plane_noise` data the biplots are shown in Figure 4.5. This is useful for checking which variables contribute most to the new principal component variables, and also to check for any problems that might have affected the fit, such as outliers, clusters or non-linearity. Interestingly, biplots are typically only made in 2D, even if the data should be summarised by more than two PCs. Occasionally you will see the biplot made for PC j vs PC k also. With the `pca_tour()` function in the `tourr` package you can view a k-D biplot. This will display the k PCs with the axes displaying the original variables, and thus see their contribution to the PCs.

```
library(ggfortify)
library(patchwork)
plane_pca <- prcomp(plane)
pl1 <- autoplot(plane_pca, loadings = TRUE,
         loadings.label = TRUE) +
  ggtitle("(a)") +
  theme_minimal() +
  theme(aspect.ratio=1)
plane_noise_pca <- prcomp(plane_noise)
pl2 <- autoplot(plane_noise_pca, loadings = TRUE,
         loadings.label = TRUE) +
  ggtitle("(b)") +
  theme_minimal() +
  theme(aspect.ratio=1)
pl1 + pl2
```

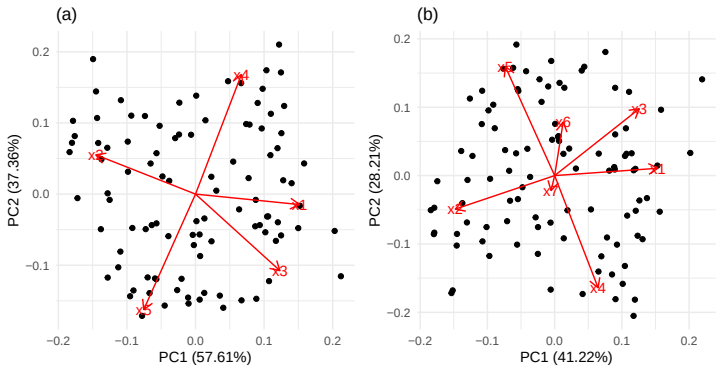


Figure 4.5: Biplots of the plane (a) and plane + noise (b) data. All five variables contribute strongly to the two principal components in (a): PC1 is primarily `x1`, `x2` and `x3` and PC2 is primarily `x4` and `x5`. In (b) the same four variables contribute in almost the same way, with variables `x6` and `x7` contributing very little. The data was constructed this way, that these two dimensions were purely noise.

It can be useful to examine this model using the tour. The model is simply a plane in high dimensions. This would be considered to be the model in the data space. The reason to do this is to check how well the model fits the data. The plane corresponding to the model should be oriented along the main direction of the points, and the spread of points around the plane should be small. We should also be able to see if there has been any strong non-linear relationship missed by the model, or outliers and clusters.

The function `pca_model()` from the `mulgar` package can be used to represent

the model as a k-D wire-frame plane. **?@fig-plane-box-model** shows the models for the `plane` and `box` data, 2D and 3D respectively.

We look at the model in the data space to check how well the model fits the data. If it fits well, the points will cluster tightly around the model representation, with little spread in other directions.

```
plane_m <- pca_model(plane_pca)
plane_m_d <- rbind(plane_m$points, plane)
animate_xy(plane_m_d, edges=plane_m$edges,
           axes="bottomleft",
           edges.col="#E7950F",
           edges.width=3)
render_gif(plane_m_d,
           grand_tour(),
           display_xy(half_range=0.9,
                      edges=plane_m$edges,
                      edges.col="#E7950F",
                      edges.width=3),
           gif_file="gifs/plane_model.gif",
           frames=500,
           width=400,
           height=400,
           loop=FALSE)
box_pca <- prcomp(box)
box_m <- pca_model(box_pca, d=3)
box_m_d <- rbind(box_m$points, box)
animate_xy(box_m_d, edges=box_m$edges,
           axes="bottomleft", edges.col="#E7950F", edges.width=3)
render_gif(box_m_d,
           grand_tour(),
           display_xy(half_range=0.9,
                      edges=box_m$edges,
                      edges.col="#E7950F",
                      edges.width=3),
           gif_file="gifs/box_model.gif",
           frames=500,
           width=400,
           height=400,
           loop=FALSE)
```

4.2.1 Example: pisa

The model for the `pisa` data is a 1D vector, shown in **?@fig-pisa-model**.

```
pisa_model <- pca_model(pisa_pca, d=1, s=2)

pisa_all <- rbind(pisa_model$points, pisa_std)
animate_xy(pisa_all, edges=pisa_model$edges,
           edges.col="#E7950F", edges.width=3)
render_gif(pisa_all,
           grand_tour(),
           display_xy(half_range=0.9,
                      edges=pisa_model$edges,
                      edges.col="#E7950F",
                      edges.width=5),
           gif_file="gifs/pisa_model.gif",
           frames=500,
           width=400,
           height=400,
           loop=FALSE)
```

4.2.2 Example: aflw

It is less useful to examine the PCA model for the `aflw` data, because the main patterns that were of interest were the exceptional players. However, we will do it anyway! **?@fig-aflw-model** shows the 4D PCA model overlain on the data. Even though the distribution of points is not as symmetric and balanced as the other examples, we can see that the cube structure mirrors the variation. We can see that the relationships between variables are not strictly linear, because the spread extends unevenly away from the box.

```
aflw_model <- pca_model(aflw_pca, d=4, s=1)

aflw_all <- rbind(aflw_model$points, aflw_std[,7:35])
animate_xy(aflw_all, edges=aflw_model$edges,
           edges.col="#E7950F",
           edges.width=3,
           half_range=0.8,
           axes="off")
render_gif(aflw_all,
           grand_tour(),
           display_xy(half_range=0.8,
```

```
                edges=aflw_model$edges,
                edges.col="#E7950F",
                edges.width=3,
                axes="off"),
          gif_file="gifs/aflw_model.gif",
          frames=500,
          width=400,
          height=400,
          loop=FALSE)
```

4.3 When relationships are not linear

4.3.1 Example: outliers

Figure 4.6 shows the scree plot for the planar data with noise and outliers. It is very similar to the scree plot on the data without the outliers (Figure 4.2). However, what we see from **?@fig-p-o-pca** is that PCA loses the outliers. The animation in (a) shows the full data, and the outliers marked by colour and labels 1, 2, are clearly unusual in some projections. When we examine the tour of the first four PCs (as suggested by the scree plot) the outliers are not unusual. They are almost contained in the point cloud. The reason is clear when all the PCs are plotted, and the outliers can be seen to be clearly detected only in PC5, PC6 and PC7.

```
plane_n_o_pca <- prcomp(plane_noise_outliers)
ggscree(plane_n_o_pca, q = 7) + theme_minimal()
```

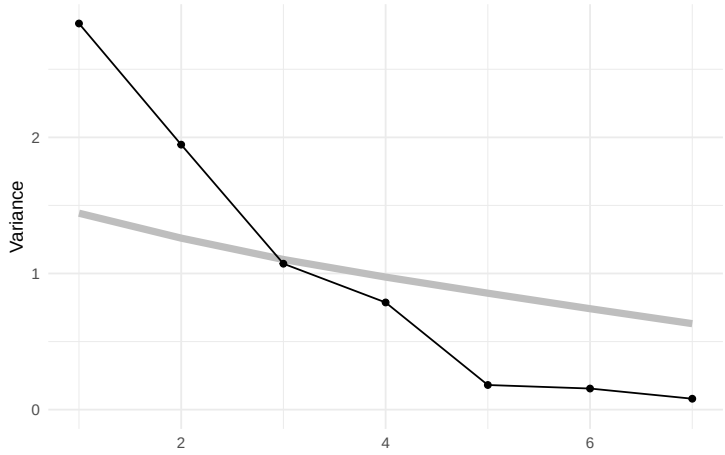


Figure 4.6: Scree plot of the planar data with noise and an outlier. It is almost the same as the data without the outliers.

```
clrs <- hcl.colors(12, "Zissou 1")
p_col <- c(rep("black", 100), clrs[11], clrs[11])
p_obs_labels <- c(rep("", 100), "1", "2")

animate_xy(plane_n_o_pca$x[,1:4],
           col=p_col,
           obs_labels=p_obs_labels)
animate_xy(plane_noise_outliers,
           col=p_col,
           obs_labels=p_obs_labels)
render_gif(plane_noise_outliers,
           grand_tour(),
           display_xy(half_range=0.8,
                      col=p_col,
             obs_labels=p_obs_labels),
           gif_file="gifs/plane_n_o_clr.gif",
           frames=500,
           width=200,
           height=200,
           loop=FALSE)
render_gif(plane_n_o_pca$x[,1:4],
           grand_tour(),
           display_xy(half_range=0.8,
                      col=p_col,
             obs_labels=p_obs_labels),
```

```
            gif_file="gifs/plane_n_o_pca.gif",
            frames=500,
            width=200,
            height=200,
            loop=FALSE)
```

```
library(GGally)
ggscatmat(plane_n_o_pca$x) + theme_minimal()
```

