Big Data Project

Carlos Isaac and Jordi Lazo

Universitat de Lleida
Escola Politècnica Superior

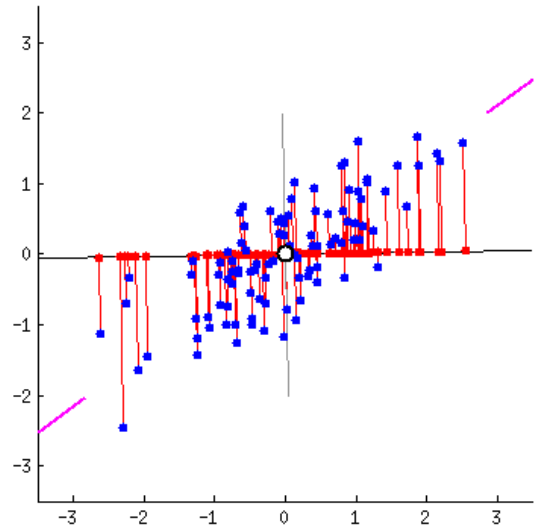
Principal Component Analysis (PCA)

Principal Component Analysis is a dimensionality reduction technique commonly used in statistics and machine learning to transform a dataset into a new coordinate system.

The goal of PCA is to simplify the data while retaining as much of the important information as possible

