

# Modèles Linéaires Appliqués / Régression Analyse Discriminante & Courbe ROC

Arthur Charpentier

UQAM

Hiver 2020 - COVID-19 # 6



## Analyse Discriminante de Fisher

Supposons que le but soit *simplement* de prédire une classe,  $\hat{y} \in \{0, 1\}$  (ou plus généralement  $\hat{y} \in \{a_1, a_2, \dots, a_J\}$ )

$$m^*(\mathbf{x}) = \operatorname{argmin}_{y \in \{0,1\}} \{\mathbb{P}[Y = y | \mathbf{X} = \mathbf{x}]\}$$

soit

$$m^*(\mathbf{x}) = \operatorname{argmin}_{y \in \{0,1\}} \left\{ \frac{\mathbb{P}[\mathbf{X} = \mathbf{x} | Y = y]}{\mathbb{P}[\mathbf{X} = \mathbf{x}]} \right\}$$

(où  $\mathbb{P}[\mathbf{X} = \mathbf{x}]$  devient  $f(\mathbf{x})$  dans le cas continu).

Si  $y$  prend deux valeurs – i.e.  $\{0, 1\}$

$$m^*(\mathbf{x}) = \begin{cases} 1 & \text{si } \mathbb{E}(Y | \mathbf{X} = \mathbf{x}) > \frac{1}{2} \\ 0 & \text{sinon} \end{cases}$$

# Analyse Discriminante de Fisher

L'ensemble

$$\mathcal{D}_S = \left\{ \mathbf{x} : \mathbb{E}(Y|\mathbf{X} = \mathbf{x}) = \frac{1}{2} \right\}$$

est appelé **frontière de décision**.

Supposons que  $\mathbf{X}|Y = 0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$  et  $\mathbf{X}|Y = 1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ , alors, si  $r_y^2$  est la distance de Mahalanobis de  $\mathbf{x}$  à  $\boldsymbol{\mu}_y$

$$r_y^2 = [\mathbf{x} - \boldsymbol{\mu}_y]^\top \boldsymbol{\Sigma}_y^{-1} [\mathbf{x} - \boldsymbol{\mu}_y] \text{ pour } y \in \{0, 1\},$$

$$m^*(\mathbf{x}) = \begin{cases} 1 & \text{si } r_1^2 < r_0^2 + 2 \log \frac{\mathbb{P}(Y = 1)}{\mathbb{P}(Y = 0)} + \log \frac{|\boldsymbol{\Sigma}_0|}{|\boldsymbol{\Sigma}_1|} \\ 0 & \text{sinon} \end{cases}$$

## Analyse Discriminante de Fisher

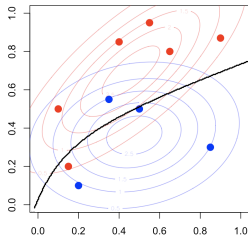
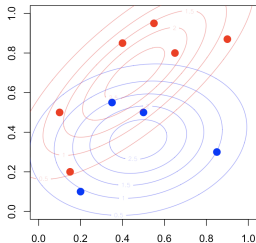
Soit  $\delta_y$  la fonction définie (pour  $y \in \{0, 1\}$ ) par

$$\delta_y(\mathbf{x}) = -\frac{1}{2} \log |\boldsymbol{\Sigma}_y| - \frac{1}{2} [\mathbf{x} - \boldsymbol{\mu}_y]^\top \boldsymbol{\Sigma}_y^{-1} [\mathbf{x} - \boldsymbol{\mu}_y] + \log \mathbb{P}(Y = y)$$

telle que la frontière de décision soit

$$\{\mathbf{x} \text{ tel que } \delta_0(\mathbf{x}) = \delta_1(\mathbf{x})\}$$

qui est **quadratique en  $\mathbf{x}$**

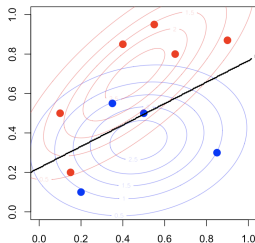
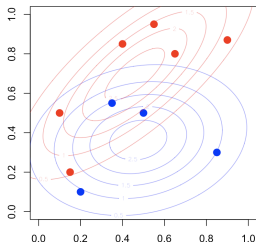


