ACT6100	Analyse de données
H2019	Solutions 3

1. (a) Le code permettant d'obtenir les résultats est présenté à la Figure 1. Ainsi, l'équation du modèle

```
plot(data1$X, data1$Y)

modele1 <- lm(Y ~ X, data = data1)
summary(modele1)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -121356.90   16241.48  -7.472 3.33e-11 ***
X               906.72      46.79  19.379  < 2e-16 ***

abline(-121356.90, 906.72, add = TRUE)

sum((predict(modele1) - data1$Y)^2)

543910771909
```

FIGURE 1 – Code informatique.

1 est

$$\widehat{Y} = -121\,356.90 + 906.72X.$$

Les valeurs p associées aux paramètres permettent de conclure que ces derniers sont significatifs. Le MSE est 543 910 771 909.

(b) Le code permettant d'obtenir les résultats est présenté à la Figure 2. Ainsi, l'équation du modèle

```
data1$X2 <- data1$X^2

modele2 <- lm(Y ~ X2 + X, data = data1)

summary(modele2)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.018e+04  2.655e+04  -0.383    0.702
X2           1.127e+00  2.251e-01   5.006 2.48e-06 ***
X            1.060e+02  1.654e+02   0.641    0.523

lines(data1$X, -1.018e+04 + 1.127e+00*data1$X2 + 1.060e+02*data1$X)

sum((predict(modele2) - data1$Y)^2)

432234571187
```

FIGURE 2 – Code informatique.

2 est

$$\widehat{Y} = -10\,180 + 1.127X^2 + 106X.$$

Les valeurs p associées aux paramètres permettent de conclure que X n'est pas une variable significative dans le modèle. Le MSE est 432 234 571 187. Si on retire la variable non significative du modèle, on obtient alors un MSE de 434 064 644 839.

(c) On a vu que le modèle le plus flexible sera automatique celui qui minimise l'erreur quadratique moyenne. Un modèle avec $K = 1$ s'ajustera parfaitement aux données et conduira à un MSE de 0. La Figure 3 illustre la situation.

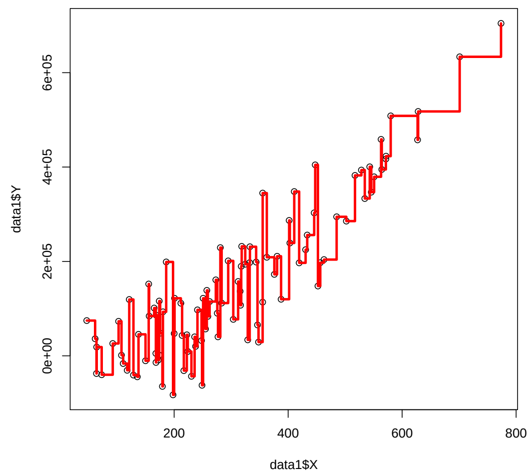

FIGURE 3 – Ajustement d'un modèle avec $K = 1$.

(d) Le code permettant d'obtenir les résultats est présenté à la Figure 4. La Figure 5 illustre le comportement de l'erreur quadratique moyenne de validation lorsque la validation croisée est utilisée. On remarque que le minimum est atteint lorsque $K = 10$. La Figure 6 illustre l'ajustement du modèle optimal. Le modèle optimal produit un MSE de 572 388 692 923.

```
FUN <- function(x){
  knn.reg(train = data1$X, y = data1$Y, k = x)$PRESS
}
MSEout <- sapply(1:50, function(x) FUN(x))

(1:50)[which(MSEout == min(MSEout))]

modele4 <- knn.reg(train = data1$X, y = data1$Y, k = 10)

sum((modele4$pred - data1$Y)^2)

572388692923
```

FIGURE 4 – Code informatique.

(e) Le code permettant d'obtenir les résultats est présenté à la Figure 7. On obtient, dans l'ordre, 6 609 874 222, 2 580 396 252, 17 333 311 211, 2 338 794 222. On remarque que le modèle 3 ($K = 1$) produit une erreur quadratique de prédiction très élevée et que le modèle optimal ($K = 10$) est celui dont l'erreur quadratique de prédiction est la plus faible.

