23-CAVAN-reduccion-gaussiana

November 4, 2017

echelonize()

```
In [1]: A = matrix(QQ,[[1,2],[3,4]])

In [2]: A.echelonize()

In [3]: A

Out[3]: [1 0]
        [0 1]
```

Volvemos a definir A e intentamos asignar a la matriz reducida un nuevo nombre B:

```
In [4]: A = matrix(QQ,[[1,2],[3,4]]);B = A.echelonize(); A; B
```

Ahora B no existe y A sigue siendo la reducida. la asignación $B = A.echelonize()$ no hace nada. Sin embargo podemos copiar el nuevo valor de A en una nueva matriz B:

```
In [5]: A = matrix(QQ,[[1,2],[3,4]]);A.echelonize(); B = copy(A)

In [6]: show(A)

[1 0]
[0 1]


In [7]: show(B)

[1 0]
[0 1]
```

Con $B = copy(A)$ podemos reproducir la matriz A, que es la reducida de la A original, en la nueva matriz B, pero vemos que la A original se ha perdido al ejecutar $A.echelonize()$.

echelon_form()

```
In [8]: A = matrix(QQ,[[1,2],[3,4]]);B=A.echelon_form()

In [9]: show(A)
```

```
[1 2]
[3 4]
```

```
In [10]: show(B)
```

```
[1 0]
[0 1]
```

Vemos que la matriz A se mantiene y podemos asignar un nombre, B, a la reducida. Si no asignamos un nombre a la reducida de A, sólo la vemos, pero no podremos calcular con ella:

```
In [11]: A = matrix(QQ,[[1,2],[3,4]]);A.echelon_form()
```

```
Out[11]: [1 0]
         [0 1]
```

echelon_form usando echelonize

Podemos reproducir el comportamiento de $echelon\_form()$ usando $echelonize()$:

```
In [12]: A = matrix(QQ,[[1,2],[3,4]])
         B = copy(A)
         A.echelonize()
         C = copy(A)
         A = copy(B)
         B = copy(C)
         show(A)
         show(B)
```

```
[1 2]
[3 4]
```

```
[1 0]
[0 1]
```