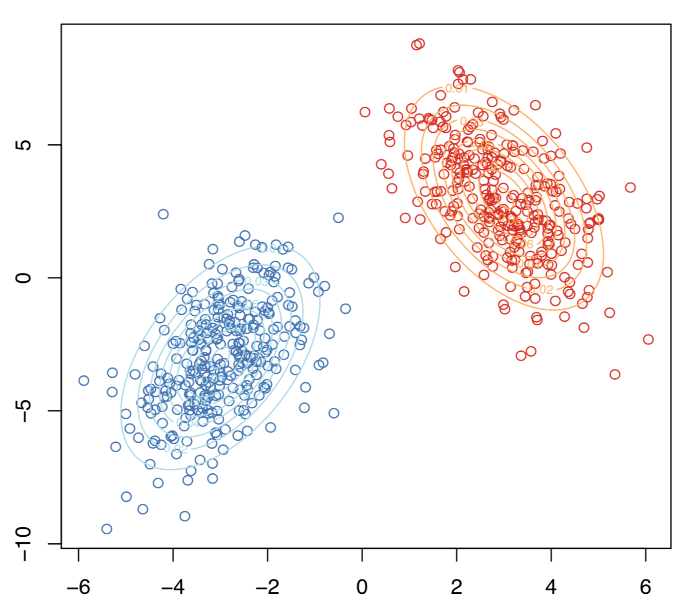
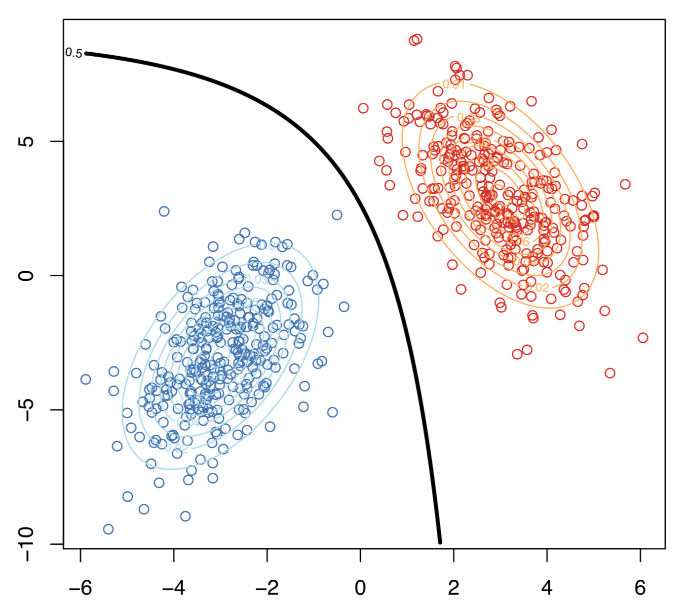
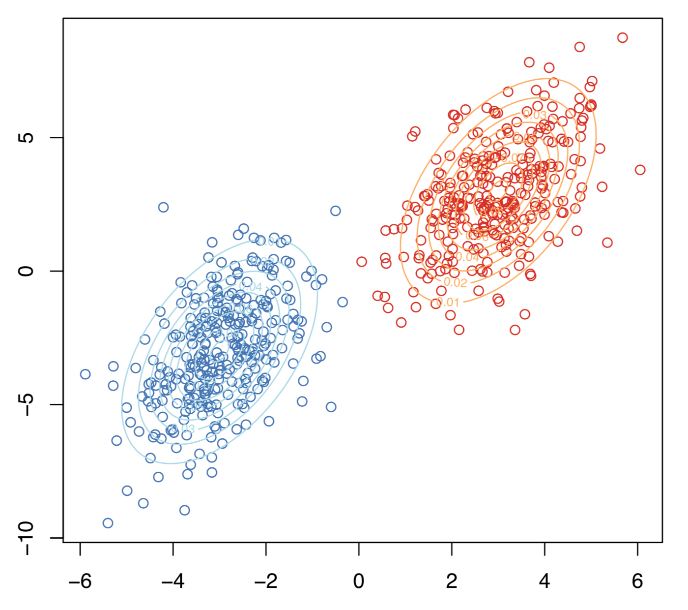
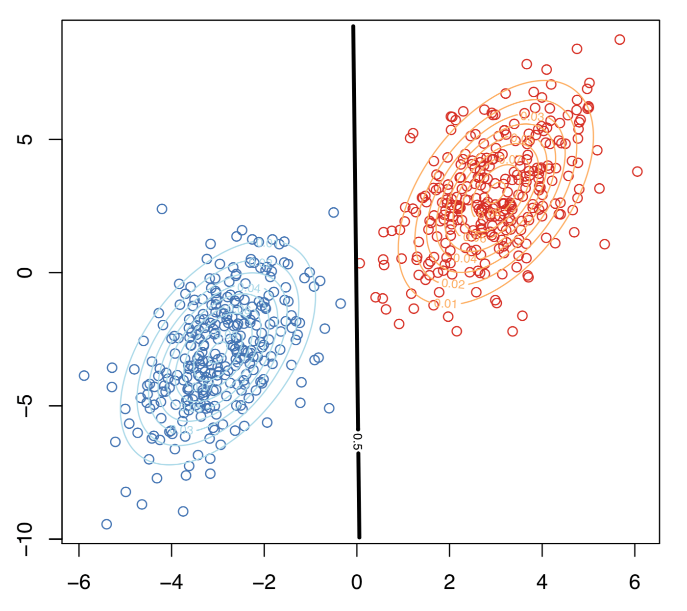
**Bayes (naive) classifier**

