

biostat

camille mathilde

22/12/2020

# Introduction

De nos jours, les maladies sont de plus en plus étudiées et de mieux en mieux comprises. De nombreux organismes et instituts recherchent des solutions pour vaincre ces maladies en trouvant des traitements. De nombreuses disciplines sont impliquées, notamment la biostatistique, qui permet d'étudier différentes situations avec des données. Dans notre cas, nous allons étudier l'occurrence des maladies coronariennes du cœur. Une étude préalable a déjà été faite sur une cohorte d'individus. Ces individus ont répondu à une date donnée à une enquête sur les habitudes alimentaires. Les réponses ont été recueillies dans une base de données que l'on nomme **Coeur**.

## Présentation de la base de données

Notre table de données contient 337 individus et 15 variables qui sont

- **Id** : l'identifiant du sujet.
- **DateEntrée** et **Date de sortie** : les dates d'entrée et de sortie de l'étude.
- **Date Naissance** : la date de naissance.
- **Statut** : si la sortie de l'enquête est due à une maladie coronarienne de cœur, alors le type de maladie est indiqué (la signification du code n'est pas précisée ici). Si l'individu est sain à la sortie de l'enquête, alors le code vaut 0.
- **Emploi** : le type d'emploi.
- **MoisEnquête** : le mois (1= Janvier, 12= Décembre) où l'individu a répondu à l'enquête sur ses pratiques alimentaires.
- **Taille/Poids** : la taille et le poids de l'individu (en cm et en kg).
- **Graisse** : la quantité moyenne de graisse ingérée par jour (g/jour).
- **Fibres** : la quantité moyenne de fibres ingérée par jour (g/jour).
- **Consommation** : la quantité de calories(/100) ingérée par jour.
- **hauteConsommation** : une variable binaire, recodage de la variable consommation.
- **MCC** : une variable binaire, recodage de la variable statut (1=MCC, 0=pas de MCC).

Nous avons également rajouté la variable **IMC** en divisant le poids par la taille au carré pour faire un lien entre la condition physique et la maladie du cœur car le poids ou la taille tout seul ne suffisent pas pour savoir si une personne est en bonne santé ou en surpoids.

Nous nous sommes également rendu compte qu'il y avait des erreurs dans le recodage de la variable **statut**. En effet, certains individus avaient contracté une maladie du cœur, mais la variable recodage **MCC** ne l'avait pas pris en compte nous avons donc rectifié ça. Nous avons aussi remarqué que certaines variables qualitatives étaient en **numeric**, ce qui posera problème pour notre étude. Nous les recodons donc en **factor**. Nous observons enfin que la table de données contient des valeurs manquantes. Nous enlevons donc chaque ligne qui contient au moins une valeur manquante. Nous passons donc de 337 à 328 individus.

Regardons les 5 premières lignes de notre table de données :

```
coeur <- readRDS('data/my_data_frame.rds')
coeur <- coeur %>% drop_na()
coeur <- coeur %>% dplyr::select(-X1)
coeur$statut <- as.factor(coeur$statut)
coeur$emploi <- as.factor(coeur$emploi)
```

```

coeur$moisEnqu_e<-as.factor(coeur$moisEnqu_e)
coeur$hauteConsomation<-as.factor(coeur$hauteConsomation)

coeur<-mutate(coeur,imc =poids /(taille/100)^2)
coeur<-coeur %>% dplyr::select(-MCC)
coeur <-mutate(coeur,MCC =case_when(
  statut!=0~1,
  TRUE~0))

pander(head(coeur))

```

Table 1: Table continues below

id	dateEntree	dateSortie	dateNaissance	statut	emploi
102	17/01/76	02/12/86	02/03/39	0	Driver
59	16/07/73	05/07/82	05/07/12	0	Driver
126	17/03/70	20/03/84	24/12/19	13	Conductor
16	16/05/69	31/12/69	17/09/06	3	Driver
247	16/03/68	25/06/79	10/07/18	13	Bank worker
272	16/03/69	13/12/73	06/03/20	3	Bank worker

Table 2: Table continues below

moisEnqu_e	consommation	taille	poids	graisse	fibre
1	22.86	181.6	88.18	9.168	1.4
7	23.88	166	58.74	9.651	0.935
3	24.95	152.4	49.9	11.25	1.248
5	22.24	171.2	89.4	7.578	1.557
3	18.54	177.8	97.07	9.147	0.991
3	20.31	175.3	61.01	8.536	0.765

hauteConsomation	imc	MCC
<=2750 KCals	26.74	0
<=2750 KCals	21.32	0
<=2750 KCals	21.48	1
<=2750 KCals	30.51	1
<=2750 KCals	30.71	1
<=2750 KCals	19.86	1

Remarquons que cet ensemble d'individu est bien une cohorte car nous avons relevé certaines covariables et les trois données fondamentales qui sont, la date d'entrée dans l'étude, la date de sortie dans l'étude et le cause de sortie dans l'étude. Nous pouvons ajouter que les covariables utilisées dans l'étude sont fixe.

