SOLUTION TP nº 2

Solution 1.

1. Créer une data frame `acteurs` qui renvoie :

```
  Mort.à Années.de.carrière Nombre.de.films  Prénom       Nom Date.du.décès
1     93                66             211  Michel   Galabru    04-01-2016
2     53                25              58   André Raimbourg    23-09-1970
3     72                48              98    Jean     Gabin    15-10-1976
4     68                37             140   Louis  de Funès    27-01-1983
5     68                31              74    Lino   Ventura    22-10-1987
6     53                32              81 Jacques  Villeret    28-01-2005
```

```
Mort.à = c(93, 53, 72, 68, 68, 53)
Années.de.carrière = c(66, 25, 48, 37, 31, 32)
Nombre.de.films = c(211, 58, 98, 140, 74, 81)
Prénom = c("Michel", "André", "Jean", "Louis", "Lino", "Jacques")
Nom = c("Galabru", "Raimbourg", "Gabin", "de Funès", "Ventura", "Villeret")
Date.du.décès = c("04-01-2016", "23-09-1970", "15-10-1976", "27-01-1983",
"22-10-1987", "28-01-2005")
acteurs = data.frame(Mort.à, Années.de.carrière, Nombre.de.films, Prénom,
Nom, Date.du.décès)
```

2. Changer le nom de la 1-ère colonne par : `"Age.du.décès"` :

```
colnames(acteurs)[1] = "Age.du.décès"
```

3. Extraire la colonne `Prénom` :

```
acteurs$Prénom
```

4. Ordonner la data frame par ordre croissant suivant l'âge de la mort :

```
acteurs[order(acteurs$Age.du.décès), ]
```

Solution 2. Le goût d'un fromage dépend de la concentration de plusieurs composés chimiques, dont :

- la concentration d'acide acétique (variable $X1$),
- la concentration d'hydrogène sulfuré (variable $X2$),
- la concentration d'acide lactique (variable $X3$).

Pour 30 types de fromage, on dispose du score moyen attribué par des goûteurs (caractère Y).
Le jeu de données `fromages` est disponible ici :

`https://chesneau.users.lmno.cnrs.fr/fromages.txt`

1. Construire une data frame `w` constituée du jeu de données `fromages` avec les noms des colonnes :

   ```
   w = read.table("https://chesneau.users.lmno.cnrs.fr/fromages.txt",
   header = T)
   ```

2. Associer (attacher) les noms aux colonnes respectives. Taper `X1` pour voir si cela a marché :

   ```
   attach(w)
   X1
   ```

3. Afficher les caractéristiques de `w` :

   ```
   str(w)
   ```

4. Donner les paramètres statistiques élémentaires pour les variables Y, $X1$, $X2$ et $X3$:

   ```
   summary(w)
   ```

5. Faite les commandes : `pairs(w)`. Que cela renvoie t'il ?

   ```
   pairs(w)
   ```

   Cela renvoie les nuages de points des variables deux à deux.

6. Construire une nouvelle data frame `ww` des individus vérifiant $X1 > 5.1$ et $X3 < 1.77$:

   ```
   ww = w[(X1 > 5.1) & (X3 < 1.77), ]
   ```

7. Afficher les caractéristiques de `ww` :

   ```
   str(ww)
   ```

8. À partir de `ww`, donner les paramètres statistiques élémentaires pour les variables Y, $X1$, $X2$ et $X3$:

   ```
   summary(ww)
   ```

Solution 3.

1. Créer dans R la fonction $f : \mathbb{N}^2 \times [0,1] \to [0,1]$ définie par :

$$f(k,n,p) = \frac{1}{2}\binom{n}{k}p^k(1-p)^{n-k} + \frac{k}{n(n+1)} :$$

```
fonc = function(k, n, p) {
(1 / 2) * choose(n, k) * p^k * (1 - p)^(n - k) + k / (n * (n + 1))
}
```

2. Utiliser des commandes R pour montrer que :

$$\sum_{k=0}^{20} f(k,20,0.4) = 1 :$$

```
x = fonc(0:20, 20, 0.4)
sum(x)
```

3. Construire une matrice A de 4 colonnes et 21 lignes de composantes $f(k, 20, p)$ avec $p \in \{0.1, 0.3, 0.6, 0.9\}$ pour les colonnes et $k \in \{0, \ldots, 20\}$ pour les lignes.
   Vérifier que la somme des valeurs des éléments des colonnes fait 1 :

```
A = cbind(fonc(0:20, 20, 0.1), fonc(0:20, 20, 0.3), fonc(0:20, 20, 0.6),
fonc(0:20, 20, 0.9))
apply(A, 2, sum)
```