Consider the follwing naive classification rulem⋆(x)=argminy{P[Y=y∣X=x]}m^\star(\mathbf{x})=\text{argmin}\_y\{\mathbb{P}[Y=y\vert\mathbf{X}=\mathbf{x}]\}m⋆(x)=argminy​{P[Y=y∣X=x]}orm⋆(x)=argminy{P[X=x∣Y=y]P[X=x]}m^\star(\mathbf{x})=\text{argmin}\_y\left\{\frac{\mathbb{P}[\mathbf{X}=\mathbf{x}\vert Y=y]}{\mathbb{P}[\mathbf{X}=\mathbf{x}]}\right\}m⋆(x)=argminy​{P[X=x]P[X=x∣Y=y]​}(where P[X=x]\mathbb{P}[\mathbf{X}=\mathbf{x}]P[X=x] is the density in the continuous case).

In the case where yyy takes two values, that will be standard {0,1}\{0,1\}{0,1} here, one can rewrite the later asm⋆(x)={1 if E(Y∣X=x)>120 otherwisem^\star(\mathbf{x})=\begin{cases}1\text{ if }\mathbb{E}(Y\vert \mathbf{X}=\mathbf{x})>\displaystyle{\frac{1}{2}}\\0\text{ otherwise}\end{cases}m⋆(x)=⎩⎪⎨⎪⎧​1 if E(Y∣X=x)>21​0 otherwise​and the setDS={x,E(Y∣X=x)=12}\mathcal{D}\_S =\left\{\mathbf{x},\mathbb{E}(Y\vert \mathbf{X}=\mathbf{x})=\frac{1}{2}\right\}DS​={x,E(Y∣X=x)=21​}is called the decision boundary.

Assume thatX∣Y=0∼N(μ0,Σ)\mathbf{X}\vert Y=0\sim\mathcal{N}(\mathbf{\mu}\_0,\mathbf{\Sigma})X∣Y=0∼N(μ0​,Σ)andX∣Y=1∼N(μ1,Σ)\mathbf{X}\vert Y=1\sim\mathcal{N}(\mathbf{\mu}\_1,\mathbf{\Sigma})X∣Y=1∼N(μ1​,Σ)then explicit expressions can be derived.m⋆(x)={1 if r12<r02+2log⁡P(Y=1)P(Y=0)+log⁡∣Σ0∣∣Σ1∣0 otherwisem^\star(\mathbf{x})=\begin{cases}1\text{ if }r\_1^2< r\_0^2+2\displaystyle{\log\frac{\mathbb{P}(Y=1)}{\mathbb{P}(Y=0)}+\log\frac{\vert\mathbf{\Sigma}\_0\vert}{\vert\mathbf{\Sigma}\_1\vert}}\\0\text{ otherwise}\end{cases}m⋆(x)=⎩⎪⎨⎪⎧​1 if r12​<r02​+2logP(Y=0)P(Y=1)​+log∣Σ1​∣∣Σ0​∣​0 otherwise​where ry2r\_y^2ry2​ is the Manalahobis distance, ry2=[X−μy]TΣy−1[X−μy]r\_y^2 = [\mathbf{X}-\mathbf{\mu}\_y]^{\text{{T}}}\mathbf{\Sigma}\_y^{-1}[\mathbf{X}-\mathbf{\mu}\_y]ry2​=[X−μy​]TΣy−1​[X−μy​]