The objective of PCA is to go from N to P variables. Where $P < N$

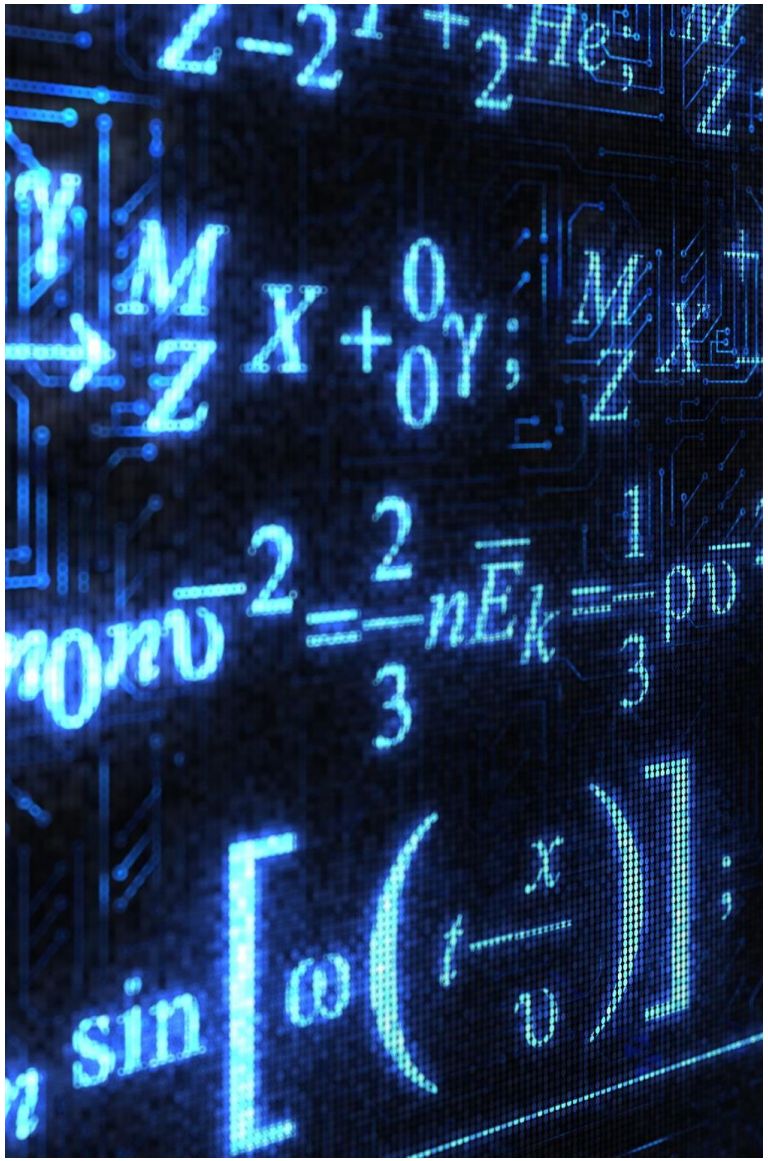
PCA Example



Fire.csv

PCA > Fire.csv

```
1   Area,Temp,RH,Days ,wind
2   1120,35,10,100,55
3   50,10,90,3,1
4   1300,36,5,0,3
5   900,20,25,24,15
6   1500,39,7,120,59
7   10,5,96,2,2
8   5,3,99,5,0
9   1600,36,12,98,67
10  20,14,89,4,10
11  25,12,97,0,1
```



Process of PCA

1- Data transformation

2- The Correlation matrix

3- Eigenvalues and Eigenvectors

4- The first and second principal component component

1- Data transformation

- Obtain a centered and reduced matrix.

PCA > R_PCA_2023_24.r

```r
data <- read.table("Fire.csv",sep=",",header=TRUE)

n<-length(data[,1]) #number of subjects
m<-length(data[1,]) #number of variables
name_v<-c() #column names, variables
name_vl<-c() #row names, individuals
for(i in 1:m){name_v[i]<-colnames(data)[i]}
for(i in 1:n){name_vl[i]<-rownames(data)[i]}
i1<-0
k<-c()
for(i in 1:n){
 for(j in 1:m){
   i1<-i1+1
  k[i1]<-data[i,j]
 }
}

X <- matrix(k,nrow=n,ncol=m,byrow = T)
## obtain the 1 matrix####
mx1<-matrix(rep(1,n),nrow=n, ncol=1)
#### 1 transpose matrix ####
mx1T<-t(mx1)
#### Identity matrix ####
Id <- diag(n)
#### Weight matrix D ###
D <- Id/(n)

#### Compute the  B matrix (centrered X matrix) ####

B<-(Id-mx1 %*% mx1T %*% D) %*% X

##Obtain the average value of each column
X[,1] #first column
mean(X[,1]) #mean first column
mean(X[,2]) #mean second column
X[,1]-mean(X[,1]) #corrected column
B[,1] #compare
X[,2]-mean(X[,2]) #corrected column
B[,2] #compare
S<-(t(B) %*% B)/(n)
D_1s<- diag(sqrt(1/diag(S)), nrow=m, ncol=m)
Y<-B %*% D_1s
```

- Centered X matrix

```
> print(B)
      [,1] [,2] [,3]  [,4]  [,5]
 [1,]  467   14  -43  64.4  33.7
 [2,] -603  -11   37 -32.6 -20.3
 [3,]  647   15  -48 -35.6 -18.3
 [4,]  247   -1  -28 -11.6  -6.3
 [5,]  847   18  -46  84.4  37.7
 [6,] -643  -16   43 -33.6 -19.3
 [7,] -648  -18   46 -30.6 -21.3
 [8,]  947   15  -41  62.4  45.7
 [9,] -633   -7   36 -31.6 -11.3
[10,] -628   -9   44 -35.6 -20.3
```

- Variance-covariance

```
> print(S)
          [,1]   [,2]     [,3]     [,4]     [,5]
[1,] 430396.0 8411.5 -26603.0 23820.70 13590.60
[2,]   8411.5  180.2   -530.8   482.30   274.50
[3,] -26603.0 -530.8   1730.0 -1393.90  -786.30
[4,]  23820.7  482.3  -1393.9  2196.04  1197.02
[5,]  13590.6  274.5   -786.3  1197.02   679.81
```

- Standard deviation

```
> print(D_ls)
             [,1]       [,2]       [,3]       [,4]       [,5]
[1,] 0.001524284 0.00000000 0.00000000 0.00000000 0.00000000
[2,] 0.000000000 0.07449423 0.00000000 0.00000000 0.00000000
[3,] 0.000000000 0.00000000 0.02404235 0.00000000 0.00000000
[4,] 0.000000000 0.00000000 0.00000000 0.02133929 0.00000000
[5,] 0.000000000 0.00000000 0.00000000 0.00000000 0.03835361
```

