Chapter 1

Table of contents

Overview

> Reading
>
> Course Notes - Chapter Introduction

Most quantitative research in the business and social sciences makes use of some kind of multivariate analysis. Research that considers only one variable at a time (a *univariate* analysis) can provide useful information – for example, about the average rate of inflation over time, the variability of a particular share's return, or the relative proportion of the population that hold a certain opinion – but it is usually in the consideration of *relationships* between two or more variables that the most interesting and useful information is to be found. For example, what other variables are related to increases in the inflation rate or the rise in the price of a particular share? Is it interest rates? Foreign exchange rates? And what causes people to prefer one opinion over another? Is it their education level? Income? The newspaper they

read? Simply put, any analysis that considers the relationship between two or more variables is a *multivariate* analysis.

The aim of this course and these notes is to cover some of the more popular methods for exploring multivariate data. The perspective that we will take when looking at these techniques will be to use the minimum amount of mathematics necessary for a solid understanding of the techniques and their interpretation. However, this does not mean "no mathematics"! Over the past twenty or so years, modern statistical software packages have made it possible to run all of the techniques that we'll cover in this course with a few clicks of a mouse, without knowing a single bit of mathematics and almost nothing about how the techniques themselves work. Clicking a mouse might give you results, but it is very difficult to know whether these results are reliable unless you know something about the underlying technique and what potential pitfalls exist. All statistical techniques, and particularly the multivariate ones, make some assumptions about the type and amount of data that should be collected and the aims of the researcher. If these are ignored, the results may not just be incorrect but misleading. In this case it would be better to put the output of an analysis in a rubbish bin than into a report or on a manager's desk. To get this understanding, a certain amount of mathematics is needed.

Having said that, the focus of the course is on the practical use and interpretation of the techniques in the analysis of real-world business and social research. The course is aimed at students who are specialising in some field of business or social science but who are not specialising in statistics, and so the techniques are illustrated mostly using real-world numerical examples rather than using mathematical arguments. The kind of statistics and mathematics that will used includes the following topics that have been covered in previous courses:

- Basic descriptive statistics (means, variances, absolute and relative frequencies, correlation)
- Hypothesis testing (z-test, t-test, F-test, χ^2-test of association)
- One-way analysis of variance and multiple linear regression
- Two-way crosstables (contingency tables), and their analysis using the χ^2 test of association
- Use of basic mathematical notation (summation' notation, vectors, matrices)

If you are unfamiliar with any of this material, it is important to go back and revise in the first few weeks of the course.

Outline of the course notes

Apart from this general introduction chapter, the course is divided into two parts of roughly equal size. In the first part, we look at techniques that a primarily ways of summarising large amounts of data and extracting its key meaning. Summarisation is concerned with taking a

large amount of data and condensing it into a simpler form that is easier to read and understand. Everyone is familiar with the idea of a "summary" section at the back of a textbook chapter, which gives the key ideas contained in the chapter. You can think of the techniques contained in the first part of the course as a summary section for numbers. We look at three techniques in Part I: correspondence analysis, factor analysis, and cluster analysis. Importantly, the techniques in Part I do not attempt to predict anything, nor explain any dependent variable. In fact, there is no dependent variable for any of the techniques in part one. This is left until Part II, which deals with what we will call "predictive" techniques. These techniques attempt to use one set of variables (the independent or predictor variables) to predict one of more other variables (the dependent or outcome variables). Multiple regression, which you would have covered in previous courses, is a typical example of a predictive technique, which we will briefly look at in this course in order to introduce other predictive models. The other techniques we will look at in Part II are: analysis of variance and covariance, discriminant analysis, classification trees, and structural equation modelling. Thus, over the course of the semester we aim to cover nine techniques which cover the bulk of multivariate analyses done in business- and social-research industries today.

