Algebraic manipulator: Vectors, matrices and lists

Software in Chemistry

Manuel Pérez Escribano & Raquel Rubert Albiol

Universitat de València

Algebraic Manipulator: Vectors

- Vectors
  - Vectors in Mathematica are written as lists
  - Lists are sets of mathematical objects separated by commas and closed in curly braces { }

```
In[1]:= vec1 = {3, 5, 8}
Out[1]= {3, 5, 8}
```

Algebraic Manipulator: Vectors

– You can operate with lists, made up of both numbers and variables

```
In[8]:= vector1 = {1, 3, 7}
        vector2 = {a, b, c}

Out[8]= {1, 3, 7}

Out[9]= {a, b, c}

In[10]:= 3 * vector1
         Pi * vector1
         número pi
         Pi * {x, y, z}
         número pi
         vector1 + vector2
         vector1 * vector2

Out[10]= {3, 9, 21}

Out[11]= {π, 3 π, 7 π}

Out[12]= {π x, π y, π z}

Out[13]= {1 + a, 3 + b, 7 + c}

Out[14]= {a, 3 b, 7 c}
```

Algebraic Manipulator: Vectors

- Assigning a list to a variable is very useful because we can refer to the *nth* element of a list just by typing:
- list[[n]]
- In this way, we can refer to a specific component

```
In[15]:= v = {4, 5, 7}
         w = {a, b, c}

Out[15]= {4, 5, 7}

Out[16]= {a, b, c}

In[17]:= vx = v[[1]]
         wy = w[[2]]

Out[17]= 4

Out[18]= b

In[19]:= vx + wy

Out[19]= 4 + b
```

Algebraic Manipulator: Vectors

- Scalar product of two vectors is calculated by:
  - Multiplying Item by Item
  - Using the dot . as an operator between two vectors
  - Using theDot[v,w] function

```
In[20]:= v = {4, 5, 7}
         w = {a, b, c}

Out[20]= {4, 5, 7}

Out[21]= {a, b, c}

In[22]:= v[[1]] * w[[1]] + v[[2]] * w[[2]] + v[[3]] * w[[3]]

Out[22]= 4 a + 5 b + 7 c
```

```
In[23]:= v.w

Out[23]= 4 a + 5 b + 7 c

In[24]:= Dot[v, w]
         producto escalar

Out[24]= 4 a + 5 b + 7 c
```

Algebraic Manipulator: Vectors

- Vector product:

-It is calculated using the

Cross[vec1,vec2] function

```
In[31]:= vec1 = {3, 5, 8};
         vec2 = {4, 3, 1};

In[33]:= Cross[vec1, vec2]
         producto vectorial

Out[33]= {-19, 29, -11}
```

Algebraic Manipulator: Vectors

- Modulus of a vector
  - It can be calculated using the Norm[vec1] function
  - It is defined by the formula:

$$\vec{v} = (v1, v2, v3)$$

$$\|\vec{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

```
In[3]:= vec1 = {3, 5, 8}
        modvec1 = Sqrt[vec1[[1]]^2 + vec1[[2]]^2 + vec1[[3]]^2]
                  raíz cuadrada
        modvec1 = Norm[vec1]
                  norma
Out[3]= {3, 5, 8}
Out[4]= 7 √2
Out[5]= 7 √2
```

Algebraic Manipulator: Vectors

- Unit vector is the vector whose modulus is 1
- It is calculated using the Normalize[vector] function
  - It is defined by the formula:

$$\vec{v} = (v1, v2, v3)$$

$$\vec{v} = \frac{\vec{v}}{\|\vec{v}\|}$$

```
In[16]:= vec1 = {3, 5, 8}
         vecuni1 = vec1 / Norm[vec1]
                           norma
         vecuni1 = Normalize[vec1]
                           normaliza
Out[16]= {3, 5, 8}
```

$$\text{Out[17]} = \left\{\frac{3}{7\sqrt{2}}, \frac{5}{7\sqrt{2}}, \frac{4\sqrt{2}}{7}\right\}$$

$$\text{Out[18]} = \left\{\frac{3}{7\sqrt{2}}, \frac{5}{7\sqrt{2}}, \frac{4\sqrt{2}}{7}\right\}$$

```
In[19]:= Norm[vecuni1]
         norma
Out[19]= 1
```