# Analyse Discriminante de Fisher

Fisher (1936) a rajouté hypothèse  $\Sigma_0 = \Sigma_1$ . Alors

$$\delta_y(\mathbf{x}) = \mathbf{x}^\top \Sigma^{-1} \mu_y - \frac{1}{2} \mu_y^\top \Sigma^{-1} \mu_y + \log \mathbb{P}(Y = y)$$

et la frontière de décision est **linéaire en  $\mathbf{x}$**



## Analyse Discriminante de Fisher

Si  $\mathbf{X}|Y = 0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})$  et  $\mathbf{X}|Y = 1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$  alors

$$\log \frac{\mathbb{P}(Y = 1|\mathbf{X} = \mathbf{x})}{\mathbb{P}(Y = 0|\mathbf{X} = \mathbf{x})}$$

est égal à

$$\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}[\boldsymbol{\mu}_y] - \frac{1}{2}[\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0]^\top \boldsymbol{\Sigma}^{-1}[\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0] + \log \frac{\mathbb{P}(Y = 1)}{\mathbb{P}(Y = 0)}$$

qui est linéaire en  $\mathbf{x}$ , autrement dit

$$\log \frac{\mathbb{P}(Y = 1|\mathbf{X} = \mathbf{x})}{\mathbb{P}(Y = 0|\mathbf{X} = \mathbf{x})} = \mathbf{x}^\top \boldsymbol{\beta}$$

ce qui rappelle la régression logistique...

## Test et Décision

$y \in \{0, 1\}$  et on va construire  $\hat{y} \in \{0, 1\}$  – via  $\hat{y} = \mathbf{1}(\hat{p} > s)$

		vérité	
		–	+
décision	–	true negative	false negative
	+	false positive	true positive

		vérité	
		–	+
décision	–	bonne décision	type 2 error
	+	type 1 error	bonne décision

On a un **tradeoff** entre les types d'erreurs, cf **base rate fallacy**.

Cf théorie des tests, où on teste  $H_0$  – qui peut être valide (ou pas) – et on doit prendre une décision : **rejeter  $H_0$**  ou **accepter  $H_0$** .

## Test et Décision

$$\text{Prevalence } \frac{200}{10,000} = 2\%$$

$$\text{Specificity } \frac{9,751}{9,800} = 99.5\%$$

$$\text{Sensitivity } \frac{100}{200} = 50\%$$

$$\text{Positive Predictive Value } \frac{100}{149} \sim 67\%$$

$$\text{Specificity } \frac{9,310}{9,800} = 95\%$$

$$\text{Positive Predictive Value } \frac{100}{590} \sim 17\%$$

	- well	+ disease	
-	9,751	100	9,851
+	49	100	149
	9,800	200	10,000

	- well	+ disease	
-	9,310	100	9,410
+	490	100	590
	9,800	200	10,000

Cf [Wainer & Savage \(2008\)](#) (until proven guilty: False positives and the war on terror)



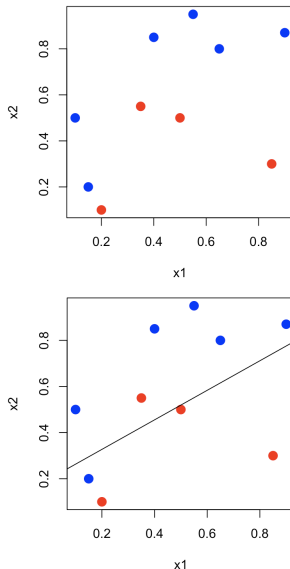
## Courbe ROC

```
1 > x1 = c(.4,.55,.65,.9,.1,.35,  
2 .5,.15,.2,.85)  
3 > x2 = c(.85,.95,.8,.87,.5,.55,  
4 .5,.2,.1,.3)  
5 > y = c(1,1,1,1,1,0,0,1,0,0)  
6 > df = data.frame(x1=x1,x2=x2,  
7 y=as.factor(y))  
8 > plot(x1,x2,col=1+y)  
9 > reg = glm(y~x1+x2,data=df,  
10 family=binomial(link = "logit"))  
11 > b = coefficients(reg)  
12 > abline(a=-b[1]/b[3],b=-b[2]/b[3])
```

$\mathbb{P}[Y = 1 | \mathbf{X} = \mathbf{x}] = \mathbb{P}[Y = 0 | \mathbf{X} = \mathbf{x}]$  si

$$e^{\mathbf{x}^\top \boldsymbol{\beta}} = 1 \text{ ou } \mathbf{x}^\top \boldsymbol{\beta} = 0$$

soit ici  $\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$ .

