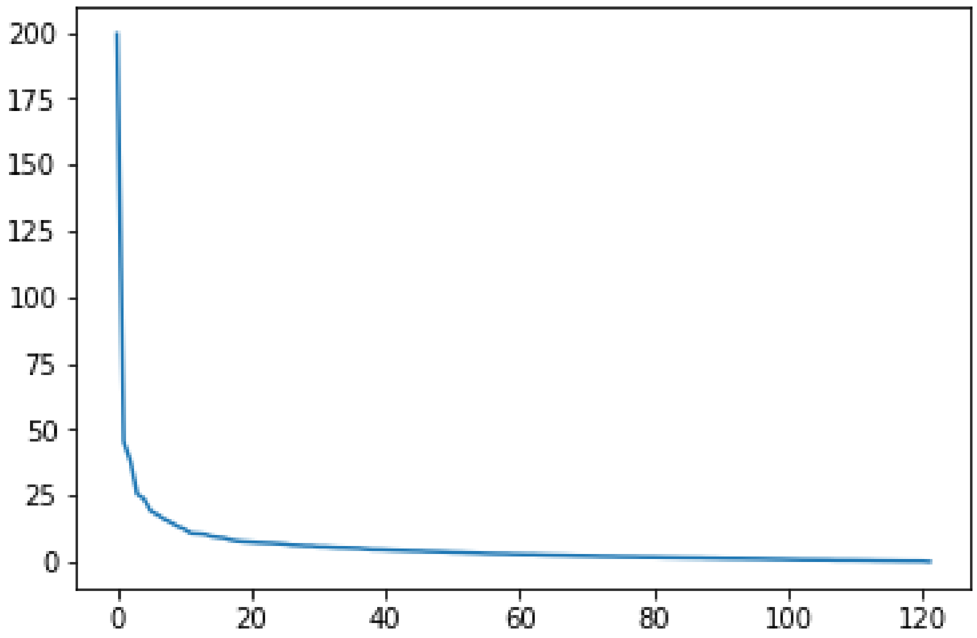
Code:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt


#Task-1: Load the crime dataset and store it as a matrix


#data was first downloaded into my machine and saves as "data.txt"

#attribute was properly labeled from repository data description


crimedata = pd.read_csv('data.txt',sep=',',names=['state','county', 'community','communityname','fold','population','householdsize','racepctblack','racePctWhite','racePctAsian','racePctHisp','agePct12t21','agePct12t29','agePct16t24','agePct65up','numbUrban','pctUrban','medIncome','pctWWage','pctWFarmSelf','pctWInvInc','pctWSocSec','pctWPubAsst','pctWRetire','medFamInc','perCapInc','whitePerCap','blackPerCap','indianPerCap','AsianPerCap','OtherPerCap','HispPerCap','NumUnderPov','PctPopUnderPov','PctLess9thGrade','PctNotHSGrad','PctBSorMore','PctUnemployed','PctEmploy','PctEmplManu','PctEmplProfServ','PctOccupManu','PctOccupMgmtProf','MalePctDivorce','MalePctNevMarr','FemalePctDiv','TotalPctDiv','PersPerFam','PctFam2Par','PctKids2Par','PctYoungKids2Par','PctTeen2Par','PctWorkMomYoungKids','PctWorkMom','NumIlleg','PctIlleg','NumImmig','PctImmigRecent','PctImmigRec5','PctImmigRec8','PctImmigRec10','PctRecentImmig','PctRecImmig5','PctRecImmig8','PctRecImmig10','PctSpeakEnglOnly','PctNotSpeakEnglWell','PctLargHouseFam','PctLargHouseOccup','PersPerOccupHous','PerOwnOccHous','PersPerRentOccHous','PctPersOwnOccup','PctPersDenseHous','PctHousLess3BR','MedNumBR','HousVacant','PctHousOccup','PctHousOwnOcc','PctVacantBoarded','PctVacMore6Mos','MedYrHousBuilt','PctHousNoPhone','PctWOFullPlumb','OwnOccLowQuart','OwnOccMedVal','OwnOccHiQuart','RentLowQ','RentMedian','RentHighQ','MedRent','MedRentPctHousInc','MedOwnCostPctInc','MedOwnCostPctIncNoMtg','NumInShelters','NumStreet','PctForeignBorn','PctBornSameState','PctSameHouse85','PctSameCity85','PctSameState85','LemasSwornFT','LemasSwFTPerPop','LemasSwFTFieldOps','LemasSwFTFieldPerPop','LemasTotalReq','LemasTotReqPerPop','PolicReqPerOffic','PolicPerPop','RacialMatchCommPol','PctPolicWhite','PctPolicBlack','PctPolicHisp','PctPolicAsian','PctPolicMinor','OfficAssgnDrugUnits','NumKindsDrugsSeiz','PolicAveOTWorked','LandArea','PopDens','PctUsePubTrans','PolicCars','PolicOperBudg','LemasPctPolicOnPatr','LemasGangUnitDeploy','LemasPctOfficDrugUn','PolicBudgPerPop','ViolentCrimesPerPop'], encoding='latin-1',engine='python',na_values=['?'])


