Name: Somit Jain

Reg. No. 20BDS0181

Course: Mathematical Modelling for Data Science(CSE3045 ELA)

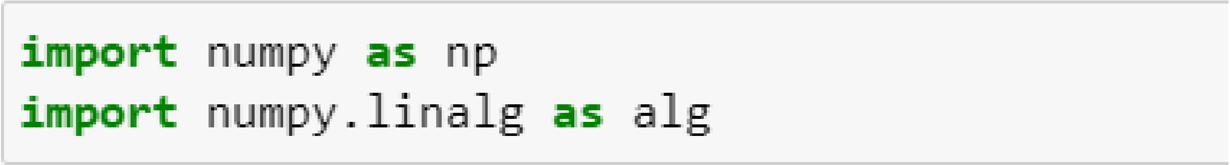
Faculty: Dr. Ilanthenral K P S K

Slot: L49+L50

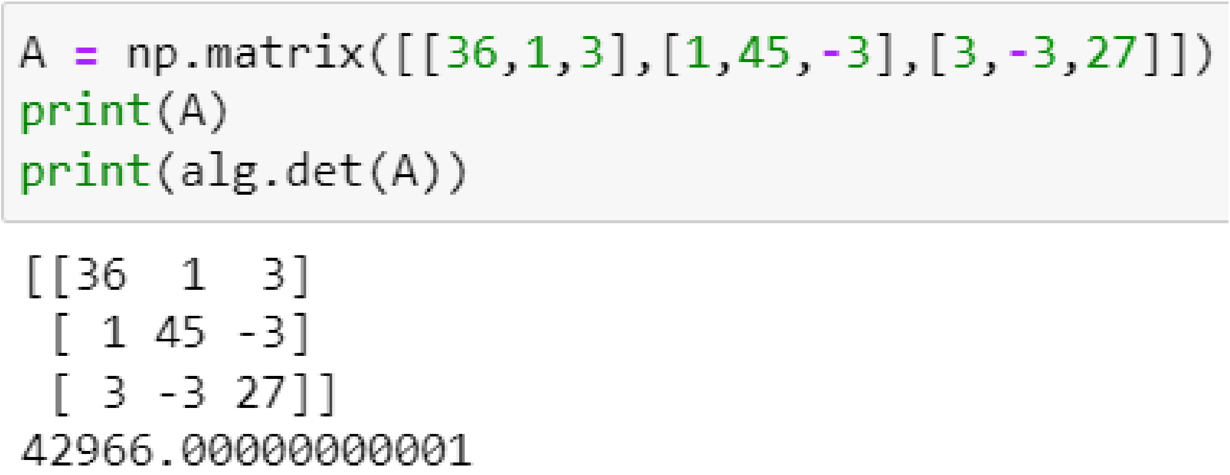
Lab-Assignment 2

Lab 3: Eigen Decomposition and Cholesky Decomposition

1. *Perform Eigen and Cholesky Decomposition using Python NumPy package linalg, explore all the possible routines along with different possible scenarios.*

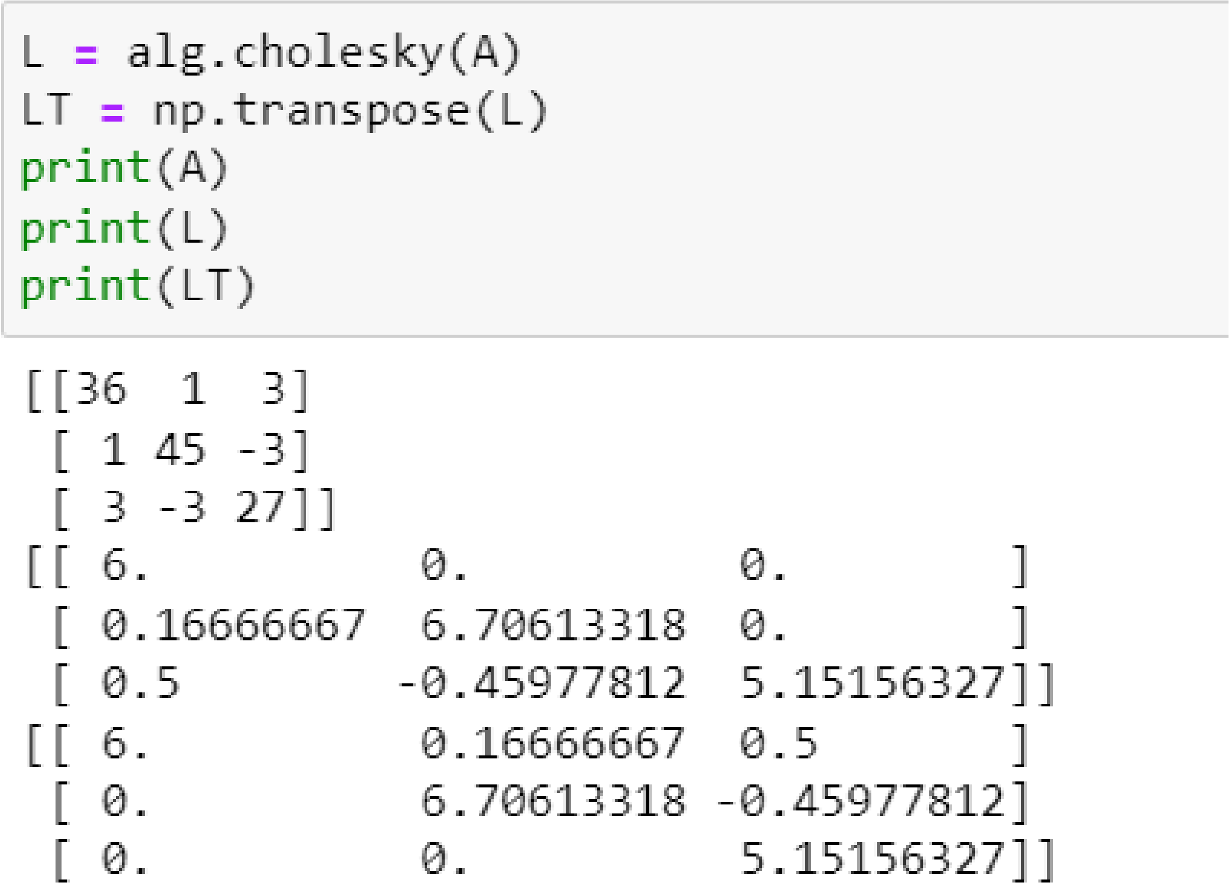


```
import numpy as np
import numpy.linalg as alg
```



```
A = np.matrix([[36,1,3],[1,45,-3],[3,-3,27]])
print(A)
print(alg.det(A))
```

