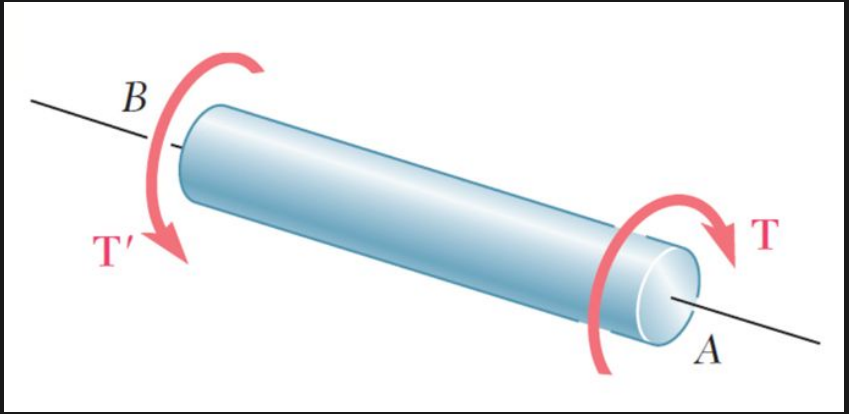
Applications of Eigenvectors

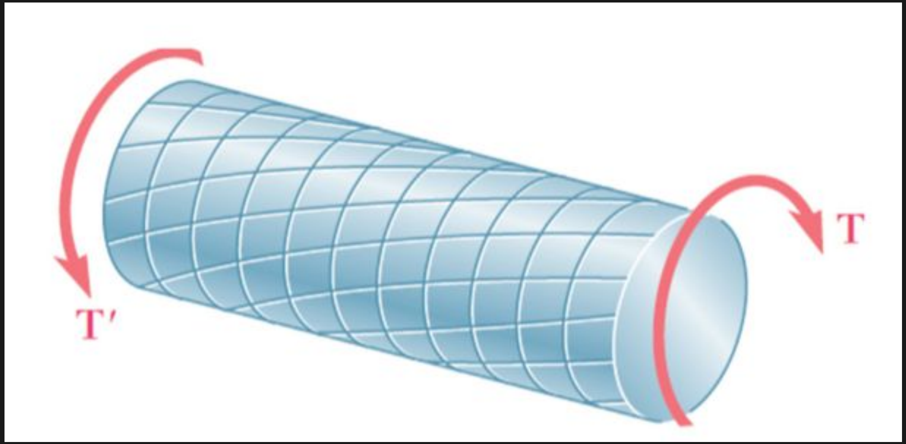
Practical Linear Algebra | Lecture 12

Mechanical Engineering



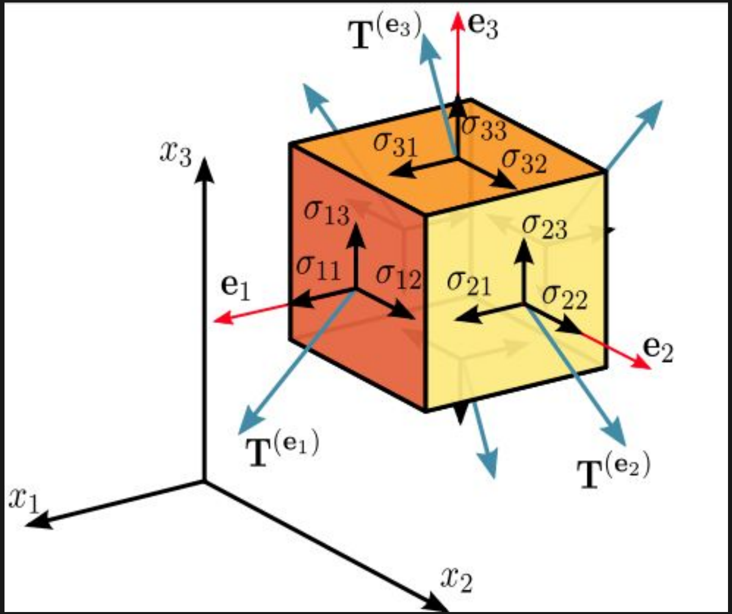
From http://me.eng.usc.ac.ir/files/1509793694994.pdf

Mechanical Engineering



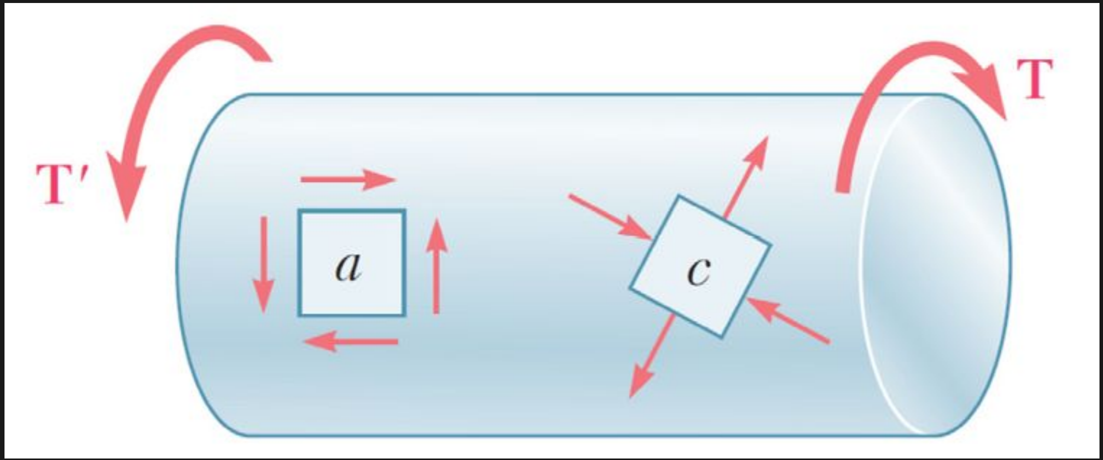
From http://me.eng.usc.ac.ir/files/1509793694994.pdf