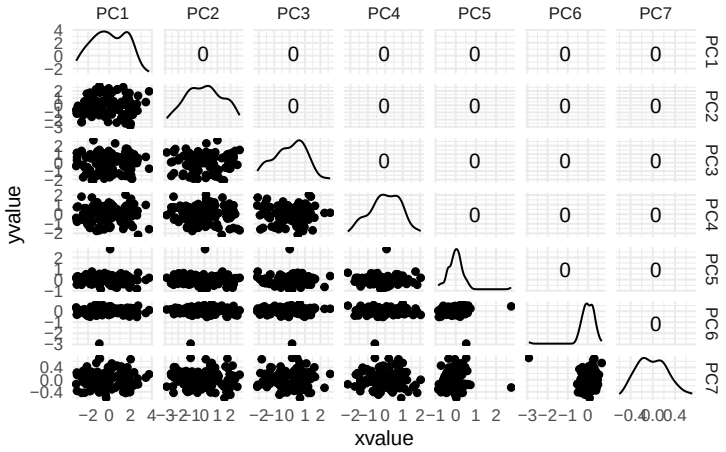


Figure 4.7: From the scatterplot matrix we can see that the outliers are present in PC5, PC6 and PC7. That means by reducing the dimensionality to the first four PCs the model has missed some important characteristics in the data.

4.3.2 Example: Non-linear associations

?@fig-plane-nonlin shows the tour of the full 5D data containing non-linear relationships in comparison with a tour of the first three PCs, as recommended by the scree plot (Figure 4.8). The PCs capture some clear and very clean non-linear relationship, but it looks like it has missed some of the complexities of the relationships. The scatterplot matrix of all 5 PCs (Figure 4.9) shows that PC4 and PC5 contain interesting features: more non-linearity, and curiously an outlier.

```
data(plane_nonlin)
plane_nonlin_pca <- prcomp(plane_nonlin)
ggscree(plane_nonlin_pca, q = 5) + theme_minimal()
```

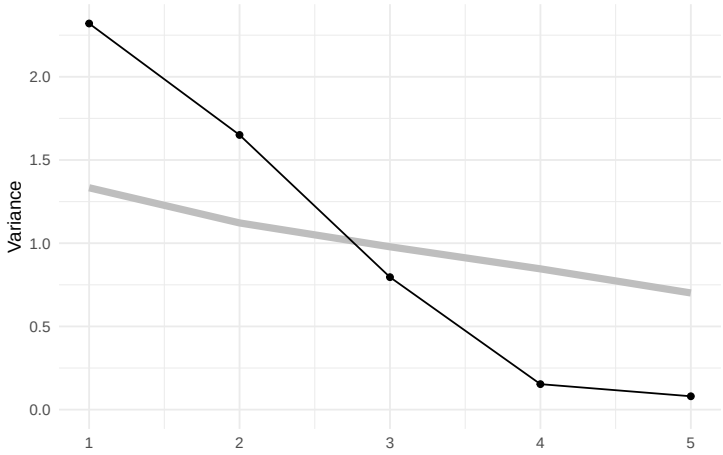


Figure 4.8: Scree plot of the non-linear data suggests three PCs.

```
animate_xy(plane_nonlin_pca$x[,1:3])
render_gif(plane_nonlin_pca$x[,1:3],
           grand_tour(),
           display_xy(half_range=0.8),
           gif_file="gifs/plane_nonlin_pca.gif",
           frames=500,
           width=200,
           height=200)
```

```
ggscatmat(plane_nonlin_pca$x)
```

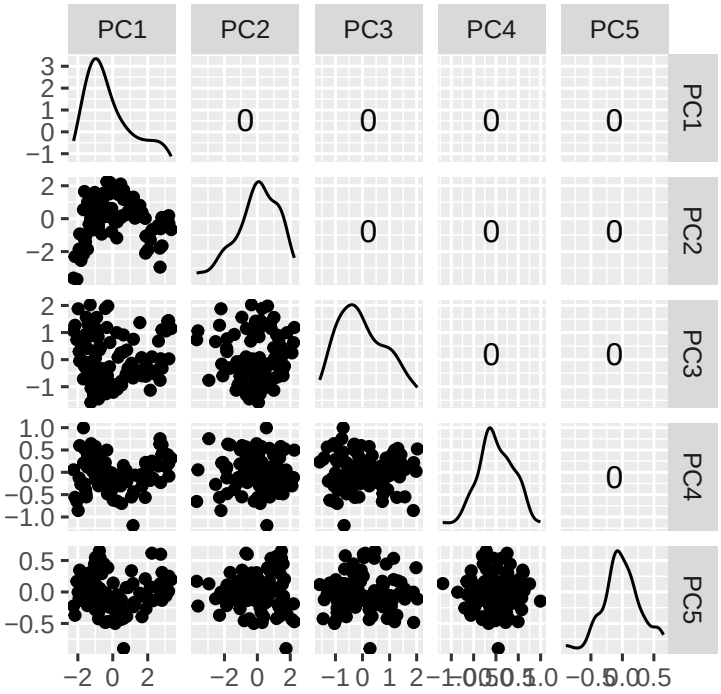


Figure 4.9: From the scatterplot matrix we can see that the there is a non-linear relationship visible in PC1 and PC2, with perhaps a small contribution from PC3. However, we can see that when the data is reduced to three PCs, it misses catching all on the non-linear relationships and also interestingly it seems that there is an unusual observation also.

One of the dangers of PCA is that interesting and curious details of the data only emerge in the lowest PCs, that are usually discarded. The tour, and examining the smaller PCs, can help to discover them.

Exercises

1. Make a scatterplot matrix of the first four PCs of the `aflw` data. Is the branch pattern visible in any pair?
2. Construct five new variables to measure these skills offense, defense, playing time, ball movement, errors. Using the tour, examine the

relationship between these variables. Map out how a few players could be characterised based on these directions of skills.
3. Symmetrise any `aflw` variables that have skewed distributions using a log or square root transformation. Then re-do the PCA. What do we learn that is different about associations between the skill variables?
4. Examine the `bushfires` data using a grand tour on the numeric variables, ignoring the `cause` (class) variable. Note any issues such as outliers, or skewness that might affect PCA. How many principal components would be recommended by the scree plot? Examine this PCA model with the data, and explain how well it does or doesn't fit.
5. Use the `pca_tour` to examine the first five PCs of the `bushfires` data. How do all of the variables contribute to this reduced space?
6. Reduce the dimension of the `sketches` data to 12 PCs. How much variation does this explain? Is there any obvious clustering in this lower dimensional space?

Project

Linear dimension reduction can optimise for other criteria, and here we will explore one example: the algorithm implemented in the `dobin` package finds a basis in which the first few directions are optimized for the detection of outliers in the data. We will examine how it performs for the `plane_noise_outliers` data (the example where outliers were hidden in the first four principal components.)

1. Start by looking up the documentation of `dobin::dobin`. How many parameters does the method depend on?
2. We first apply the function to the `plane_noise_outliers` data using default values for all parameters.
3. Recall that the outliers were added in rows 101 and 102 of the data. Make a scatter plots showing the projection onto the first, second and third component, using color to highlight the outliers. Are they visible as outliers with three components?
4. Adjust the `frac` parameter of the `dobin` function to `frac = 0.99` and repeat the graphical evaluation from point 3. How does it compare to the previous solution?

