# ICA Demo

## Examples of Independent Component Analysis

STAT 32950-24620

Spring 2025 (wk8)



## Independent Component Analysis objectives

- Model the observed vector $\boldsymbol{X}$ as linear transformation of latent vector $\boldsymbol{S}$, which has **independent, non-Gaussian components**.
- Recover the components of $\boldsymbol{S}$.

The population model for the noiseless, equal dimensional case:

$$\boldsymbol{X} = A\boldsymbol{S}$$

$$\begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_p \end{bmatrix} = A_{p\times p} \begin{bmatrix} S_1 \\ \vdots \\ S_i \\ \vdots \\ S_p \end{bmatrix}$$



## ICA properties

- The components of $\boldsymbol{S}$ are independent, thus the joint density of $\boldsymbol{S}$ can be written as the product $f(\boldsymbol{s}) = \prod_{i=1}^{p} f_i(\boldsymbol{s}_i)$.
- The components $S_i$ are **non-Gaussian**.
  - At most one Gaussian component is allowed in ICA.
  - If all components are Gaussian, ICA won't work (use PCA).
- The **mixing matrix** $A$ is invertible, **unmixing matrix** $A^{-1}$ exists.
- The goal is to recover the independent "signal" resources $\boldsymbol{S}$.

**The identifiability issues**

Similar to another latent variable model Factor Analysis (FA), Independent Component Analysis (ICA) lacks of identifiability.

The recovery of components of $\boldsymbol{S}$ is not unique.

- The variance of $\boldsymbol{S}$ can not be determined (Impose $Cov(\boldsymbol{S}) = \boldsymbol{I}_p$)
- The sign of $S_i$ can not be determined.
- The order of $S_i$ can not be determined.



## How to recover components $S_i$'s?

The desired independent components are linear combinations of the components of the original $\boldsymbol{X}$:

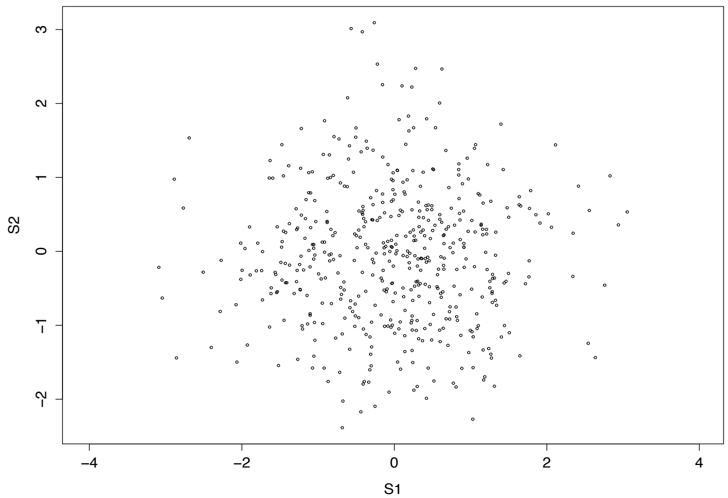
$$\boldsymbol{S} = \begin{bmatrix} S_1 \\ \vdots \\ S_i \\ \vdots \\ S_p \end{bmatrix} = A^{-1}\boldsymbol{X} = \begin{bmatrix} \boldsymbol{w}_1'\boldsymbol{X} \\ \vdots \\ \boldsymbol{w}_i'\boldsymbol{X} \\ \vdots \\ \boldsymbol{w}_p\boldsymbol{X} \end{bmatrix}$$

The idea of ICA is selecting $\boldsymbol{w}_i$'s so that $\boldsymbol{w}_i'\boldsymbol{X}$ is as far away from normal distribution as possible.

Measures of non-Gaussianity: Skewness, Kurtosis, Entropy.

