Universidad Nacional Autónoma de México
Facultad de Ingeniería

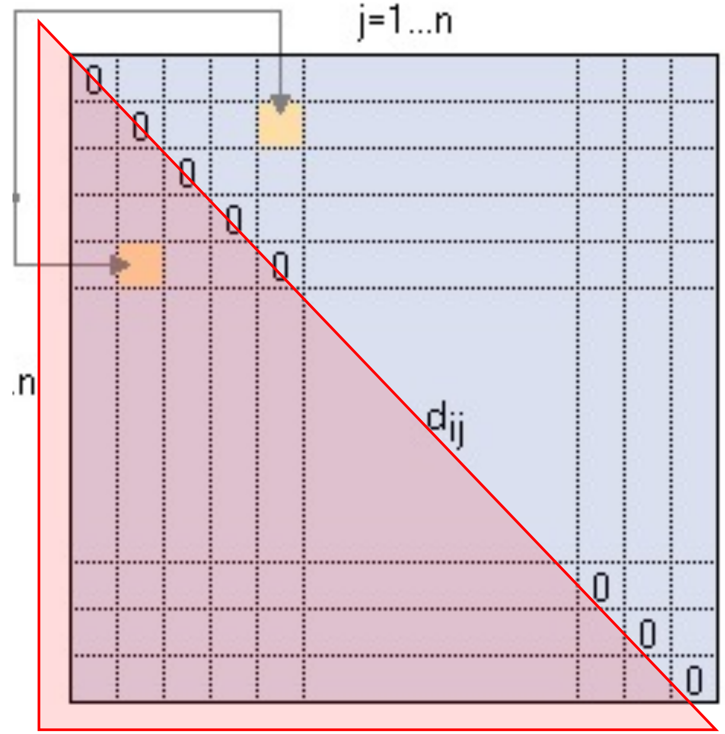
Métricas de distancia
Práctica 3

Guillermo Molero-Castillo
guillermo.molero@ingenieria.unam.edu

Septiembre, 2021

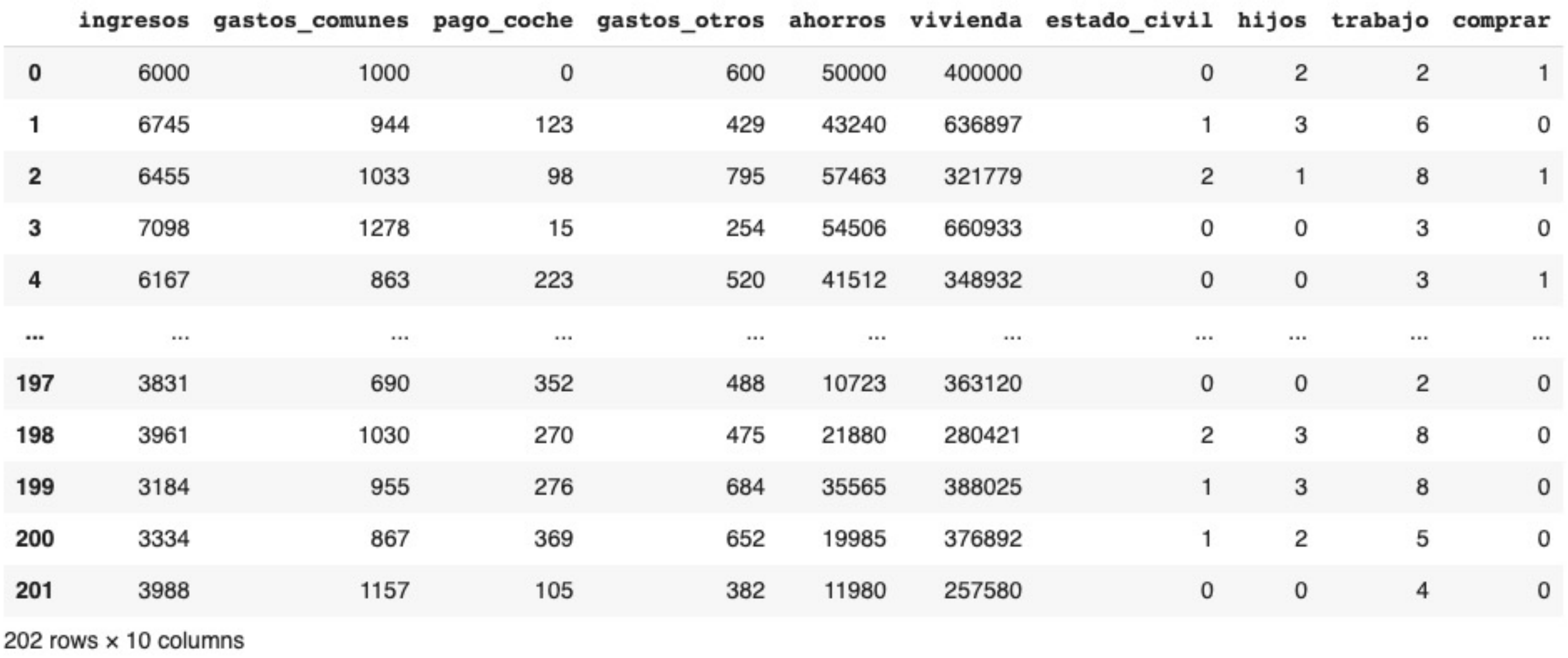
Práctica

Objetivo. Obtener las matrices de distancia (Euclidiana, Chebyshev, Manhattan, Minkowski) en Google Colab a partir de una matriz de datos.



Práctica

Fuente de datos



	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
0	6000	1000	0	600	50000	400000	0	2	2	1
1	6745	944	123	429	43240	636897	1	3	6	0
2	6455	1033	98	795	57463	321779	2	1	8	1
3	7098	1278	15	254	54506	660933	0	0	3	0
4	6167	863	223	520	41512	348932	0	0	3	1
...	...	...	...	...	...	...	...	...	...	...
197	3831	690	352	488	10723	363120	0	0	2	0
198	3961	1030	270	475	21880	280421	2	3	8	0
199	3184	955	276	684	35565	388025	1	3	8	0
200	3334	867	369	652	19985	376892	1	2	5	0
201	3988	1157	105	382	11980	257580	0	0	4	0

202 rows × 10 columns

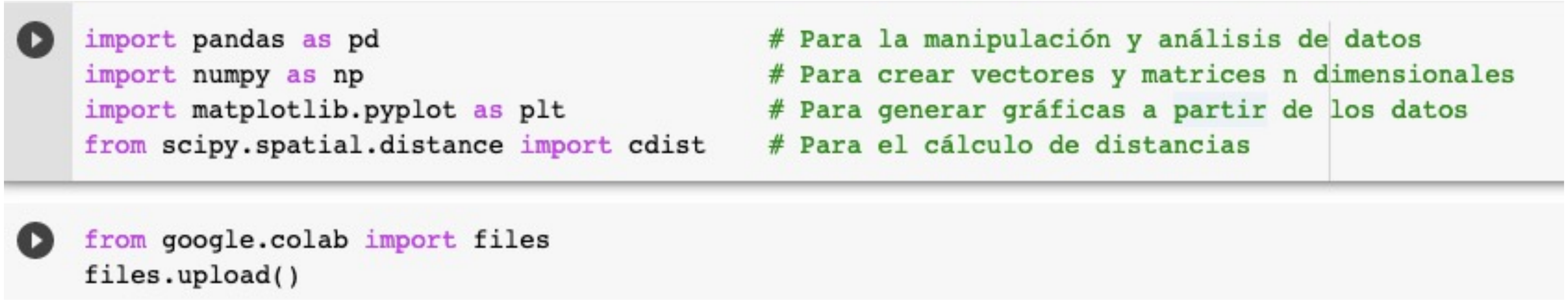
Práctica

Fuente de datos

- ingresos: son ingresos mensuales de 1 o 2 personas, si están casados.
- gastos_comunes: son gastos mensuales de 1 o 2 personas, si están casados.
- pago_coche
- gastos_otros
- ahorros
- vivienda: valor de la vivienda.
- estado_civil: 0-soltero, 1-casado, 2-divorciado
- hijos: cantidad de hijos menores (no trabajan).
- trabajo: 0-sin trabajo, 1-autonomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autonomo, 8-empresarios o empresario y autónomo
- comprar: 0-alquilar, 1-comprar casa a través de crédito hipotecario con tasa fija a 30 años.