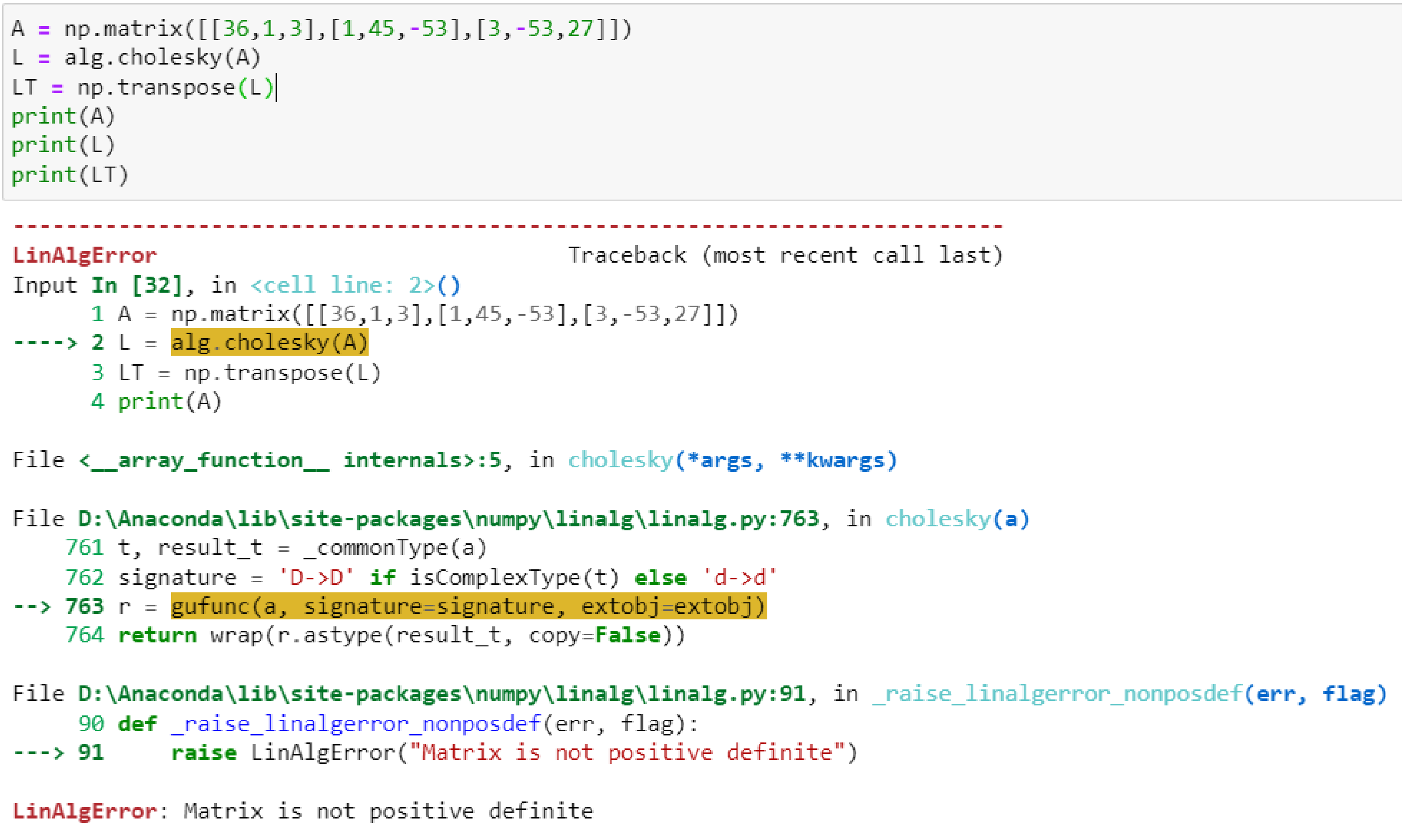
```
[[36  1  3]
 [ 1 45 -3]
 [ 3 -3 27]]
42966.00000000001
```



```
L = alg.cholesky(A)
LT = np.transpose(L)
print(A)
print(L)
print(LT)
```

```
[[36  1  3]
 [ 1 45 -3]
 [ 3 -3 27]]
[[ 6.          0.          0.        ]
 [ 0.16666667  6.70613318  0.        ]
 [ 0.5        -0.45977812  5.15156327]]
[[ 6.          0.16666667  0.5       ]
 [ 0.          6.70613318 -0.45977812]
 [ 0.          0.          5.15156327]]
```

