Exercice : Test de Kolmogorov-Smirnov

Code

Hide

Hide

```
set.seed(254)
k=1:100
l=1:100
for (i in 1:100)
{
  u=rexp(1,1)
  l[i]=u
}
so=l[order(l)]
so
```

```
  [1] 0.005143292 0.007998401 0.035032275 0.038825120 0.053518175 0.079166044 0.10
7159502 0.131127127 0.136118233
 [10] 0.158363834 0.160438983 0.181117203 0.205961673 0.214485342 0.218351417 0.22
3184937 0.224276716 0.228771869
 [19] 0.232283868 0.233438427 0.236372354 0.245644187 0.248222437 0.260990661 0.26
1556915 0.267238777 0.272209368
 [28] 0.272860450 0.276354577 0.278804836 0.294241696 0.310352448 0.322630604 0.34
0961270 0.366683055 0.368288236
 [37] 0.397390094 0.413780637 0.418461236 0.420138591 0.450135275 0.450721797 0.46
3897370 0.539803244 0.576316005
 [46] 0.605506711 0.621298573 0.639958588 0.643892045 0.667425530 0.668248098 0.67
8926278 0.692402035 0.711499895
 [55] 0.742813652 0.767528851 0.769268951 0.778684772 0.860676320 0.863709202 0.93
9297402 1.004607570 1.030692153
 [64] 1.053208095 1.090745795 1.104893157 1.248641727 1.254266503 1.277752831 1.28
5727725 1.349126834 1.373063784
 [73] 1.407445110 1.411408625 1.427030606 1.451181516 1.456112422 1.463243833 1.47
6518646 1.482341621 1.501251490
 [82] 1.673895370 1.720761325 1.771965988 1.784164976 1.795787865 1.878004692 2.01
6635647 2.145343950 2.356171688
 [91] 2.417059082 2.491645159 2.650724599 2.750335669 2.918883867 3.131893150 3.42
0829050 4.086011892 4.841131080
[100] 5.591181144
```

Hide

Hide

```
#les statistiques d'ordre
l2=1:100
for (i in 1:100)
{
    l2[i]=i/100
}
#le zalpha du theoreme de Massart
alph=0.05
zalpha=sqrt(1/2*log(2/alph))
zalpha/10
```

```
[1] 0.1358102
```

