

Rapport de biostatistique

Pierre et Florian

Sommaire

I. Introduction	3
II. Import des packages	4
III. Présentation des données	9
A. Variables quantitatives	9
B. Variables qualitatives	10
IV. Importation et sauvegarde de la base	14
V. Mécanismes utilisés pour effacer les données	15
VI. Analyse descriptive de la base de données	17
A. Visualisation globale des données	17
B. Analyse descriptive univariée	22
1. Variables quantitatives	22
2. Variables qualitatives	32
C. Analyse descriptive bivariée	38
1. Variables quantitatives - quantitatives	39
2. Variables qualitatives - qualitatives	71
3. Variables quantitatives - qualitatives	76
VII. Imputation des valeurs manquantes : trois techniques utilisées	182
A. Régression linéaire binaire	182
1. Visualisation des données avec des boîtes à moustaches pour FORMATION, PARTICULIERS, TUTORAT, BOURSE, EMPLOI, LOGEMENT, ARGENT, CAF ET GENRE	182

2. Imputation	193
B. Remplacement par la moyenne	242
1. Visualisation des données manquantes pour la variable MOYENNE . .	242
2. Imputation	243
C. MICE	248
VIII. Conclusion	266

I. Introduction

L'objectif de ce dossier est de traiter les valeurs manquantes dans notre base de données afin d'assurer une analyse fiable et complète. Pour ce faire, nous appliquerons trois méthodes d'imputation : la régression logistique binaire, le remplacement par la moyenne, et l'imputation multiple MICE (imputations multivariées par équations chainées).

La base de données sélectionnée est issue d'une étude visant à explorer l'impact de la gestion budgétaire sur les performances académiques des étudiants des Pays de la Loire (niveaux Bac+1 et au-delà) durant l'année universitaire 2023-2024. Afin d'examiner cette relation, un questionnaire a été diffusé auprès des étudiants via les réseaux sociaux et les messageries universitaires.

Ce jeu de données contient 7 variables quantitatives et 15 variables qualitatives, décrivant les caractéristiques de 135 étudiants.

II. Import des packages

```
library(kableExtra)
```

Warning: package 'kableExtra' was built under R version 4.4.3

```
library(knitr)
```

Warning: package 'knitr' was built under R version 4.4.3

```
library(openxlsx)
```

Warning: package 'openxlsx' was built under R version 4.4.3

```
library(car)
```

Warning: package 'car' was built under R version 4.4.3

Loading required package: carData

```
library(MASS)  
library(tidyverse)
```

Warning: package 'tidyverse' was built under R version 4.4.3

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
v dplyr     1.1.4     v readr     2.1.5  
v forcats   1.0.0     v stringr   1.5.1  
v ggplot2   3.5.1     v tibble    3.2.1  
v lubridate  1.9.4     v tidyr    1.3.1  
v purrr    1.0.2
```

```
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter()     masks stats::filter()
x dplyr::group_rows() masks kableExtra::group_rows()
x dplyr::lag()        masks stats::lag()
x dplyr::recode()     masks car::recode()
x dplyr::select()     masks MASS::select()
x purrr::some()       masks car::some()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts t
```

```
library(EnvStats)
```

Attaching package: 'EnvStats'

The following object is masked from 'package:MASS':

boxcox

The following object is masked from 'package:car':

qqPlot

The following objects are masked from 'package:stats':

predict, predict.lm

```
library(stats)
library(lmtest)
```

Warning: package 'lmtest' was built under R version 4.4.3

Loading required package: zoo

Warning: package 'zoo' was built under R version 4.4.3

Attaching package: 'zoo'

```
The following objects are masked from 'package:base':
```

```
as.Date, as.Date.numeric
```

```
library(PerformanceAnalytics)
```

```
Warning: package 'PerformanceAnalytics' was built under R version 4.4.3
```

```
Loading required package: xts
```

```
Warning: package 'xts' was built under R version 4.4.3
```

```
#####
# Warning from 'xts' package #####
#
# The dplyr lag() function breaks how base R's lag() function is supposed to #
# work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or #
# source() into this session won't work correctly. #
#
# Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
# conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #
# dplyr from breaking base R's lag() function. #
#
# Code in packages is not affected. It's protected by R's namespace mechanism #
# Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning. #
#
#####
```