## Statistique descriptive

Notre jeu de donnée présente 5 variables quantitatives et 9 variables qualitatives.

Faisons un sommaire des variables quantitatives :

consommation	fibre	graisse	taille	poids
Min. :17.48	Min. :0.605	Min. : 7.26	Min. :152.4	Min. : 46.72
1st Qu.:25.46	1st Qu.:1.367	1st Qu.:11.15	1st Qu.:168.9	1st Qu.: 64.64
Median :28.11	Median :1.679	Median :12.60	Median :173.0	Median : 72.80
Mean :28.35	Mean :1.723	Mean :12.76	Mean :173.4	Mean : 72.40
3rd Qu.:31.10	3rd Qu.:1.939	3rd Qu.:14.02	3rd Qu.:177.8	3rd Qu.: 79.44
Max. :43.96	Max. :5.351	Max. :21.63	Max. :190.5	Max. :106.14

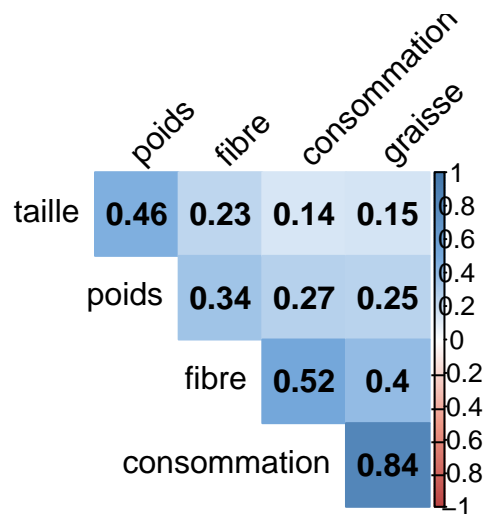
Les individus mangent en moyenne 2835 calories par jours. Ils ingèrent de plus 12.76 grammes de gras par jour, mesurent 173 cm et pèsent 72.40 kilo-gramme en moyenne.

Faisons maintenant un sommaire des variables qualitatives :

statut	emploi	moisEnqu_e	hauteConsomation
0 :252	Bank worker:147	11 : 39	<=2750 KCals:149
3 : 18	Conductor : 83	1 : 37	>2750 KCals :179
13 : 18	Driver : 98	3 : 37	NA
12 : 12	NA	2 : 34	NA
5 : 10	NA	5 : 34	NA
1 : 8	NA	12 : 33	NA
(Other): 10	NA	(Other):114	NA

Il y a 2 fois plus de **Bank worker** que de **Conductor** ou de **Driver**. Il y a 149 personnes qui mangent moins de 2750 calories par jours et 179 personnes qui en mangent plus.

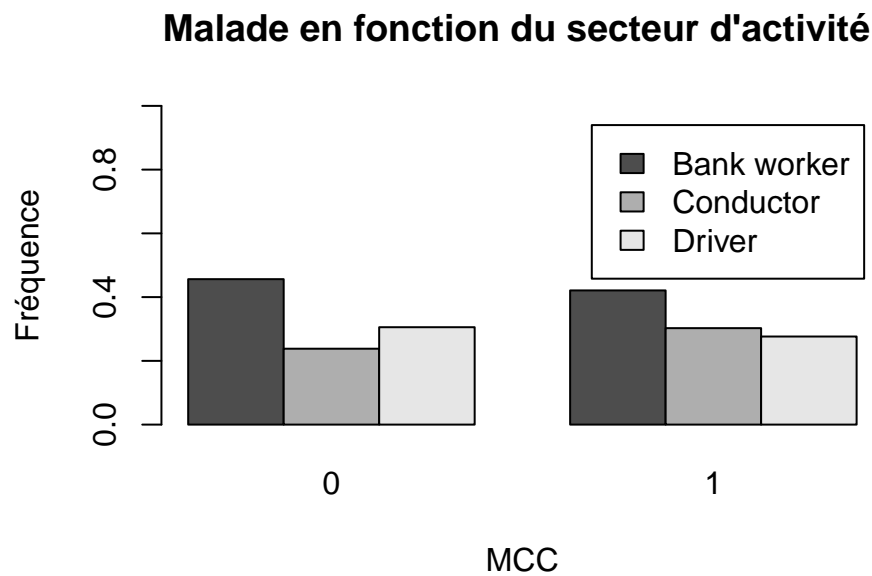
Regardons désormais la corrélation entre les variables quantitatives :



Nous observons que toutes les variables sont corrélées positivement.