Mechanical Engineering



From https://en.wikipedia.org/wiki/Cauchy_stress_tensor

Mechanical Engineering



From http://me.eng.usc.ac.ir/files/1509793694994.pdf

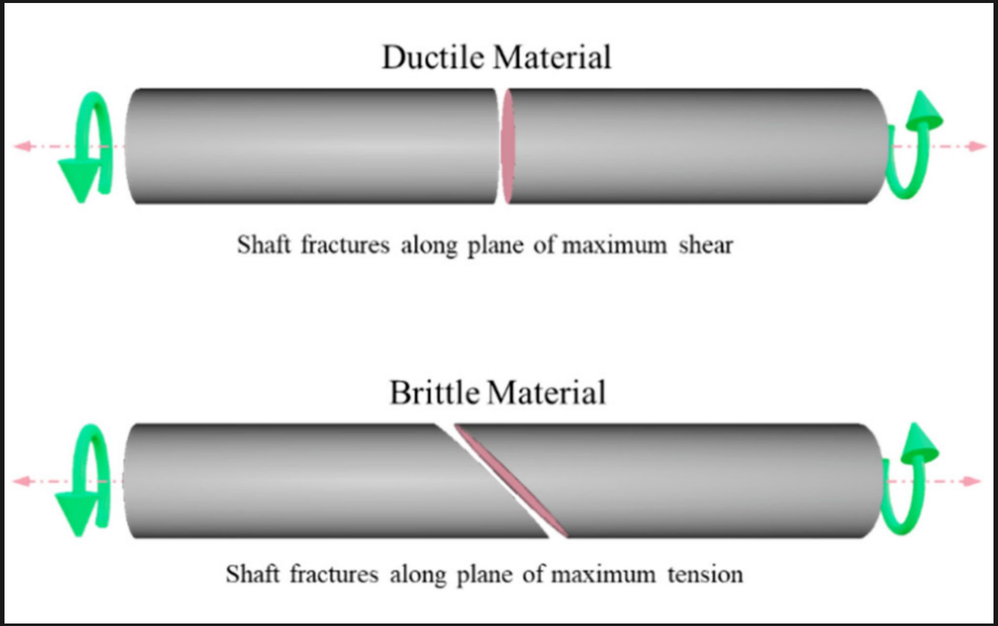
Mechanical Engineering

From https://www.youtube.com/watch?v=s4LfFwCidBc

Mechanical Engineering

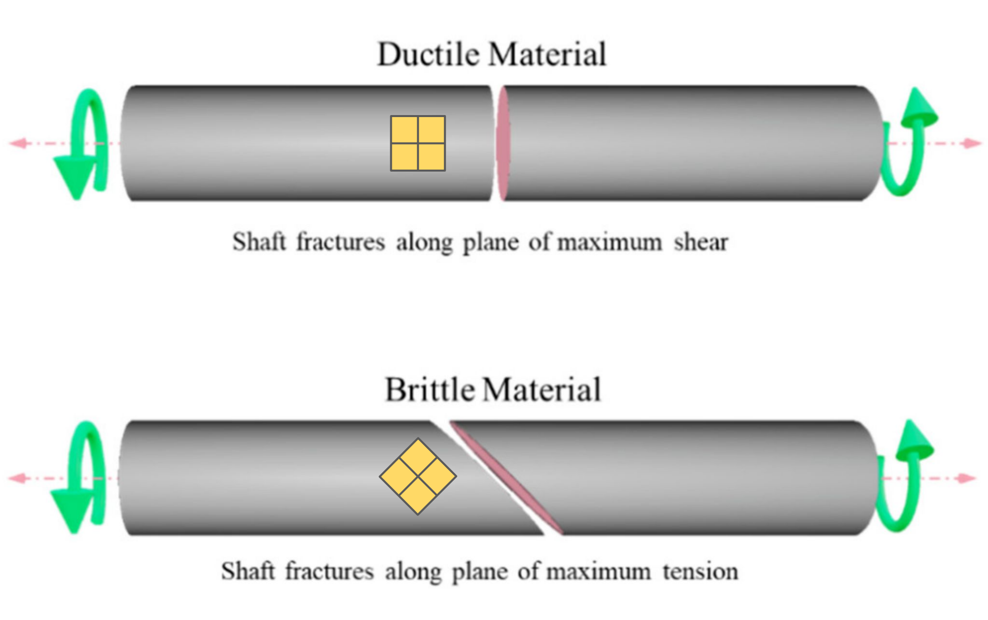
From https://www.youtube.com/watch?v=s4LfFwCidBc

Mechanical Engineering



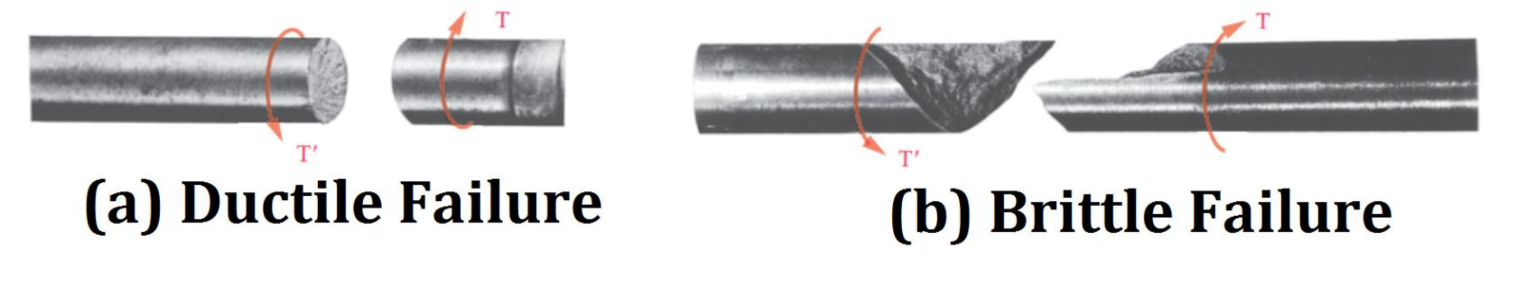
From https://www.mdpi.com/1996-1944/14/21/6521

Mechanical Engineering



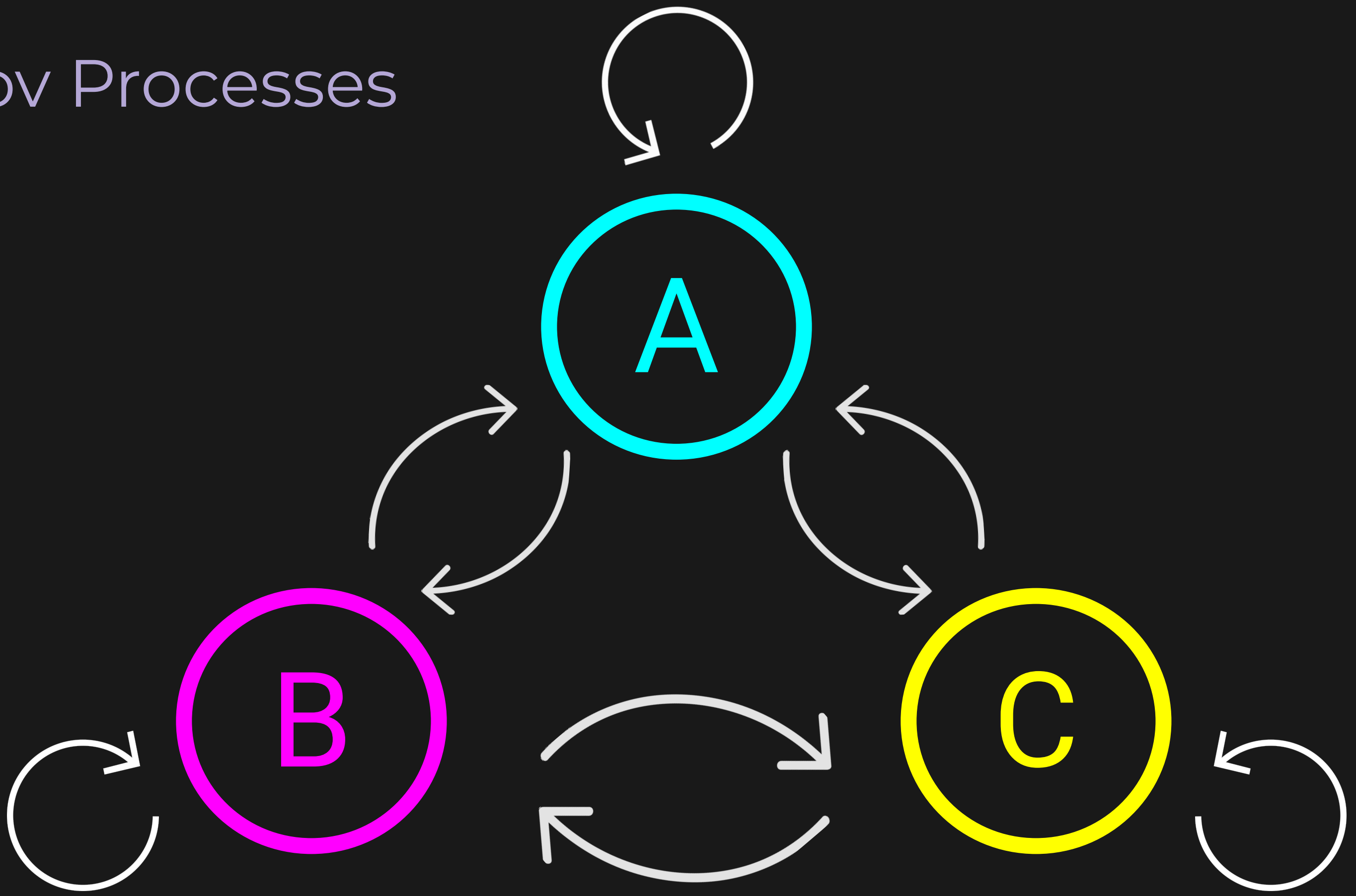
From https://www.mdpi.com/1996-1944/14/21/6521

