ML Homework

2023-04-09

C'est parti

European Social Survey (ESS) est une enquête réalisée tous les deux ans qui vise à mesurer et comprendre les attitudes, les croyances et les valeurs des personnes en Europe. Cette enquête est menée dans plus de 30 pays et couvre un large éventail de sujets, tels que la confiance sociale et politique, l'inégalité, la migration, la santé et le bien-être.

L'ESS fournit des informations précieuses sur le climat social et politique en Europe et permet de mieux comprendre les enjeux actuels dans ces domaines.

L'objectif est d'aller sur ce même site, trouver un dataset qui répond aux éxégences (déjà citée dans le fichier de devoir) Pour commencer, il nous faut un dataset: celui que nous avons choisi est: **ESS round 9 - 2018. Timing of life, Justice and fairness**

1)

On va charger le fichier CSV:

```
ess_data <- read.csv("ESS9e03_1.csv", header = TRUE, na.strings = 999)
```

un head pour voir ce qu'il y a dedans.

```
head(ess_data[,1:8], 3)
```

```
##        name essround edition   proddate idno cntry   dweight   pspwght
## 1 ESS9e03_1        9     3.1 17.02.2021   27    AT 0.5811743 0.2181114
## 2 ESS9e03_1        9     3.1 17.02.2021  137    AT 1.0627724 0.4134733
## 3 ESS9e03_1        9     3.1 17.02.2021  194    AT 1.3765086 2.2702928
```

```
# pour tout voir il vous faut uncommenter la ligne ci-dessous
#head(ess_data, 3)
```

Après avoir fouillé le fichier (pas tout, mais assez de temps pour trouver des VARs que nous jugeons bonnes et pertinentes), nous avons décidé de partir sur les variables suivantes:

wkhtot : Total hours normally worked per week in main job overtime included.

grspnum : What is your usual gross pay.

cntry : Country.

explication

Ok maintenant, il va falloir expliquer pourquoi ces VARs, comme cité ci-dessus, **wkhtot** fait référence à la durée de travail par semaine, et **grspnum** correspond au salaire brut de la personne intérogée.

Donc on voudrais savoir, s'il y a bien une relation (**ou pas**) entre ces deux variables dans la vraie vie, et comment l'une influence l'autre, ou bien tout simplement on n'arrive pas à trouver une relation entre les deux.

pour simplifier les choses, on se pose la question:

Est-ce en travaillant plus, on gagne plus ?

cette question revient souvent en politique et voici un exemple:

comme disait Sarkozy "Il faut laisser ceux qui veulent travailler plus pour gagner plus le faire"

ou bien comme disaint Mélenchon: "Il faut travailler moins pour travailler tous".

pour optimiser les requetes et ne pas avoir besoin de charger toutes les colonnes, il serait bon de garder que celles qui nous intéressent, par conséquent:

```
data = subset(ess_data, select = c("wkhtot",  "grspnum" , "cntry"))
```

Après avoir inspecter les colonnes en questions, il parait que certaines contiennes des valeurs d'exception, pae ex 666666666, 777777777 ... qui sont déjà définies dans le fichier téléchargé.

Donc il va falloir les éliminer pour ne pas tomber dans le piege de valeurs extreme qui pourraient surgir à l'une des extrémités de la distribution. (l'ex de Bernard Arnault).

Pour ce faire on met en pratique nos compétences limitées en R, mais qui fonctionnent bien sûr.

Chacune de ces colonnes contient des valeurs extrèmes qui sont en réalité des chiffres codés pour dire : 666 Not applicable*.

777 Refusal*.

888 Don't know*.

999 No answer*.

```
# data <- subset(data, !data$stfeco %in% c(77, 88, 99))
data <- subset(data, !data$grspnum %in% c(666666666, 777777777, 888888888, 999999999))
data <- subset(data, !data$wkhtot %in% c(666, 777, 888, 999))
```

Maintenant que les valeurs extrèmes sont éliminées, il nous est demandé de diviser (split) le dataset en deux parties afin de pouvoir appliquer une régression linéaire sur l'ensemble de variables choisies.

