Advanced Quantitative Methods
Math preliminaries

Johan A. Elkink
jos.elkink@ucd.ie

December 20, 2010

If you encounter any mistakes in this text, please inform the author.

Contents

1 Introduction

This document provides a brief overview of the mathematical skills that are useful for the Advanced Quantitative Methods course. The contents are primarily intended as a reference, rather than as material to learn by heart. This is also the reason that in many sections there is a list of rules that you can apply, without much discussion of why these rules apply. They are simply there as reference when dealing with matrices, or expectations, or variances in equations.

The contents of Sections 2 and 3 are primarily based on Magnus and Neudecker (1999, 3-11).

2 Vectors

2.1 Definition

A **vector** is most easily perceived as just a list of values. For example, we might have a vector $\mathbf{v}$ with values 3, 2 and 6, which we would write as $\mathbf{v} = \begin{bmatrix} 3 & 2 & 6 \end{bmatrix}$. The order of the values matters, so if $\mathbf{w} = \begin{bmatrix} 3 & 6 & 2 \end{bmatrix}$, then $\mathbf{v}$ is different from $\mathbf{w}$, i.e. $\mathbf{v} \neq \mathbf{w}$. A vector can have any dimension and any kind of values. Let us take as a running example $\mathbf{x} = \begin{bmatrix} 8 & -2 & 6 & -8 & 5 & 15 & -5 & -16 \end{bmatrix}$. In R you could enter this with:

```
x <- c(8, -2, 6, -8, 5, 15, -5, -16)
```

The c here stands for "**concatenate**", i.e. concatening the collection of values between parentheses into one vector.[1]

We often use **subscripts** to indicate a specific element in a vector. In our running example, we could say, e.g., $x_4 = -8$. Or more generally, we can say that x_i indicates the ith element in $\mathbf{x}$, so if $i = 2$ this is -2, if $i = 7$ this is -5, etc. To indicate that we use i in this fashion, we can write $\mathbf{x} = [x_i]_{n \times 1}$, which means $\mathbf{x}$ is a vector of which a typical element is x_i, and which has n elements. In R, we use square brackets to indicate a particular element, for example:

```
x[5]
```

or more generically:

```
i <- 3
x[i]
```

The latter can be used inside a loop:

```
for (i in 1:8) {
  cat(x[i], "squared is", x[i]^2, "\n")
}
```

We usually also have an indicator of the dimension of the vector, for example n. We might say that, because there are 8 elements in vector $\mathbf{x}$, $n = 8$. To make explicit that we use $\mathbf{x}$ as the name of the vector, i as the indicator for the element and n as the dimension, we would write $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & ... & x_i & ... & x_n \end{bmatrix}$.

[1] This note will assume that the reader has basic familiarity with R syntax and will not elaborate on the form R commands take in general.

Although in many contexts we do not need to specify this, vectors can be either row vectors or column vectors. This will become a little clearer when we are talking about matrices, below. A **column vector** looks like this:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

while a **row vector** looks like this:

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}$$

We can transpose one into the other, usually indicated with a ór a T, as in $\mathbf{x}'$ or $\mathbf{x}^T$. So, the **transpose** of row vector $\mathbf{v}$ looks like:

$$\mathbf{v}' = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

When not specified, we assume column vectors. R also assumes a column vector and to get the row vector, we can use the t function:

```
t(x)
```

2.2 Summation

An often used symbol in statistics classes or textbooks is the **summation** symbol, $\sum$. Losely translated, this means "sum up this series of numbers". For example, if I have a vector $\mathbf{x} = \begin{bmatrix} 3 & 2 & 5 & 1 & 3 \end{bmatrix}$, then $\sum_i x_i = 3+2+5+1+3 = 14$. In R, this example would be:

```
x <- c(3,2,5,1,3)
sum(x)
```

Usually, **subscripts** are used to indicate exactly what series of elements are summed. For example, if we say that we use i as the indicator for the elements of $\mathbf{x}$, as in when $i = 3$ then $x_i = x_3 = 5$, then we would use the notation $\sum_i x_i$. If we also specify the number of elements as, say, n, so $n = 5$ because $\mathbf{x}$ has 5 elements, then we can write $\sum_{i=1}^{n} x_i$ to mean the sum of the elements of $\mathbf{x}$, counting from $i = 1$ to n.

2.3 Addition and subtraction

When two vectors are of the same size, they can be added or subtracted. We do this by simply adding or subtracting the individual elements. For example, if $\mathbf{v} = \begin{bmatrix} 4 & 2 & 3 & 1 \end{bmatrix}'$ and $\mathbf{w} = \begin{bmatrix} 5 & 2 & 1 & 1 \end{bmatrix}'$, then

$$\mathbf{v} + \mathbf{w} = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 5 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4+5 \\ 2+2 \\ 3+1 \\ 1+1 \end{bmatrix} = \begin{bmatrix} 9 \\ 4 \\ 4 \\ 2 \end{bmatrix}$$

and $\mathbf{v} - \mathbf{w} = \begin{bmatrix} 4-5 & 2-2 & 3-1 & 1-1 \end{bmatrix}' = \begin{bmatrix} -1 & 0 & 2 & 0 \end{bmatrix}'$. In R this would be:

```
v <- c(4,2,3,1)
w <- c(5,2,1,1)
v + w
v - w
```

Note that this implies that $\mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v}$, but $\mathbf{v} - \mathbf{w} \neq \mathbf{w} - \mathbf{v}$.

2.4 Multiplication with scalar

In statistics, we do not often add or subtract vectors. We do often multiply them, however. There are three ways of multiplying vectors: multiplying with a **scalar** (a single value) and two forms of multiplying vectors with each other, the inner and the outer products.

When we multiply a vector with a scalar, each element is multiplied by the same value. For example, if we take $\mathbf{v}$ from the example above, then

$$4\mathbf{v} = 4 \cdot \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \cdot 4 \\ 4 \cdot 2 \\ 4 \cdot 3 \\ 4 \cdot 1 \end{bmatrix} = \begin{bmatrix} 16 \\ 8 \\ 12 \\ 4 \end{bmatrix}$$

And in R:

```
4 * v
```

More abstractly, if we have a vector $\mathbf{x} = [x_i]_{n\times 1}$, so we have a vector $\mathbf{x}$ which consists of elements indicated by x_i, then we can say that for any scalar a, $a\mathbf{x} = [ax_i]_{n\times 1}$. Note that if in this case $a = -1$, we get the negative values of $\mathbf{x}$. We can also simply write this as $-1\mathbf{x} = -\mathbf{x}$. And combined with the previous topic, addition: $\mathbf{v} - \mathbf{w} = \mathbf{v} + (-1)\mathbf{w} = (-1)\mathbf{w} + \mathbf{v} = -\mathbf{w} + \mathbf{v} = -(\mathbf{w} - \mathbf{v})$. This is all obvious with regular numbers, but it is worth verifying that the same holds for vectors.