5

Non-linear dimension reduction

5.1 Background

Non-linear dimension reduction (NLDR) aims to find a low-dimensional representation of the high-dimensional data that shows the main features of the data. In statistics, it dates back to Kruskal (1964a)'s work on multidimensional scaling (MDS). Some techniques only require an interpoint similarity or distance matrix as the main ingredient, rather than the full data. We'll focus on when the full data is available here, so we can also compare structure perceived using the tour on the high-dimensional space, relative to structure revealed in the low-dimensional embedding.

There are many methods available for generating non-linear low dimensional representations of the data. MDS is a classical technique that minimises the difference between two interpoint distance matrices, the distance between points in the high-dimensions, and in the low-dimensional representations. A good resource for learning about MDS is Borg & Groenen (2005).

```
library(mulgar)
library(Rtsne)
library(uwot)
library(ggplot2)
library(patchwork)
set.seed(42)
cnl_tsne <- Rtsne(clusters_nonlin)
cnl_umap <- umap(clusters_nonlin)
n1 <- ggplot(as.data.frame(cnl_tsne$Y), aes(x=V1, y=V2)) +
  geom_point() +
  ggtitle("(a) t-SNE") +
  theme_minimal() +
  theme(aspect.ratio=1)
n2 <- ggplot(as.data.frame(cnl_umap), aes(x=V1, y=V2)) +
  geom_point() +
  ggtitle("(b) UMAP") +
```

```
  theme_minimal() +
  theme(aspect.ratio=1)
n1 + n2
```

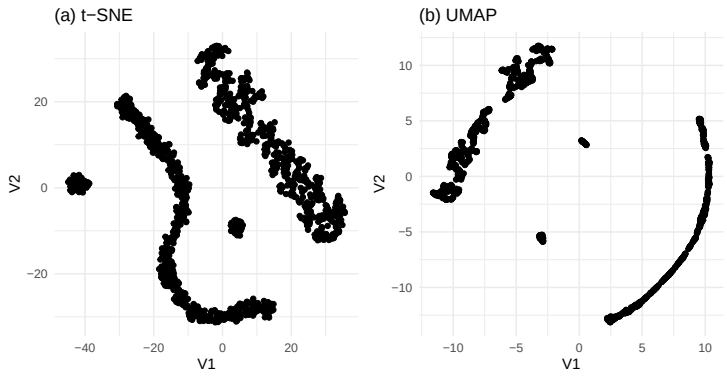


Figure 5.1: Two non-linear embeddings of the non-linear clusters data: (a) t-SNE, (b) UMAP. Both suggest four clusters, with two being non-linear in some form.

Figure 5.1 show two NLDR views of the `clusters_nonlin` data set from the `mulgar` package. Both suggest that there are four clusters, and that some clusters are non-linearly shaped. They disagree on the type of non-linear pattern, where t-SNE represents one cluster as a wavy-shape and UMAP both have a simple parabolic shape. Popular methods in current use include t-SNE (Maaten & Hinton, 2008), UMAP (McInnes et al., 2018) and PHATE (Moon et al., 2019).

```
library(tourr)
render_gif(clusters_nonlin,
           grand_tour(),
           display_xy(),
           gif_file = "gifs/clusters_nonlin.gif",
           frames = 500,
           width = 300,
           height = 300)
```

The full 4D data is shown with a grand tour in **?@fig-clusters-nonlin**. The four clusters suggested by the NLDR methods can be seen. We also get a better sense of the relative size and proximity of the clusters. There are two small spherical clusters, one quite close to the end of the large sine wave cluster. The fourth cluster is relatively small, and has a slight curve, like a bent rod. The

t-SNE representation is slightly more accurate than the UMAP representation. We would expect that the wavy cluster is the sine wave seen in the tour.

NLDR can provide useful low-dimensional summaries of high-dimensional structure but you need to check whether it is a sensible and accurate representation by comparing with what is perceivd from a tour.

5.2 Linking NLDR representation with tour view

NLDR can produce useful low-dimensional summaries of structure in high-dimensional data, like those shown in Figure 5.1. However, there are numerous pitfalls. The fitting procedure can produce very different representations depending on the parameter choices, and even the random number seeding the fit. (You can check this by changing the `set.seed` in the code above, and by changing from the default parameters.) Also, it may not be possible to represent the high-dimensional structures faithfully low dimensions. For these reasons, one needs to connect the NLDR view with a tour of the data, to help assess its usefulness and accuracy. For example, with this data, we would want to know which of the two curved clusters in the UMAP representation correspond to the sine wave cluster.

5.2.1 Using `liminal`

Figure 5.2 shows how the NLDR plot can be linked to a tour view, using the `liminal` package, to better understand how well the structure of the data is represented. Here we see learn that the smile in the UMAP embedding is the small bent rod cluster, and that the unibrow is the sine wave.

```
library(liminal)
umap_df <- data.frame(umapX = cnl_umap[, 1],
                      umapY = cnl_umap[, 2])
limn_tour_link(
  umap_df,
  clusters_nonlin,
  cols = x1:x4
)
```

Figure 5.2: Two screenshots from liminal showing which clusters match between the UMAP representation and the tour animation. The smile corresponds to the small bent rod cluster. The unibrow matches to the sine wave cluster.

5.2.2 Using `detourr`

?@fig-detourr-clusters-nonlin shows how the linking is achieved using `detourr`. It uses a shared data object, as made possible by the `crosstalk` package, and the UMAP view is made interactive using `plotly`.

```
library(detourr)
library(dplyr)
library(crosstalk)
library(plotly)
umap_df <- data.frame(umapX = cnl_umap[, 1],
                      umapY = cnl_umap[, 2])
cnl_df <- bind_cols(clusters_nonlin, umap_df)
shared_cnl <- SharedData$new(cnl_df)

detour_plot <- detour(shared_cnl, tour_aes(
  projection = starts_with("x"))) |>
    tour_path(grand_tour(2),
                    max_bases=50, fps = 60) |>
       show_scatter(alpha = 0.7, axes = FALSE,
                    width = "100%", height = "450px")

umap_plot <- plot_ly(shared_cnl,
                    x = ~umapX,
                    y = ~umapY,
                    color = I("black"),
                    height = 450) %>%
    highlight(on = "plotly_selected",
              off = "plotly_doubleclick") %>%
    add_trace(type = "scatter",
              mode = "markers")

bscols(
     detour_plot, umap_plot,
     widths = c(5, 6)
 )
```

5.3 Example: fake_trees

Figure 5.3 shows a more complex example, using the fake_trees data. We know that the 10D data has a main branch, and 9 branches (clusters) attached to it, absed on our explorations in the earlier chapters. The t-SNE view, where points are coloured by the known branch ids, is very helpful for seeing the linear branch structure.

What we can't tell is that there is a main branch from which all of the others extend. We also can't tell which of the clusters corresponds to this branch. Linking the plot with a tour helps with this. Although, not shown in the sequence of snapshots in Figure 5.3, the main branch is actually the dark blue cluster, which is separated into three pieces by t-SNE.

```
library(liminal)
library(Rtsne)
data(fake_trees)
set.seed(2020)
tsne <- Rtsne::Rtsne(dplyr::select(fake_trees, dplyr::starts_with("dim")))
tsne_df <- data.frame(tsneX = tsne$Y[, 1],
                      tsneY = tsne$Y[, 2])
limn_tour_link(
  tsne_df,
  fake_trees,
  cols = dim1:dim10,
  color = branches
)
```

Figure 5.3: Three snapshots of using the liminal linked views to explore how t-SNE has summarised the fake_trees data in 2D.

The t-SNE representation clearly shows the linear structures of the data, but viewing this 10D data with the tour shows that t-SNE makes several inaccurate breaks of some of the branches.

Exercises

1. Using the `penguins_sub` data generate a 2D representation using t-SNE. Plot the points mapping the colour to species. What is most surprising? (Hint: Are the three species represented by three distinct clusters?)
2. Re-do the t-SNE representation with different parameter choices. Are the results different each time, or could they be considered to be equivalent?
3. Use `liminal` to link the t-SNE representation to a tour of the penguins. Highlight the points that have been placed in an awkward position by t-SNE from others in their species. Watch them relative to the others in their species in the tour view, and think about whether there is any rationale for the awkward placement.
4. Use UMAP to make the 2D representation, and use `liminal` to link with a tour to explore the result.
5. Repeat 1-4 using `detourr`.