Práctica

1. Importar las bibliotecas necesarias

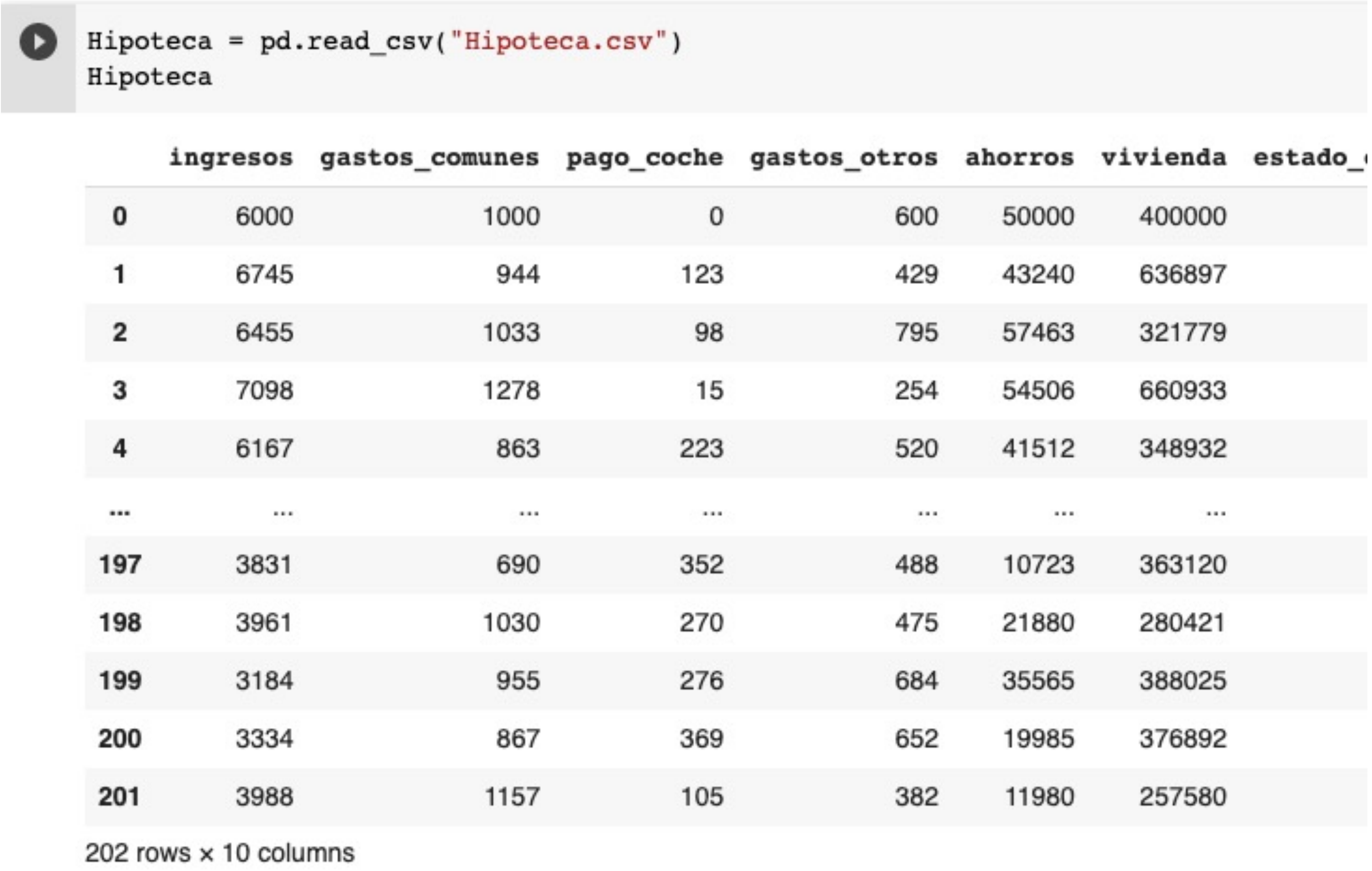


```
import pandas as pd                        # Para la manipulación y análisis de datos
import numpy as np                         # Para crear vectores y matrices n dimensionales
import matplotlib.pyplot as plt            # Para generar gráficas a partir de los datos
from scipy.spatial.distance import cdist   # Para el cálculo de distancias
```

```
from google.colab import files
files.upload()
```

Práctica

2. Importar los datos



```
Hipoteca = pd.read_csv("Hipoteca.csv")
Hipoteca
```

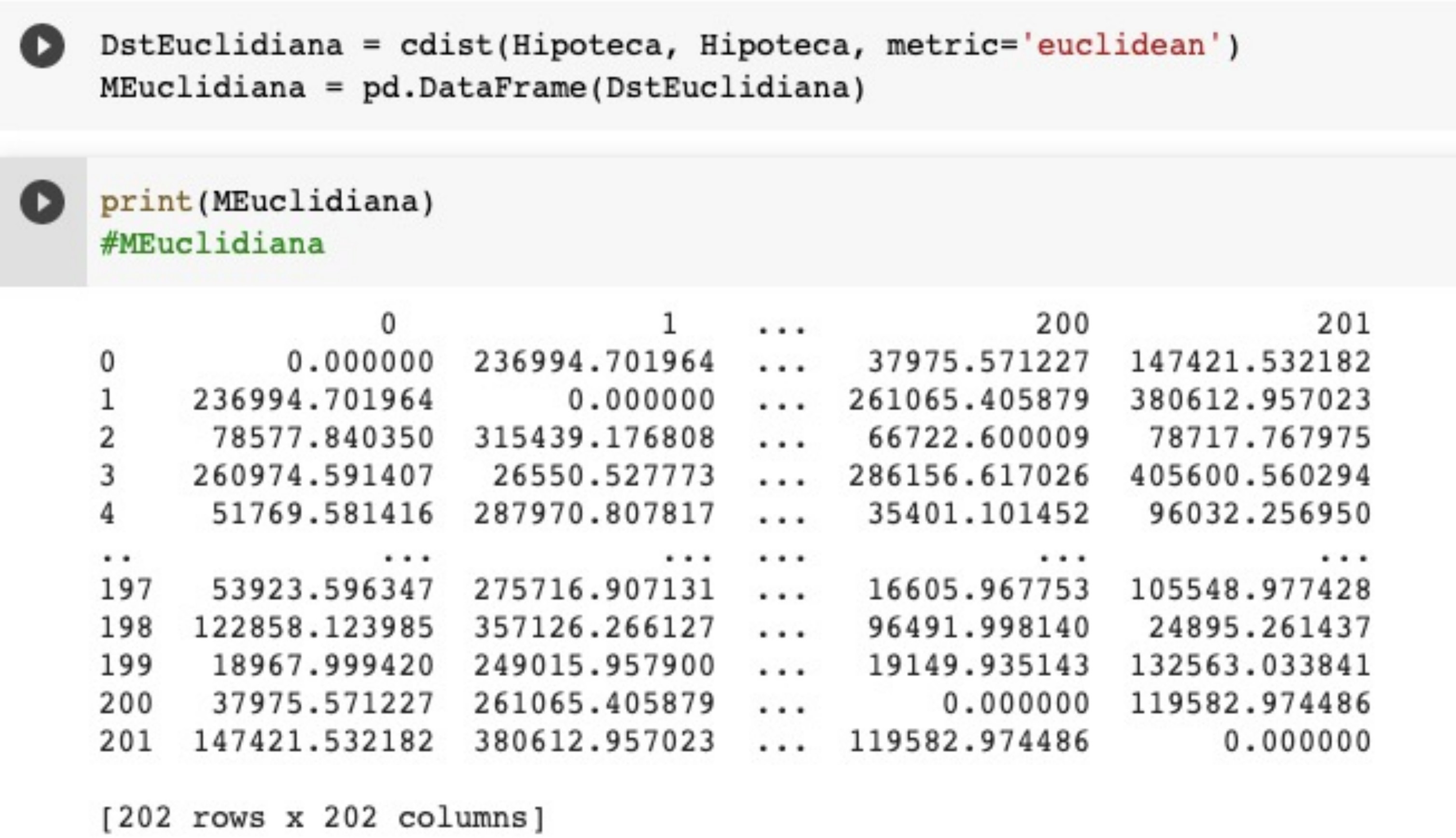
	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_
0	6000	1000	0	600	50000	400000	
1	6745	944	123	429	43240	636897	
2	6455	1033	98	795	57463	321779	
3	7098	1278	15	254	54506	660933	
4	6167	863	223	520	41512	348932	
...	...	...	...	...	...	...	
197	3831	690	352	488	10723	363120	
198	3961	1030	270	475	21880	280421	
199	3184	955	276	684	35565	388025	
200	3334	867	369	652	19985	376892	
201	3988	1157	105	382	11980	257580	