Mechanical Engineering

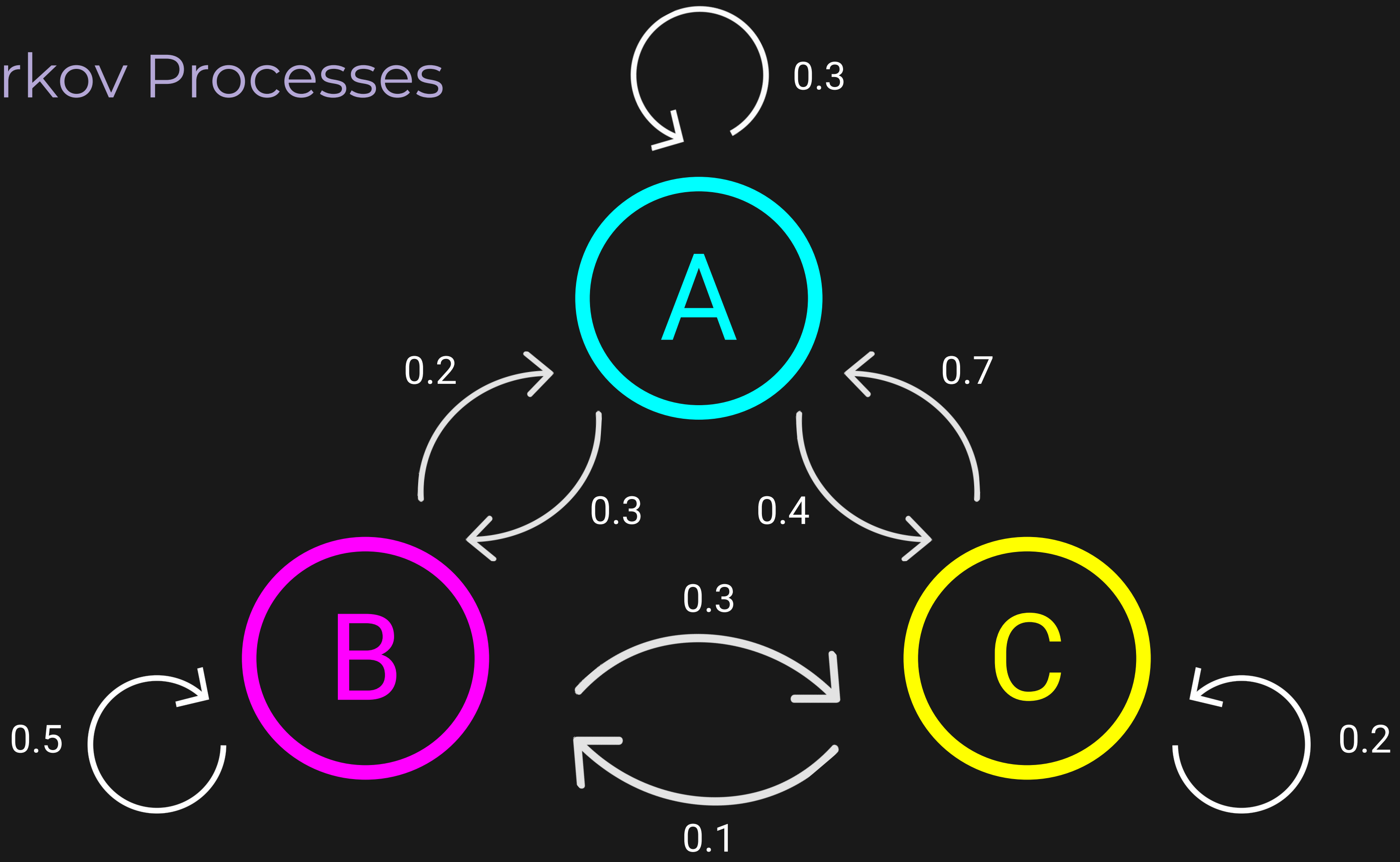


From http://me.eng.usc.ac.ir/files/1509793694994.pdf

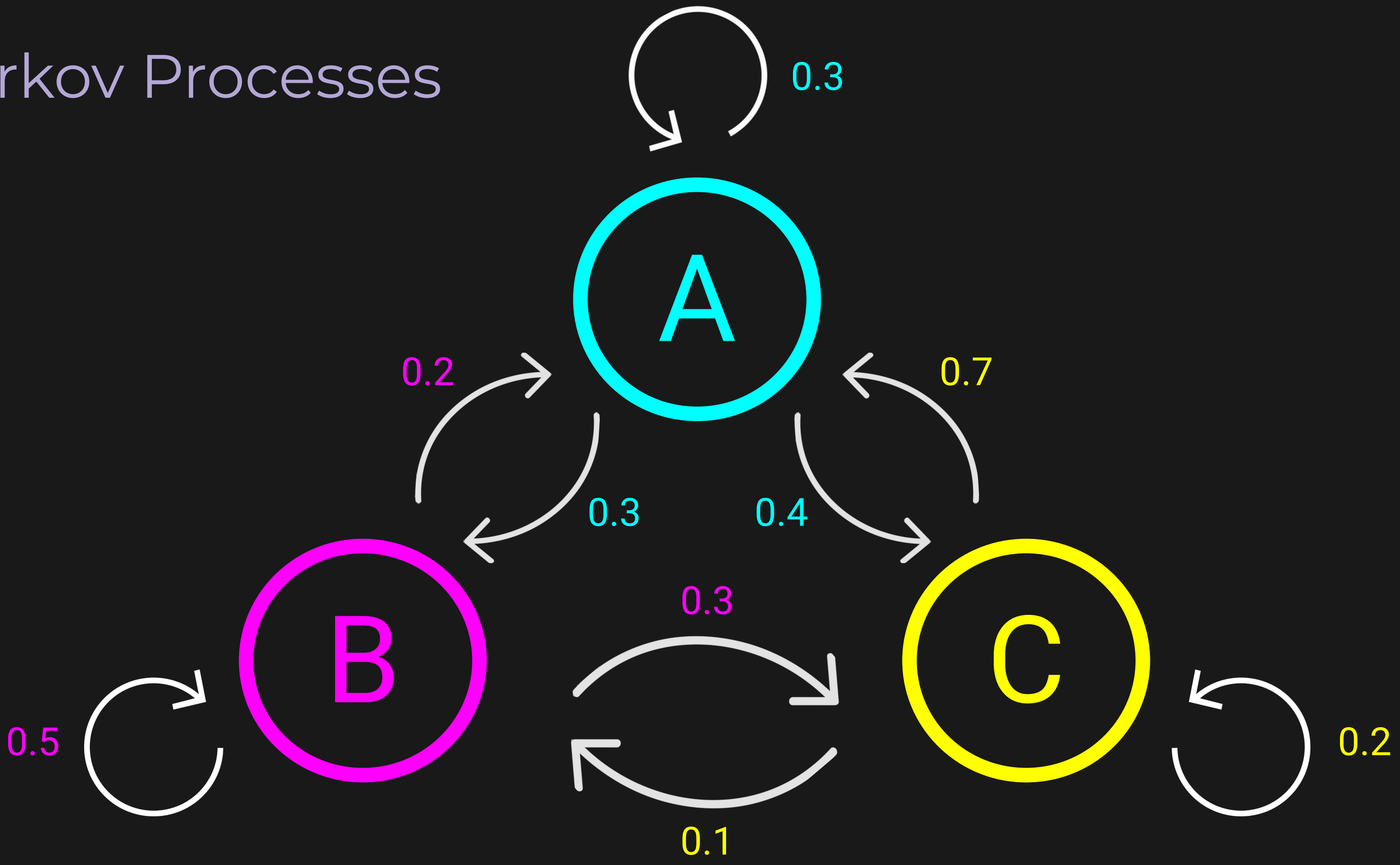
Markov Processes



Markov Processes



Markov Processes



Stochastic Matrices

- A state vector describes the probabilities of being in each state at one time instant

$$x = \begin{bmatrix} x_A \\ x_B \\ x_C \end{bmatrix} \qquad x^{(0)} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Stochastic Matrices

- A stochastic matrix propagates a state vector from one time instant to the next time instant
  - Also called a Markov matrix or a transition matrix

$$A = \begin{bmatrix} 0.3 & 0.2 & 0.7 \\ 0.3 & 0.5 & 0.1 \\ 0.4 & 0.3 & 0.2 \end{bmatrix}$$

$$x^{(1)} = Ax^{(0)}$$

Stochastic Matrices

$$x^{(1)} = Ax^{(0)}$$

$$x^{(1)} = \begin{bmatrix} 0.3 & 0.2 & 0.7 \\ 0.3 & 0.5 & 0.1 \\ 0.4 & 0.3 & 0.2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix}$$

Stochastic Matrices

$$x^{(3)} = AAAx^{(0)}$$

$$x^{(n)} = A^n x^{(0)}$$

Eigendecomposition Shortcut

$$x^{(n)} = A^n x^{(0)}$$