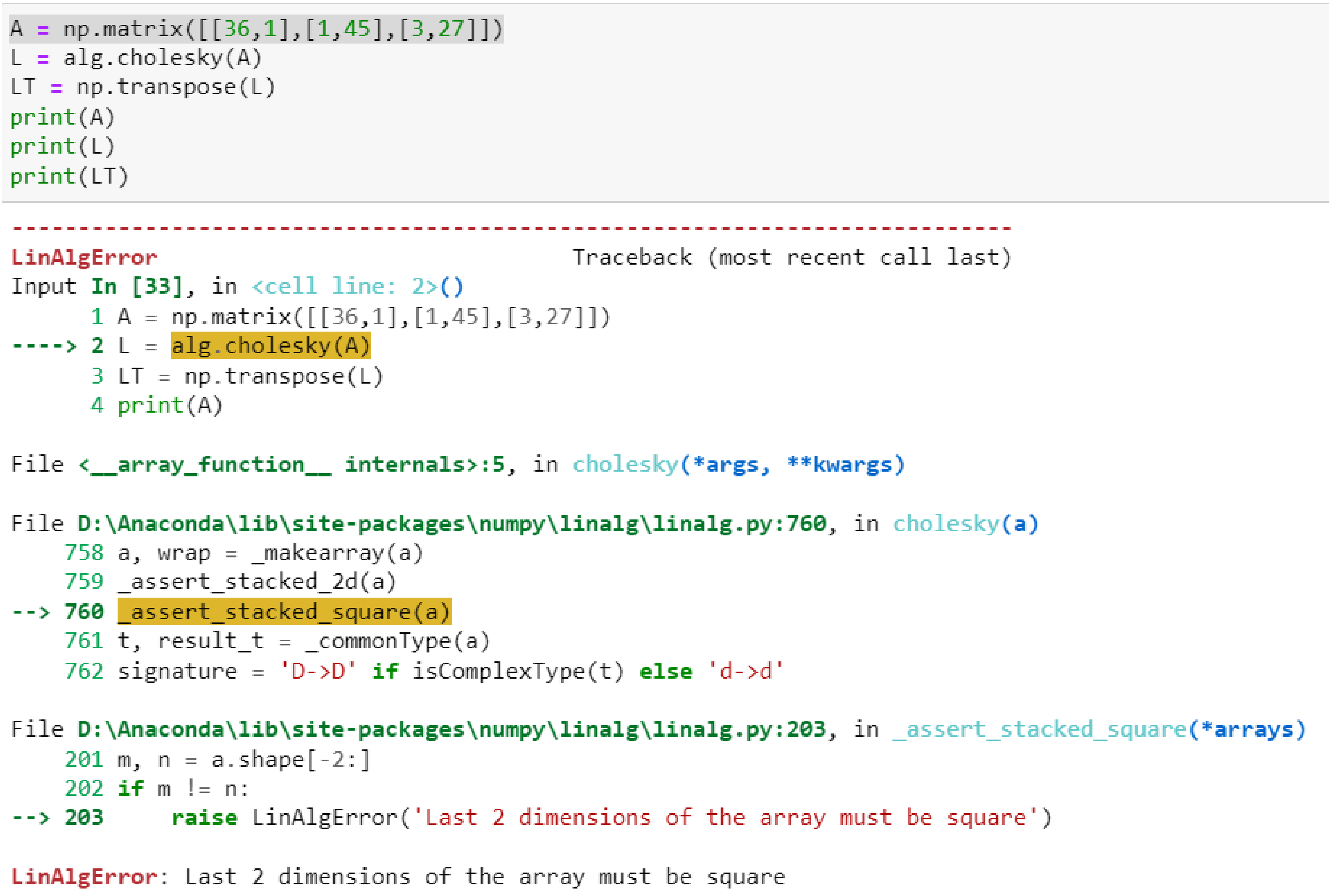
When matrix is not positive definite:



```
A = np.matrix([[36,1,3],[1,45,-53],[3,-53,27]])
L = alg.cholesky(A)
LT = np.transpose(L)
print(A)
print(L)
print(LT)
```

```
---------------------------------------------------------------------------
LinAlgError                               Traceback (most recent call last)
Input In [32], in <cell line: 2>()
      1 A = np.matrix([[36,1,3],[1,45,-53],[3,-53,27]])
----> 2 L = alg.cholesky(A)
      3 LT = np.transpose(L)
      4 print(A)

File <__array_function__ internals>:5, in cholesky(*args, **kwargs)

File D:\Anaconda\lib\site-packages\numpy\linalg\linalg.py:763, in cholesky(a)
    761 t, result_t = _commonType(a)
    762 signature = 'D->D' if isComplexType(t) else 'd->d'
--> 763 r = gufunc(a, signature=signature, extobj=extobj)
    764 return wrap(r.astype(result_t, copy=False))

File D:\Anaconda\lib\site-packages\numpy\linalg\linalg.py:91, in _raise_linalgerror_nonposdef(err, flag)
     90 def _raise_linalgerror_nonposdef(err, flag):
---> 91     raise LinAlgError("Matrix is not positive definite")

LinAlgError: Matrix is not positive definite
```