Let δy\delta\_yδy​be defined asδy(x)=−12log⁡∣Σy∣−12[x−μy]TΣy−1[x−μy]+log⁡P(Y=y)\delta\_y(\mathbf{x})=-\frac{1}{2}\log\vert\mathbf{\Sigma}\_y\vert-\frac{1}{2}[{\color{blue}{\mathbf{x}}}-\mathbf{\mu}\_y]^{\text{{T}}}\mathbf{\Sigma}\_y^{-1}[{\color{blue}{\mathbf{x}}}-\mathbf{\mu}\_y]+\log\mathbb{P}(Y=y)δy​(x)=−21​log∣Σy​∣−21​[x−μy​]TΣy−1​[x−μy​]+logP(Y=y)the decision boundary of this classifier is {x such that δ0(x)=δ1(x)}\{\mathbf{x}\text{ such that }\delta\_0(\mathbf{x})=\delta\_1(\mathbf{x})\}{x such that δ0​(x)=δ1​(x)}which is quadratic in x{\color{blue}{\mathbf{x}}}x. This is the quadratic discriminant analysis. This can be visualized bellow.  
  
The decision boundary is here  
  
But that can’t be the linear discriminant analysis, right? I mean, the frontier is not linear… Actually, in Fisher’s [seminal paper](https://digital.library.adelaide.edu.au/dspace/bitstream/2440/15227/1/138.pdf), it was assumed that Σ0=Σ1\mathbf{\Sigma}\_0=\mathbf{\Sigma}\_1Σ0​=Σ1​.

In that case, actually, δy(x)=xTΣ−1μy−12μyTΣ−1μy+log⁡P(Y=y)\delta\_y(\mathbf{x})={\color{blue}{\mathbf{x}}}^{\text{T}}\mathbf{\Sigma}^{-1}\mathbf{\mu}\_y-\frac{1}{2}\mathbf{\mu}\_y^{\text{T}}\mathbf{\Sigma}^{-1}\mathbf{\mu}\_y+\log\mathbb{P}(Y=y)δy​(x)=xTΣ−1μy​−21​μyT​Σ−1μy​+logP(Y=y) and the decision frontier is now linear in x{\color{blue}{\mathbf{x}}}x. This is the linear discriminant analysis. This can be visualized bellow  
  
Here the two samples have the same variance matrix and the frontier is  


**Link with the logistic regression**

Assume as previously thatX∣Y=0∼N(μ0,Σ)\mathbf{X}\vert Y=0\sim\mathcal{N}(\mathbf{\mu}\_0,\mathbf{\Sigma})X∣Y=0∼N(μ0​,Σ)andX∣Y=1∼N(μ1,Σ)\mathbf{X}\vert Y=1\sim\mathcal{N}(\mathbf{\mu}\_1,\mathbf{\Sigma})X∣Y=1∼N(μ1​,Σ)thenlog⁡P(Y=1∣X=x)P(Y=0∣X=x)\log\frac{\mathbb{P}(Y=1\vert \mathbf{X}=\mathbf{x})}{\mathbb{P}(Y=0\vert \mathbf{X}=\mathbf{x})}logP(Y=0∣X=x)P(Y=1∣X=x)​is equal to xTΣ−1[μy]−12[μ1−μ0]TΣ−1[μ1−μ0]+log⁡P(Y=1)P(Y=0)\mathbf{x}^{\text{{T}}}\mathbf{\Sigma}^{-1}[\mathbf{\mu}\_y]-\frac{1}{2}[\mathbf{\mu}\_1-\mathbf{\mu}\_0]^{\text{{T}}}\mathbf{\Sigma}^{-1}[\mathbf{\mu}\_1-\mathbf{\mu}\_0]+\log\frac{\mathbb{P}(Y=1)}{\mathbb{P}(Y=0)}xTΣ−1[μy​]−21​[μ1​−μ0​]TΣ−1[μ1​−μ0​]+logP(Y=0)P(Y=1)​which is linear in x\mathbf{x}xlog⁡P(Y=1∣X=x)P(Y=0∣X=x)=xTβ\log\frac{\mathbb{P}(Y=1\vert \mathbf{X}=\mathbf{x})}{\mathbb{P}(Y=0\vert \mathbf{X}=\mathbf{x})}=\mathbf{x}^{\text{{T}}}\mathbf{\beta}logP(Y=0∣X=x)P(Y=1∣X=x)​=xTβHence, when each groups have Gaussian distributions with identical variance matrix, then LDA and the logistic regression lead to the same classification rule.