la 1ere (80%) pour l'entrainement, et la 2eme pour tester (20%).

```
train_indices <- sample(nrow(data), floor(0.8*nrow(data)))

train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]
```

2)

Il parait que tout est prêt pour qu'on fasse une regression linéaire. pour ce faire on fait appel à la fonction **lm()** en R.

mais avant cela, il serait génial si on s'arrete une minute pour comprendre ce concept.

en simple mots, on plotte nos données, on observe le nuage de points que produit ce nuage, on essaie de trouver un pattern que l'on puisse représenter par une droite, de type

$$Y = A * X + B$$

le **X** étant le variable indépendante, ou bien explicative dans ce cas la durée de travail qu'un employée puisse faire.

le **A** est un chiffre qui représente graphiquement la pente ou l'inclinaison de la droite mais en d'autre terme Elle correspond à la variation de la valeur de Y lorsque X augmente/ diminue d'une unité.

C'est à dire, dans ce cas, que se passe-t-il si on travaille plus, ou moins, quel impacte aurait ce comprentement sur la variable Y qui le salaire gagné.

et puis vient le **B** : ceci représente la valeur de la variable dépendante lorsque la variable indépendante est égale à zéro.

et Finalement **Y**, ce qu'on appelle la variable dépendante ou expliquée, dans ce scénario le salaire gagné.

maintenant que ceci est expliqué, on continue et on applique ceci sur notre nuage de point, en appellant la fonction **lm()**.

remarque: on applique la regression linèaire sur le training dataset. dans ce contexte, on aimerait représenter la Variable **grspnum** qui est le salaire par la variable indépendante **wkhtot** la durée de travail. autrement dit, on cherche à trouver une relation entre les deux qui puisse s'exprimer comme ceci

$$grspnum = A * wkhtot + B$$

```
model <- lm(grspnum~wkhtot , data = train_data )
```

3)

il parait que le model est produit, prêt à être utiliser. allons voir ce que contient ce dernier en appliquant le **summary()** sur ce model.

```
summary(model)
```

```
##
## Call:
## lm(formula = grspnum ~ wkhtot, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -185606  -55387  -49512  -25173 9990868
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  15487.0     5286.0   2.930   0.0034 **
## wkhtot        1015.0      124.1   8.176 3.18e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209800 on 13857 degrees of freedom
```

```
##   (237 observations deleted due to missingness)
## Multiple R-squared:  0.004801,   Adjusted R-squared:  0.00473
## F-statistic: 66.85 on 1 and 13857 DF,  p-value: 3.177e-16
```

on utilise la fonction summary pour obtenir un petit résumé de notre model de régression linéaire. cela comprend la formule du modèle qui montre la variable dépendante **Y** et variable explicative **X**, et aussi les valeurs des coefficients de régression estimés, l'erreur standard, les t-values et les p-values associées pour chaque variable prédictive dans le modèle.

mais si on regarde bien, dans la section coefficients, on remarque que **l'intercept** et **wkhtot** cela veut dire que le modele puisse s'ecrire de la façon suivante:

$$Y = A * X + B$$

4)

Y : variable expliquée (dépendante) => salaire.

X : variable explicative (indépendante) les heures de travail.

Intecept: 1223.80.

la pente (slope) : 29.58

Ok, c'est bien d'avoir généré un modele mais à quoi sert-il s'il ne predit rien. pour tester, on va utiliser le dataset de test pour voir si ce modele est capable de prédire le salaire à partir d'une certaine durée de travail / semaine

On utilise la fonction **predict** en passant le dataset de test.

6)

```
predictions <- predict(model, newdata = test_data)
```

predict contient les Y / salaire predit par le modele.

pour tester on peut dire, combien touche un employée qui fait 40 heures de travail en général. et ça donne :

```
prediction_40_heures_sem <- predict(model, newdata = data.frame(wkhtot = 40))
prediction_40_heures_sem
```

```
##        1
## 56086.75
```

Quand on voit ce chiffre on a limpression qu'on est dans un monde parallel, mais en réalité ceci est dû aux valeurs extrèmes qu'on a dans le dataset, encore une fois on n'est pas tous des Bernard Arnault ni Elon Musk, on est des gens normaux.