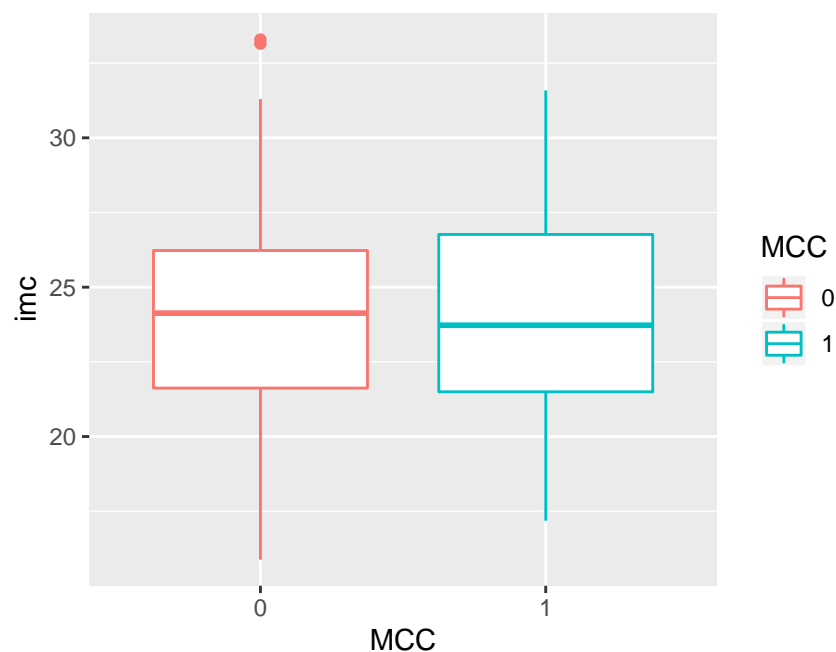
La variable **consommation** est très corrélée avec la variable **graisse** mais elle est très peu corrélée avec la variable **taille**. La variable **taille** est peu corrélée avec la variable **graisse**.

Nous allons maintenant regarder le lien entre la maladie et le secteur d'activité.



Nous observons que la proportion de conducteur est plus élevée chez les malades que chez les non malades.

Nous allons maintenant regarder le lien entre l'IMC et les maladies coronariennes.



Nous remarquons que la médiane de l'IMC du groupe de personnes malades est à peine plus basse que la médiane pour le groupe de personnes non malade. Une simple analyse descriptive ne suffit pas pour obtenir des résultats bien concluants, nous allons continuer avec des méthodes plus poussées. Nous commencerons par des régressions logistiques.

## regression logistique binaire

Nous allons commencer par une régression logistique qui nous permettra d'expliquer la variable **MCC** en fonction de toutes les autres covariables. Rappelons que dans la régression logistique ce n'est pas la réponse binaire qui est modélisée mais la probabilité de réalisation d'une des deux modalités (avoir une maladie coronarienne ou non).

Nous allons commencer par regarder le meilleur modèle en comparant les AIC puis nous étudierons ce modèle.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.6.3
```

```
##
```

```
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      select
```

```
coeurlogbin<-coeur%>%dplyr::select(-id,-dateEntree,-dateSortie,-dateNaissance,-statut,-hauteConsommation)
```

```
res<-glm(MCC~.,family = binomial(logit),data=coeurlogbin)
```

```
stepAIC(res)
```

```
## Start:  AIC=377.8
```

```
## MCC ~ emploi + moisEnqu_e + consommation + taille + poids + graisse +
```

```
##      fibre + imc
```

```
##
```

```
##
```

```
## - moisEnqu_e    Df Deviance    AIC
```

```
## - emploi        2   342.97 360.97
```

```
## - taille        1   337.85 375.85
```

```
## - consommation  1   337.86 375.86
```

```
## - imc           1   338.15 376.15
```

```
## - poids         1   338.16 376.16
```

```
## - graisse       1   338.24 376.24
```

```
## - fibre         1   338.73 376.73
```

```
## <none>          337.80 377.80
```

```
##
```

```
## Step:  AIC=360.97
```

```
## MCC ~ emploi + consommation + taille + poids + graisse + fibre +
```

```
##      imc
```

```
##
```

```
##
```

```
## - emploi        2   343.90 357.90
```

```
## - taille        1   342.97 358.97
```

```
## - consommation  1   342.99 358.99
```

```
## - poids         1   343.13 359.13
```

```
## - imc           1   343.14 359.14
```

```

## - graisse      1   343.27 359.27
## - fibre        1   343.96 359.96
## <none>          342.97 360.97
##
## Step: AIC=357.9
## MCC ~ consommation + taille + poids + graisse + fibre + imc
##
##           Df Deviance    AIC
## - consommation  1   343.90 355.90
## - taille        1   343.91 355.91
## - poids         1   344.08 356.08
## - imc           1   344.09 356.09
## - graisse       1   344.13 356.13
## - fibre         1   344.68 356.68
## <none>          343.90 357.90
##
## Step: AIC=355.9
## MCC ~ taille + poids + graisse + fibre + imc
##
##           Df Deviance    AIC
## - taille      1   343.91 353.91
## - poids       1   344.09 354.09
## - imc         1   344.09 354.09
## - graisse    1   344.51 354.51
## - fibre      1   344.79 354.79
## <none>        343.90 355.90
##
## Step: AIC=353.91
## MCC ~ poids + graisse + fibre + imc
##
##           Df Deviance    AIC
## - graisse    1   344.52 352.52
## - fibre      1   344.79 352.79
## <none>        343.91 353.91
## - imc       1   349.27 357.27
## - poids     1   350.12 358.12
##
## Step: AIC=352.52
## MCC ~ poids + fibre + imc
##
##           Df Deviance    AIC
## - fibre      1   346.17 352.17
## <none>        344.52 352.52
## - imc       1   349.90 355.90
## - poids     1   351.05 357.05
##
## Step: AIC=352.17
## MCC ~ poids + imc
##
##           Df Deviance    AIC
## <none>        346.17 352.17
## - imc       1   351.99 355.99
## - poids     1   355.01 359.01

```

```
##
## Call: glm(formula = MCC ~ poids + imc, family = binomial(logit), data = coeurlogbin)
##
## Coefficients:
## (Intercept)      poids      imc
##   -0.68843    -0.07396    0.19954
##
## Degrees of Freedom: 327 Total (i.e. Null); 325 Residual
## Null Deviance:      355.1
## Residual Deviance: 346.2    AIC: 352.2

reslog<-glm(MCC~imc,family = binomial(logit),data=coeurlogbin)
reslog
```

```
##
## Call: glm(formula = MCC ~ imc, family = binomial(logit), data = coeurlogbin)
##
## Coefficients:
## (Intercept)      imc
##   -0.88265    -0.01315
##
## Degrees of Freedom: 327 Total (i.e. Null); 326 Residual
## Null Deviance:      355.1
## Residual Deviance: 355    AIC: 359

confint(reslog)
```

## Waiting for profiling to be done...

```
##              2.5 %    97.5 %
## (Intercept) -2.85833907 1.0777684
## imc         -0.09488249 0.0677505
```

Les modalités de références sont :

- Pour la variable **emploi** : Bank worker
- Pour la variable **moisEnquete** : 1 (Janvier)

## régression polytomique ordonné

```
coeur <-mutate(coeur,consom_dec =case_when(
  consommation<23~"peu",
  consommation>22 & consommation<30~"moyen",
  consommation>29 ~"beaucoup"))
coeur$consom_dec<-as.factor(coeur$consom_dec)

coeurlogbin1<-coeur%>%dplyr::select(-id,-dateEntree,-dateSortie,-dateNaissance,-statut,-hauteConsomation)
modele<-polr(consom_dec~.,data=coeurlogbin1)
stepAIC(modele)

## Start:  AIC=325.09
## consom_dec ~ emploi + moisEnqu_e + taille + poids + graisse +
##      fibre + imc + MCC
##
##              Df    AIC
## - moisEnqu_e 11 316.92
```



```

## - emploi      2 322.04
## - MCC         1 323.09
## - poids       1 324.30
## - taille      1 324.43
## - imc         1 324.47
## <none>        325.09
## - fibre       1 343.66
## - graisse     1 532.55
##
## Step: AIC=316.92
## consom_dec ~ emploi + taille + poids + graisse + fibre + imc +
##      MCC
##
##           Df    AIC
## - emploi   2 313.65
## - MCC      1 314.95
## - poids    1 316.07
## - taille   1 316.08
## - imc      1 316.20
## <none>     316.92
## - fibre    1 337.12
## - graisse  1 520.53
##
## Step: AIC=313.65
## consom_dec ~ taille + poids + graisse + fibre + imc + MCC
##
##           Df    AIC
## - MCC      1 311.71
## - taille   1 312.55
## - poids    1 312.62
## - imc      1 312.71
## <none>     313.65
## - fibre    1 334.19
## - graisse  1 518.76
##
## Step: AIC=311.71
## consom_dec ~ taille + poids + graisse + fibre + imc
##
##           Df    AIC
## - taille   1 310.59
## - poids    1 310.66
## - imc      1 310.74
## <none>     311.71
## - fibre    1 332.24
## - graisse  1 517.42
##
## Step: AIC=310.59
## consom_dec ~ poids + graisse + fibre + imc
##
##           Df    AIC
## - poids    1 308.69
## - imc      1 308.94
## <none>     310.59
## - fibre    1 332.25

```