Each technique is illustrated with a detailed example, with more concise descriptions of further examples given in the final part of each chapter, called "Further examples". These are intended as a basis for discussion in class or for self-study. For the practical implementation of the methods studied in this course, the R software package will be used. The R system is an open-source software project for analysing data and constructing graphics. It provides a general computer language for performing tasks like organizing data, statistical analyses, simulation studies, model fitting, building of complex graphics and many more. The R language was introduced in 1996, but in the first decade of the twenty-first century interest in R has exceeded all possible expectations. Apart from a well maintained core system with new releases every few months there are currently literally thousands of researchers contributing add-on packages on cutting-edge developments in statistics and data analysis. R is available in the Scilabs on campus. If you would like to install R on your own laptop / PC, go to the website http://www.R-project.org. To download R to your own computer: Navigate to ..../bin/windows/base and save the file R-x.0.x.-win.exe on your computer. Click this file to start the installation procedure and select the defaults unless you have a good reason not to do so. The core R system that is installed includes several packages. Apart from these installed packages several thousands of dedicated contributed packages are available to be downloaded by users in need of specific analyses. Many users of R prefer working with RStudio. This is a free and open source integrated development environment for R which works with the standard version of R available from CRAN (Comprehensive R Archive Network available at the website address given above). It can be downloaded from the RStudio home page www.rstudio.com to be run from your desktop (Windows, Mac or Linux). In this course we will be using the RStudio environment.

Before diving into the techniques, it is necessary to make sure that everyone is on the same mathematical footing, at least as far as basics are concerned. The remainder of this section revises some of the most important ideas behind matrices and the data filling them, and

describes how mathematical notation will be used in the remainder of the notes. It may be useful to have a quick read through these sections now to see how familiar you are with the content, and then to refer back to them as you read through the chapters to come.

Data and the data matrix

In earlier mathematics courses, you would have learned that a *matrix* is simply a rectangular or square arrangement of numbers. For example,

$$\mathbf{X} = \begin{bmatrix} 10 & 3 & 49 & 23 \\ 9 & 20 & 94 & 1 \end{bmatrix}$$

is a matrix. Specifically it is a matrix with dimension 2×4, or a "2×4 matrix" for short, because it has two rows and four columns. A *vector* is a special case of a matrix with only one row (called a row vector) or one column (called a column vector). Of course, any number on its own is also a special case of a matrix; one with one row *and* one column. This is called a *scalar*.

If we want to talk generally about matrices, without referring to specific number like in the matrix above, it is common to use x's in place of the numbers in the matrix e.g.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \end{bmatrix}$$

The whole matrix is usually denoted with a bold capital i.e. $\mathbf{X}$. It is important to realise that each x in the matrix above is simply a placeholder for a particular value to come. The first matrix (with the numbers) can *only* refer to one particular matrix, but the second one (with the x's) can be used to refer to *any* 2×4 matrix. Also, we can refer to any position in the $\mathbf{X}$ matrix using subscript notation.

Take x_{23} for example. The "23" is actually two subscripts put together (a "2" and a "3"). The first subscript (the "2") indicates what row the particular x we are interested in is in (the second row), and the second subscript (the "3") indicates what column the x is in (the third column). In the first matrix above, $x_{23} = 94$. Of course, if we have a vector, then there will be only one subscript, because all elements are in the same row (or column if it is a column vector). In fact, there is nothing stopping us having more than two subscripts too (for matrices of more than two dimensions), but we will not need to go this extra step for this course.

Why are we going over all this? Suppose that we have given out a survey and collected responses from 6 people on 5 questions. We can arrange these responses in the format of a table, as shown below:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \\ x_{61} & x_{62} & x_{63} & x_{64} & x_{65} \end{bmatrix}$$

We will call a table or matrix that is set up like this to contain data collected from a survey or some other piece of research a *data table* or a *data matrix*. There is not real difference between it an the other matrices we were talking about earlier, just a special application. The things that we are collecting data from (which could be people, shares, countries, animal species, songs ... anything you can collect data on) are called *cases* or *responses*. These appear as separate *rows* in the data matrix. The pieces of information that we use to describe each case are called *variables* or *attributes* and these appear in the *columns* of the data matrix. The x's, remember, are simply placeholders for values to come. Specifically, the values to come may be numbers, or they may be words. It is perfectly allowable for the first column of x's to be, for example, the first names of each person, e.g. $x_{11} =$ Iris. Of course, this will affect the type of analysis we can do later on that variable (for example, it wouldn't make sense to calculate a mean' first name).

