ACT6100	**Analyse de données**
H2019	**Solutions 5**

1. (a) Avec un paramètre de régularisation de $\lambda = 0$, il s'agit d'une régression linéaire simple. On a alors

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{8} X_i Y_i}{\sum_{i=1}^{8} X_i^2} = \frac{35}{16} = 2.1875$$
$$\widehat{\beta}_0 = \frac{\sum_{i=1}^{8} Y_i}{8} = 40.$$

Ainsi, l'équation du modèle est $\widehat{Y} = 40 + 2.1875X$.

(b) Pour la régression Ridge en général, on a vu que l'ordonnée à l'origine n'était pas affectée et donc que $\widehat{\beta}_0 = 40$. La solution générale, en dimension p, est

$$\widehat{\boldsymbol{\beta}}^{\boldsymbol{R}} = \left(\boldsymbol{X}^{**T}\boldsymbol{X}^{**} + \lambda \boldsymbol{I}_p\right)^{-1} \boldsymbol{X}^{**T}\boldsymbol{Y}$$

qui devient, lorsque $p = 1$,

$$= \frac{\sum_{i=1}^{8} X_i Y_i}{\sum_{i=1}^{8} X_i^2 + \lambda(1)}$$
$$= \frac{35}{16 + \lambda}.$$

Pour $\lambda = 4$, on obtient $\widehat{\beta}_1 = 1.75$ (la valeur est, comme prévue, plus près de 0 que celle obtenue dans la régression linéaire classique). L'équation du modèle est alors $\widehat{Y} = 40 + 1.75X$. On pouvait également utiliser cette approche pour trouver la solution à la sous-question précédente.

Par la suite, on obtient les prédictions et on calcule l'erreur quadratique moyenne :

$$\widehat{Y}_1 = 40 + 1.75(-2) = 36.5$$
$$\widehat{Y}_2 = \widehat{Y}_3 = \widehat{Y}_4 = 38.25$$
$$\widehat{Y}_5 = 40.00$$
$$\widehat{Y}_6 = 41.75$$
$$\widehat{Y}_7 = \widehat{Y}_8 = 43.50$$
$$\to MSE = \frac{\sum_{i=1}^{8} \left(\widehat{Y}_i - Y_i\right)^2}{8}$$
$$= 1.8125.$$

2. (a) Pour trouver un optimum de la fonction objectif, on calcule la dérivée partielle par rapport à

β_0

$$
\begin{aligned}
\frac{\partial}{\partial \beta_0} & \left(\sum_{i=1} = n\left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\right) \\
&= \sum_{i=1}^{n} 2\left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij}\right)(-1) = 0 \\
&\rightarrow \sum_{i=1}^{n} Y_i - n\beta_0 - \sum_{i=1}^{n}\sum_{j=1}^{p} \beta_j X_{ij} = 0 \\
&\rightarrow \widehat{\beta}_0 = \frac{\sum_{i=1}^{n} Y_i}{n} - \sum_{j=1}^{p} \beta_j \frac{\sum_{i=1}^{n} X_{ij}}{n} \\
&\rightarrow \widehat{\beta}_0 = \bar{Y} - \sum_{j=1}^{p} \beta_j \bar{X}_j.
\end{aligned}
$$

Le calcul de la dérivée seconde permet de vérifier qu'il s'agit bien d'un minimum local

$$
\begin{aligned}
\frac{\partial^2}{\partial \beta_0^2} & \left(\sum_{i=1}^{n} 2\left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij}\right)(-1)\right) \\
&= 2n > 0.
\end{aligned}
$$

(b) On a

$$
\begin{aligned}
&\arg\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n}\left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij}\right)^2 + \lambda \sum_{j=1} \beta_j^2 \\
&\arg\min_{\beta_1,\ldots,\beta_p} \sum_{i=1}^{n}\left(Y_i - \bar{Y} - \sum_{j=1}^{p} \beta_j \bar{X}_j - \sum_{j=1}^{p} \beta_j X_{ij}\right)^2 + \lambda \sum_{j=1} \beta_j^2 \\
&\arg\min_{\beta_1,\ldots,\beta_p} \sum_{i=1}^{n}\left(Y_i^* - \sum_{j=1}^{p} \beta_j X_{ij}^*\right)^2 + \lambda \sum_{j=1} \beta_j^2.
\end{aligned}
$$