- Centered and reduced X matrix(Y)

```
> print(Y)
            [,1]       [,2]       [,3]       [,4]       [,5]
 [1,]  0.7118406  1.04291915 -1.0338211  1.3742500  1.2925166
 [2,] -0.9191432 -0.81943648  0.8895670 -0.6956607 -0.7785782
 [3,]  0.9862117  1.11741338 -1.1540329 -0.7596786 -0.7018710
 [4,]  0.3764981 -0.07449423 -0.6731859 -0.2475357 -0.2416277
 [5,]  1.2910685  1.34089605 -1.1059482  1.8010357  1.4459310
 [6,] -0.9801146 -1.19190760  1.0338211 -0.7170000 -0.7402246
 [7,] -0.9877360 -1.34089605  1.1059482 -0.6529821 -0.8169319
 [8,]  1.4434969  1.11741338 -0.9857364  1.3315714  1.7527599
 [9,] -0.9648718 -0.52145958  0.8655247 -0.6743214 -0.4333958
[10,] -0.9572503 -0.67044803  1.0578635 -0.7596786 -0.7785782
```

2- The Correlation matrix

```
R<-cor(X,method="pearson")
```

	[1,]	[2,]	[3,]	[4,]	[5,]
[1,]	1.000000	0.9551288	-0.9749300	0.7748190	0.7945308
[2,]	0.9551288	1.000000	-0.9506715	0.7666899	0.7842801
[3,]	-0.9749300	-0.9506715	1.000000	-0.7151357	-0.7250558
[4,]	0.7748190	0.7666899	-0.7151357	1.000000	0.9796874
[5,]	0.7945308	0.7842801	-0.7250558	0.9796874	1.000000

It can be proved that the R matrix is the variance covariance matrix of the centered and reduced matrix Y

3- Eigenvalues and Eigenvectors

- Eigenvalues represent the magnitude or the amount of variance captured by each eigenvector (principal component). In simple terms, an eigenvalue indicates the significance of its corresponding eigenvector – larger eigenvalues correspond to more significant, more informative eigenvectors.
- Eigenvectors in PCA are essentially the directions in which the data varies the most.
- Each eigenvector found in PCA denotes a principal component, which is a direction in the multi-dimensional space.

Eig$values

Index	Value
[1]	4.37087167
[2]	0.53644798
[3]	0.05300381
[4]	0.02571692
[5]	0.01395962

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.4613613	0.3077012	-0.38705705	0.53256518	-0.50894687
[2,]	-0.4570019	0.3063995	0.83211591	-0.02878238	-0.06342974
[3,]	0.4479731	-0.4445988	0.38867989	0.46276349	-0.48624056
[4,]	-0.4320330	-0.5691288	-0.08022448	-0.53158240	-0.44768743
[5,]	-0.4369926	-0.5383899	0.01618418	0.46777774	0.54781004

Eig$vectors

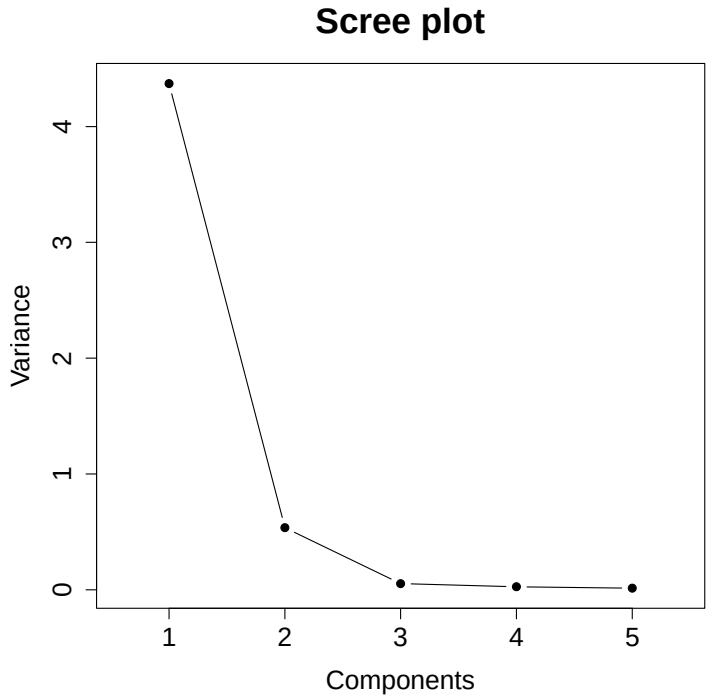
4- The first and second principal component