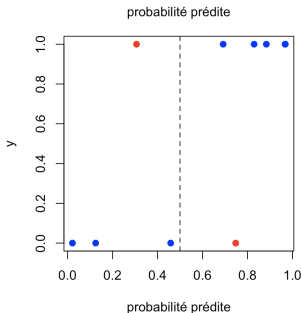
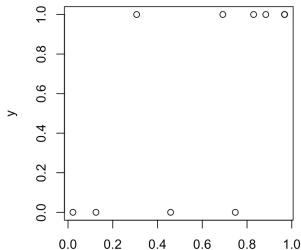


## Courbe ROC

```
1 > Y = df$y
2 > S = predict(reg,type="response")
3 > plot(S,y)
4 > seuil = .5
5 > Yhat = (S>seuil)*1
6 > plot(S,y,col=1+(y==Yhat))
7 > abline(v=seuil,lty=2)
```

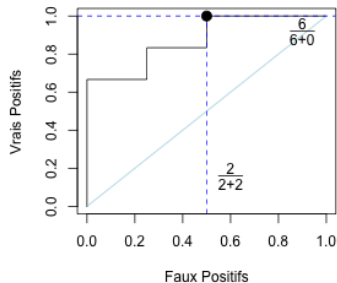
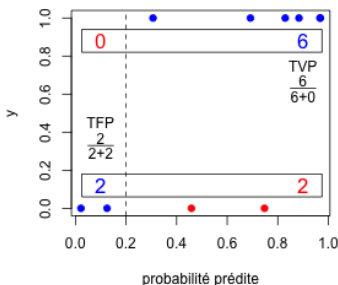
$$\hat{y}_i = \begin{cases} 1 & \text{si } \hat{p}_i > \text{seuil} \\ 0 & \text{si } \hat{p}_i \leq \text{seuil} \end{cases}$$

```
1 > table(Yhat,Y)
2     Y
3 Yhat 0 1
4     0 3 1
5     1 1 5
```



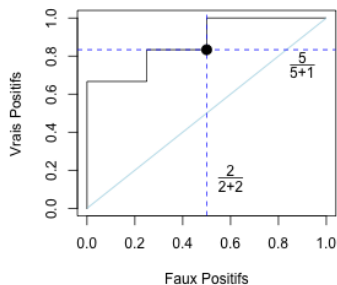
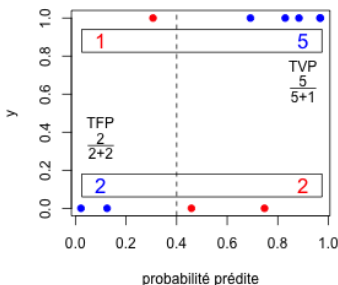
## Courbe ROC

$$FPR = \frac{\mathbb{P}[y = 0, \hat{y} = 1]}{\mathbb{P}[y = 0]} \text{ et } TPR = \frac{\mathbb{P}[y = 1, \hat{y} = 1]}{\mathbb{P}[y = 1]}$$



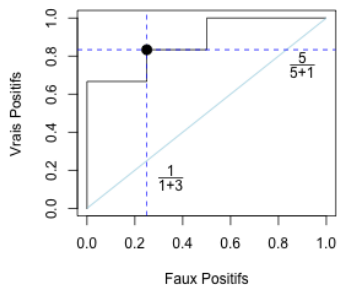
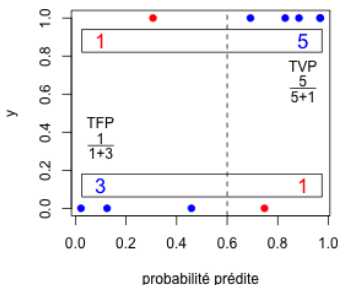
## Courbe ROC

$$FPR = \frac{\mathbb{P}[y = 0, \hat{y} = 1]}{\mathbb{P}[y = 0]} \text{ et } TPR = \frac{\mathbb{P}[y = 1, \hat{y} = 1]}{\mathbb{P}[y = 1]}$$



