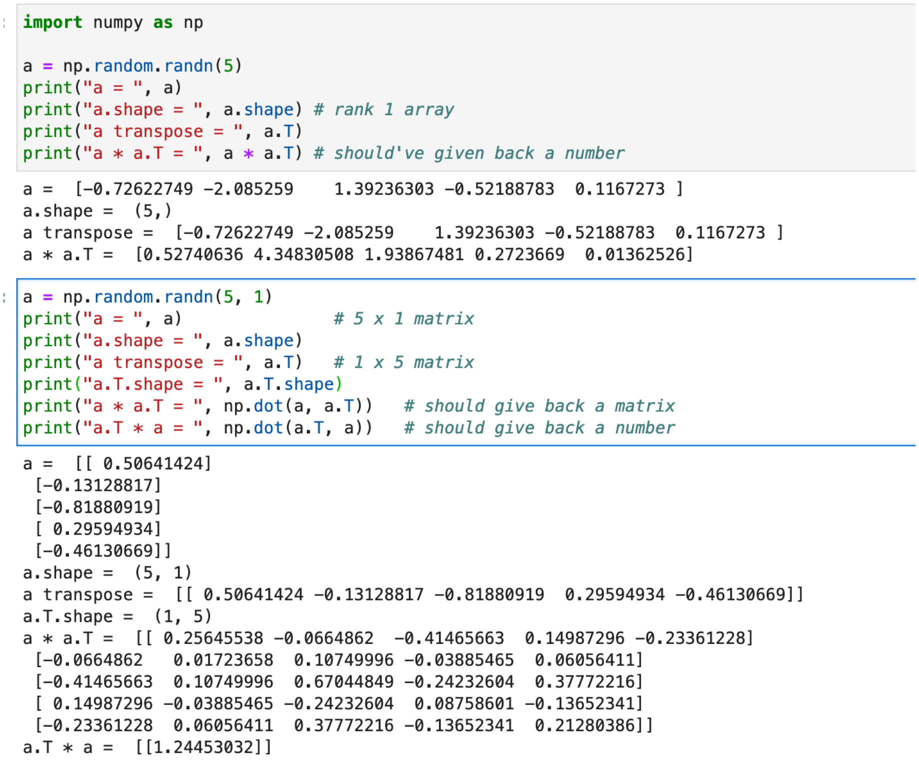
Note on Python/numpy vector



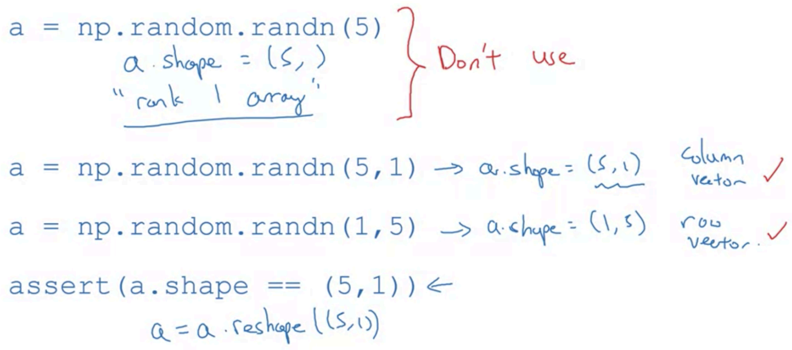
```
import numpy as np

a = np.random.randn(5)
print("a = ", a)
print("a.shape = ", a.shape) # rank 1 array
print("a transpose = ", a.T)
print("a * a.T = ", a * a.T) # should've given back a number
```

```
a =  [-0.72622749 -2.085259    1.39236303 -0.52188783  0.1167273 ]
a.shape =  (5,)
a transpose =  [-0.72622749 -2.085259    1.39236303 -0.52188783  0.1167273 ]
a * a.T =  [0.52740636 4.34830508 1.93867481 0.2723669  0.01362526]
```

```
a = np.random.randn(5, 1)
print("a = ", a)              # 5 x 1 matrix
print("a.shape = ", a.shape)
print("a transpose = ", a.T)   # 1 x 5 matrix
print("a.T.shape = ", a.T.shape)
print("a * a.T = ", np.dot(a, a.T))    # should give back a matrix
print("a.T * a = ", np.dot(a.T, a))    # should give back a number
```

```
a =  [[ 0.50641424]
 [-0.13128817]
 [-0.81880919]
 [ 0.29594934]
 [-0.46130669]]
a.shape =  (5, 1)
a transpose =  [[ 0.50641424 -0.13128817 -0.81880919  0.29594934 -0.46130669]]
a.T.shape =  (1, 5)
a * a.T =  [[ 0.25645538 -0.0664862  -0.41465663  0.14987296 -0.23361228]
 [-0.0664862   0.01723658  0.10749996 -0.03885465  0.06056411]
 [-0.41465663  0.10749996  0.67044849 -0.24232604  0.37772216]
 [ 0.14987296 -0.03885465 -0.24232604  0.08758601 -0.13652341]
 [-0.23361228  0.06056411  0.37772216 -0.13652341  0.21280386]]
a.T * a =  [[1.24453032]]
```



```
a = np.random.randn(5)
      a.shape = (5, )
      "rank 1 array"
```

Don't use

```
a = np.random.randn(5,1) → a.shape = (5,1)
```

column vector ✓

```
a = np.random.randn(1,5) → a.shape = (1,5)
```

row vector. ✓

```
assert(a.shape == (5,1)) ←
      a = a.reshape((5,1))
```

=> Don't use rank 1 vectors
Use assert as a way to ensure your code run without bugs