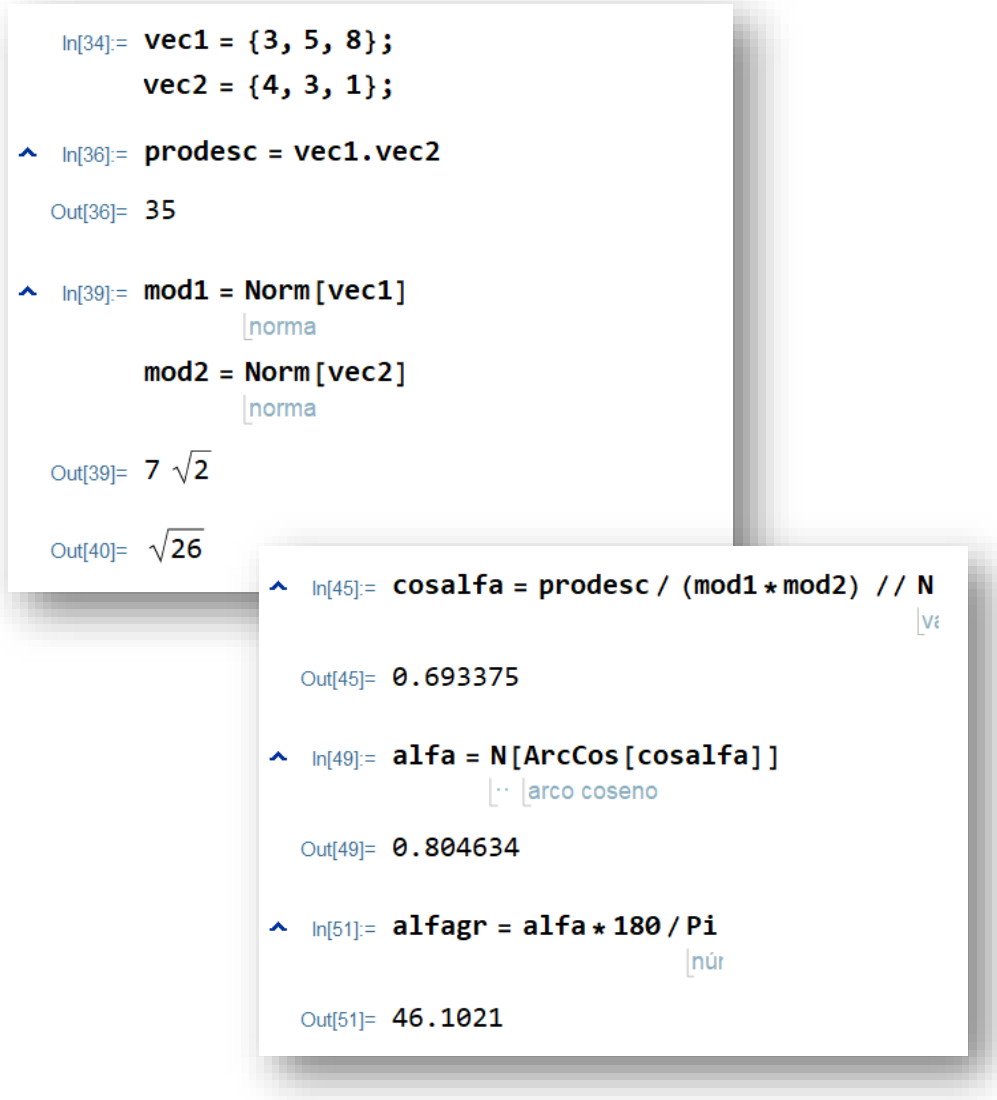
Algebraic Manipulator: Vectors

- We can also calculate the angle formed by two vectors. The formula we use is:

$$\cos(\alpha) = \frac{vector1 \cdot vector2}{\|vector1\| \cdot \|vector2\|}$$

- Therefore, we can calculate the angle as:

$$\alpha = \arccos\left(\frac{vector1 \cdot vector2}{\|vector1\| \cdot \|vector2\|}\right) = \arccos(\cos(\alpha))$$