`R` makes a distinction between matrices of numeric values and data frames containing all different types of data.

```
load("data/survey.data.RData")
survey.data
```

```
  Person Q1     Q2       Q3       Q4 Q5
1   John  6 639020 52.82732 21.91701 16
2  Sally  1 153860 47.71114 17.41023 14
3   Jane  4 138180 51.62570 21.96218 16
4    Tom  2 101360 48.69179 18.88657 13
5   Rick  2 564000 48.81529 19.48836 11
6    Amy  3 328830 49.02871 21.84143  9
```

Above we have a data frame called survey.data. We can ask `R` whether survey.data is a data frame with the function `is.data.frame()`:

```
is.data.frame(survey.data)
```

```
[1] TRUE
```

We can ask R whether survey.data is a matrix with the function is.matrix().

```
is.matrix(survey.data)
```

```
[1] FALSE
```

We can convert a data frame to a matrix with the function as.matrix(). Similarly, a matrix can be converted into a data frame with the function as.data.frame().

```
X <- as.matrix(survey.data[,-1])
X
```

```
     Q1     Q2       Q3       Q4 Q5
[1,]  6 639020 52.82732 21.91701 16
[2,]  1 153860 47.71114 17.41023 14
[3,]  4 138180 51.62570 21.96218 16
[4,]  2 101360 48.69179 18.88657 13
[5,]  2 564000 48.81529 19.48836 11
[6,]  3 328830 49.02871 21.84143  9
```

Above we have the matrix called X.

```
is.data.frame(X)
```

```
[1] FALSE
```

```
is.matrix(X)
```

```
[1] TRUE
```

Notice that when the matrix X was created from the data frame survey.data, the first column, containing non-numeric values were excluded. Has it not been excluded, all entries in the matrix will be converted to text, even the numeric values.

```
as.matrix(survey.data)
```

```
     Person  Q1  Q2       Q3         Q4         Q5
[1,] "John"  "6" "639020" "52.82732" "21.91701" "16"
[2,] "Sally" "1" "153860" "47.71114" "17.41023" "14"
[3,] "Jane"  "4" "138180" "51.62570" "21.96218" "16"
[4,] "Tom"   "2" "101360" "48.69179" "18.88657" "13"
[5,] "Rick"  "2" "564000" "48.81529" "19.48836" "11"
[6,] "Amy"   "3" "328830" "49.02871" "21.84143" " 9"
```

At this point it is probably worth spending a little time discussing different data types. There are two main types of data that we need to distinguish between: *numerical* variables, and *categorical* variables.

> **!** Important
>
> Numerical variables
> **Numerical variables** are measurements that can be recorded on a quantitative scale where the intervals between two values on the scale have some meaning. Essentially, this means that (a) the variable contains numbers rather than words or symbols, (b) the gaps between two numbers have some actual meaning. Examples of numerical variables are height, age, and number of children.
> **Categorical variables** are measurements of individuals in terms of groups or categories where the gap between categories have no intrinsic meaning. A typical example of a categorical variable is race, where the gap betweenblack' and white' has no proper interpretation, language, political affiliation, country of birth, and many other demographic variables.

It is vitally important to be able to distinguish between different data types because to a large extent these dictate what statistical techniques can be used. For example, it makes good sense to calculate the mean of a continuous variable but (as we have seen) no sense at all to calculate the mean of a categorical variable. The same idea extends to multivariate analysis. Some of the techniques we will look at work on correlation coefficients, which cannot be calculated for strictly categorical variables like race.

One further point on data types: some textbooks further divide numerical variables into *ratio-scaled* numerical variables and *interval-scaled* numerical variables; and divide categorical variables into *ordinal* categorical variables and *nominal* categorical variables. For the purposes of deciding which multivariate technique to use, this is an unnecessary detail and it is sufficient to know whether a variable is numerical or categorical. For the sake of completeness these additional terms are briefly described below. Ratio-scaled numerical variables are those that have a natural zero point (like age, height, and income). These are called "ratio-scaled" because the are not sensitive to units of measurement (if I am three times your height in meters I am also three times your height if it is measured in centimeters). This means that ratio-scaled variables have an arbitrary scale. Interval-scaled variables are still numeric but do not have a

natural zero point (IQ, temperature in degrees Celcius, and most Likert-type rating scales are of this type). Interval-scaled variables therefore have an arbitrary zero point *and* an arbitrary scale. Ordinal categorical variables are those where the categories can be ordered even if the gaps between them cannot be interpreted (such as level of education, which can be ordered: none, primary-school, high-school, undergraduate degree, postgraduate degree). In contrast, the categories of a nominal categorical variable cannot be ordered in any meaningful way (such as race or language group). It is also common to further classify numerical variables as *continuous* if they can take on any intermediate value on the scale (e.g. height) or *discrete* if the values a variable can take on are limited in some way (e.g. number of children).

