EC221: A (Hopefully!) Intuitive Guide to Linear Algebra

Tom Glinnan[*]

January 2022

Contents:

Introduction: What Math do you actually need for the LT of EC221?

If you look at the slides on Moodle, I wouldn't blame you for thinking they're quite math-heavy. When I took EC221 as an undergrad I remember thinking that my half-remembered MA100 wasn't really enough by itself to give me any insight into what was going on at the start. One general tip that I would keep in mind for *any* quantitative module is:

> If you don't understand a proof, it's almost certainly
> because you don't understand the underlying math

So getting up to speed on the math required for the LT can make things much more manageable, and allow you to see the intuition behind what's going on in the course, not just drowning in symbols. Going back to the math itself is a great idea if you ever get completely confused

The main two areas of math used in EC221 are **linear algebra** (matrices etc.) and **statistics**, with linear algebra being the one people often forget. A fair enough question to ask is **why** it's used so much. There are really two reasons:

[*] It goes without saying: please feel free to come to office hours or send me an email if you want to discuss any of this! Email me too if there are any corrections

- By expressing our methods in a matrix form, we get much cleaner proofs which allow you to see what's going on underneath without messy algebra. For example, deriving OLS using sums is painful when there are many regressors, but if we use matrices it's literally the same proof as in the bivariate regression ($y_i = \alpha + \beta x_i + \varepsilon_i$) case
- Because we can interpret matrices as linear transformations, we get new interpretations of what our methods are doing, which we can't see if we only use sums

Please **do not** think that you need to have a perfect understanding of linear algebra before you start EC221 - it's something that we'll go over and apply during the term. For many people, the concepts don't 'sink in' until they've seen them applied somewhere. However, it is worth having a look at some basics if you have time, and it's worth looking back on them for clarification during the term. There are lots of resources to help get you up to speed:

- Will Matcham (the course director) runs help lectures which supplement the main material. Lectures 1 and 2 are a review of all of the Linear Algebra you'll have seen in MA100 (plus a little new stuff). I'd really recommend you go or watch the recording if you have the time
- On Moodle there are written up notes on linear algebra, probability and statistics, which cover essentially the same material as Will does
- The notes I'm writing here, which complement the others by covering roughly the same material but focusing less on the formalities and more on the intuition and how we apply it

To be able to do correct proofs in matrix form (point 1 of why we use matrices), you only really need a formal understanding of linear algebra. However, understanding what's really going on under the surface allows you to appreciate point 2, which is why I'm writing this. Linear Algebra is used all over math, so it's far from a useless thing to study. One final note: while I've written notes tailored to metrics here, the best way to learn Linear Algebra properly is without a doubt by watching the following playlist from the great YouTube channel 3Blue1Brown[1]:

`https://www.youtube.com/playlist?app=desktop&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab`

The focus there is on visualising what all of the main concepts in Linear Algebra (rank, determinant, matrix operations, eigenvalues) really mean. It's a great resource especially if you have a specific topic you're confused about

The Basics - what is a matrix and why do we care?

The key to understanding matrices is to realise that there's always two ways we can think of them. A matrix is both:

- A table with objects in it - usually numbers - which has columns and rows

[1] The playlist on multivariable calculus is similarly fantastic if you ever need to re-look at that. I'd really recommend the whole channel if you like math - it has 4m subscribers for a reason

- A linear transformation - if I have a matrix A and a vector x then Ax is another vector. In other words, multiplying by A represents a function which takes in one vector and gives me another

In any given situation, we sometimes have to think of them in one way, and sometimes in the other. Once we know what things we can do with matrices (transpose, take inverses, etc), we usually forget about interpretation and just do rearranging and algebra - but interpreting any assumption or equation that involves matrices might require us to view matrices as a table or as a linear transformation. One final tip: many matrix concepts are generalisations of things that hold for ordinary numbers, so it can be good to build intuition by thinking what would happen if the matrix were a 1×1 matrix just containing a constant a

Different ways to view a DGP

The big reason why matrices matter in metrics is because of the first interpretation: our data naturally comes in tables with rows and columns. If we have data that doesn't come as numbers, we can usually replace them with numbers by creating dummmy variables[2]. To illustrate why this matters to us, our usual assumptions on the data generating process (DGP) often look something like this[3]:

$$y_i = a_1 + a_2 x_{2i} + \varepsilon_i \qquad \mathbb{E}(\varepsilon_i | x_{2i}) = 0$$