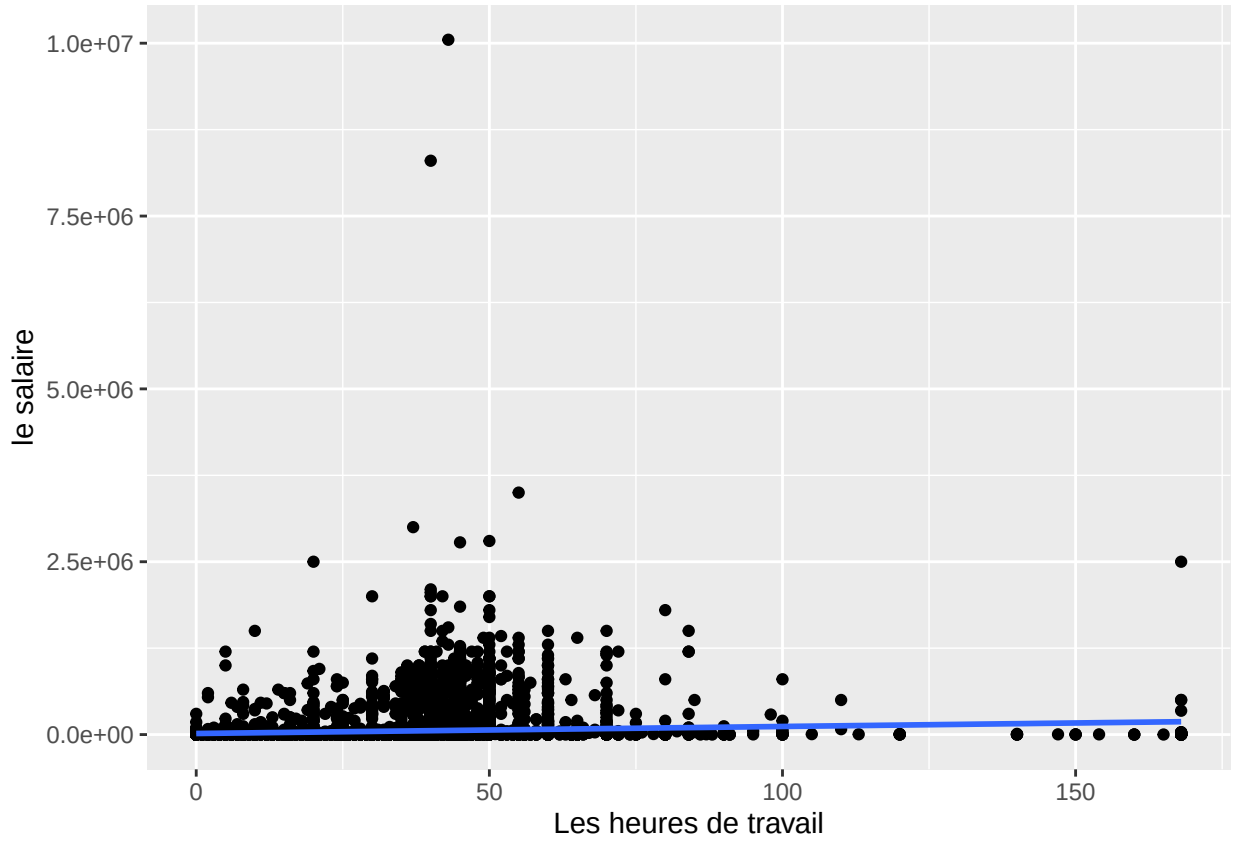
pour visualiser ce qui ce passe, on préfere faire appel à la Bib ggplot.

```
ggplot(train_data, aes(x = wkhtot, y = grspnum )) +
  geom_point() +
  stat_smooth(method = "lm") +
   xlab("Les heures de travail") + ylab("le salaire")
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 237 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 237 rows containing missing values (`geom_point()`).
```



une chose qui semble anormale, est que la durée de travail par semaine est un peu exagérée, car il ne peut exister nulle part un être humain qui puisse travailler 160 heures par semaine, à moins qu'il soit un robot, d'ailleurs même Elon musk ne puisse faire que 80 H par semaines.