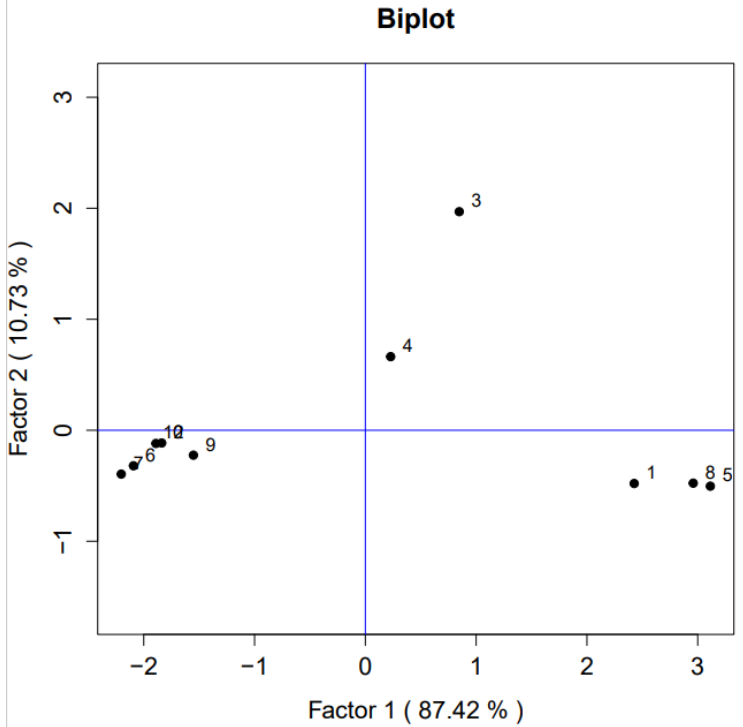
- The first principal component captures the maximum variance in the dataset.
- It is the direction in the data space along which the data varies the most. In mathematical terms, it is the eigenvector corresponding to the largest eigenvalue of the covariance matrix of the data.
- The second principal component is orthogonal to the first principal component, ensuring that it is uncorrelated with the first component.
- It captures the maximum amount of variance that is left after removing the variance captured by the first principal component. In other words, it is the direction of the second most considerable variance in the data.

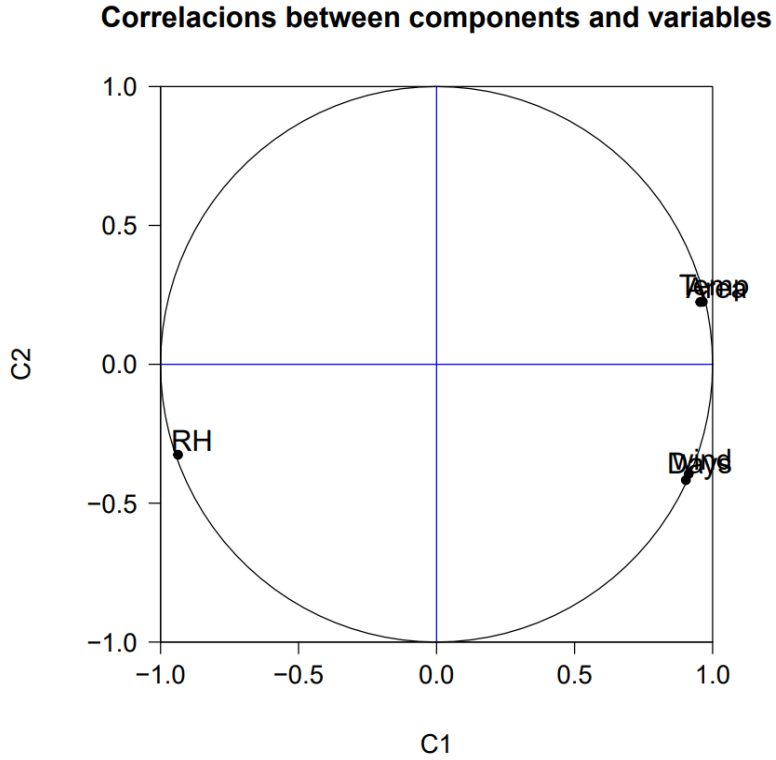
	[,1]	[,2]
[1,]	-0.4613613	0.3077012
[2,]	-0.4570019	0.3063995
[3,]	0.4479731	-0.4445988
[4,]	-0.4320330	-0.5691288
[5,]	-0.4369926	-0.5383899