```
## - graisse 1 515.77
##
## Step: AIC=308.69
## consom_dec ~ graisse + fibre + imc
##
##           Df    AIC
## - imc      1 307.08
## <none>      308.69
## - fibre    1 330.53
## - graisse  1 514.32
##
## Step: AIC=307.08
## consom_dec ~ graisse + fibre
##
##           Df    AIC
## <none>      307.08
## - fibre    1 331.70
## - graisse  1 515.49
##
## Call:
## polr(formula = consom_dec ~ graisse + fibre, data = coeurlogbin1)
##
## Coefficients:
##   graisse   fibre
## -1.289518 -1.808077
##
## Intercepts:
## beaucoup|moyen      moyen|peu
##      -20.93183      -14.60442
##
## Residual Deviance: 299.079
## AIC: 307.079
```

```
modele2<-polr(consom_dec~graisse+fibre,data=coeurlogbin1)
modele2
```

```
## Call:
## polr(formula = consom_dec ~ graisse + fibre, data = coeurlogbin1)
##
## Coefficients:
##   graisse   fibre
## -1.289518 -1.808077
##
## Intercepts:
## beaucoup|moyen      moyen|peu
##      -20.93183      -14.60442
##
## Residual Deviance: 299.079
## AIC: 307.079
```

```
confint(modele2)
```

```
## Waiting for profiling to be done...
##
## Re-fitting to get Hessian
```

```
##           2.5 %    97.5 %
## graisse -1.566429 -1.051114
## fibre   -2.599604 -1.074953
```

La commande `polr` utilisée renvoie l'opposé du coefficient  $\beta$  considéré donc nous obtenons les résultats suivants pour nos coefficients:

#### QUESTION SUR LES MOINS

$$\ln\left(\frac{\text{odds}(Y \leq \text{moyen} | \text{graisse} = x_1 + 1, \text{fibre} = x_2)}{\text{odds}(Y \leq \text{moyen} | \text{graisse} = x_1, \text{fibre} = x_2)}\right) = \beta_{\text{graisse}} = 1.29$$

A consommation de fibre fixée, augmenter la consommation de graisse de 1 g/jour va diviser  $\text{odds}(Y \leq \text{moyen})$  par au moins  $\exp(-1.56)$ . QUEST

$$\ln\left(\frac{\text{odds}(Y \leq \text{moyen} | \text{fibre} = x_1 + 1, \text{graisse} = x_2)}{\text{odds}(Y \leq \text{moyen} | \text{fibre} = x_1, \text{graisse} = x_2)}\right) = \beta_{\text{fibre}} = 1.81$$

A consommation de graisse fixée, augmenter la consommation de fibre de 1 g/jour va diviser  $\text{odds}(Y \leq \text{moyen})$  par au moins  $\exp(-2.599)$ . QUEST

## régression polytomique non-ordonné

```
library(nnet)
```

```
## Warning: package 'nnet' was built under R version 3.6.3
```

```
modele3<-multinom(emploi~.,data=coeurlogbin1)
```

```
## # weights:  63 (40 variable)
## initial  value 360.344831
## iter   10 value 302.374870
## iter   20 value 278.075103
## iter   30 value 277.745433
## iter   40 value 276.134417
## iter   50 value 275.672548
## final   value 274.952497
## converged
```

```
stepAIC(modele3)
```

```
## Start:  AIC=629.9
## emploi ~ moisEnqu_e + taille + poids + graisse + fibre + imc +
##       MCC + consom_dec
##
## # weights:  30 (18 variable)
## initial  value 360.344831
## iter   10 value 316.508406
## iter   20 value 298.066550
## iter   30 value 296.225944
## iter   40 value 295.433298
## iter   50 value 295.415665
## iter   60 value 295.405218
## iter   70 value 295.366644
```