```
Attaching package: 'xts'
```

```
The following objects are masked from 'package:dplyr':
```

```
first, last
```

```
Attaching package: 'PerformanceAnalytics'
```

```
The following objects are masked from 'package:EnvStats':
```

```
kurtosis, skewness
```

```
The following object is masked from 'package:graphics':
```

```
legend
```

```
library(corrplot)
```

```
Warning: package 'corrplot' was built under R version 4.4.3
```

```
corrplot 0.95 loaded
```

```
library(sjPlot)
```

```
Warning: package 'sjPlot' was built under R version 4.4.3
```

```
library(ggplot2)
library(leaps)
```

```
Warning: package 'leaps' was built under R version 4.4.3
```

```
library(AER)
```

```
Warning: package 'AER' was built under R version 4.4.3
```

```
Loading required package: sandwich
```

```
Warning: package 'sandwich' was built under R version 4.4.3
```

```
Loading required package: survival
```

```
library(naniar)
```

```
Warning: package 'naniar' was built under R version 4.4.3
```

```
library(grid)
library(mice)
```

Warning: package 'mice' was built under R version 4.4.3

Attaching package: 'mice'

The following object is masked from 'package:stats':

filter

The following objects are masked from 'package:base':

cbind, rbind

III. Présentation des données

A. Variables quantitatives

Dans les variables quantitatives, nous retrouvons la moyenne générale, MOYENNE, un indicateur qui synthétise les résultats obtenus dans l'ensemble des matières étudiées. Cet indicateur reflète la performance académique globale d'un étudiant.

Nous avons également le taux d'assiduité, ASSIDUITE, codé sur une échelle de 1 à 10 (1 correspondant à un étudiant qui ne fréquente jamais les cours et 10 à un étudiant qui les fréquente toujours).

La variable RESTAURANT représente le nombre moyen de restaurants ou fast-foods fréquentés mensuellement par un étudiant.

De plus, le niveau de stress financier, STRESS, codé également de 1 à 10, reflète la perception des étudiants quant à leur situation financière.

Enfin, nous trouvons les variables SOMMEIL et TRAJET : la première correspond au temps de sommeil moyen quotidien des étudiants, tandis que la seconde mesure le temps de trajet moyen nécessaire pour se rendre à leur établissement universitaire.

```
quantis <- data.frame(
  Variables = c("MOYENNE", "ASSIDUITE", "RESTAURANT", "AGE",
                "STRESS", "SOMMEIL", "TRAJET"),
  Définitions = c(
    "Moyenne générale de l'étudiant.",
    "Taux d'assiduité aux cours magistraux et aux travaux dirigés.",
    "Nombre de restaurants ou fast-foods par mois.",
    "Âge de l'étudiant.",
    "Degré de stress financier.",
    "Durée moyenne de sommeil par nuit.",
    "Temps de trajet pour se rendre à l'établissement."
  ),
  Modalités = c(
    "Nombres réels de 0 à 20.",
    "Nombres entiers de 1 (aucune) à 10 (extrêmement élevée).",
    "Nombres réels positifs.",
    "Nombres entiers positifs.",
    "Nombres entiers de 1 (aucun) à 10 (extrêmement élevé).",
    "Nombres réels positifs.",
    "Nombres entiers positifs."
```

```

),
  stringsAsFactors = FALSE
)

quantis |>
  kable("latex",
    booktabs = T,
    caption = "Variables quantitatives") |>
  kable_styling(
    latex_options = c("scale_down",
                      "HOLD_position",
                      full_width = TRUE,
                      font_size = 12)) |>
  column_spec(1:3,
              background = "cyan") |>
  row_spec(0,
            background = "blue",
            color = "white")

```

Table 1: Variables quantitatives

Variables	Définitions	Modalités
MOYENNE	Moyenne générale de l'étudiant.	Nombres réels de 0 à 20.
ASSIDUITE	Taux d'assiduité aux cours magistraux et aux travaux dirigés.	Nombres entiers de 1 (aucune) à 10 (extrêmement élevée).
RESTAURANT	Nombre de restaurants ou fast-foods par mois.	Nombres réels positifs.
AGE	Âge de l'étudiant.	Nombres entiers positifs.
STRESS	Degré de stress financier.	Nombres entiers de 1 (aucun) à 10 (extrêmement élevé).
SOMMEIL	Durée moyenne de sommeil par nuit.	Nombres réels positifs.
TRAJET	Temps de trajet pour se rendre à l'établissement.	Nombres entiers positifs.

B. Variables qualitatives

Dans les variables qualitatives, nous retrouvons FORMATION, qui correspond à la filière suivie par l'étudiant, comme économie et gestion ou d'autres domaines.

Nous avons également REVISIONS, une variable qui indique le nombre d'heures de révisions effectuées chaque jour en moyenne, classées en trois catégories : 0-1 heure, 1-2 heures, et 2 heures ou plus.

La variable PARTICULIERS renseigne si l'étudiant a suivi des cours particuliers durant ses études, avec deux modalités possibles : oui ou non.

De plus, la variable TUTORAT indique si l'étudiant a participé à des sessions de tutorat pendant l'année, également codée en oui ou non.

La variable BOURSE identifie les étudiants boursiers, avec les mêmes modalités : oui ou non.

Nous avons également EMPLOI, qui signale si l'étudiant occupe un emploi en parallèle de ses études.

La variable LOGEMENT fait référence au loyer payé par l'étudiant, et la variable ARGENT correspond au montant d'argent reçu en dehors des revenus issus d'un emploi étudiant, de la bourse ou des aides de la CAF.

En outre, la variable DEPENSES estime les dépenses mensuelles de l'étudiant dans les loisirs, réparties en deux catégories : moins de 100 euros et 100 euros ou plus.

La variable CAF mentionne si l'étudiant bénéficie d'aides de la CAF, et la variable TRANSPORT concerne le budget mensuel moyen consacré aux transports, classé en trois catégories : 0-15 euros, 15-30 euros, et 30 euros ou plus.

La variable GENRE reflète le genre de l'étudiant, avec les modalités homme et femme.

La variable STATUT représente le statut social des parents, classé en trois groupes : classe aisée, classe moyenne, et classe populaire.

La variable SANTE identifie si l'étudiant a rencontré des problèmes de santé durant l'année.

Enfin, la variable STRUCTURE indique le type de structure de formation fréquentée par l'étudiant, comme une université ou une autre institution.