202 rows × 10 columns

Práctica

3. Matrices de distancias

a) Euclidiana



```
DstEuclidiana = cdist(Hipoteca, Hipoteca, metric='euclidean')
MEuclidiana = pd.DataFrame(DstEuclidiana)
```

```
print(MEuclidiana)
#MEuclidiana
```

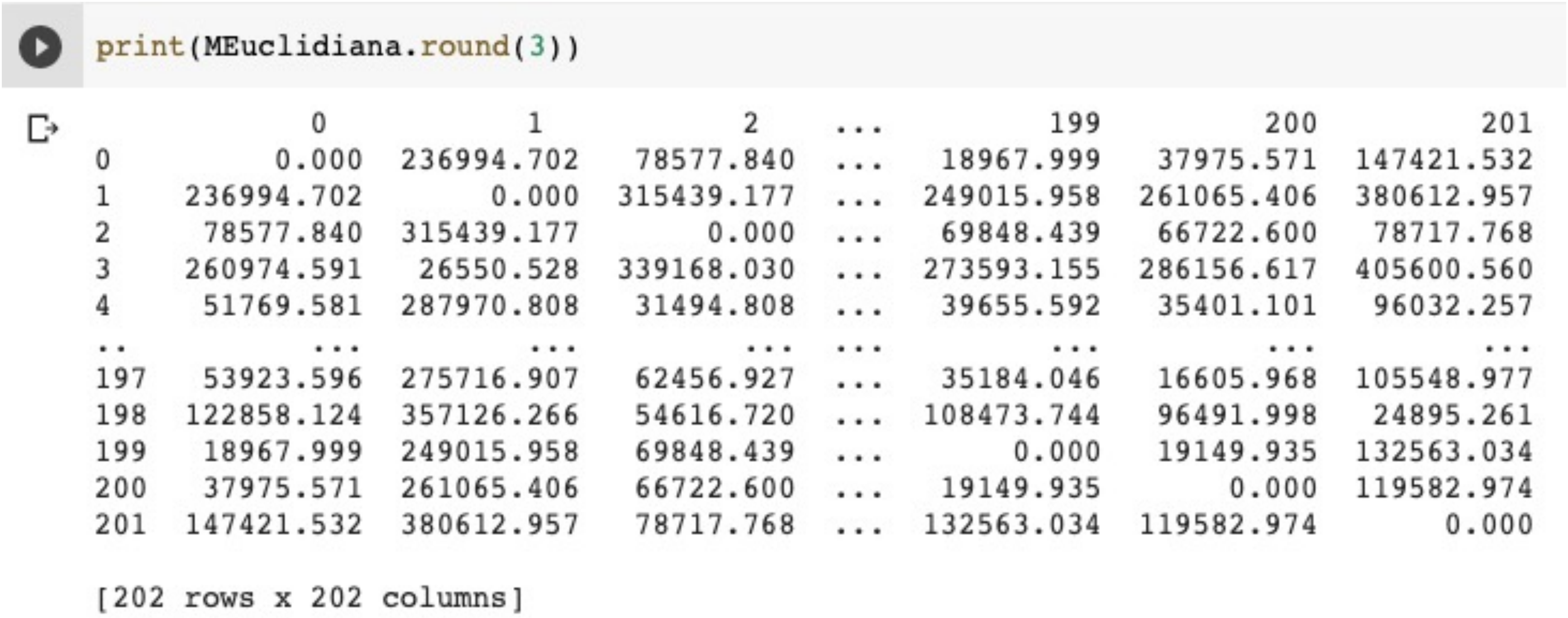
```
                 0              1  ...            200            201
0         0.000000  236994.701964  ...   37975.571227  147421.532182
1    236994.701964       0.000000  ...  261065.405879  380612.957023
2     78577.840350  315439.176808  ...   66722.600009   78717.767975
3    260974.591407   26550.527773  ...  286156.617026  405600.560294
4     51769.581416  287970.807817  ...   35401.101452   96032.256950
..             ...            ...  ...            ...            ...
197   53923.596347  275716.907131  ...   16605.967753  105548.977428
198  122858.123985  357126.266127  ...   96491.998140   24895.261437
199   18967.999420  249015.957900  ...   19149.935143  132563.033841
200   37975.571227  261065.405879  ...       0.000000  119582.974486
201  147421.532182  380612.957023  ...  119582.974486       0.000000

[202 rows x 202 columns]
```

Práctica

3. Matrices de distancias

a) Euclidiana



```
print(MEuclidiana.round(3))
```

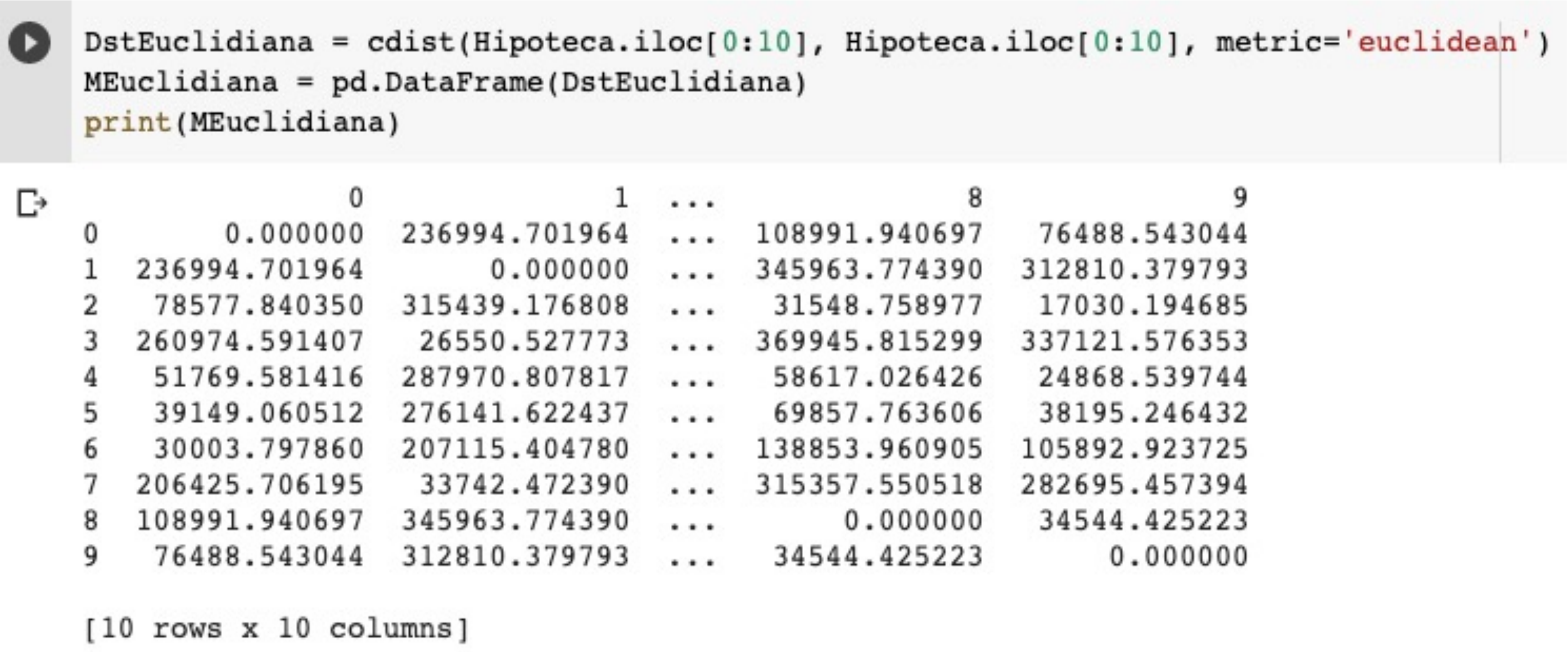
```
              0           1           2  ...         199         200         201
0         0.000  236994.702   78577.840  ...   18967.999   37975.571  147421.532
1    236994.702       0.000  315439.177  ...  249015.958  261065.406  380612.957
2     78577.840  315439.177       0.000  ...   69848.439   66722.600   78717.768
3    260974.591   26550.528  339168.030  ...  273593.155  286156.617  405600.560
4     51769.581  287970.808   31494.808  ...   39655.592   35401.101   96032.257
..          ...         ...         ...  ...         ...         ...         ...
197   53923.596  275716.907   62456.927  ...   35184.046   16605.968  105548.977
198  122858.124  357126.266   54616.720  ...  108473.744   96491.998   24895.261
199   18967.999  249015.958   69848.439  ...       0.000   19149.935  132563.034
200   37975.571  261065.406   66722.600  ...   19149.935       0.000  119582.974
201  147421.532  380612.957   78717.768  ...  132563.034  119582.974       0.000

[202 rows x 202 columns]
```

