2. Linear algebra for applied statistics

- Linear algebra is the math of vectors and matrices.
- In statistics, the main purpose of linear algebra is to organize data and write down the manipulations we want to do to them.
- A **vector** of length n is also called an n-**tuple** or a **sequence** of length n.
- We can suppose that each data point is a **real number**. We write $\mathcal{R}$ for the set of real numbers, and $\mathcal{R}^n$ for the set of vectors of n real numbers.
- Write the US life expectancy at birth for 2011 to 2015 as $\mathbf{y} = (y_1, y_2, y_3, y_4, y_5) = (79.0, 79.1, 79.0, 79.0, 78.9)$. We see $\mathbf{y} \in \mathcal{R}^5$.
- A numerical dataset with n datapoints is a vector in $\mathcal{R}^n$.
- Qualitative data can be written as a vector in $\mathcal{R}^n$ by assigning a number for each qualitative level.
- We use bold font for vectors, and italic font for **elements** of the vector.

More perspectives on vectors

Question 2.1. You may or may not have seen vectors in other contexts. In physics, a vector is a quantity with magnitude and direction. How does that fit in with our definition?
This physics definition is a polar coordinate representation of a vector in $\mathcal{R}^2$ and $\mathcal{R}^3$. Our definition is a Cartesian coordinate representation.

Question 2.2. How can I write a boldface $\mathbf{x}$ in handwriting?
An underscore, $\underset{\sim}{x}$, is conventional handwriting to represent bold font. In physics and mathematics, vectors are sometimes written as $\vec{x}$, but we will not do that here.

Adding vectors and multiplying by a scalar

- For a dataset, the **index** i of the element y_i of the vector y might correspond to a measurement on the ith member of a population, the outcome of the ith group in an experiment, or the ith observation out of a sequence of observations on a system. Recall i is called an **observational unit**, or just **unit**.
- We might want to add two quantities u_i and v_i for unit i.
- Using vector notation, if $\mathbf{u} = (u_1, u_2, \ldots, u_n)$, $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ we define the **vector sum** $\mathbf{y} = \mathbf{u} + \mathbf{v}$ to be the **elementwise sum** $y_i = u_i + v_i$, adding up the corresponding elements for each unit.
- We might also want to rescale each element by the same factor. To change a measurement y_i in inches to a new measurement z_i in mm, we rescale with the **scalar** $\alpha = 25.4$. We want $z_i = \alpha y_i$ for each i. This is written in vector notation as **multplication of a vector by a scalar**, $\mathbf{z} = \alpha \mathbf{y}$.

An example of vector addition and scalar multiplication

An ecologist measures the pH of ten Michigan lakes at two points in the summer. Set up vector notation to describe her data algebraically. Write a vector calculation giving the average pH in each lake.

- Let x_i be the first pH measurement in lake i, for $i \in \{1, 2, \ldots, 10\}$.
- Write $\mathbf{x} = (x_1, \ldots, x_{10})$ for the vector of the first pH measurement in each of the 10 lakes.
- Let $\mathbf{y} = (y_1, \ldots, y_{10})$ be the vector of second measurements.
- Let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{10})$ be the average pH for each of the 10 lakes.
- For each lake i, the mean is $\mu_i = \frac{1}{2}(x_i + y_i)$.
- In vector notation, this is

$$\boldsymbol{\mu} = \frac{1}{2}(\mathbf{x} + \mathbf{y})$$

.

Vectors and scalars in R

- We have seen in Chapter 1 that R has vectors. An R vector of length 1 is a scalar, and we have seen that R follows the usual mathematical rules of vector addition and multiplication by a scalar.
- R also allows adding a scalar to a vector

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

```
x+2

## [1] 3 4 5
```

- Mathematically, adding scalars to vectors is **not allowed**. Instead, we define the **vector of ones**, $\mathbf{1} = (1, 1, \ldots, 1)$, and write $\mathbf{x} + 2 \times \mathbf{1}$.

Question 2.3. Why does R choose to break the usual rules of mathematics here?

Convenience. It is quite common to want to add a fixed amount to each data point, so R makes this easy.

Matrices

- Matrices let us store and manipulate p quantitites for each of n units.
- An $n \times p$ matrix $\mathbb{A}$ is a numerical array with n rows and p columns,

$$\mathbb{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{np} \end{bmatrix}$$

- Blackboard bold capital letters, $\mathbb{A}$, $\mathbb{B}$, $\mathbb{X}$, $\mathbb{Z}$, etc, denote matrices.
- We are reserving plain capital letters for random variables.
- We say $\mathbb{A} = [a_{ij}]_{n \times p}$ as an abbreviation for writing the full $n \times p$ matrix.
- We write $\underset{n \times p}{\mathbb{A}}$ to emphasize the dimension.
- Keeping track of matrix dimensions can be helpful!