```
qualis <- data.frame(  
  Variables = c("FORMATION", "REVISIONS", "PARTICULIERS",  
    "TUTORAT", "BOURSE", "EMPLOI",  
    "LOGEMENT", "ARGENT", "DEPENSES",  
    "CAF", "TRANSPORT", "GENRE",  
    "STATUT", "SANTE", "STRUCTURE"  
  ),  
  Définitions = c(  
    "Formation de l'étudiant.",  
    "Nombre d'heures de révisions par jour en moyenne.",  
    "Suivi de cours particuliers durant les études.",  
    "Participation à du tutorat pendant l'année.",  
    "Étudiant boursier.",  
    "Emploi étudiant",
```

```

    "Loyer payé par l'étudiant.",  

    "Argent reçu en dehors du travail étudiant, de la bourse et des aides de la CAF.",  

    "Estimation des dépenses mensuelles dans les loisirs.",  

    "Aides de la CAF.",  

    "Budget transport par mois en moyenne.",  

    "Genre de l'étudiant.",  

    "Statut social des parents.",  

    "Problèmes de santé durant l'année.",  

    "Structure de formation."  

),  

Modalités = c(  

  "Économie et gestion \\" Autres",  

  "0 - 1h \\" 1 - 2h \\" 2h et plus",  

  "Oui \\" Non",  

  "0 - 100 € \\" 100 € et plus",  

  "Oui \\" Non",  

  " 0 - 15 € \\" 15 - 30 € \\" 30 € et plus",  

  " Homme \\" Femme",  

  "Classe aisée \\" Classe moyenne \\" Classe populaire",  

  "Oui \\" Non",  

  "Université \\" Autres"  

),  

stringsAsFactors = FALSE  

)  

qualis |>  

  kable("latex",  

    booktabs = T,  

    caption = "Variables qualitatives") |>  

kable_styling(  

  latex_options = c("scale_down",  

                  "HOLD_position",  

                  full_width = TRUE,  

                  font_size = 12)) |>  

column_spec(1:3,

```

```

background = "cyan") |>
row_spec(0,
         background = "blue",
         color = "white")

```

Table 2: Variables qualitatives

Variables	Définitions	Modalités
FORMATION	Formation de l'étudiant.	Economie et gestion \ Autres
REVISIONS	Nombre d'heures de révisions par jour en moyenne.	0 - 1h \ 1 - 2h \ 2h et plus
PARTICULIERS	Suivi de cours particuliers durant les études.	Oui \ Non
TUTORAT	Participation à du tutorat pendant l'année.	Oui \ Non
BOURSE	Étudiant boursier.	Oui \ Non
EMPLOI	Emploi étudiant	Oui \ Non
LOGEMENT	Loyer payé par l'étudiant.	Oui \ Non
ARGENT	Argent reçu en dehors du travail étudiant, de la bourse et des aides de la CAF.	Oui \ Non
DEPENSES	Estimation des dépenses mensuelles dans les loisirs.	0 - 100 € \ 100 € et plus
CAF	Aides de la CAF.	Oui \ Non
TRANSPORT	Budget transport par mois en moyenne.	0 - 15 € \ 15 - 30 € \ 30 € et plus
GENRE	Genre de l'étudiant.	Homme \ Femme
STATUT	Statut social des parents.	Classe aisée \ Classe moyenne \ Classe populaire
SANTE	Problèmes de santé durant l'année.	Oui \ Non
STRUCTURE	Structure de formation.	Université \ Autres

IV. Importation et sauvegarde de la base

```
Budget <- read.xlsx("data/student_budget_data_2023_2024.xlsx")

write_csv(Budget,
          file = "Budget.csv")

View(Budget)
```

V. Mécanismes utilisés pour effacer les données

Pour mener à bien notre analyse et remplacer les valeurs manquantes, nous allons dans un premier temps supprimer des valeurs afin d'introduire aléatoirement des données manquantes (NA), reproduisant ainsi une situation de données incomplètes.

Pour ce faire, nous utilisons le code présenté ci-dessous.

Nous avons choisi de retirer 10 % des données de notre base, soit un total de 315 valeurs.

```
# Calcul du nombre total de cellules
cellules <- nrow(Budget) * ncol(Budget)

# Nombre de cellules à supprimer (10 %)
sup <- round(0.1 * cellules)

# Conversion de toutes les colonnes en caractères
Budget_chr <- Budget |>
  mutate(across(everything(), as.character))

# Copie sous forme longue
Budget_long <- Budget_chr |>
  pivot_longer(everything(),
    names_to = "variables",
    values_to = "valeurs") |>

# Identifiant de chaque variable
group_by(variables) |>
  mutate(ID = row_number()) |>
  ungroup()

# Sélection aléatoire de 10 % des cellules selon une distribution uniforme
set.seed(123)
indices <- integer(0)
while(length(indices) < sup) {
  indices <- unique(c(indices, round(runif(sup - length(indices), min = 1, max = nrow(
}))}

# Remplacement des cellules sélectionnées par NA
Budget_long_na <- Budget_long |>
  mutate(valeurs = case_when(row_number() %in% indices ~ NA_character_ ,
```

```

TRUE ~ valeurs))

# Format large
Budget2 <- Budget_long_na |>
  pivot_wider(names_from = variables,
              values_from = valeurs) |>
  select(-ID)

print(Budget2)