pour assurer qu'on parle de gens normaux qui touche entre 900 et 9000 on va se limiter cette fois-ci à 9000 euro comme salaire max et 900 comme salaire Min.

et en même temps, on suppose que la personne intérogée ait le cerveau de Elon msuk (mais pas sa fortune) on va se limiter à 80 heures par semaine.

```
data <- data[data$grspnum <= 9000,]
data <- data[data$grspnum > 980,]


data <- data[data$wkhtot <= 80,]
```

encore un petit paramétrage, pour assurer que tout est bon dès le début.

```
train_indices <- sample(nrow(data), floor(0.8*nrow(data)))

train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]
```

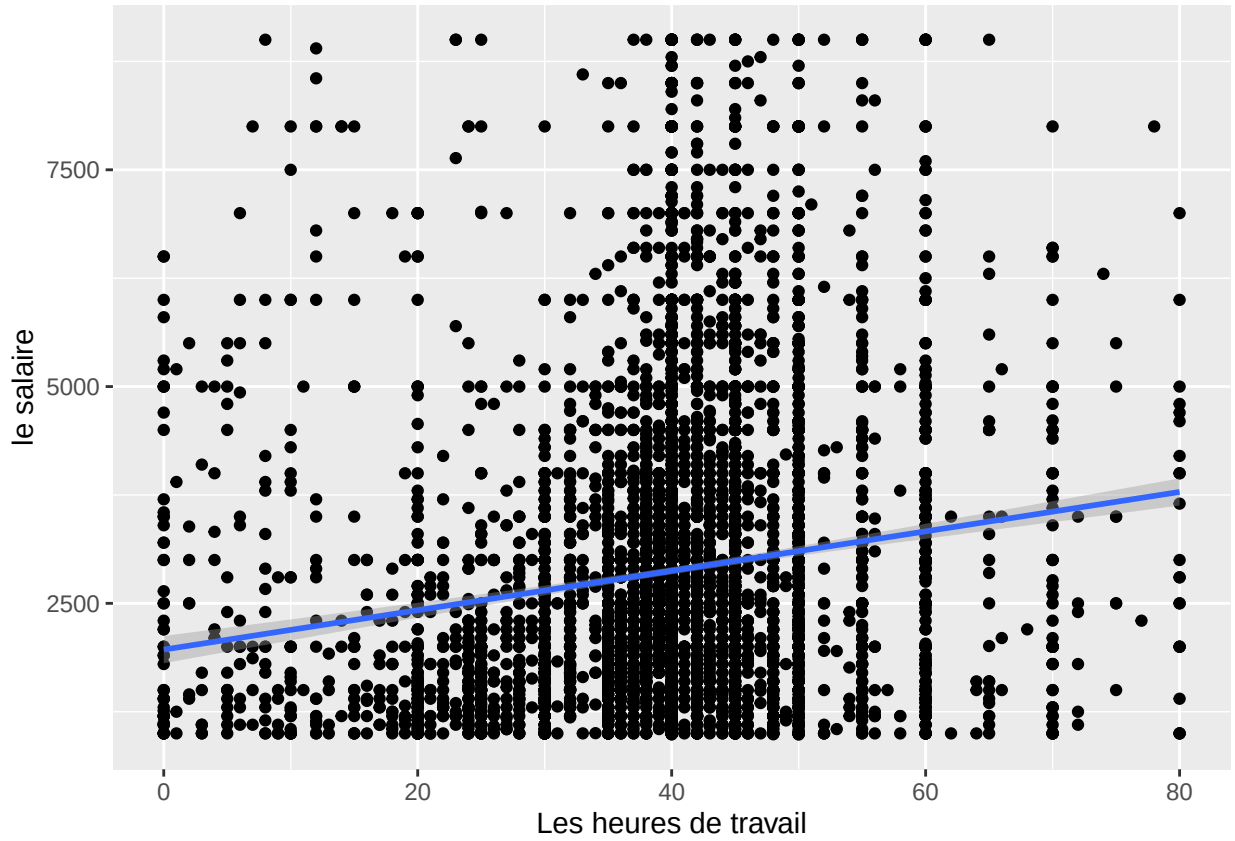
maiantenant que les paramaetres sont en place on replot et on analyse :

```
ggplot(train_data, aes(x = wkhtot, y = grspnum )) +
  geom_point() +
  stat_smooth(method = "lm") +
   xlab("Les heures de travail") + ylab("le salaire")
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 159 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 159 rows containing missing values (`geom_point()`).
```



