Q1

```python
import numpy as np
from sklearn.mixture import GaussianMixture
from scipy.stats import multivariate_normal
import matplotlib.pyplot as plt

x = np.loadtxt("gmm_mixture.csv", delimiter=",", dtype=float)


components = np.array([2, 3, 4, 5, 6, 7, 8, 9, 10])
likelihood_dataset = np.zeros(9)

for i in range(2, 11):
    gmm = GaussianMixture(n_components=i, n_init=10)
    gmm.fit(x)
    # print("means")
    means = gmm.means_
    # print(means)
    # print("covariances")
    covariances = gmm.covariances_
    # print(covariances)
    # print("weights")
    weights = gmm.weights_
    # print(gmm.weights_)

    # likelihood_dataset = 1.0
    # for dataitem in x:
    #     likelihood_item = 0.0000000000000
    #     for group in range(0, i):
    #         # likelihood_item += weights[group] * multivariate_normal.pdf(dataitem, mean=means[group], cov=covariances[group])
    #         # likelihood_item += weights[group] * (1/(((2*np.pi)**(len(dataitem)/2))*np.sqrt(np.linalg.det(covariances[group]))))*np.exp(-0.5*np.dot(np.dot((dataitem-means
    #     likelihood_dataset *= likelihood_item

    print(gmm.score(x))
    likelihood_dataset[i-2] = np.exp(gmm.score(x))

    # print("likelihood when J = " + str(i))
    # print("%.8f" % likelihood_dataset[i-2])


plt.grid()
plt.plot(components,likelihood_dataset)
plt.xlabel('J')
plt.ylabel('likelihood')
plt.show()
```

(C):

```
means
[[-2.33183714  1.00813233]
 [ 2.37461544  1.35243087]]
covariances
[[[ 1.07436028 -0.21875163]
  [-0.21875163  5.50764875]]

 [[ 1.06836247  0.42504138]
  [ 0.42504138  4.94169212]]]
weights
[0.60077442 0.39922558]
likelihood when J = 2
0.01308483
```

(D):

```
means
[[ 2.49628637  2.46308988]
 [-2.49478905  2.49946996]
 [-0.70380971 -2.04396713]]
covariances
[[[ 1.01703782 -0.03577669]
  [-0.03577669  1.10624868]]

 [[ 0.99177281  0.49609886]
  [ 0.49609886  1.05173229]]

 [[ 4.5690307  -0.22259568]
  [-0.22259568  0.96134876]]]
weights
[0.30305006 0.40138959 0.29556035]
likelihood when J = 3
0.01691473
```