# A tibble: 135 x 22
  FORMATION ASSIDUITE REVISIONS PARTICULIERS TUTORAT MOYENNE BOURSE EMPLOI
  <chr>      <chr>     <chr>     <chr>      <chr>     <chr>     <chr>     <chr>
1 Economie et g~ 9   <NA>       Oui        Oui      15       Non      Non
2 Economie et g~ 8   <NA>       Non        Non     13.5     <NA>      Oui
3 Economie et g~ 10  2h et pl~ Non        Non      15       Oui      Oui
4 Economie et g~ 9   0 - 1h    Non        Oui      14       Non      <NA>
5 Economie et g~ 8   2h et pl~ Non        Non     15.8     Non      Non
6 Economie et g~ 10  0 - 1h    Non        Non      13       Non      Oui
7 Economie et g~ 8   1 - 2h    <NA>      <NA>     13.8     Non      Non
8 Autres            10        0 - 1h    <NA>      Non      16       Non      Oui
9 Economie et g~ 10  2h et pl~ Non        <NA>     13.4     Non      Non
10 Economie et g~ 5  0 - 1h    Non        Non      12.5     Non      Non
# i 125 more rows
# i 14 more variables: LOGEMENT <chr>, ARGENT <chr>, RESTAURANT <chr>,
# DEPENSES <chr>, CAF <chr>, TRANSPORT <chr>, GENRE <chr>, AGE <chr>,
# STRESS <chr>, STATUT <chr>, SOMMEIL <chr>, SANTE <chr>, STRUCTURE <chr>,
# TRAJET <chr>

write_csv(Budget2,
          file = "Budget2.csv")

```

Nous avons 135 individus et 22 variables, ce qui donne $22 * 135 = 2970$ cellules.

```
sum(is.na(Budget2))
```

```
[1] 297
```

Nous obtenons bien $0,1 * 2970 = 297$ valeurs manquantes.

VI. Analyse descriptive de la base de données

A. Visualisation globale des données

```
View(Budget2)
```

Vérifions la nature des variables.

```
str(Budget2)
```

```
tibble [135 x 22] (S3: tbl_df/tbl/data.frame)
$ FORMATION    : chr [1:135] "Economie et gestion" "Economie et gestion" "Economie et
$ ASSIDUITE    : chr [1:135] "9" "8" "10" "9" ...
$ REVISIONS    : chr [1:135] NA NA "2h et plus" "0 - 1h" ...
$ PARTICULIERS: chr [1:135] "Oui" "Non" "Non" "Non" ...
$ TUTORAT      : chr [1:135] "Oui" "Non" "Non" "Oui" ...
$ MOYENNE      : chr [1:135] "15" "13.5" "15" "14" ...
$ BOURSE        : chr [1:135] "Non" NA "Oui" "Non" ...
$ EMPLOI        : chr [1:135] "Non" "Oui" "Oui" NA ...
$ LOGEMENT      : chr [1:135] "Non" "Non" "Non" "Non" ...
$ ARGENT        : chr [1:135] "Oui" NA "Non" "Non" ...
$ RESTAURANT    : chr [1:135] "4" "2" "1" "4" ...
$ DEPENSES      : chr [1:135] "100 € et plus" "100 € et plus" "0 - 100 €" "0 - 100 €" ...
$ CAF           : chr [1:135] "Non" "Non" "Non" "Non" ...
$ TRANSPORT     : chr [1:135] "30 € et plus" "30 € et plus" "0 - 15 €" "15 - 30 €" ...
$ GENRE         : chr [1:135] "Homme" "Homme" "Femme" "Femme" ...
$ AGE           : chr [1:135] "22" "19" NA "20" ...
$ STRESS         : chr [1:135] "1" "2" "1" "2" ...
$ STATUT        : chr [1:135] "Classe aisée" "Classe moyenne" "Classe moyenne" "Classe
$ SOMMEIL       : chr [1:135] "7" "8" "7" "7" ...
$ SANTE          : chr [1:135] NA "Non" "Non" "Non" ...
$ STRUCTURE     : chr [1:135] "Université" "Université" "Université" "Université" ...
$ TRAJET         : chr [1:135] "10" "80" "35" "30" ...
```

Modifions le type de données des variables (passage de caractère à numérique ou facteur).

```

Budget2 <- read_csv(
  "Budget2.csv",
  col_types = c("fnffffnffffnffffnnfnffn")
)