$$x^{(n)} = (Q\Lambda Q^{-1})^n x^{(0)}$$

$$x^{(n)} = (Q\Lambda Q^{-1})(Q\Lambda Q^{-1})(Q\Lambda Q^{-1})\cdots(Q\Lambda Q^{-1})x^{(0)}$$

$$x^{(n)} = Q\Lambda(Q^{-1}Q)\Lambda(Q^{-1}Q)\Lambda Q^{-1}\ldots Q\Lambda Q^{-1}x^{(0)}$$

Eigendecomposition Shortcut

$$x^{(n)} = Q\Lambda\Lambda\Lambda Q^{-1} \cdots Q\Lambda Q^{-1} x^{(0)}$$

$$x^{(n)} = Q\Lambda^n Q^{-1} x^{(0)}$$

$$x^{(n)} = Q \begin{bmatrix} \lambda_1^n & 0 & \cdots & 0 \\ 0 & \lambda_2^n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_k^n \end{bmatrix} Q^{-1} x^{(0)}$$

Eigenvectors in NumPy

```
import numpy as np

A = np.array([
    [2, 4, 6],
    [8, -1, 4],
    [-13, 4, 0]
    ])

eigVals, eigVecs = np.linalg.eig(A)
print("Eigenvalues of 3 x 3 example matrix:\n", eigVals, "\n")
print("Eigenvectors of 3 x 3 example matrix:\n", eigVecs, "\n")

identityMatrix = np.eye(5)
eigVals, eigVecs = np.linalg.eig(identityMatrix)
print("Eigenvalues of 5 x 5 identity matrix:\n", eigVals, "\n")
print("Eigenvectors of 5 x 5 identity matrix:\n", eigVecs, "\n")
```