Práctica

3. Matrices de distancias

a) Euclidiana



```
DstEuclidiana = cdist(Hipoteca.iloc[0:10], Hipoteca.iloc[0:10], metric='euclidean')
MEuclidiana = pd.DataFrame(DstEuclidiana)
print(MEuclidiana)
```

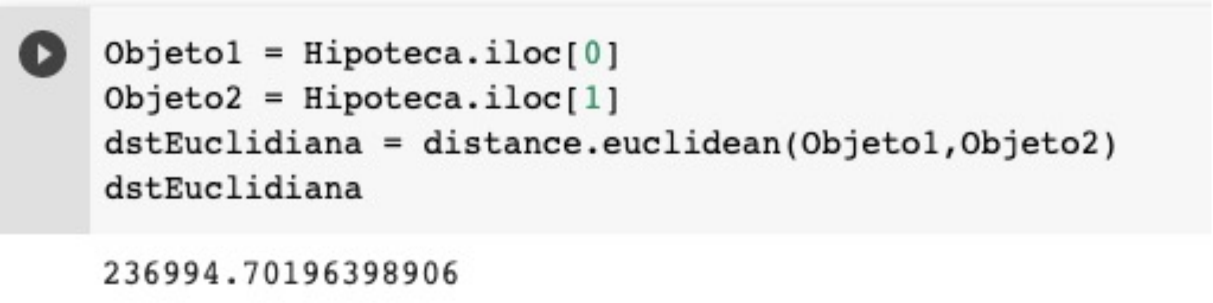
```
               0              1  ...              8              9
0       0.000000  236994.701964  ...  108991.940697   76488.543044
1  236994.701964       0.000000  ...  345963.774390  312810.379793
2   78577.840350  315439.176808  ...   31548.758977   17030.194685
3  260974.591407   26550.527773  ...  369945.815299  337121.576353
4   51769.581416  287970.807817  ...   58617.026426   24868.539744
5   39149.060512  276141.622437  ...   69857.763606   38195.246432
6   30003.797860  207115.404780  ...  138853.960905  105892.923725
7  206425.706195   33742.472390  ...  315357.550518  282695.457394
8  108991.940697  345963.774390  ...       0.000000   34544.425223
9   76488.543044  312810.379793  ...   34544.425223       0.000000

[10 rows x 10 columns]
```

Práctica

3. Matrices de distancias

a) Euclidiana (entre dos objetos)



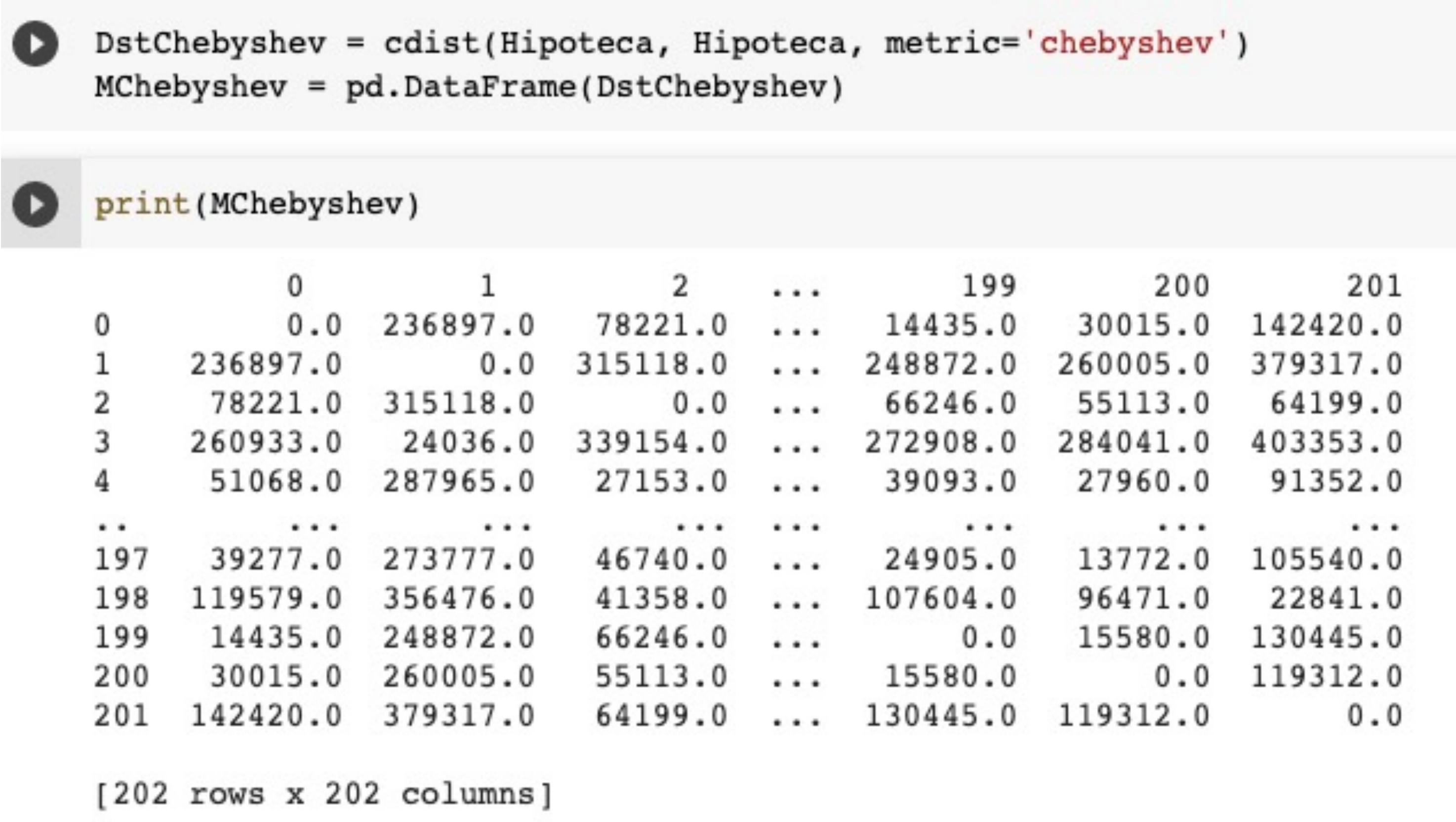
```
Objeto1 = Hipoteca.iloc[0]
Objeto2 = Hipoteca.iloc[1]
dstEuclidiana = distance.euclidean(Objeto1,Objeto2)
dstEuclidiana
```

```
236994.70196398906
```

Práctica

3. Matrices de distancias

b) Chebyshev



```
DstChebyshev = cdist(Hipoteca, Hipoteca, metric='chebyshev')
MChebyshev = pd.DataFrame(DstChebyshev)
```

```
print(MChebyshev)
```