References

Abbott, E. (1884). *Flatland: A romance of many dimensions*. Dover Publications.

Ahlberg, C., Williamson, C., & Shneiderman, B. (1991). Dynamic Queries for Information Exploration: An Implementation and Evaluation. *ACM CHI '92 Conference Proceedings*, 619–626.

Anderson, E. (1957). A Semigraphical Method for the Analysis of Complex Problems. *Proceedings of the National Academy of Science, 13*, 923–927.

Andrews, D. F. (1972). Plots of High-dimensional Data. *Biometrics*, *28*, 125–136.

Andrews, D. F., Gnanadesikan, R., & Warner, J. L. (1971). Transformations of Multivariate Data. *Biometrics*, *27*, 825–840.

Anselin, L., & Bao, S. (1997). Exploratory Spatial Data Analysis Linking SpaceStat and ArcView. In M. M. Fischer & A. Getis (Eds.), *Recent Developments in Spatial Analysis* (pp. 35–59). Springer.

Arnold, J. B. (2021). *Ggthemes: Extra themes, scales and geoms for ggplot2*. https://github.com/jrnold/ggthemes

ASA Statistical Graphics Section. (2023). *Video Library*. https://community.amstat.org/jointscsg-section/media/videos.

Asimov, D. (1985). The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal of Scientific and Statistical Computing*, *6*(1), 128–143.

Auguie, B. (2017). *gridExtra: Miscellaneous functions for "grid" graphics*. https://CRAN.R-project.org/package=gridExtra

Australian Bureau of Agricultural and Resource Economics and Sciences. (2018). *Forests of Australia*. https://www.agriculture.gov.au/abares/forestsaustralia/forest-data-maps-and-tools/spatial-data/forest-cover

Becker, R. A., & Chambers, J. M. (1984). *S: An environment for data analysis and graphics*. Wadsworth.

Becker, R. A., & Cleveland, W. S. (1988). Brushing Scatterplots. In W. S. Cleveland & M. E. McGill (Eds.), *Dynamic graphics for statistics* (pp. 201–224). Wadsworth.

Becker, R., Cleveland, W. S., & Shyu, M.-J. (1996). The Visual Design and Control of Trellis Displays. *Journal of Computational and Graphical Statistics*, *6*(1), 123–155.

Bederson, B. B., & Schneiderman, B. (2003). *The craft of information visualization: Readings and reflections*. Morgan Kaufmann.

Bellman, R. (1961). *Adaptive control processes : A guided tour*.

Bickel, P. J., Kur, G., & Nadler, B. (2018). Projection pursuit in high dimen-

sions. *Proceedings of the National Academy of Sciences*, *115*, 9151–9156. https://doi.org/10.1073/pnas.1801177115

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Boehmke, B., & Greenwell, B. M. (2019). *Hands-on machine learning with r (1st ed.)*. Chapman; Hall/CRC. https://doi.org/10.1201/9780367816377

Boelaert, J., Ollion, E., & Sodoge, J. (2022). *aweSOM: Interactive self-organizing maps*. https://CRAN.R-project.org/package=aweSOM

Bonneau, G.-P., Ertl, T., & Nielson, G. M. (Eds.). (2006). *Scientific visualization: The visual extraction of knowledge from data*. Springer.

Borg, I., & Groenen, P. J. F. (2005). *Modern Multidimensional Scaling*. Springer.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Cutler, A., Liaw, A., & Wiener, M. (2022). *randomForest: Breiman and cutler's random forests for classification and regression*. https://www.stat.berkeley.edu/~breiman/RandomForests/

Breiman, L., Friedman, J., Olshen, C., & Stone, C. (1984). *Classification and Regression Trees*. Wadsworth; Brooks/Cole.

Buja, A. (1996). Interactive Graphical Methods in the Analysis of Customer Panel Data: Comment. *Journal of Business & Economic Statistics*, *14*(1), 128–129.

Buja, A., & Asimov, D. (1986). Grand Tour Methods: An Outline. *Computing Science and Statistics*, *17*, 63–67.

Buja, A., Asimov, D., Hurley, C., & McDonald, J. A. (1988). Elements of a Viewing Pipeline for Data Analysis. In W. S. Cleveland & M. E. McGill (Eds.), *Dynamic graphics for statistics* (pp. 277–308). Wadsworth.

Buja, A., Cook, D., Asimov, D., & Hurley, C. (1997). *Dynamic Projections in High-Dimensional Visualization: Theory and Computational Methods*. AT&T Labs.

Buja, A., Cook, D., Asimov, D., & Hurley, C. (2005). Computational Methods for High-Dimensional Rotations in Data Visualization. In C. R. Rao, E. J. Wegman, & J. L. Solka (Eds.), *Handbook of statistics: Data mining and visualization* (pp. 391–414). Elsevier/North-Holland.

Buja, A., Cook, D., & Swayne, D. (1996). Interactive High-Dimensional Data Visualization. *Journal of Computational and Graphical Statistics*, *5*(1), 78–99.

Buja, A., Hurley, C., & McDonald, J. A. (1986). A Data Viewer for Multivariate Data. *Computing Science and Statistics*, *17*(1), 171–174.

Buja, A., & Swayne, D. F. (2002). Visualization Methodology for Multidimensional Scaling. *Journal of Classification*, *19*(1), 7–43.

Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H., & Chen, L. (2008). Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, *17*(2), 444–472. https://doi.org/10.1198/106186008X318440

Buja, A., & Tukey, P. (Eds.). (1991). *Computing and Graphics in Statistics*. Springer-Verlag.

Card, S. K., Mackinlay, J. D., & Schneiderman, B. (1999). *Readings in information visualization*. Morgan Kaufmann Publishers.

Carr, D. B., Wegman, E. J., & Luo, Q. (1996). *ExplorN: Design Considerations Past and Present* (Technical Report No. 129). Center for Computational Statistics, George Mason University.

Chatfield, C. (1995). *Problem solving: A statistician's guide*. Chapman; Hall/CRC Press.

Chen, C., Härdle, W., & Unwin, A. (Eds.). (2006). *Handbook of computational statistics (volume III) data visualization*. Springer.

Chen, C.-H., Härdle, W., & Unwin, A. (Eds.). (2007). *Handbook of Data Visualization*. Springer.

Cheng, B., & Titterington, M. (1994). Neural Networks: A Review from a Statistical Perspective. *Statistical Science*, *9*(1), 2–30.

Cheng, J., & Sievert, C. (2021). *Crosstalk: Inter-widget interactivity for HTML widgets*. https://rstudio.github.io/crosstalk/

Chernoff, H. (1973). The Use of Faces to Represent Points in k-dimensional Space Graphically. *Journal of the American Statistical Association*, *68*, 361–368.

Cleveland, W. S. (1979). Robust Localy Weighted Regression and Smoothing Scatterplots. *Journal of American Statistics Association*, *74*, 829–836.

Cleveland, W. S. (1993). *Visualizing Data*. Hobart Press.

Cleveland, W. S., & McGill, M. E. (Eds.). (1988). *Dynamic graphics for statistics*. Wadsworth.

Cook, D., & Buja, A. (1997). Manual Controls For High-Dimensional Data Projections. *Journal of Computational and Graphical Statistics*, *6*(4), 464–480.

Cook, D., Buja, A., & Cabrera, J. (1993). Projection Pursuit Indexes Based on Orthonormal Function Expansions. *Journal of Computational and Graphical Statistics*, *2*(3), 225–250.

Cook, D., Buja, A., Cabrera, J., & Hurley, C. (1995b). Grand Tour and Projection Pursuit. *Journal of Computational and Graphical Statistics*, *4*(3), 155–172.

Cook, D., Buja, A., Cabrera, J., & Hurley, C. (1995a). Grand Tour and Projection Pursuit. *Journal of Computational and Graphical Statistics*, *4*(3), 155–172.

Cook, D., Hofmann, H., Lee, E.-K., Yang, H., Nikolau, B., & Wurtele, E. (2007). Exploring Gene Expression Data, Using Plots. *Journal of Data Science*, *5*(2), 151–182.