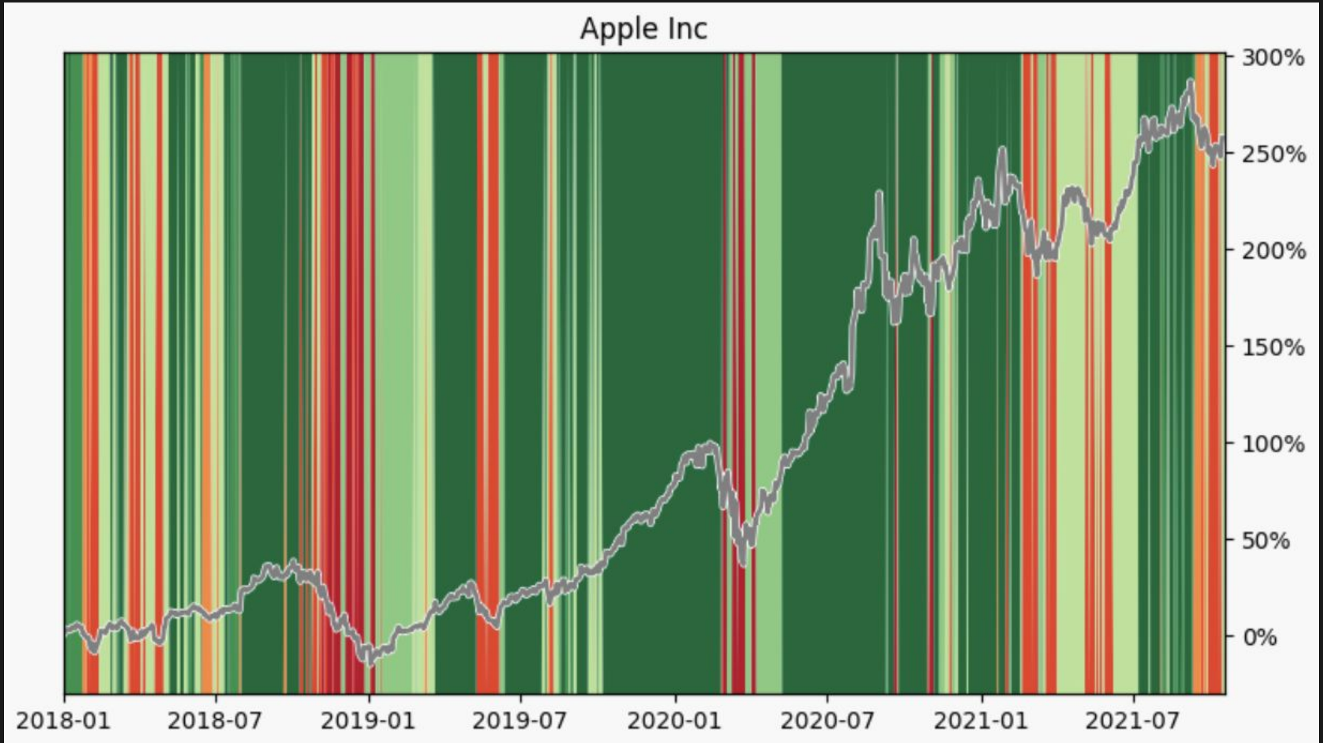
Eigenvectors in NumPy

```
Eigenvalues of 3 x 3 example matrix:
 [-3.09680938+0.j          2.04840469+6.04078438j  2.04840469-6.04078438j]

Eigenvectors of 3 x 3 example matrix:
 [[-0.09755463+0.j          0.23029775+0.43122305j  0.23029775-0.43122305j]
 [-0.78784428+0.j          0.39293866+0.35301307j  0.39293866-0.35301307j]
 [ 0.60809908+0.j         -0.69425544+0.j         -0.69425544-0.j        ]]

Eigenvalues of 5 x 5 identity matrix:
 [1. 1. 1. 1. 1.]

Eigenvectors of 5 x 5 identity matrix:
 [[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

```python
import numpy as np

# number of transitions
n = 10000

# stochastic matrix
A = np.array([
    [0.3, 0.2, 0.7],
    [0.3, 0.5, 0.1],
    [0.4, 0.3, 0.2]
    ])

# initial state vector
x0 = np.array([0, 1, 0])

# calculate xn (final state vector) with simple matrix multiplication
xn = x0

# relatively slow and tedious
for step in range(n):
    xn = A.dot(xn)

print("x0:\n", x0, "\n")
print("x1:\n", A.dot(x0), "\n")
print("final state vector (xn):\n", xn, "\n")

# calculate xn (final state vector) via spectral theorem shortcut
_, eigVecs = np.linalg.eig(A)
diagonalizedMatrix = np.linalg.inv(eigVecs).dot(A).dot(eigVecs)
mainDiag = np.diag(diagonalizedMatrix)
B = np.eye(A.shape[0])

# this is much faster!
for i in range(B.shape[0]):
    B[i, i] = mainDiag[i] ** n

finalStochasticMatrix = eigVecs.dot(B).dot(np.linalg.inv(eigVecs))

print("final stochastic matrix:\n", finalStochasticMatrix, "\n")
print("final state vector (xn) again:\n", finalStochasticMatrix.dot(x0), "\n")
```

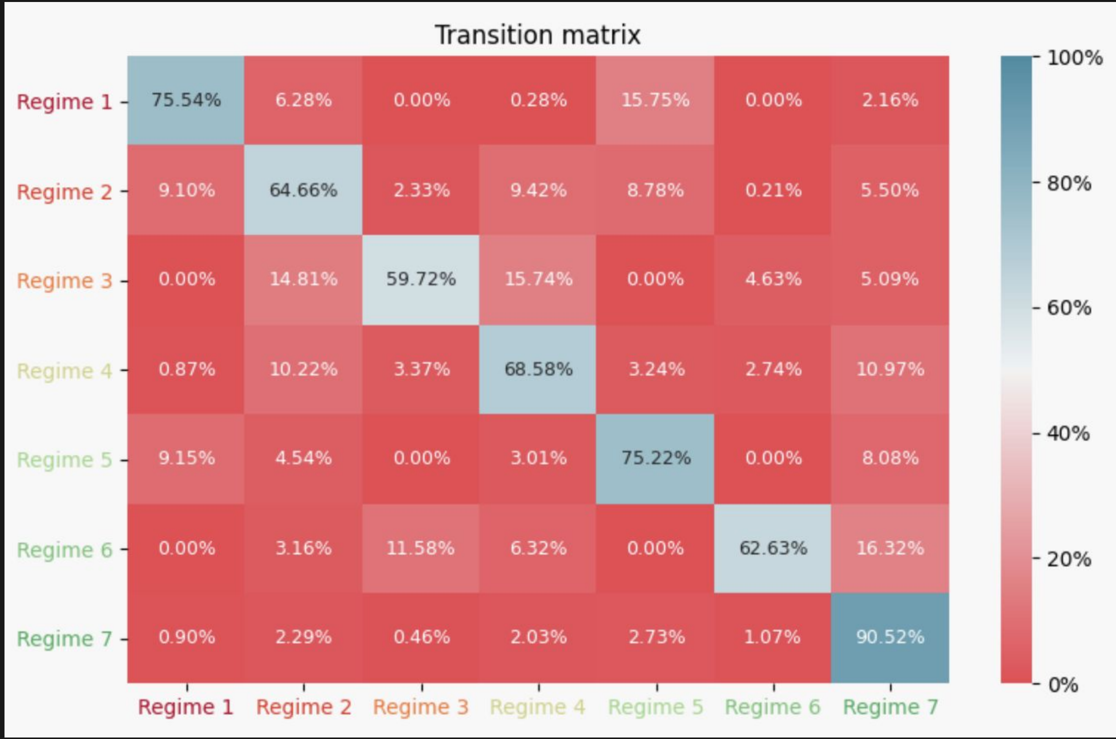
```
x0:
 [0 1 0]

x1:
 [0.2 0.5 0.3]

final state vector (xn):
 [0.39361702 0.29787234 0.30851064]

final stochastic matrix:
 [[0.39361702 0.39361702 0.39361702]
 [0.29787234 0.29787234 0.29787234]
 [0.30851064 0.30851064 0.30851064]]

