1. Write Steps Involved in PCA.

STEP 1: STANDARDIZATION

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (for example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.

STEP 2: COVARIANCE MATRIX COMPUTATION

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS

Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data.

What you first need to know about eigenvectors and eigenvalues is that they always come in pairs, so that every eigenvector has an eigenvalue. Also, their number is equal to the number of dimensions of the data. For example, for a 3-dimensional data set, there are 3 variables, therefore there are 3 eigenvectors with 3 corresponding eigenvalues.

It is eigenvectors and eigenvalues who are behind all the magic of principal components because the eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance (most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.

By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.

STEP 4: CREATE A FEATURE VECTOR

As we saw in the previous step, computing the eigenvectors and ordering them by their eigenvalues in descending order, allow us to find the principal components in order of significance. In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call Feature vector.

So, the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep. This makes it the first step towards dimensionality reduction, because if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions.

STEP 5: RECAST THE DATA ALONG THE PRINCIPAL COMPONENTS AXES

In the previous steps, apart from standardization, you do not make any changes on the data, you just select the principal components and form the feature vector, but the input data set remains always in terms of the original axes (i.e, in terms of the initial variables).

In this step, which is the last one, the aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis). This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

2. Perform dimensionality reduction using PCA on the US Arrests dataset (enclosed herewith). What variance can be explained by PC1 & PC2?

In [49]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [4]:
```
df=pd.read_csv("USArrests.csv")
```

In [5]:
```
df.head()
```

Out[5]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
0	Alabama	13.2	236	58	21.2
1	Alaska	10.0	263	48	44.5
2	Arizona	8.1	294	80	31.0
3	Arkansas	8.8	190	50	19.5
4	California	9.0	276	91	40.6

In [7]:
```
df.drop("Unnamed: 0",axis=1,inplace=True)
```

In [9]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

In [14]:
```
df_sc=sc.fit_transform(df)
```

In [17]:
```
from sklearn.decomposition import PCA
```

In [18]:
```
pca=PCA()
```

In [24]:
```
df_pca=pca.fit_transform(df_sc)
```

In [29]:
```
component_names = [f"PC{i+1}" for i in range(df_pca.shape[1])]
pd.DataFrame(df_pca,columns=component_names)
```

Out[29]:

	PC1	PC2	PC3	PC4
0	0.985566	1.133392	-0.444269	0.156267
1	1.950138	1.073213	2.040003	-0.438583
2	1.763164	-0.745957	0.054781	-0.834653
3	-0.141420	1.119797	0.114574	-0.182811
4	2.523980	-1.542934	0.598557	-0.341996
5	1.514563	-0.987555	1.095007	0.001465
6	-1.358647	-1.088928	-0.643258	-0.118469
7	0.047709	-0.325359	-0.718633	-0.881978
8	3.013042	0.039229	-0.576829	-0.096285
9	1.639283	1.278942	-0.342460	1.076797
10	-0.912657	-1.570460	0.050782	0.902807
11	-1.639800	0.210973	0.259801	-0.499104
12	1.378911	-0.681841	-0.677496	-0.122021
13	-0.505461	-0.151563	0.228055	0.424666
14	-2.253646	-0.104054	0.164564	0.017556
15	-0.796881	-0.270165	0.025553	0.206496
16	-0.750859	0.958440	-0.028369	0.670557
17	1.564818	0.871055	-0.783480	0.454728
18	-2.396829	0.376392	-0.065682	-0.330460
19	1.763369	0.427655	-0.157250	-0.559070
20	-0.486166	-1.474496	-0.609497	-0.179599
21	2.108441	-0.155397	0.384869	0.102372
22	-1.692682	-0.632261	0.153070	0.067317
23	0.996494	2.393796	-0.740808	0.215508
24	0.696787	-0.263355	0.377444	0.225824
25	-1.185452	0.536874	0.246889	0.123742
26	-1.265637	-0.193954	0.175574	0.015893
27	2.874395	-0.775600	1.163380	0.314515
28	-2.383915	-0.018082	0.036855	-0.033137
29	0.181566	-1.449506	-0.764454	0.243383
30	1.980024	0.142849	0.183692	-0.339534
31	1.682577	-0.823184	-0.643075	-0.013484
32	1.123379	2.228003	-0.863572	-0.954382
33	-2.992226	0.599119	0.301277	-0.253987
34	-0.225965	-0.742238	-0.031139	0.473916
35	-0.311783	-0.287854	-0.015310	0.010332
36	0.059122	-0.541411	0.939833	-0.237781
37	-0.888416	-0.571100	-0.400629	0.359061
38	-0.863772	-1.491978	-1.369946	-0.613569
39	1.320724	1.933405	-0.300538	-0.131467
40	-1.987775	0.823343	0.389293	-0.109572
41	0.999742	0.860251	0.188083	0.652864
42	1.355138	-0.412481	-0.492069	0.643195
43	-0.550565	-1.471505	0.293728	-0.082314
44	-2.801412	1.402288	0.841263	-0.144890
45	-0.096335	0.199735	0.011713	0.211371
46	-0.216903	-0.970124	0.624871	-0.220848
47	-2.108585	1.424847	0.104775	0.131909
48	-2.079714	-0.611269	-0.138865	0.184104
49	-0.629427	0.321013	-0.240659	-0.166652