In words: we can think of each data point (x_{2i}, y_i) as being generated by a process where God or nature takes a value of x_{2i}, draws a random value of ε_i from some distribution satisfying $\mathbb{E}(\varepsilon_i | x_{2i}) = 0$ and puts them together in this way to get y_i for some constants a_1, a_2. This is the **scalar** way of viewing the DGP. It's quite easy to see that if we define x_i and β as column vectors:

$$x_i = \begin{bmatrix} 1 \\ x_{2i} \end{bmatrix} \qquad \text{and} \qquad \beta = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

then $a_1 + a_2 x_{2i} = \beta' x_i = x_i' \beta$, where $'$ represents the transpose. Therefore we can equivalently write the DGP as[4]:

$$y_i = x_i' \beta + \varepsilon_i \qquad \mathbb{E}(\varepsilon_i | x_i) = 0$$

This is the **vector** way of viewing the DGP, which is useful for analysing (for example) multivariable regression - no matter how many variables we have, the equation looks like this (just with different sized vectors for β and x_i). Both the scalar and vector ways of viewing the DGP focus on one data

[2] For example, 'yes' and 'no' on a survey can be represented by 1 and 0

[3] See the other GitHub document which introduces the LT to see what I mean by this

[4] Since $\mathbb{E}(\varepsilon_i | x_i) = 0$ iff $\mathbb{E}(\varepsilon_i | x_{2i}) = 0$, since the only difference between x_i and x_{2i} is the constant 1

point at a time, but one thing we can also do is stack these equations together. In other words:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \end{bmatrix} \beta + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$= \begin{bmatrix} 1 & x_{2,1} \\ 1 & x_{2,2} \\ \vdots & \vdots \\ 1 & x_{2,n} \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Which we refer to as[5]:

$$y = X\beta + \varepsilon \qquad \mathbb{E}(\varepsilon|X) = 0$$

This is the **matrix** way of viewing the DGP. We can see from the above that all three are equivalent - they're just different notations for the same thing. In the LT of the course we often focus on the matrix interpretation (as it is the cleanest to use, once we understand linear algebra), but sometimes we go for the vector interpretation too[6]. Just like the different ways to view matrices, we want to be able to understand the equivalence between these forms so that we can convert between them. All of the concepts and estimators you will learn about in the LT *can* be written using sums - it's just harder to analyse and more messy to think about (at least to someone who understands linear algebra!)

The Basic Concepts

3Blue1Brown's videos and the lectures by Will are good for explaining many of these[7]. I'll do a quick run through of the main concepts[8]:

- **Matrix Transpose A':** this just refers to taking a matrix A and flipping it on the diagonal (the one which runs from the top left to the bottom right - when we refer to 'the diagonal' it's always this one). To put it another way: this is interchanging the rows and columns. We use this because sometimes if we have non-square matrices, we can't multiply them together unless we take transposes[9]

- **A linear transformation:** as said above, we can think of a matrix as this. It's a special type of function that takes in a vector, and outputs another vector. To see what it does: see

[5] Since $\mathbb{E}(\varepsilon_i|x_i) = 0$ iff $\mathbb{E}(\varepsilon|x) = 0$ under the assumption that data is sampled *iid* from the population / DGP

[6] In the section of asymptotics and the Law of Large numbers it turns out to be useful

[7] With Will's being more formal and more tailored to EC221, but 3Blue1Brown's being much more visual and focussed on intuition

[8] And I'm happy to discuss more with you if you're confused

[9] For example, if X has n rows and k columns, it represents a transformation from $\mathbb{R}^k \to \mathbb{R}^n$; if we have a vector $a \in \mathbb{R}^k$ then $Xa \in \mathbb{R}^n$. We therefore can't define something like XX, as the second time we go to use the X it's expecting to receive a vector in $\mathbb{R}^k$, and instead gets one in $\mathbb{R}^n$. However X' has k rows and n columns, so represents a function from $\mathbb{R}^n \to \mathbb{R}^k$. Therefore, $X'X$ is well-defined: it's a function from $\mathbb{R}^k \to \mathbb{R}^k$

the figure of $\mathbb{R}^2$ with grid lines drawn on it. If we apply a matrix A to every point, the grid lines will have been stretched or squished, and maybe rotated, but they'll still be straight and parallel to each other. They won't be curved, or squashed by different amounts in different places