(c) On a (il faut se rappeler un peu comment faire du calcul matriciel [1] de base, sinon ça devient un peu lourd...)

$$
\begin{aligned}
&\arg\min_{\boldsymbol{\beta}^*} \left(\boldsymbol{Y}^* - \boldsymbol{X}^*\boldsymbol{\beta}^*\right)^T \left(\boldsymbol{Y}^* - \boldsymbol{X}^*\boldsymbol{\beta}^*\right) + \lambda \boldsymbol{\beta}^{*T}\boldsymbol{\beta}^* \\
&\arg\min_{\boldsymbol{\beta}^*} \boldsymbol{Y}^{*T}\boldsymbol{Y}^* - 2\boldsymbol{\beta}^{*T}\boldsymbol{X}^{*T}\boldsymbol{Y}^* + \boldsymbol{\beta}^{*T}\boldsymbol{X}^{*T}\boldsymbol{X}^*\boldsymbol{\beta}^* + \lambda\boldsymbol{\beta}^{*T}\boldsymbol{I}_p\boldsymbol{\beta}^* \\
&\quad \frac{\partial}{\partial \boldsymbol{\beta}^*}\left(\boldsymbol{Y}^{*T}\boldsymbol{Y}^* - 2\boldsymbol{\beta}^{*T}\boldsymbol{X}^{*T}\boldsymbol{Y}^* + \boldsymbol{\beta}^{*T}\boldsymbol{X}^{*T}\boldsymbol{X}^*\boldsymbol{\beta}^* + \lambda\boldsymbol{\beta}^{*T}\boldsymbol{I}_p\boldsymbol{\beta}^*\right) \\
&\qquad = -2\boldsymbol{X}^{*T}\boldsymbol{Y}^* + 2\boldsymbol{X}^{*T}\boldsymbol{X}^*\boldsymbol{\beta}^* + 2\lambda\mathbf{I}_p\boldsymbol{\beta}^* = 0 \\
&\quad \rightarrow \widehat{\boldsymbol{\beta}}^* = \left(\boldsymbol{X}^{*T}\boldsymbol{X}^* + \lambda\boldsymbol{I}_p\right)^{-1}\boldsymbol{X}^{*T}\boldsymbol{Y}^*.
\end{aligned}
$$

3. (a) Le code permettant d'importer et de construire la base de données est présenté à la Figure 1.
   (b) Le code permettant d'ajuster le modèle est présenté à la Figure 2.
   (c) i. Avec les résultats présentés à la Figure 2, on a, pour Y, le cout total d'un sinistre (sachant qu'il y a eu sinistre), on a

$$
\begin{aligned}
E[Y] &= \exp(7.5711 - 0.0008 * 400 - 0.0449 + 0.0321 + 0.3270 * 45 * 0.0137) \\
&= 1\,702.572.
\end{aligned}
$$

1. N'hésitez pas à vérifier en dimension 2 les propriétés dont vous n'êtes pas certaine ou certain.

```
data(freMPL3)
data <- freMPL3
data <- data[data$ClaimAmount > 0,]
```