[1]	4.37087167
[2]	0.53644798

Results of PCA



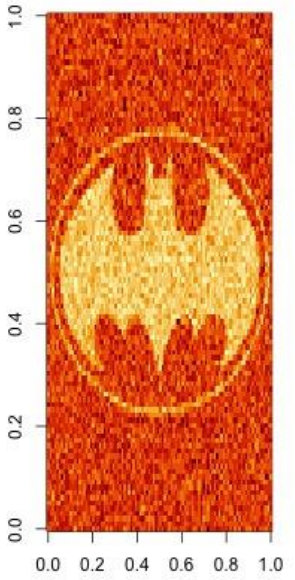




Expectation-Maximization algorithm (EM)

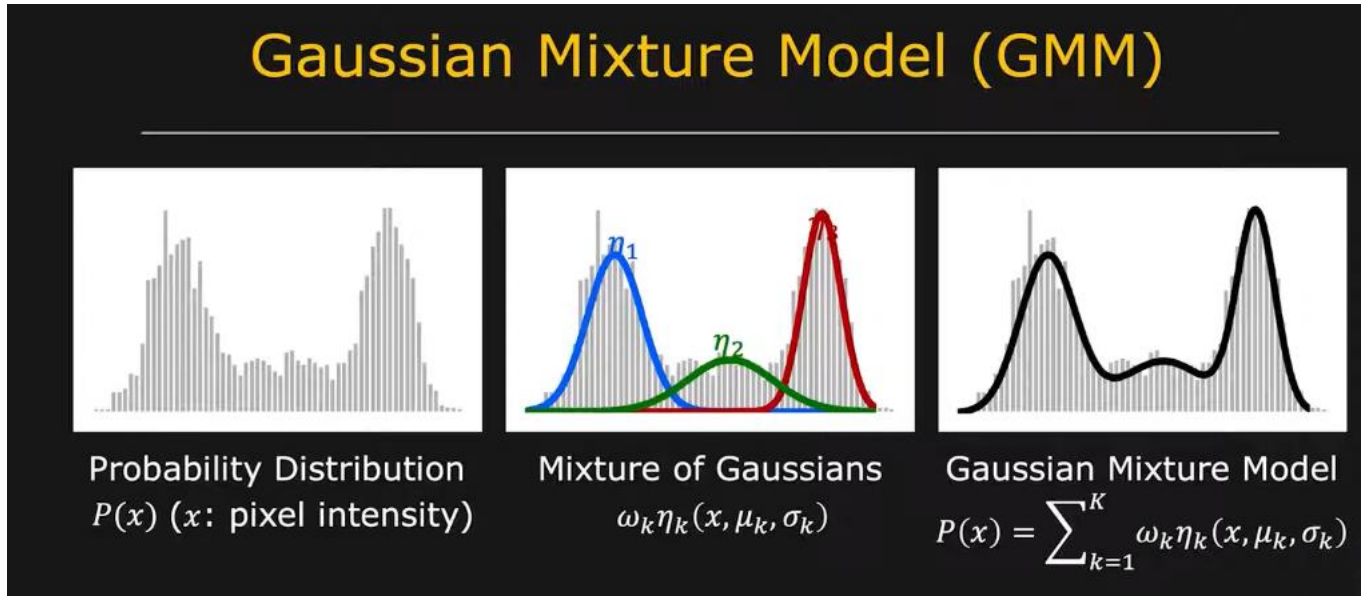
- The Expectation-Maximization (EM) algorithm is a statistical technique used in computing maximum likelihood estimates of parameters in models when the data are incomplete or have missing values.