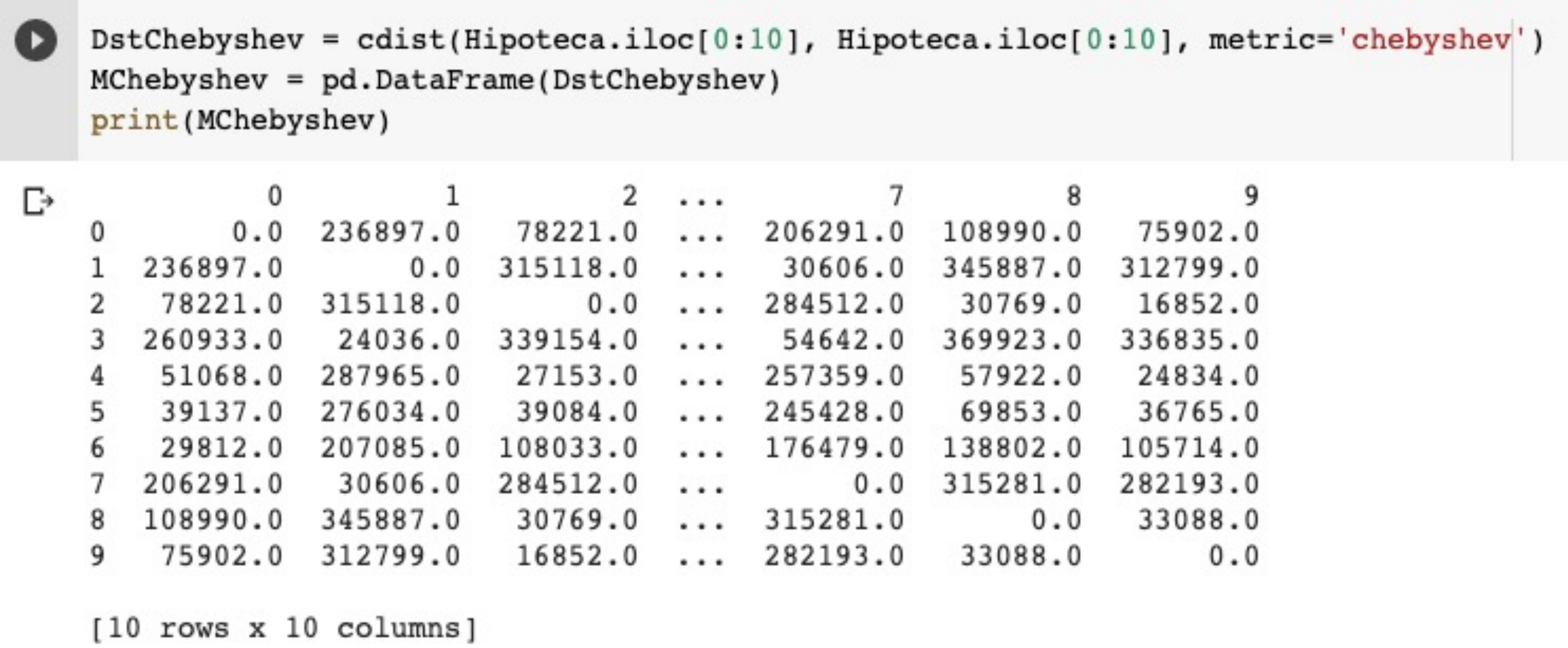
```
            0         1         2    ...       199       200       201
0         0.0  236897.0   78221.0    ...   14435.0   30015.0  142420.0
1    236897.0       0.0  315118.0    ...  248872.0  260005.0  379317.0
2     78221.0  315118.0       0.0    ...   66246.0   55113.0   64199.0
3    260933.0   24036.0  339154.0    ...  272908.0  284041.0  403353.0
4     51068.0  287965.0   27153.0    ...   39093.0   27960.0   91352.0
..        ...       ...       ...    ...       ...       ...       ...
197   39277.0  273777.0   46740.0    ...   24905.0   13772.0  105540.0
198  119579.0  356476.0   41358.0    ...  107604.0   96471.0   22841.0
199   14435.0  248872.0   66246.0    ...       0.0   15580.0  130445.0
200   30015.0  260005.0   55113.0    ...   15580.0       0.0  119312.0
201  142420.0  379317.0   64199.0    ...  130445.0  119312.0       0.0

[202 rows x 202 columns]
```

Práctica

3. Matrices de distancias

b) Chebyshev



```
DstChebyshev = cdist(Hipoteca.iloc[0:10], Hipoteca.iloc[0:10], metric='chebyshev')
MChebyshev = pd.DataFrame(DstChebyshev)
print(MChebyshev)
```

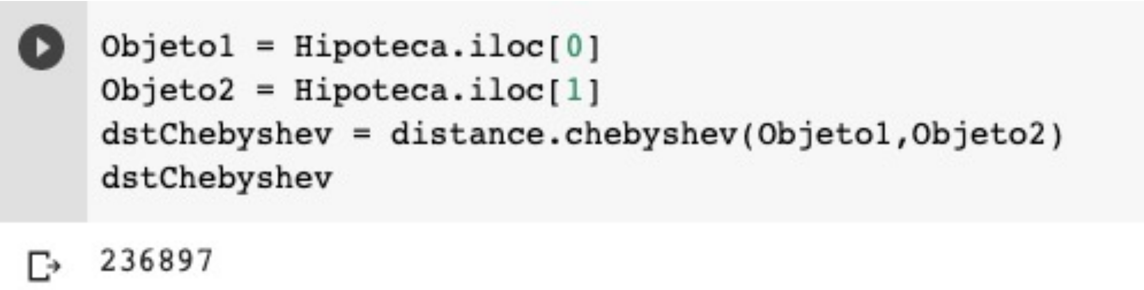
```
          0         1         2  ...         7         8         9
0       0.0  236897.0   78221.0  ...  206291.0  108990.0   75902.0
1  236897.0       0.0  315118.0  ...   30606.0  345887.0  312799.0
2   78221.0  315118.0       0.0  ...  284512.0   30769.0   16852.0
3  260933.0   24036.0  339154.0  ...   54642.0  369923.0  336835.0
4   51068.0  287965.0   27153.0  ...  257359.0   57922.0   24834.0
5   39137.0  276034.0   39084.0  ...  245428.0   69853.0   36765.0
6   29812.0  207085.0  108033.0  ...  176479.0  138802.0  105714.0
7  206291.0   30606.0  284512.0  ...       0.0  315281.0  282193.0
8  108990.0  345887.0   30769.0  ...  315281.0       0.0   33088.0
9   75902.0  312799.0   16852.0  ...  282193.0   33088.0       0.0

[10 rows x 10 columns]
```

Práctica

3. Matrices de distancias

b) Chebyshev (entre dos objetos)