Figure 1 – Code informatique.

```
data1 <- data[,c(1,2,5,6,7,8,10,21)]
head(data1)

modele1 <- glm(ClaimAmount ~ LicAge + VehAge + Gender + MariStat + SocioCateg
               + DrivAge, family = Gamma(link = "log"), data = data1,
               offset = log(Exposure))
summary(modele1)

                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      7.5710910  0.3240725  23.362  < 2e-16 ***
LicAge          -0.0008203  0.0007727  -1.062 0.288630
VehAge1          0.1294967  0.1574067   0.823 0.410847
VehAge10+       -0.1471522  0.3170181  -0.464 0.642605
VehAge2         -0.0980039  0.1566142  -0.626 0.531586
VehAge3         -0.0448791  0.1695603  -0.265 0.791302
VehAge4         -0.1953874  0.1713965  -1.140 0.254522
VehAge5          0.7147311  0.1907686   3.747 0.000188 ***
VehAge6-7        0.4353038  0.1915252   2.273 0.023210 *
VehAge8-9        0.2439549  0.2466899   0.989 0.322902
GenderMale       0.0321148  0.0978605   0.328 0.742840
MariStatOther    0.2287443  0.1209635   1.891 0.058859 .
SocioCategCSP20  0.0255151  0.8166161   0.031 0.975079
SocioCategCSP21 -1.4977180  1.6061031  -0.933 0.351255
SocioCategCSP22 -0.6301074  1.1438103  -0.551 0.581814
SocioCategCSP26  0.4710595  0.3158833   1.491 0.136156
SocioCategCSP37  0.0138807  0.4183706   0.033 0.973538
SocioCategCSP42 -0.1137252  0.3396833  -0.335 0.737835
SocioCategCSP46  0.0694105  0.3664637   0.189 0.849806
SocioCategCSP47 -1.1527185  1.1430011  -1.009 0.313414
SocioCategCSP48  0.1746436  0.3388862   0.515 0.606405
SocioCategCSP49 -0.1603568  0.4584226  -0.350 0.726548
SocioCategCSP50  0.3269536  0.2057376   1.589 0.112280
SocioCategCSP55  0.1298638  0.2479113   0.524 0.600491
SocioCategCSP6  -0.8686871  0.7359607  -1.180 0.238094
SocioCategCSP60 -0.0116380  0.2485941  -0.047 0.962668
SocioCategCSP65 -0.4241183  1.1552812  -0.367 0.713599
SocioCategCSP66  0.0827468  0.3305943   0.250 0.802400
SocioCategCSP74 -3.4982896  1.1484951  -3.046 0.002369 **
SocioCategCSP77 -2.1169095  0.8202913  -2.581 0.009977 **
DrivAge          0.0137474  0.0091838   1.497 0.134671
```

Figure 2 – Code informatique.

ii. Le coefficient associé à la variable **DrivAge** n'a pas une valeur significativement différente de 0 (la valeur p du test est supérieure à 0.05).

(d) Le code permettant de sélectionner la valeur de $K = 10$ à l'aide du critère AIC est présenté à la Figure 3. Le code permettant de sélectionner la valeur de $K = 2$ à l'aide du critère MSE et de la 2-validation croisée est présenté à la Figure 4.

```
data1$DrivAge2 <- data1$DrivAge^2
data1$DrivAge3 <- data1$DrivAge^3
data1$DrivAge4 <- data1$DrivAge^4
data1$DrivAge5 <- data1$DrivAge^5
...

M1 <- glm(ClaimAmount ~ DrivAge, family = Gamma(link = "log"), data = data1,
          offset = log(Exposure))$aic
M2 <- glm(ClaimAmount ~ DrivAge + DrivAge2, family = Gamma(link = "log"),
          data = data1, offset = log(Exposure))$aic
M3 <- glm(ClaimAmount ~ DrivAge + DrivAge2+ DrivAge3, family = Gamma(link = "log"),
          data = data1, offset = log(Exposure))$aic
M4 <- glm(ClaimAmount ~ DrivAge + DrivAge2 + DrivAge3 + DrivAge4,
          family = Gamma(link = "log"), data = data1, offset = log(Exposure))$aic
...

c(M1, M2, M3, M4, M5, M6, M7, M8, M9, M10)
22004.34 22005.14 22007.08 22008.74 22010.35 22011.74 22013.72
22014.87 22005.53 21994.72
```