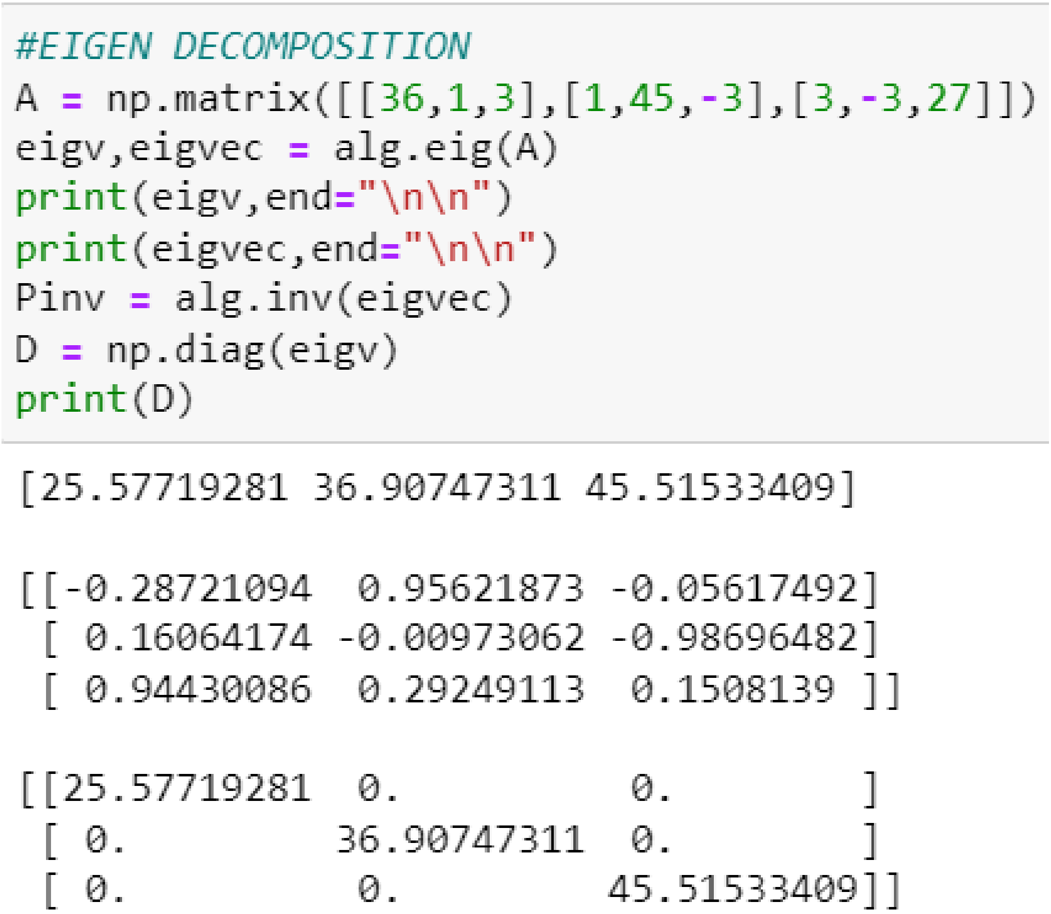
When matrix is not square matrix:



```
A = np.matrix([[36,1],[1,45],[3,27]])
L = alg.cholesky(A)
LT = np.transpose(L)
print(A)
print(L)
print(LT)
```

```
---------------------------------------------------------------------------
LinAlgError                               Traceback (most recent call last)
Input In [33], in <cell line: 2>()
      1 A = np.matrix([[36,1],[1,45],[3,27]])
----> 2 L = alg.cholesky(A)
      3 LT = np.transpose(L)
      4 print(A)

File <__array_function__ internals>:5, in cholesky(*args, **kwargs)

File D:\Anaconda\lib\site-packages\numpy\linalg\linalg.py:760, in cholesky(a)
    758 a, wrap = _makearray(a)
    759 _assert_stacked_2d(a)
--> 760 _assert_stacked_square(a)
    761 t, result_t = _commonType(a)
    762 signature = 'D->D' if isComplexType(t) else 'd->d'

File D:\Anaconda\lib\site-packages\numpy\linalg\linalg.py:203, in _assert_stacked_square(*arrays)
    201 m, n = a.shape[-2:]
    202 if m != n:
--> 203     raise LinAlgError('Last 2 dimensions of the array must be square')

LinAlgError: Last 2 dimensions of the array must be square
```