Cook, D., & Laa, U. (2023). *Mulgar: Functions for pre-processing data for multivariate data visualisation using tours*.

Cook, D., Lee, E.-K., Buja, A., & Wickham, H. (2006). Grand Tours, Projection Pursuit Guided Tours and Manual Controls. In C.-H. Chen, W. Härdle, & A. Unwin (Eds.), *Handbook of Data Visualization*. Springer.

Cook, D., Majure, J. J., Symanzik, J., & Cressie, N. (1996). Dynamic Graphics in a GIS: Exploring and Analyzing Multivariate Spatial Data using

Linked Software. *Computational Statistics: Special Issue on Computer Aided Analyses of Spatial Data*, *11*(4), 467–480.

Cook, D., & Swayne, D. F. (2007). *Interactive and dynamic graphics for data analysis: With R and GGobi*. Springer-Verlag. https://doi.org/10.1007/978-0-387-71762-3

Cortes, C., Pregibon, D., & Volinsky, C. (2003). Computational Methods for Dynamic Graphs. *Journal of Computational & Graphical Statistics*, *12*(4), 950–970.

Cortes, C., & Vapnik, V. N. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297.

d'Ocagne, M. (1885). *Coordonnées Parallèles et Axiales: Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées paralléles*. Gauthier-Villars.

Dalgaard, P. (2002). *Introductory statistics with R*. Springer.

Dasu, T., Swayne, D. F., & Poole, D. (2005). Grouping Multivariate Time Series: A Case Study. *Proceedings of the IEEE Workshop on Temporal Data Mining: Algorithms, Theory and Applications, in Conjunction with the Conference on Data Mining, Houston, November 27, 2005*, 25–32.

de Vries, A., & Ripley, B. D. (2022). *Ggdendro: Create dendrograms and tree diagrams using ggplot2*. https://github.com/andrie/ggdendro

Department of Environment, Land, Water & Planning. (2019). *Fire Origins - Current and Historical*. https://discover.data.vic.gov.au/dataset/fire-origins-current-and-historical

Department of Environment, Land, Water & Planning. (2020a). *CFA - Fire Station*. https://discover.data.vic.gov.au/dataset/cfa-fire-station-vmfeat-geomark_point

Department of Environment, Land, Water & Planning. (2020b). *Recreation Sites*. https://discover.data.vic.gov.au/dataset/recreation-sites

Diaconis, P., & Freedman, D. (1984b). Asymptotics of Graphical Projection Pursuit. *Annals of Statistics*, *12*, 793–815.

Diaconis, P., & Freedman, D. (1984a). Asymptotics of graphical projection pursuit. *Annals of Statistics*, *12*(3), 793–815. https://doi.org/10.1214/aos/1176346703

Dykes, J., MacEachren, A. M., & Kraak, M.-J. (2005). *Exploring geovisualization*. Elsevier.

Everitt, B. S., Landau, S., & Leese, M. (2001). *Cluster Analysis (4th ed)*. Edward Arnold.

Fienberg, S. E. (1979). Graphical Methods in Statistics. *Journal of American Statistical Association*, *33*(4), 165–178.

Fisher, R. A. (1936b). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, *7*, 179–188.

Fisher, R. A. (1936a). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

Fisher, R. A. (1938). The Statistical Utilization of Multiple Measurements. *Annals of Eugenics*, *8*, 376–386.

Fisherkeller, M. A., Friedman, J. H., & Tukey, J. W. (1973). *PRIM-9, an interactive multidimensional data display and analysis system.* https://www.youtube.com/watch?v=B7XoW2qiFUA

Fisherkeller, M. A., Friedman, J. H., & Tukey, J. W. (1974). PRIM-9, an interactive multidimensional data display and analysis system. In W. S. Cleveland (Ed.), *The collected works of john w. Tukey: Graphics 1965-1985, volume v* (pp. 340–346).

Forbes, J., Cook, D., & Hyndman, R. J. (2020). Spatial modelling of the two-party preferred vote in australian federal elections: 2001–2016. *Australian & New Zealand Journal of Statistics*, *62*(2), 168–185. https://doi.org/https://doi.org/10.1111/anzs.12292

Ford, B. J. (1992). *Images of science: A history of scientific illustration.* The British Library.

Forgy, E. (1965). Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, *21*(3), 768–769.

Fraley, C., & Raftery, A. E. (2002). Model-based Clustering, Discriminant Analysis, Density Estimation. *Journal of the American Statistical Association*, *97*, 611–631.

Fraley, C., Raftery, A. E., & Scrucca, L. (2022). *Mclust: Gaussian mixture modelling for model-based clustering, classification, and density estimation.* https://mclust-org.github.io/mclust/

Friedman, J. H. (1987). Exploratory Projection Pursuit. *Journal of American Statistical Association*, *82*, 249–266.

Friedman, J. H., & Tukey, J. W. (1974). A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Transactions on Computing C*, *23*, 881–889.

Friendly, M., & Denis, D. J. (2004). *Milestones in the history of thematic cartography, statistical graphics, and data visualization.* http://www.math.yorku.ca/SCS/Gallery/milestone/.

Furnas, G. W., & Buja, A. (1994). Prosection Views: Dimensional Inference Through Sections and Projections. *Journal of Computational and Graphical Statistics*, *3*(4), 323–385.

Gabriel, K. R. (1971). The Biplot Graphical Display of Matrices with Applications to Principal Component Analysis. *Biometrika*, *58*, 453–467.

Gentle, J. E., Härdle, W., & Mori, Y. (Eds.). (2004). *Handbook of computational statistics: Concepts and methods.* Springer.

Giordani, P., Ferraro, M. B., & Martella, F. (2020). *An introduction to clustering with r.* Springer Singapore. https://doi.org/10.1007/978-981-13-0553-5

Glover, D. M., & Hopke, P. K. (1992). Exploration of Multivariate Chemical Data by Projection Pursuit. *Chemometrics and Intelligent Laboratory Systems*, *16*, 45–59.

Good, P. (2005). *Permutation, Parametric, and Bootstrap Tests of Hypotheses.* Springer.

Gower, J. C., & Hand, D. J. (1996). *Biplots*. Chapman; Hall.

Hajibaba, H., Karlsson, L., & Dolnicar, S. (2016). Residents open their homes to tourists when disaster strikes. *Journal of Travel Research*, *56*(8), 1065–1078.

Hansen, C., & Johnson, C. R. (2004). *Visualization handbook*. Academic Press.

Harrison, P. (2022). *Langevitour: Langevin tour*. https://logarithmic.net/langevitour/

Hart, C., & Wang, E. (2022). *Detourr: Portable and performant tour animations*. https://casperhart.github.io/detourr/

Hartigan, J. A., & Kleiner, B. (1981). Mosaics for Contingency Tables. *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, 268–273.

Hartigan, J., & Kleiner, B. (1984). A Mosaic of Television Ratings. *The American Statistician*, *38*, 32–35.

Haslett, J., Bradley, R., Craig, P., Unwin, A., & Wills, G. (1991). Dynamic Graphics for Exploring Spatial Data with Application to Locating Global and Local Anomalies. *The American Statistician*, *45*(3), 234–242.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.). (2015). *Handbook of cluster analysis*. Chapman; Hall/CRC. https://doi.org/10.1201/b19706

Hofmann, H. (2001). *Graphical Tools for the Exploration of Multivariate Categorical Data*. Books on Demand.

Hofmann, H. (2003). Constructing and Reading Mosaicplots. *Computational Statistics and Data Analysis*, *43*(4), 565–580.

Hofmann, H., & Theus, M. (1998). Selection Sequences in MANET. *Computational Statistics*, *13*(1), 77–87.

Horikoshi, M., & Tang, Y. (2018). *Ggfortify: Data visualization tools for statistical analysis results*. https://CRAN.R-project.org/package=ggfortify

Horikoshi, M., & Tang, Y. (2023). *Ggfortify: Data visualization tools for statistical analysis results*. https://github.com/sinhrks/ggfortify

Horst, A., Hill, A., & Gorman, K. (2022). *Palmerpenguins: Palmer archipelago (antarctica) penguin data*. https://CRAN.R-project.org/package=palmerpenguins

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417--441. https://doi.org/10.1037/h0071325

Huber, P. J. (1985). Projection Pursuit (with discussion). *Annals of Statistics*, *13*, 435–525.