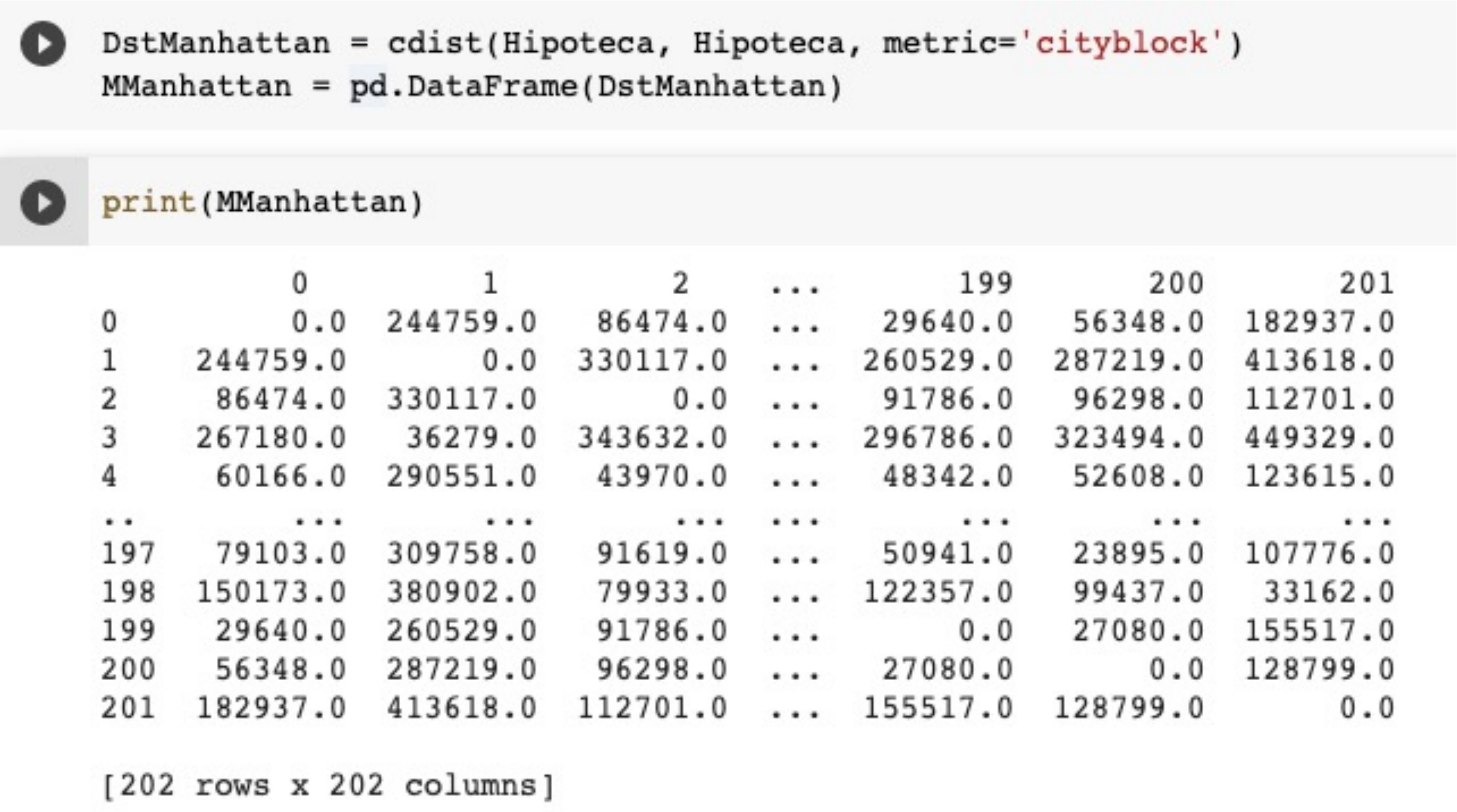
```
Objeto1 = Hipoteca.iloc[0]
Objeto2 = Hipoteca.iloc[1]
dstChebyshev = distance.chebyshev(Objeto1,Objeto2)
dstChebyshev
```

```
236897
```

Práctica

3. Matrices de distancias

c) Manhattan



```
DstManhattan = cdist(Hipoteca, Hipoteca, metric='cityblock')
MManhattan = pd.DataFrame(DstManhattan)
```

```
print(MManhattan)
```

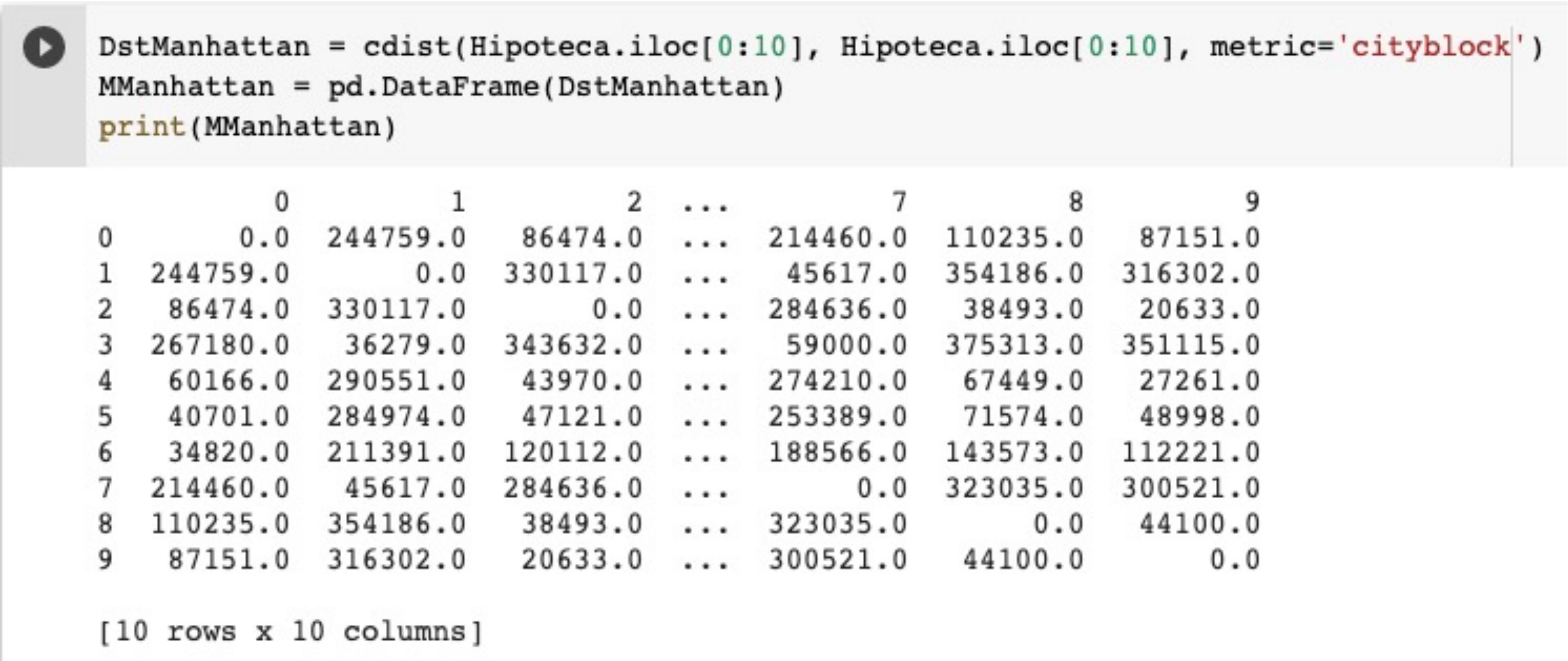
```
            0         1         2  ...       199       200       201
0         0.0  244759.0   86474.0  ...   29640.0   56348.0  182937.0
1    244759.0       0.0  330117.0  ...  260529.0  287219.0  413618.0
2     86474.0  330117.0       0.0  ...   91786.0   96298.0  112701.0
3    267180.0   36279.0  343632.0  ...  296786.0  323494.0  449329.0
4     60166.0  290551.0   43970.0  ...   48342.0   52608.0  123615.0
..        ...       ...       ...  ...       ...       ...       ...
197   79103.0  309758.0   91619.0  ...   50941.0   23895.0  107776.0
198  150173.0  380902.0   79933.0  ...  122357.0   99437.0   33162.0
199   29640.0  260529.0   91786.0  ...       0.0   27080.0  155517.0
200   56348.0  287219.0   96298.0  ...   27080.0       0.0  128799.0
201  182937.0  413618.0  112701.0  ...  155517.0  128799.0       0.0

[202 rows x 202 columns]
```

Práctica

3. Matrices de distancias

c) Manhattan



```
DstManhattan = cdist(Hipoteca.iloc[0:10], Hipoteca.iloc[0:10], metric='cityblock')
MManhattan = pd.DataFrame(DstManhattan)
print(MManhattan)
```

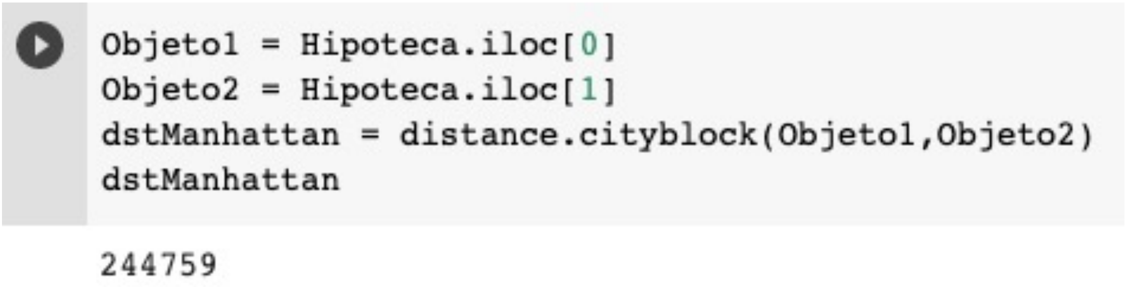
```
          0         1         2  ...         7         8         9
0       0.0  244759.0   86474.0  ...  214460.0  110235.0   87151.0
1  244759.0       0.0  330117.0  ...   45617.0  354186.0  316302.0
2   86474.0  330117.0       0.0  ...  284636.0   38493.0   20633.0
3  267180.0   36279.0  343632.0  ...   59000.0  375313.0  351115.0
4   60166.0  290551.0   43970.0  ...  274210.0   67449.0   27261.0
5   40701.0  284974.0   47121.0  ...  253389.0   71574.0   48998.0
6   34820.0  211391.0  120112.0  ...  188566.0  143573.0  112221.0
7  214460.0   45617.0  284636.0  ...       0.0  323035.0  300521.0
8  110235.0  354186.0   38493.0  ...  323035.0       0.0   44100.0
9   87151.0  316302.0   20633.0  ...  300521.0   44100.0       0.0

[10 rows x 10 columns]
```