## Example 1 - Input sources

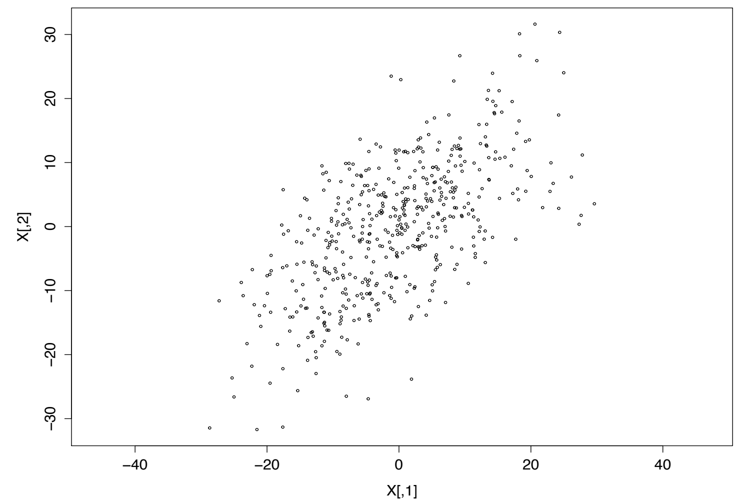
Example 1 — Normal inputs S with 2 independent components






## Example 1 - Observed data

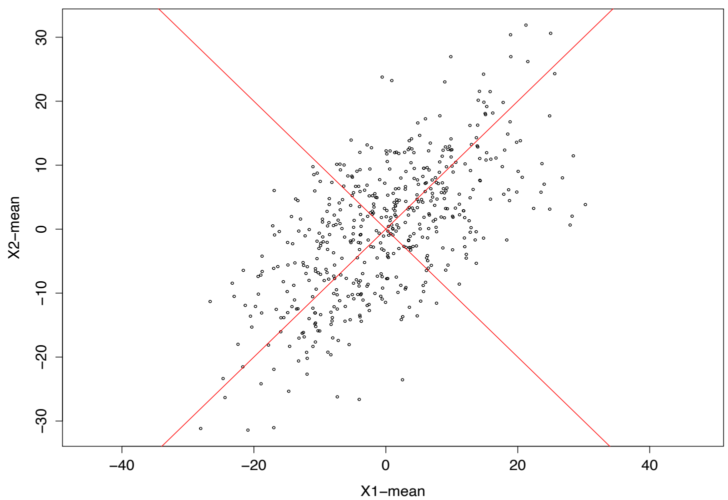
Example 1 — Observation data X = AS (inputs transformed)






## Example 1 - Recover sources by PCA

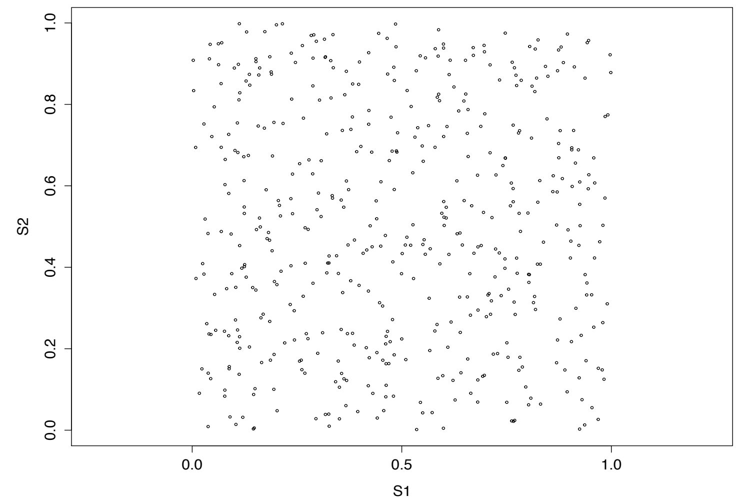
Example 1 — PCA on transformed data ("recovering" the inputs well)