FIGURE 3 – Code informatique.

```
### Création de deux bases pour la 2-validation croisée
set.seed(100)
indice <- sample(1:length(data1$Exposure), 623, replace = FALSE)
dataA <- data1[indice,]
dataB <- data1[-indice,]

### Fonction qui calcule le MSE pour chacun des modèles avec x comme
### base d'entrainement et y comme base d'évaluation

FUN <- function(x, y)
{
  M1 <- glm(ClaimAmount ~ DrivAge, family = Gamma(link = "log"), data = x,
            offset = log(Exposure))
  M2 <- glm(ClaimAmount ~ DrivAge + DrivAge2, family = Gamma(link = "log"),
            data = x, offset = log(Exposure))
  M3 <- glm(ClaimAmount ~ DrivAge + DrivAge2+ DrivAge3, family = Gamma(link = "log"),
            data = x, offset = log(Exposure))
  M4 <- glm(ClaimAmount ~ DrivAge + DrivAge2 + DrivAge3 + DrivAge4,
            family = Gamma(link = "log"), data = x, offset = log(Exposure))
  ...
  MSE1 <- mean((predict(M1, newdata = y, type = "response") - y$ClaimAmount)^2)
  MSE2 <- mean((predict(M2, newdata = y, type = "response") - y$ClaimAmount)^2)
  MSE3 <- mean((predict(M3, newdata = y, type = "response") - y$ClaimAmount)^2)
  MSE4 <- mean((predict(M4, newdata = y, type = "response") - y$ClaimAmount)^2)
  ...
  c(MSE1, MSE2, MSE3, MSE4, MSE5, MSE6, MSE7, MSE8, MSE9, MSE10)
  }

(FUN(dataA, dataB) + FUN(dataB, dataA))/2
```

FIGURE 4 – Code informatique.

(e) Le code permettant d'obtenir le modèle Ridge et les résultats sont présentés à la Figure 5.

```
X <- as.matrix(data1[,c(2,7)])
Y <- round(as.matrix(data1[,8]),0)
OS <- as.matrix(data1[,1])

cv.fit <- cv.glmnet(X, Y, nfolds = 10, alpha = 0, family = "poisson",
                    offset = log(OS))

coef(cv.fit)
                      1
(Intercept) 8.042979e+00
LicAge      2.099967e-38
DrivAge     1.168982e-36

cv.fit$lambda.min
18376.2
```

FIGURE 5 – Code informatique.

(f) Le code permettant d'obtenir le modèle Ridge et les résultats sont présentés à la Figure 6.

```
### Pour utiliser la fonction cv.glmnet avec une (ou plusieurs) variables
### explicatives catégorielles, il faut construire manuellement les
### variables dichotomiques avec la fonction model.matrix
X <- model.matrix(~VehUsage - 1, data = data)
head(X, 3)
 VehUsagePrivate VehUsagePrivate+trip to office VehUsageProfessional
              0                              0                    1
              1                              0                    0
              0                              0                    1
   VehUsageProfessional run
35                        0
52                        0
60                        0

Y <- round(as.matrix(data1[,8]),0)
OS <- as.matrix(data1[,1])
cv.fit <- cv.glmnet(X, Y, nfolds = 10, alpha = 0, family = "poisson",
                    offset = log(OS))
coef(cv.fit)

(Intercept)                     8.042979e+00
VehUsagePrivate                 9.611938e-35
VehUsagePrivate+trip to office  8.361573e-35
VehUsageProfessional           -1.952645e-34
VehUsageProfessional run       -2.098311e-34
```

FIGURE 6 – Code informatique.

4. On a

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^R &= \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{X}^T\mathbf{Y} \\
&= \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} \\
&= \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\widehat{\boldsymbol{\beta}}.
\end{aligned}$$

Si $\lambda = 0$, alors on a

$$E\left[\widehat{\boldsymbol{\beta}}^R\right] = E\left[\widehat{\boldsymbol{\beta}}\right] = \boldsymbol{\beta}$$

et sinon

$$\begin{aligned} E\left[\widehat{\boldsymbol{\beta}}^R\right] &= E\left[\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\widehat{\boldsymbol{\beta}}\right] \\ &\neq \boldsymbol{\beta}. \end{aligned}$$