Standardisation of data

When analysing numerical data, it often happens that different variables are measured on scales of very different sizes. For example, in the above matrix the first question might ask one how many children one has and the second question might ask for one's income in Rands. Clearly, the scale of possible values for the first question (between 0 and perhaps 15) is *much* smaller than for the second (between 0 and perhaps several million Rand). For reasons that will become clearer later on, this can cause enormous problems in some multivariate techniques by giving too much influence to the variables measured on larger scales. In order to put all variables on an equal footing, it is often necessary to *standardise* the data. Because several techniques require standardised data we consider it in this introductory chapter, but it is important to realise that *not all* the techniques need the data to be standardised. Moreover, in cases where all numerical variables are measured on the same scale (e.g.all on a 1 to 5 Likert rating scale) there will be no need to standardise either.

There are several different ways to standardise data, but the only one that we will use is to standardise the data so that each variable has a mean of zero and a standard deviation of one. In order to do this we carry out the following steps:

1. Calculate the mean and standard deviation of each variable in the data matrix (i.e. these are the column means and the column standard deviations)
2. Subtract each element in the data matrix by its column mean.
3. Divide the resulting "element minus mean" by its column standard deviation

We will illustrate the standardisation of a data matrix using the following example. Suppose that information on three variables (income, number of children, and age) has been collected from five individuals. The data is contained in the following table.

Table 1: Unstandardized Data with Summary Statistics

Person	Income	No Children	Age
a	10000	0	40
b	0	3	23
c	300000	2	32
d	150000	2	35
e	1000000	1	58
$\bar{x}$	292000	1.6	37.6
s	414210	1.140	12.973

where we use the usual mathematical notation $\bar{x}$ to denote the mean and s to denote the standard deviation. Note that the variables are measured on very different scales. To standardise the data, we simply follow the steps above. For example, the standardised income of person a is given by

$$\frac{10\,000 - 292\,000}{414\,210} = -0.681$$

to three decimal places. Similarly the standardised number of children for person d is given by $(2 - 1.6)/1.140 = 0.351$. You can check for yourself that the *new* column means and standard deviations are all zero and one respectively. Since the mean of all the variables is zero, it is possible to see at a glance which observations are below average (those that are negative) and which are above average (those that are positive).

Person	Standardised data		
	Income	N.Children	Age
a	-0.681	-1.403	0.185
b	-0.705	1.228	-1.125
c	0.019	0.351	-0.432
d	-0.343	0.351	-0.200
e	1.709	-0.526	1.572

The relevance of standardising data may not seem clear to you at the moment. Just bear this section in mind as you continue through the notes and refer back to it when the issue of standardisation reappears.

```
X <- matrix (c(10000,0,300000,150000,1000000,0,3,2,2,1,40,23,32,35,58), ncol=3,
             dimnames=list(c("a","b","c","d","e"),
                           c("Income","No Children","Age")))
```

In R we can create a matrix with the ***matrix()*** *function. The values in the matrix are concatenated with the operator* `c()`*. Notice that the values needs to be entered column wise by default. The names for the two dimensions are specified by* ***dimnames=list("row names", "column names")****. Notice below that the row names appear to the left.*

They are text, but are not part of the CONTENT of the matrix. The matrix X:5 × 3 contains only numeric values.

```
X
```

```
   Income No Children Age
a   10000           0  40
b       0           3  23
c  300000           2  32
d  150000           2  35
e 1000000           1  58
```

To calculate the means we apply to X, column wise (indicated by 2; 1 for row wise) the function `mean()`.

```
xbar <- apply(X,2,mean)
xbar
```

```
     Income No Children         Age
   292000.0         1.6        37.6
```