```

## final value 295.357814
## converged
## # weights: 60 (38 variable)
## initial value 360.344831
## iter 10 value 296.204644
## iter 20 value 277.918392
## iter 30 value 277.759987
## iter 40 value 277.757814
## final value 277.757807
## converged
## # weights: 60 (38 variable)
## initial value 360.344831
## iter 10 value 321.862774
## iter 20 value 292.668891
## iter 30 value 277.873218
## iter 40 value 276.441638
## final value 276.435260
## converged
## # weights: 60 (38 variable)
## initial value 360.344831
## iter 10 value 288.834223
## iter 20 value 278.639603
## iter 30 value 278.477928
## iter 40 value 276.589255
## iter 50 value 275.948755
## final value 275.686537
## converged
## # weights: 60 (38 variable)
## initial value 360.344831
## iter 10 value 308.528179
## iter 20 value 280.866090
## iter 30 value 280.611759
## iter 40 value 278.518118
## iter 50 value 277.754389
## final value 277.526923
## converged
## # weights: 60 (38 variable)
## initial value 360.344831
## iter 10 value 311.237508
## iter 20 value 288.482132
## iter 30 value 277.516004
## iter 40 value 276.426566
## final value 276.393089
## converged
## # weights: 60 (38 variable)
## initial value 360.344831
## iter 10 value 302.811283
## iter 20 value 278.319607
## iter 30 value 278.018976
## iter 40 value 276.463528
## iter 50 value 275.848419
## final value 275.189810
## converged
## # weights: 57 (36 variable)

```

```

## initial value 360.344831
## iter 10 value 303.628700
## iter 20 value 279.702097
## iter 30 value 279.473136
## iter 40 value 277.454786
## iter 50 value 276.662905
## final value 276.507102
## converged
##           Df      AIC
## - consom_dec  4 625.01
## - MCC         2 626.38
## - moisEnqu_e 22 626.72
## - graisse     2 627.37
## - imc         2 628.79
## - poids       2 628.87
## <none>        629.90
## - fibre       2 631.05
## - taille      2 631.52
## # weights: 57 (36 variable)
## initial value 360.344831
## iter 10 value 303.628700
## iter 20 value 279.702097
## iter 30 value 279.473136
## iter 40 value 277.454786
## iter 50 value 276.662905
## final value 276.507102
## converged
##
## Step: AIC=625.01
## emploi ~ moisEnqu_e + taille + poids + graisse + fibre + imc +
##      MCC
##
## # weights: 24 (14 variable)
## initial value 360.344831
## iter 10 value 314.759817
## iter 20 value 299.396193
## iter 30 value 297.548454
## iter 40 value 297.546035
## iter 50 value 296.954813
## iter 60 value 296.699244
## iter 70 value 296.692320
## iter 80 value 296.687415
## final value 296.686925
## converged
## # weights: 54 (34 variable)
## initial value 360.344831
## iter 10 value 289.231804
## iter 20 value 279.528995
## iter 30 value 279.490853
## final value 279.490564
## converged
## # weights: 54 (34 variable)
## initial value 360.344831
## iter 10 value 321.185922

```

```

## iter 20 value 290.648247
## iter 30 value 279.274078
## iter 40 value 278.146786
## final value 278.146660
## converged
## # weights: 54 (34 variable)
## initial value 360.344831
## iter 10 value 288.978592
## iter 20 value 281.695828
## iter 30 value 281.580465
## iter 40 value 279.686137
## iter 50 value 278.792376
## final value 278.792171
## converged
## # weights: 54 (34 variable)
## initial value 360.344831
## iter 10 value 304.140504
## iter 20 value 282.825680
## iter 30 value 282.614246
## iter 40 value 280.375572
## iter 50 value 279.328037
## final value 279.321947
## converged
## # weights: 54 (34 variable)
## initial value 360.344831
## iter 10 value 309.216075
## iter 20 value 287.667609
## iter 30 value 278.344704
## iter 40 value 278.103725
## final value 278.103377
## converged
## # weights: 54 (34 variable)
## initial value 360.344831
## iter 10 value 304.021337
## iter 20 value 279.863638
## iter 30 value 279.701265
## iter 40 value 277.671116
## iter 50 value 276.731259
## final value 276.731055
## converged
##           Df      AIC
## - moisEnqu_e 22 621.37
## - MCC         2 621.46
## - imc         2 624.21
## - poids       2 624.29
## <none>        625.01
## - graisse     2 625.58
## - fibre       2 626.64
## - taille      2 626.98
## # weights: 24 (14 variable)
## initial value 360.344831
## iter 10 value 314.759817
## iter 20 value 299.396193
## iter 30 value 297.548454

```

