**Problem** Shrinkage

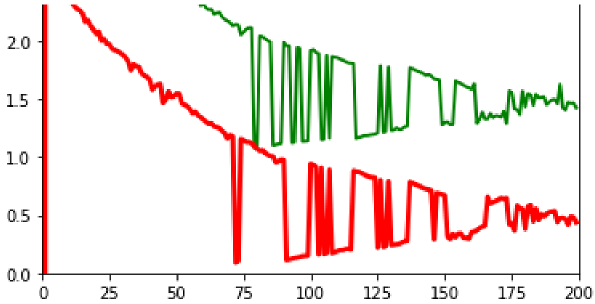
A point raised by Ledoit and Wolf (2003, 2004) is that the covariance matrix $\sum$ is poorly estimated by its sample counterpart and can be greatly improved by their shrinkage estimators when $p$ (number of assets) is not small in comparison with $n$ (number of observations), which is often the case in portfolio management.

**Question**

**My solution**

In [2]:

```
import cvxpy as cp
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA

# simulate X from normal dist. with size 200*400
X = np.random.normal(0,1,(200,400))

# number of factors is 3
pca_X_3 = PCA(3)
# reference for PCA: https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60
princomp_X_3 = pca_X_3.fit_transform(X)
principalDf = pd.DataFrame(data = princomp_X_3,
              columns = ['PrincipalComp 1', 'PrincipalComp 2', 'PrincipalComp 3'])

# reference for projection: https://stackoverflow.com/questions/17836880/orthogonal-projection-wit
h-numpy
Proj = np.dot(np.linalg.inv(X.T.dot(X)), X.T)
Proj = np.dot(X, Proj)
# Generate estimator with the sample variance of the residuals
# after projecting X to the space orthonormal to PCA loadings
sigma_square = np.var(X - Proj @ X)

# S
inner_term = np.eye(400) - (1/400) * np.dot(np.ones(400), np.ones(400).T)
S = (1/400) * np.dot(X, np.dot(inner_term, X.T))

# F
beta_hat = principalDf.to_numpy()
F = np.dot(beta_hat, beta_hat.T) + sigma_square * np.eye(200)

# Solve optimization problem
alpha = cp.Variable()
obj_fn = cp.Minimize(cp.norm(alpha * F + (1 - alpha) * S, "fro"))
problem = cp.Problem(obj_fn)
problem.solve()

S_eval = np.linalg.eigvals(S)
R_eval = np.linalg.eigvals(alpha.value * F + (1 - alpha.value) * S)

plt.plot(R_eval, label="$\lambda$'s of R", color='green',linewidth=2.0)
plt.plot(S_eval, label="$\lambda$'s of S", color='red',linewidth=3.0)
plt.axis([0,200,0,3])
plt.legend()
plt.show()
```




In [ ]: