Facteurs influençants la prise des transports en commun pour l'agglomération grenobloise

RACHIDI Mustapha & SAUNIER Florent & SAADALLAH Malek

Janvier 2023

```
#J'ai tenté l'image
knitr::include_graphics("Image_BUS_TRAM.jpg")
```

Introduction

Ce projet se base sur des données récoltées en 2010 dans la région Grenobloise. L'étude a pour but de déterminer les facteurs influençants la prise des transports en commun . Pour cela nous nous sommes pris comme limites : le réseau Mtag qui comprend les bus qualifiés de "ville" (Nous n'avons pas pris en compte les bus régionaux comme par exemple le bus Grenoble - Chamrousse) et le réseau du trammway dont les lignes depuis 2010 ont été augmentées.

Articles de la littérature

Familiarisation avec la base de données

La base de données contient 30 702 lignes et 116 colonnes ce qui correspond à nos variables , on peut la qualifier de base de données "moyenne" mais qui saura nous occuper. Concernant le nombre de valeurs manquantes, toutes variables confondues nous avons 971 658 valeurs manquantes soit 27.3% de notre base de données. De plus, 0% des lignes ont toutes leurs valeurs et c'est 21% des colonnes qui n'ont pas de valeurs manquantes. Il peut être intéressant de voir où sont les valeurs manquantes.
L'échantillon comporte 5189 personnes

Visualisation valeurs manquantes titre à changer peut ête

En annexe, quelques graphiques permettant de visualiser quelles variables ont le plus de valeurs manquantes. Ces graphiques nous permettrons d'adopter un regard critique sur les variables que nous choisirons par la suite. Cependant, on peut établir quelques critères avec r : ration de valeurs manquantes dans la colonne.

Bon : r<=5% Moyen : 5%<r<=20% Mauvais 20%<r<=45% Très mauvais : r>45%

Plusieurs variables ont entre 80% 99% de valeurs manquantes JAI TROUVÉ PQ c'est jusque que beaucoup de gens n'ont tout simplement pas plus de 1 véhicule, ce qui fait que les variables correspondantes sont vides. À CHANGER

```
sample <- sample(c(TRUE,FALSE),size=nrow(DB_projet_full),replace=TRUE,prob=c(0.7,0.3))
DB_BBB <-DB_projet_full[!sample,]
DB_BBB$freqtcu2<-DB_BBB$freqtcu
```

Variables du projet

Frecqtcu : Variable d'intérêt (Y) catégorielle qui indique la fréquence d'utilisation des transports en communs chez une personne.
Elle prend les valeurs :

1 : Utilisation des transports en commun tous les jours
2 : Utilisation des transports en commun au moins deux fois par semaine
3 : Utilisation des transports en commun au moins deux fois par mois
4: Utilisation des transports en commun très rare
5 : Utilisation des transports en commun inexistante

Nous avons décidié de construire frecqtcu de manière à ce qu'elle prenne la valeur 0 ou 1

```
DB_projet_full$freqtcu2<-DB_projet_full$freqtcu
 DB_projet_full<-DB_projet_full%>%mutate(freqtcu=ifelse(freqtcu<=3,1,0))
DB_projet_full$freqtcu<-factor(DB_projet_full$freqtcu)
```

Pour toutes les personnes qui prennent les transports de manière : régulière/tous les jours , au moins deux fois par semaine et au moins deux fois par mois se sont vues attribuées la valeur 1 car le "au moins" présage une prise des transports en communs plus élevée.

Tailmng : Variable qui indique le nombre de personnes composant le ménage.

```
DB_projet_full<-rename(DB_projet_full,"tailmng"="NO_PERS")
```

On change simplement le nom de la variable "NO_PERS" qui indique le nombre de personne dans le ménage
Permis :Variable indiquant si la personne effectuant le trajet possède le permis ou pas.

```
DB_projet_full<-DB_projet_full%>%mutate(permis=ifelse(any(permis==1 | permis==3),"YES","NO"))
DB_projet_full$permis<-factor(DB_projet_full$permis)
```

Car_ownership : Variable indiquant si la personne effectuant le trajet possède une voiture

```
DB_projet_full<-DB_projet_full%>%mutate(car_ownership=ifelse(DB_projet_full$VP_DISPO>0 & (DB_projet_ful

DB_projet_full$car_ownership<-factor(DB_projet_full$car_ownership)
```

Cette variable dépent de trois variables qui sont VP_dispo qui doit être strictement supérieur à 0, puis GENRE (type de véhicule utilisé) , nous avons exclu les campings cars car notre sujet se prête au milieu urbain et de POSSE (Est ce que la voiture appartient à la personne).Nous nous sommes contentés de prendre exclusivement les véhicules possedés par la personne.