2.5 Inner product

If two vectors are of the same dimension, we can multiply them together. The **inner product** is a product of a row vector with a column vector, while the outer product is the product of a column vector with a row vector. One cannot multiply two row vectors or two column vectors - but we can of course use the transpose to transform one into the other. So, if we have vectors $\mathbf{v} = \begin{bmatrix} 1 & 4 & 2 & 3 \end{bmatrix}'$ and $\mathbf{w} = \begin{bmatrix} 5 & 4 & 1 & \frac{1}{2} \end{bmatrix}'$, then we can get the inner product $\mathbf{v}'\mathbf{w}$. The óperator transforms the column vector $\mathbf{v}$ into a row vector, so now we multiply row vector $\mathbf{v}'$ with column vector $\mathbf{w}$, resulting in the inner product. The inner product is calculated by adding the products of the individual elements. Abstractly, if $\mathbf{v} = [v_i]_{n\times 1}$ and $\mathbf{w} = [w_i]_{n\times 1}$, then $\mathbf{v}'\mathbf{w} = \sum_i (v_i \cdot w_i)$. So, $\mathbf{v}'\mathbf{w} = 1 \cdot 5 + 4 \cdot 4 + 2 \cdot 1 + 3 \cdot \frac{1}{2} = 5 + 16 + 2 + 1\frac{1}{2} = 24\frac{1}{2}$. In R:

```
v <- c(1,4,2,3)
w <- c(5,4,1,.5)
t(v) %*% w
```

Note that this implies that $\mathbf{v}'\mathbf{w} = \mathbf{w}'\mathbf{v}$. An inner product always results in a scalar, a single value.

The length of a vector $\mathbf{v}$, written as $\|\mathbf{v}\|$, is defined as $\|\mathbf{v}\| = (\mathbf{v}'\mathbf{v})^{1/2}$, which is also called the Euclidean norm.

An interesting typical form in which we encounter this is the inner product of a vector and its transpose, for example $\mathbf{x}'\mathbf{x}$. If $\mathbf{x} = [x_i]_{n\times 1}$ then $\mathbf{x}'\mathbf{x} = \sum_i (x_i \cdot x_i) = \sum_i x_i^2$. This is, for obvious reasons, called the **sum of squares** of $\mathbf{x}$. So $\mathbf{v}'\mathbf{v} = 1^2 + 4^2 + 2^2 + 3^2 = 1 + 16 + 4 + 9 = 30$, or in R:

```
t(v) %*% v
```

or perhaps more simply:

```
sum(v^2)
```

Since in typical **linear regression** models, the goal is to minimize the sum of squares of the errors, you can see how this is a relevant inner product.

2.6 Outer product

The **outer product** gets us into matrices, so this forms a nice bridge to the next section. If we have vectors $\mathbf{v} = \begin{bmatrix}6 & 2 & 4\end{bmatrix}'$ and $\mathbf{w} = \begin{bmatrix}0.5 & 1.5 & 2\end{bmatrix}'$ then we have an outer product $\mathbf{v}\mathbf{w}'$, which looks like this:

$$\mathbf{v}\mathbf{w}' = \begin{bmatrix}6\\2\\4\end{bmatrix} \cdot \begin{bmatrix}0.5 & 1.5 & 2\end{bmatrix} = \begin{bmatrix}6\cdot 0.5 & 6\cdot 1.5 & 6\cdot 2\\ 2\cdot 0.5 & 2\cdot 1.5 & 2\cdot 2\\ 4\cdot 0.5 & 4\cdot 1.5 & 4\cdot 2\end{bmatrix} = \begin{bmatrix}3 & 9 & 12\\ 1 & 3 & 4\\ 2 & 6 & 8\end{bmatrix}$$

Or in R:

```
v <- c(6,2,4)
w <- c(0.5,1.5,2)
v %*% t(w)
```

Note that $\mathbf{v}\mathbf{w}' \neq \mathbf{w}\mathbf{v}'$, but that $\mathbf{v}\mathbf{w}' = (\mathbf{w}\mathbf{v}')'$ (you can see this easiest if you try $\mathbf{v}\mathbf{w}'$ and $\mathbf{w}\mathbf{v}'$ in R) - this and anything else we can say about this product is about matrices, however, so let us turn to that topic.

3 Matrices

3.1 Definition

A matrix is a collection of values organised in rows and columns. For example, we could have a matrix $\mathbf{X}$ like this:

$$\mathbf{X} = \begin{bmatrix}4 & 2 & 1 & 0\\ 5 & 0.5 & 0.25 & 1\\ 0 & 0 & 2 & 2\end{bmatrix}$$

This would be a matrix with 3 rows and 4 columns, which we say to be of dimension 3×4, written as $\mathbf{X}_{3\times 4}$. Note that we *always* say row first, column second. Also note that vectors are, of course, special types of matrices, with

either just one row or just one column - hence all rules applicable to matrices also apply to vectors.[2]

When we use **subscripts**, we use the same order, row then columns. So with x_{ij} we refer to the element in $\mathbf{X}$ located at the intersection of the ith row and the jth column, e.g. $x_{23} = 0.25$. We can use the same typical element formulation as with vectors: $\mathbf{X} = [x_{ij}]_{n\times k}$. So if we say $\mathbf{M}_{n\times k} = [m_{ij}]_{n\times k}$, then we are saying we have a matrix called $\mathbf{M}$, with n rows, k columns, and that we are using i as an indicator for the row and j as an indicator for the column.

One way of looking at a matrix is to see it as a collection of **row vector**s or a collection of **column vector**s, bound together. $\mathbf{X}$ above can be seen as a combination of the column vectors $\mathbf{x}_{\bullet 1} = \begin{bmatrix}4 & 5 & 0\end{bmatrix}'$, $\mathbf{x}_{\bullet 2} = \begin{bmatrix}2 & 0.5 & 0\end{bmatrix}'$, $\mathbf{x}_{\bullet 3} = \begin{bmatrix}1 & 0.25 & 2\end{bmatrix}'$ and $\mathbf{x}_{\bullet 4} = \begin{bmatrix}0 & 1 & 2\end{bmatrix}'$, for example. In R, this is also the easiest way to manually enter a matrix, using cbind to combine a collection of vectors into a matrix as if they are column vectors, and rbind to combine a collection of vectors into a matrix as if they are row vectors:

```
x1 <- c(4,5,0)
x2 <- c(2,.5,0)
x3 <- c(1,.25,2)
x4 <- c(0,1,2)
X <- cbind(x1, x2, x3, x4)
```

or just directly:

```
X <- cbind(c(4,5,0), c(2,.5,0), c(1,.25,2), c(0,1,2))
```

In R, the indexing of matrices is in the same order as in mathematics in general, rows first, columns second, using a comma to separate the two. So we can access x_{23} with:[3]

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

By leaving out either the row or the column number, you can get the respective column or row vector, respectively. For example, the second column vector of $\mathbf{X}$ is:

```
X[,2]
```

Note that the result is always a column vector, so:

```
X[3,]
```

returns a column vector, the elements of which represent the 3rd row of $\mathbf{X}$.

The **transpose** of a matrix turns all column vectors into row vectors and vice versa. So we get:

$$\mathbf{X}' = \begin{bmatrix}4 & 5 & 0\\ 2 & 0.5 & 0\\ 1 & 0.25 & 2\\ 0 & 1 & 2\end{bmatrix}$$

[2]Of course, some rules only apply to specific types of matrices, whereby vectors might per definition not be of that type, hence these rules would not apply to vectors. All rules that apply to matrices in general, apply to vectors.

[3]I consistently use normal typeface for a single value (e.g. x), bold for a vector (e.g. $\mathbf{x}$), and a capitalized bold typeface for a matrix (e.g. $\mathbf{X}$). So an element, which is a single value, of matrix $\mathbf{X}$ is x_{ij}. When accessing a single element in R, however, R needs to know the matrix name itself. So if I use a capital X as the name of the matrix, then I also need to use the capital X when accessing an individual element, as in this example.

