In [ ]:
```python
import sympy as sp
from scipy import *
from sympy import *
init_printing()
from IPython.display import display, Latex, HTML, Math
import numpy as np
import pandas as pd
```

→ V consists of eig.vecs of A^TA (right singular vectors)

→ U consists of eig.vecs of AA^T (left singular vectors)

In [ ]:
```python
# right singular vectors
A = Matrix([[2,0,0],
            [0,2,1],
            [0,1,2],
            [0,0,0]])
AtA = A.T*A
vecs1 = AtA.eigenvects() # eigenvalues/vectors from AtA
A
```

Out[ ]:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

In [ ]:
```python
vecs1
```

Out[ ]:

$$\left[\left(1,\ 1,\ \left[\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \right] \right),\ \left(4,\ 1,\ \left[\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right] \right),\ \left(9,\ 1,\ \left[\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right] \right) \right]$$

In [ ]:
```python
# We get singular values
s1 = sqrt(vecs1[2][0])
s2 = sqrt(vecs1[1][0])
s3 = sqrt(vecs1[0][0])

s1,s2,s3
```

Out[ ]: $(3,\ 2,\ 1)$

In [ ]:
```python
# We get the eigenvectors from AtA
v1 = vecs1[2][2][0].normalized()
v2 = vecs1[1][2][0].normalized()
v3 = vecs1[0][2][0].normalized()

v1,v2,v3
```

Out[ ]:

$$\left(\begin{bmatrix} 0 \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix},\ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},\ \begin{bmatrix} 0 \\ -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \right)$$

In [ ]:
```
# We derive U vectors from v1..v3
u1 = (s1**-1)*A*v1
u2 = (s2**-1)*A*v2
u3 = (s3**-1)*A*v3
U1 = u1.row_join(u2).row_join(u3)
# We need one more vector in order to have 4 eigenvectors

Ut = u1.T.col_join(u2.T).col_join(u3.T)
u4 = Ut.nullspace()[0].normalized()
U = U1.row_join(u4)
display(U)
```

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In [ ]:
```
# We set up Vt
V = v1.row_join(v2).row_join(v3)
Vt = V.T
Vt
```

Out[ ]:
$$\begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 1 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

In [ ]:
```
# Create S with same shape as A (4 x 3)
S = diag(s1, s2, s3).col_join(zeros(1,3))
S
```

Out[ ]:
$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

In [ ]:
```
display(Math('U \Sigma V^T = {}{}{}'.format(latex(U), latex(S), latex(Vt))))

display(A == U*S*Vt)
```

$$U\Sigma V^T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 1 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

True

In [ ]:
```
# Now let's try from AAt
#left singular vector
AAt = A*A.T
vecs2 = AAt.eigenvects() # eigenvalues/vectors from AAt
vecs2
```

Out[ ]:

$$\left[\left(0, \ 1, \ \left[\left[\begin{matrix}0\\0\\0\\1\end{matrix}\right] \right] \right), \ \left(1, \ 1, \ \left[\left[\begin{matrix}0\\-1\\1\\0\end{matrix}\right] \right] \right), \ \left(4, \ 1, \ \left[\left[\begin{matrix}1\\0\\0\\0\end{matrix}\right] \right] \right), \ \left(9, \ 1, \ \left[\left[\begin{matrix}0\\1\\1\\0\end{matrix}\right] \right] \right) \right]$$

In [ ]:

```
# We get singular values !! NOT TAKING 0
s1 = sqrt(vecs2[3][0])
s2 = sqrt(vecs2[2][0])
s3 = sqrt(vecs2[1][0])
```

In [ ]:

```
# We obtain all eigenvalues from AAT and construct U
u1 = vecs2[3][2][0].normalized()
u2 = vecs2[2][2][0].normalized()
u3 = vecs2[1][2][0].normalized()
u4 = vecs2[0][2][0].normalized()
U = u1.row_join(u2).row_join(u3).row_join(u4)
U
```

Out[ ]:

$$\left[\begin{matrix}0 & 1 & 0 & 0\\\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0\\\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0\\0 & 0 & 0 & 1\end{matrix}\right]$$

In [ ]:

```
# We construct v from u1..u3. Note the v's are the rows in Vt, e.g. v1^T
v1 = (s1**-1) * u1.T * A
v2 = (s2**-1) * u2.T * A
v3 = (s3**-1) * u3.T * A

Vt = v1.col_join(v2).col_join(v3)
```

In [ ]:

```
# Let is construct S
S = diag(s1, s2, s3).col_join(zeros(1,3))
S
```

Out[ ]:

$$\left[\begin{matrix}3 & 0 & 0\\0 & 2 & 0\\0 & 0 & 1\\0 & 0 & 0\end{matrix}\right]$$

In [ ]:

```
display(Math('U \Sigma V^T = {}{}{}'.format(latex(U), latex(S), latex(Vt))))

display(A == U*S*Vt)
```

$$U \Sigma V^T = \left[\begin{matrix}0 & 1 & 0 & 0\\\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0\\\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0\\0 & 0 & 0 & 1\end{matrix}\right]\left[\begin{matrix}3 & 0 & 0\\0 & 2 & 0\\0 & 0 & 1\\0 & 0 & 0\end{matrix}\right]\left[\begin{matrix}0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\\1 & 0 & 0\\0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\end{matrix}\right]$$

```
True
```