#datadrame was stored to excel file

writer = pd.ExcelWriter('output_2.xlsx')

crimedata.to_excel(writer,'Sheet1')
```

```
writer.save()


#feature matrix extraction and missing value replacement with mean
df=crimedata.drop(['state','county', 'community','communityname','fold','ViolentCrimesPerPop'], axis=1)
df=df.fillna(df.mean())
print(df)


#Task-2: Compute the eigenvectors and eigenvalues
df=df.values
m = np.asmatrix(df)


print(m)
print(m.shape)


#eignevalue only defined for square matrix.Data is not square matrix
#values, vectors = np.linalg.eig(m)
#print(values)
#print(vectors)


#Singular value decomposition using numpy function
u, s, vh = np.linalg.svd(m, full_matrices=True)
#print(u.shapre),print(s.shape),print(vh.shape)
#SVD is usually described for the factorization of a 2D matrix A.
#The higher-dimensional case will be discussed below.
#In the 2D case, SVD is written as A = U S V^H, where A = a, U= u, S= \mathtt{np.diag}(s) and V^H = vh.
#The 1D array s contains the singular values of a and u and vh are unitary.
#The rows of vh are the eigenvectors of A^H A and the columns of u are the eigenvectors of A A^H.
#In both cases the corresponding (possibly non-zero) eigenvalues are given by s**2.
```

#task-3- Report a table with the top 20 eigenvalues, is there a clear point where you could cut o_ the #dimensions?

```
plt.plot(s)
array([1.99265184e+02, 4.52777778e+01, 3.80249413e+01, 2.55167068e+01,
    2.37178608e+01, 1.91912938e+01, 1.76521547e+01, 1.60281819e+01,
    1.47647753e+01, 1.32948669e+01, 1.22509574e+01, 1.06720545e+01,
    1.05332430e+01, 1.03464017e+01, 9.54261448e+00, 9.16899923e+00,
    8.83913698e+00, 8.13406379e+00, 7.72554309e+00, 7.51147904e+00,
    7.36422858e+00, 7.18773286e+00, 7.05902943e+00, 6.95786519e+00,
    6.77921500e+00, 6.58162820e+00, 6.23875753e+00, 6.12052879e+00,
    5.91653452e+00, 5.87762084e+00, 5.60797111e+00, 5.58480136e+00,
    5.41249323e+00, 5.28124431e+00, 5.22657999e+00, 5.11347823e+00,
    5.03282648e+00, 4.79723790e+00, 4.58151142e+00, 4.55016626e+00,
    4.46317852e+00, 4.33522553e+00, 4.26516552e+00, 4.17257444e+00,
    4.07378248e+00, 4.03834807e+00, 3.89151747e+00, 3.82452223e+00,
    3.75000667e+00, 3.64992548e+00, 3.51564830e+00, 3.47032241e+00,
    3.36073170e+00, 3.28630495e+00, 3.18522586e+00, 3.05719284e+00,
    2.97631016e+00, 2.92007576e+00, 2.87811806e+00, 2.77376894e+00,
    2.72831420e+00, 2.66345777e+00, 2.65938843e+00, 2.55565722e+00,
    2.48703129e+00, 2.42700937e+00, 2.39068887e+00, 2.34157702e+00,
    2.25802622e+00, 2.20631020e+00, 2.15230013e+00, 2.09558441e+00,
    2.00616656e+00, 1.97381375e+00, 1.93665709e+00, 1.86005691e+00,
    1.82828180e+00, 1.80492631e+00, 1.73037247e+00, 1.70281966e+00,
    1.68238211e+00, 1.62226642e+00, 1.56929254e+00, 1.54332491e+00,
    1.49473845e+00, 1.46319934e+00, 1.43982189e+00, 1.41165062e+00,
    1.33362745e+00, 1.30726883e+00, 1.25031592e+00, 1.20913070e+00,
    1.18075508e+00, 1.15744894e+00, 1.10834698e+00, 1.08775591e+00,
    1.04757203e+00, 9.89679589e-01, 9.31885590e-01, 9.06703941e-01,
    8.32788084e-01, 7.84805535e-01, 7.23990377e-01, 7.04975201e-01,
    6.88913179e-01, 6.63693013e-01, 6.42062918e-01, 5.59514137e-01,
    5.48798800e-01, 5.39320282e-01, 5.20548692e-01, 4.79239403e-01,
    4.56640876e-01, 3.85088605e-01, 3.65029238e-01, 3.41262213e-01,
    2.99025996e-01, 2.62145188e-01, 2.38825116e-01, 2.33573565e-01,
    2.02425538e-01, 6.58347346e-03])
```


200
175
150
125
100
75
50
25
0
0
20
40
60
80
100
120