5. (a) Le code permettant d'obtenir les résultats es présenté à la Figure 7 et le graphique est présenté à la Figure 8. On observe qu'une valeur $\alpha = 0$ est optimale et que le modèle est en fait

```
library(ggplot2)

data(mtcars)

X <- model.matrix(mpg ~ ., data=mtcars)[,-1]
Y <- mtcars$mpg

alphas <- seq(0, 1, by=.01)
mses <- numeric(101)

for(i in 1:101){
  cvfits <- cv.glmnet(X, Y, alpha = alphas[i], nfolds = 32)
  loc <- which(cvfits$lambda == cvfits$lambda.min)
  mses[i] <- cvfits$cvm[loc]
}

RES <- data.frame(mse=mses, alpha=alphas)

plot1 <- ggplot(RES, aes(x=alpha, y=mse)) +
  geom_point(shape=1) +
  ylab("MSE (validation croisée)") +
  xlab("alpha") +
  ggtitle("")
plot1
```

FIGURE 7 – Code informatique.

purement Ridge. Ce résultat s'explique probablement par la forte multicolinéarité présente dans les données.

(b) Le code permettant d'obtenir les résultats es présenté à la Figure 9 et le graphique est présenté à la Figure 10. On observe qu'une valeur $\alpha = 1$ est optimale et que le modèle est en fait purement Lasso.

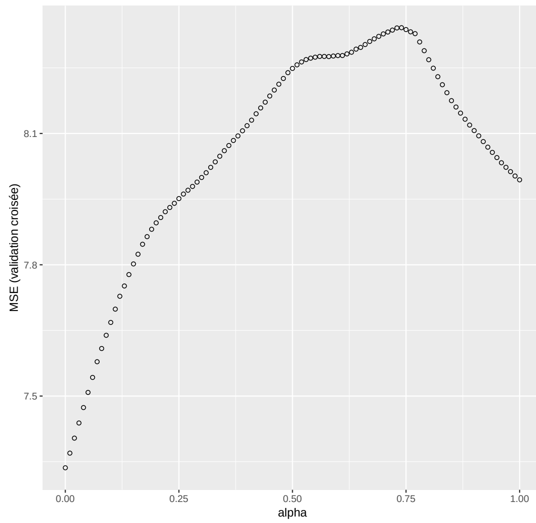

FIGURE 8 – Évolution de l'erreur quadratique moyenne en fonction de la valeur de α.

```
load("/.../QuickStartExample.RData")

X <- x
Y <- y

alphas <- seq(0, 1, by=.01)
mses <- numeric(101)


for(i in 1:101){
  cvfits <- cv.glmnet(X, Y, alpha = alphas[i], nfolds=100)
  loc <- which(cvfits$lambda==cvfits$lambda.min)
  mses[i] <- cvfits$cvm[loc]
}

RES <- data.frame(mse=mses, alpha=alphas)

plot1 <- ggplot(RES, aes(x = alpha, y = mse)) +
  geom_point(shape = 1) +
  ylab("Erreur quadratique moyenne") +
  xlab("alpha") +
  ggtitle("")

plot1
```

FIGURE 9 – Code informatique.

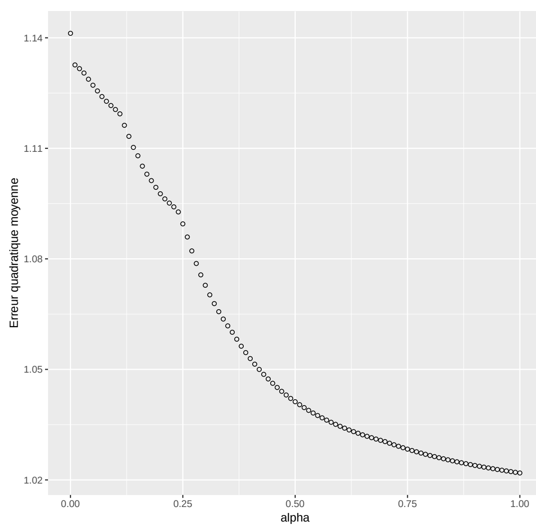

FIGURE 10 – Évolution de l'erreur quadratique moyenne en fonction de la valeur de α.