Similarly, the function `sd()` is applied to each column to calculate the standard deviations.

```
s <- apply(X,2,sd)
s
```

```
      Income  No Children          Age
4.142101e+05 1.140175e+00 1.297305e+01
```

Any numeric calculations can be performed by simply typing the expression at the R command prompt "`>`".

```
(10000-292000)/414210
```

```
[1] -0.6808141
```

R has the ability to operate on a whole vector (or matrix) at once. Here the standardised values for Age is calculated by subtracting the mean from the values in column 2 and dividing resulting "column minus mean" by the standard deviation.

```
(X[,2]-1.6)/1.14
```

```
         a          b          c          d          e
-1.4035088  1.2280702  0.3508772  0.3508772 -0.5263158
```

The expressions above is simply for illustration purposes. The function `scale()` performs all the standardisation calculations in a single step. The output is again a matrix of size 5×3, but additional attributes are provided: first the mean called "`scaled:center`", then the standard deviations called "`scaled:scale`".

```
scale(X)
```

```
       Income No Children        Age
a -0.68081393  -1.4032928  0.1849989
b -0.70495627   1.2278812 -1.1254101
c  0.01931387   0.3508232 -0.4316641
d -0.34282120   0.3508232 -0.2004155
e  1.70927752  -0.5262348  1.5724908
attr(,"scaled:center")
     Income No Children         Age
   292000.0         1.6        37.6
attr(,"scaled:scale")
      Income  No Children          Age
4.142101e+05 1.140175e+00 1.297305e+01
```

Mathematical notation

Let us begin by taking another look at the general $\mathbf{X}$ matrix from earlier in the chapter.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \\ x_{61} & x_{62} & x_{63} & x_{64} & x_{65} \end{bmatrix}$$

We have already discussed the use of subscript notation, using the example of x_{23} – where the first subscript that the x we are interested in is in the second row and the third column of the data matrix. We can make this one step more general by referring to a general subscript i for the rows and j for the columns, to give x_{ij}. In the same way that the x's are just placeholders for value to come, so is i and so is j. Thus, we can insert any value from 1 to 6 into i and any value from 1 to 5 into j, and refer to a specific element of $\mathbf{X}$.

This becomes important because we don't want to have to write out the whole matrix every time we want to do something with $\mathbf{X}$. The general i and j subscripts allow us to be much more concise. For example, suppose that the x's are scores on 5 different tests. The score achieved by student i on test j is given by x_{ij}. Suppose now that we want to find student 3's average mark, which we label as $\bar{x}_3$. In order to do this we need to add up all 5 test scores and divide by 5. We could write

$$\bar{x}_3 = \frac{x_{31} + x_{32} + x_{33} + x_{34} + x_{35}}{5}$$

or we could write

$$\bar{x}_3 = \frac{1}{5} \sum_{j=1}^{5} x_{3j}$$

The summation sign (Σ) indicates that we add up all the terms following the sign, by letting j take each of the values in turn between the "limits of summation" (which are 1 and 5 respectively). In this case, it does not save much time or space to use summation notation, but in some cases it does. For example, if we now want refer to the average test score of *any* of the individuals in our data set, we have two choices: either use the "full" notation and write

out all the averages

$$\begin{aligned}
\bar{x}_1 &= (x_{11} + x_{12} + x_{13} + x_{14} + x_{15})/5 \\
\bar{x}_2 &= (x_{21} + x_{22} + x_{23} + x_{24} + x_{25})/5 \\
\bar{x}_3 &= (x_{31} + x_{32} + x_{33} + x_{34} + x_{35})/5 \\
\bar{x}_4 &= (x_{41} + x_{42} + x_{43} + x_{44} + x_{45})/5 \\
\bar{x}_5 &= (x_{51} + x_{52} + x_{53} + x_{54} + x_{55})/5 \\
\bar{x}_6 &= (x_{61} + x_{62} + x_{63} + x_{64} + x_{65})/5
\end{aligned}$$

or use the other general subscript i and write the average test score of person i is given by

$$\bar{x}_i = \frac{1}{5}\sum_{j=1}^{5} x_{ij} \qquad (i = 1, \ldots, 6)$$

where the $(i = 1, \ldots, 6)$ part indicates that i can take on any value from 1 to 6. The compactness of the summation notation is clear to see. Now, suppose that we want to weight the different test scores differently. This is typically what happens in the calculation of a year mark. Suppose that the x's are marks are 5 class tests, which are count w_1, w_2, w_3, w_4, w_5 towards the final mark. Note that we have introduced another variable here, w_j, to denote the weight attached to test j. Then the year mark obtained by student i is given by