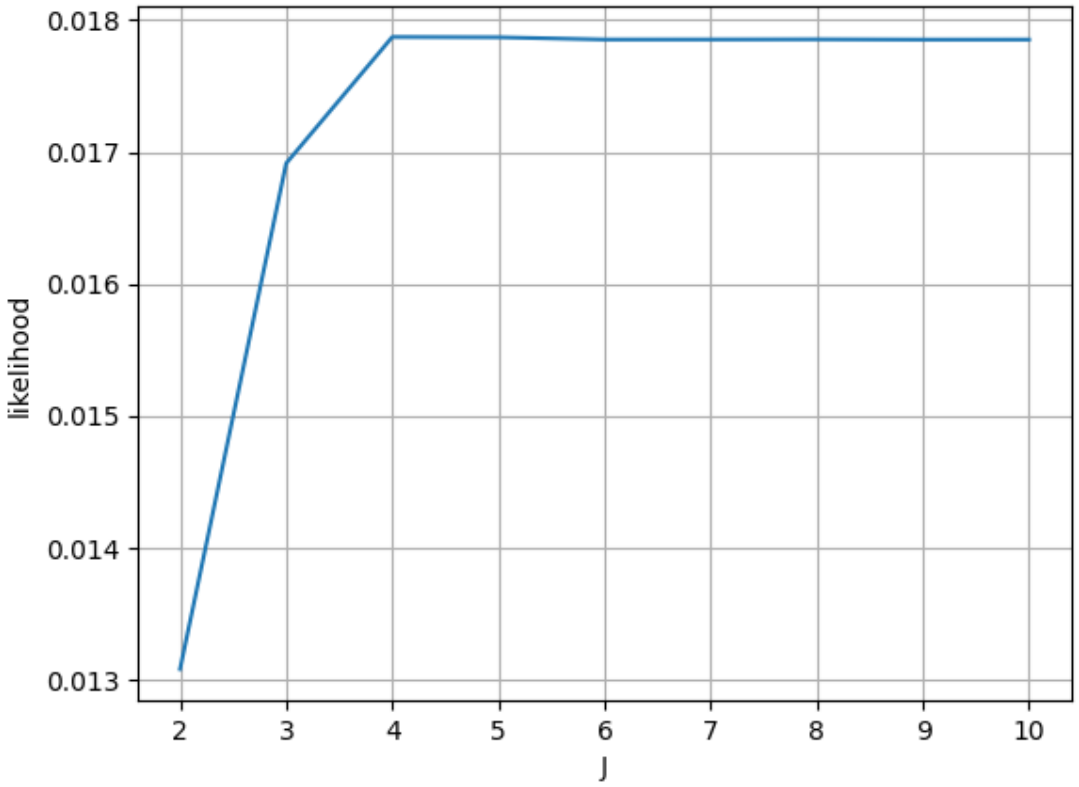
```
means
[[-2.49442365  2.51197673]
 [ 1.98812567 -1.98130459]
 [ 2.49234079  2.50536627]
 [-2.03123184 -2.00490212]]
covariances
[[[ 1.00249489  0.50000281]
  [ 0.50000281  1.02879499]]

 [[ 1.09203627  0.06314679]
  [ 0.06314679  1.12824698]]

 [[ 1.02647445 -0.02593194]
  [-0.02593194  1.00599639]]

 [[ 1.02304681 -0.27781611]
  [-0.27781611  0.99321289]]]
weights
[0.39928922 0.10320096 0.29809998 0.19940984]
likelihood when J = 4
0.01787178
```

```
means
[[-3.02467488  1.95888255]
 [ 2.48867342  2.50625466]
 [ 2.0030284  -1.97952209]
 [-2.03622139 -2.00046037]
 [-1.91571355  3.12079301]]
covariances
[[[ 0.68717479  0.1370166 ]
  [ 0.1370166   0.65349646]]

 [[ 1.03337274 -0.03209704]
  [-0.03209704  1.01374655]]

 [[ 1.06516195  0.04744636]
  [ 0.04744636  1.114907  ]]

 [[ 1.07700039 -0.31030019]
  [-0.31030019  1.00437965]]

 [[ 0.65911169  0.17771053]
  [ 0.17771053  0.6889159 ]]]
weights
[0.20669524 0.2989081  0.10201842 0.20132376 0.19105448]
likelihood when J = 5
0.01785462
```

```
means
[[ 2.42108305  3.13442996]
 [-3.0257938   1.95975261]
 [ 2.01258639 -1.93625044]
 [-2.03616975 -1.99910667]
 [-1.93758602  3.09806193]
 [ 2.56362661  1.84142912]]
covariances
[[[ 1.06303441  0.04097219]
  [ 0.04097219  0.58850105]]

 [[ 0.69675983  0.14574704]
  [ 0.14574704  0.66126714]]

 [[ 1.07155039  0.06816554]
  [ 0.06816554  1.18504148]]

 [[ 1.07400694 -0.30801533]
  [-0.30801533  1.00489921]]

 [[ 0.67327799  0.19637703]
  [ 0.19637703  0.70974992]]

 [[ 0.99235983 -0.0158816 ]
  [-0.0158816   0.50884112]]]
weights
[0.15696294 0.20280444 0.10440942 0.20132737 0.19483538 0.13966046]
likelihood when J = 6
0.01785618
```

likelihood when J = 7
0.01785247
likelihood when J = 8
0.01785355

likelihood when J = 9
0.01785048
likelihood when J = 10
0.01784818
(E):



(F):
when J equals to 4 the likelihood is maximum: 0.01787178

```
means
[[-2.49442365  2.51197673]
 [ 1.98812567 -1.98130459]
 [ 2.49234079  2.50536627]
 [-2.03123184 -2.00490212]]
covariances
[[[ 1.00249489  0.50000281]
  [ 0.50000281  1.02879499]]

 [[ 1.09203627  0.06314679]
  [ 0.06314679  1.12824698]]

 [[ 1.02647445 -0.02593194]
  [-0.02593194  1.00599639]]

 [[ 1.02304681 -0.27781611]
  [-0.27781611  0.99321289]]]
weights
[0.39928922 0.10320096 0.29809998 0.19940984]
likelihood when J = 4
0.01787178
```