Algebraic Manipulator: Matrices

- Matrices
  - To build a matrix, we can use curly brackets {{x1,x2},{x3,x4}}
  - Or, the Class Assistant (Palettes):
  - Control + "," to add columns
  - Control + *Enter* to generate rows

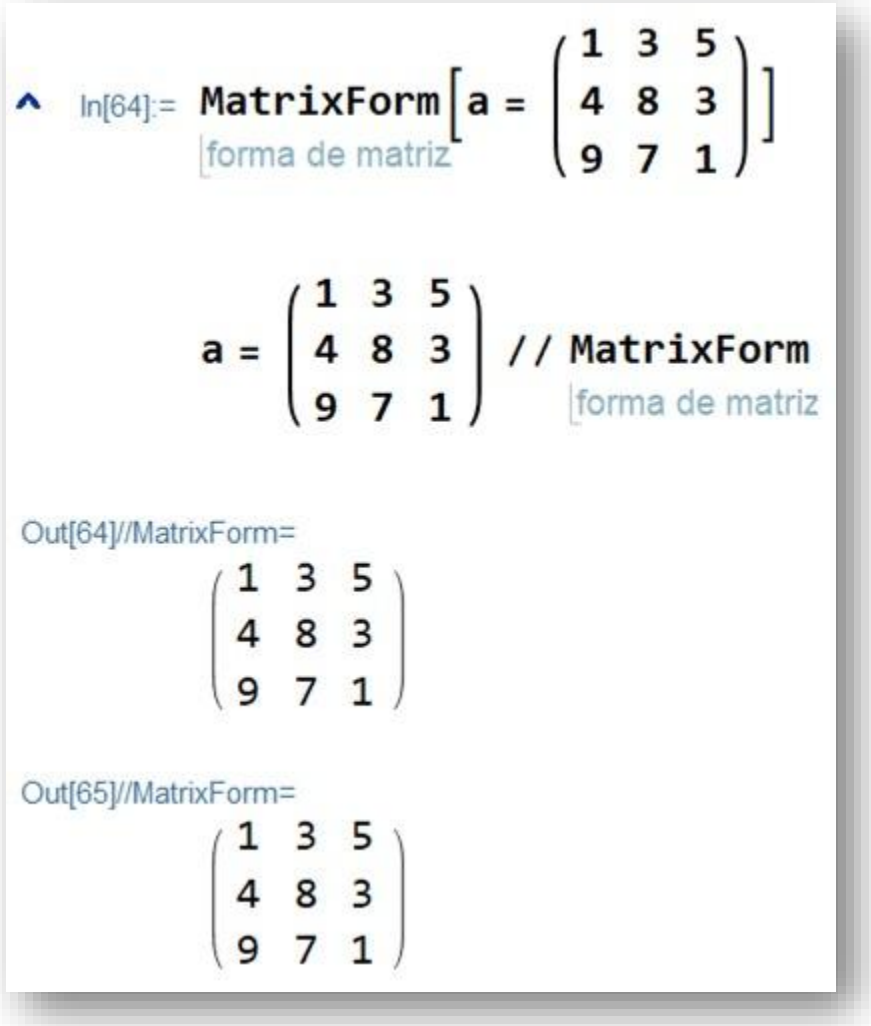
$$a = \begin{pmatrix} x1 & x2 \\ x3 & x4 \end{pmatrix}$$

$$b = \begin{pmatrix} x1 & x2 & x3 \\ x4 & x5 & x6 \\ x7 & x8 & x9 \end{pmatrix}$$

```
c = {{x1, x2}, {x3, x4}}
```