Country	*GDP*	*Gini*	*Debt*	*System*
Netherlands	673.5	30.9	58.2	PR
United Kingdom	2236	34	51.8	Maj
Ireland	189	32	44.2	PR
Germany	2925	27	66	PR
Sweden	345.1	23	36.7	PR

Table 1: Mini data set. Source: *CIA World Factbook*, https://www.cia.gov/library/publications/the-world-factbook. The GDP is measured in billions of dollars, 2008 estimate, at purchasing power parity. The public debt is measured as percentage of GDP in 2008. System refers to the political system, divided in proporational (PR) and majoritarian (Maj) systems.

3.2 Example data matrix

A very typical definition of the **data matrix** in a linear regression model is $\mathbf{X}_{n\times k} = [x_{ij}]_{n\times k}$, which implies that we have n cases in our data set (respondents of a survey, countries, wars, or whatever else is our unit of analysis that we collected data on), k variables that we measured on those cases, and that we will use i to indicate the case and j to indicate the variable. So to put it in a slightly more politics and economic context, imagine we have the data set provided in Table 1. Since we can only store numbers in a data set, we will turn the last variable into a dummy variable. A **dummy variable** takes only the values 0 and 1 and is often used for categorical variables in regression models. We can define the following data matrix:

$$\mathbf{X} = \begin{bmatrix} 673.5 & 30.9 & 58.2 & 1 \\ 2236 & 34 & 51.8 & 0 \\ 189 & 32 & 44.2 & 1 \\ 2925 & 27 & 66 & 1 \\ 345.1 & 23 & 36.7 & 1 \end{bmatrix}$$

In this particular example, $k = 4$ and $n = 5$. Typically, we also want to add a constant to the model:

$$\mathbf{X} = \begin{bmatrix} 1 & 673.5 & 30.9 & 58.2 & 1 \\ 1 & 2236 & 34 & 51.8 & 0 \\ 1 & 189 & 32 & 44.2 & 1 \\ 1 & 2925 & 27 & 66 & 1 \\ 1 & 345.1 & 23 & 36.7 & 1 \end{bmatrix}$$

so that $k = 5$. In R:

```
X <- cbind(1,
           c(673.5, 2236, 189, 2925, 345.1),
           c(30.9, 34, 32, 27, 23),
           c(58.2, 51.8, 44.2, 66., 36.7),
           c(1,0,1,1,1))
```

Note that you can add names to columns and rows of a matrix, if this helps the interpretation:

```
colnames(X) <- c("Constant","GDP","Gini","Debt","System")
rownames(X) <- c("NL","UK","IE","DE","SE")
```

You can now use both indicators or names to get a particular value or vector:

```
X[3,2]
X["NL","Gini"]
X["IE",]
X[,"Debt"]
```

Usually, data sets in R are stored in data frames rather than in matrices. One has to convert to a matrix, however, if one wants to do manual calculations or regressions. To get the dimensions of a matrix, we can use the dim function in R:

```
dim(X)
n <- dim(X)[1]
k <- dim(X)[2]
```

3.3 Addition and subtraction

The addition and subtraction of matrices is very similar to that of vectors, namely just an element-by-element addition or subtraction. Matrices need to have the exact same dimensions to be able to add or subtract. For example:

$$\begin{bmatrix} 3 & 2 \\ 1 & 4 \\ 5 & 7 \end{bmatrix} + \begin{bmatrix} 1 & 0.5 \\ 1 & 0.5 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 2.5 \\ 2 & 4.5 \\ 7 & 10 \end{bmatrix}$$

Or, in other words, if $\mathbf{X} = [x_{ij}]_{n\times k}$ and $\mathbf{M} = [m_{ij}]_{n\times k}$, then $\mathbf{X} + \mathbf{M} = [x_{ij} + m_{ij}]_{n\times k}$ and $\mathbf{X} - \mathbf{M} = [x_{ij} - m_{ij}]_{n\times k}$. Hence, $\mathbf{X} + \mathbf{M} = \mathbf{M} + \mathbf{X}$ and $\mathbf{X} - \mathbf{M} \neq \mathbf{M} - \mathbf{X}$. Note that:

$$(\mathbf{A}')' = \mathbf{A}$$
$$(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$$

3.4 Multiplication with scalar

The multiplication of a matrix with a scalar is, again, very similar to vectors. So, if $\mathbf{X} = [x_{ij}]_{n\times k}$, then $a\mathbf{X} = [ax_{ij}]_{n\times k}$. We can derive from this that $a(\mathbf{X}+\mathbf{M}) = a\mathbf{X} + a\mathbf{M}$ and the same with subtraction. Furthermore, $(a + b)\mathbf{X} = a\mathbf{X} + b\mathbf{X}$ and $a(b\mathbf{X}) = (ab)\mathbf{X}$. These rules are easy to derive, so you might want to do so as an exercise. Again, $-\mathbf{X} = (-1)\mathbf{X} = [-x_{ij}]_{n\times k}$.

3.5 Multiplication

Multiplying matrices with each other is in line with how vectors are multiplied - of course, since vectors are matrices - but it might look slightly more confusing. We can multiply two matrices if and only if the number of columns of the first matrix is identical to the number of rows of the second. So if we have matrices

$\mathbf{X}_{n\times k}$ and $\mathbf{M}_{p\times q}$, then we can only calculate $\mathbf{XM}$ if $p = k$ and $\mathbf{MX}$ if $n = q$.[4] The product of two matrices is the matrix that consists of every **inner product** of row vectors of the first matrix with column vectors of the second. So if we have $\mathbf{X}_{n\times k}$ and $\mathbf{M}_{k\times q}$ then $\mathbf{XM} = [\mathbf{x}_{i\bullet}\mathbf{m}_{\bullet j}]_{n\times q} = [\sum_{c=1}^{k}(x_{ic}\cdot m_{cj})]_{n\times q}$. $\mathbf{XM}$ is thus an $n \times q$ matrix and its ijth element is $\sum_{c=1}^{k} x_{ic}m_{cj}$, the inner product of the ith row of $\mathbf{X}$ and the jth column of $\mathbf{M}$. This would be a sufficient definition, but it is much easier to see in practice:

$$\begin{aligned}
\mathbf{X} &= \begin{bmatrix} 1 & 1 & 2 \\ 0.5 & 0.5 & 3 \end{bmatrix} \\
\mathbf{M} &= \begin{bmatrix} 3 & 2 \\ 1 & 4 \\ 5 & 7 \end{bmatrix} \\
\mathbf{XM} &= \begin{bmatrix} 1\cdot 3 + 1\cdot 1 + 2\cdot 5 & 1\cdot 2 + 1\cdot 4 + 2\cdot 7 \\ 0.5\cdot 3 + 0.5\cdot 1 + 3\cdot 5 & 0.5\cdot 2 + 0.5\cdot 4 + 3\cdot 7 \end{bmatrix} \\
&= \begin{bmatrix} 14 & 20 \\ 17 & 24 \end{bmatrix} \\
\mathbf{MX} &= \begin{bmatrix} 3\cdot 1 + 2\cdot 0.5 & 3\cdot 1 + 2\cdot 0.5 & 3\cdot 2 + 2\cdot 3 \\ 1\cdot 1 + 4\cdot 0.5 & 1\cdot 1 + 4\cdot 0.5 & 1\cdot 2 + 4\cdot 3 \\ 5\cdot 1 + 7\cdot 0.5 & 5\cdot 1 + 7\cdot 0.5 & 5\cdot 2 + 7\cdot 3 \end{bmatrix} \\
&= \begin{bmatrix} 4 & 4 & 12 \\ 3 & 3 & 14 \\ 8.5 & 8.5 & 31 \end{bmatrix}
\end{aligned}$$

It is advisable to practice with some random small matrices, to get a feel for how this multiplication works. You should do this practice on paper, not in R. We can do the above example also in R:

```
X <- rbind(c(1,1,2), c(.5,.5,3))
M <- cbind(c(3,1,5), c(2,4,7))
X %*% M
M %*% X
```

Note from the above that $\mathbf{XM} \neq \mathbf{MX}$, which is different from algebra with single values and has significant implications for matrix algebra.