Création de la nouvelle base de données

Variables complémentaires
Grâce aux variables précédentes et aux articles que l'on a trouvé dans la littérature, nous allons construire notre base de données pour notre modèle.
Nous exploiterons un ensemble de caractéristiques socio-économiques puis certaines variables liées au "confort" du trajet.

Restriction géographique Définissons ce que l'on entend par "transports en communs".
Pour notre étude nous nous concentrons sur les transports en communs de la société MTag,c'est à dire les tram et bus du réseau.
Notre délimitation géographique sera simplement les terminaux des différentes lignes de tram/bus confondues.
Par la suite, quand on parlera de transports en communs, on se refère à la définition au dessus.

Toutes les zones répertoriées dans le vecteur "Vec_zone" ont au moins un arrêt du réseau Mtag.

Restriction sur l'âge

Il est nécessaire de préciser que les mineurs se déplacent majoritairement via les transports en communs car ils n'ont tout simplement pas le choix...
Pour ne pas être biaisé, il est judicieux de filtrer les mineurs de notre base de données ainsi que les personnes âgées de plus de 80ans.

Notre nouvelle base de données comprend maintenant 10 879 observations et 22 variables ## Analyse Univariée

Analyse Univariée : freqtcu

Dans notre base de données, il y a 46% des gens qui prennent les transports en communs de manière plus ou moins régulière.
Analyse Univariée :permis

Toutes les personnes de notre échantillonnage possède le permis de conduire.
Analyse Univariée : tailmng

Pour ce qu'il en est de tailmng, la moyenne étant plus élevée que la médiane nous avons une asymétrie du côté droit , c'est à dire qu'il y a une concentration plus importante de valeurs à gauche de la moyenne.

Analyse Univariée : car_ownership

84.4% des gens qui ont le permis sont propriétaires d'un véhicule dans notre étude.
Pas de conclusions hâtives, cela sera explicité dans l'analyse bivariée.

Analyse Bivariée

Pour cette partie, nous allons faire appel à plusieurs tests statistiques pour tenter de comprendre les relations qu'il peut y avoir entre nos varialbes.
Le test statistique du Chi^2 est utile pour établir ou non une relation entre deux variables qualitatives.
Tandis que le test statistique d

```
DB_var_zg<-New_DB_filtered[!duplicated(New_DB_filtered$id_pers),]
#DB_var_zg<-allgreI faire avec allegreI
#DB_var_zg<-New_DB_filtered%>%group_by(tir) faire un group_by tir et trouver des variables relatives au
```

Création des variables pour l'analyse bivariée

```
#DB_var_zg : zone grenoble
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Plus_jeune=sum(min(age))) #variable plus jeune de la régi
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_actif=sum(OCCU1=="TravailPleinT" | OCCU1=="TravailPart"
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_inactif=sum(OCCU1=="Chomeur" | OCCU1=="Reste_auFoyer"
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_retraites=sum(OCCU1=="Retraite")) #Nb_retraités
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_etu=sum(OCCU1=="Scolaire" | OCCU1=="Etudiant" | OCCU1==

DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_log_collec=sum(TYPE_HAB=="GRAND_COLLECTIF" | TYPE_HAB==
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_log_indiv=sum(TYPE_HAB=="INDIVIDUEL_ISO" | TYPE_HAB=="

DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_proprietaire=sum(TYPE_OCU=="PROPRIETAIRE" | TYPE_OCU=="
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_locataire=sum(TYPE_OCU=="LOCATAIRE" | TYPE_OCU=="AUTRE_

DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_parking=sum(LIEU_STAT1=="PARKING PUBLIC" | LIEU_ST
#DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_garage=sum(LIEU_STAT1=="GARAGE/BOX"))
#DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_rue=sum(LIEU_STAT1=="RUE"))
```

```
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_parking=sum(LIEU_STAT1=="PARKING PUBLIC" | LIEU_ST
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_garage=sum(LIEU_STAT1=="GARAGE/BOX"))
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_rue=sum(LIEU_STAT1=="RUE"))

DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_interdit=sum(TYPE_STAT1=="INTERDIT"))
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_gratuit=sum(TYPE_STAT1=="GRATUIT" | TYPE_STAT1=="
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Nb_stat_payant=sum(TYPE_STAT1=="PAYANT"))

DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Haut_stat_social=sum(csp ==21 | csp==22 | csp==23 |csp==31
DB_var_zg<-DB_var_zg%>%group_by(tir)%>%mutate(Bas_stat_social=sum(csp==10 | csp==47 | csp==48 |csp==56
```

