Solution Exercice #3, Série 5

Francis Duval

18 février 2020

Activer les librairies utiles.

```
library(here)
library(tidyverse)
library(glue)
library(magrittr)
library(glmnet)
```

Lire la base de données `credit.csv`.

```
credit <- load(here("0_data", "freMPL3.rda"))
```

a)

```
data <- freMPL3 %>% filter(ClaimAmount > 0)
```

b)

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

summary(gamma_fit)
```

```
##
## Call:
## glm(formula = ClaimAmount ~ LicAge + VehAge + Gender + MariStat +
##     SocioCateg + DrivAge, family = Gamma(link = "log"), data = data,
##     offset = log(Exposure))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8065  -1.1024  -0.5599   0.1153   6.0984
##
## Coefficients:
##                     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
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 2.500965)
##
##     Null deviance: 1882.1  on 1246  degrees of freedom
## Residual deviance: 1721.3  on 1216  degrees of freedom
## AIC: 21928
##
## Number of Fisher Scoring iterations: 9
```

i)

```r
assure_df <- tibble(
  LicAge = 400,
  VehAge = factor(3),
  Gender = factor("Male"),
  MariStat = factor("Alone"),
  SocioCateg = factor("CSP50"),
  DrivAge = 45,
  Exposure = 1
)

as.numeric(predict(gamma_fit, newdata = assure_df, type = "response"))
```

```
## [1] 3553.833
```

ii)

Cette affirmation est fort probablement fausse puisque la valeur-p associée au paramètre `DrivAge` est trop élevée.

c)

Premièrement, créer une base de données avec seulement les variables `Exposure`, `ClaimAmount` et `DrivAge`.

```r
data_drv_age <- data %>% select(ClaimAmount, Exposure, DrivAge)
```

i)

Créer une fonction qui prend en entrée la base `df` et qui renvoie la même base de données avec les `k` puissances de la variable `var`.

```
ajout_puissances <- function(df, var, k) {
  if (k == 1) {
    return(df)
  }

  new_vars <- map_dfc(2:k, ~ df[[var]] ^ .x) %>%
    setNames(glue("{var}_{2:k}"))

  res <- bind_cols(df, new_vars)
  return(res)
}
```

Ajuster le GLM gamma pour $K = 1, \ldots, 10$.

```
data_drv_age_ls <- map(1:10, ~ ajout_puissances(df = data_drv_age, var = "DrivAge", k = .x))

gamma_fit_ls <- map(
  data_drv_age_ls,
  ~ glm(ClaimAmount ~ ., family = Gamma(link = "log"), data = .x, offset = log(Exposure))
)
```

Calculer l'AIC pour chaque modèle ajuster et choisir K tel que l'AIC est le plus petit.

```
(AICs <- map_dbl(gamma_fit_ls, AIC))
```

```
##  [1] 21327.55 21329.20 21328.50 21330.44 21332.19 21334.04 21336.03
##  [8] 21334.10 21333.05 21319.97
```

```
which.min(AICs)
```

```
## [1] 10
```

Avec le critère AIC, on choisit donc $K = 10$.

ii)

Premièrement, créer une fonction qui prend en entrée la base de données, la variable explicative et le nombre de puissances k et renvoie l'erreur quadratique moyenne de validation croisée.

```
mse_2_folds_gamma <- function(df, var, k) {
  dat <- ajout_puissances(df, var, k)

  folds <-  seq(1, nrow(dat)) %>%
      cut(breaks = 2, labels = F) %>%
      sample()

  responses_ls <- dat %>%
    mutate(folds = folds) %>%
    group_split(folds) %>%
    map(~ pull(., ClaimAmount))

  gamma_fit_ls <- map(
    1:2,
    ~ glm(ClaimAmount ~ ., family = Gamma(link = "log"), data = dat[folds != .x, ], offset = log(Exposure))
  )

  predictions_ls <- map(1:2, ~ predict(gamma_fit_ls[[.x]], newdata = dat[folds == .x, ], type = "response"))
```

```r
  res <- map2(responses_ls, predictions_ls, ~ mean((.x - .y) ^ 2)) %>%
    flatten_dbl() %>%
    mean()

  return(res)
}
```

Ensuite, ajuster les modèles pour K = 1, . . . , 10 et regarder quelle valeur mène à la plus petite erreur quadratique moyenne.

```r
(MSEs <- map_dbl(1:10, ~ mse_2_folds_gamma(data_drv_age, var = "DrivAge", k = .x)))
```

```
##  [1] 4.312187e+06 4.300823e+06 4.301520e+06 4.319982e+06 4.311656e+06
##  [6] 4.329723e+06 4.317668e+06 4.386675e+06 2.145221e+20 4.306240e+06
```

```r
which.min(MSEs)
```

```
## [1] 2
```

Avec le critère de l'erreur quadratique moyenne de 2-validation croisée, on choisit donc K = 2.

d)

```r
ridge_cv <- cv.glmnet(
  x = as.matrix(data[c("DrivAge", "LicAge")]),
  y = as.matrix(data["ClaimAmount"]),
  nfolds = 10,
  alpha = 0,
  family = "poisson",
  offset = log(as.matrix(data["Exposure"]))
)

coef(ridge_cv)
```

```
## 3 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) 8.042976e+00
## DrivAge     1.168809e-36
## LicAge      2.096173e-38
```

```r
ridge_cv$lambda.min
```

```
## [1] 18373.48
```

La valeur de lambda sélectionnée est 1.8373481×10^4.

e)

```r
ridge_cv_2 <- cv.glmnet(
  x = model.matrix(~ VehUsage - 1, data = data),
  y = as.matrix(data["ClaimAmount"]),
  nfolds = 20,
  alpha = 0,
  family = "poisson",
  offset = log(as.matrix(data["Exposure"]))
)
```

Les valeurs des paramètres obtenus sont:

```r
coef(ridge_cv_2)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                                             1
## (Intercept)                      8.042976e+00
## VehUsagePrivate                  9.612068e-35
## VehUsagePrivate+trip to office   8.361128e-35
## VehUsageProfessional            -1.952646e-34
## VehUsageProfessional run        -2.097956e-34
```