```

## iter 40 value 297.546035
## iter 50 value 296.954813
## iter 60 value 296.699244
## iter 70 value 296.692320
## iter 80 value 296.687415
## final value 296.686925
## converged
##
## Step: AIC=621.37
## emploi ~ taille + poids + graisse + fibre + imc + MCC
##
## # weights: 21 (12 variable)
## initial value 360.344831
## iter 10 value 302.046780
## final value 299.413690
## converged
## # weights: 21 (12 variable)
## initial value 360.344831
## iter 10 value 335.039006
## iter 20 value 298.498079
## final value 298.213880
## converged
## # weights: 21 (12 variable)
## initial value 360.344831
## iter 10 value 302.321679
## iter 20 value 299.941809
## final value 298.610827
## converged
## # weights: 21 (12 variable)
## initial value 360.344831
## iter 10 value 313.528594
## iter 20 value 300.878425
## final value 299.073481
## converged
## # weights: 21 (12 variable)
## initial value 360.344831
## iter 10 value 325.224947
## iter 20 value 298.292821
## final value 298.177326
## converged
## # weights: 21 (12 variable)
## initial value 360.344831
## iter 10 value 314.444586
## iter 20 value 299.446814
## final value 298.031618
## converged
##           Df    AIC
## - MCC      2 620.06
## - imc      2 620.35
## - poids    2 620.43
## - graisse  2 621.22
## <none>      621.37
## - fibre    2 622.15
## - taille   2 622.83

```

```

## # weights: 21 (12 variable)
## initial value 360.344831
## iter 10 value 314.444586
## iter 20 value 299.446814
## final value 298.031618
## converged
##
## Step: AIC=620.06
## emploi ~ taille + poids + graisse + fibre + imc
##
## # weights: 18 (10 variable)
## initial value 360.344831
## iter 10 value 302.048974
## final value 299.905371
## converged
## # weights: 18 (10 variable)
## initial value 360.344831
## iter 10 value 334.373478
## iter 20 value 298.727750
## final value 298.727139
## converged
## # weights: 18 (10 variable)
## initial value 360.344831
## iter 10 value 301.681838
## iter 20 value 299.546691
## iter 30 value 299.112795
## iter 40 value 298.527382
## iter 50 value 298.451213
## iter 60 value 298.256435
## final value 298.250675
## converged
## # weights: 18 (10 variable)
## initial value 360.344831
## iter 10 value 302.392570
## iter 20 value 299.890081
## iter 30 value 299.474015
## iter 40 value 298.799732
## iter 50 value 298.739425
## iter 60 value 298.533113
## final value 298.505150
## converged
## # weights: 18 (10 variable)
## initial value 360.344831
## iter 10 value 325.491389
## iter 20 value 298.698452
## final value 298.691152
## converged
##           Df    AIC
## - graisse  2 616.50
## - fibre    2 617.01
## - imc       2 617.38
## - poids    2 617.45
## - taille   2 619.81
## <none>      620.06

```



```

## # weights: 18 (10 variable)
## initial value 360.344831
## iter 10 value 301.681838
## iter 20 value 299.546691
## iter 30 value 299.112795
## iter 40 value 298.527382
## iter 50 value 298.451213
## iter 60 value 298.256435
## final value 298.250675
## converged
##
## Step: AIC=616.5
## emploi ~ taille + poids + fibre + imc
##
## # weights: 15 (8 variable)
## initial value 360.344831
## iter 10 value 300.952043
## final value 300.941512
## converged
## # weights: 15 (8 variable)
## initial value 360.344831
## iter 10 value 310.400830
## final value 299.823267
## converged
## # weights: 15 (8 variable)
## initial value 360.344831
## iter 10 value 302.896646
## iter 20 value 301.009332
## iter 30 value 300.090371
## iter 40 value 299.981044
## iter 50 value 299.786775
## final value 299.781681
## converged
## # weights: 15 (8 variable)
## initial value 360.344831
## iter 10 value 318.686408
## iter 20 value 299.771438
## iter 30 value 299.767701
## final value 299.767465
## converged
##           Df      AIC
## - imc      2 615.53
## - fibre    2 615.56
## - poids    2 615.65
## <none>      616.50
## - taille   2 617.88
## # weights: 15 (8 variable)
## initial value 360.344831
## iter 10 value 318.686408
## iter 20 value 299.771438
## iter 30 value 299.767701
## final value 299.767465
## converged
##

```