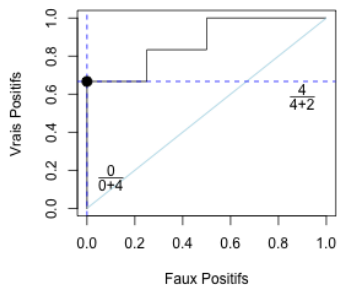
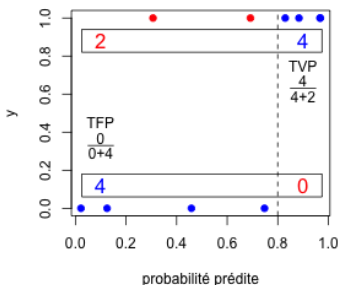
## Courbe ROC

$$FPR = \frac{\mathbb{P}[y = 0, \hat{y} = 1]}{\mathbb{P}[y = 0]} \text{ et } TPR = \frac{\mathbb{P}[y = 1, \hat{y} = 1]}{\mathbb{P}[y = 1]}$$



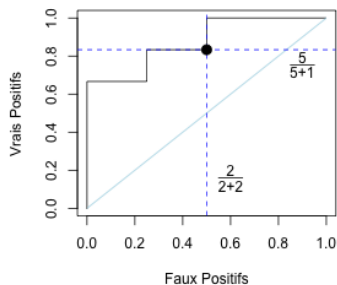
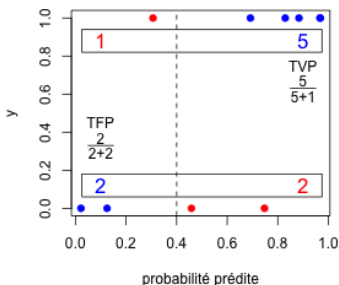
## Courbe ROC

$$FPR = \frac{\mathbb{P}[y = 0, \hat{y} = 1]}{\mathbb{P}[y = 0]} \text{ et } TPR = \frac{\mathbb{P}[y = 1, \hat{y} = 1]}{\mathbb{P}[y = 1]}$$



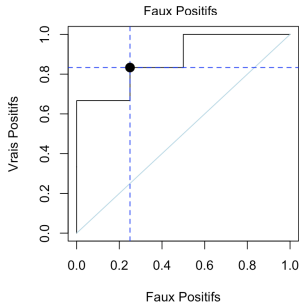
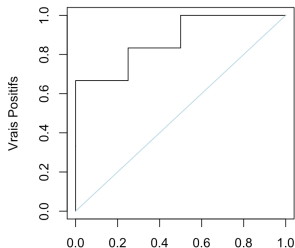
## Courbe ROC

$$FPR = \frac{\mathbb{P}[y = 0, \hat{y} = 1]}{\mathbb{P}[y = 0]} \text{ et } TPR = \frac{\mathbb{P}[y = 1, \hat{y} = 1]}{\mathbb{P}[y = 1]}$$



## Courbe ROC

```
1 > roc.curve=function(s){
2   Ps=(S>s)*1
3   FP=sum((Ps==1)*(Y==0))/sum(Y==0)
4   TP=sum((Ps==1)*(Y==1))/sum(Y==1)
5   vect=c(FP,TP)
6   names(vect)=c("FPR","TPR")
7   return(vect) }
8 > u = seq(0,1,length=251)
9 > V = Vectorize(roc.curve)(u)
10 > plot(t(V),type="s")
11 > table(Yhat,Y)
12     Y
13 Yhat 0 1
14     0 3 1
15     1 1 5
16 > sum((Yhat)*(Y==0))/sum(Y==0)
17 [1] 0.25
18 > sum((Yhat==1)*(Y==1))/sum(Y==1)
19 [1] 0.8333333
```