Práctica

3. Matrices de distancias

c) Manhattan (entre dos puntos)



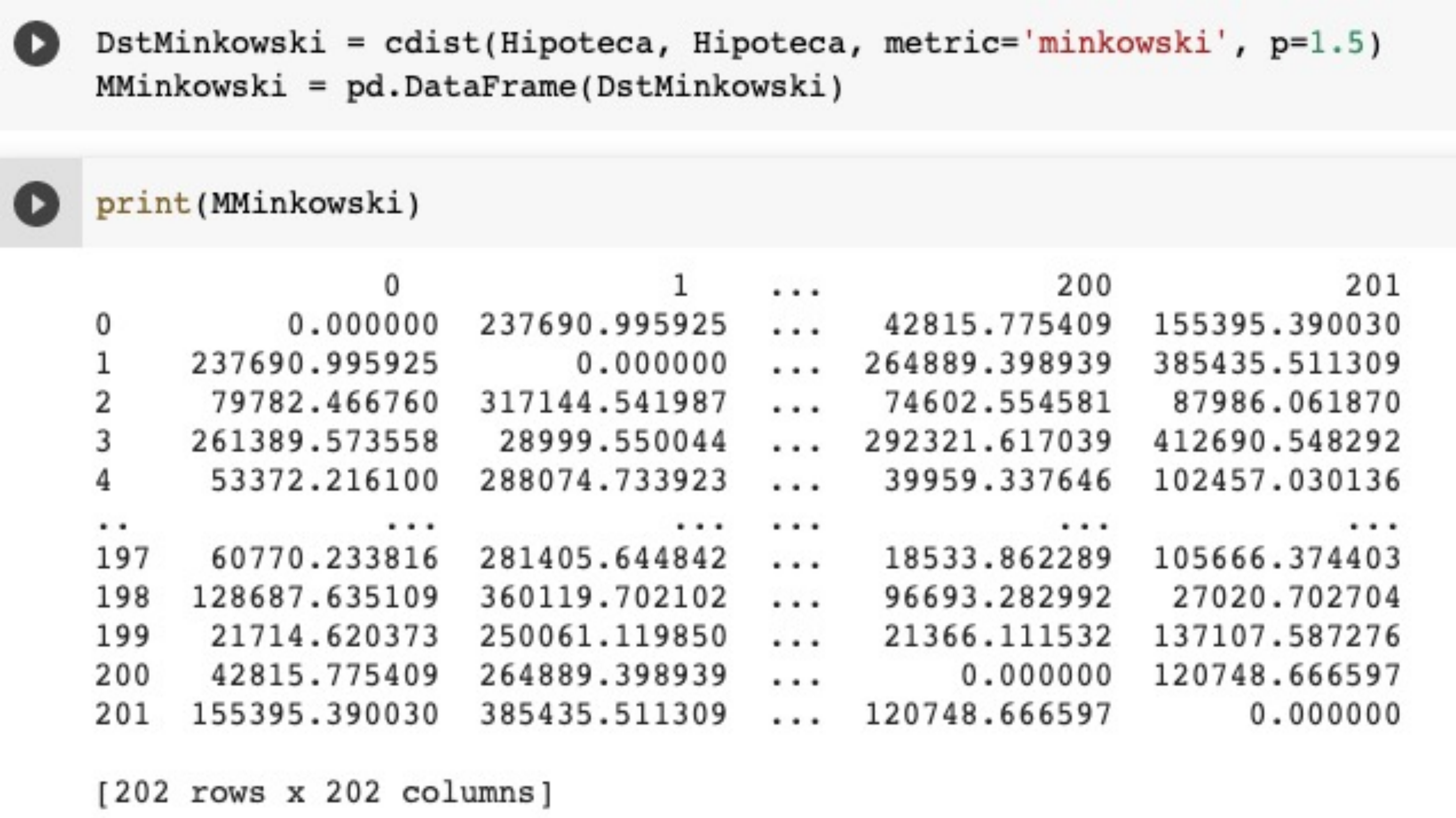
```
Objeto1 = Hipoteca.iloc[0]
Objeto2 = Hipoteca.iloc[1]
dstManhattan = distance.cityblock(Objeto1,Objeto2)
dstManhattan
```

```
244759
```

Práctica

3. Matrices de distancias

d) Minkowski



```
DstMinkowski = cdist(Hipoteca, Hipoteca, metric='minkowski', p=1.5)
MMinkowski = pd.DataFrame(DstMinkowski)
```

```
print(MMinkowski)
```

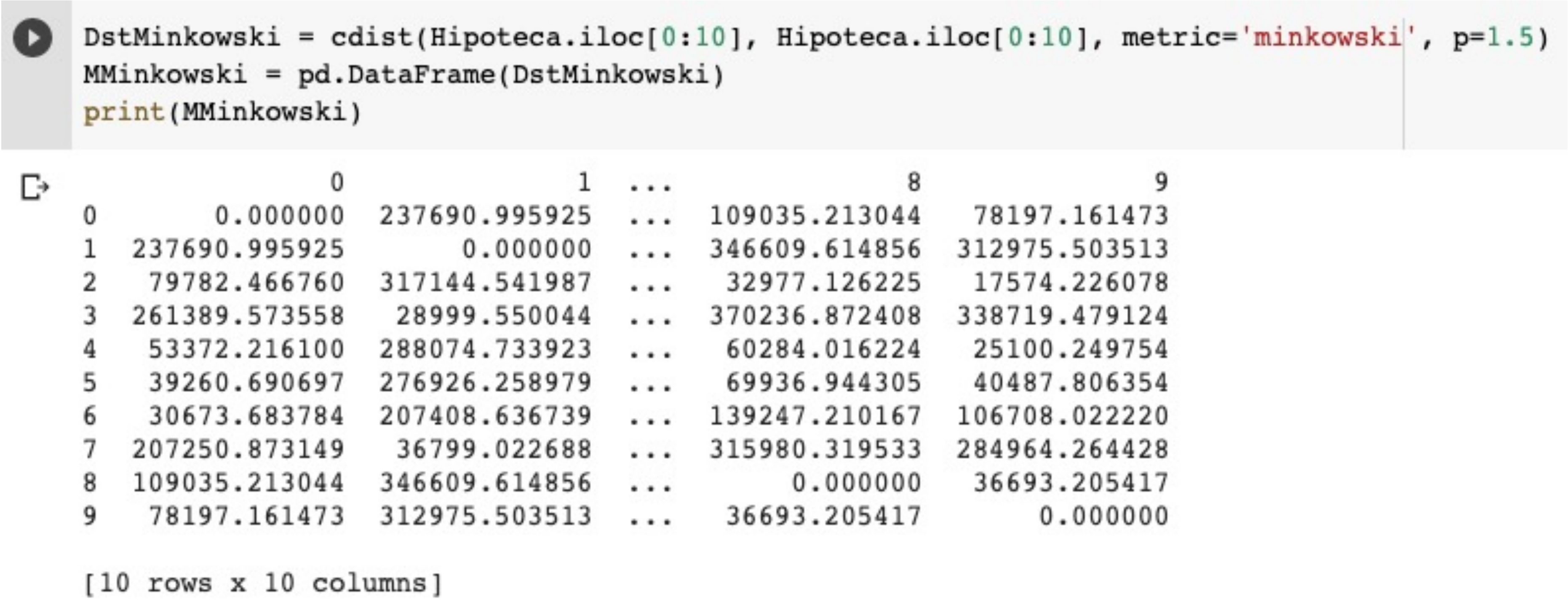
```
                0              1  ...            200            201
0        0.000000  237690.995925  ...   42815.775409  155395.390030
1   237690.995925       0.000000  ...  264889.398939  385435.511309
2    79782.466760  317144.541987  ...   74602.554581   87986.061870
3   261389.573558   28999.550044  ...  292321.617039  412690.548292
4    53372.216100  288074.733923  ...   39959.337646  102457.030136
..            ...            ...  ...            ...            ...
197  60770.233816  281405.644842  ...   18533.862289  105666.374403
198 128687.635109  360119.702102  ...   96693.282992   27020.702704
199  21714.620373  250061.119850  ...   21366.111532  137107.587276
200  42815.775409  264889.398939  ...       0.000000  120748.666597
201 155395.390030  385435.511309  ...  120748.666597       0.000000

[202 rows x 202 columns]
```