$$\bar{x}_i = \frac{w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + w_4 x_{i4} + w_5 x_{i5}}{5} \qquad (i = 1, \ldots, 6)$$

or we could just write

$$\bar{x}_i = \frac{1}{5}\sum_{j=1}^{5} w_j x_{ij} \qquad (i = 1, \ldots, 6)$$

Note that the w's only have a j subscript because they do not differ over students (the weighting is the same for all students) and so do not need an i subscript. Suppose that the weightings *did* differ over students (say because the weights needed to be adjusted if students miss a test for medical reasons). Then the w's would be able to differ over students, we would need to include a subscript i, and we would have an average mark given by

$$\bar{x}_i = \frac{1}{5}\sum_{j=1}^{5} w_{ij} x_{ij} \qquad (i = 1, \ldots, 6)$$

Try writing out the average year mark for student 4 as a practice exercise. Finally, suppose that we want to work out a class average, labelled C. For this we need to add together each student's average mark $\bar{x}_i$ and then divide by the number of students in the class, 6. This can be written as

$$C = \sum_{i=1}^{6} \bar{x}_i = \frac{1}{6}\sum_{i=1}^{6}\frac{1}{5}\sum_{j=1}^{5} w_{ij} x_{ij} = \frac{1}{30}\sum_{i=1}^{6}\sum_{j=1}^{5} w_{ij} x_{ij} \qquad (i = 1, \ldots, 6)$$

Once again, try writing this out in full as an exercise and to see the usefulness of the summarised notation!

Introduction to Singular Value Decomposition (SVD)

There is one result from matrix algebra that we will use extensively in the methods discussed in this course. Without going into the detail of the mathematics, the **singular value decomposition (SVD)** can be viewed as a "black box". The results of the SVD can be stated in terms of matrices or individual elements of a matrix.

Mathematical Definition

Any matrix $\mathbf{X}$ can be expressed as the product of three matrices $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}'$. The ij-th element of $\mathbf{X}$, denoted as x_{ij}, is expressed as:

$$x_{ij} = \sum_{k=1}^{r} u_{ik} d_k v_{jk}$$

The values d_k have only one subscript because the matrix $\mathbf{D}$ is a **diagonal matrix** with zeros for all off-diagonal elements.

If $\mathbf{X}$ is the matrix of standardised data (e.g., Income, Number of Children, and Age), then $r = 3$, and $\mathbf{D}$ will contain three diagonal values. These values in $\mathbf{D}$ are called the **singular values**. Singular values are always non-negative ($d_k \geq 0$), and we order them such that $d_1 \geq d_2 \geq ... \geq d_r$.

Dimension Reduction

The SVD is the basis for approximating multivariate data by **dimension reduction**. Working with too many variables makes it difficult to discern interrelationships. We often seek a matrix $\mathbf{X}^*$ that is "simpler" than $\mathbf{X}$ but remains a good approximation.

Least Squares Approximation

We aim to find the least squares solution $\mathbf{X}^*$ that minimizes the sum of squared differences between the elements of $\mathbf{X}$ and $\mathbf{X}^*$:

$$\min \sum_i \sum_j (x_{ij} - x_{ij}^*)^2$$

According to **Huygens' Principle**, the approximation necessarily includes the centroid (mean), so we center the data matrix before approximating. If the data is already standardised, it is already centered.