Permis

```
#On va se ref aux fréquences : Test de Fisher exact : permet de tester si les fréquences entières obser
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## v readr     2.1.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.3.2
```

```
library(rstatix) #chargement librairie
```

```
## Warning: package 'rstatix' was built under R version 4.3.2
```

```
##
## Attaching package: 'rstatix'
##
## The following object is masked from 'package:stats':
##
##     filter
```

```
##Permis
table(DB_var_zg$freqtcu,DB_var_zg$permis)
```

```
##
##      YES
##   0 1388
##   1 1125
```

```
chisq.test(table(DB_var_zg$freqtcu,DB_var_zg$permis)) #pour deux variables qualitatives
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$permis)
## X-squared = 27.524, df = 1, p-value = 1.551e-07
```

```
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$tailmng,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$tailmng, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4206 -1.3221  0.5794  1.5794  2.0720
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.51916    0.07135  49.319   <2e-16 ***
## DB_var_zg$tailmng -0.09854    0.04252  -2.317   0.0206 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.454 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.002134,   Adjusted R-squared:  0.001737
## F-statistic:  5.37 on 1 and 2511 DF,  p-value: 0.02056
```

```
#cor.test(table(New_DB_filtered$freqtcu,New_DB_filtered$permis))
```

tailmng

```r
#On va se ref aux fréquences : Test de Fisher exact : permet de tester si les fréquences entières obser
table(DB_var_zg$freqtcu,DB_var_zg$tailmng)
```

```
##
##       1   2   3   4   5   6
##   0 785 511  77  14   0   1
##   1 616 412  77  18   2   0
```

```r
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$tailmng)
)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$tailmng)
## p-value = 0.2005
## alternative hypothesis: two.sided
```

```r
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$tailmng,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$tailmng, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4206 -1.3221  0.5794  1.5794  2.0720
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.51916    0.07135  49.319   <2e-16 ***
## DB_var_zg$tailmng -0.09854    0.04252  -2.317   0.0206 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.454 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.002134,   Adjusted R-squared:  0.001737
## F-statistic:  5.37 on 1 and 2511 DF,  p-value: 0.02056
```

car_ownership

```r
table(DB_var_zg$freqtcu,DB_var_zg$car_ownership)
```

```
##
##      NON  OUI
##   0   91 1269
##   1  265  841
```

```r
chisq.test(table(DB_var_zg$freqtcu,DB_var_zg$car_ownership))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$car_ownership)
## X-squared = 145.87, df = 1, p-value < 2.2e-16
```