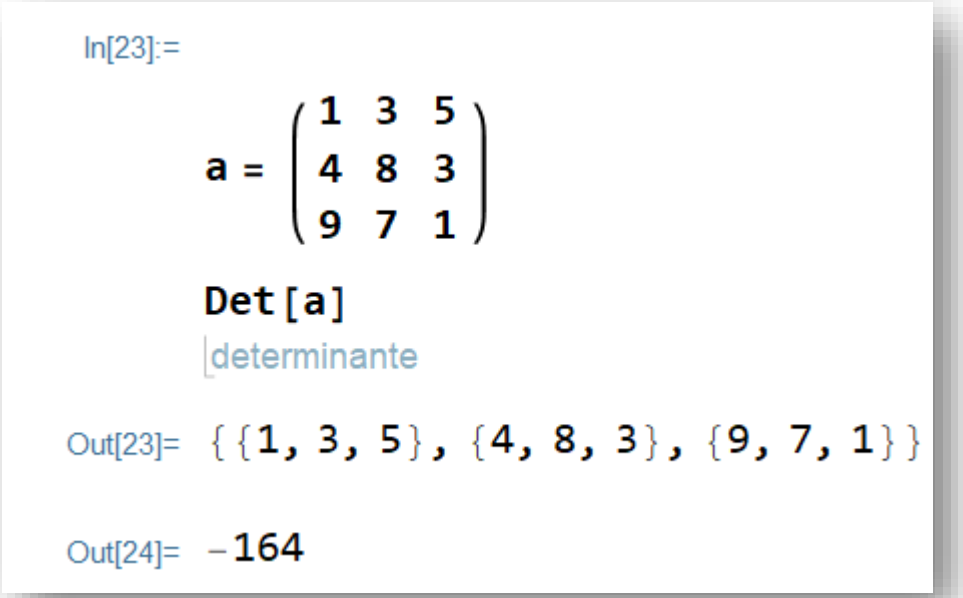
Algebraic Manipulator: Matrices

- To obtain the result in the form of a matrix we use the MatrixForm function
- Can be written at the end of the matrix using //MatrixForm
- You can also write it at the beginning MatrixForm[]



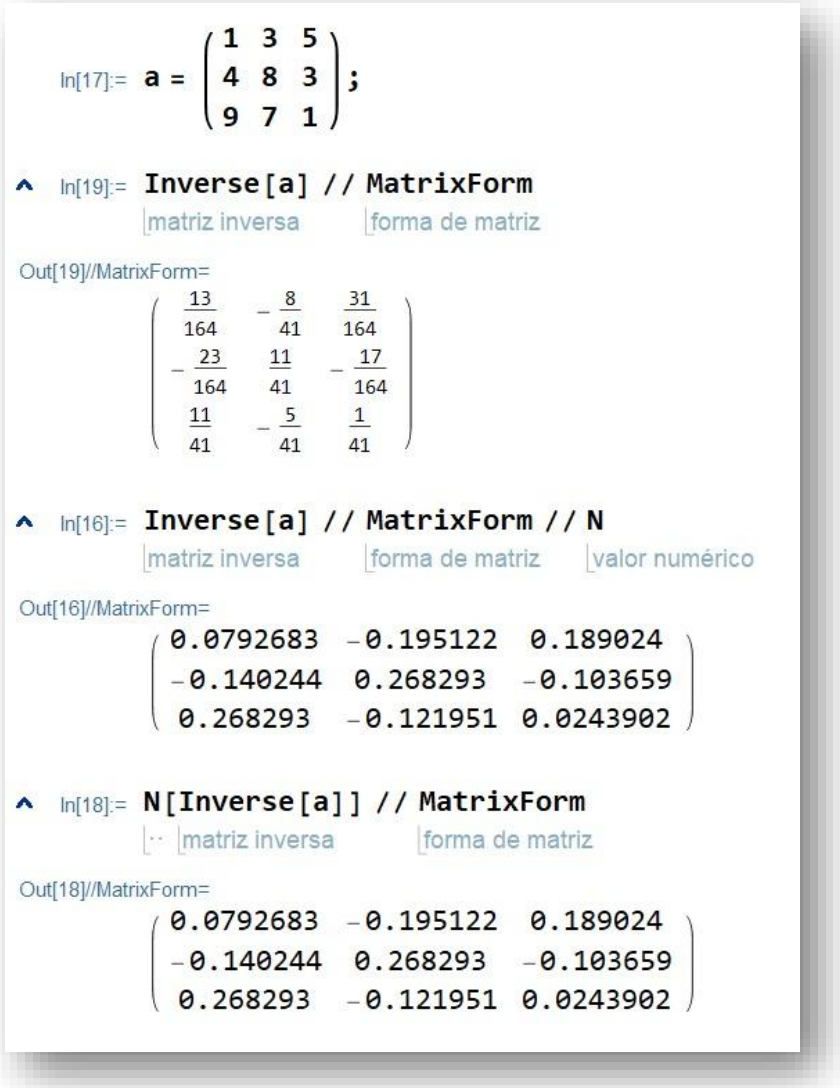
Algebraic Manipulator: Matrices

- To calculate the determinant of a matrix:
  - We will use the Det[Matriz] function