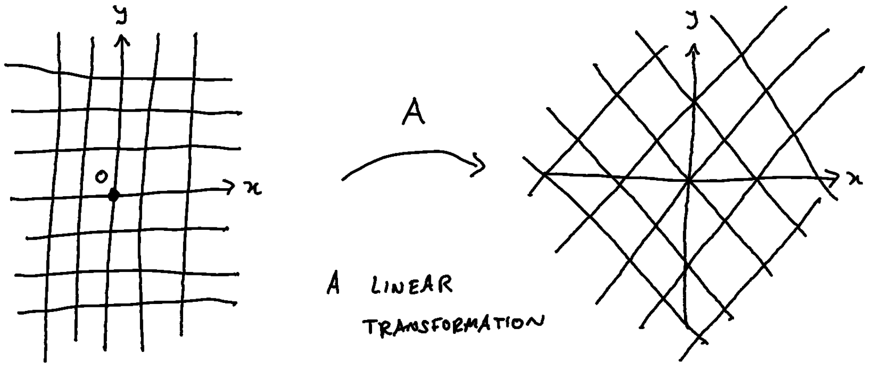

Figure 1: Grid Lines under a Linear Transformation

- **Matrix Inverse A^{-1}:** simply put, this is the inverse of the function that A is. In other words, if A takes in x and gives you y, then A^{-1} takes in y and gives you x
- **Multiplying matrices:** this represents the composition of the two functions: doing one and then the other. In other words, ABx is the function which first does B, then does A[10]
- **The unit square / cube:** because of the property about grid lines, we can analyse what a matrix does by looking at what it does to one square on the grid - namely the square that goes from 0 to 1 on both the x and y axes. This is referred to as the unit square[11]. In higher dimensions, we can refer to this as the unit cube, or if we're being extra technical we can call it the unit hypercube
- **Determinant:** the absolute value of the determinant of a matrix is the area of the unit square after the matrix A has been applied to every point on $\mathbb{R}^2$. The sign of the determinant represents whether the unit square has been 'flipped' over an axis[12]. For example, $det\,A = 5$ means that the unit square has not been flipped and its area has been stretched by 5; $det\,A = -1/3$ means it has been flipped, and then its squished into a space 1/3 of the area of the original square. We know that A stretches and squishes - $det\,A$ effectively measures how much it does this. A special case is when $det\,A = 0$. The unit square now has no area - in other words, the

[10] For the more math-y types, this is why the associative property $(AB)C = A(BC)$ holds trivially for matrices. They're both ways of describing the function which does C, then B, then A

[11] 'Unit' means 1 - it has area 1 and is like the typical example of a square on the grid

[12] More technically, $det\,A > 0$ means it has been flipped over an even number of axes (including 0), and $det\,A < 0$ means it has been flipped over an odd number of them

lines have all been squished into one. As a consequence, we can't undo the transformation, as if we have access to some vector on this line we don't know where it originally came from. This is why we have the result that a matrix can be inverted - A^{-1} exists - if and only if $det\, A \neq 0$