```

## Step: AIC=615.53
## emploi ~ taille + poids + fibre
##
## # weights: 12 (6 variable)
## initial value 360.344831
## iter 10 value 319.810141
## final value 319.810118
## converged
## # weights: 12 (6 variable)
## initial value 360.344831
## iter 10 value 312.168601
## iter 20 value 310.056817
## final value 310.032865
## converged
## # weights: 12 (6 variable)
## initial value 360.344831
## iter 10 value 302.773746
## iter 20 value 301.538107
## final value 301.526738
## converged
##           Df      AIC
## - fibre    2 615.05
## <none>      615.53
## - poids    2 632.07
## - taille   2 651.62
## # weights: 12 (6 variable)
## initial value 360.344831
## iter 10 value 302.773746
## iter 20 value 301.538107
## final value 301.526738
## converged
##
## Step: AIC=615.05
## emploi ~ taille + poids
##
## # weights: 9 (4 variable)
## initial value 360.344831
## final value 322.148152
## converged
## # weights: 9 (4 variable)
## initial value 360.344831
## iter 10 value 312.904158
## iter 20 value 312.078675
## final value 312.077839
## converged
##           Df      AIC
## <none>      615.05
## - poids    2 632.16
## - taille   2 652.30

## Call:
## multinom(formula = emploi ~ taille + poids, data = coeurlogbin1)
##
## Coefficients:

```

```
##           (Intercept)      taille      poids
## Conductor    34.13104 -0.1694273 -0.07620753
## Driver       22.11401 -0.1231291 -0.01424113
##
## Residual Deviance: 603.0535
## AIC: 615.0535

modele4<-multinom(emploi~poids+taille,data=coeurlogbin1)

## # weights:  12 (6 variable)
## initial value 360.344831
## iter  10 value 302.773746
## iter  20 value 301.538107
## final value 301.526738
## converged

modele4

## Call:
## multinom(formula = emploi ~ poids + taille, data = coeurlogbin1)
##
## Coefficients:
##           (Intercept)      poids      taille
## Conductor    34.13104 -0.07620753 -0.1694273
## Driver       22.11401 -0.01424113 -0.1231291
##
## Residual Deviance: 603.0535
## AIC: 615.0535

confint(modele4)

## , , Conductor
##
##           2.5 %      97.5 %
## (Intercept) 30.3445677 37.91750699
## poids      -0.1114239 -0.04099117
## taille     -0.1959546 -0.14290007
##
## , , Driver
##
##           2.5 %      97.5 %
## (Intercept) 16.95239383 27.27563010
## poids      -0.04357888  0.01509662
## taille     -0.15673368 -0.08952445
```

Bank worker est la modalité de référence pour emploi.

$$\beta_{poids|conductor} = \ln \left( \frac{\frac{P(conductor|poids=x_1+1,taille=x_2)}{P(Bankworker|poids=x_1+1,taille=x_2)}}{\frac{P(conductor|poids=x_1,taille=x_2)}{P(Bankworker|poids=x_1,taille=x_2)}} \right) = -0.07620753$$

QUESTION POUR LES MO DE REF X1 X2

Toutes les autres covariables fixées à leurs modalités de référence, un 1kg en plus va multiplier par au moins  $\exp(-0.1114239)$  = la préférence de conductor par rapport à Bank worker.

$$\beta_{taille|conductor} = \ln\left(\frac{P(conductor|taille=x_1+1,poids=x_2)}{P(Bankworker|taille=x_1+1,poids=x_2)}\right) = -0.1694273$$

## QUESTION POUR LES MO DE REF X1 X2

Toutes les autres covariables fixées à leurs modalités de référence, un 1cm en plus va multiplier par au moins  $\exp(-0.1959546)$  = la préférence de conductor par rapport à Bank worker.

## Modèle de Cox

Recodage au format date.

```
library(lubridate)

## Warning: package 'lubridate' was built under R version 3.6.2
##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##      date

coeur$dateEntree<-as.Date(coeur$dateEntree,format="%d/%m/%y")
year(coeur$dateEntree)<-1900+year(coeur$dateEntree) %% 100
coeur$dateSortie<-as.Date(coeur$dateSortie,format="%d/%m/%y")
year(coeur$dateSortie)<-1900+year(coeur$dateSortie) %% 100

coeur<-coeur%>% mutate(time = coeur$dateSortie - coeur$dateEntree )

coeurcox<-coeur%>%dplyr::select(-id,-dateEntree,-dateSortie,-dateNaissance,-statut,-hauteConsomation)
library(survival)

## Warning: package 'survival' was built under R version 3.6.3

survie=Surv(coeur$time,coeur$MCC)
# res=coxph(survie~.-moisEnqu_e,data=coeurcox)
# res
```

## DIAPO TRUC EN PLUS