Algebraic Manipulator: Matrices

- To calculate the inverse matrix of another matrix:
  - We use the function Inverse[Matrix]



Algebraic Manipulator: Matrices

- To multiply matrices:
  - The operator "." should be used

In[25]:=

$$a = \begin{pmatrix} 1 & 3 & 5 \\ 4 & 8 & 3 \\ 9 & 7 & 1 \end{pmatrix};$$

$$b = \begin{pmatrix} 8 & 3 & 5 \\ 4 & 0 & 4 \\ 9 & 7 & 1 \end{pmatrix};$$

```
a.b // MatrixForm
```

forma de matriz

Out[27]//MatrixForm=

$$\begin{pmatrix} 65 & 38 & 22 \\ 91 & 33 & 55 \\ 109 & 34 & 74 \end{pmatrix}$$

Algebraic Manipulator: Matrices

- The inverse of a matrix multiplied by the same matrix results in the unit matrix

In[28]:=

$$a = \begin{pmatrix} 1 & 3 & 5 \\ 4 & 8 & 3 \\ 9 & 7 & 1 \end{pmatrix};$$

```
Inverse[a] // MatrixForm
```
matriz inversa | forma de matriz

```
a.Inverse[a] // MatrixForm
```
matriz inversa | forma de matriz

Out[29]//MatrixForm=

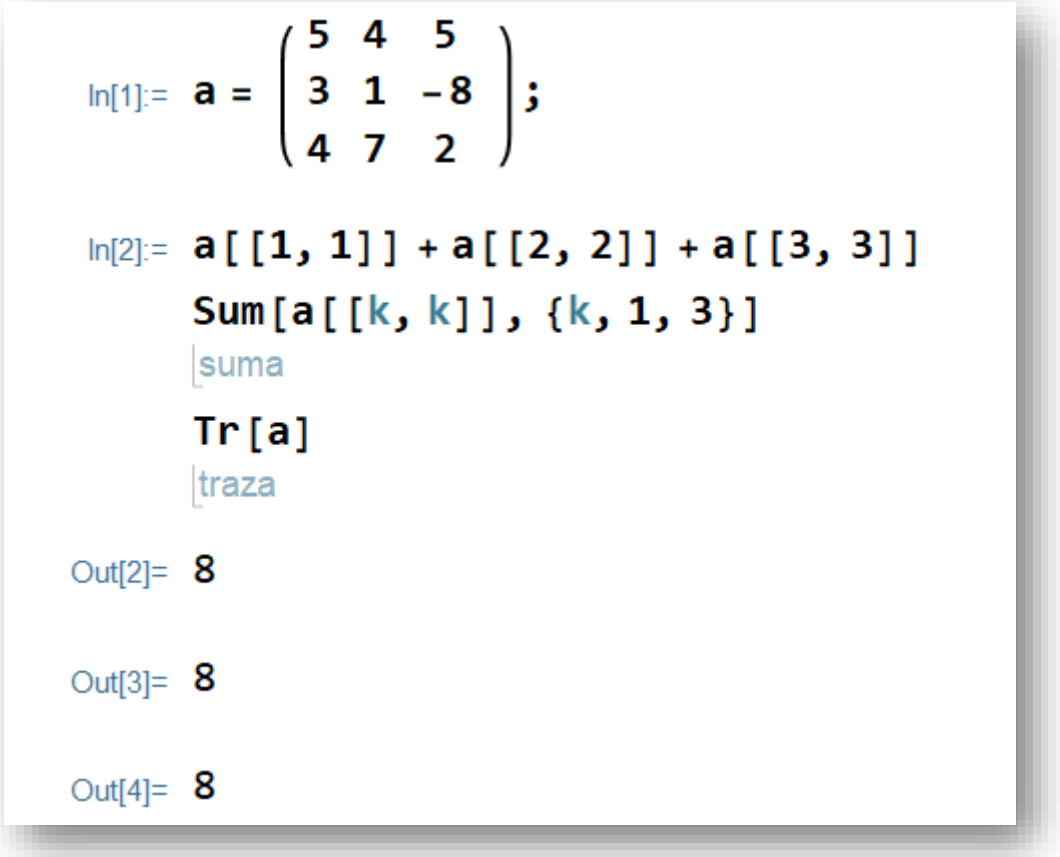
$$\begin{pmatrix} \frac{13}{164} & -\frac{8}{41} & \frac{31}{164} \\ -\frac{23}{164} & \frac{11}{41} & -\frac{17}{164} \\ \frac{11}{41} & -\frac{5}{41} & \frac{1}{41} \end{pmatrix}$$

