Fiche TD $N =^{\circ}$ **04** - Nonparametric Regression

Exercice 01: Consider the distribution with joint pdf

$$f(x,y) = \begin{cases} e^x + y, & \text{if } 0 \le x \le 1 \text{ and } 0 \le y \le 1; \\ 0, & \text{otherwise.} \end{cases}$$

1. Déterminer la densité marginale $f_X(x)$
2. Déterminer la densité marginale $f_Y(y)$
3. Les v.a.r. X et Y sont-elles indépendantes ?
4. Calculer l'estimateur de Nadaraya-Watson de la function $r(x) = \mathbb{E}(Y/X = x)$

Exercice 02: prove that

$$\int_{\mathbb{R}} \frac{y}{h} \sum_{i=1}^{n} K\left(\frac{X - x_i}{h}\right) K\left(\frac{Y - y_i}{h}\right) dy = \sum_{i=1}^{n} K\left(\frac{X - x_i}{h}\right) y_i.$$

Exercice 03: For any fixed x, we write $L(c) = \sum_{i=1}^{n} K\left(\frac{X-x_i}{h}\right)(y_i - c)^2$
Show that Nadaraya-Watson estimator is the solution of optimization of $L(c)$.so

$$\widehat{r}_n(x) = \arg\min_{c \in \mathbb{R}} \sum_{i=1}^{n} K\left(\frac{X - x_i}{h}\right)(y_i - c)^2$$

Exercice 04: Soit un échantillon d'observations $(Y_1, Y_2, ..., Y_N)$ d'une variable aléatoire Y observé aux dates $1, 2, ..., N$. On suppose pour simplifier que ces observations sont classées par ordre croissant. On considère l'éstimateur à noyau donné par :

$$m(t) = \sum_{j=1}^{N} W_j(t) Y_j$$

où les poids $W_j(t)$ vérifient : $W_j(t) = \dfrac{K_h(t - t_j)}{\sum_{j=1}^{N} K_h(t - t_j)}$ et $\sum_{j=1}^{N} W_j = 1$

où $K(.)$ désigne une fonction kernel. Montrez que l'éstimateur $m(t_i)$ sera nécessairement compris entre le minimum et le maximum des observations Y_j ; c'est à dire que :

$$Y_1 \le m(t_i) \le Y_N$$

Exercice 05:simulation sous R:
Données de geyser Old Faithful : pour 272 eruptions du geyser, on a enregistré les durées d'eruptions et les temps entre les éruptions successives.
Nous cherchons à estimer la densité du temps d'attente entre les éruptions successives.voici le code sous R :

```
summary(faithful)
hist(faithful$waiting, probability=T)
breaks=c(40,50,55,60,70,75,77.5,80,82.5,85,90,100)
hist(faithful$waiting,breaks,probability=T)
```

1. **Estimation de la densité par noyau dans R :**

```
layout(matrix(1:3, ncol = 3))
 plot(density(faithful$waiting))
 plot(density(faithful$waiting,bw=8))
 plot(density(faithful$waiting,bw=0.8))
```

2. **problème de régression :**
   Exemple : données de moto :mesures d'accélération de tête dans un accident simulé de moto.

   ```
   library(MASS); plot(mcycle)
   ```

   (a) **Régression polynomiale**

   ```
   attach(mcycle)
   fit3<-lm(accel~times+I(times^2)+I(times^3))
   fit6<-lm(accel~times+I(times^2)+I(times^3)+I(times^4)+I(times^5)+I(times^6))
    fit9<-lm(accel~times+I(times^2)+I(times^3)+I(times^4)+I(times^5)+I(times^6)+
   + I(times^7)+I(times^8)+I(times^9))
    plot(times, accel)
    lines(times, fit3$fitted, lty=1)
    lines(times, fit6$fitted, lty=2)
    lines(times, fit9$fitted, lty=3)
    legend(40, -70, c("fit3", "fit6", "fit9"), lty=c(1,2,3))
   ```

   (b) **Estimateur par noyau de Nadaraya-Watson**

   ```
   plot(times, accel)
   lines(ksmooth(times, accel, "normal", bandwidth=1), lty=1)
   lines(ksmooth(times, accel, "normal", bandwidth=5), lty=2)
   legend(40, -100, legend=c("bandwidth=1", "bandwidth=5"), lty=c(1,2))
   ```

   (c) **Lissage par polynômes locaux (LOESS)** : L'idee : ajuster un polynôme aux données localement

   ```
    attach(mcycle)
    mcycle.1 <- loess(accel~times, span=0.1)
    mcycle.2 <- loess(accel~times, span=0.2)
    mcycle.3 <- loess(accel~times, span=1)
    prtimes<- matrix((0:1000)*((max(times)-min(times))/1000)+min(times), ncol=1)
    praccel.1 <-predict(mcycle.1, prtimes)
    praccel.2 <-predict(mcycle.2, prtimes)
    praccel.3 <-predict(mcycle.3, prtimes)
    plot(mcycle, pch="+")
    lines(prtimes, praccel.1, lty=1)
    lines(prtimes, praccel.2, lty=2)
    lines(prtimes, praccel.3, lty=3)
    legend(40, -90, legend=c("span=0.1", "span=0.2", "span=1"), lty=c(1,2,3),bty="n")
   ```

   span=α contrôle la proportion des points utilisés dans le voisinage local; Le degré du polynôme est = 2.