

Number of PCs->	0N	1N	2N	3N	4N	0c	1c	2c	3c	4c
Dataset I	4.542471	0.38345	0.175563	0.141784	0.160838	4.543119	0.384614	0.177815	0.144441	0.160838
Dataset II	4.542471	0.641093	0.715628	0.908393	1.115658	4.549539	0.648642	0.750621	0.941973	1.115658
Dataset III	4.542471	1.290372	1.96724	2.650841	3.65328	4.557473	1.323462	2.119748	3.02738	3.65328
Dataset IV	4.542471	0.799943	0.828083	0.98495	1.194	4.566199	0.840614	1.20709	1.271192	1.194
Dataset V	4.542471	1.917768	3.331722	4.548257	5.139267	4.919928	2.835679	4.651435	4.971247	5.139267

```

import numpy as np
import pandas as pd
import math
from sklearn import metrics
from sklearn.decomposition import PCA
from sklearn.metrics import mean_squared_error
import numpy.linalg as linalg
from sklearn.preprocessing import StandardScaler
from numpy.linalg import inv

```

```

origin_data1 = pd.read_csv("dataI.csv")
origin_data2 = pd.read_csv("dataII.csv")
origin_data3 = pd.read_csv("dataIII.csv")
origin_data4 = pd.read_csv("dataIV.csv")
origin_data5 = pd.read_csv("dataV.csv")
origin_noisyless= pd.read_csv("iris.csv")

```

```

#StandardScaler.inverse_transform(pca_data1,pc_data1_1)
#doing PCA with data_1
pca_data0 = PCA(n_components=0)
pc_data1_0 = pca_data0.fit_transform(origin_data1) #this is before reconstructing, has only 1 column
pca_data1_recons = np.dot(pc_data1_0, pca_data0.components_) + pca_data0.mean_
print(mean_sqr(pca0_data1_recons))
print(mean_squared_error(pca0_data1_recons, origin_noisyless)*4)

```

```

pca_data1 = PCA(n_components=1)
pc_data1_1 = pca_data1.fit_transform(origin_data1) #this is before reconstructing, has only 1 column
pcal_data1_recons = np.dot(pc_data1_1, pca_data1.components_) + pca_data1.mean_
print(mean_squared_error(pcal_data1_recons, origin_noisyless)*4)

```

```

#noiseless for data1
pcan_data1 = PCA(n_components=0)
pcan_data1.fit(origin_noisyless)
pcant_data1 = pcant_data1.transform(origin_data1)
pcan_data1_recons = np.dot(pcant_data1, pcant_data1.components_) + pcant_data1.mean_
print(mean_squared_error(pcan_data1_recons, origin_noisyless)*4)

```

```

pcan_data1 = PCA(n_components=1)
pcan_data1.fit(origin_noisyless)
pcant_data1 = pcant_data1.transform(origin_data1)
pcan_data1_recons = np.dot(pcant_data1, pcant_data1.components_) + pcant_data1.mean_
print(mean_squared_error(pcan_data1_recons, origin_noisyless)*4)

```

```

pcan_data1 = PCA(n_components=2)

```

```

pca2_data2 = PCA(n_components=2)
pc_data2_2 = pca2_data2.fit_transform(origin_data2)
pca2_data2_recons = np.dot(pc_data2_2, pca2_data2.components_) + pca2_data2.mean_
print(mean_squared_error(pca2_data2_recons, origin_noisyless)*4)
df = pd.DataFrame(pca2_data2_recons)
df.to_csv("data2_n=2.csv")

```