Out[30]//MatrixForm=

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

Algebraic Manipulator: Matrices

- The trace of a matrix is the sum of the elements of the main diagonal
  - It can be calculated by manually adding the terms
  - With the function Sum[]
  - With the function Tr[Matrix]



Algebraic Manipulator: Lists

- Lists
  - Remember that a list is a set of mathematical objects separated by commas and closed with curly brackets {}.

```
In[36]:= llista1 = {1, 2, 3}
Out[36]= {1, 2, 3}

In[37]:= 2 * llista1
Out[37]= {2, 4, 6}
```

Algebraic Manipulator: Lists

- Lists can also be constructed with the function Table[]
  - Construct a list with the same item repeated n times
  - We can also build a list by repetition of lists

```
In[38]:= Table[x, 5]
         tabla

Out[38]= {x, x, x, x, x}
```

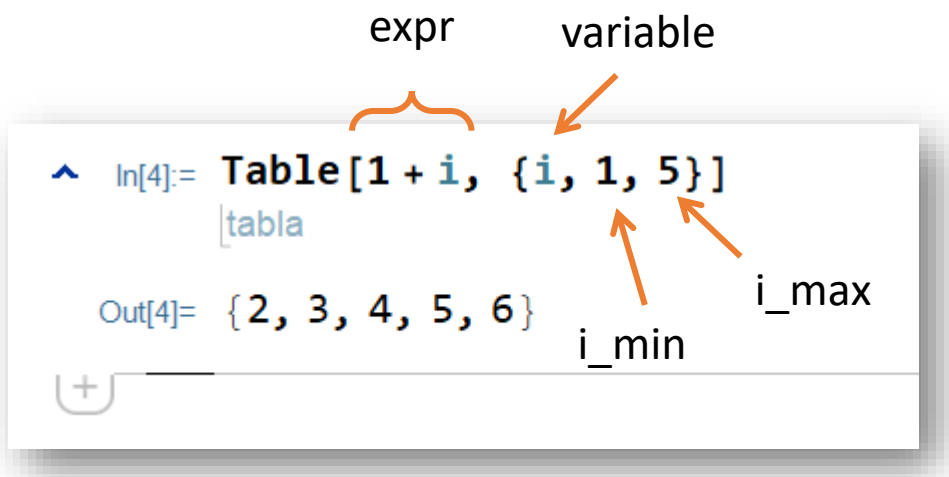
```
In[39]:= Table[{1, 2}, 4]
         tabla

Out[39]= {{1, 2}, {1, 2}, {1, 2}, {1, 2}}
```

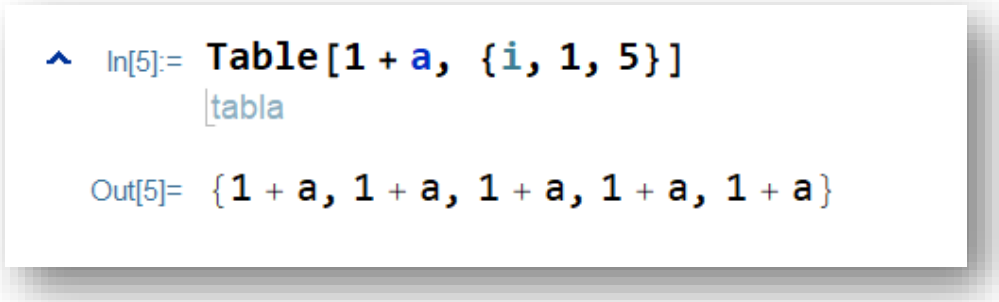
Algebraic Manipulator: Lists

– At the same time, we can construct a table from an expression based on *i*, where the values of *i* vary from a minimum to a maximum value