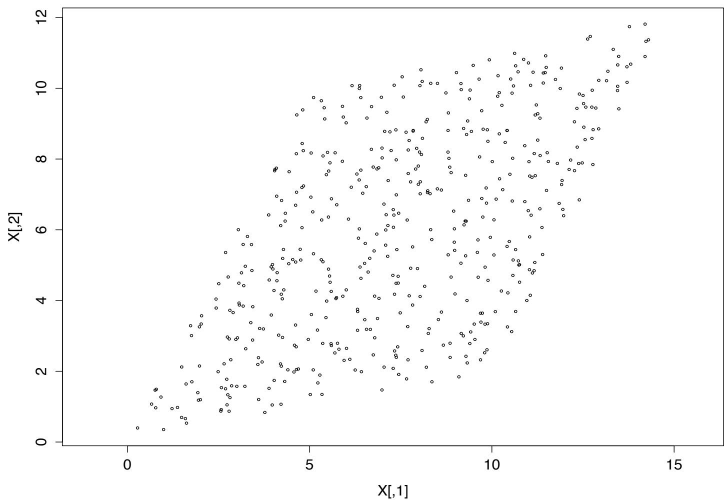

## Example 2 - Input sources

Example 2 — Uniform inputs S with two independent component



## Example 2 - Observed data

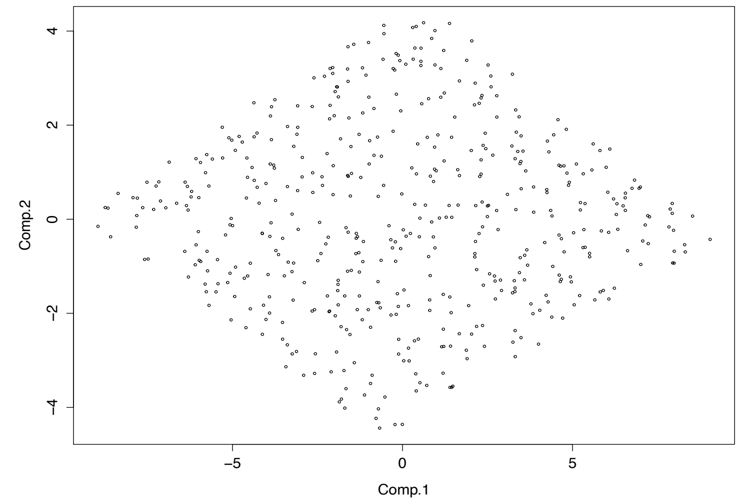
Example 2 — Observations of data X = AS (inputs transformed)






## Example 2 - Recover sources by PCA

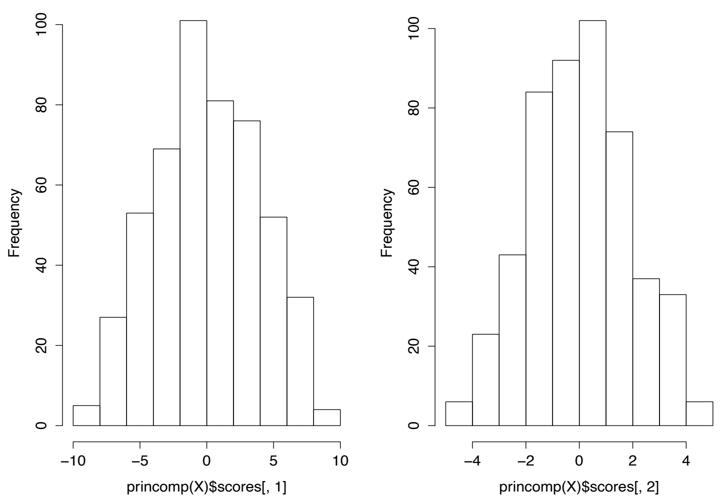
Example 2 — PCA on transformed data






## Example 2 - Recovered components by PCA

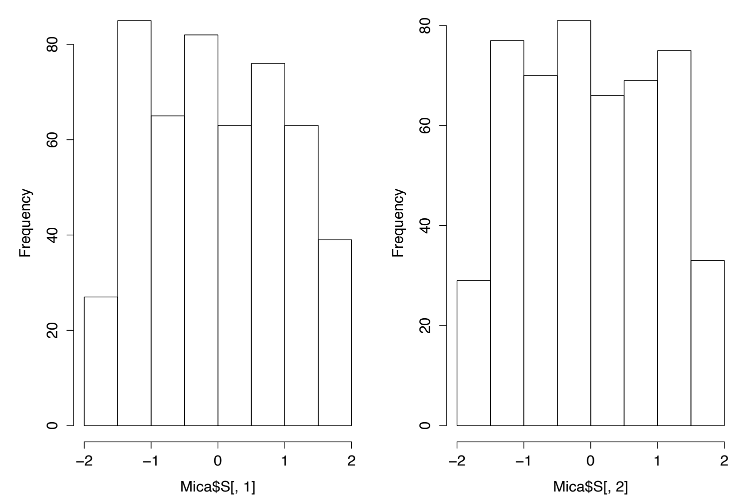
Example 2 — PCA components (Do they "recover" the inputs?)