Autres variables

```r
table(DB_var_zg$freqtcu,DB_var_zg$Nb_actif) #Nb_actif
```

```
##
##       2   3   4   5   6   7   8   9  10  11
##   0  59 170 115 288 303 139 120  52  65  77
##   1  58 118  89 277 251 125  55  54  38  60
```

```r
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_actif),simulate.p.value=TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_actif)
## p-value = 0.004498
## alternative hypothesis: two.sided
```

```r
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$Nb_actif,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$Nb_actif, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4757 -1.3704  0.6086  1.5875  1.7138
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.24408    0.08110  40.002   <2e-16 ***
## DB_var_zg$Nb_actif   0.02105    0.01286   1.638    0.102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.455 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.001067,   Adjusted R-squared:  0.0006689
## F-statistic: 2.681 on 1 and 2511 DF,  p-value: 0.1016
```

```r
table(DB_var_zg$freqtcu,DB_var_zg$Nb_inactif) #Nb inactif
```

```
##
##       0   1   2   3   4   5   6   8   9  10  11  13
##   0  13 139  80 444 222 179  33  92  72  25  29  60
##   1  36  35 157 373 190 148  20  37  46  36  33  14
```

```r
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_inactif),simulate.p.value=TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_inactif)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```r
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$Nb_inactif,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$Nb_inactif, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7170 -1.3115  0.6074  1.4452  1.8102
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.18980    0.05313  60.038  < 2e-16 ***
## DB_var_zg$Nb_inactif  0.04055    0.01013   4.002 6.47e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.451 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.006337,   Adjusted R-squared:  0.005941
## F-statistic: 16.01 on 1 and 2511 DF,  p-value: 6.472e-05
```

```r
#Nb_retraite
table(DB_var_zg$freqtcu,DB_var_zg$Nb_retraites)
```

```
##
##       5   6   7   8   9  10  11  12  13  14  15  17  18  19  20  23  24  27
##   0  59  35  40 127  47  83  35 127 187 141  66 114  82  54  50  49  46  46
##   1  34  82   4  52  80  79  67  98 193 152  49  67  57  11  65   3  15  17
```

```r
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_retraites),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_retraites)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```r
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$Nb_retraites,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$Nb_retraites, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8535 -1.3172  0.4683  1.3968  1.9331
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.888103   0.083385  34.636  < 2e-16 ***
## DB_var_zg$Nb_retraites 0.035756   0.005829   6.134 9.91e-10 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.445 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.01476,    Adjusted R-squared:  0.01437
## F-statistic: 37.63 on 1 and 2511 DF,  p-value: 9.907e-10
```

```
#Nb_etu
table(DB_var_zg$freqtcu,DB_var_zg$Nb_etu)
```

```
##
##       1   2   3   4   5   6   7   8   9  10  11  13  14  15  16  56
##   0  49 119 149 250 202  39  90 151  99  34  26  24  41  67  13  35
##   1   3  64  67 115 116  31  92 149  95  31  43  40  79  63  55  82
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_etu),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_etu)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$Nb_etu,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$Nb_etu, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6061 -1.3492  0.4796  1.4510  2.9636
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.634593   0.037124   97.90   <2e-16 ***
## DB_var_zg$Nb_etu -0.028539   0.002567  -11.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.421 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.04692,    Adjusted R-squared:  0.04654
## F-statistic: 123.6 on 1 and 2511 DF,  p-value: < 2.2e-16
```

```
#Nb_log_col
table(DB_var_zg$freqtcu,DB_var_zg$Nb_log_collec)
```

```
##
##      4  6  8  9 15 16 19 27 28 29 30 35 38 43 44 46 47 48 50 51 52 53 55 59 60
##   0 39 40 51 46 54 86 86 72 33 49 89 23 20 20 36 39 31 46 57 24 51 49 65 26 25
##   1  7  2 19 15 11  7 50 40 20  3 42 25 28 27 28 60 65 22 56 31 69 15 54 43 36
##
##     61 62 63 66 74 94
```

```
##   0 53 13 39 31 60 35
##   1 73 55 88 38 14 82
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_log_collec),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_log_collec)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$Nb_log_collec,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$Nb_log_collec, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1501 -1.2110  0.1434  1.0847  2.6106
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.228321   0.063189   66.92   <2e-16 ***
## DB_var_zg$Nb_log_collec -0.019563   0.001291  -15.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.393 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.08384,    Adjusted R-squared:  0.08347
## F-statistic: 229.8 on 1 and 2511 DF,  p-value: < 2.2e-16
```

```
#Nb_log_ind