Multiplying a matrix with a vector is now a very straightforward extension: by taking the vector as a matrix with just one column or row, the rules are simply the same:

$$\begin{bmatrix} 2 & 4 \\ 1 & 2 \\ 1 & 8 \\ 3 & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 3 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 3\cdot 2 + \frac{1}{2}\cdot 4 \\ 3\cdot 1 + \frac{1}{2}\cdot 2 \\ 3\cdot 1 + \frac{1}{2}\cdot 8 \\ 3\cdot 3 + \frac{1}{2}\cdot \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 7 \\ 9\frac{1}{4} \end{bmatrix}$$

Since the number of columns of the **transpose** of a matrix is the same as the number of rows of that matrix, we can always multiply a matrix with its

[4] It is common in mathematics not to write the multiplication operator explicitly, so while we could write $a \times b$ or $a \cdot b$ to indicate "a times b", we will generally simply write ab, unless this leads to confusion. In matrix algebra there is something called a dot product, which we do not use in this text, therefore, to avoid confusion, the $\cdot$ notation will not be used for matrices.

transpose, as in $\mathbf{X}'\mathbf{X}$ or $\mathbf{X}\mathbf{X}'$. If the matrix is a square matrix, i.e. the number of rows and columns are the same, then we can also calculate $\mathbf{X}\mathbf{X} = \mathbf{X}^2$. Verify that the diagonal of $\mathbf{X}'\mathbf{X}$ contains the **sum of squares** of the column vectors of $\mathbf{X}$.

Note that, as with vectors, the **transpose** of a multiplication is the multiplication of the transposed matrices, in reverse order: $(\mathbf{X}\mathbf{M})' = \mathbf{M}'\mathbf{X}'$. The following rules can be derived from the above:

$$(\mathbf{A}\mathbf{B})\mathbf{C} = \mathbf{A}(\mathbf{B}\mathbf{C})$$
$$\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{A}\mathbf{B} + \mathbf{A}\mathbf{C}$$
$$(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{A}\mathbf{C} + \mathbf{B}\mathbf{C}$$

3.6 Special matrices

There are a number of matrices that are in some way special. This section will just list them and, where useful, mention some typical characteristics.

A matrix which consists of only zeros is a **null matrix**. We usually denote this matrix simply by a 0 - it is generally clear from the context that it concerns a matrix rather than simply the number zero.

A **square matrix** is a matrix with a number of columns identical to the number of rows.

A **diagonal matrix** is a square matrix with values on the diagonal and zeros everywhere else (i.e. $x_{ij} = 0 \quad \forall \quad i \neq j$),[5] for example:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

Note that if $\mathbf{X}$ is diagonal, $\mathbf{X}' = \mathbf{X}$.

An **identity matrix** is a diagonal matrix with only ones, i.e. $x_{ij} = 1 \quad \forall \quad i = j$ and $x_{ij} = 0 \quad \forall \quad i \neq j$. For example:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The identity matrix is usually denoted $\mathbf{I}$, or sometimes includes the size, as in $\mathbf{I}_4$. Note that $\mathbf{I}_n\mathbf{X}_{n\times k} = \mathbf{X}_{n\times k}\mathbf{I}_k = \mathbf{X}_{n\times k}$.

A matrix is **symmetric** if $\mathbf{X}' = \mathbf{X}$. Every diagonal matrix is symmetric.

A matrix is **idempotent** if $\mathbf{X}^2 = \mathbf{X}$. Every identity matrix is idempotent.

A matrix is **orthogonal** if $\mathbf{X}\mathbf{X}' = \mathbf{X}'\mathbf{X} = \mathbf{I}$.[6] If either $\mathbf{X}\mathbf{X}' = \mathbf{I}$ or $\mathbf{X}'\mathbf{X} = \mathbf{I}$ holds for a matrix, but not both, the matrix is said to be semi-orthogonal.

If there is a matrix $\mathbf{A}$ such that $\mathbf{A}^2 = \mathbf{X}$, then $\mathbf{A}$ is said to be the **square root** of matrix $\mathbf{X}$, i.e. $\mathbf{X}^{1/2}$. There can be more than one square root matrix for any $\mathbf{X}$.

[5] $\forall$ means "for all" or "everywhere where".

[6] In other words, as discussed below, if $\mathbf{X}^{-1} = \mathbf{X}'$, $\mathbf{X}$ is orthogonal.

3.7 Linear and quadratic forms

The expression $\mathbf{v}'\mathbf{x}$ is called a **linear form** in $\mathbf{x}$ and the expression $\mathbf{x}'\mathbf{A}\mathbf{x}$ a **quadratic form**. $\mathbf{x}'\mathbf{A}\mathbf{y}$ is called a bilinear form. Note that:

$$\mathbf{v}'\mathbf{x} = \sum_i v_i x_i$$
$$\mathbf{x}'\mathbf{A}\mathbf{x} = \sum_j \sum_i a_{ij} x_i x_j$$

If we assume that $\mathbf{A}$ is a symmetric matrix, we call this matrix **positive definite** iff[7] $\mathbf{x}'\mathbf{A}\mathbf{x} > 0 \quad \forall \quad \mathbf{x} \neq 0$ and positive semidefinite iff $\mathbf{x}'\mathbf{A}\mathbf{x} \geq 0 \quad \forall \quad \mathbf{x} \neq 0$.

If a matrix is the result of the multiplication of a matrix with its transpose, i.e. either $\mathbf{B}'\mathbf{B}$ or $\mathbf{B}\mathbf{B}'$, then this matrix is positive semidefinite.[8] If $\mathbf{B}$ is of full rank (see below), then $\mathbf{B}'\mathbf{B}$ is positive definite.

3.8 Matrix rank

If, for a particular matrix $\mathbf{X}$, $\mathbf{X}\mathbf{v} = 0$ only holds when $\mathbf{v} = 0$, then the columns of $\mathbf{X}$ are said to be **linearly independent**. If there is any $\mathbf{v} \neq 0$ at all where $\mathbf{X}\mathbf{v} = 0$, then the columns of $\mathbf{X}$ are linearly dependent. Similarly, if for a particular matrix $\mathbf{X}$, $\mathbf{X}\mathbf{v} = 0$ only holds when $\mathbf{v} = 0$, then the rows of $\mathbf{X}$ are said to be linearly independent.

The column **rank** is the maximum number of linearly independent columns of a matrix. Since the column rank is always equal to the row rank, we can simply talk of the rank of a matrix, denoted by $r(\mathbf{X})$ (in case of matrix $\mathbf{X}$). The maximum value for the rank is either the number of rows or the number of columns of a matrix, whichever is the lowest. A matrix which has a rank equal to the minimum of the number of rows and number of columns is said to be of **full rank**.