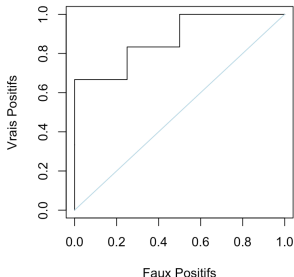




## Courbe ROC

De nombreux packages permettent de tracer des courbes ROC, dont ROCR (ou pROC, plotROC)

```
1 > library(ROCR)
2 > pred = prediction(S,Y)
3 > plot(performance(pred,"tpr","fpr"))
4 > auc.perf = performance(pred,
5   measure = "auc")
6 > auc.perf@y.values[[1]]
[1] 0.875
```

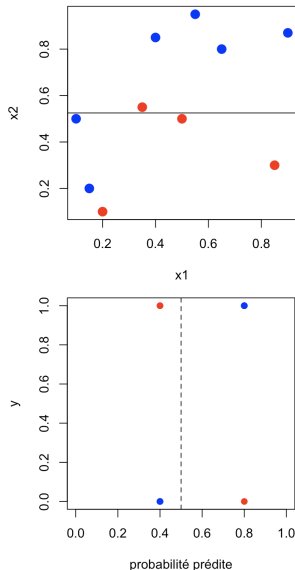


L'**AUC** – aire sous la courbe – donne une idée de la qualité de la classification.

## Courbe ROC

Régression de  $y$  sur  $1_{[.525, \infty)}(x_2)$

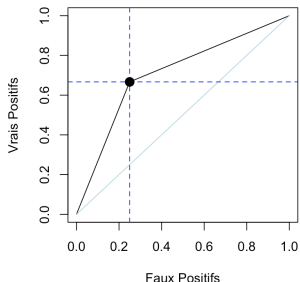
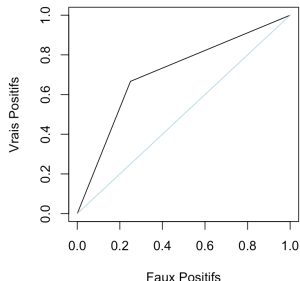
```
1 > reg = glm(y~I(x2> .525),data=df,  
2 family=binomial(link = "logit"))  
3 > abline(h=.525)  
4 > Y = df$y  
5 > S = predict(reg,type="response")  
6 > plot(S,y,xlim=0:1)  
7 > seuil = .5  
8 > Yhat = (S>seuil)*1  
9 > table(Yhat,Y)  
10      Y  
11 Yhat 0 1  
12    0 3 |\aftergroup\rcrd|2|\aftergroup\blackcolor|  
13    1 |\aftergroup\rcrd|4|\aftergroup\blackcolor| 5
```



## Courbe ROC

La courbe ROC est linéaire par morceaux

```
1 > pred = prediction(S,Y)
2 > plot(performance(pred,"tpr","fpr"))
3 > table(Yhat,Y)
4      Y
5 Yhat 0 1
6      0 3 2
7      1 1 4
8 > auc.perf = performance(pred,
9     measure = "auc")
10 > auc.perf@y.values[[1]]
[1] 0.7083333
```

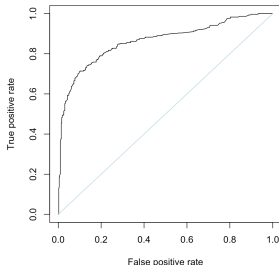


# Survie des Passagers du Titanic

y : indicatrice de survie d'un passager du Titanic

```
1 > loc = "http://freakonometrics.free.fr/titanic.RData"  
2 > download.file(loc, "titanic.RData")  
3 > load("titanic.RData")  
4 > base = base[!is.na(base$Age),1:7]  
5 > reg = glm(Survived ~ Sex+poly(Age,3)+Pclass+SibSp,  
6   family = "binomial", data = base)
```

```
1 > library(ROCR)  
2 > Y = base$Survived  
3 > S = predict(reg,type="response")  
4 > pred = prediction(S,Y)  
5 > plot(performance(pred,"tpr","fpr"))  
6 > performance(pred, measure = "auc")  
7   @y.values[[1]]  
8 [1] 0.8627358
```