table(DB_var_zg$freqtcu,DB_var_zg$Nb_log_indiv)
```

```
##
##       0   1   2   3   4   6   9  11  13  16  17  20  21  22  23  25  29  31  33
##   0 142  53 117  95  93  72  20  83  36  46  51  41  33  35  31  40  39 133  54
##   1 141 114 188 156  79  40  28  34  28  22  12  13  20  82  27   4   7  24  11
##
##      34  40  49  52
##   0  38  39  46  51
##   1  30  31  15  19
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_log_indiv),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_log_indiv)
## p-value = 0.0004998
```

```
## alternative hypothesis: two.sided
```

```
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$Nb_log_indiv,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$Nb_log_indiv, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2157 -1.1261  0.2513  1.2513  1.9406
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.059365   0.039033   78.38   <2e-16 ***
## DB_var_zg$Nb_log_indiv 0.022237   0.001936   11.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.419 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.04991,    Adjusted R-squared:  0.04953
## F-statistic: 131.9 on 1 and 2511 DF,  p-value: < 2.2e-16
```

```
#Nb_propriétaire
table(DB_var_zg$freqtcu,DB_var_zg$Nb_proprietaire)
```

```
##
##       0   7  12  18  20  22  25  26  27  28  30  31  32  33  37  38  39  42  44
##   0  60  20  29  25  13  38  37  18  27  71 127  59 107  34  79  45  71  95  79
##   1  14  27  33  36  55  78  66  29  25  92 112  53  47  31  37  73  29  18  43
##
##      46  47  48  52  53  60  62
##   0  35  31  47  97  54  51  39
##   1  82  38  19  27  11  19  31
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_proprietaire),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_proprietaire)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$Nb_proprietaire,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$Nb_proprietaire,
##     data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7516 -1.3061  0.4433  1.3876  2.1115
```

```
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.888479   0.079356  36.399  < 2e-16 ***
## DB_var_zg$Nb_proprietaire 0.013921   0.002146   6.486 1.06e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.444 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.01648,    Adjusted R-squared:  0.01609
## F-statistic: 42.07 on 1 and 2511 DF,  p-value: 1.059e-10
```

```
#Nb_locataire
table(DB_var_zg$freqtcu,DB_var_zg$Nb_locataire)
```

```
##
##       0   1   2   3   4   5   8   9  10  11  12  13  14  15  16  17  18  19  20
##   0  98  46  91  95  64 127  51  75  49  62  31  36  76  47  26  22  20  34  59
##   1  70  15  21  82  67  35  12  36   3  32  27  28  40  19  43  42  28  25  53
##
##      21  22  23  29  30  38  60
##   0  70  57  46  34  24  13  35
##   1  97  67  22  84  40  55  82
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_locataire),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_locataire)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```
summary(lm(DB_var_zg$freqtcu2~DB_var_zg$Nb_locataire,data=DB_var_zg))
```

```
##
## Call:
## lm(formula = DB_var_zg$freqtcu2 ~ DB_var_zg$Nb_locataire, data = DB_var_zg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7948 -1.2101  0.3165  1.2633  2.8760
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.794834   0.042144   90.04   <2e-16 ***
## DB_var_zg$Nb_locataire -0.027847   0.002054  -13.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.405 on 2511 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.06823,    Adjusted R-squared:  0.06785
```

```
## F-statistic: 183.9 on 1 and 2511 DF,  p-value: < 2.2e-16
```

```
#Nb_stat_parking
table(DB_var_zg$freqtcu,DB_var_zg$Nb_stat_parking)
```

```
##
##       0   1   2   3   4   5   6   7   9  10  11  12  13  16  18  19  20  24  29
##   0  51  80 100  20 191  55  51 159 109 112  46  50  74  22  77  36  49  35  25
##   1  63  46  14  27  88  67  85 153  35 127  17  75  43  42  60  28  15  82  36
##
##      33
##   0  46
##   1  22
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_stat_parking),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_stat_parking)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```
#Nb_stat_garage
table(DB_var_zg$freqtcu,DB_var_zg$Nb_stat_parking)
```

```
##
##       0   1   2   3   4   5   6   7   9  10  11  12  13  16  18  19  20  24  29
##   0  51  80 100  20 191  55  51 159 109 112  46  50  74  22  77  36  49  35  25
##   1  63  46  14  27  88  67  85 153  35 127  17  75  43  42  60  28  15  82  36
##
##      33
##   0  46
##   1  22
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_stat_parking),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_stat_parking)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```
#Nb_stat_rue
table(DB_var_zg$freqtcu,DB_var_zg$Nb_stat_rue)
```

```
##
##       0   2   4   5   6   7   8  11  12  14  15  16  17  18  19  24  25
##   0  46  93 120 112 153  23 130  90  13 114 120  24  58  56  91  51  94
##   1  15  18  73  42 179  25  77  79  55 119 128  31  65  69  93  12  45
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_stat_rue),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_stat_rue)
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```
#Nb_stat_interdit
table(DB_var_zg$freqtcu,DB_var_zg$Nb_stat_interdit)
```