The abstract definition of the rank is somewhat difficult to follow intuitively. Perhaps an example will help. Consider the following three matrices:

$$\mathbf{A} = \begin{bmatrix} 3 & 5 & 1 \\ 2 & 2 & 1 \\ 1 & 4 & 2 \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} 3 & 5 & 9 \\ 2 & 2 & 6 \\ 1 & 4 & 3 \end{bmatrix}$$
$$\mathbf{C} = \begin{bmatrix} 3 & 5 & 11\frac{1}{2} \\ 2 & 2 & 7 \\ 1 & 4 & 5 \end{bmatrix}$$

[7] The term "iff", often used in mathematics and philosophy, stands for "if and only if".

[8] A straightforward proof for $\mathbf{B}'\mathbf{B}$ can be found in Davidson and MacKinnon (1993, 787) and uses much of the matrix algebra we studied above: $\mathbf{x}'\mathbf{B}'\mathbf{B}\mathbf{x} = (\mathbf{B}\mathbf{x})'(\mathbf{B}\mathbf{x}) = \|\mathbf{B}\mathbf{x}\|^2 \geq 0$. Similarly, $\mathbf{x}'\mathbf{B}\mathbf{B}'\mathbf{x} = (\mathbf{B}'\mathbf{x})'(\mathbf{B}'\mathbf{x}) = \|\mathbf{B}'\mathbf{x}\|^2 \geq 0$. So since both are sums of squares, they are by definition positive. Because we did not make any assumption about the rank of $\mathbf{B}$, it might be possible that there is a $\mathbf{x} \neq 0$ such that $\mathbf{B}\mathbf{x} = 0$, so that we cannot be sure that these matrices are positive definite.

For matrix $\mathbf{A}$, the three columns are independent and $r(\mathbf{A}) = 3$. There is no $\mathbf{v} \neq 0$ such that $\mathbf{Av} = 0$ (of course, if $\mathbf{v} = 0$, $\mathbf{Av} = \mathbf{A}0 = 0$).[9]

For matrix $\mathbf{B}$, however, the first and the last column vectors are linear combinations of each other. That is to say, $\mathbf{b}_{\bullet 1} = \frac{1}{3}\mathbf{b}_{\bullet 3}$. We thus cannot say that the columns of $\mathbf{B}$ are linearly independent. In fact, at most two columns ($\mathbf{b}_{\bullet 1}$ and $\mathbf{b}_{\bullet 2}$ or $\mathbf{b}_{\bullet 2}$ and $\mathbf{b}_{\bullet 3}$) are independent, so $r(\mathbf{B}) = 2$. We could construct a matrix $\mathbf{v}$ such that $\mathbf{Bv} = 0$, namely $\mathbf{v} = \begin{bmatrix} 1 & 0 & -\frac{1}{3} \end{bmatrix}'$ (there is an infinite number of such vectors, of the form $\mathbf{v} = \begin{bmatrix} \alpha & 0 & -\frac{1}{3}\alpha \end{bmatrix}'$).

For matrix $\mathbf{C}$ we have a similar story, and $r(\mathbf{C}) = 2$ as well. In this case, one cannot express one of the column vectors as a linear combination of another column vector, but one can express any of the column vectors as a linear combination of the two other column vectors. For example, $\mathbf{c}_{\bullet 3} = 3\mathbf{c}_{\bullet 1} + \frac{1}{2}\mathbf{c}_{\bullet 2}$ and thus when $\mathbf{v} = \begin{bmatrix} 3 & \frac{1}{2} & -1 \end{bmatrix}'$, $\mathbf{Cv} = 0$.

If $r(\mathbf{X}) = 0$ then $\mathbf{X} = 0$ and vice versa, if $\mathbf{X} = 0$ then $r(\mathbf{X}) = 0$. Furthermore, the following holds: $r(\mathbf{A}) = r(\mathbf{A}') = r(\mathbf{A}'\mathbf{A}) = r(\mathbf{A}\mathbf{A}')$. And simply for completeness sake:

$$\begin{aligned} r(\mathbf{AB}) &= min(r(\mathbf{A}), r(\mathbf{B})) \\ r(\mathbf{AB}) &= r(\mathbf{A}) \qquad \text{if } \mathbf{B} \quad \text{is square and of full rank} \\ r(\mathbf{A} + \mathbf{B}) &\leq r(\mathbf{A}) + r(\mathbf{B}) \end{aligned}$$

A square matrix which is of full rank is called **non-singular**, while a square matrix which is of less than full rank is called **singular**.

In R, the rank can be found using a function available in the Matrix package:

```
library(Matrix)
rankMatrix(X)
```

Ranks are important in regression analysis: when the data matrix is not of full rank, there is a problem of perfect **multicollinearity**.

3.9 Matrix inverse

If $\mathbf{A}$ is a square, non-singular matrix there exists a matrix $\mathbf{B}$ such that $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$. We denote this matrix $\mathbf{B}$ as $\mathbf{A}^{-1}$ and it is called the **inverse** of $\mathbf{A}$. For a matrix is non square or singular, the inverse does not exist; otherwise there is only one possible solution of $\mathbf{A}^{-1}$. So by definition, $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$, if $\mathbf{A}$ is non-singular.

The calculation of the inverse is somewhat cumbersome and will not be detailed here - it is much easier to use software for this purpose. For example in R:

```
X <- cbind(c(1,4,2),c(4,2,2),c(1,1,1))
solve(X)
```

The concept of the inverse is rather important, however, and as you will see below (§5), a key part of linear regression analysis. It is the closest we get in matrix algebra to division.

[9] Note that a discussion of how one can establish whether such $\mathbf{v}$ exists, or what the rank is if it is not full rank, is beyond the scope of this text.

Note that we stated in §3.8 that $r(\mathbf{X}'\mathbf{X}) = r(\mathbf{X})$, so if $\mathbf{X}$ is of full rank, $\mathbf{X}'\mathbf{X}$ is also of full rank. $\mathbf{X}'\mathbf{X}$ is by definition a square matrix, so if $\mathbf{X}$ is of full rank, we can call $\mathbf{X}'\mathbf{X}$ a non-singular matrix, and the inverse will exist. If $\mathbf{X}$ is not of full rank, i.e. there are **linear dependencies** between columns, then $\mathbf{X}'\mathbf{X}$ will be singular and the inverse will not exist. Since $(\mathbf{X}'\mathbf{X})^{-1}$ is part of the **ordinary least squares** estimation procedure, this linear dependency would render estimation impossible. This is what is called perfect **multicollinearity**.

Some handy rules are:

$$(\mathbf{A}^{-1})' = (\mathbf{A}')^{-1}$$
$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

3.10 Trace of a matrix

The **trace** of a square matrix is simply the sum of the diagonal elements. So if $\mathbf{X} = [x_{ij}]_{n\times k}$, then the trace $tr(\mathbf{X}) = \sum_{i=1}^{n} x_{ii}$. For example:

$$\mathbf{X} = \begin{bmatrix} 7 & 1 & 2 \\ 9 & 1 & 3 \\ 12 & 2 & 5 \end{bmatrix}$$
$$tr(\mathbf{X}) = 7 + 1 + 5 = 13$$

The following rules hold for traces:

$$\begin{aligned} tr(\mathbf{A} + \mathbf{B}) &= tr(\mathbf{A}) + tr(\mathbf{B}) \\ tr(c\mathbf{A}) &= c \cdot tr(\mathbf{A}) \\ tr(\mathbf{A}') &= tr(\mathbf{A}) \\ tr(\mathbf{AB}) &= tr(\mathbf{BA}) \\ tr(\mathbf{ABC}) &= tr(\mathbf{CAB}) = tr(\mathbf{BCA}) \\ tr(\mathbf{A}'\mathbf{A}) &\geq 0 \\ \mathbf{v}'\mathbf{w} &= tr(\mathbf{v}\mathbf{w}') \quad \text{if } \mathbf{v}\mathbf{w}' \text{ is a square matrix} \end{aligned}$$

whereby $tr(\mathbf{A}'\mathbf{A})$ is only 0 when $\mathbf{A}$ is a null matrix. An example of the use of this kind of permutations is (Davidson and MacKinnon, 1993, 774):

$$tr(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') = tr(\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}) = tr(\mathbf{I}) = k$$

for a data matrix $\mathbf{X}_{n\times k}$.