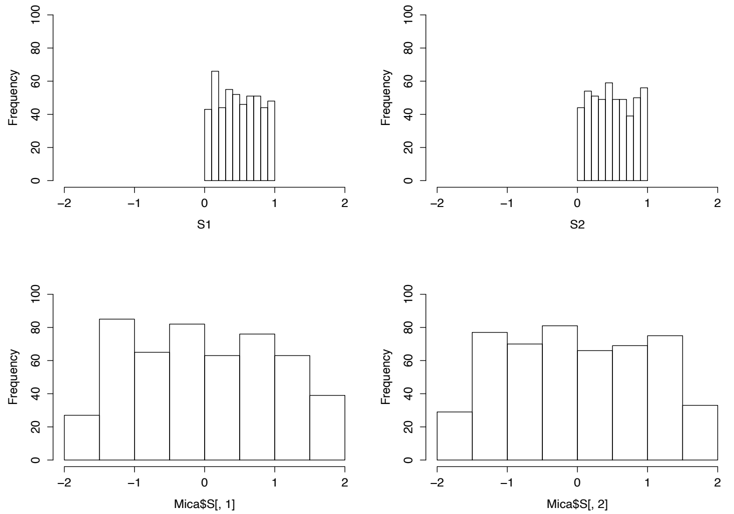

## Example 2 - Recovered components by ICA

Example 2 — ICA components (Does ICA recover the inputs?)



## Example 2 - Comparison of input and ICA recoveries

Example 2 — ICA input (top) vs ICA recoveries (bottom)






## Example 2 - Code

Example 2 — code (uniform inputs)

```
library(fastICA)

N=500
S1=runif(N)
S2=runif(N)
S=cbind(S1,S2)
# Mixing
A=matrix(c(5, 10, 10, 2),2,2)
X=S%*%A

#======= PCA =======#

Mpca = princomp(X,cor=T)

summary(Mpca,loading=T)
plot(X,asp=1,cex=.5)
plot(X-matrix(rep(1,1000),500,2)%*%diag(colMeans(X)),asp=1,cex=.5,xlab="X1-mean",ylab="X2-mean")
abline(0, Mpca$loading[1,1]/Mpca$loading[2,1],col=2)
abline(0, Mpca$loading[1,2]/Mpca$loading[2,2],col=2)
#plot(princomp(X)$scores[,1:2],cex = .5)
par(mfrow=c(1,2))
hist(princomp(X)$scores[,1],main="")
hist(princomp(X)$scores[,2],main="")
```



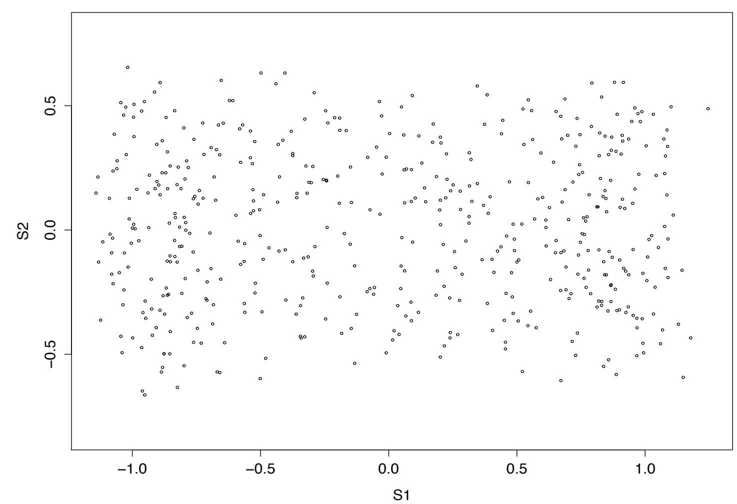
Example 2 — ICA part (uniform inputs); Correlation matrix comparisons

```
#======= ICA =======#
Mica=fastICA(X,2)

# ICA recovered components
hist(Mica$S[,1],main="")
hist(Mica$S[,2],main="")
#plot(Mica$S[,1],Mica$S[,2])
#plot(1:500,fastICA(X,2)$S[,1])
#plot(Mica$X)   # Data
#plot(Mica$X %*% Mica$K)  #PCA
#plot(Mica$S)
par(mfrow=c(2,2))
hist(S1,main="",xlim=c(-2,2),ylim=c(0,100))
hist(S2,main="",xlim=c(-2,2),ylim=c(0,100))
hist(Mica$S[,1],main="",,xlim=c(-2,2),ylim=c(0,100))
hist(Mica$S[,2],main="",,xlim=c(-2,2),ylim=c(0,100))
cor(S)
S1          S2
S1 1.000000000 0.006070144
S2 0.006070144 1.000000000
cor(X)
[,1]      [,2]
[1,] 1.0000000 0.6154696
[2,] 0.6154696 1.0000000

cor(Mica$S)
[,1]         [,2]
[1,] 1.000000e+00 1.997374e-16
[2,] 1.997374e-16 1.000000e+00

cor(princomp(X)$scores)
Comp.1        Comp.2
Comp.1  1.000000e+00 -9.495089e-16
Comp.2 -9.495089e-16  1.000000e+00
```



