Le but de ce T.P. est d'illustrer le principe de l'estimation non paramétrique d'une fonction de régression par la méthode du noyau ainsi que la validation croisée pour choisir un paramètre de lissage (fenêtre) en régression non paramétrique.

Partie 1 : Simulation de données

On suppose que l'on dispose de l'observation de n var $Y_i, i = 1, \ldots, n$ indépendantes, et l'on se place dans le cadre du modèle de régression standard :

$$Y_i = f(x_i) + \epsilon_i,\ i = 1, \ldots, n,$$
$$\epsilon_1, \ldots, \epsilon_n \text{ i.i.d. avec } \mathbb{E}(\epsilon_i) = 0,\ \mathrm{Var}(\epsilon_i) = \sigma^2,$$

où les prédicteurs x_i sont **déterministes** et à valeurs dans $\mathbb{R}$ et f est une fonction inconnue que l'on cherche à estimer. Dans ce TP, on va estimer les 4 fonctions de régression suivantes à l'aide d'un estimateur de Nadaray-Watson :

```
# Ensemble de 4 fonctions tests
T = 1000
t = (1:T)/T

truef1 = 0.5 + (0.2*cos(4*pi*t)) + (0.1*cos(24*pi*t))
truef2 = 0.2 + 0.6*(t > 1/3 & t <= 0.75)
truef3 = 4*sin(4*pi*t) - sign(t - .3) - sign(.72 - t) + 5
truef4 = sqrt(t*(1-t))*sin((2*pi*1.05) /(t+.05)) + 0.5
```

Sous R, un estimateur de Nadaraya-Watson (qui est un estimateur par polynôme locaux de degré zéro) peut s'implémenter à partir de la fonction `locpoly` de la librairie `KernSmooth`.

On commence par se créer 4 jeux de données $Y_1, \ldots, Y_4$, à partir desquels on va estimer les fonctions

1) Utiliser le code ci-dessous pour créer puis visualiser les données pour chacune des 4 fonctions de régression $f_1, \ldots, f_4$.

```
# Choix des points du design : n valeurs regulierement espacees sur [0,1]
n <- 100
x <- (1:n)/n

# Data 1
rsnr = 5 #rapport signal sur bruit
f1 = 0.5 + (0.2*cos(4*pi*x)) + (0.1*cos(24*pi*x))
sigma1 <- sd(f1)/rsnr # Niveau de bruit
Y1 <- f1 + rnorm(n,mean=0,sd=sigma1)
```

```
# Visualisation des donnees avec la fonction de regression
dev.new()
plot(x,Y1,type="p",pch=19)
lines(t,truef1, type = "l", col="red", lty="dotdash", lwd=2)
```

2) Utiliser le code ci-dessous pour ajuster un estimateur à noyau à chacun des 4 jeux de données $Y_1, \ldots, Y_4$ qui correspondent aux fonctions $f_1, \ldots, f_4$.

```
library(KernSmooth)

# Estimation a noyau
h = 0.1
hatf1 = locpoly(x,Y1,degree=0,bandwidth=h,gridsize=T,range.x=c(0,1))$y

plot(t,truef1, type = "l", col="red", lty="dotdash", lwd=2)
lines(t, hatf1,type="l",col="blue", pch=19)
```

3) Pour chaque fonction de régression, faites varier le paramètre de fenêtre h pour l'estimateur à noyau. Utiliser la connaissance de la vraie fonction de régression f pour choisir la valeur optimale h^* qui minimise la norme L^2 entre f et $\hat{f}_h$ i.e.

$$h^* = \arg\min_{h>0} \left\{ \int (\hat{f}_h(x) - f(x))^2 dx \right\}.$$

Comparer graphiquement f et $\hat{f}_{h^*}$.

Partie 2 : Validation croisée

On note $\mathcal{A} = \{(Y_i, x_i), i = 1, \ldots, n\}$, l'ensemble des données et on suppose que l'on veut évaluer l'erreur quadratique intégrée moyenne :

$$R(h) = \mathbb{E} \int_{\mathbb{R}} (\hat{f}_{n,h}(x) - f(x))^2 dx,$$

pour un estimateur $\hat{f}_{n,h}$ de f qui dépend d'un paramètre de lissage h et qui est construit à partir de l'ensemble complet des données $\mathcal{A}$.

On note $\mathcal{A}_{-i} = \{(Y_j, x_j), j = 1, \ldots, n, j \neq i\}$ l'ensemble des données privé de la i-ème observation, et on construit alors un estimateur $\hat{f}_{n,h,-i}$ à partir de cet ensemble incomplet de données $\mathcal{A}_{-i}$. Un estimateur de $R(h)$ est alors donné par

$$\hat{R}(h) = \sum_{i=1}^{n} (x_{i+1} - x_i)(Y_i - \hat{f}_{n,h,-i}(x_i))^2.$$

Un choix "optimal" de h est alors obtenu par minimisation de $\hat{R}(h)$ i.e.

$$\hat{h}_n = \arg\min_{h \in \mathbb{R}^+} \hat{R}(h).$$

4) Reprenez les exemples simulés précédents pour choisir un meilleur paramètre de fenêtre pour l'estimation des fonctions de régression $f_1, \ldots, f_4$. Dans chaque cas, tracer l'estimateur obtenu et comparer le à la vraie fonction de régression.