encore une fois on regénère le modele pour voir si celui-ci s'adapte aux nouveaux params

ce modele s'ecrit comme ceci: les coefficient générés par la fonction summary (intercept et wkhtot)

```
model <- lm(grspnum~wkhtot , data = train_data )
summary(model)
```

```
##
## Call:
## lm(formula = grspnum ~ wkhtot, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2784.9 -1212.1  -473.9   710.6  6851.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1966.591     79.984   24.59   <2e-16 ***
## wkhtot        22.728      1.928   11.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1696 on 6572 degrees of freedom
##   (159 observations deleted due to missingness)
## Multiple R-squared:  0.02071,    Adjusted R-squared:  0.02056
## F-statistic:   139 on 1 and 6572 DF,  p-value: < 2.2e-16
```

et donne comme résultat pour les 40 H de travail par semaine :

```
prediction_40_heures_sem <- predict(model, newdata = data.frame(wkhtot = 40))
prediction_40_heures_sem
```

```
##        1
## 2875.724
```

Pas mal comme résultat, car ceci correspond presque à la réalité

un autre example pour finir : combien touche une personne qui travaille 30 heures par semaine

```
prediction_40_heures_sem <- predict(model, newdata = data.frame(wkhtot = 30))
prediction_40_heures_sem
```

```
##       1
## 2648.44
```

remarque IMPORTANTE

ce dataset couvre différents domaine au sein de l'union européenne, par conséquent comprend l'ensemble des pays européens Cela veut dire que certains pays ont une économie plus développée que d'autres, mais pas forcément une meilleure vie. le salaire moyen les aides sociales aussi qui different d'un pays à un autre, ce pour cela on trouve que certains travaillent pour la même période et gagnent moins.

en Turquie par ex le salaire moyen est de 600 euro, c'est peu pour un français mais raisonnable pour un turc.

Modele Français

pour justifier cela, on peut garder qu'un seul pays, dans ce cas on veut bien garder la france et on remet tous les params à jour

```r
data <- data[data$cntry == "FR",]
data <- data[data$grspnum <= 9000,]
data <- data[data$grspnum > 980,]
data <- data[data$wkhtot <= 80,]

train_indices <- sample(nrow(data), floor(0.8*nrow(data)))

train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]
```

on applique le **lm()** avec la nouvelle configuration.

```r
model <- lm(grspnum~wkhtot , data = train_data )
summary(model)
```

```
##
## Call:
## lm(formula = grspnum ~ wkhtot, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2127.6  -648.6  -280.0   339.7  6614.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  901.882    244.950   3.682 0.000259 ***
## wkhtot        37.095      6.172   6.010 3.78e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1076 on 460 degrees of freedom
##   (164 observations deleted due to missingness)
## Multiple R-squared:  0.0728, Adjusted R-squared:  0.07079
## F-statistic: 36.12 on 1 and 460 DF,  p-value: 3.783e-09
```