2. (a) L'équation du modèle est $Y = \beta_0 + \beta_1 X$. On doit déterminer les valeurs de β_0 et β_1 qui minimisent

$$\text{MSE} = \frac{1}{5} \sum_{i=1}^{5} (Y_i - \beta_0 - \beta_1 X_i)^2 .$$

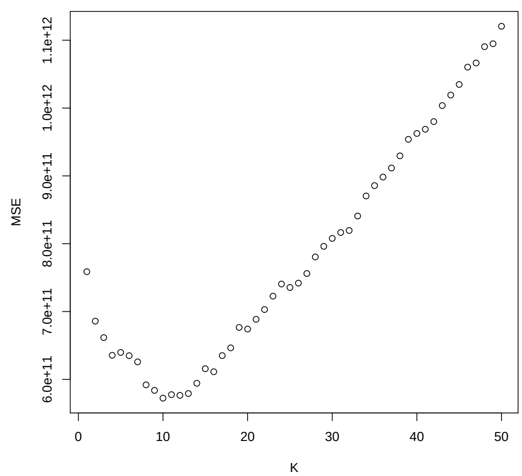

FIGURE 5 – Erreur quadratique moyenne de validation avec validation croisée.

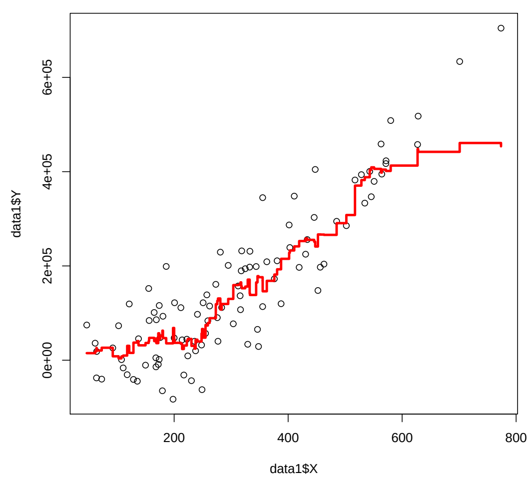

FIGURE 6 – Ajustement d'un modèle avec $K = 10$.

```
P1 <- -121356.90 + 906.72*Xpred
P2 <- -1.018e+04 + 1.127e+00*Xpred^2 + 1.060e+02*Xpred
P3 <- 231164.472
P4 <- knn.reg(train = matrix(data1$X, ncol = 1),
      test = matrix(Xpred, ncol = 1), y = data1$Y, k = 10)$pred

c("M1" = (P1-Ypred)^2, "M2" = (P2-Ypred)^2,
  "M3" = (P3-Ypred)^2, "M4" = (P4-Ypred)^2)
```

FIGURE 7 – Code informatique.

Les dérivées partielles sont

$$\frac{\partial \text{MSE}}{\partial \beta_0} = -2 \sum_{i=1}^{5} (Y_i - \beta_0 - \beta_1 X_i) = 0$$

$$\frac{\partial \text{MSE}}{\partial \beta_1} = -2 \sum_{i=1}^{5} (Y_i - \beta_0 - \beta_1 X_i) X_i = 0.$$

En résolvant ce système d'équations, on obtient

$$\widehat{\beta}_0 = \frac{\sum_{i=1}^5 Y_i - \widehat{\beta}_1 \sum_{i=1}^5 X_i}{5} = 5.1327$$
$$\widehat{\beta}_1 = \frac{5\sum_{i=1}^5 Y_i X_i - \sum_{i=1}^5 Y_i \sum_{i=1}^5 X_i}{5\sum_{i=1}^5 X_i^2 - \left(\sum_{i=1}^5 X_i\right)^2} = -0.159292.$$

Ainsi, l'équation du modèle est $Y = 5.1327 - 0.159292X$.

(b) Il faut déterminer, pour toutes les valeurs de $X \in \mathbb{R}$, les deux plus proches voisins et calculer la moyenne des Y correspondants. Ainsi, pour $x \leq 2.5$, les deux valeurs de X les plus proches sont 1 et 2, et on obtient $\widehat{f}(X) = (4+8)/2 = 6$. On a alors

$$\widehat{f}(X) = \begin{cases} 6, & X \leq 2.5 \\ 5, & 2.5 < X \leq 4.5 \\ 2.5, & 4.5 < X \leq 6.5 \\ 4, & X > 6.5 \end{cases}$$

Le choix de fermer les intervalles à droite est totalement arbitraire.