Matrix addition

- matrices are added elementwise.
- We can only add matrices with the same dimensions.
- Let $\mathbb{A} = [a_{ij}]_{n \times p}$ and $\mathbb{B} = [b_{ij}]_{n \times p}$.
- Since both $\mathbb{A}$ and $\mathbb{B}$ are $n \times p$ we can add them

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{np} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & & & \vdots \\ b_{n1} & b_{n2} & \dots & b_{np} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1p} + b_{1p} \\ a_{21} + b_{22} & a_{22} + b_{22} & \dots & a_{2p} + b_{2p} \\ \vdots & & & \vdots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \dots & a_{np} + b_{np} \end{bmatrix}$$

- Focus on one element (say, the top left element in red) and then check that the pattern repeats.

Matrix addition in R

- R agrees with our mathematical definition of matrix addition.
- If `A` and `B` are two R matrices of matching dimension, then

```
C <- A + B
```

gives a matrix with the elementwise sums.

- If the dimensions of `A` and `B` don't match, R gives an error message.
- Curiously, R does allow adding a vector to a matrix using the recycling rule. This is **not allowed** in math.

```
A <- matrix(1:6,2,3)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
A+c(0,10)
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]   12   14   16
```

Matrix multiplication

- The rule for matrix multiplication seems strange at first. We'll see why it is useful.
- The (i, j) entry of the product $\mathbb{A}\,\mathbb{B}$ comes from multiplying each element in row i of $\mathbb{A}$ with the corresponding element in column j of $\mathbb{B}$ and adding up these terms.
- For this rule to work, the number of columns of $\mathbb{A}$ must match the number of rows of $\mathbb{B}$.
- We start with the 2×2 case, where $\mathbb{A} = [a_{ij}]_{2\times 2}$ and $\mathbb{B} = [b_{ij}]_{2\times 2}$.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

- In red: notice how the $(1, 2)$ entry of $\mathbb{A}\,\mathbb{B}$ comes from **sliding** row 1 of $\mathbb{A}$ down column 2 of $\mathbb{B}$ and adding corresponding terms.

Let's practice multiplying 2×2 matrices

Question 2.4. Evaluate the matrix product

$$\begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ -2 & 0 \end{bmatrix}$$

Answer:

$$\begin{bmatrix} 4 & 8 \\ -5 & 4 \end{bmatrix}$$

The dimension of a matrix product

- We can multiply $\underset{n\times p}{\mathbb{A}}\ \underset{p\times q}{\mathbb{B}}$ and the result is an $n \times q$ matrix.

Question 2.5. What do you think is the dimension of the triple product

$$\underset{n\times p}{\mathbb{A}}\ \underset{p\times q}{\mathbb{B}}\ \underset{q\times r}{\mathbb{C}}$$

- The rule is: when multiplying matrices, the middle dimensions have to pair up, and the resulting matrix has the outside dimensions.
- Notice that $\underset{p\times q}{\mathbb{B}}\ \underset{n\times p}{\mathbb{A}}$ does not exist.
- In general, $\mathbb{A}\,\mathbb{B} \neq \mathbb{B}\,\mathbb{A}$. Matrix multiplication **is not commutative**.

Matrix multiplication in R

- R interprets the usual multiplication operator $*$ as elementwise multiplication.
- Matrix multiplication is done with %*%.
- Let's check this by an example.

```
U <- matrix(c(2,1,2,-1),2)
U
##      [,1] [,2]
## [1,]    2    2
## [2,]    1   -1
V <- matrix(c(3,1,1,2),2)
V
##      [,1] [,2]
## [1,]    3    1
## [2,]    1    2
```

```
U * V
##      [,1] [,2]
## [1,]    6    2
## [2,]    1   -2
U %*% V
##      [,1] [,2]
## [1,]    8    6
## [2,]    2   -1
```

Matrix multiplication and linear equations

- A linear system of n equations with p unknown variables, $x_1, \ldots, x_p$ is

$$\left.\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}\,x_2 & + & \ldots & + & a_{1p}\,x_p & = & b_1 \\ a_{21}x_1 & + & a_{22}\,x_2 & + & \ldots & + & a_{2p}\,x_p & = & b_2 \\ \vdots & & & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}\,x_2 & + & \ldots & + & a_{np}\,x_p & = & b_n \end{array}\right\} \quad \text{(L1)}$$

- The rule for matrix multiplication lets us write this as $\mathbb{A}\mathbf{x} = \mathbf{b}$.

