```
clear
clc
rotx(0.2)
```

```
ans = 3×3
    1.0000         0         0
         0    0.9801   -0.1987
         0    0.1987    0.9801
```

```
R = rotx(30, 'deg')
```

```
R = 3×3
    1.0000         0         0
         0    0.8660   -0.5000
         0    0.5000    0.8660
```

```
det(R)
```

```
ans = 1
```

```
inv(R)
```

```
ans = 3×3
    1.0000         0         0
         0    0.8660    0.5000
         0   -0.5000    0.8660
```

```
R' %will be same as above
```

```
ans = 3×3
    1.0000         0         0
         0    0.8660    0.5000
         0   -0.5000    0.8660
```

```
roty(0.2)
```

```
ans = 3×3
    0.9801         0    0.1987
         0    1.0000         0
   -0.1987         0    0.9801
```

```
rotz(0.3)
```

```
ans = 3×3
    0.9553   -0.2955         0
    0.2955    0.9553         0
         0         0    1.0000
```

```
trplot(ans)

rotx(pi/2)*roty(pi/2)
```

```
ans = 3×3
     0     0     1
     1     0     0
     0     1     0
```

```
roty(pi/2)*rotx(pi/2) %will be different than above
```

```
ans = 3×3
     0     1     0
     0     0    -1
    -1     0     0
```

```
eul2r(0.1, 0.2, 0.3) %ZYZ euler angles to rotation matrix
```

```
ans = 3×3
    0.9021   -0.3836    0.1977
    0.3875    0.9216    0.0198
   -0.1898    0.0587    0.9801
```

```
tr2eul(ans) %rotation matrix to euler angles
```

```
ans = 1×3
    0.1000    0.2000    0.3000
```

```
%%there are two sets of euler angles that result in the same rotation
%%matrix

rpy2r(0.1, 0.2, 0.3) %roll pitch yaw angles to rotation matrix
```

```
ans = 3×3
    0.9363   -0.2751    0.2184
    0.2896    0.9564   -0.0370
   -0.1987    0.0978    0.9752
```

```
tr2rpy(ans) %rotation matrix to roll pitch yaw angles
```

```
ans = 1×3
    0.1000    0.2000    0.3000
```

```
rpy2r(0.3, pi/2, 0.5)
```

```
ans = 3×3
         0   -0.1987    0.9801
         0    0.9801    0.1987
   -1.0000         0         0
```

```
rpy2r(0, pi/2, 0.8)
```

```
ans = 3×3
         0   -0.7174    0.6967
         0    0.6967    0.7174
   -1.0000         0         0
```

```
%%both are equal because pitch = pi/2 so now roll and
%%yaw are equivilant (gimbal lock) and the rotation = roll + yaw

R = eul2r(0.1, 0.2, 0.3)
```

```
R = 3×3
    0.9021   -0.3836    0.1977
    0.3875    0.9216    0.0198
   -0.1898    0.0587    0.9801
```

```
trplot(R)
eig(R) %eigenvalues of the rotation matrix R
```

```
ans = 3×1 complex
   0.9019 + 0.4319i
   0.9019 - 0.4319i
   1.0000 + 0.0000i
```

```
[v,e] = eig(R) %each column of v is an eigenvector
```

```
v = 3×3 complex
   0.7064 + 0.0000i   0.7064 + 0.0000i   0.0450 + 0.0000i
  -0.0143 - 0.6318i  -0.0143 + 0.6318i   0.4486 + 0.0000i
  -0.0284 + 0.3175i  -0.0284 - 0.3175i   0.8926 + 0.0000i
e = 3×3 complex
   0.9019 + 0.4319i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.9019 - 0.4319i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   1.0000 + 0.0000i
```

```
%%eigenvector is real when eigevalue is 1

[th,v] = tr2angvec(R) %th: angle of rotation needed, v: vector of rotation
```

```
th = 0.4466
v = 1×3
    0.0450    0.4486    0.8926
```

```

angvec2r(th, v) %find rotation matrix R from 'th' and 'v'
```

```
ans = 3×3
    0.9021   -0.3836    0.1977
    0.3875    0.9216    0.0198
   -0.1898    0.0587    0.9801
```

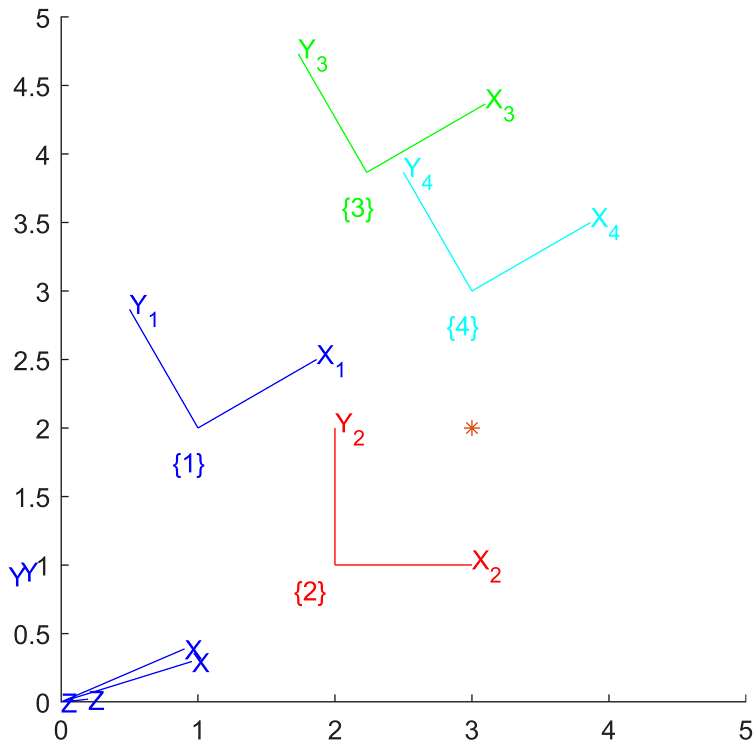
```

q = UnitQuaternion(R) % q = s <v1,v2,v3>
```

```
q = 

0.97517 < 0.0099667, 0.099335, 0.19768 >
```

```
q.plot()
```




```
inv(q)
```

```
ans =
0.97517 < -0.0099667, -0.099335, -0.19768 >
```

```
q*inv(q) %0 rotation
```

```
ans =
1 < 0, 0, 0 >
```

```
q/q %=q*inv(q)
```

```
ans =
1 < 0, 0, 0 >
```

```
q*[1 0 0]'
```

```
ans = 3×1
    0.9021
    0.3875
```

```
    -0.1898
```

```
q0 = UnitQuaternion
```

```
q0 =

1 < 0, 0, 0 >
```

```
q0.interp(q, 1) %interpolation between q0 and q.
```

```
ans =

0.97517 < 0.0099667, 0.099335, 0.19768 >
```

```
%%0 is initial q. 1 is final q. 0.5 is interpolation halfway between
%%initial q and final q
```