Hide

Hide

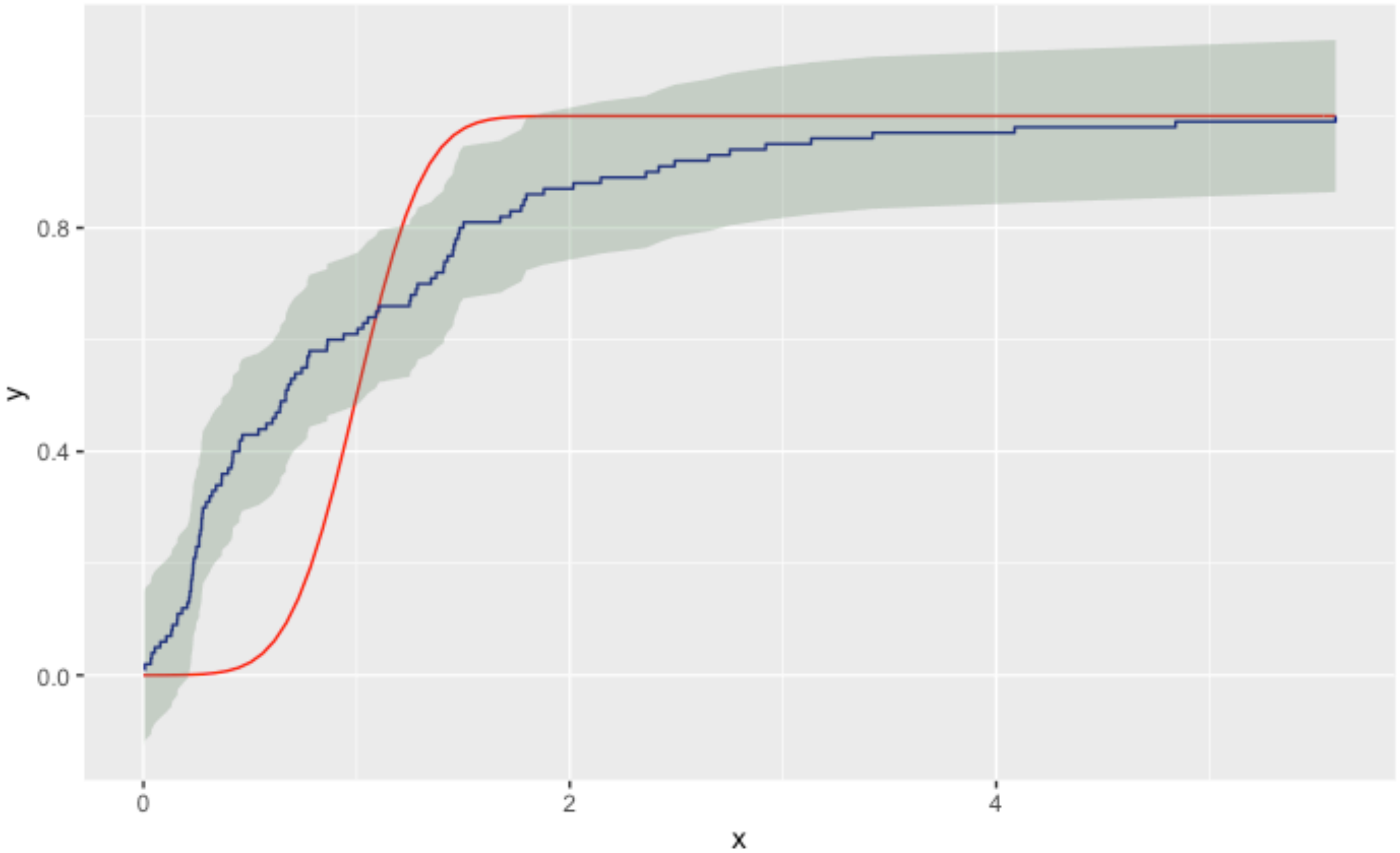
```
plotdata <- data.frame(x=so, y=l2, lower = (l2-zalpha/10), upper = (l2+zalpha/10),
qdown=qnorm((l2-zalpha/10)), qup=qnorm((l2+zalpha/10)))
```

```
NaNs producedNaNs produced
```