str(Budget2)

```

`spc_tbl_` [135 x 22] (S3: spec_tbl_df/tbl_df/tbl/data.frame)

	\$ FORMATION	\$ ASSIDUITE	\$ REVISIONS	\$ PARTICULIERS	\$ TUTORAT	\$ MOYENNE	\$ BOURSE	\$ EMPLOI	\$ LOGEMENT	\$ ARGENT	\$ RESTAURANT	\$ DEPENSES	\$ CAF	\$ TRANSPORT	\$ GENRE	\$ AGE	\$ STRESS	\$ STATUT	\$ SOMMEIL	\$ SANTE	\$ STRUCTURE	\$ TRAJET
- attr(*, "spec")=	Factor w/ 2 levels "Economie et gestion",...: 1 1 1 1 1 1 1 2 1 1 ...	num [1:135] 9 8 10 9 8 10 8 10 10 5 ...	Factor w/ 3 levels "2h et plus","0 - 1h",...: NA NA 1 2 1 2 3 2 1 2 ...	Factor w/ 2 levels "Oui","Non": 1 2 2 2 2 2 NA NA 2 2 ...	Factor w/ 2 levels "Oui","Non": 1 2 2 1 2 2 NA 2 NA 2 ...	num [1:135] 15 13.5 15 14 15.8 13 13.8 16 13.4 12.5 ...	Factor w/ 2 levels "Non","Oui": 1 NA 2 1 1 1 1 1 1 1 ...	Factor w/ 2 levels "Non","Oui": 1 2 2 NA 1 2 1 2 1 1 ...	Factor w/ 2 levels "Non","Oui": 1 1 1 1 2 1 1 NA 2 2 ...	Factor w/ 2 levels "Oui","Non": 1 NA 2 2 1 2 1 1 1 1 ...	num [1:135] 4 2 1 4 2 2 NA 4 2 5 ...	Factor w/ 2 levels "100 € et plus",...: 1 1 2 2 2 1 1 1 2 2 ...	Factor w/ 2 levels "Non","Oui": 1 1 1 1 2 1 1 1 2 2 ...	Factor w/ 3 levels "30 € et plus",...: 1 1 2 3 2 3 3 1 1 3 ...	Factor w/ 2 levels "Homme","Femme": 1 1 2 2 2 2 2 2 2 1 ...	num [1:135] 22 19 NA 20 22 NA 21 20 24 NA ...	num [1:135] 1 2 1 2 3 8 4 3 5 NA ...	Factor w/ 3 levels "Classe aisée",...: 1 2 2 2 2 2 2 2 NA 2 ...	num [1:135] 7 8 7 7 8 7 7 NA 6 7 ...	Factor w/ 2 levels "Non","Oui": NA 1 1 1 1 2 1 2 2 1 ...	Factor w/ 2 levels "Université","Autres": 1 1 1 1 1 1 1 NA 1 1 ...	num [1:135] 10 80 35 30 30 60 30 60 45 25 ...
.. cols(FORMATION = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),	ASSIDUITE = col_number(),	REVISIONS = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),	PARTICULIERS = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),	TUTORAT = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),	MOYENNE = col_number(),	BOURSE = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),	EMPLOI = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),														

```

.. LOGEMENT = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. ARGENT = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. RESTAURANT = col_number(),
.. DEPENSES = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. CAF = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. TRANSPORT = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. GENRE = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. AGE = col_number(),
.. STRESS = col_number(),
.. STATUT = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. SOMMEIL = col_number(),
.. SANTE = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. STRUCTURE = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
.. TRAJET = col_number()
.. )
- attr(*, "problems")=<externalptr>

```

`glimpse(Budget2)`

```

Rows: 135
Columns: 22
$ FORMATION      <fct> Economie et gestion, Economie et gestion, Economie et ges-
$ ASSIDUITE      <dbl> 9, 8, 10, 9, 8, 10, 8, 10, 10, 5, NA, 8, 10, 9, 9, 5, ~
$ REVISIONS      <fct> NA, NA, 2h et plus, 0 - 1h, 2h et plus, 0 - 1h, 1 - 2h, 0~
$ PARTICULIERS   <fct> Oui, Non, Non, Non, Non, Non, NA, NA, Non, Non, Non, Non, ~
$ TUTORAT        <fct> Oui, Non, Non, Oui, Non, Non, NA, Non, NA, Non, Oui, Non, ~
$ MOYENNE         <dbl> 15.00, 13.50, 15.00, 14.00, 15.80, 13.00, 13.80, 16.00, 1~
$ BOURSE          <fct> Non, NA, Oui, Non, Non, Non, Non, Non, Non, Non, Non~
$ EMPLOI          <fct> Non, Oui, Oui, NA, Non, Oui, Non, Oui, Non, Non, Non~
$ LOGEMENT        <fct> Non, Non, Non, Non, Oui, Non, Non, NA, Oui, Oui, NA, Non, ~
$ ARGENT          <fct> Oui, NA, Non, Non, Oui, Non, Oui, Oui, Oui, Non, Oui~
$ RESTAURANT      <dbl> 4, 2, 1, 4, 2, 2, NA, 4, 2, 5, 2, 2, 2, 2, 2, NA, 3, 1~
$ DEPENSES        <fct> 100 € et plus, 100 € et plus, 0 - 100 €, 0 - 100 €, 0 - 1~
$ CAF             <fct> Non, Non, Non, Non, Oui, Non, Non, Oui, Oui, Non, Ou~
$ TRANSPORT       <fct> 30 € et plus, 30 € et plus, 0 - 15 €, 15 - 30 €, 0 - 15 €~
$ GENRE           <fct> Homme, Homme, Femme, Femme, Femme, Femme, Fem~
$ AGE             <dbl> 22, 19, NA, 20, 22, NA, 21, 20, 24, NA, 20, 20, 20, 23, 2~
$ STRESS          <dbl> 1, 2, 1, 2, 3, 8, 4, 3, 5, NA, 2, 4, 1, 1, 1, 1, 3, NA~
$ STATUT          <fct> Classe aisée, Classe moyenne, Classe moyenne, Classe moye~
$ SOMMEIL         <dbl> 7.00, 8.00, 7.00, 7.00, 8.00, 7.00, 7.00, NA, 6.00, 7.00, ~

```

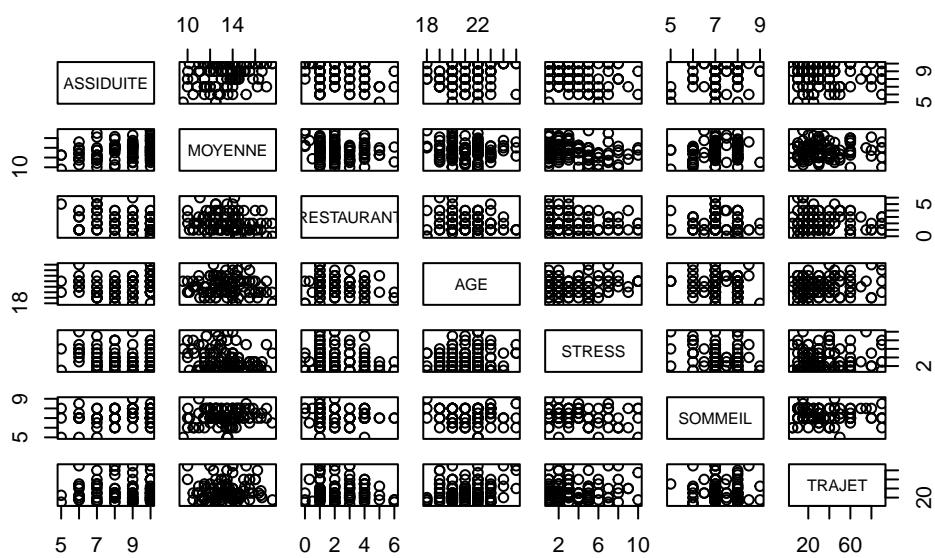
```
$ SANTE      <fct> NA, Non, Non, Non, Non, Oui, Non, Oui, Oui, Non, Non, Non~  
$ STRUCTURE   <fct> Université, Université, Université, Université, Universit~  
$ TRAJET       <dbl> 10, 80, 35, 30, 30, 60, 30, 60, 45, 25, 20, 20, 25, 10, 1~
```