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1p} \\ a_{21} & a_{22} & \ldots & a_{2p} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{np} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- In red: the first entry in the matrix product $\mathbb{A}\mathbf{x}$ comes from sliding the first row of $\mathbb{A}$ down the column of values in $\mathbf{x}$, multiplying each pair of numbers, and adding up these products. This matches the left hand side of the first line of (L1).

Using matrices to solve a system of linear equations

- We've seen how matrices can represent a system of linear equations as $\mathbb{A}\mathbf{x} = \mathbf{b}$.
- For a basic linear algebra equation $ax = b$ we would divide through by a, or equivalently multiply through by a^{-1}, to find $x = a^{-1}b$ when $a \neq 0$.
- Is there a **matrix inverse** $\mathbb{A}^{-1}$ such that $\mathbf{x} = \mathbb{A}^{-1}\mathbf{b}$ solves the system of linear equations $\mathbb{A}\mathbf{x} = \mathbf{b}$?
- We will see that there is an inverse $\mathbb{A}^{-1}$ when the system of linear equations has a unique solution. Since software can compute this inverse, we can solve systems of linear equations easily. This is useful in statistics for fitting linear models to datasets. Understanding when this inverse exists, and what to do when it doesn't, will help us develop appropriate models for data analysis.
- From the previous slide, we can only expect $\mathbb{A}^{-1}$ to exist when $p = n$, in which case $\mathbb{A}$ is called a **square matrix**.

A matrix product example

Question 2.6. Let $\mathbb{U} = \begin{bmatrix} 2 & 2 \\ 1 & -1 \end{bmatrix}$ and $\mathbb{V} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$. Calculate $\mathbb{U}\mathbb{V}$.

We can check our working in R.

```
U <- matrix(
   c(2,1,2,-1),2)
U
```

```
##      [,1] [,2]
## [1,]    2    2
## [2,]    1   -1
```

```
V <- matrix(
   c(3,1,1,2),2)
V
```

```
##      [,1] [,2]
## [1,]    3    1
## [2,]    1    2
```

```
U %*% V
```

```
##      [,1] [,2]
## [1,]    8    6
## [2,]    2   -1
```

Addition of matrices and multiplication by a scalar

- If $\mathbb{A} = [a_{ij}]_{p\times q}$ and $\mathbb{B} = [b_{ij}]_{p\times q}$ then the **matrix sum** $\mathbb{A} + \mathbb{B}$ is computed elementwise, just like for vectors:

$$\begin{bmatrix} a_{11} & \dots & a_{1q} \\ \vdots & \ddots & \vdots \\ a_{p1} & \dots & a_{pq} \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \dots & b_{pq} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & \dots & a_{1q}+b_{1q} \\ \vdots & \ddots & \vdots \\ a_{p1}+b_{p1} & \dots & a_{pq}+b_{pq} \end{bmatrix}$$

- **Scalar times matrix** multiplication is also computed elementwise:

$$s\mathbb{A} = s\begin{bmatrix} a_{11} & \dots & a_{1q} \\ \vdots & \ddots & \vdots \\ a_{p1} & \dots & a_{pq} \end{bmatrix} = \begin{bmatrix} s\,a_{11} & \dots & s\,a_{1q} \\ \vdots & \ddots & \vdots \\ s\,a_{p1} & \dots & s\,a_{pq} \end{bmatrix}$$

- Scalar times matrix multiplication does commute: $s\mathbb{A} = \mathbb{A}s$.
- Matrix and scalar multiplication both have a **distributive** property: $\mathbb{U}(\mathbb{V}+\mathbb{W}) = \mathbb{U}\mathbb{V} + \mathbb{U}\mathbb{W}$, and $s(\mathbb{V}+\mathbb{W}) = s\mathbb{V} + s\mathbb{W}$,

The identity matrix

- The $p \times p$ **identity matrix**, $\mathbb{I}_p$, is a square matrix with 1's on the diagonal and 0's everywhere else:

$$\mathbb{I}_p = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

- Check that for any $p \times p$ matrix $\mathbb{A}$, we have $\mathbb{I}_p\mathbb{A} = \mathbb{A}\mathbb{I}_p = \mathbb{A}$. Also, for any vector $\mathbf{v} \in \mathcal{R}^p$ we have $\mathbb{I}_p\mathbf{v} = \mathbf{v}$.