Hurley, C. (1987). *The data viewer: An interactive program for data analysis* [PhD thesis]. University of Washington.

Iannone, R., Cheng, J., Schloerke, B., Hughes, E., Lauer, A., & Seo, J. (2023). *Gt: Easily create presentation-ready display tables*. https://CRAN.R-project.org/package=gt

Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, *5*, 299–314.

Ihaka, R., Murrell, P., Hornik, K., Fisher, J. C., Stauffer, R., Wilke, C. O., McWhite, C. D., & Zeileis, A. (2023). *Colorspace: A toolbox for manipulating and assessing colors and palettes*. https://CRAN.R-project.org/package=colorspace

Inselberg, A. (1985). The Plane with Parallel Coordinates. *The Visual Computer*, *1*, 69–91.

Iowa State University. (2020). *ASOS-AWOS-METAR data download*. https://mesonet.agron.iastate.edu/request/download.phtml?network=AU___ASOS

Johnson, D., & Travis, J. (2007). *Flatland: The movie*. https://round-drum-w7xh.squarespace.com/our-story.

Johnson, R. A., & Wichern, D. W. (2002). *Applied multivariate statistical analysis (5th ed)*. Prentice-Hall.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Phil. Trans. R. Soc. A.*, *374*, 20150202. https://doi.org/10.1098/rsta.2015.0202

Jones, M. C., & Sibson, R. (1987). What is Projection Pursuit? (With discussion). *Journal of the Royal Statistical Society, Series A*, *150*, 1–36.

Kassambara, A. (2017). *Practical guide to cluster analysis in r: Unsupervised machine learning*. STHDA.

Kassambara, A. (2023). *Ggpubr: ggplot2 based publication ready plots*. https://rpkgs.datanovia.com/ggpubr/

Kohonen, T. (2001). *Self-Organizing Maps (3rd ed)*. Springer.

Koschat, M. A., & Swayne, D. F. (1996). Interactive Graphical Methods in the Analysis of Customer Panel Data (with discussion). *Journal of Business and Economic Statistics*, *14*(1), 113–132.

Krijthe, J. (2022). *Rtsne: T-distributed stochastic neighbor embedding using a barnes-hut implementation*. https://github.com/jkrijthe/Rtsne

Kruskal, J. B. (1964a). Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. *Psychometrika*, *29*, 1–27.

Kruskal, J. B. (1964b). Nonmetric Multidimensional Scaling: A Numerical Method. *Psychometrika*, *29*, 115–129.

Kruskal, J. B., & Wish, M. (1978). *Multidimensional Scaling*. Sage Publications.

Kuhn, M., & Wickham, H. (2020). *Tidymodels: A collection of packages for modeling and machine learning using tidyverse principles*. https://www.tidymodels.org

Kuhn, M., & Wickham, H. (2023). *Tidymodels: Easily install and load the tidymodels packages*. https://CRAN.R-project.org/package=tidymodels

Laa, U., Cook, D., & Valencia, G. (2020a). A slice tour for finding hollowness in high-dimensional data. *Journal of Computational and Graphical Statistics*, *29*(3), 681–687. https://doi.org/10.1080/10618600.2020.1777140

Laa, U., Cook, D., & Valencia, G. (2020b). A slice tour for finding hollowness in high-dimensional data. *Journal of Computational and Graphical Statistics*, *29*(3), 681–687. https://doi.org/10.1080/10618600.2020.1777140

Lancaster, H. O. (1965). The helmert matrices. *The American Mathematical Monthly*, *72*(1), 4–12.

Laurent, S. (2023). *Cxhull: Convex hull*. https://github.com/stla/cxhull

Lee, E.-K. (2018). PPtreeViz: An r package for visualizing projection pursuit classification trees. *Journal of Statistical Software*, *83*(8), 1–30. https://doi.org/10.18637/jss.v083.i08

Lee, E.-K., & Cook, D. (2009). A projection pursuit index for large p small n data. *Statistics and Computing*, *20*, 381–392. https://doi.org/10.1007/s11222-009-9131-1

Lee, E.-K., Cook, D., Klinke, S., & Lumley, T. (2005). Projection Pursuit for Exploratory Supervised Classification. *Journal of Computational and Graphical Statistics*, *14*(4), 831–846.

Lee, S. (2021). *Liminal: Multivariate data visualization with tours and embeddings*. https://CRAN.R-project.org/package=liminal

Lee, S., Cook, D., Silva, N. da, Laa, U., Spyrison, N., Wang, E., & Zhang, H. S. (2022). The state-of-the-art on tours for dynamic visualization of high-dimensional data. *WIREs Computational Statistics*, *14*(4), e1573. https://doi.org/10.1002/wics.1573

Lee, Y. D., Cook, D., Park, J., & Lee, E.-K. (2013). PPtree: Projection pursuit classification tree. *Electronic Journal of Statistics*, *7*(none), 1369–1386. https://doi.org/10.1214/13-EJS810

Leisch, F., & Gruen, B. (2023). *CRAN task view: Cluster analysis & finite mixture models*. https://cran.r-project.org/web/views/Cluster.html.

Leisch, F., & Grün, B. (2020). *MSA: Market segmentation analysis*.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22. https://CRAN.R-project.org/doc/Rnews/

Littman, M. L., Swayne, D. F., Dean, N., & Buja, A. (1992). Visualizing the Embedding of Objects in Euclidean Space. *Computing Science and Statistics: Proceedings of the 24th Symposium on the Interface*, 208–217.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489

Longley, P. A., Maguire, D. J., Goodchild, M. F., & Rhind, D. W. (2005). *Geographic information systems and science*. John Wiley & Sons.

Loperfido, N. (2018). Skewness-based projection pursuit: A computational approach. *Computational Statistics & Data Analysis*, *120*, 42–57. https://doi.org/https://doi.org/10.1016/j.csda.2017.11.001

Maaten, L. van der, & Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.*, *9*(Nov), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam & J. Neyman (Eds.), *Proc. Of the fifth berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297). University of California Press.

Maindonald, J., & Braun, J. (2003). *Data analysis and graphics using r - an example-based approach*. Cambridge University Press.

Martin, E. (1965). *Flatland*. http://www.der.org/films/flatland.html.

McFarlane, M., & Young, F. W. (1994). Graphical Sensitivity Analysis for Multidimensional Scaling. *Journal of Computational and Graphical Statistics*, *3*, 23–33.

McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform manifold approximation and projection for dimension reduction*. http://arxiv.org/abs/1802.03426

McNeil, D. (1977). *Interactive Data Analysis*. John Wiley & Sons.

McVicar, T. (2011). *Near-surface wind speed. v10. CSIRO. Data collection*. https://doi.org/10.25919/5c5106acbcb02

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2023). *e1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), TU wien*. https://CRAN.R-project.org/package=e1071

Milborrow, S. (2022). *Rpart.plot: Plot rpart models: An enhanced version of plot.rpart*. http://www.milbo.org/rpart-plot/index.html

Mock, T. (2022). *gtExtras: Extending gt for beautiful HTML tables*. https://CRAN.R-project.org/package=gtExtras

Moon, K. R., Dijk, D. van, Wang, Z., Gigante, S., Burkhardt, D. B., Chen, W. S., Yim, K., Elzen, A. van den, Hirn, M. J., Coifman, R. R., Ivanova, N. B., Wolf, G., & Krishnaswamy, S. (2019). Visualizing structure and transitions for biological data exploration. *Nature Biotechnology*, *37*, 1482–1492. https://doi.org/10.1038/s41587-019-0336-3

Murrell, P. (2005). *R graphics*. Chapman & Hall/CRC.

OpenStreetMap contributors. (2020). *Planet dump retrieved from https://planet.osm.org* .https://www.openstreetmap.org.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Pedersen, T. L. (2023). *Patchwork: The composer of plots*. https://CRAN.R-project.org/package=patchwork

Perisic, I., & Posse, C. (2005). Projection pursuit indices based on the empirical distribution function. *Journal of Computational and Graphical Statistics*, *14*(3), 700–715. https://doi.org/10.1198/106186005X69440

Polzehl, J. (1995). Projection Pursuit Discriminant Analysis. *Computational Statistics and Data Analysis*, *20*, 141–157.