```
save(Budget2, file="Budget.rda")
```

```
# Variables quantitatives  
  
quantis <- c("ASSIDUITE", "MOYENNE", "RESTAURANT", "AGE", "STRESS", "SOMMEIL", "TRAJET"  
  
# Variables qualitatives  
  
qualis <- c("FORMATION", "REVISIONS", "PARTICULIERS", "TUTORAT", "BOURSE",  
           "EMPLOI", "LOGEMENT", "ARGENT", "DEPENSES", "CAF", "TRANSPORT",  
           "GENRE", "STATUT", "SANTE", "STRUCTURE")
```

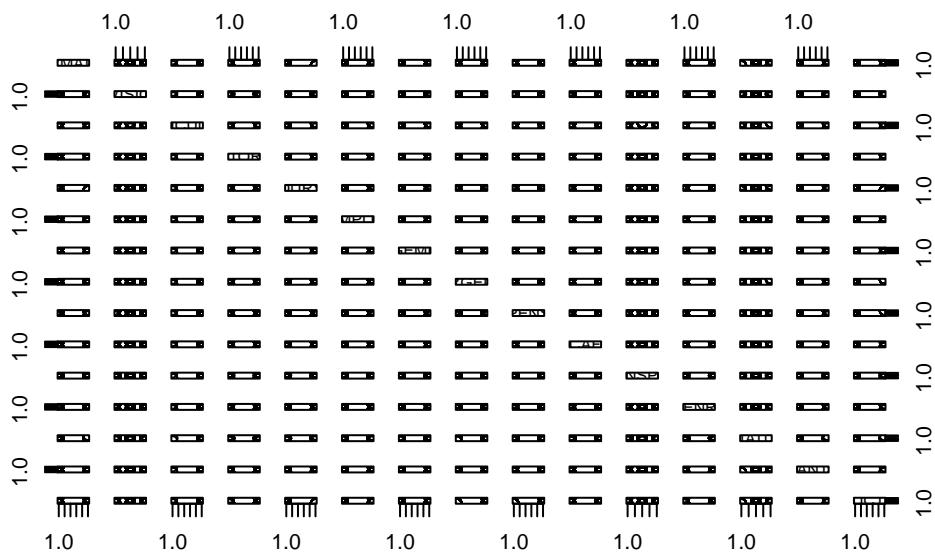
Représenter toutes les variables sur un même graphique étant illisible, nous avons décidé de construire un graphique en fonction de la nature des variables, quantitative et qualitative.