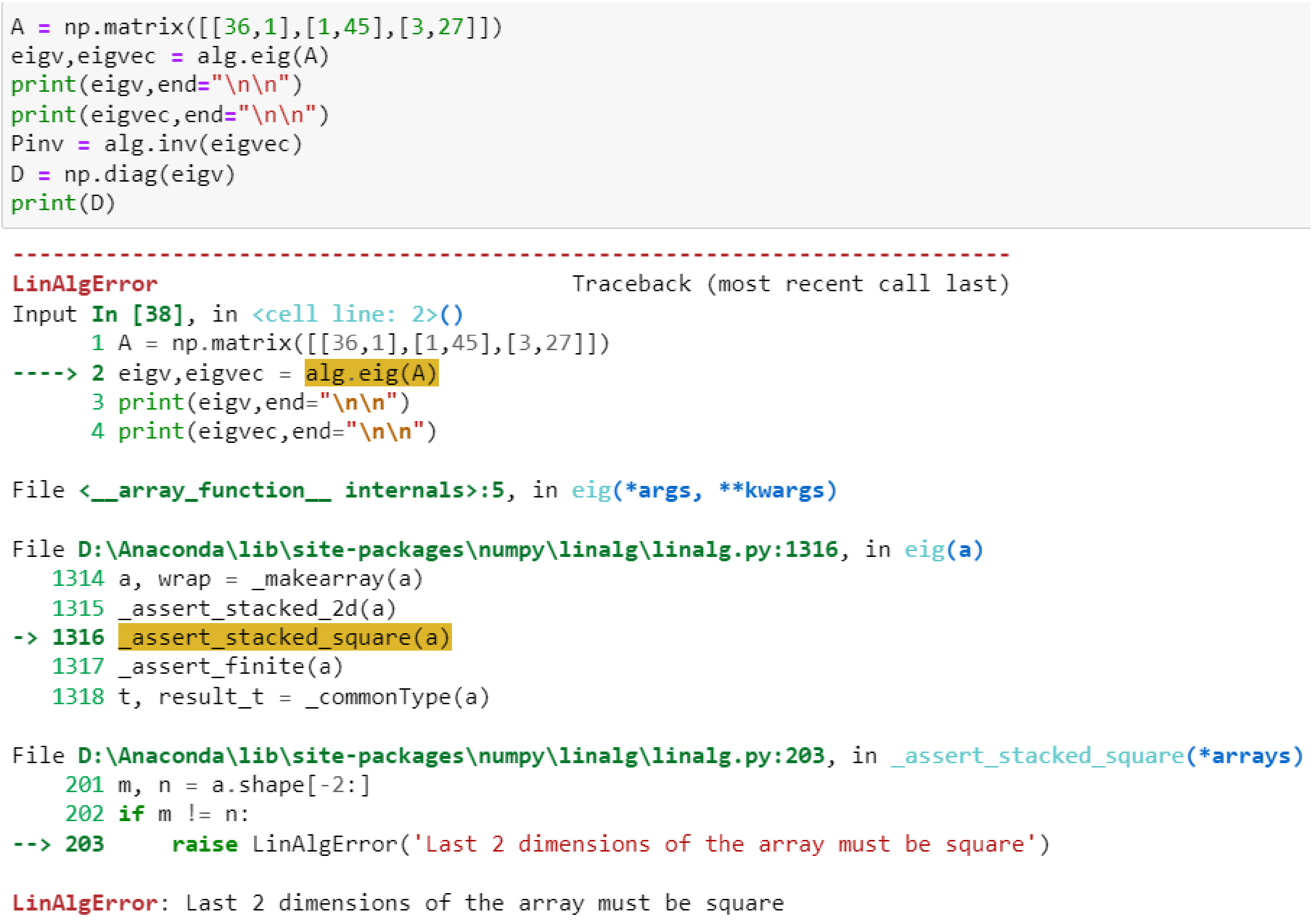
EIGEN VALUES:



```
#EIGEN DECOMPOSITION
A = np.matrix([[36,1,3],[1,45,-3],[3,-3,27]])
eigv,eigvec = alg.eig(A)
print(eigv,end="\n\n")
print(eigvec,end="\n\n")
Pinv = alg.inv(eigvec)
D = np.diag(eigv)
print(D)
```

```
[25.57719281 36.90747311 45.51533409]

[[-0.28721094  0.95621873 -0.05617492]
 [ 0.16064174 -0.00973062 -0.98696482]
 [ 0.94430086  0.29249113  0.1508139 ]]

[[25.57719281  0.          0.        ]
 [ 0.         36.90747311  0.        ]
 [ 0.          0.         45.51533409]]
```