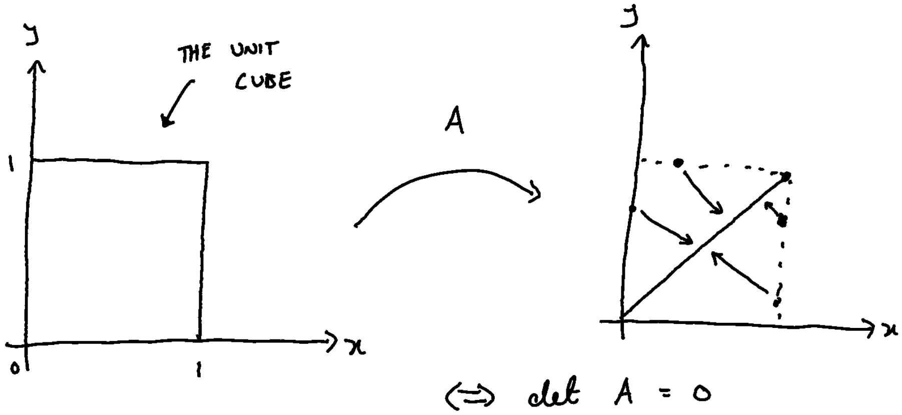

Figure 2: A transformation with $det\, A = 0$

- **Trace:** Simply, just the sum of the elements on the diagonal (top left to bottom right) of A. While it doesn't have a great visual interpretation, we usually use it as a trick in proofs because of two properties it has[13]
- **Rank and the Column Space:** the column space is the name for the image of the function: it's every point y that can be expressed as $y = Ax$ for some x. The name comes from the fact that (as it turns out) this is equal to the span of the vectors which make up the columns of A. The rank of a matrix is the dimension of the column space. The situation mentioned above where all of the vectors get squashed into a smaller line is one where the column space is 1-dimensional, even though the input space was 2-dimensional. For the same reasons as above, this gives us another way to look at invertability: A is invertible if and only if A has full rank - rank which is the same as the size as the input

We use all of these properties fluently when doing proofs, so it's good to be aware of them. I'll talk about them in class. We can see that it's useful to understand both ideas of what a matrix is (a table and a linear transformation): concepts such as the transpose and the trace are easy to understand if we think of a matrix as a table, but hard if we think of it as a transformation; and the determinant has a really complicated formula used to calculate it, but the visual image is actually quite simple

[13] That $\text{tr}(AB) = \text{tr}(BA)$ for any matrices A, B and tr applied to a constant is that constant (as a constant is a 1×1 matrix). The trick is used because we can't ordinarily swap the order of matrices: remember AB refers to doing B and then A, and BA refers to doing A and then B - if A represents flipping a vector in the x axis and B represents rotation 90^o then we can see that the order matters! However, in some special cases we can write a constant as equalling the tr of that constant, then use the $\text{tr}(AB) = \text{tr}(BA)$ property. This is used heavily in proofs looking at the variance of estimators

Slightly more advanced concepts

These aren't necessarily harder to understand, but rely on the previous concepts to define. They come up in proofs and in understanding metrics, so they're worth looking at:

Positive Definiteness

It's easy to say what it means for a number to be positive or negative, but how can we generalise this to matrices? Unfortunately one approach that doesn't work is taking our definition to be that all of the elements are positive, as we can find matrices which have this but also have behaviour that positive numbers don't. A better definition comes from considering the following: if we take any number $a > 0$, then for any $x \in \mathbb{R}$ that's not 0, it's always true that $ax^2 > 0$. Similarly, if $a < 0$ then $ax^2 < 0$, and if $a = 0$ then $ax^2 = 0$. Positive definiteness is all about generalising this to matrices. The matrix version[14] of the expression ax^2 is $x'Ax$, which is known as a **quadratic form**. Simply: a matrix A is positive definite $x'Ax > 0$ for all vectors $x \neq 0$ and negative definite if $x'Ax < 0$ for all $x \neq 0$. If $x'Ax \geq 0$ for all x, but possibly 0 for some $x \neq 0$, we call it positive semi-definite (psd). Similarly, we can define negative semi-definite. We care about doing this because:

- Some objects we care about can be written as quadratic forms, such as the F and T-test statistics, so we can analyse them directly like this
- Many objects we care about are positive definite or psd matrices, such as the matrix $X'X$ which is key to OLS. Given positive definiteness is roughly the matrix version of being positive, it might also make sense that positive definite matrices are always invertible (since multiplying by A^{-1} is the matrix version of 'dividing by A'). Theorems about how pd / psd matrices behave can help us prove things about methods where they're a part

Projections and Idempotency

Think about how an old-school cinema projector works: some light gets shone at a screen, and produces an image there. If you moved the projector a little closer to the screen, and then suitably adjusted the angle and settings, you could get it to produce the same image on the screen. In general in physics, a projection is all about multiple inputs getting mapped to the same output. Here: there are a big set of places you could place a projector (in 3d space) and still create the same image on a screen (in 2d space). This is the rough concept we are trying to take for ourselves

How can we formalise this? If we think about it, a definition that works is that a projection is a function where applying it twice gives the same result as applying it once. Therefore applying it any number of times gives the same result as applying it once: once you project an image onto the closest point to it on the screen, applying that transformation again won't do anything (as the closest point to it on the screen is now itself)