- We can often write $\mathbb{I}$ in place of $\mathbb{I}_p$ since the dimension of $\mathbb{I}$ is always evident from the context.

Question 2.7. Suppose $\mathbb{B}$ is a $n \times q$ matrix and $\mathbb{I}$ is an identity matrix.
(i) If we write $\mathbb{B}\mathbb{I}$, what must be the dimension of $\mathbb{I}$? Find a simplification of $\mathbb{B}\mathbb{I}$.

(ii) How about if we write $\mathbb{I}\mathbb{B}$?

Inverting a 2×2 matrix

- Let $\mathbb{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ be a general 2×2 matrix.
- Let $\mathbb{A}^{-1} = \dfrac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

Question 2.8. Compute the matrix product $\mathbb{A}\,\mathbb{A}^{-1}$

- $\mathbb{A}^{-1}$ is called the **inverse** of $\mathbb{A}$.

The identity matrix

- $\mathbb{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is the 2×2 **identity matrix**.
- We can check that $\mathbb{A}\,\mathbb{A}^{-1} = \mathbb{A}^{-1}\,\mathbb{A} = \mathbb{I}$.
- Also, for any 2×2 matrix $\mathbb{B}$, $\mathbb{I}\,\mathbb{B} = \mathbb{B}\mathbb{I} = \mathbb{B}$.
- We see that $\mathbb{I}$ plays the role of 1 in the usual arithmetic.
- For numbers we are familiar with the corresponding results $a^{-1}\dot{a} = a\dot{a}^{-1} = 1$ and $1 \times a = a \times 1 = a$.

The determinant of a 2×2 matrix

- Recall that $\mathbb{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$
- We call $ad - bc$ the **determinant** of $\mathbb{A}$, and we write

$$\det(A) = ad - bc.$$

- We can see from the formula for $\mathbb{A}^{-1}$ that the inverse of $\mathbb{A}$ exists if and only if $ad - bc \neq 0$.

Finding the matrix inverse and determinant in R

- The R function `det()` finds the determinant of a square matrix, and `solve()` finds the inverse if it exists.

```
A <- matrix(runif(9),3,3)
round(A,2)
```

```
##      [,1] [,2] [,3]
## [1,] 0.27 0.91 0.94
## [2,] 0.37 0.20 0.66
## [3,] 0.57 0.90 0.63
```

```
A_inv <- solve(A)
round(A_inv,2)
```

```
##       [,1]  [,2]  [,3]
## [1,] -2.18  1.30  1.91
## [2,]  0.68 -1.75  0.82
## [3,]  1.02  1.32 -1.33
```

```
A %*% A_inv
```

```
##                [,1] [,2] [,3]
## [1,]  1.000000e+00    0    0
## [2,] -1.110223e-16    1    0
## [3,]  0.000000e+00    0    1
```

```
det(A) ; det(A_inv)
```

```
## [1] 0.2139161
## [1] 4.674729
```

Question 2.9. Why is `A%*%A_inv` not exactly equal to the identity

Using R to solve a set of linear equations

Worked example. Suppose we want to solve

$$\begin{array}{rcrcrcrcl} w & + & 2x & - & 3y & + & 4z & = & 0 \\ 2w & - & 2x & + & y & + & z & = & 1 \\ -w & - & x & + & 4y & - & z & = & 2 \\ 3w & - & x & - & 8y & + & 2z & = & 3 \end{array}$$

How do we do this using R?

1. Write the system as a matrix equation $\mathbb{A}\mathbf{x} = \mathbf{b}$,

$$\begin{bmatrix} 1 & 2 & -3 & 4 \\ 2 & -2 & 1 & 1 \\ -1 & -1 & 4 & -1 \\ 3 & -1 & -8 & 2 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

Using R to solve a set of linear equations, continued...

2. Enter the matrix $\mathbb{A}$ and vector $\mathbf{b}$ into R.

```
A <- rbind( c( 1, 2,-3, 4),
            c( 2,-2, 1, 1),
            c(-1,-1, 4,-1),
            c( 3,-1,-8, 2))
b <- c(0,1,2,3)
```

3. Compute the matrix solution to the linear system, $\mathbf{x} = \mathbb{A}^{-1}\mathbf{b}$.

Question 2.10. Which of these correctly computes $\mathbf{x}$ and why?

```
round(solve(A) %*% b,2)
```

```
##       [,1]
## [1,] -3.75
## [2,] -3.58
## [3,] -0.80
## [4,]  2.13
```

```
round(solve(A) * b,2)
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,]  0.00  0.00  0.00  0.00
## [2,] -0.09  0.24 -1.15 -0.51
## [3,]  0.00  0.40 -0.40 -0.40
## [4,]  1.09 -0.44  2.35  0.71
```