final state vector (xn) again:
 [0.39361702 0.29787234 0.30851064]
```

Finance



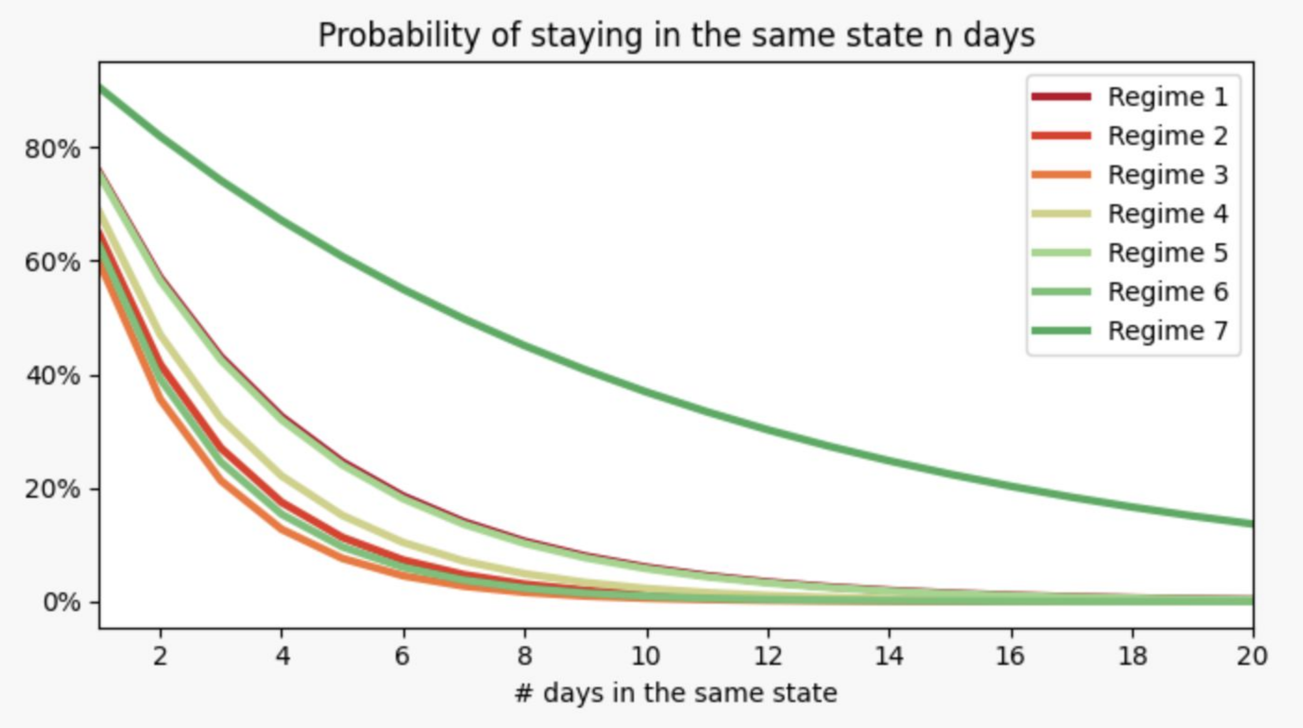
From https://quantdare.com/markov-chain-as-market-predictor/

Finance



From https://quantdare.com/markov-chain-as-market-predictor/

Finance



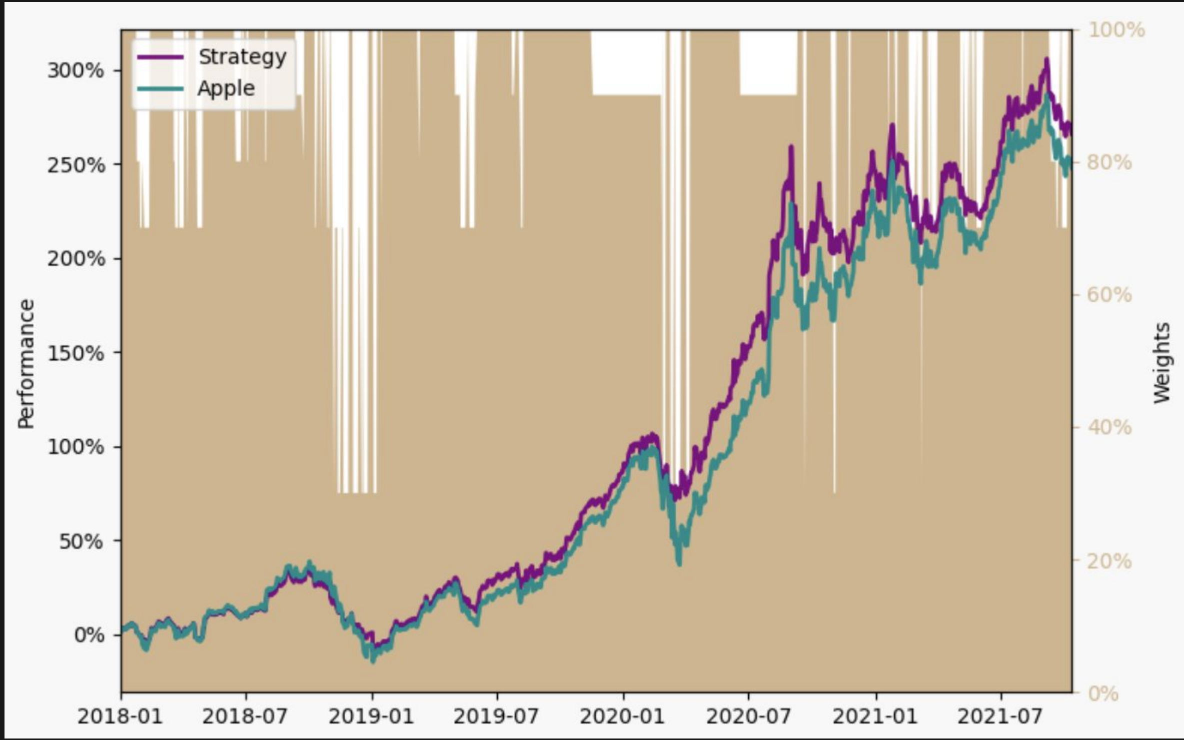
From https://quantdare.com/markov-chain-as-market-predictor/

Finance

- Basic strategy:
  - If it is a strong bear, that is regime is 1, the investment will be 30%
  - If it is a bear, that is regime is 2, the investment will be 70%
  - If it is neutral negative, that is regime is 3, the investment will be 80%
  - If regime is between neutral positive and strong bull, that is regime is 4, 5, 6, or 7, the investment will be 100%
- Based on Markov chain analysis:
  - If regime 7 has been kept more than 30 days, the investment will be reduced 10% up to 90%
  - If regime 1 has been kept more than 9 days, the investment will be increased 10% up to 100%
  - If regime 2 or 3 have been kept more than 6 days, the investment will be increased 10% up to 100%

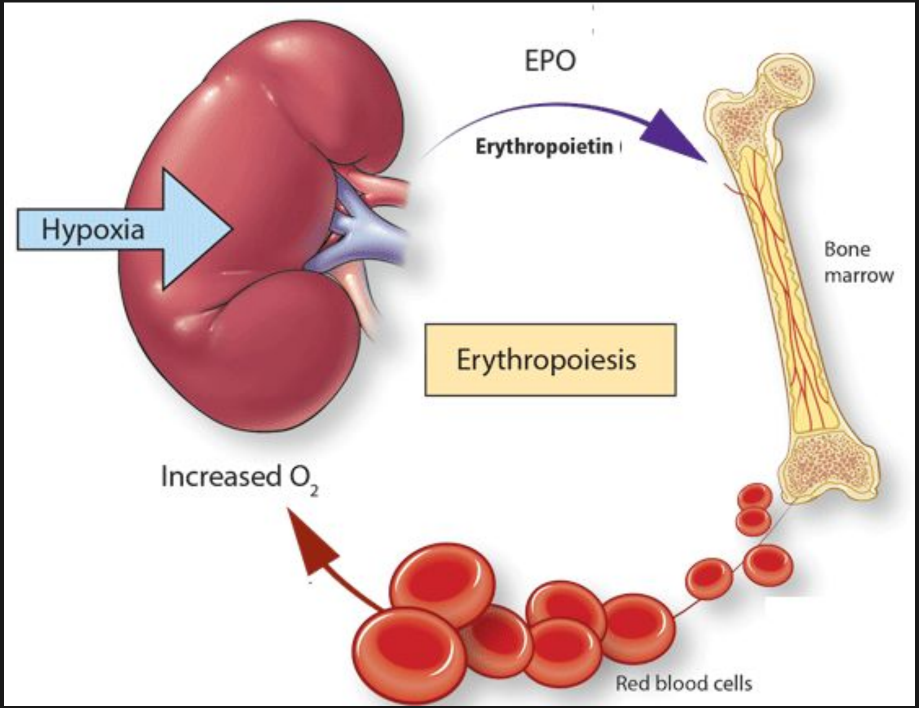
From https://quantdare.com/markov-chain-as-market-predictor/

Finance



From https://quantdare.com/markov-chain-as-market-predictor/

Red Blood Cells