1. Initialization: Start with initial guesses for the parameters.
2. Repeat until convergence:
    1. E-step: Estimate the expected value of the missing data given the observed data and the current parameter estimates.
    2. M-step: Maximize the expected log-likelihood obtained in the E-step to update the parameter estimates.
3. Termination: The algorithm stops when the changes in parameters are below a certain threshold.

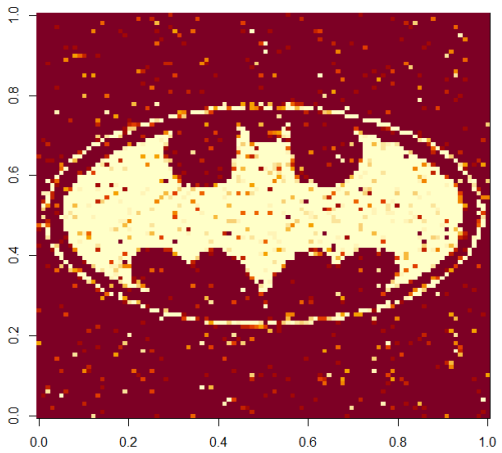


Gaussian Mixture Model

- Gaussian Mixture Model (GMM) is a probabilistic model that aims to find underlying Gaussian distributions (also called components) in a dataset.
- Gaussian Components: Each component represents a cluster or a group in the data. Every Gaussian is defined by parameters such as mean and variance (in univariate) or covariance matrix (in multivariate).
- Mixing Coefficients: These are the probabilities that a randomly selected observation comes from one of the Gaussian components. They essentially represent the proportion of each component in the overall distribution.

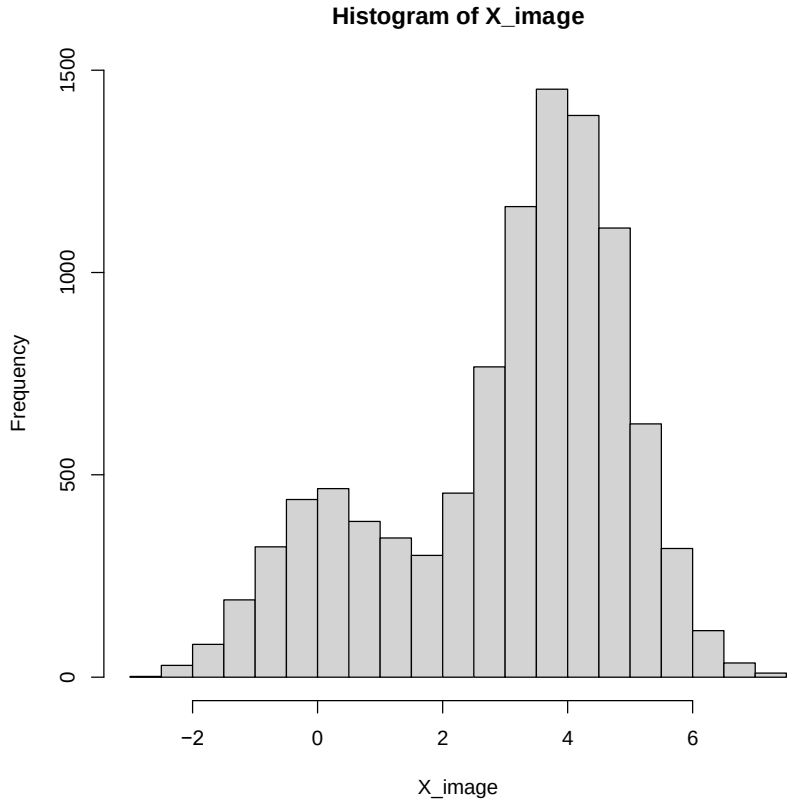


Parameters evolution



Initial Values:
μ1 =4.0
μ2 =2.0
V1 =2.15
V2 =2.15
π1 =0.75

Final Values:
μ1 =3.9369
μ2 =0.1667
V1 =1.0421
V2 =1.0104
π1 =0.7584



Results of EM