4 Derivatives

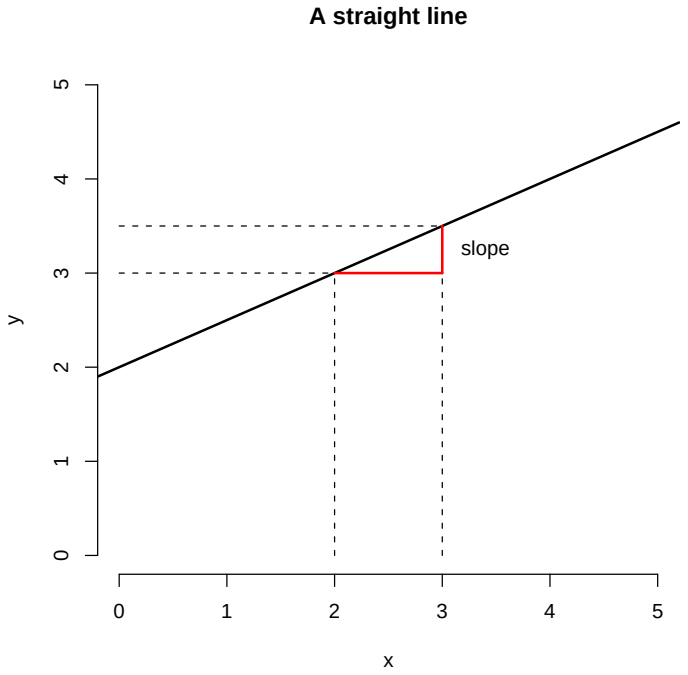
4.1 Concept

Derivatives is an extensive topic in calculus. Here we will only devote very little time to the topic, but the most important thing it to have an idea of the concept of derivates and of finding maximum and minimum points on a curve using derivatives. Derivatives are important throughout the course: the β-coefficient in a regression analysis represents the slope of the line, i.e., the derivative of a straight line; the estimation of the β-coefficients in an ordinary least squares

regression takes place through taking a derivative; and the Maximum Likelihood (ML) estimation in regression is based on taking the derivative of the log-likelihood function.

4.1.1 Straight lines

For a straight line, the **slope** of the line is the increase in y for one point increase in x.



The equation of the above line is $y = 2 + \frac{1}{2}x$. You can see that if x increases by one, y will increase by $\frac{1}{2}x = \frac{1}{2} \cdot 1 = \frac{1}{2}$. Another way of formulating this is to say that the **first derivative** of y is $\frac{1}{2}$, or:[10]

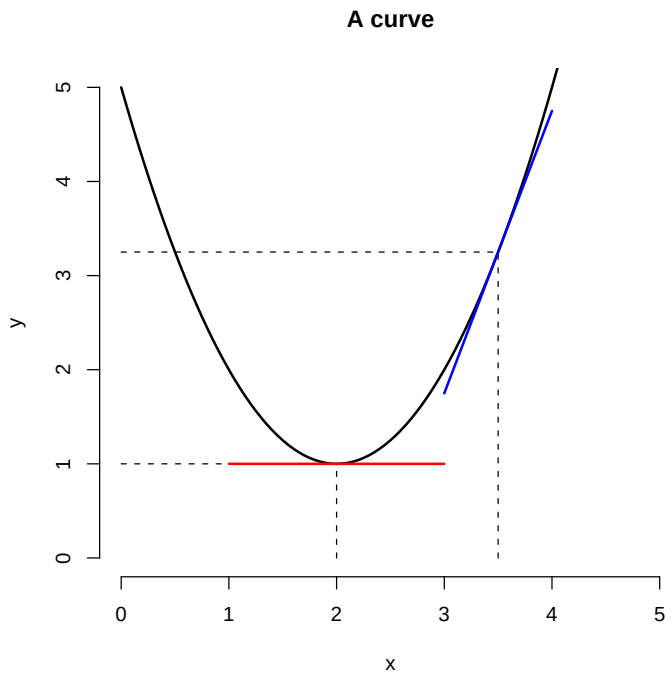
$$\frac{dy}{dx} = \frac{1}{2}$$

The derivative thus tells you something about how quickly y changes, given changes in x.

4.1.2 Curves

The story does not change much for quadratic forms - also here, the derivative tells you how much y changes, given changes in x.

[10] Notations for the derivative vary. $\frac{dy}{dx}$ is a good notation, because it makes explicit what function the derivative is taken from, with respect to what other variable. Another common formulation is that for function $f(x)$, the derivative is $f'(x)$ and the second derivative $f''(x)$. We will consistently use the former notation.



The equation for this curve is $y = x^2 - 4x + 5$. As you can see, the amount by which y changes given a change in x depends on where we are on the curve. Whereas with a straight line, the derivative is just a constant, here the derivative is itself a function of x. Explaining all rules of derivatives is beyond the scope of this text, but a straightforward pair of rules that helps with most basic functions is:

$$\frac{d(ax^b)}{dx} = bax^{b-1}$$
$$\frac{d(a+b)}{dx} = \frac{da}{dx} + \frac{db}{dx}$$

So we can apply both to get the derivative of y:

$$\frac{d(x^2)}{dx} = 2x$$
$$\frac{d(-4x)}{dx} = -4$$
$$\frac{dy}{dx} = \frac{d(x^2 - 4x + 5)}{dx} = 2x - 4$$

We can see that the derivative is a function of x, so we can find the derivate at a specific point on the curve by putting in the value of x. For example for $x = 2$:

$$\frac{dy}{dx} = 2x - 4 = 2 \cdot 2 - 4 = 0$$

And for $x = 3\frac{1}{2}$:

$$\frac{dy}{dx} = 2x - 4 = 2 \cdot 3\frac{1}{2} - 4 = 3$$

Note that there is something special about $\frac{dy}{dx} = 0$: at this point on the curve, the curve so to speak changes direction, from increasing to decreasing or vice versa. Always when a curve changes "direction" like this, the derivative is zero - but not vice versa, not everywhere where the derivative is zero does the curve change direction. Another and better term for this kind of point is that the curve reached a (local) minimum or maximum. So if we want to find where the curve reaches a (local) minimum or maximum, we can equating the derivative to zero. We gave the answer already, but the following steps could be taken to find the minimum of this curve:

$$\begin{aligned} \frac{dy}{dx} = 2x - 4 &= 0 \\ 2x &= 4 \\ x &= \frac{4}{2} = 2 \end{aligned}$$

So there is a minimum or maximum at $x = 2$. This concept of finding the minimum or maximum is crucial for regression analysis. With ordinary least squares, we estimate the β-coefficients by finding the minimum of the sum of squared errors. With maximum likelihood, we estimate the β-coefficients by finding the maximum point of the (log)likelihood function.