Table[expr(*i*), {*i*, *i_min*, *i_max*}]

– If *i* does not appear in the expression, expression replicas are obtained





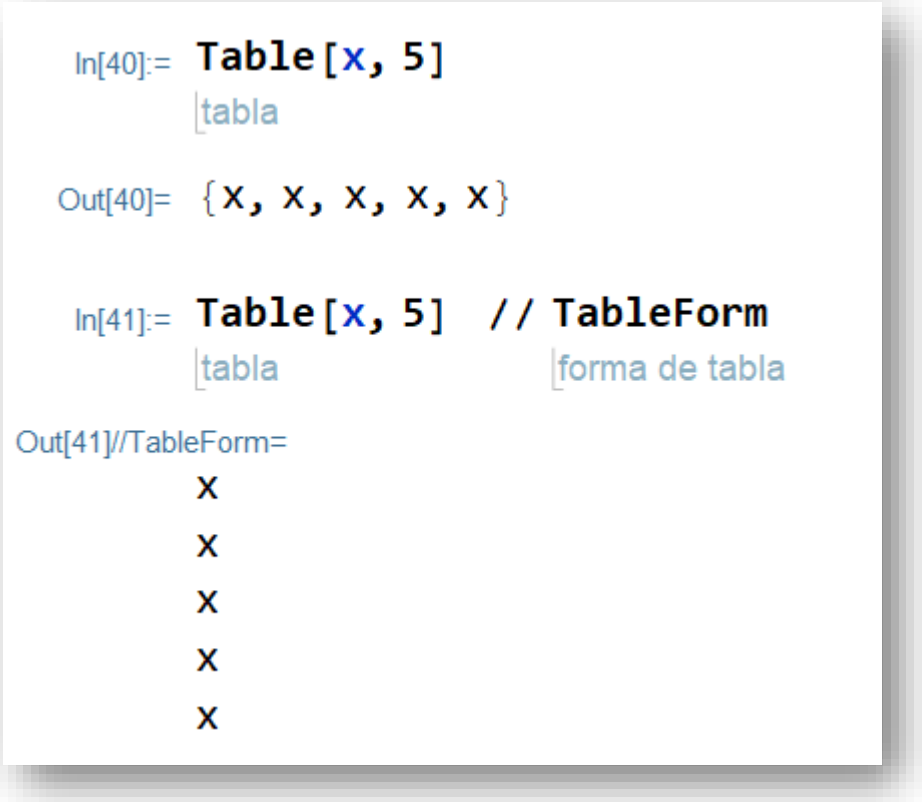
Algebraic Manipulator: Lists

– We can add additional instructions for two variables:

  - Table[expr, {*i*, *i_min*, *i_max*}, {*j*, *j_min*, *j_max*}]

– Table[expr, {*i*, *i_min*, *i_max*, *incre*}] increments *i* between *i_min* and *i_max* using *incre* as increment

```
In[13]:= Table[a * i + j, {i, 1, 5}, {j, 1, 3}]
         tabla

Out[13]= {{1 + a, 2 + a, 3 + a},
          {1 + 2 a, 2 + 2 a, 3 + 2 a}, {1 + 3 a, 2 + 3 a, 3 + 3 a},
          {1 + 4 a, 2 + 4 a, 3 + 4 a}, {1 + 5 a, 2 + 5 a, 3 + 5 a}}

In[12]:= Table[i a, {i, 0, 20, 2}]
         tabla

Out[12]= {0, 2 a, 4 a, 6 a, 8 a, 10 a, 12 a, 14 a, 16 a, 18 a, 20 a}
```

Algebraic Manipulator: Lists

- To maintain the table formatting we can use the command //TableForm