Observe furthermore that the slope is proportional to Σ−1[μ1−μ0]\mathbf{\Sigma}^{-1}[\mathbf{\mu}\_1-\mathbf{\mu}\_0]Σ−1[μ1​−μ0​], as stated in Fisher’s article. But to obtain such a relationship, he observe that the ratio of between and within variances (in the two groups) wasvariance betweenvariance within=[ωμ1−ωμ0]2ωTΣ1ω+ωTΣ0ω\frac{\text{variance between}}{\text{variance within}}=\frac{[\mathbf{\omega}\mathbf{\mu}\_1-\mathbf{\omega}\mathbf{\mu}\_0]^2}{\mathbf{\omega}^{\text{T}}\mathbf{\Sigma}\_1\mathbf{\omega}+\mathbf{\omega}^{\text{T}}\mathbf{\Sigma}\_0\mathbf{\omega}}variance withinvariance between​=ωTΣ1​ω+ωTΣ0​ω[ωμ1​−ωμ0​]2​which is maximal when ω\mathbf{\omega}ω is proportional to Σ−1[μ1−μ0]\mathbf{\Sigma}^{-1}[\mathbf{\mu}\_1-\mathbf{\mu}\_0]Σ−1[μ1​−μ0​], when Σ0=Σ1\mathbf{\Sigma}\_0=\mathbf{\Sigma}\_1Σ0​=Σ1​.

**Homebrew linear discriminant analysis**

To compute vector ω\mathbf{\omega}ω

|  |
| --- |
| m0 = **apply**(myocarde[myocarde$PRONO=="0",1:7],2,**mean**)  m1 = **apply**(myocarde[myocarde$PRONO=="1",1:7],2,**mean**)  Sigma = **var**(myocarde[,1:7])  omega = **solve**(Sigma)%\*%(m1-m0)  omega  [,1]  FRCAR -0.012909708542  INCAR 1.088582058796  INSYS -0.019390084344  PRDIA -0.025817110020  PAPUL 0.020441287970  PVENT -0.038298291091  REPUL -0.001371677757 |

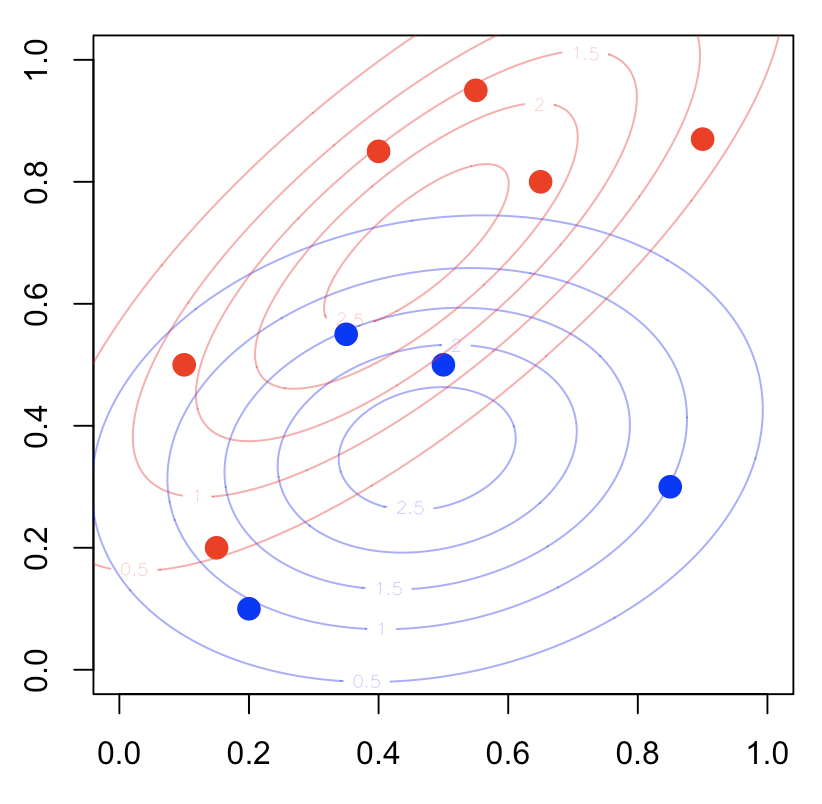
For the constant – in the equation ωTx+b=0\omega^T\mathbf{x}+b=0ωTx+b=0 – if we have equiprobable probabilities, use