[14] A sidenote: the idea that pd is the 'matrix version' of positive works as good intuition for why the second derivative test for a local minimum of a function $f(x_1, ..., x_n)$ is that its Hessian at the point you want to test is pd. Since the Hessian is the matrix of second derivatives of the function, this is just the matrix version of the usual test for a one-dimensional function: that is, $f''(x^*) > 0$

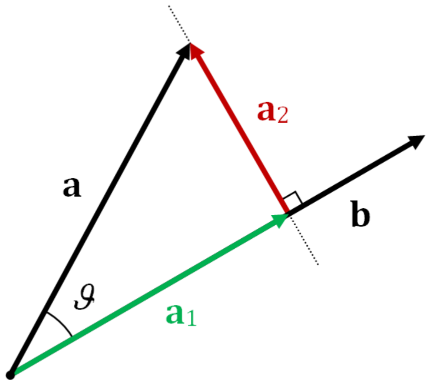

Figure 3: From Wikipedia: Vector Projection of a onto the space spanned by b is a_1

In function notation, a projection is when $f(x) = f(f(x))$ for all x. For matrices, this happens when $AA = A$, and if so we say that A is **idempotent**[15]. Projections and Idempotency are essentially referring to the same thing

We care about this in metrics because some of the matrices we end up using are idempotent, and so we can use theorems about idempotent matrices to help with proofs. Also it can help us with some intuition: it turns out we can think of the predicted value of y by OLS - what you get by running the predict command in Stata - as Py, where P is an idempotent matrix ($PP = P$). This gives us an interpretation of OLS - it's projecting y onto a certain space, which turns out to be the space spanned by the columns of X. I'll talk about this more when we get to it[16]

Diagonalisation and Eigen-stuff

One thing to note is that the grid lines act as a co-ordinate system for $\mathbb{R}^2$ - you could communicate any point to someone else by telling them its value on the grid, and you could do this in a unique way[17]. We motivated a linear transformation by saying that, while it might stretch, squish or rotate grid lines, it doesn't bend them or make them curved. If the matrix doesn't squash things down to a smaller number of dimensions, then we can use these new grid lines as a *different co-ordinate system* for expressing the points in $\mathbb{R}^2$.[18] In fact we can say more: it turns out that if we have the co-ordinates of a point in the one co-ordinate system (basis) - call them x, the co-ordinates in the new system are given by $A^{-1}x$

[15] Coming from the Greek root of 'the same power' - doing it once has the same effect as doing it many times

[16] For those of you who have taken a more advanced linear algebra course such as MA212, this might seem like I'm talking about a Moore-Penrose Pseudoinverse here (a generalisation of an inverse which exists under more general conditions). In fact, the OLS estimator $\hat{\beta}$ is **exactly** the left pseudoinverse of y

[17] In case you've not thought about it in this way before, we define a *basis* as a linearly independent set which spans the whole space - these are exactly the conditions for this set to act as a 'good' coordinate system (every point can be expressed in it, in one way only)

[18] Formally: if A is invertible and $e_1, ...e_n$ are basis vectors for the space, then so are $Ae_1, ..., Ae_n$

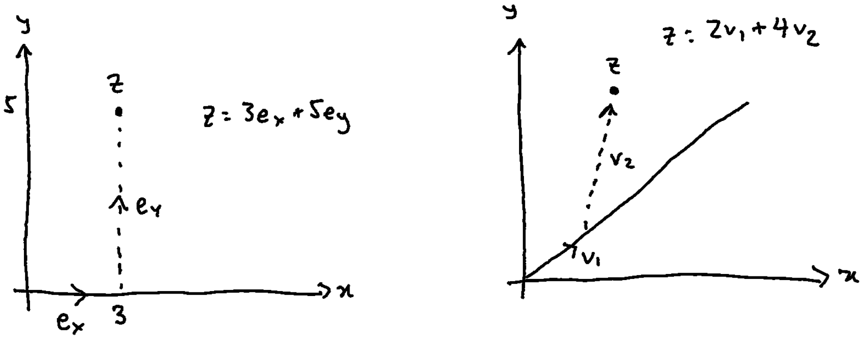

Figure 4: Representing z in a new basis

What is diagonalisation all about? It comes from the the idea that some linear transformations are much nicer than others. As we saw above, if A, B, C, D are linear transformations then so is $ABCD$, which (although still not breaking any of our rules about grid lines) might be a complicated mixture of stretches, squishes, rotations, and other things all done in a specific order. In some sense, the 'nicest' transformations are those which do something very simple: just stretches. These are represented by the diagonal matrices:

$$D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{bmatrix}$$

because if you put in any $x = [x_1, ..., x_n]$ after it you'll see that $Dx = [\lambda_1 x_1, ..., \lambda_n x_n]$ is just a stretch of x - it doesn't do any rotations or anything like that

