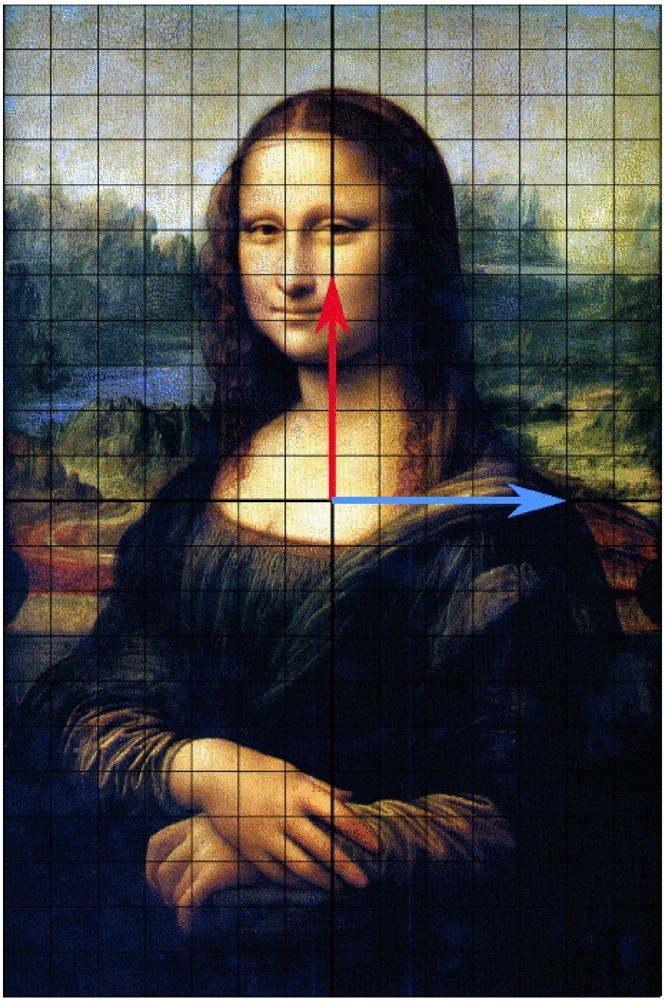
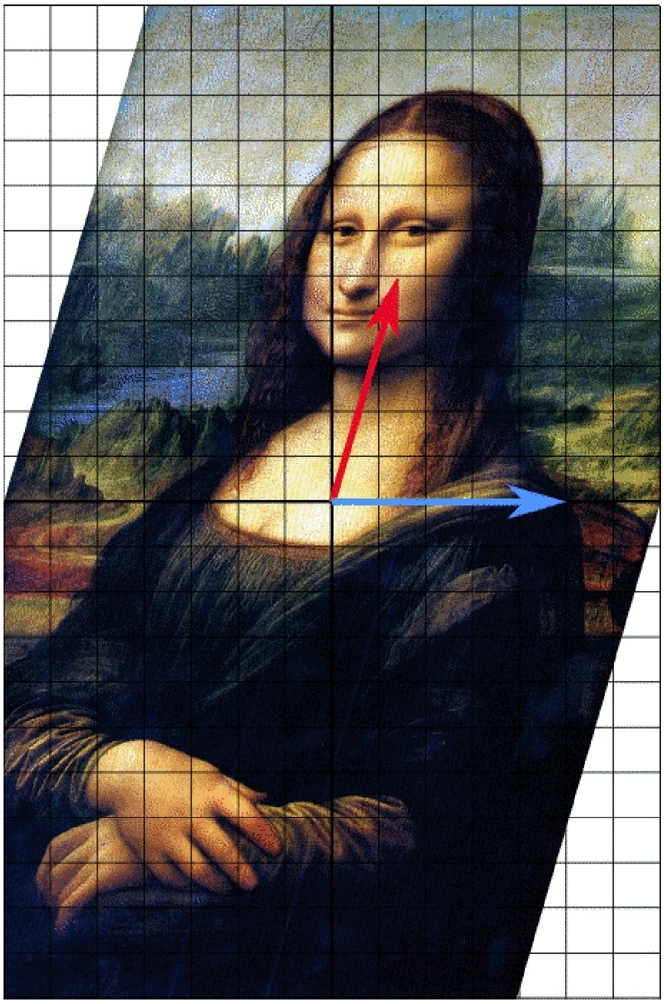




Hands-on demo of eigenvalue and eigenvector

In [43]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [44]:

```
A = np.random.rand(2, 2)
A
```

Out[44]:

```
array([[0.52709121, 0.98410474],
       [0.86879566, 0.18281998]])
```