4.1.3 Multiple dimensions

A function can have more than two dimensions like in the two plots above. For example $y = 2x^2 + 4z^3 + 3z + 5$ would be a function in three-dimensional space. We can take the derivative in two ways:

$$\begin{aligned} \frac{dy}{dx} &= 4x \\ \frac{dy}{dz} &= 12z^2 + 3 \end{aligned}$$

We call these functions **partial derivative**. In regression analysis we usually have to deal with this situation, since we are usually interested in more than one independent variable, so we are dealing with derivatives in higher dimensional spaces.

It is also possible for variables to "interact", for example $y = 2x^3 - 2xz + 4z$:

$$\begin{aligned} \frac{dy}{dx} &= 6x^2 - 2z \\ \frac{dy}{dz} &= -2x + 4 \end{aligned}$$

The derivative of y relative to z thus depends on the value of x.

4.2 Matrix algebra and derivatives

The derivative of a matrix with respect to a scalar is simply the matrix of each of the derivatives of each of the elements of the first matrix, with respect to the scalar. So if we have matrix $\mathbf{X} = [x_{ij}]_{n \times k}$ then $\frac{d\mathbf{X}}{d\alpha} = [\frac{dx_{ij}}{d\alpha}]_{n \times k}$. For example,

if:

$$\mathbf{M} = \begin{bmatrix} x^2 & 2x & 3 \\ 2x+4 & 3x^3 & 2x \\ 3 & 2 & 4x \end{bmatrix}$$

then the derivative of $\mathbf{M}$ to x is:

$$\frac{d\mathbf{M}}{dx} = \begin{bmatrix} 2x & 2 & 0 \\ 2 & 9x^2 & 2 \\ 0 & 0 & 4 \end{bmatrix}$$

The above takes derivatives of matrix $\mathbf{M}$ with respect to a scalar, x. What happens when we take the derivative with respect to a vector? When taking the derivative of a function with respect to a vector, we get a vector of derivatives for each of the elements of the original vector. This means that in the case of a matrix, we get a three-dimensional matrix, with layers for each derivative. This is somewhat difficult to imagine, so it is easier to think of this in terms of **partial derivatives**, looking at one element of the vector at a time (Gentle, 2007, 152). So the results of a derivative are:

Take derivative of ...	*With respect to ...*	*Result*
function	scalar	function
$n \times 1$ vector	scalar	$n \times 1$ vector
$n \times m$ matrix	scalar	$n \times m$ matrix
function	$n \times 1$ vector	$n \times 1$ vector
$n \times 1$ vector	$1 \times m$ vector	$n \times m$ matrix
$n \times m$ matrix	$1 \times p$ vector	$n \times m \times p$ matrix

In terms of linear and quadratic forms, we can use the following rules (Harville, 1997, 295-296):

$$\frac{d(\mathbf{v}'\mathbf{x})}{d\mathbf{x}} = \mathbf{v}$$
$$\frac{d(\mathbf{x}'\mathbf{A}\mathbf{x})}{d\mathbf{x}} = (\mathbf{A} + \mathbf{A}')\mathbf{x}$$
$$\frac{d\mathbf{B}\mathbf{x}}{d\mathbf{x}} = \mathbf{B}$$

A matrix of partial derivatives is called a **Jacobian matrix**, so in the last equation, $\mathbf{B}$ is the Jacobian matrix of $\mathbf{Bx}$.

The derivative tells you something about the slope of the curve[11] at a particular point. You can also take the derivative of the derivative, which will tell you how quickly this slope changes at a particular point. If you want a comparison to physics, you can compare the function with a description of your location, the derivative with a description of your speed, and the second derivative with a description of your acceleration. We can write the second derivative, for example, as $\frac{d^2y}{dx_1 dx_2}$, which can be read as the function that you acquire by first taking the derivative of y with respect to x_1, and the derivative of this with respect to x_2. For quadratic forms we have (Harville, 1997, 296):

$$\frac{d^2(\mathbf{x}'\mathbf{A}\mathbf{x})}{d\mathbf{x}d\mathbf{x}'} = \mathbf{A} + \mathbf{A}'$$

[11] Since we are usually talking about vectors of dimension higher than 2, the more appropriate term is surface.

You can imagine that we can now generate a matrix $\mathbf{H}$, such that $\mathbf{H} = [\frac{d^2y}{dx_i dx_j}]_{n \times k}$, which is called the **Hessian matrix** (Greene, 2003, 838), so this looks like:

$$\mathbf{H} = \begin{bmatrix} \frac{d^2y}{dx_1 dx_1} & \frac{d^2y}{dx_1 dx_2} & \cdots & \frac{d^2y}{dx_1 dx_k} \\ \frac{d^2y}{dx_2 dx_1} & \frac{d^2y}{dx_2 dx_2} & \cdots & \frac{d^2y}{dx_2 dx_k} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{d^2y}{dx_k dx_1} & \frac{d^2y}{dx_k dx_2} & \cdots & \frac{d^2y}{dx_k dx_k} \end{bmatrix} = \frac{d^2\mathbf{y}}{d\mathbf{x}d\mathbf{x}'}$$

5 Deriving the OLS estimator

Let us look at the derivation of the estimator for **ordinary least squares** (OLS). We have the following equation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

whereby $\mathbf{y}$ is a vector of dimension n; $\mathbf{X}$ is a data matrix with dimensions $n \times k$ (we can assume that the first column contains ones to have a constant in the model, but this is irrelevant to the remainder of this section); $\boldsymbol{\beta}$ is a vector of dimension k; and $\boldsymbol{\varepsilon}$ is a vector of dimension n. $\mathbf{y}$ represents the dependent variable in the model; $\mathbf{X}$ the independent variables; $\boldsymbol{\beta}$ the regression coefficients in the model that generated the data in nature - so we are assuming these are the "real" coefficients; $\boldsymbol{\varepsilon}$ contains the residual variation in $\mathbf{y}$. We will use the $\hat{\boldsymbol{\beta}}$ notation to refer to the estimated values of $\boldsymbol{\beta}$.

Note that the β-coefficients thus represent the slope of $\mathbf{y}$, i.e. how much $\mathbf{y}$ changes on a unit change in $\mathbf{x}_i$:

$$\frac{d\mathbf{y}}{dx_i} = \boldsymbol{\beta}_i$$

OLS is based on the idea that we can estimate $\hat{\boldsymbol{\beta}}$ by minimizing the squared sum of the errors, i.e. by minimizing $\mathbf{e}'\mathbf{e}$. So we want to minimize:

$$\begin{aligned} \mathbf{e}'\mathbf{e} &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= (\mathbf{y}' - \boldsymbol{\beta}'\mathbf{X}')(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

Since we want to find the minimum of this equation (we want to minimize the estimation errors) and since we want to find the optimal values of $\boldsymbol{\beta}$ to get this minimum, we are interested in (note that since $\mathbf{y}'\mathbf{X}\boldsymbol{\beta}$ is a scalar, it is identical to its transpose):

$$\begin{aligned} \frac{d(\mathbf{e}'\mathbf{e})}{d\boldsymbol{\beta}} &= \frac{d(\mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta})}{d\boldsymbol{\beta}} \\ &= -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

So we can find the estimate of $\boldsymbol{\beta}$ (i.e. $\hat{\boldsymbol{\beta}}$) by setting this equation to zero:

$$\begin{aligned}
-2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} &= 0 \\
2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} &= 2\mathbf{X}'\mathbf{y} \\
\frac{1}{2}(\mathbf{X}'\mathbf{X})^{-1} \cdot 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} &= \frac{1}{2}(\mathbf{X}'\mathbf{X})^{-1} \cdot 2\mathbf{X}'\mathbf{y} \\
(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\
\hat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}
\end{aligned}$$

With this, you can implement your own OLS estimator quite straightforwardly in R:[12]

```
bhat <- solve(t(X) %*% X) %*% t(X) %*% y
```

Note that there is a difference between residuals ($\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$), which are assumed to exist in nature, and errors ($\mathbf{e} \equiv \hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$), which are part of the model estimation.