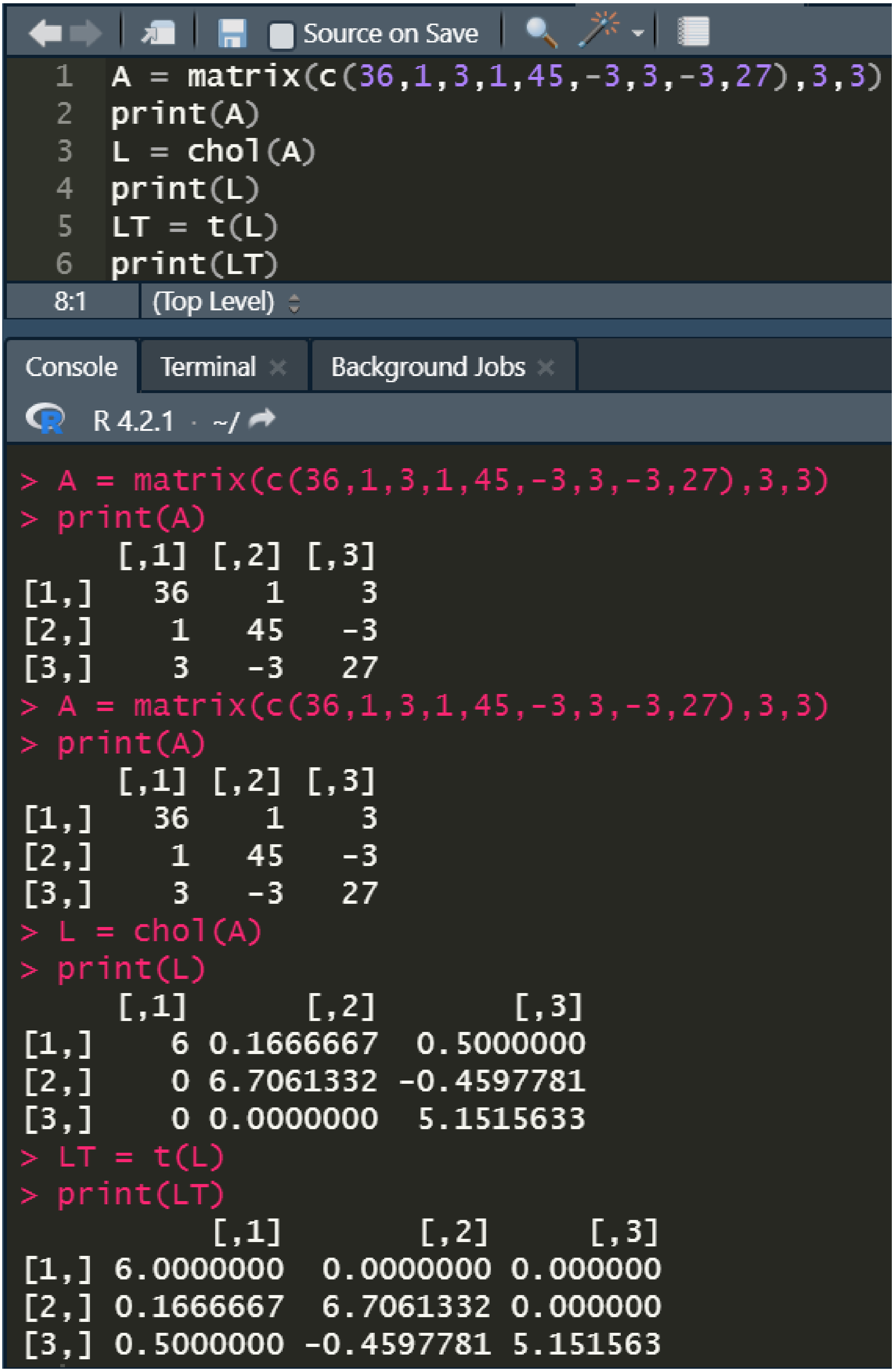
When matrix is not square matrix:



```
A = np.matrix([[36,1],[1,45],[3,27]])
eigv,eigvec = alg.eig(A)
print(eigv,end="\n\n")
print(eigvec,end="\n\n")
Pinv = alg.inv(eigvec)
D = np.diag(eigv)
print(D)
```