(c) La Table 1 présente les prédictions obtenues. Par exemple, pour la première ligne, les deux plus

i	Y_i	X_i	$\widehat{Y}_i$
1	4	1	5.0
2	2	4	6.0
3	8	2	3.0
4	5	9	2.5
5	3	7	3.5

Table 1 – Base de données et valeurs prédites.

proches voisins sont $X = 2$ et $X = 4$ ce qui donne $\widehat{Y} = (8+2)/2 = 5$. L'erreur quadratique moyenne de validation est

$$\text{vMSE} = \frac{(5-4)^2 + (6-2)^2 + \cdots + (3.5-3)^2}{5} = \frac{48.5}{5}.$$

3. (a) On remarque dans la base de données quelques assurés dont l'âge est étrange $(0, 4, 5, \ldots$ ans). Pour l'analyse, le code présenté à la Figure 8 permet de conserver uniquement les assurés de 16 ans ou plus. On remarque également que certains assurés ont une exposition nulle (0). Le code présenté à la Figure 8 permet également des les retirer de la base de données.

```
### Vérifier les âges
table(swmotorcycle$OwnerAge)

### Retirer les assurés de moins de 17 ans
swmotorcycle <- swmotorcycle[swmotorcycle$OwnerAge >= 16, ]

### Retirer les assurés avec une exposition nulle
swmotorcycle <- swmotorcycle[swmotorcycle$Exposure > 0, ]
```

Figure 8 – Code informatique.

(b) Le code présenté à la Figure 9 permet d'ajuster le modèle sur les données nettoyées et d'obtenir les valeurs des paramètres.

(c) **(i)** Le code présenté à la Figure 10 permet d'ajuster le modèle sur les données nettoyées et d'obtenir les prédictions. **(ii)** Le code présenté à la Figure 11 permet d'ajuster le modèle sur les données nettoyées et d'obtenir les prédictions.

```
modele1 <- glm(ClaimNb ~ OwnerAge, family = poisson(link = "log"),
               offset = log(Exposure), data = swmotorcycle)

summary(modele1)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.13447    0.11952  -17.86   <2e-16 ***
OwnerAge    -0.06035    0.00319  -18.92   <2e-16 ***
```

FIGURE 9 – Code informatique.

```
FUN <- function(x){
swmotorcycle$OwnerAgeClass <- cut(swmotorcycle$OwnerAge, c(0, x, 100))
modele <- glm(ClaimNb ~ OwnerAgeClass, family = poisson(link = "log"),
              offset = log(Exposure), data = swmotorcycle)
sum((predict(modele, type = "response") - swmotorcycle$ClaimNb)^2)
}

MSEin <- sapply(17:90, function(x) FUN(x))

(17:90)[which(MSEin == min(MSEin))]

swmotorcycle$OwnerAgeClass <- cut(swmotorcycle$OwnerAge, c(0, 77, 100))

modele2 <- glm(ClaimNb ~ OwnerAgeClass, family = poisson(link = "log"),
               offset = log(Exposure), data = swmotorcycle)

coef(modele2)

exp(-4.543029)

0.01064113

exp(-4.543029 - 11.312377)

1.300423e-07
```

FIGURE 10 – Code informatique.

```
FUN <- function(w){
  dataTrain$OwnerAgeClass <- cut(dataTrain$OwnerAge, c(0, w, 100))
  dataValidation$OwnerAgeClass <- cut(dataValidation$OwnerAge, c(0, w, 100))
  modele <- glm(ClaimNb ~ OwnerAgeClass, family = poisson(link = "log"),
            offset = log(Exposure), data = dataTrain)
  sum((predict(modele, type = "response", newdata = dataValidation) -
       dataValidation$ClaimNb)^2)
}

set.seed(100)
n <- length(swmotorcycle$OwnerAge)
indice <- matrix(sample(1:n, n, replace = FALSE), nrow = 12)

FUN2 <- function(x){
train <- as.vector(indice[-x, ])
valid <- as.vector(indice[x, ])
dataTrain <<- swmotorcycle[train, ]
dataValidation <<- swmotorcycle[valid, ]
sapply(25:85, function(y) FUN(y))
}
MSEout <- sapply(1:12, function(x) FUN2(x))

MSE <- rowMeans(MSEout)

optimalK <- (25:85)[which(MSE == min(MSE))]

swmotorcycle$OwnerAgeClass <- cut(swmotorcycle$OwnerAge, c(0, optimalK, 100))
modele3 <- glm(ClaimNb ~ OwnerAgeClass, family = poisson(link = "log"),
               offset = log(Exposure), data = swmotorcycle)

coef(modele3)

exp(-3.463325)

0.03132543

exp(-3.463325 -1.598827 )

0.006331919
```

Figure 11 – Code informatique.