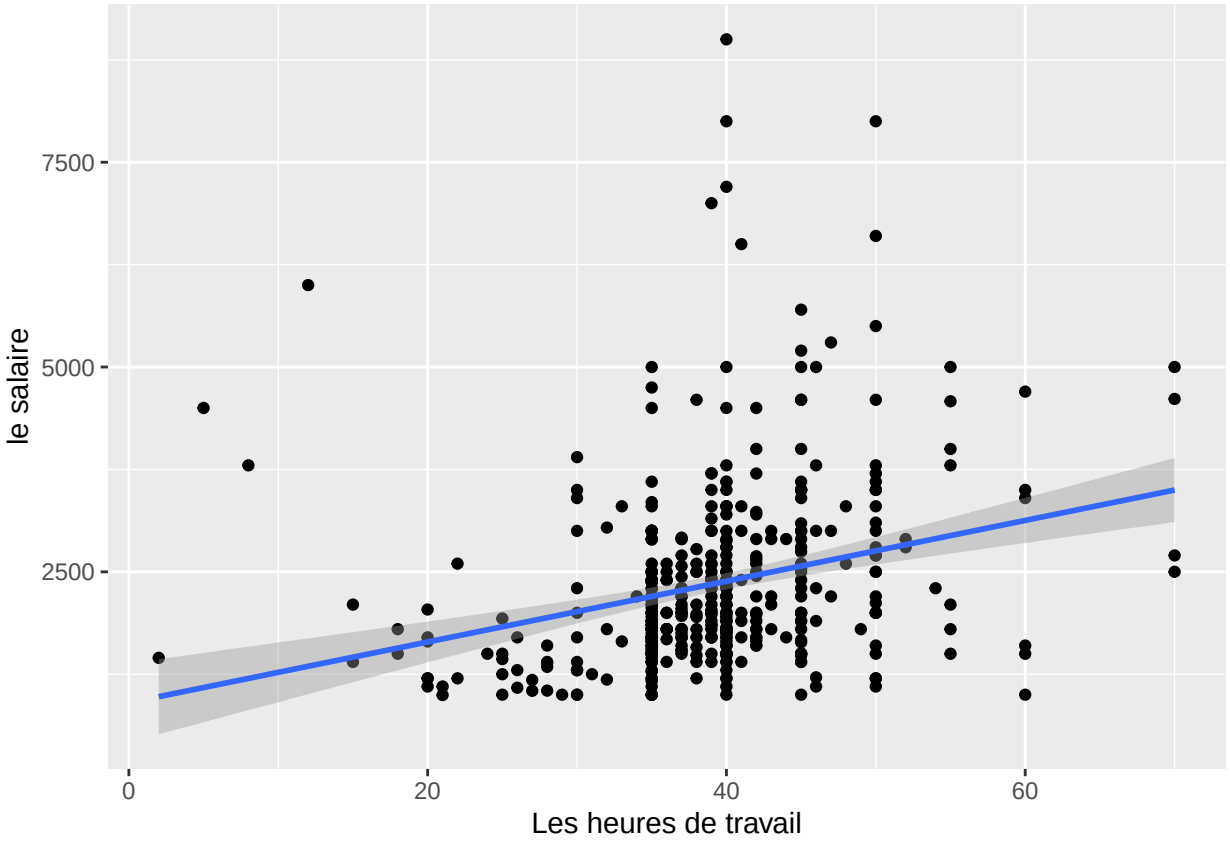
on visualise le modele.

```r
ggplot(train_data, aes(x = wkhtot, y = grspnum )) +
  geom_point() +
  stat_smooth(method = "lm") +
   xlab("Les heures de travail") + ylab("le salaire")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 164 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 164 rows containing missing values (`geom_point()`).
```



allons tester ce modele (**France**) avec une durée de 35 heures, cela donne:

```
prediction_35_heures_FR <- predict(model, newdata = data.frame(wkhtot = 30))
prediction_35_heures_FR
```

```
##        1
## 2014.736
```

Conclusions

Bien que ce modèle donne des résultats qui sont proches de la réalité, on remarque certaines anomalies telle que:

Pour la même durée de travail, des gens touchent moins que d'autres, ou l'inverse , c'est à dire ils touchent beaucoup plus par rapport à la moyenne, ceci peut se justifier :

Certains travaux rapportent beaucoup plus que d'autres, un médecin qui travaille 40 heures et touche 8000 euro, son saaire est loin d'etre comparé à celui d'un caissier qui travaille pour la même période et touche 1700 euro ou moins.

certes, travailler plus peut faire gagner plus, mais pas dans toutes les situations Donc le salaire aussi dépdend de la nature du travaille.

7)

Méthode alternative

Une méthode de régression qui permet une interprétation facile du rôle de chaque prédicteur est la régression Lasso.

Dans la régression Lasso, l'algorithme pénalise la taille absolue des coefficients, ce qui a pour effet de réduire certains coefficients à zéro, cela peut aider à identifier les variables les plus importantes dans la prédiction de la variable de réponse.

Cela a pour effet de réaliser une sélection de fonctionnalités, ce qui signifie que seuls les prédicteurs les plus importants sont conservés dans le modèle.