The transpose of a matrix

- Sometimes we want to switch the rows and columns of a matrix.
- For example, we usually suppose that each column of a data matrix is a measurement variable (say, height and weight) and each row of a data matrix is an object being measured (say, a row for each person). However, what if the data were stored in a matrix where columns corresponded to objects?
- Switching rows and columns is called **transposing** the matrix.
- The **transpose** of $\mathbb{A}$ is denoted mathematically by $\mathbb{A}^{\mathrm{T}}$ and in R by `t(A)`.

$$\mathbb{A} = \begin{bmatrix} 1 & 2 & -3 \\ 2 & -2 & 1 \\ -1 & -1 & 4 \\ 3 & -1 & -8 \end{bmatrix}, \qquad \mathbb{A}^{\mathrm{T}} = \begin{bmatrix} 1 & 2 & -1 & 3 \\ 2 & -2 & -1 & -1 \\ -3 & 1 & 4 & -8 \end{bmatrix}$$

- If $\mathbb{A}$ has dimension $n \times p$, then $\mathbb{A}^{\mathrm{T}}$ is $p \times n$.

More properties of matrices

- The following material is not essential for this course. However, it may help reinforce understanding to see more ways in which matrix addition and multiplication behave similarly, or differently, from usual arithmetic.

Associative property. We are used to the associative property of addition and multiplication for numbers: $a + (b + c) = (a + b) + c$ and $a \times (b \times c) = (a \times b) \times c$. You can check that matrix addition and multiplication also have the associative property: for matrices of appropriate size, $\mathbb{A} + (\mathbb{B} + \mathbb{C}) = (\mathbb{A} + \mathbb{B}) + \mathbb{C}$ and $\mathbb{A}(\mathbb{B}\,\mathbb{C}) = (\mathbb{A}\,\mathbb{B})\mathbb{C}$.

Inverse of a product. For square invertible matrices $\mathbb{A}$ and $\mathbb{B}$, we can check that $(\mathbb{A}\mathbb{B})^{-1} = \mathbb{B}^{-1}\mathbb{A}^{-1}$. The change of order may seem weird. To demonstrate that this inverse works correctly,

$$(\mathbb{A}\,\mathbb{B})^{-1}(\mathbb{A}\,\mathbb{B}) = \mathbb{B}^{-1}\mathbb{A}^{-1}\mathbb{A}\,\mathbb{B} = \mathbb{B}^{-1}\mathbb{I}\,\mathbb{B} = \mathbb{B}^{-1}\mathbb{B} = \mathbb{I}.$$

Note that we have repeatedly used the associative property of matrix multiplication, and we have been careful not to accidentally commute (recall that, in general, $\mathbb{C}\,\mathbb{D} \neq \mathbb{D}\,\mathbb{C}$).

More properties of matrices, continued

Transpose of a sum. Convince yourself that $(\mathbb{A}+\mathbb{B})^{\mathrm{T}} = \mathbb{A}^{\mathrm{T}} + \mathbb{B}^{\mathrm{T}}$. If you like, calculate an example in R to check.
Transpose of a product. The rule is $(\mathbb{A}\,\mathbb{B})^{\mathrm{T}} = \mathbb{B}^{\mathrm{T}}\mathbb{A}^{\mathrm{T}}$.
Question 2.11. Suppose that $\mathbb{A}$ has dimension $n \times p$ and $\mathbb{B}$ is $p \times q$. Check that this formula for $(\mathbb{A}\,\mathbb{B})^{\mathrm{T}}$ has the right dimension.

Example:

```
A <- matrix(1:3,4,3); B <- matrix(1:6,3,2)
```

```
t(A %*% B)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   14   11   11   14
## [2,]   32   29   29   32
```

```
t(B) %*% t(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   14   11   11   14
## [2,]   32   29   29   32
```

More properties of matrices, continued

A matrix commutes with its inverse: $\mathbb{A}^{-1}\mathbb{A} = \mathbb{A}\mathbb{A}^{-1} = \mathbb{I}$.

- Recall that, in general, matrix multiplication does not commute ($\mathbb{AB} \neq \mathbb{BA}$).
- We can check, using R, that a matrix does commute with its inverse.

```
A <- matrix(runif(9),3,3)
```

```
A_inv <- solve(A)
```

```
round( A %*% A_inv, 3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

```
round( A_inv %*% A, 3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```