Análisis componentes principales: <svd> (sin toolboxes extra) versus <pca> (statistics and ML toolbox)

Este código ejecutó sin errores en Matlab **R2022a**.

Presentación en vídeo: http://personales.upv.es/asala/YT/V/pcaissvd.html

Objetivo: ilustrar que los comandos "**svd**" y "**pca**" producen el mismo resultado.

Generar Datos

```
N=1000;
X0=randn(N,4); %four standard normal random variables
TT=diag([4.5 4 3 2])*[1 2 3 3;1 3 -1 2;0 0 0.06 0.4;0.2 0 0.1 0.6]';
X=X0*TT+[5 4 3 -1]; %scale and mix things and move mean
```

Procesar Datos

Si los datos fueran en unidades arbitrarias no comparables, estandarizamos:

```
X=zscore(X);
std(X)
```

```
ans = 1×4
    1.0000    1.0000    1.0000    1.0000
```

```
[U,S,V]=svd(X,'econ');
[coeff,score,latent]=pca(X); %Statistics and ML toolbox
```

Los dos comandos sacan lo mismo. Comparemos:

```
diag(S)'/sqrt(N-1) %correction needed to have "standard dev." units as output
```

```
ans = 1×4
    1.5966    0.9637    0.5694    0.4448
```

```
sqrt(latent')
```

```
ans = 1×4
    1.5966    0.9637    0.5694    0.4448
```

```
V
```

```
V = 4×4
    0.5318    0.2772   -0.8002   -0.0025
    0.3924    0.7454    0.5186    0.1464
    0.5086   -0.5109    0.1590    0.6746
    0.5518   -0.3264    0.2559   -0.7235
```

```
coeff %identical to V except for sign changes (irrelevant)
```

```
coeff = 4×4
    0.5318    0.2772    0.8002    0.0025
    0.3924    0.7454   -0.5186   -0.1464
    0.5086   -0.5109   -0.1590   -0.6746
    0.5518   -0.3264   -0.2559    0.7235
```

```
%US=XV
U*S %identical to "score" except for sign changes (irrelevant)
```

```
ans = 1000×4
   -0.3910   -1.1917    0.2408   -0.0411
    1.2578   -2.9915   -0.1253    0.0271
    3.7607   -0.8059   -1.6580   -0.7293
   -1.2287   -0.9800   -0.1331   -0.0762
    0.8531    0.3515   -0.1200   -0.2461
    0.9066    0.8012   -0.2194    0.3800
    2.4905   -0.6085    0.7167   -0.3315
   -0.2843   -1.8776   -0.0315    0.7120
   -0.5703   -1.8373    0.6291    0.1043
    0.4006   -0.2137    0.0128    0.0145
     ⋮
```

```
score
```

```
score = 1000×4
   -0.3910   -1.1917   -0.2408    0.0411
    1.2578   -2.9915    0.1253   -0.0271
    3.7607   -0.8059    1.6580    0.7293
   -1.2287   -0.9800    0.1331    0.0762
    0.8531    0.3515    0.1200    0.2461
    0.9066    0.8012    0.2194   -0.3800
    2.4905   -0.6085   -0.7167    0.3315
   -0.2843   -1.8776    0.0315   -0.7120
   -0.5703   -1.8373   -0.6291   -0.1043
    0.4006   -0.2137   -0.0128   -0.0145
     ⋮
```

```
std(score) %should be S/sqrt(N-1) or latent variable variance
```

```
ans = 1×4
    1.5966    0.9637    0.5694    0.4448
```