|  |
| --- |
| b = (**t**(m1)%\*%**solve**(Sigma)%\*%m1-**t**(m0)%\*%**solve**(Sigma)%\*%m0)/2 |

**Application (on the small dataset)**

In order to visualize what’s going on, consider the small dataset, with only two covariates,

|  |
| --- |
| x = **c**(.4,.55,.65,.9,.1,.35,.5,.15,.2,.85)  y = **c**(.85,.95,.8,.87,.5,.55,.5,.2,.1,.3)  z = **c**(1,1,1,1,1,0,0,1,0,0)  **df** = **data.frame**(x1=x,x2=y,y=**as.factor**(z))  m0 = **apply**(**df**[**df**$y=="0",1:2],2,**mean**)  m1 = **apply**(**df**[**df**$y=="1",1:2],2,**mean**)  Sigma = **var**(**df**[,1:2])  omega = **solve**(Sigma)%\*%(m1-m0)  omega  [,1]  x1 -2.640613174  x2 4.858705676 |

  
Using R regular function, we get

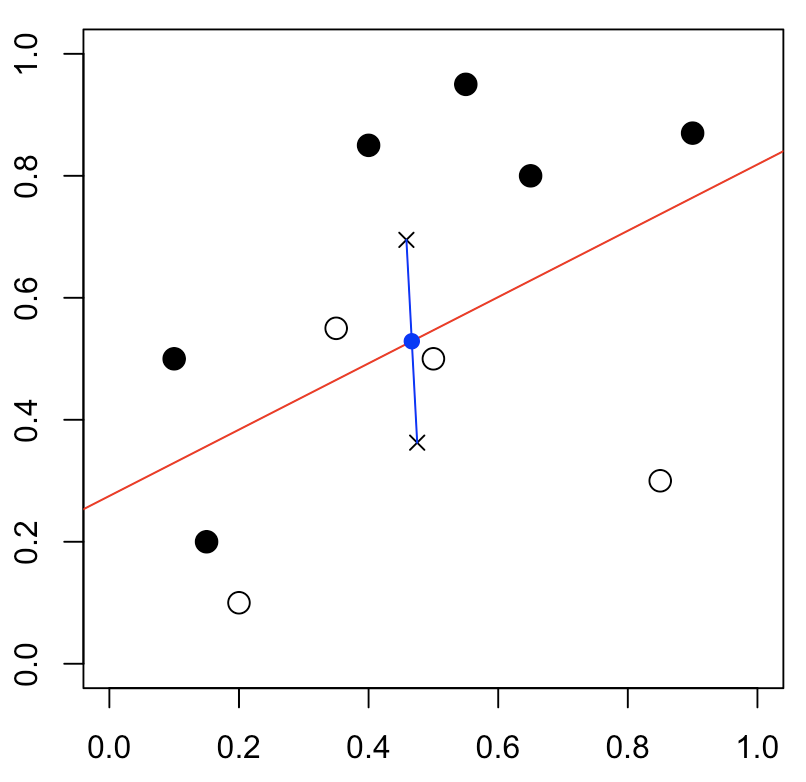
|  |
| --- |
| **library**(MASS)  fit\_lda = lda(y ~x1+x2 , **data**=**df**)  fit\_lda    Coefficients of linear discriminants:  LD1  x1 -2.588389554  x2 4.762614663 |

which is the same coefficient as the one we got with our own code. For the constant, use

|  |
| --- |
| b = (**t**(m1)%\*%**solve**(Sigma)%\*%m1-**t**(m0)%\*%**solve**(Sigma)%\*%m0)/2 |