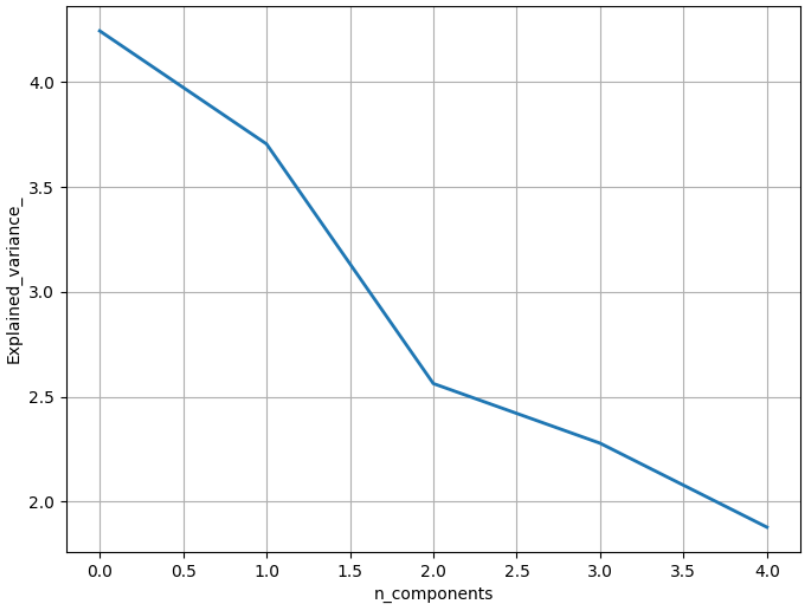
Q2

codework2_2.py 3 | pca_data.csv

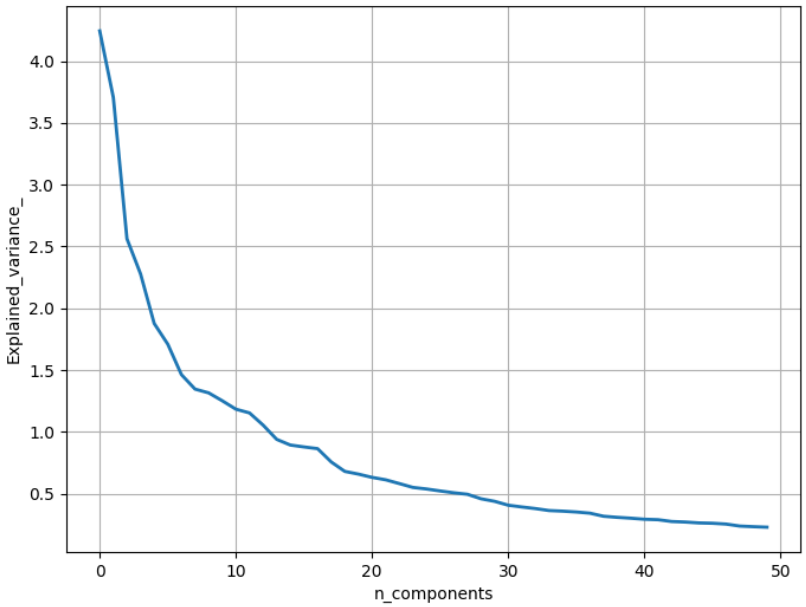
codework2_2.py > ...

```python
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

x = np.loadtxt("pca_data.csv", delimiter=",", dtype=float)

pca = PCA(n_components = 200)
pca.fit(x)

plt.figure(1, figsize=(8, 6))
plt.grid()
plt.plot(pca.explained_variance_, linewidth=2)
plt.axis('tight')
plt.xlabel('n_components')
plt.ylabel('Explained_variance_')
plt.show()
```

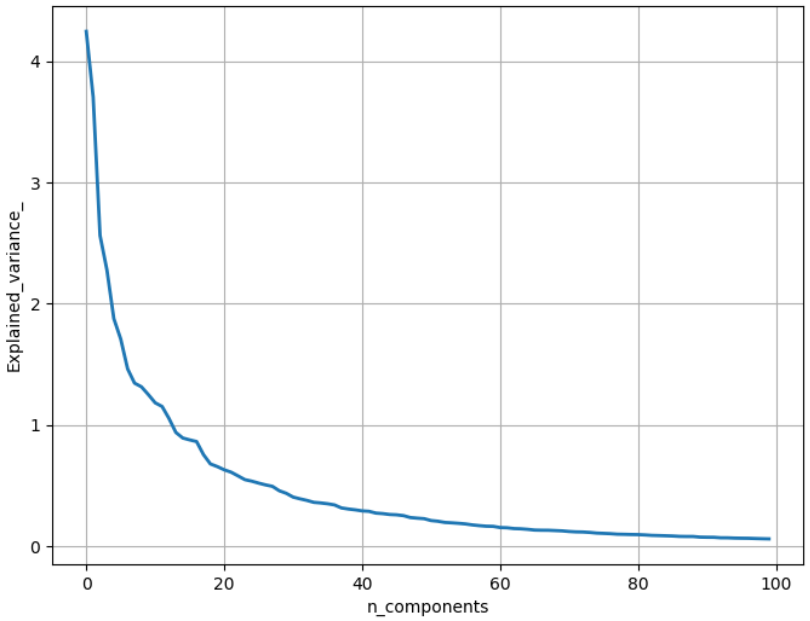
when n = 5



when n = 50



when n = 100



when n = 200