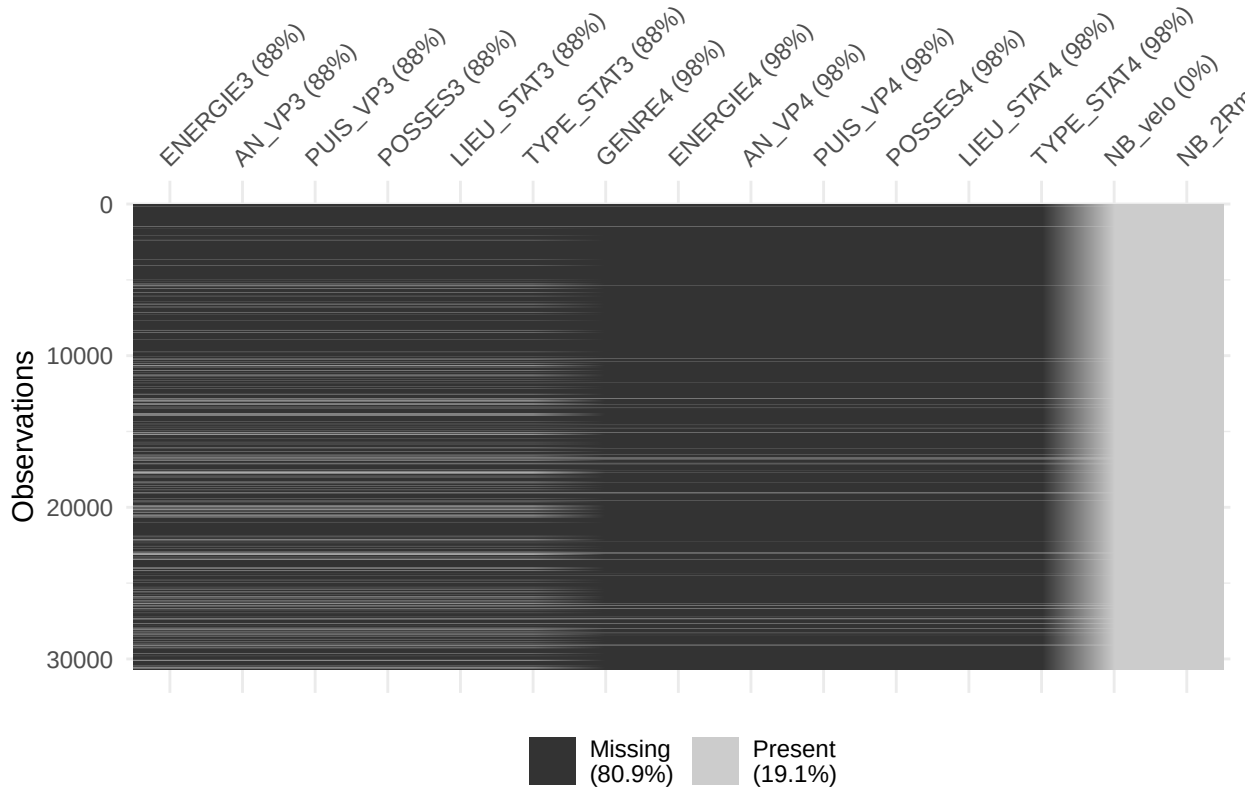
```
##
##        0    1    6
##   0 1335   33   20
##   1 1034   64   27
```

```
fisher.test(table(DB_var_zg$freqtcu,DB_var_zg$Nb_stat_interdit),simulate.p.value = TRUE
)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table(DB_var_zg$freqtcu, DB_var_zg$Nb_stat_interdit)
## p-value = 0.0009995
## alternative hypothesis: two.sided
```

Annexes

```
data_2<-DB_projet_full[,c(30:44)]
vis_miss(
  data_2,
  cluster = FALSE,
  sort_miss = FALSE,
  show_perc = TRUE,
  show_perc_col = TRUE,
  large_data_size = 9e+06,
  warn_large_data = TRUE
)
```



Listes variables à plus de 80% de valeurs manquantes

-motoracc -situveil -STAT_TRAV -TYPE_STAT4 -LIEU_STAT4 -POSSES4 -PUIS_VP4 -AN_VP4 -ENERGIE4 -GENRE4 -TYPE_STAT3 -LIEU_STAT3 -POSSES3 -PUIS_VP3 -AN_VP3 -ENERGIE3 -motdeacc -nbarret -abonpeage