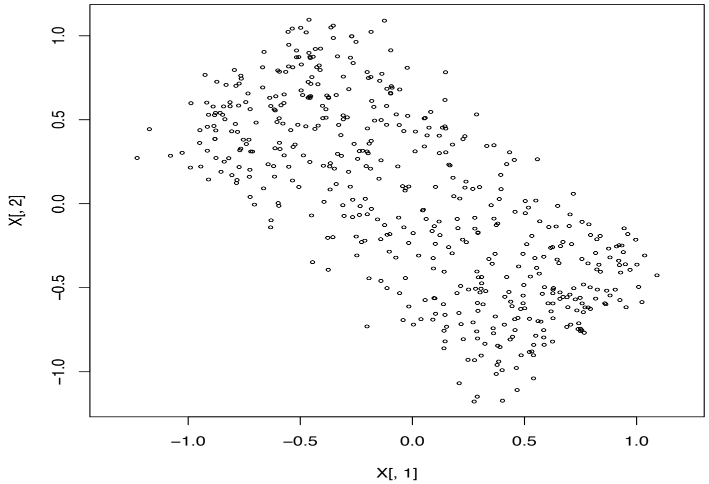
## Example 3 - Input sources

Example 3 — noisy signal inputs



## Example 3 - observed data

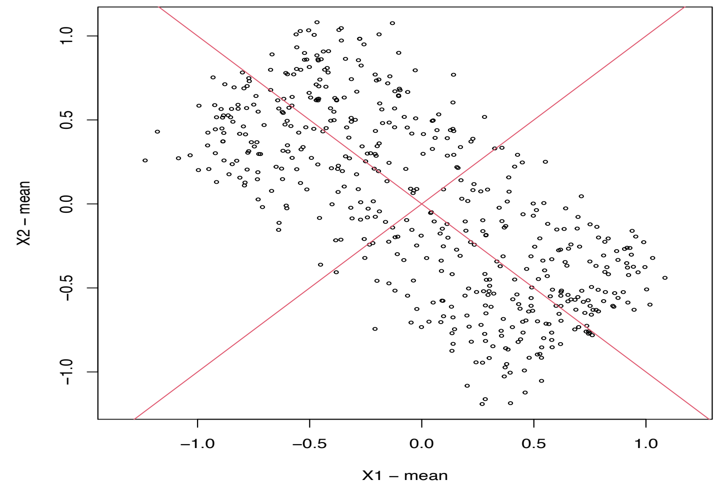
Example 3 — Data $\boldsymbol{X} = A\boldsymbol{S} + \epsilon$ (observed)