In addition to the estimation of the regression coefficients, we need to know the standard errors, or more specifically, the variance-covariance matrix of the regression coefficients. The variance-covariance matrix looks like this (Gujarati, 2003, 935):

$$V(\hat{\boldsymbol{\beta}}) = \begin{bmatrix} var(\hat{\boldsymbol{\beta}}_1) & cov(\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2) & ... & cov(\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_k) \\ cov(\hat{\boldsymbol{\beta}}_2, \hat{\boldsymbol{\beta}}_1) & var(\hat{\boldsymbol{\beta}}_2) & ... & cov(\hat{\boldsymbol{\beta}}_2, \hat{\boldsymbol{\beta}}_k) \\ ... & ... & ... & ... \\ cov(\hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\beta}}_1) & cov(\hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\beta}}_2) & ... & var(\hat{\boldsymbol{\beta}}_k) \end{bmatrix}$$

This matrix is very important to keep in mind not only for understanding the OLS estimator, but also for the class on bootstrapping and simulation.

The variance-covariance matrix of $\hat{\boldsymbol{\beta}}$ is defined by:

$$V(\hat{\boldsymbol{\beta}}) = E[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})'],$$

where $E[x]$ is the expectation of x - we will discuss this below. To get the value of $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$, we need to move around a bit with the equations from above (Gujarati, 2003, 956-957):

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\
&= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}) \\
&= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon} \\
&= \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon} \\
\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}
\end{aligned}$$

[12] Note that the OLS estimator as implemented in R takes a few steps in preparation, which you would now have to do manually. The dataset usually consists of many variables, including those you do not want to use in your regression, and is usually stored as a dataframe. The matrix multiplication here assumes a matrix and $\mathbf{X}$ should contain only the variables you are interested in, including the constant. Furthermore, a dataframe often contains missing values, but the matrix multiplication cannot deal with those - so you need to remove the missings first. This is normally done using listwise deletion - i.e. if there is any missing values for a particular unit (respondent, country, etc.) in either y or $\mathbf{X}$, the row is completely removed from both.

Now we can use this to work out the variance-covariance matrix:

$$\begin{aligned} V(\hat{\boldsymbol{\beta}}) &= E[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})'] \\ &= E[((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon})((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon})'] \\ &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}] \end{aligned}$$

Because one of the assumptions of OLS is that $\mathbf{X}$ is fixed, while only the error terms are considered random, we get:

$$V(\hat{\boldsymbol{\beta}}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}']\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

And another assumption we make is that the variance of the error term is constant (**homoscedasticity**), indicated by σ^2, with no covariances between errors (no **autocorrelation**), we can say that $E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'] = \sigma^2\mathbf{I}$, so that:

$$\begin{aligned} V(\hat{\boldsymbol{\beta}}) &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\sigma^2\mathbf{I}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\ &= \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \end{aligned}$$

So the standard errors on $\hat{\boldsymbol{\beta}}$ are the square root of the diagonal values of this variance-covariance matrix.[13]

In R we can estimate our OLS coefficients and standard errors using matrix algebra with the following code:[14]

```
n <- dim(X)[1]
k <- dim(X)[2]
bhat <- solve(t(X) %*% X) %*% t(X) %*% y
e <- y - X %*% bhat
s2 <- t(e) %*% e / (n - k)
V <- s2 %x% solve(t(X) %*% X)
se <- sqrt(diag(V))
```

[13]Note that there is an alternative route to arrive at $V(\hat{\boldsymbol{\beta}})$, which does not require the assumption of fixed $\mathbf{X}$, using the law of iterated expectations. As above, $V(\hat{\boldsymbol{\beta}}) = E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}]$, from where we can proceed:

$$\begin{aligned} V(\hat{\boldsymbol{\beta}}) &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}] \\ &= E_{\mathbf{X}}[E_{\boldsymbol{\varepsilon}|\mathbf{X}}[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}]] \\ &= E_{\mathbf{X}}[E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}|\mathbf{X}]] \\ &= E_{\mathbf{X}}[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'|\mathbf{X}]\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}] \end{aligned}$$

Given $\mathbf{X}$, the expectation $E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'|\mathbf{X}] = \sigma^2\mathbf{I}$, therefore:

$$\begin{aligned} V(\hat{\boldsymbol{\beta}}) &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\sigma^2\mathbf{I}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}] \\ &= E_{\mathbf{X}}[\sigma^2(\mathbf{X}'\mathbf{X})^{-1}] \\ &= \sigma^2 E[(\mathbf{X}'\mathbf{X})^{-1}] \end{aligned}$$

which is then approximated by $V(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$.

[14]In R, %*% is the symbol to multiply two matrices, while %x% is the symbol to get the kronecker product of two matrices. A kronecker product $\mathbf{A} \otimes \mathbf{B}$ means that every element in $\mathbf{A}$ is multiplied with the entire matrix $\mathbf{B}$, such that the resulting matrix is much larger than the two separate. Since in the R code s2 is a scalar, but seen by R as a matrix of size 1×1, we use the kronecker product to avoid errors. If you see a scalar as a 1×1 matrix, then the kronecker product is the same thing as a multiplication with a scalar.

```
t <- bhat / se

print(cbind(bhat, se, t))
```

To generate a fake example in R, you can do something like the following (where rnorm is a function to generate a collection of random values from the normal distribution, with parameters number of values, mean of distribution, standard deviation of distribution):

```
x1 <- rnorm(100,3,2)
x2 <- rnorm(100,2,4)
x3 <- rnorm(100,0,2)
y <- 3 + 2*x1 + x2 + 3*x3 + rnorm(100,0,2)
X <- cbind(1, x1, x2, x3)
```

So the "real" equation here is $\mathbf{y} = 3 + 2\mathbf{x}_1 + \mathbf{x}_2 + 3\mathbf{x}_3 + \boldsymbol{\varepsilon}$ and you can use the above code to try to find estimates for $\boldsymbol{\beta}$. You should, of course, get close to $\hat{\boldsymbol{\beta}} = (3, 2, 1, 3)'$. To do it using built-in functions, you just use:

```
summary(lm(y ~ x1 + x2 + x3))
```

References

Davidson, Russell and James G. MacKinnon. 1993. *Estimation and inference in econometrics*. Oxford: Oxford University Press.

Gentle, James E. 2007. *Matrix algebra: theory, computations, and applications in statistics*. New York: Springer.

Greene, William H. 2003. *Econometric Analysis*. 5th ed. Upper Saddle River: Prentice Hall.

Gujarati, Damodar N. 2003. *Basic econometrics*. 4th ed. Boston: McGraw-Hill.

Harville, David A. 1997. *Matrix algebra from a statistician's perspective*. New York: Springer-Verlag.

Magnus, Jan R. and Heinz Neudecker. 1999. *Matrix differential calculus with applications in statistics and econometrics*. revised ed. Chichester: John Wiley & Sons.