# Survie des Passagers du Titanic

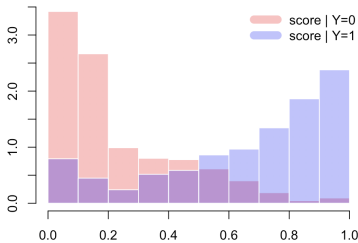
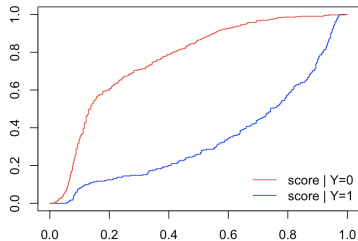
Kolmogorov-Smirnov (KS) :  
Comparer les distributions de  
( $S|Y = 0$ ) et ( $S|Y = 1$ )

$$d = \sup_{x \in [0,1]} \{|\hat{F}_1(x) - \hat{F}_0(x)|\}$$

où

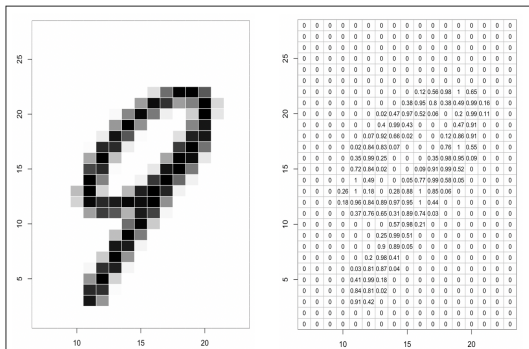
$$\hat{F}_1(x) = \frac{1}{n_1} \sum_{i:y_i=1} \mathbf{1}(s_i \leq x)$$

$$\hat{F}_0(x) = \frac{1}{n_0} \sum_{i:y_i=0} \mathbf{1}(s_i \leq x)$$



# Classification... sur des images

$(y_i, \mathbf{x}_i)$ , où  $y \in \{0, 1, 2, 3, \dots, 9\}$  et  $\mathbf{x}_i \in \mathcal{M}_{28,28} = [0, 1]^{28 \times 28}$



$\mathbf{x}_i \in \mathcal{M}_{28,28}$

$y_i \in \{0, 1, \dots, 9\}$

```
1 > library(keras)
2 > mnist = dataset_mnist()
```

## Classification... sur des images

Here  $\{(y_i, \mathbf{x}_i)\}$  with  $y_i = "3"$  and  $\mathbf{x}_i \in [0, 1]^{28 \times 28}$

```
1 > library(keras)
2 > mnist = dataset_mnist()
3 > n = 1000
4 > V = mnist$train$x[1:n,,]
5 > MV = NULL
6 for(i in 1:n) MV=cbind(MV,as.vector(V[i,,]))
7 > MV = t(MV)
8 > df = data.frame(y=mnist$train$y[1:n],x=MV)
```

# Classification... sur des images

Peut-on reconnaître les '1' ?

```
1 > reg=glm((y==1)~.,data=df,family=binomial)
2 Warning messages:
3 1: glm.fit: algorithm did not converge
4 2: glm.fit: fitted probabilities numerically 0 or 1
   occurred
```

Problème numérique !

On a  $k = 784$  variables explicatives... il faut réduire la dimension !

```
1 > library(factoextra)
2 > pca=prcomp(MV)
3 > res.ind = get_pca_ind(pca)
4 > PTS = res.ind$coord
5 > k=3
6 > dfpca = data.frame(y=mnist$train$y[1:n],x=PTS[,1:k])
7 > reg1 = glm((y==1)~.,data=dfpca,family=binomial)
8 > reg8 = glm((y==8)~.,data=dfpca,family=binomial)
```



# Classification... sur des images

On essaye de reconnaître les '1' et les '8'

```
1 > library(ROCR)
2 > Y1 = as.numeric((df$y == 1)*1)
3 > S1 = predict(reg1,type="response")
4 > pred1 = prediction(S1,Y1)
5 > plot(performance(pred1,"tpr","fpr"))
6 > Y8 = as.numeric((df$y == 8)*1)
7 > S8 = predict(reg8,type="response")
8 > pred8 = prediction(S8,Y8)
9 > plot(performance(pred8,"tpr","fpr"))
```

On utilise ici seulement  
les 3 premières composantes principales