```
# Graphiques croisés des variables quantitatives  
  
plot(Budget2[, quantis])
```



```
# Graphiques croisés des variables qualitatives
```

```
plot(Budget2[, qualis])
```



B. Analyse descriptive univariée

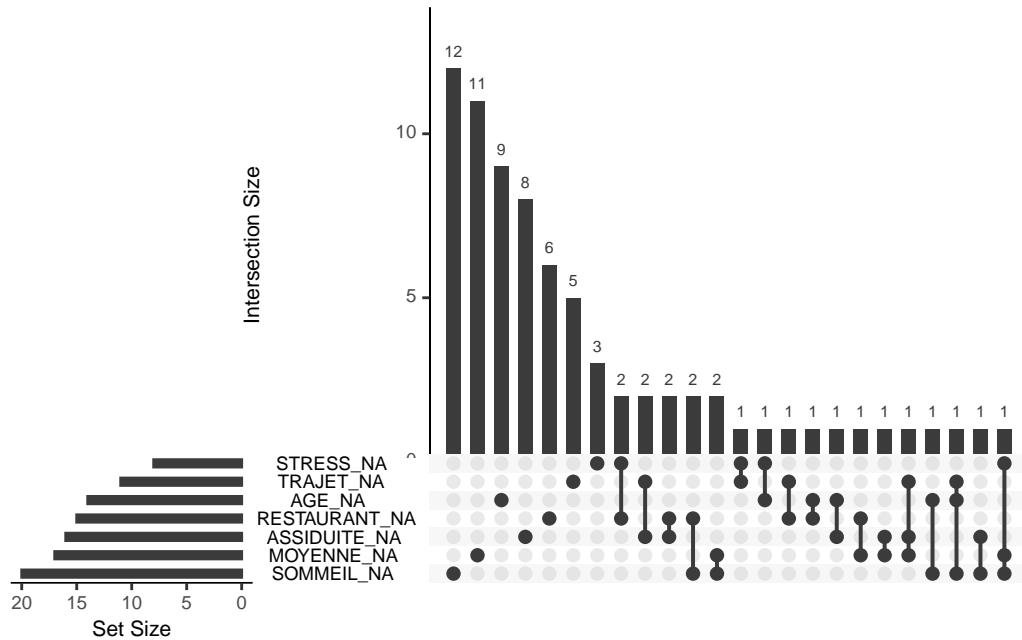
Désormais, nous allons procéder à une analyse descriptive univariée des variables explicatives et expliquée. Cette étape est cruciale pour nous permettre de mesurer la dispersion des données selon chaque variable. En effet, une analyse univariée a pour but de décrire et mesurer la répartition des données d'une seule variable. Nous débuterons par l'étude des variables quantitatives (1), suivie de l'examen des valeurs atypiques dans ces variables (2), et enfin, nous analyserons les variables qualitatives (3).

1. Variables quantitatives

```
# Statistiques  
  
summary(Budget2[, quantis])
```

ASSIDUITE	MOYENNE	RESTAURANT	AGE
Min. : 5.000	Min. : 9.60	Min. : 0.000	Min. : 18.0
1st Qu.: 8.000	1st Qu.: 12.07	1st Qu.: 1.000	1st Qu.: 20.0
Median : 9.000	Median : 13.50	Median : 2.000	Median : 21.0
Mean : 8.555	Mean : 13.40	Mean : 2.148	Mean : 21.1
3rd Qu.: 10.000	3rd Qu.: 14.38	3rd Qu.: 3.000	3rd Qu.: 22.0
Max. : 10.000	Max. : 17.53	Max. : 6.000	Max. : 25.0
NA's : 16	NA's : 17	NA's : 15	NA's : 14
STRESS	SOMMEIL	TRAJET	
Min. : 1.000	Min. : 5.000	Min. : 5.00	
1st Qu.: 1.000	1st Qu.: 7.000	1st Qu.: 20.00	
Median : 3.000	Median : 7.000	Median : 30.00	
Mean : 3.488	Mean : 7.217	Mean : 33.65	
3rd Qu.: 5.000	3rd Qu.: 8.000	3rd Qu.: 45.00	
Max. : 10.000	Max. : 9.000	Max. : 90.00	
NA's : 8	NA's : 20	NA's : 11	

```
# Graphique des données manquantes pour les variables quantitatives  
  
naniar::gg_miss_upset(Budget2[, quantis],  
                        nsets = 8,  
                        nintersects = 150)
```



Concernant les variables quantitatives, nous pouvons observer qu'il y a un total de 101 valeurs manquantes (NA).

Les statistiques et ce graphique indiquent les valeurs manquantes suivantes :

- 16 valeurs manquantes pour la variable ASSIDUITE.
- 17 valeurs manquantes pour la variable MOYENNE.
- 15 valeurs manquantes pour la variable RESTAURANT.
- 14 valeurs manquantes pour la variable AGE.
- 8 valeurs manquantes pour la variable STRESS.
- 20 valeurs manquantes pour la variable SOMMEIL.
- 11 valeurs manquantes pour la variable TRAJET.

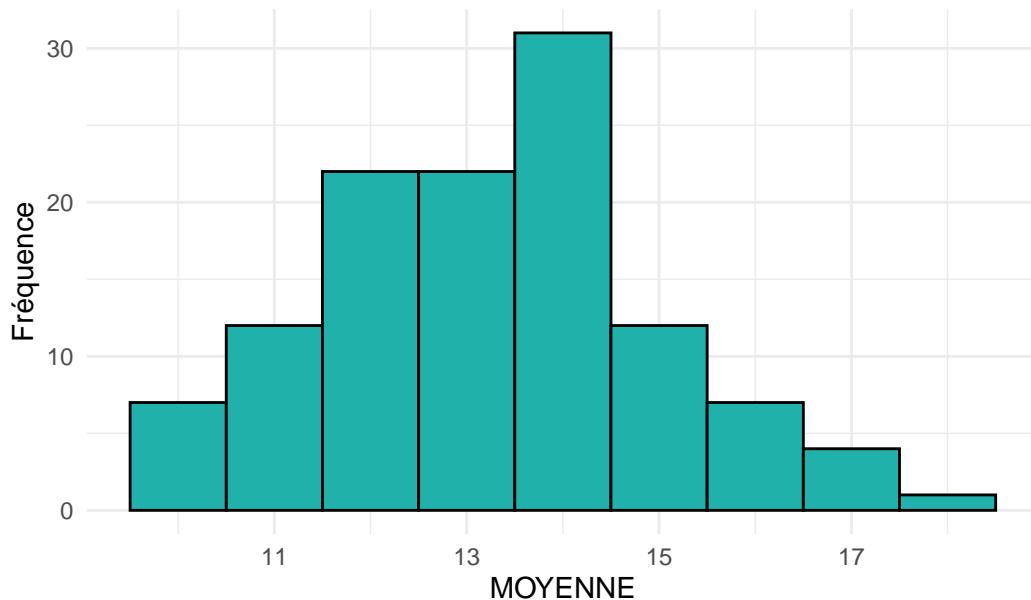
```
# Vecteur de couleurs
couleurs <- c("lightseagreen", "lightpink", "mediumseagreen", "lightsalmon", "lightcor
```