## Example 3 - Recover sources by ICA

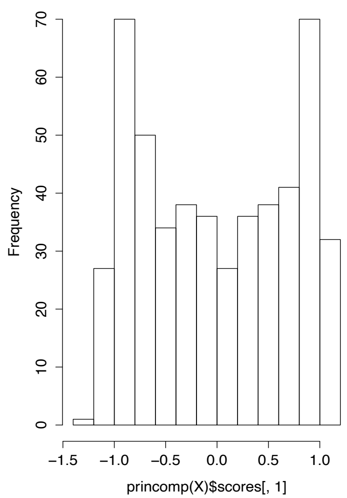
Example 3 — PCA on transformed signals

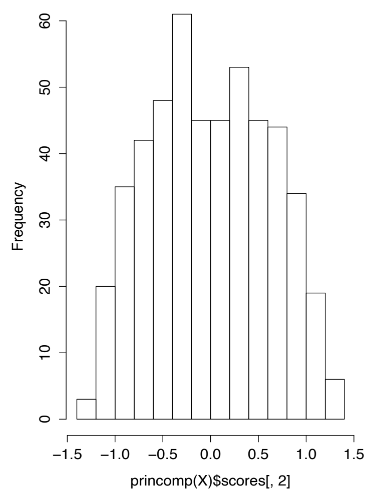





## Example 3 - Recovered components by PCA

Example 3 — PCA components

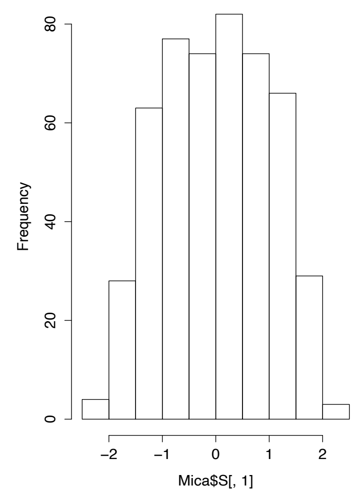


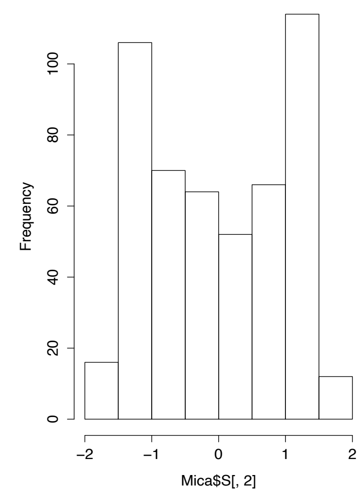






## Example 3 - Recovered components by ICA

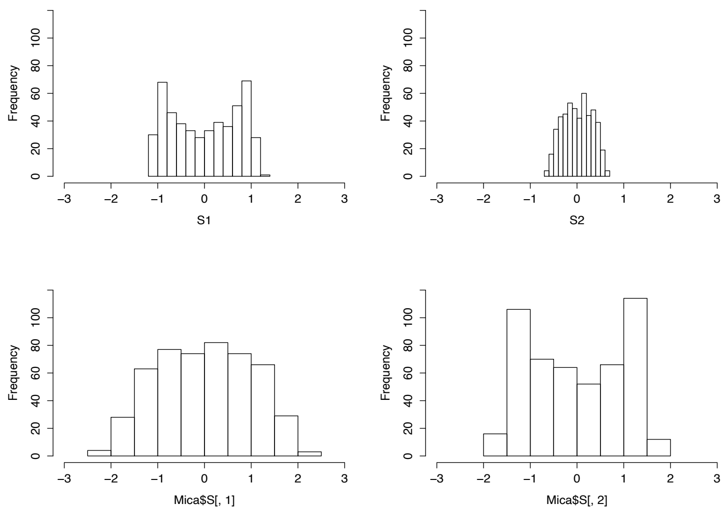
Example 3 — ICA components



## Example 3 - Comparison of input and ICA recoveries

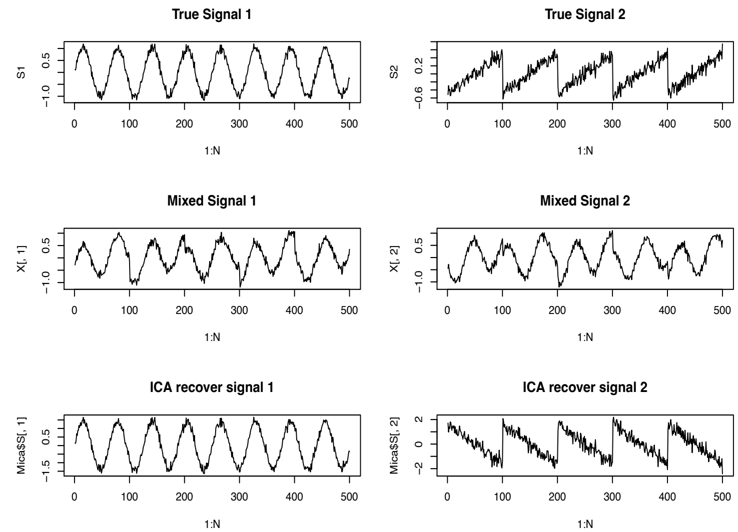
Example 3 — ICA inputs (top) vs recoveries (bottom)