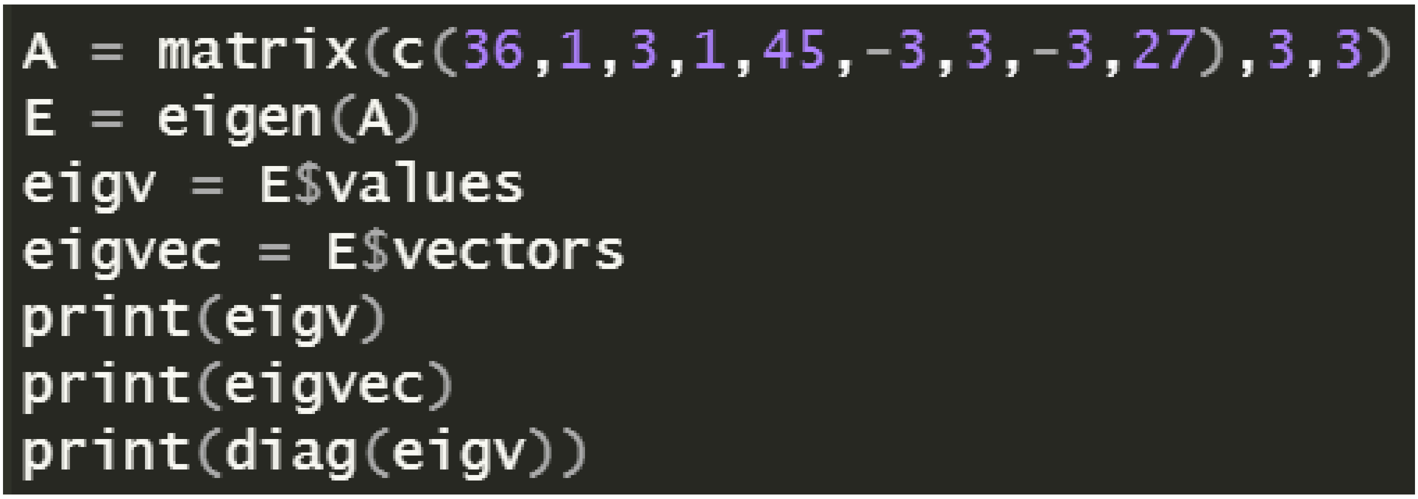
```
---------------------------------------------------------------------------
LinAlgError                               Traceback (most recent call last)
Input In [38], in <cell line: 2>()
      1 A = np.matrix([[36,1],[1,45],[3,27]])
----> 2 eigv,eigvec = alg.eig(A)
      3 print(eigv,end="\n\n")
      4 print(eigvec,end="\n\n")

File <__array_function__ internals>:5, in eig(*args, **kwargs)

File D:\Anaconda\lib\site-packages\numpy\linalg\linalg.py:1316, in eig(a)
   1314 a, wrap = _makearray(a)
   1315 _assert_stacked_2d(a)
-> 1316 _assert_stacked_square(a)
   1317 _assert_finite(a)
   1318 t, result_t = _commonType(a)

File D:\Anaconda\lib\site-packages\numpy\linalg\linalg.py:203, in _assert_stacked_square(*arrays)
    201 m, n = a.shape[-2:]
    202 if m != n:
--> 203     raise LinAlgError('Last 2 dimensions of the array must be square')

LinAlgError: Last 2 dimensions of the array must be square
```