Solution 4. Créer dans R la fonction $f : \mathbb{R}^n \to \mathbb{R}^{n-2}$ définie par :

$$f(x_1, \ldots, x_n) = \left(\frac{x_1 + x_2 + x_3}{3}, \frac{x_2 + x_3 + x_4}{3}, \ldots, \frac{x_{n-2} + x_{n-1} + x_n}{3} \right) :$$

```
fonc = function(x){
n = length(x)
( x[1:(n - 2)] + x[2:(n - 1)] + x[3:n] ) / 3
}
```

Solution 5.

1. Créer dans R une fonction `fonc` décrivant la fonction $f : \mathbb{R} \to [0, \infty[$ définie par :

$$f(x) = \begin{cases} \frac{1}{24} x^4 e^{-x} & \text{si } x \geq 0, \\ 0 & \text{sinon :} \end{cases}$$

```
fonc = function(x) {
if (x<0)
0
else
(1/ 24) *  x^4 * exp(-x)
}
```

   ou plus simplement :

```
fonc = function(x) {
ifelse(x >=0, (1/ 24) *  x^4 * exp(-x), 0)
}
```

2. Reproduire et commenter les commandes suivantes :

```
curve(fonc(x), 0, 10, col = "cyan", lwd = 4)
```

   Cela trace le graphe de la fonction $f(x)$ avec $x \in [0, 10]$ de couleur bleue cyan (et une épaisseur de trait de niveau "4").

Solution 6.

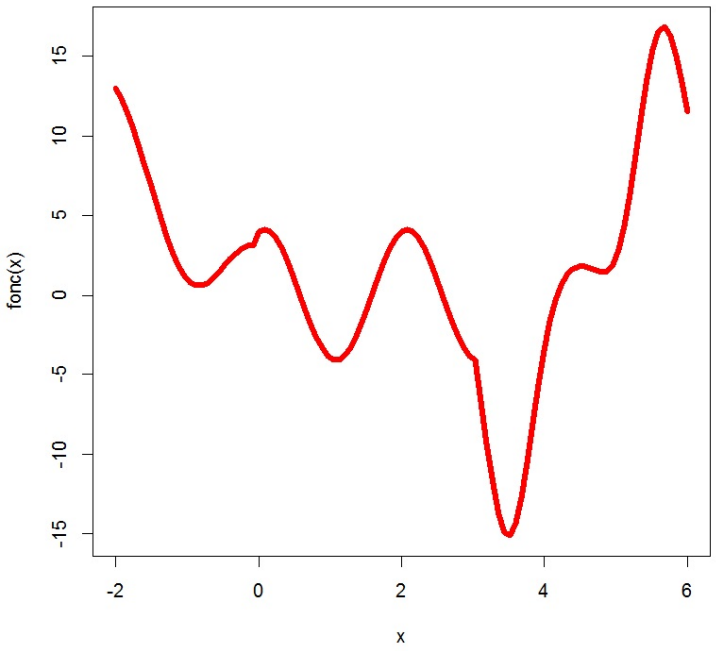
1. Créer dans R une fonction `fonc` décrivant la fonction $f : \mathbb{R} \to [0, \infty[$ définie par :

$$f(x) = \begin{cases} x^2 - 3x + 3\cos(\pi x) & \text{si } x < 0, \\ 3\cos(\pi x) + \cos(\pi x) + \sin(\pi x) & \text{si } x \in [0, 3], \\ 2\sin(\pi x) + \dfrac{1}{3}(x - 3\sin(\pi x))^2 \cos(x) & \text{si } x > 3, \end{cases}$$

   (on pourra utiliser la commande `ifelse`*)* :

```
fonc = function(x) {
ifelse(x  < 0,
x^2 - 3 * x + 3 * cos(pi * x),
ifelse(x <= 3,
 3 * cos(pi * x) + cos(pi * x) + sin(pi * x),
 2 * sin(pi * x) + (1 / 3) * (x-3 * sin(pi * x))^2 * cos(x)))
}
```

2. Reproduire et commenter les commandes suivantes :

   `curve(fonc(x), -2, 6, col = "red", lwd = 5)`

   Cela trace le graphe de la fonction $f(x)$ avec $x \in [-2, 6]$ de couleur rouge (et une épaisseur de trait de niveau "5") :