## Example 3 - True signal, mixed input, recovery by ICA

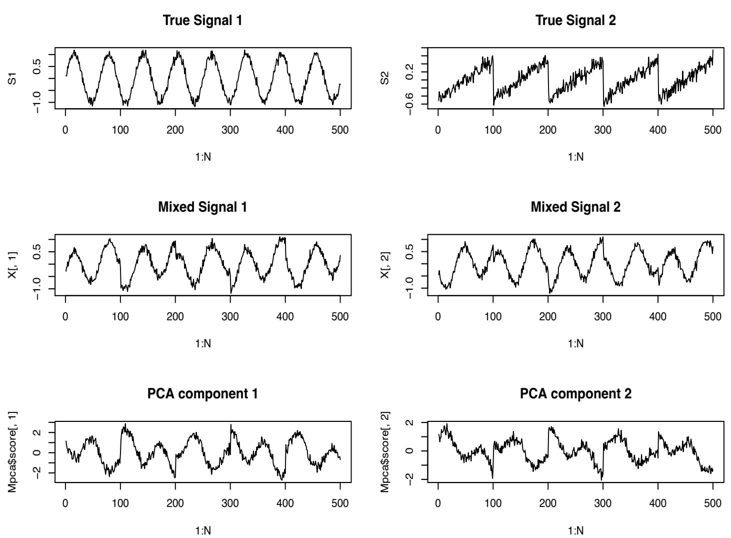
Ex 3 — True signal (top) vs Mixed input (mid) vs ICA recoveries (bottom)






## Example 3 - True signal, mixed input, recovery by PCA

Ex 3 — True signal (top) vs Mixed input (mid) vs PCA components (bottom)






## Example 3 - Code

Example 3 — partial code (noisy signal inputs)

```
S1=sin((1:500)/10)+.1*rnorm(N); S2=rep((((1:100)-50)/100),5)+.1*rnorm(N)
S=cbind(S1,S2)
A=matrix(c(1/sqrt(2), -1/sqrt(2), 1/sqrt(2), 1/sqrt(2)), ,2,2)
X=S%*%A

#====== PCA =======#
Mpca = princomp(X,cor=T)

plot(X-matrix(rep(1,2*N),N,2)%*%diag(colMeans(X)),asp=1,cex=.5,xlab="X1 - mean",ylab="X2 - mean")
abline(0, Mpca$loading[1,1]/Mpca$loading[2,1],col=2)
abline(0, Mpca$loading[1,2]/Mpca$loading[2,2],col=2)
par(mfrow=c(1,2))
hist(princomp(X)$scores[,1],main="")
hist(princomp(X)$scores[,2],main="")

#====== ICA =======#
Mica=fastICA(X,2)

par(mfrow=c(2,2))
plot(1:N,S1,cex=.5,type="l")
plot(1:N,S2,cex=.5,type="l")
plot(1:N,Mica$S[,1],cex=.5,type="l")
plot(1:N,Mica$S[,2],cex=.5,type="l")

par(mfrow=c(3,2))
plot(1:N,S1,cex=.5,type="l"); title("True Signal 1")
plot(1:N,S2,cex=.5,type="l"); title("True Signal 2")
plot(1:N,X[,1],cex=.5,type="l"); title("Mixed Signal 1")
plot(1:N,X[,2],cex=.5,type="l"); title("Mixed Signal 2")
plot(1:N,Mica$S[,1],cex=.5,type="l"); title("ICA recover signal 1")
plot(1:N,Mica$S[,2],cex=.5,type="l"); title("ICA recover signal 2")
```