From https://thefactfactor.com/facts/pure_science/biology/human-biology/erythrocytes/4557/

Red Blood Cells

- Red blood cells (RBCs) are constantly being created and destroyed every day, but their number is stable over the long run
- The spleen filters out and destroys a fixed fraction of RBCs each day
- Bone marrow produces new RBCs each day proportional to the number destroyed the previous day
- Problem: What is the RBC count on the nth day?

From https://tglab.princeton.edu/wp-content/uploads/2011/03/RBC.pdf

Red Blood Cells

$$R[n+1] = (1-f)R[n] + M[n]$$

$$M[n+1] = \gamma f R[n]$$

$$\begin{bmatrix} R[n+1] \\ M[n+1] \end{bmatrix} = \begin{bmatrix} 1-f & 1 \\ \gamma f & 0 \end{bmatrix} \begin{bmatrix} R[n] \\ M[n] \end{bmatrix}$$

Red Blood Cells

$$\begin{bmatrix} R[n] \\ M[n] \end{bmatrix} = \begin{bmatrix} 1-f & 1 \\ \gamma f & 0 \end{bmatrix}^n \begin{bmatrix} R[0] \\ M[0] \end{bmatrix}$$

$$\begin{bmatrix} R[n] \\ M[n] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ f & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -f \end{bmatrix}^n \begin{bmatrix} \frac{1}{1+f} & \frac{1}{1+f} \\ \frac{f}{1+f} & \frac{-1}{1+f} \end{bmatrix} \begin{bmatrix} R[0] \\ M[0] \end{bmatrix}$$