In [45]:

```
# Compute eigenvalues (lambdas) and eigenvectors (v) of A
lambdas, v = np.linalg.eig(A)
```

In [46]:

```
v
```

Out[46]:

```
array([[ 0.7881902 , -0.66250373],
       [ 0.61543172,  0.74905861]])
```

Let's confirm that $Av = \lambda v$ for the first eigenvector:

In [47]:

```
first_eigenvector = v[:,0]
first_eigenvector
```

Out[47]:

```
array([0.7881902 , 0.61543172])
```

In [48]:

```
lambda_first = lambdas[0]
lambda_first
```

Out[48]:

```
1.2954961819004978
```

In [49]:

```
A_first_eigenvector = np.dot(A, first_eigenvector)
A_first_eigenvector
```

Out[49]:

```
array([1.0210974 , 0.79728944])
```

In [50]:

```
lambda_first * first_eigenvector
```

Out[50]:

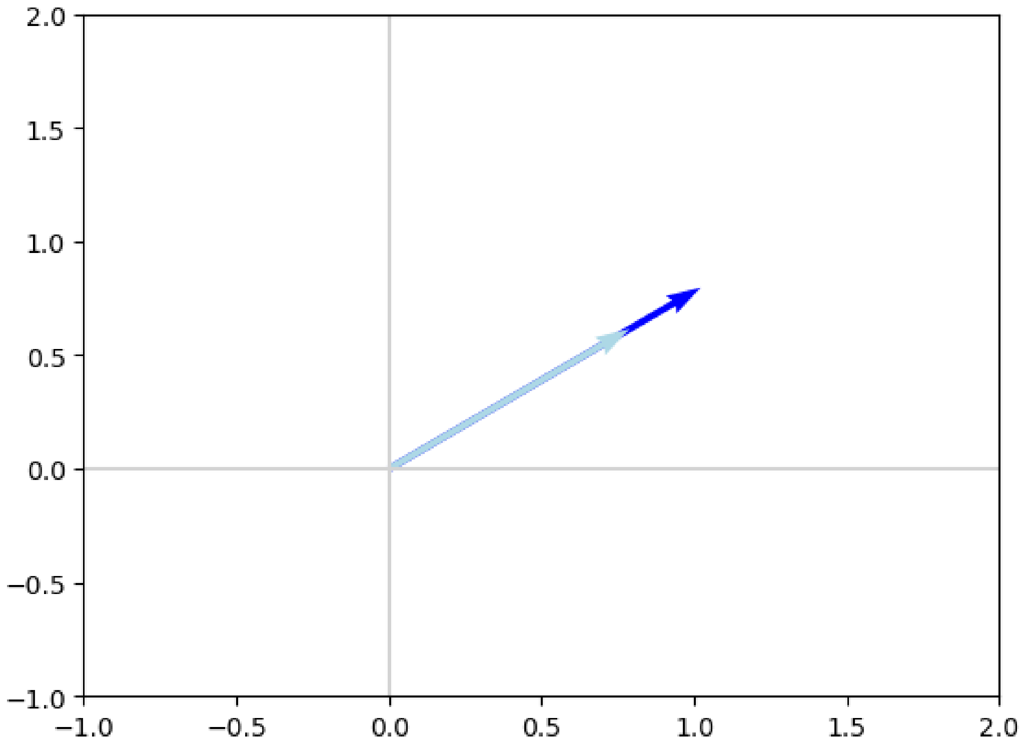
```
array([1.0210974 , 0.79728944])
```

In [51]:

```
import matplotlib.pyplot as plt
```

In [52]:

```
# plot_vector is pre-created function, you can use other plotting techniques
plot_vectors([A_first_eigenvector, first_eigenvector], ['blue', 'lightblue'])
plt.xlim(-1, 2)
_ = plt.ylim(-1, 2)
```



In [53]:

```
second_eigenvector = v[:,1]
second_eigenvector
```

Out[53]:

```
array([-0.66250373,  0.74905861])
```

In [54]:

```
lambda_second = lambdas[1]
lambda_second
```

Out[54]:

```
-0.5855849956987036
```

In [55]:

```
A_second_eigenvector = np.dot(A, second_eigenvector)
A_second_eigenvector
```

Out[55]:

```
array([ 0.38795224, -0.43863749])
```

In [56]:

```
lambda_second * second_eigenvector
```

Out[56]:

```
array([ 0.38795224, -0.43863749])
```

In [57]:

```
plot_vectors([A_first_eigenvector, first_eigenvector, A_second_eigenvector, second_eigenv
ector],
             ['blue', 'lightblue', 'green', 'lightgreen'])
plt.xlim(-1, 4)
_ = plt.ylim(-3, 2)
```