Posse, C. (1992). Projection Pursuit Discriminant Analysis for Two Groups. *Communications in Statistics, Part A – Theory and Methods*, *21*, 1–19.

Posse, C. (1995). Tools for Two-dimensional Projection Pursuit. *Journal of Computational and Graphical Statistics*, *4*(2), 83–100.

P-Tree System. (2020). *JAXA Himawari Monitor - User's Guide*. https://www.eorc.jaxa.jp/ptree/userguide.html

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Rao, C. R. (1948). The Utilization of Multiple Measurements in Problems of Biological Classification (with discussion). *Journal of the Royal Statistical Society, Series B*, *10*, 159–203.

Rao, C. R. (Ed.). (1993). *Handbook of Statistics, Vol. 9*. Elsevier Science Publishers.

Rao, C. R., Wegman, E. J., & Solka, J. L. (Eds.). (2006). *Handbook of Statistics: Data Mining and Visualization*. Elsevier/North-Holland.

Ripley, B. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Ripley, B. (2023a). *MASS: Support functions and datasets for venables and ripley's MASS*. http://www.stats.ox.ac.uk/pub/MASS4/

Ripley, B. (2023b). *Nnet: Feed-forward neural networks and multinomial log-linear models*. http://www.stats.ox.ac.uk/pub/MASS4/

Rothkopf, E. Z. (1957). A Measure of Stimulus Similarity and Errors in Some Paired-associate Learning Tasks. *Journal of Experimental Psychology*, *53*, 94–101.

Schloerke, B. (2016). *Geozoo: Zoo of geometric objects*. https://CRAN.R-project.org/package=geozoo

Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., Elberg, A., & Crowley, J. (2023). *GGally: Extension to ggplot2*.

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, *8*(1), 289–317. https://doi.org/10.32614/RJ-2016-021

Shepard, R. N. (1962). The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function, I and II. *Psychometrika*, *27*, 125-139 and 219-246.

Sievert, C. (2020). *Interactive web-based data visualization with r, plotly, and shiny*. Chapman; Hall/CRC. https://plotly-r.com

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., & Despouy, P. (2023). *Plotly: Create interactive web graphics via plotly.js*. https://CRAN.R-project.org/package=plotly

Sjoberg, D. D., Larmarange, J., Curry, M., Lavery, J., Whiting, K., & Zabor, E. C. (2023). *Gtsummary: Presentation-ready data summary and analytic result tables*. https://CRAN.R-project.org/package=gtsummary

Sjoberg, D. D., Whiting, K., Curry, M., Lavery, J. A., & Larmarange, J. (2021). Reproducible summary tables with the gtsummary package. *The R Journal*, *13*, 570–580. https://doi.org/10.32614/RJ-2021-053

Slowikowski, K. (2023). *Ggrepel: Automatically position non-overlapping text labels with ggplot2*. https://github.com/slowkow/ggrepel

Sparks, A. H., Carroll, J., Goldie, J., Marchiori, D., Melloy, P., Padgham, M., Parsonage, H., & Pembleton, K. (2020). *bomrang: Australian government bureau of meteorology (BOM) data client*. https://CRAN.R-project.org/package=bomrang

Spence, R. (2007). *Information visualization: Design for interaction*. Prentice Hall.

Stauffer, R., Mayr, G. J., Dabernig, M., & Zeileis, A. (2009). Somewhere over the rainbow: How to make effective use of colors in meteorological visualizations. *Bulletin of the American Meteorological Society*, *96*(2), 203–216. https://doi.org/10.1175/BAMS-D-13-00155.1

Sutherland, P., Rossini, A., Lumley, T., Lewin-Koh, N., Dickerson, J., Cox, Z., & Cook, D. (2000b). Orca: A Visualization Toolkit for High-Dimensional Data. *Journal of Computational and Graphical Statistics*, *9*(3), 509–529.

Sutherland, P., Rossini, A., Lumley, T., Lewin-Koh, N., Dickerson, J., Cox, Z., & Cook, D. (2000a). Orca: A visualization toolkit for high-dimensional data. *Journal of Computational and Graphical Statistics*, *9*(3), 509–529. https://doi.org/10.1080/10618600.2000.10474896

Swayne, D. F., Buja, A., & Temple Lang, D. (2004). Exploratory visual analysis of graphs in GGobi. In J. Antoch (Ed.), *CompStat: Proceedings in computational statistics, 16th symposium*. Physica-Verlag.

Swayne, D. F., Cook, D., & Buja, A. (1992). XGobi: Interactive Dynamic Graphics in the X Window System with a Link to S. *American Statistical Association 1991 Proceedings of the Section on Statistical Graphics*, 1–8.

Swayne, D. F., Cook, D., & Buja, A. (1998). XGobi: Interactive dynamic data visualization in the x window system. *Journal of Computational and Graphical Statistics*, *7*(1), 113–130. https://doi.org/10.1080/10618600.1998.10474764

Swayne, D. F., & Klinke, S. (1998). Editorial commentary. *Computational Statistics: Special Issue on The Use of Interactive Graphics*, *14*(1).

Swayne, D. F., Temple Lang, D., Buja, A., & Cook, D. (2003). GGobi: Evolving from XGobi into an Extensible Framework for Interactive Data Visualization. *Computational Statistics & Data Analysis*, *43*, 423–444.

Swayne, D., & Buja, A. (1998). Missing Data in Interactive High-Dimensional Data Visualization. *Computational Statistics*, *13*(1), 15–26.

Symanzik, J. (2002). New applications of the image grand tour. *Computing Science and Statistics*, *34*, 500--512. https://math.usu.edu/symanzik/papers/2002_interface.pdf

Symanzik, J. (2004). Interactive and Dynamic Graphics. In J. E. Gentle, W. Härdle, & Y. Mori (Eds.), *Handbook of computational statistics: Concepts and methods* (pp. 293–336). Springer.

Takatsuka, M., & Gahegan, M. (2002). GeoVISTA Studio: A Codeless Visual Programming Environment for Geoscientific Data Analysis and Visualization. *The Journal of Computers and Geosciences*, *28*(10), 1131–1144.

Tang, Y., Horikoshi, M., & Li, W. (2016). Ggfortify: Unified interface to visualize statistical result of popular r packages. *The R Journal*, *8*(2), 474–485. https://doi.org/10.32614/RJ-2016-060

Tarpey, T., Li, L., & Flury, B. (1995). Principal points and self–consistent points of elliptical distributions. *The Annals of Statistics*, *23*, 103–112.

Temple Lang, D., Swayne, D., Wickham, H., & Lawrence, M. (2006). *rggobi: An Interface between R and GGobi*. http://www.R-project.org.

Therneau, T., & Atkinson, B. (2022). *Rpart: Recursive partitioning and regression trees*. https://CRAN.R-project.org/package=rpart

Theus, M. (2002). Interactive Data Visualization Using Mondrian. *Journal of Statistical Software*, *7*(11), http://www.jstatsoft.org.

Theus, M., Hofmann, H., & Wilhelm, A. F. X. (1998). Selection Sequences – Interactive Analysis of Massive Data Sets. *Computing Science and Statistics*, *29*(1), 439–444.

Thompson, G. L. (1993). Generalized Permutation Polytopes and Exploratory Graphical Methods for Ranked Data. *The Annals of Statistics*, *21*, 1401–1430.

Tierney, L. (1991). *LispStat: An Object-Orientated Environment for Statistical Computing and Dynamic Graphics*. John Wiley & Sons.

Tierney, N., & Cook, D. (2023a). Expanding tidy data principles to facilitate missing data exploration, visualization and assessment of imputations. *Journal of Statistical Software*, *105*(7), 1–31. https://doi.org/10.18637/jss.v105.i07

Tierney, N., & Cook, D. (2023b). Expanding tidy data principles to facilitate missing data exploration, visualization and assessment of imputations. *Journal of Statistical Software*, *105*(7), 1–31. https://doi.org/10.18637/jss.v105.i07

Tierney, N., Cook, D., McBain, M., & Fay, C. (2023). *Naniar: Data structures, summaries, and visualisations for missing data*. https://github.com/njtierney/naniar

Torgerson, W. S. (1952). Multidimensional Scaling. 1. Theory and Method. *Psychometrika*, *17*, 401–419.