We know that not all of our complicated linear transformations can be written as just stretches[19]. **What diagonalisation does** is to ask a different question: if we used a different co-ordinate system from the standard one, could we think of it as a stretch then? In other words, we're looking for some matrix D such that for any z in the new co-ordinate system, $z' = Dz$ represents the same thing as A. The co-ordinates in the new system are $z = P^{-1}x$ and $z' = P^{-1}x'$ for some P[20], where $x' = Ax$. If we can do this, we'll have matrices P and D such that $P^{-1}x' = DP^{-1}x$ for all x. Plugging in $x' = Ax$ gives $P^{-1}Ax = DP^{-1}x$, and so $Ax = PDP^{-1}x$ for all x. As this holds no matter which x we put in, we have that:

$$A = PDP^{-1}$$

If we take a step back: what we've concluded so far is that if there exists a different co-ordinate system in which A becomes only a stretch, we can write $A = PDP^{-1}$ for some matrices P and D. In other words, we can do the transformation represented by A by firstly transforming to new

[19] For example, a rotation can't be, if we're restricting D to contain real numbers

[20] Which reflects which co-ordinates we are changing into

co-ordinates (P^{-1}), then applying a stretch (D), then transforming back to the original co-ordinates ($P = (P^{-1})^{-1}$)

Eigenvalues and Eigenvalues are all about how we find P and D. It turns out that the best choice for our new grid lines are vectors which don't change after having A applied to them - they only stretch. To find these, we solve the *eigenvalue problem*: find a $\lambda \in \mathbb{R}$ and a vector v such that

$$Av = \lambda v$$

Then simply, P is the matrix which has these v as its columns, and D has the eigenvalues λ on the diagonal

You'll probably be pleased to hear that we don't have to think through all of this logic every time we apply diagonalisation. It's helpful to understand the story of where it comes from, but for proofs the important thing is that you understand which matrices can be diagonalised, and that if it can be then it can be written as $A = PDP^{-1}$ for an invertible matrix P and a diagonal matrix D. One set of such matrices which can be are the symmetric[21] matrices. If you have a symmetric matrix in front of you, it's often helpful just to write it in its diagonalised form, and see if that helps! Often we analyse quadratic forms in this way

Calculus in Matrix notation

The main idea is that, in the spirit of using matrices to do our metrics because they can be neater to work with, we often want to collect all of our derivatives into matrices. For example, if the gradient $\nabla f(x)$ is defined as usual:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_k} \end{bmatrix}$$

then we can see that if $f(x) = \sum_i a_i x_i$ for constants a_i then:

$$\nabla f(x) = a$$

since $\partial f / \partial x_i = a_i$, and we form the gradient ∇f by stacking all of these partial derivatives

Because of this we make the following definition: if a and x are both vectors, and $f(x) = a'x$, then

$$\nabla f(x) = a$$

By a similar argument we can define other calculus 'rules' for matrices, such as:

$$\nabla\ x'Ax = (A + A')x \qquad \text{and} \qquad \nabla\ x'x = 2x$$

[21] ie, $A = A'$. More technically, their entries need to also be real numbers

We care about the Gradient ∇f (the vector of first derivatives) and the Hessian $\nabla^2 f$ (the matrix of second derivatives) because they can help us do maximisation. If we have a function $f : \mathbb{R}^n \to \mathbb{R}$, then we (locally) minimise f by finding the vector x^* such that

$$\nabla f(x^*) = 0 \qquad \text{and} \qquad \nabla^2 f(x^*) \text{ is positive semi definite}$$

as this is the 'matrix version' of the 1-dimensional case - $f : \mathbb{R} \to \mathbb{R}$ is (locally) minimised where we have that $f'(x^*) = 0$ and $f''(x^*) > 0$

We can prove the rules we've written above - Q4 of PS1 is effectively asking us to do that. After this, we can talk about maximising functions purely in matrix notation, so we never leave the world of matrices. This gives us very clean proofs - see the document on the derivation of OLS using matrices on Github