Práctica

3. Matrices de distancias

d) Minkowski



```
DstMinkowski = cdist(Hipoteca.iloc[0:10], Hipoteca.iloc[0:10], metric='minkowski', p=1.5)
MMinkowski = pd.DataFrame(DstMinkowski)
print(MMinkowski)
```

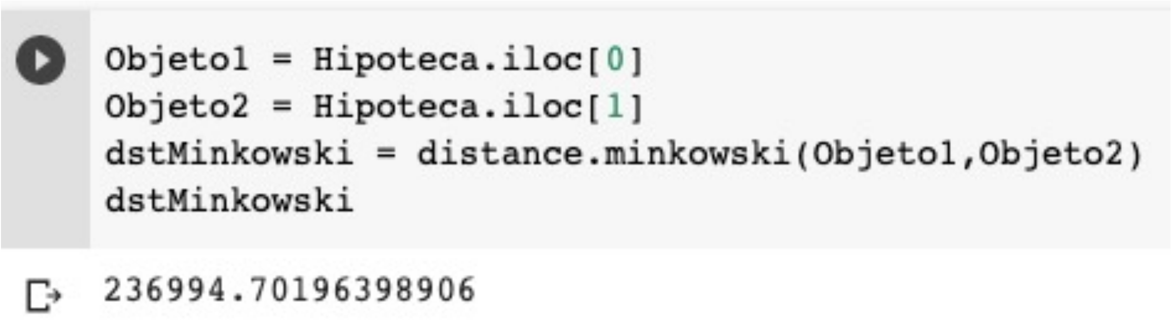
```
                0              1  ...              8              9
0        0.000000  237690.995925  ...  109035.213044   78197.161473
1   237690.995925       0.000000  ...  346609.614856  312975.503513
2    79782.466760  317144.541987  ...   32977.126225   17574.226078
3   261389.573558   28999.550044  ...  370236.872408  338719.479124
4    53372.216100  288074.733923  ...   60284.016224   25100.249754
5    39260.690697  276926.258979  ...   69936.944305   40487.806354
6    30673.683784  207408.636739  ...  139247.210167  106708.022220
7   207250.873149   36799.022688  ...  315980.319533  284964.264428
8   109035.213044  346609.614856  ...       0.000000   36693.205417
9    78197.161473  312975.503513  ...   36693.205417       0.000000

[10 rows x 10 columns]
```

Práctica

3. Matrices de distancias

d) Minkowski (entre dos puntos)

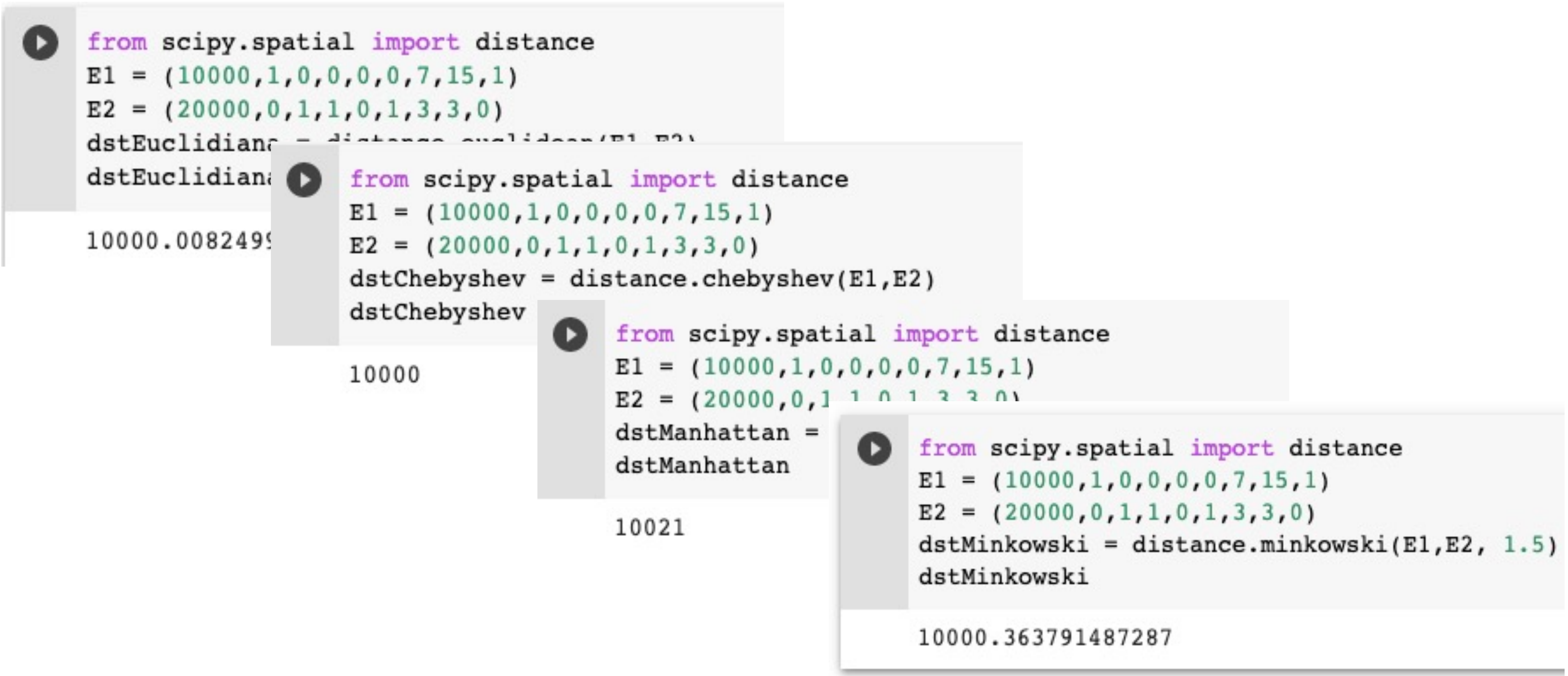


```
Objeto1 = Hipoteca.iloc[0]
Objeto2 = Hipoteca.iloc[1]
dstMinkowski = distance.minkowski(Objeto1,Objeto2)
dstMinkowski
```

```
236994.70196398906
```

Práctica

Otras mediciones



```
from scipy.spatial import distance
E1 = (10000,1,0,0,0,0,7,15,1)
E2 = (20000,0,1,1,0,1,3,3,0)
dstEuclidiana = distance.euclidean(E1,E2)
dstEuclidiana
```

```
10000.008249
```

```
from scipy.spatial import distance
E1 = (10000,1,0,0,0,0,7,15,1)
E2 = (20000,0,1,1,0,1,3,3,0)
dstChebyshev = distance.chebyshev(E1,E2)
dstChebyshev
```

```
10000
```

```
from scipy.spatial import distance
E1 = (10000,1,0,0,0,0,7,15,1)
E2 = (20000,0,1,1,0,1,3,3,0)
dstManhattan =
dstManhattan
```

```
10021
```

```
from scipy.spatial import distance
E1 = (10000,1,0,0,0,0,7,15,1)
E2 = (20000,0,1,1,0,1,3,3,0)
dstMinkowski = distance.minkowski(E1,E2, 1.5)
dstMinkowski
```

```
10000.363791487287
```