Tufte, E. (1983). *The visual display of quantitative information*. Graphics Press.

Tufte, E. (1990). *Envisioning information*. Graphics Press.

Tukey, J. W. (1965). The Technical Tools of Statistics. *The American Statistician*, *19*, 23–28.

Unwin, A. R., Hawkins, G., Hofmann, H., & Siegl, B. (1996). Interactive Graphics for Data Sets with Missing Values - MANET. *Journal of Computational and Graphical Statistics*, *5*(2), 113–122.

Unwin, A., Hofmann, H., & Wilhelm, A. (2002). Direct Manipulation Graphics for Data Mining. *Journal of Image and Graphics*, *2*(1), 49–65.

Unwin, A., Theus, M., & Hofmann, H. (2006). *Graphics of Large Datasets: Visualizing a Million*. Springer.

Unwin, A., Volinsky, C., & Winkler, S. (2003). Parallel Coordinates for Exploratory Modelling Analysis. *Comput. Stat. Data Anal.*, *43*(4), 553–564. https://doi.org/{\tt http://dx.doi.org/10.1016/S0167-9473(02)00292-X}

Urbanek, S., & Theus, M. (2003). iPlots: High Interaction Graphics for R. In K. Hornik, F. Leisch, & A. Zeileis (Eds.), *Proceedings of the 3rd international workshop on distributed statistical computing (DSC 2003)*.

Vaidyanathan, R., Xie, Y., Allaire, J., Cheng, J., Sievert, C., & Russell, K. (2023). *Htmlwidgets: HTML widgets for r*. https://github.com/ramnathv/htmlwidgets

van der Maaten, L. J. P. (2014). Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, *15*, 3221–3245.

van der Maaten, L. J. P., & Hinton, G. E. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Vapnik, V. N. (1999). *The Nature of Statistical Learning Theory*. Springer.

Velleman, P. F., & Velleman, A. Y. (1985). *Data desk handbook*. Data Description, Inc.

Venables, W. N., & Ripley, B. (2002a). *Modern Applied Statistics with S*. Springer-Verlag.

Venables, W. N., & Ripley, B. D. (2002b). *Modern applied statistics with s* (Fourth). Springer. https://www.stats.ox.ac.uk/pub/MASS4/

Venables, W. N., & Ripley, B. D. (2002c). *Modern applied statistics with s* (Fourth). Springer. https://www.stats.ox.ac.uk/pub/MASS4/

Wainer, H. (2000). *Visual Revelations (2nd ed)*. LEA, Inc.

Wainer, H., & Spence, I. (eds). (2005a). *The Commercial and Political Atlas, Representing, by means of Stained Copper-Plate Charts, The Progress of the Commerce, Revenues, Expenditure, and Debts of England, during the whole of the Eighteenth Century, by William Playfair*. Cambridge University Press.

Wainer, H., & Spence, I. (eds). (2005b). *The Statistical Breviary; Shewing on a Principle entirely new, the resources of every state and kingdom in Europe; illustrated with Stained Copper-Plate Charts, representing the physical powers of each distinct nation with ease and perspicuity by William Playfair*. Cambridge University Press.

Wang, P. C. C. (Ed.). (1978). *Graphical Representation of Multivariate Data*. Academic Press.

Wegman, E. (1990). Hyperdimensional Data Analysis Using Parallel Coordinates. *Journal of American Statistics Association*, *85*, 664–675.

Wegman, E. J. (1991). *The Grand Tour in k-Dimensions* (Technical Report No. 68). Center for Computational Statistics, George Mason University.

Wegman, E. J., & Carr, D. B. (1993). *Statistical Graphics and Visualization* (C. R. Rao, Ed.; pp. 857–958). Elsevier Science Publishers.

Wegman, E. J., Poston, W. L., & Solka, J. L. (1998). Image Grand Tour. *Automatic Target Recognition VIII - Proceedings of SPIE, 3371*, 286–294.

Wehrens, R., & Buydens, L. M. C. (2007). Self- and super-organizing maps in R: The kohonen package. *Journal of Statistical Software*, *21*(5), 1–19. https://doi.org/10.18637/jss.v021.i05

Wehrens, R., & Kruisselbrink, J. (2018). Flexible self-organizing maps in kohonen 3.0. *Journal of Statistical Software*, *87*(7), 1–18. https://doi.org/10.18637/jss.v087.i07

Wehrens, R., & Kruisselbrink, J. (2023). *Kohonen: Supervised and unsupervised self-organising maps*. https://CRAN.R-project.org/package=kohonen

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H. (2022). *Classifly: Explore classification models in high dimensions*. http://had.co.nz/classifly

Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., Woo, K., Yutani, H., & Dunnington, D. (2023). *ggplot2: Create elegant data visualisations using the grammar of graphics*. https://CRAN.R-project.org/package=ggplot2

Wickham, H., & Cook, D. (2023). *Tourr: Tour methods for multivariate data visualisation*. https://github.com/ggobi/tourr

Wickham, H., Cook, D., & Hofmann, H. (2015). Visualizing statistical models: Removing the blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *8*(4), 203–225. https://doi.org/10.1002/sam.11271

Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2011a). Tourr: An R Package for Exploring Multivariate Data with Projections. *Journal of Statistical Software*, *40*(2). https://doi.org/10.18637/jss.v040.i02

Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2011b). tourr: An R package for exploring multivariate data with projections. *Journal of Statistical Software*, *40*(2), 1–18. https://doi.org/10.18637/jss.v040.i02

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation*. https://CRAN.R-project.org/package=dplyr

Wickham, H., Hester, J., & Bryan, J. (2023). *Readr: Read rectangular text data*. https://CRAN.R-project.org/package=readr

Wilhelm, A. F. X., Wegman, E. J., & Symanzik, J. (1999). Visual Clustering and Classification: The Oronsay Particle Size Data Set Revisited. *Computational Statistics: Special Issue on Interactive Graphical Data Analysis*, *14*(1), 109–146.

Wilkinson, L. (2005). *The grammar of graphics*. Springer.

Wills, G. (1999). NicheWorks – Interactive Visualization of Very Large Graphs. *Journal of Computational and Graphical Statistics*, *8*(2), 190–212.

Xie, Y., Hofmann, H., & Cheng, X. (2014). Reactive Programming for Interactive Graphics. *Statistical Science*, *29*(2), 201–213. https://doi.org/10.1214/14-STS477

Young, F. W., Valero-Mora, P. M., & Friendly, M. (2006). *Visual Statistics: Seeing Data with Dynamic Interactive Graphics*. John Wiley & Sons.

Zeileis, A., Fisher, J. C., Hornik, K., Ihaka, R., McWhite, C. D., Murrell, P., Stauffer, R., & Wilke, C. O. (2020). colorspace: A toolbox for manipulating and assessing colors and palettes. *Journal of Statistical Software*, *96*(1), 1–49. https://doi.org/10.18637/jss.v096.i01

Zeileis, A., Hornik, K., & Murrell, P. (2009). Escaping RGBland: Selecting colors for statistical graphics. *Computational Statistics & Data Analysis*, *53*(9), 3259–3270. https://doi.org/10.1016/j.csda.2008.11.033

Zhang, C., Ye, J., & Wang, X. (2023). A computational perspective on projection pursuit in high dimensions: Feasible or infeasible feature extraction.

International Statistical Review, *91*(1), 140–161. https://doi.org/10.1111/insr.12517

Zhang, H. S., Cook, D., Laa, U., Langrené, N., & Menéndez, P. (2021). Visual diagnostics for constrained optimisation with application to guided tours. *The R Journal*, *13*(2), 624–641. https://doi.org/10.32614/RJ-2021-105

Zhang, H. S., Cook, D., Laa, U., Langrené, N., & Menéndez, P. (2022). *Ferrn: Facilitate exploration of touRR optimisatioN*. https://github.com/huizezhang-sherry/ferrn/

Zhu, H. (2021). *kableExtra: Construct complex table with kable and pipe syntax*. https://CRAN.R-project.org/package=kableExtra

Index