Best 2D approximation: $\mathbf{X}^* = \sum_{k=1}^{2} u_{ik} d_k v_{jk}$ **Best 1D approximation:** $\mathbf{X}^* = u_{i1} d_1 v_{j1}$

R Programming Exercise

Use the interactive console below to perform the SVD on a standardized dataset.

```r
# Step 1: Create the sample data matrix X
X <- matrix(c(50, 2, 35,
              20, 4, 45,
              40, 3, 30,
              35, 3, 32,
              60, 5, 25),
            nrow = 5, byrow = TRUE)
colnames(X) <- c("Income", "Children", "Age")

# Step 2: Standardise the matrix
X.std <- scale(X)

# Step 3: Compute SVD
res.svd <- svd(X.std)

# Extract components
U <- res.svd$u
D <- diag(res.svd$d)
V <- res.svd$v

# Step 4: Construct the best 2-dimensional approximation (X.star)
# We use only the first two columns/elements
```

```r
X.star <- U[, 1:2] %*% D[1:2, 1:2] %*% t(V[, 1:2])

# View results
print("Standardized Matrix:")
```

```
[1] "Standardized Matrix:"
```

```r
print(X.std)
```

```
          Income   Children        Age
[1,]  0.59344243 -1.2278812  0.2151580
[2,] -1.38469899  0.5262348  1.5598952
[3,] -0.06593805 -0.3508232 -0.4572107
[4,] -0.39562828 -0.3508232 -0.1882632
[5,]  1.25282290  1.4032928 -1.1295793
attr(,"scaled:center")
  Income Children      Age
    41.0      3.4     33.4
attr(,"scaled:scale")
   Income  Children       Age
15.165751  1.140175  7.436397
```

```r
print("2D Approximation (X.star):")
```

```
[1] "2D Approximation (X.star):"
```

```r
print(X.star)
```

```
            [,1]       [,2]       [,3]
[1,]  0.26048350 -1.2630881 -0.1244430
[2,] -1.51239121  0.5127327  1.4296557
[3,]  0.21160127 -0.3214764 -0.1741349
[4,] -0.09109489 -0.3186221  0.1223452
[5,]  1.13140134  1.3904538 -1.2534230
```

A word of caution on practical data analysis

One of the main aims of this course is to put you in a position of being able to perform the multivariate statistical analysis of your own research projects, in whatever field this may be. Most of the examples used in these notes are themselves real-world studies, and so you will get some idea of some of the complexities involved in gathering and analysing data. Having said that, there is an obvious need in an introductory course like this one to choose data sets that work' and that can be used to illustrate the techniques. We therefore do not discuss many of the practical difficulties which inevitably arise when doing your own original research. As a result when these difficulties arise when it comes to doing your own research, you may look back on this course and think why weren't we taught that?' Unfortunately, the kinds of problems that can arise are so varied and require such different solutions that it is not possible to teach in a course such as this one. As Bartholemew et al. put it, only when one has a clear idea of where one is going is it possible to know the important questions which arise". However, the following broad areas should be borne in mind whenever conducting an original analysis.

i Missing Data

Missing data can cause severe problems for many of the techniques we will consider. Most techniques will simply drop cases which possess missing data on *any* of the variables to be included in the analysis. When the number of variables is large, as is often the case in multivariate analyses, this can result in a substantial proportion of the sample being dropped. This proportion should always be noted early in the analysis. Another critical question to ask iswhy is the data missing?" and 'does the missing data introduce any bias into the results?" Often, it is the people with the most extreme views that turn up as missing data by refusing to answer certain questions, which is clearly biasing. Possible solutions are *mean replacement* or other *imputation* (replacement) techniques, but these are beyond the scope of this course.

! Sample Sizes

It is a general rule that the bigger the model you fit, the greater the number of cases you need. In univariate analysis and simple hypothesis testing, the calculation ofrequired' sample sizes is reasonably straightforward, but in multivariate analysis there are only very rough guidelines where any exist at all. As a *very* rough guideline, most techniques require at least 10 respondents per parameter estimated. That means that in order to estimate a regression model with four independent variable, you need at least 50 respondents (not forgetting the constant term β_0, there are 5 parameters to be estimated). When sample sizes are small, one should be very careful about drawing strong conclusions. This is a particular problem in student research, where sample sizes are typically very small.

Transformations

Many statistical techniques assume that data are normally distributed. Although it is again beyond the scope of this course, it is often possible to transform data that is not normally distributed into something that *is* normally distributed by using some kind of transforming function. Taking the logarithm of a set of numbers, for example, often works, as does taking the square (both of these transformations work by sucking in' the tails of the non-normal distributions). Where transformations do not help, the analyst must make a decision about whether the data is approximately normal' or 'normal enough' to continue, or whether it is necessary to use other methods (like non-parametric statistics, which tend to be harder to use but do not make any distributional assumptions).