```
#########################################################################

# TD - Exercises in R on kriging


# Bertrand Iooss & Fabrice Gamboa - Barranquilla - October 2015


#########################################################################


rm(list=ls())

graphics.off()


#################################################################


########## Part 1


set.seed(12345)


################

# a) Function


myfunc <- function(x){

  return( sin(30*(x - 0.9)^4)*cos(2*(x - 0.9)) + (x - 0.9)/2)

}


# vizualisation

ntest <- 1000

xtest <- seq(0,1,le=ntest)
```

```
ytest <- myfunc(xtest)


x11()
plot(xtest,ytest,type="l",xlab="x",ylab="y")


################
#b) Simple kriging


library(DiceKriging)
help(package="DiceKriging")
?km


# design of experiments
nx <- 5
x <- seq(0,1,le=nx)
y <- myfunc(x)


# plot the design points
points(x,y)


# parameters of kriging
mu = 0
sig2 = 0.5
theta = 0.2


# Building the kriging
krig <- km(~1,design=data.frame(x=x), response=y, covtype="gauss",
```

```
        coef.var=sig2, coef.trend=mu, coef.cov=theta)


# predictions of kriging
ypred <- predict(object=krig, newdata=xtest, type="SK", checkNames=F)
Q2 = 1 - mean((ytest-ypred$mean)^2)/var(ytest)
print(Q2)


# plot the kriging model
x11()
plot(xtest,ytest,type="l",xlab="x",ylab="y",lwd=2,
    main="kriging predictions",ylim=c(min(ypred$lower95),max(ypred$upper95)))
points(x,y,col=2,pch=2,lwd=2)
lines(xtest,ypred$mean,col=4,lwd=2)
lines(xtest,ypred$lower95,col=4,lty=2)
lines(xtest,ypred$upper95,col=4,lty=2)


# using DiceView
library(DiceView)


x11()
sectionview(krig,ylim=c(min(ypred$lower95),max(ypred$upper95)))
lines(xtest,ytest,type="l",xlab="x",ylab="y",lwd=2)


################
#3) Kriging with unkown hyperparameters


nx <- 7
```

```
x <- seq(0,1,le=nx)
y <- myfunc(x)

krig <- km(~1,design=data.frame(x=x), response=y, covtype="gauss",
        coef.var=sig2, coef.trend=mu, coef.cov=theta)
ypred <- predict(object=krig, newdata=xtest, type="UK", checkNames=F)
Q2 = 1 - mean((ytest-ypred$mean)^2)/var(ytest)
print(Q2)
x11()
sectionview(krig,ylim=c(min(ypred$lower95),max(ypred$upper95)))
lines(xtest,ytest,type="l",xlab="x",ylab="y",lwd=2)

krig <- km(~1,design=data.frame(x=x), response=y, covtype="matern5_2")
ypred <- predict(object=krig, newdata=xtest, type="UK", checkNames=F)
Q2 = 1 - mean((ytest-ypred$mean)^2)/var(ytest)
print(Q2)
x11()
sectionview(krig,ylim=c(min(ypred$lower95),max(ypred$upper95)))
lines(xtest,ytest,type="l",xlab="x",ylab="y",lwd=2)



################
#4) Adaptive design

xadd <- which.max(ypred$sd)/ntest
yadd <- myfunc(xadd)
points(xadd,yadd,lwd=2)
```

```
x <- c(x,xadd)
y <- c(y,yadd)

krig <- km(~1,design=data.frame(x=x), response=y, covtype="matern5_2")
ypred <- predict(object=krig, newdata=xtest, type="SK", checkNames=F)
Q2 = 1 - mean((ytest-ypred$mean)^2)/var(ytest)

x11()
sectionview(krig,ylim=c(min(ypred$lower95),max(ypred$upper95)))
lines(xtest,ytest,type="l",xlab="x",ylab="y",lwd=2)

for (i in 1:4){
 xadd <- which.max(ypred$sd)/ntest
 yadd <- myfunc(xadd)
 points(xadd,yadd,lwd=2)
 x <- c(x,xadd)
 y <- c(y,yadd)

 krig <- km(~1,design=data.frame(x=x), response=y, covtype="matern5_2")
 ypred <- predict(object=krig, newdata=xtest, type="SK", checkNames=F)
 Q2 = c(Q2,1 - mean((ytest-ypred$mean)^2)/var(ytest))
 x11()
 sectionview(krig,ylim=c(min(ypred$lower95),max(ypred$upper95)))
 lines(xtest,ytest,type="l",xlab="x",ylab="y",lwd=2)
}
```

```
x11()
plot(seq(8,12),Q2)
print(Q2)

################
#5) Quantile estimation

# nx <- 8
# x <- seq(0,1,le=nx)
# y <- myfunc(x)
# krig <- km(~1,design=data.frame(x=x), response=y, covtype="matern5_2")
# ypred <- predict(object=krig, newdata=xtest, type="SK", checkNames=F)

x11()
sectionview(krig,ylim=c(min(ypred$lower95),max(ypred$upper95)))
lines(xtest,ytest,type="l",xlab="x",ylab="y",lwd=2)

q1 = quantile(ytest,0.95)
print(c("True 0.95-quantile = ",q1))
abline(h=q1)

q2 = quantile(ypred$mean,0.95)
print(c("Kriging 0.95-quantile = ",q2))
abline(h=q2,col=4)

nsim = 1e4
zsim = simulate(krig,nsim=nsim,cond=TRUE,newdata=data.frame(x=xtest),nugget.sim=1e-5)
```

```
q=rep(0,nsim)
for (i in 1:nsim){
  q[i] = quantile(zsim[i,],0.95)
}
for (i in 1:10) lines(xtest,zsim[i,],col=2)

q3 = mean(q)
print(c("Quantile by conditional simul. =",q3))
abline(h=q3,col=2)

res <- quantile(q,c(0.05,0.95))
print(c("0.95-quantile CI =",res))

# Histogram of the output
x11()
hist(ytest)
abline(v=q1)
abline(v=q2,col=4)
abline(v=q3,col=2)
```