In [31]:
```
pca.explained_variance_
```

Out[31]:
```
array([2.53085875, 1.00996444, 0.36383998, 0.17696948])
```

In [32]:
```
pca.explained_variance_ratio_
```

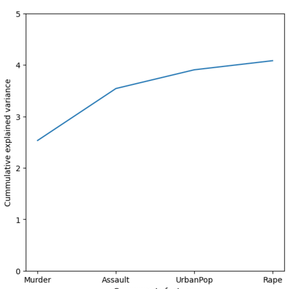
Out[32]:
```
array([0.62006039, 0.24744129, 0.0891408 , 0.04335752])
```

In [35]:
```
evc = np.cumsum(pca.explained_variance_)
print(evc)
```

```
[2.53085875 3.5408232  3.90446318 4.08163265]
```

In [50]:
```
features = ['Murder', 'Assault', 'UrbanPop', 'Rape']
plt.figure(figsize=(6,6))
sns.lineplot(x=features, y=evc)
plt.xlabel("Components features")
plt.ylabel("Cummulative explained variance")
plt.ylim(0,5)
plt.show
```

Out[50]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



3. Why Dimension Reduction is an Important Concept in Data Science?

- It reduces the time and storage space required.
- It helps Remove multi-collinearity which improves the interpretation of the parameters of the machine learning model.
- It becomes easier to visualize the data when reduced to very low dimensions such as 2D or 3D.
- It avoids the curse of dimensionality.
- It removes irrelevant features from the data, Because having irrelevant features in the data can decrease the accuracy of the models and make your model learn based on irrelevant features.

4. Explore Other Dimension Reduction Methods other than PCA. Explain it.

Among machine learning algorithms, there are many basic linear dimension reduction methods including PCA, ICA, linear discriminant analysis (LDA), LFA, and LPP, but most of them are based on the projection of data, which makes data from the dimension reduction difficult to interpret. Now there are two widely used dimension reduction methods, PCA and LDA, which will be introduced in this section.

Linear Discriminant Analysis

LDA, also known as Fisher linear discriminant, is a common way of dimension reduction in machine learning. Unlike PCA, it needs to enter known labels, which means it needs supervised conditions. The concept is to find a low-dimensional space in a high-dimensional space, which can make the mean (center points) distance of two or more types of data be the farthest and the intraclasses variance of each type to be as small as possible. As shown in the figure Fig. 2.9, μ represents the center of the data classes, while the size of the red and blue circles represents the size of variance. It can be seen that by projecting the data onto the diagonal on the left, the centers of red and blue datasets can be separated.