Hide

Hide

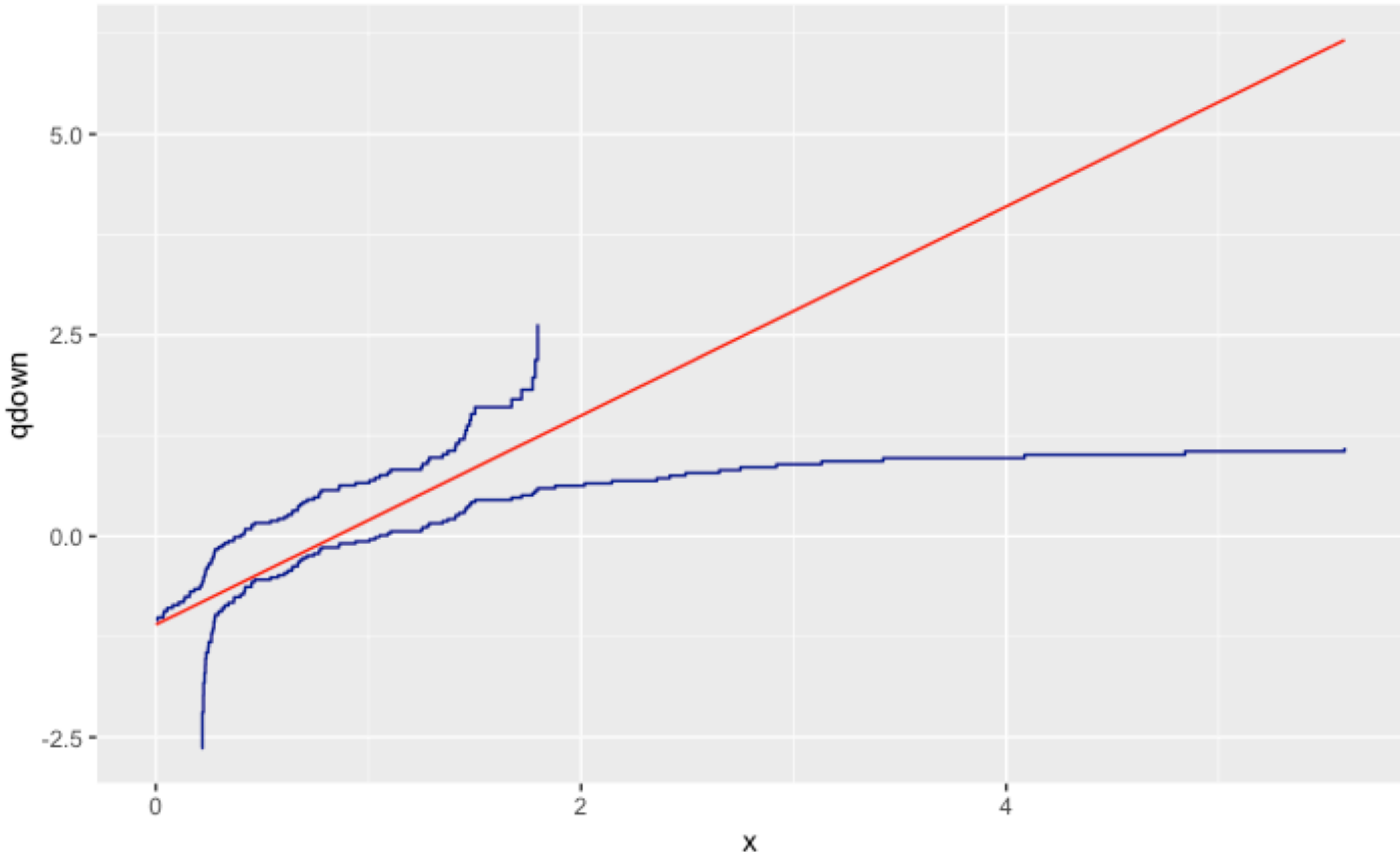
```
p4 <- ggplot(data.frame(x = c(0, 4)), aes(x = x)) +
  stat_function(fun =test, colour="red") +geom_step(data=plotdata,aes(so,l2),colou
r="darkblue")+geom_ribbon(data=plotdata, aes(ymin=lower, ymax=upper, x=x, fill = "
band"), alpha = 0.3, fill="darkseagreen4")
p4
```