2. *Using R, perform the Eigen Decomposition and Cholesky Decomposition.*

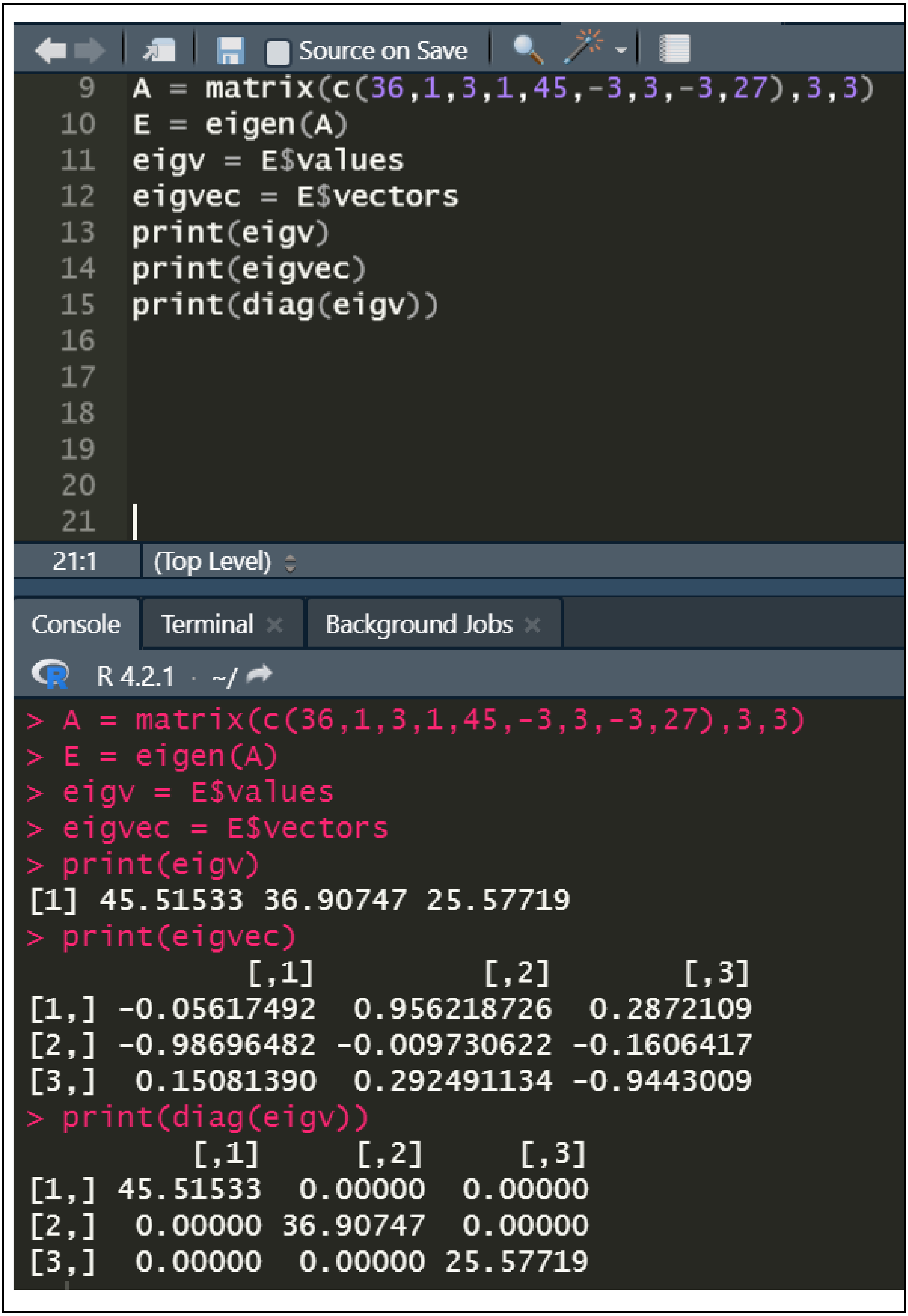


```r
A = matrix(c(36,1,3,1,45,-3,3,-3,27),3,3)
print(A)
L = chol(A)
print(L)
LT = t(L)
print(LT)
```

```
> A = matrix(c(36,1,3,1,45,-3,3,-3,27),3,3)
> print(A)
     [,1] [,2] [,3]
[1,]   36    1    3
[2,]    1   45   -3
[3,]    3   -3   27
> A = matrix(c(36,1,3,1,45,-3,3,-3,27),3,3)
> print(A)
     [,1] [,2] [,3]
[1,]   36    1    3
[2,]    1   45   -3
[3,]    3   -3   27
> L = chol(A)
> print(L)
     [,1]      [,2]       [,3]
[1,]    6 0.1666667  0.5000000
[2,]    0 6.7061332 -0.4597781
[3,]    0 0.0000000  5.1515633
> LT = t(L)
> print(LT)
          [,1]       [,2]     [,3]
[1,] 6.0000000  0.0000000 0.000000
[2,] 0.1666667  6.7061332 0.000000
[3,] 0.5000000 -0.4597781 5.151563
```



```r
A = matrix(c(36,1,3,1,45,-3,3,-3,27),3,3)
E = eigen(A)
eigv = E$values
eigvec = E$vectors
print(eigv)
print(eigvec)
print(diag(eigv))
```



```r
A = matrix(c(36,1,3,1,45,-3,3,-3,27),3,3)
E = eigen(A)
eigv = E$values
eigvec = E$vectors
print(eigv)
print(eigvec)
print(diag(eigv))
```

```
> A = matrix(c(36,1,3,1,45,-3,3,-3,27),3,3)
> E = eigen(A)
> eigv = E$values
> eigvec = E$vectors
> print(eigv)
[1] 45.51533 36.90747 25.57719
> print(eigvec)
            [,1]         [,2]       [,3]
[1,] -0.05617492  0.956218726  0.2872109
[2,] -0.98696482 -0.009730622 -0.1606417
[3,]  0.15081390  0.292491134 -0.9443009
> print(diag(eigv))
         [,1]     [,2]     [,3]
[1,] 45.51533  0.00000  0.00000
[2,]  0.00000 36.90747  0.00000
[3,]  0.00000  0.00000 25.57719
```