$$\begin{bmatrix} R[n] \\ M[n] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ f & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & (-f)^n \end{bmatrix} \begin{bmatrix} \frac{1}{1+f} & \frac{1}{1+f} \\ \frac{f}{1+f} & \frac{-1}{1+f} \end{bmatrix} \begin{bmatrix} R[0] \\ M[0] \end{bmatrix}$$

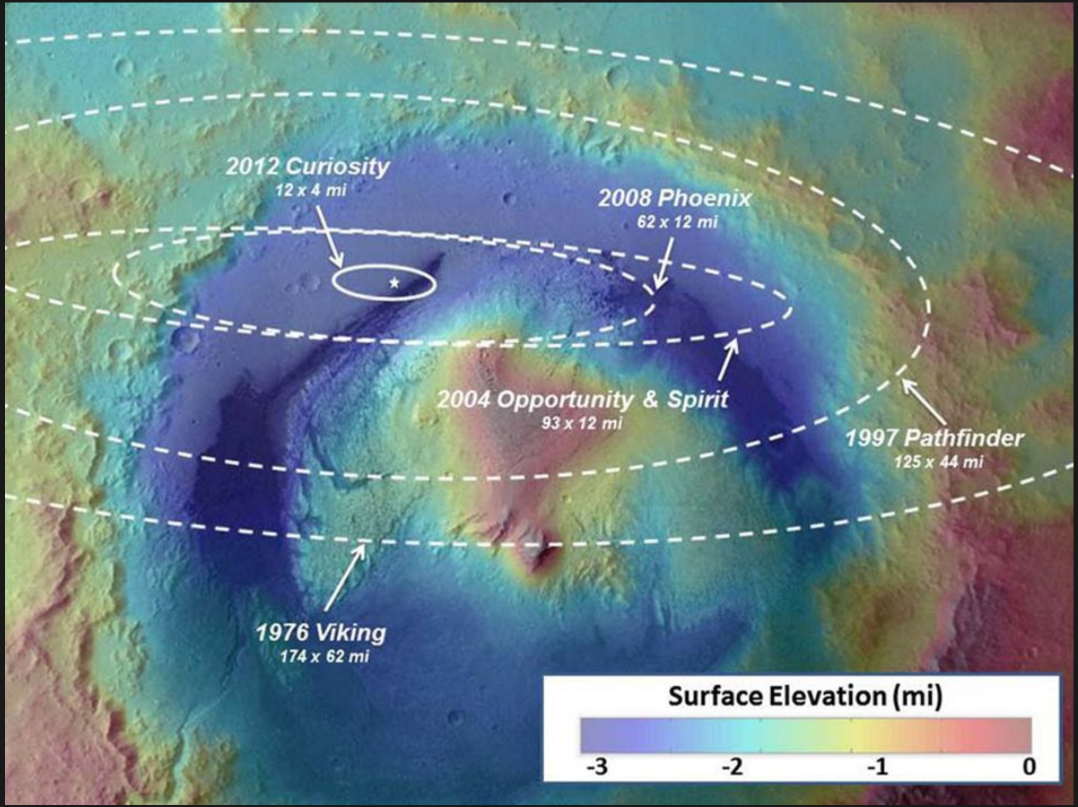
Red Blood Cells

$$R[n] = \frac{1 + (f)(-f)^n}{1+f} R[0] + \frac{1 - (-f)^n}{1+f} M[0]$$

$$0 \le f \le 1 \quad \Longrightarrow \quad \lim_{n \to \infty} R[n] = \frac{1}{1+f} R[0] + \frac{1}{1+f} M[0]$$

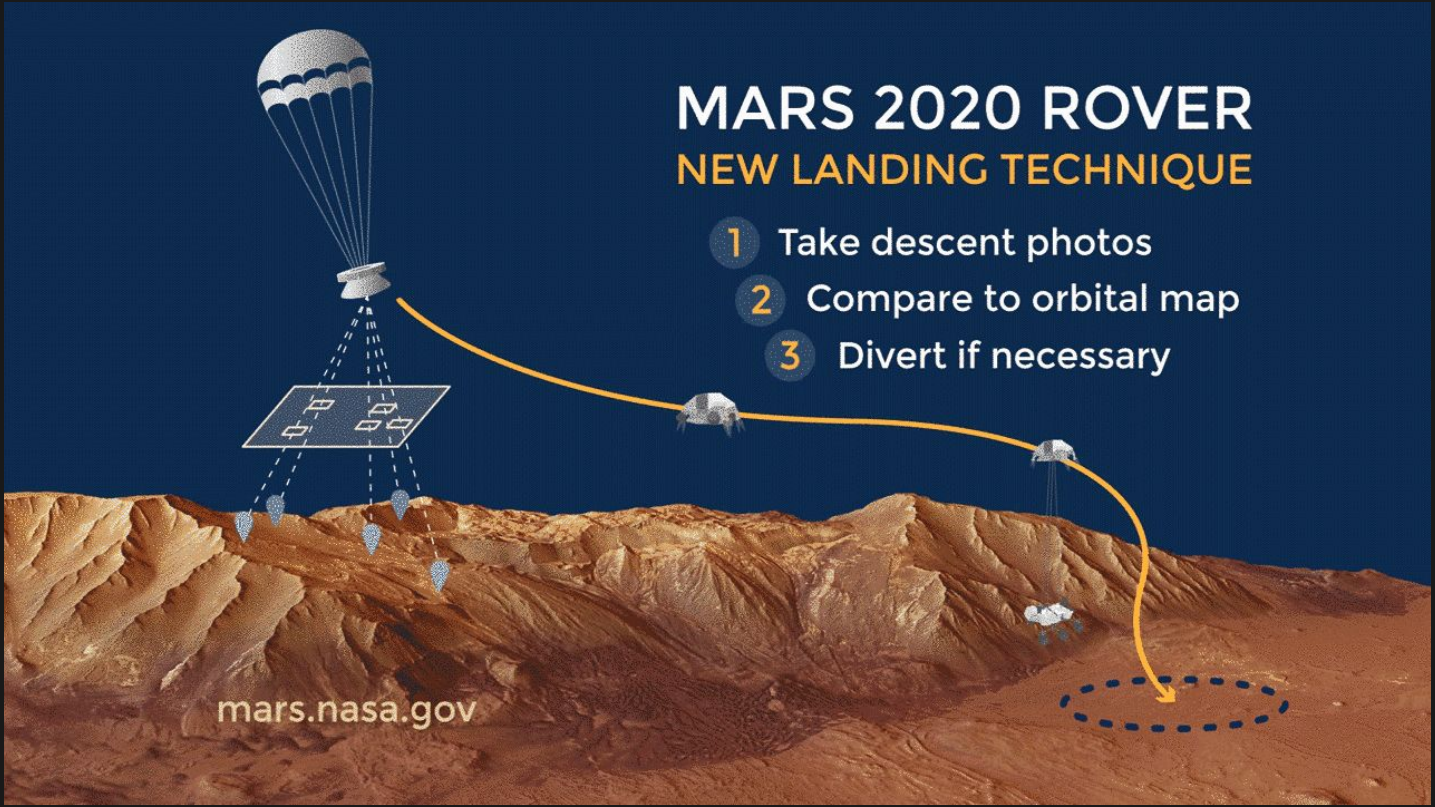
$$= \frac{R[0] + M[0]}{1+f}$$

Ellipsoids

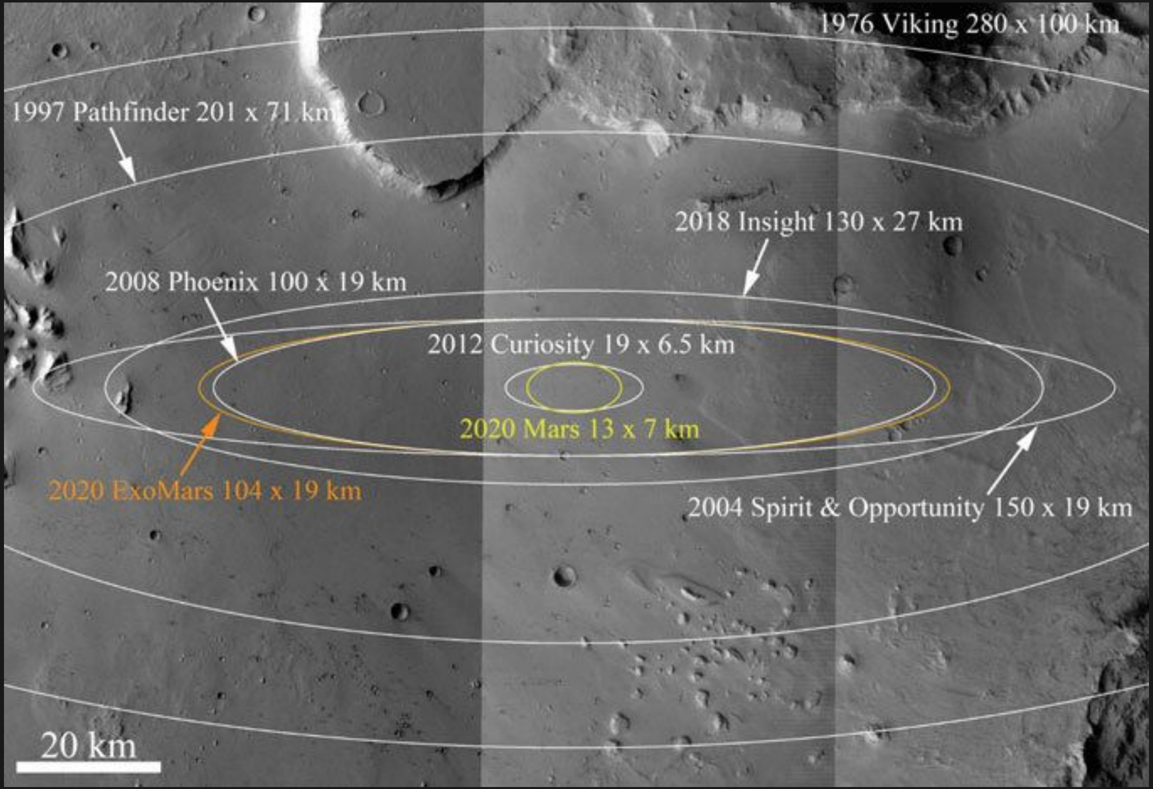


From https://www.nasa.gov/mission_pages/msl/multimedia/pia16039.html

Ellipsoids


MARS 2020 ROVER
NEW LANDING TECHNIQUE
1 Take descent photos
2 Compare to orbital map
3 Divert if necessary
mars.nasa.gov

Ellipsoids



From https://www.researchgate.net/publication/331289183_Planetary_Mapping_for_Landing_Sites_Selection_The_Mars_Case_Study

Ellipsoids

$$\mathcal{E} = \{x \mid x^T A x \leq 1\}$$

- If A is a symmetric matrix that satisfies a *special property*, then this set of vectors forms an ellipsoid
- Eigenvectors and eigenvalues of A determine the directions and lengths of the semiaxes, respectively
- We'll see in the next lecture what this special property is

Next Time

- Quadratic forms
- Matrix norms