If we plot it, we get the red straight line

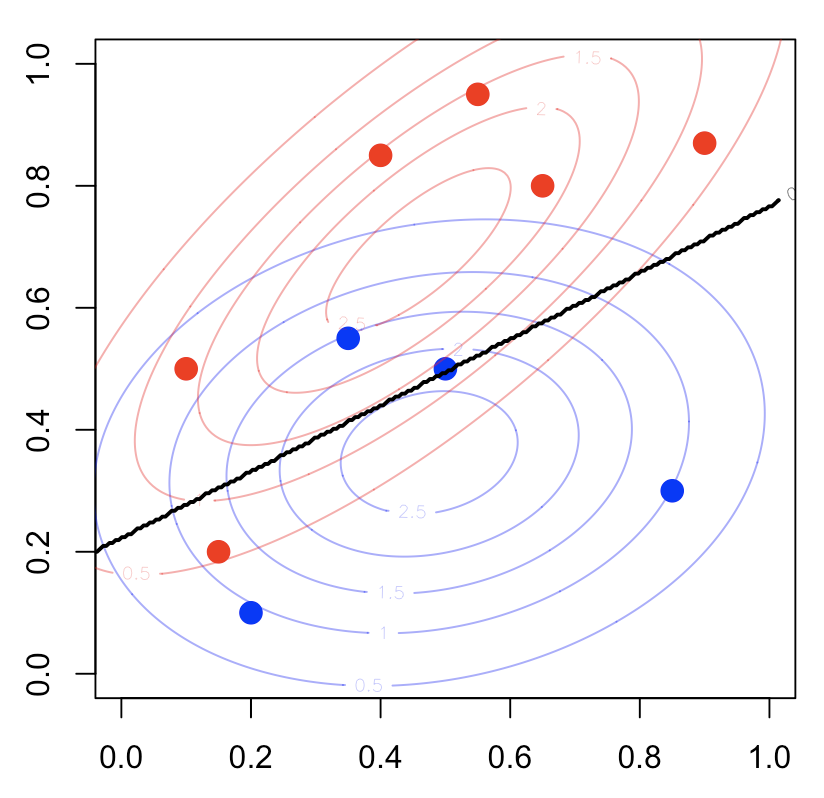
|  |
| --- |
| **plot**(**df**$x1,**df**$x2,pch=**c**(1,19)[1+(**df**$y=="1")])  **abline**(a=b/omega[2],b=-omega[1]/omega[2],**col**="red") |

  
As we can see (with the blue points), our red line intersects the middle of the segment of the two barycenters

|  |
| --- |
| **points**(m0["x1"],m0["x2"],pch=4)  **points**(m1["x1"],m1["x2"],pch=4)  **segments**(m0["x1"],m0["x2"],m1["x1"],m1["x2"],**col**="blue")  **points**(.5\*m0["x1"]+.5\*m1["x1"],.5\*m0["x2"]+.5\*m1["x2"],**col**="blue",pch=19) |

Of course, we can also use R function

|  |
| --- |
| predlda = **function**(x,y) **predict**(fit\_lda, **data.frame**(x1=x,x2=y))$class==1  vv=**outer**(vu,vu,predlda)  **contour**(vu,vu,vv,add=TRUE,lwd=2,**levels** = .5) |

  
One can also consider the quadratic discriminent analysis since it might be difficult to argue that Σ0=Σ1\mathbf{\Sigma}\_0=\mathbf{\Sigma}\_1Σ0​=Σ1​

|  |
| --- |
| fit\_qda = qda(y ~x1+x2 , **data**=**df**) |

The separation curve is here

|  |
| --- |
| **plot**(**df**$x1,**df**$x2,pch=19,  **col**=**c**("blue","red")[1+(**df**$y=="1")])  predqda=**function**(x,y) **predict**(fit\_qda, **data.frame**(x1=x,x2=y))$class==1  vv=**outer**(vu,vu,predlda)  **contour**(vu,vu,vv,add=TRUE,lwd=2,**levels** = .5) |