Hide

Hide

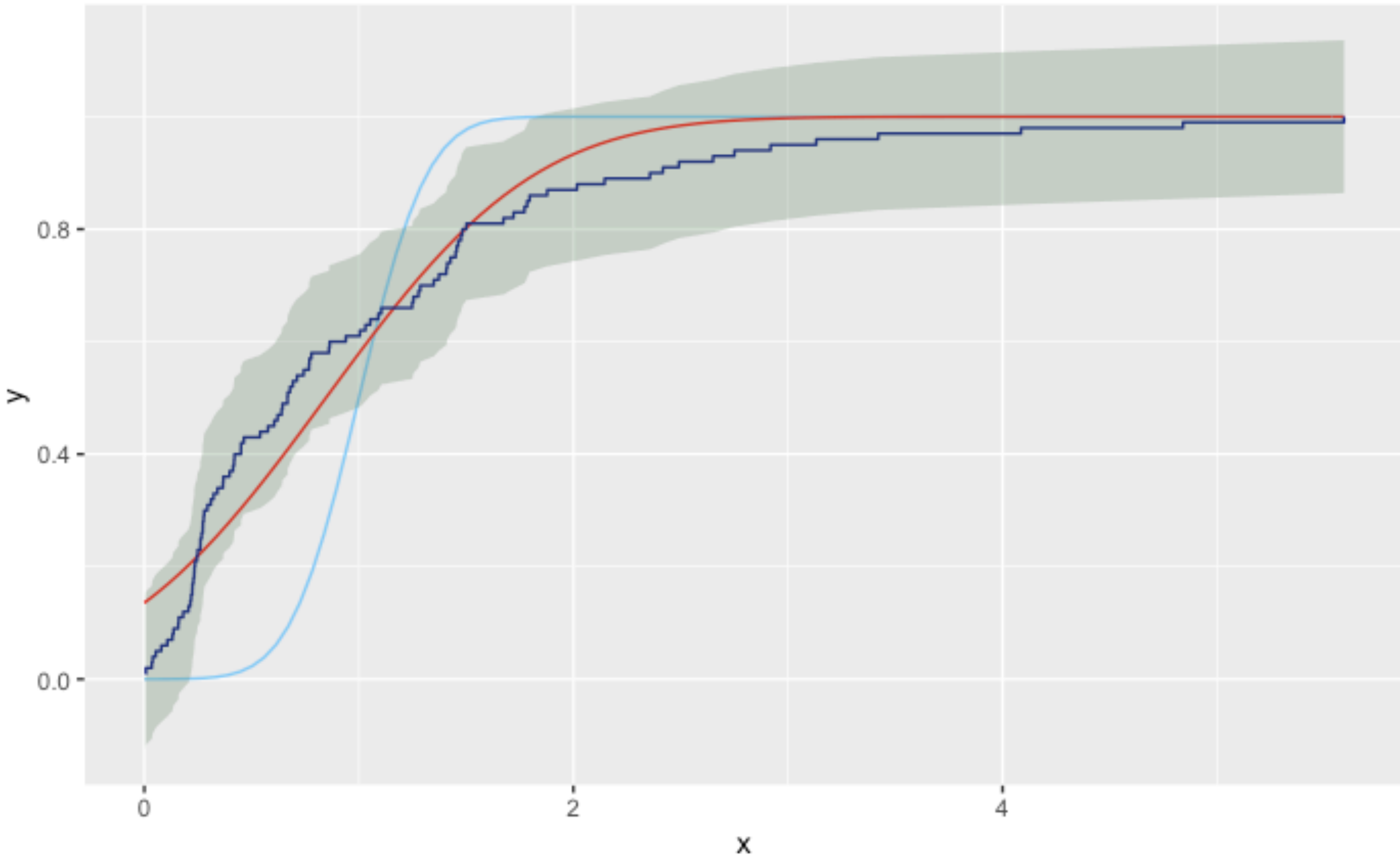
```
#Apres plusieurs essais on otient une droite comprise entre les deux courbes
a=1.3
b=-1.1
droite<-function(x){a*x+b}
ptest<- ggplot(data.frame(x = c(0, 4)), aes(x = x)) +geom_step(data=plotdata,aes(s
o,qdown),colour="darkblue")+geom_step(data=plotdata,aes(so,qup),colour="darkblue")
+stat_function(fun =droite, colour="red")
ptest
```



Hide

Hide

```
#On superpose la loi normale correspondante sur le graphe des fonctions de reparti
tion
ppp<- ggplot(data.frame(x = c(0, 4)), aes(x = x)) + stat_function(fun =test, colou
r="lightskyblue")+
  stat_function(fun =test_norm, colour="red") +geom_step(data=plotdata,aes(so,l2),
colour="darkblue")+geom_ribbon(data=plotdata, aes(ymin=lower, ymax=upper, x=x, fil
l = "band"), alpha = 0.3, fill="darkseagreen4")
ppp
```



La fonction de repartition de la N(1,1/2) n est pas dans la bande de confiance de la fonction de repartition empirique basee sur les X1,…Xn iid~Exp(1). On rejette donc l hypothese nulle au niveau 5%.

Pour le modele general avec un test portant sur la famille des loi normales, on trouve une loi qui appartient a l intervalle et ne pouvont donc pas rejeter l hypothese nulle.