```
# Distribution des valeurs des variables quantitatives
```

```
par(mfrow = c(1, 1))
```

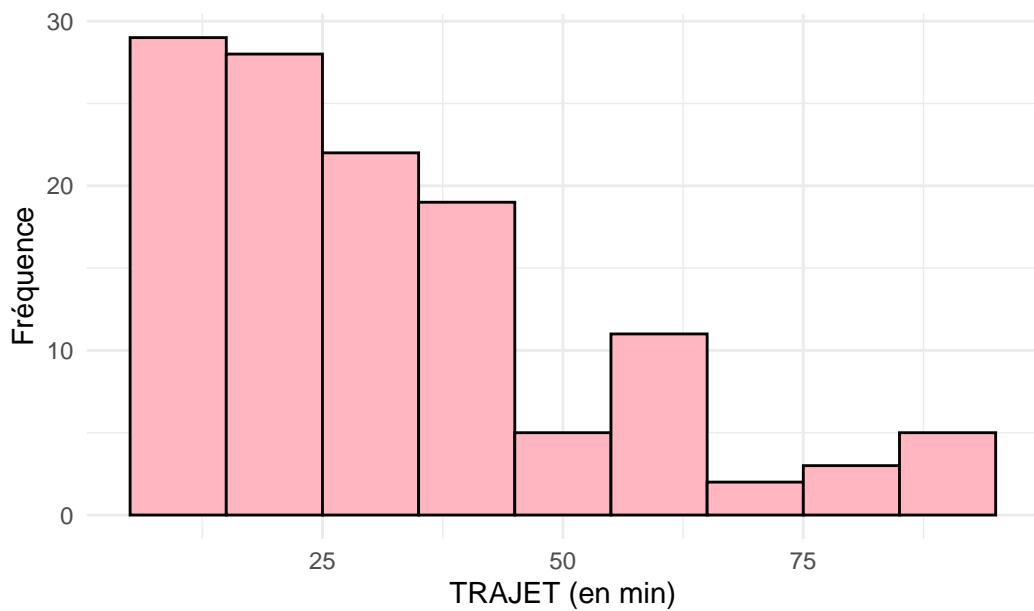
```
# Graphique pour MOYENNE
ggplot(Budget2, aes(x = MOYENNE)) +
  geom_histogram(binwidth = 1, fill = couleurs[1], color = "black") +
  labs(title = "", x = "MOYENNE", y = "Fréquence") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_bin()`).



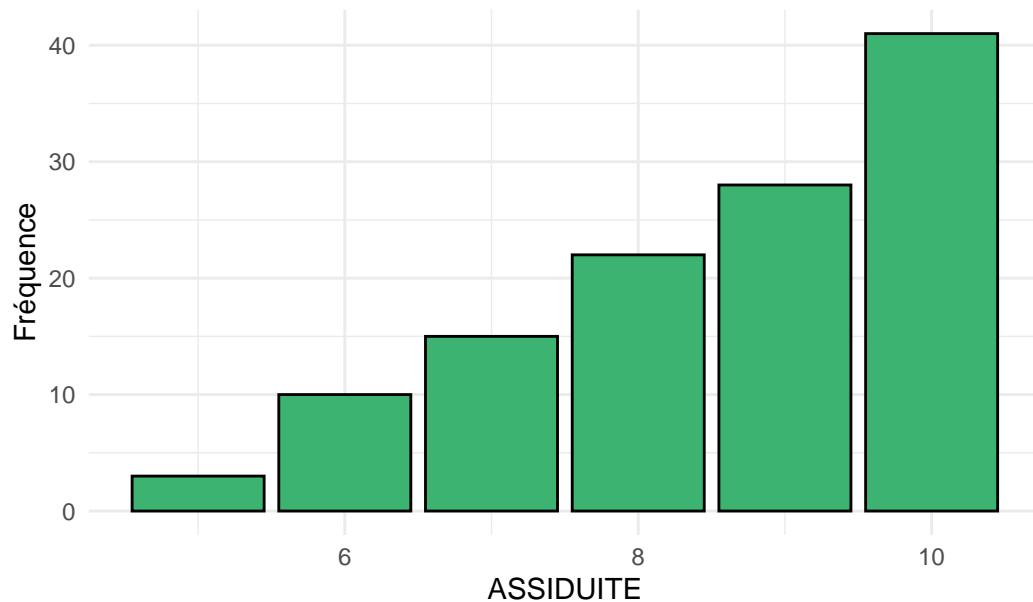
```
# Graphique pour TRAJET
ggplot(Budget2, aes(x = TRAJET)) +
  geom_histogram(binwidth = 10, fill = couleurs[2], color = "black") +
  labs(title = "", x = "TRAJET (en min)", y = "Fréquence") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_bin()`).



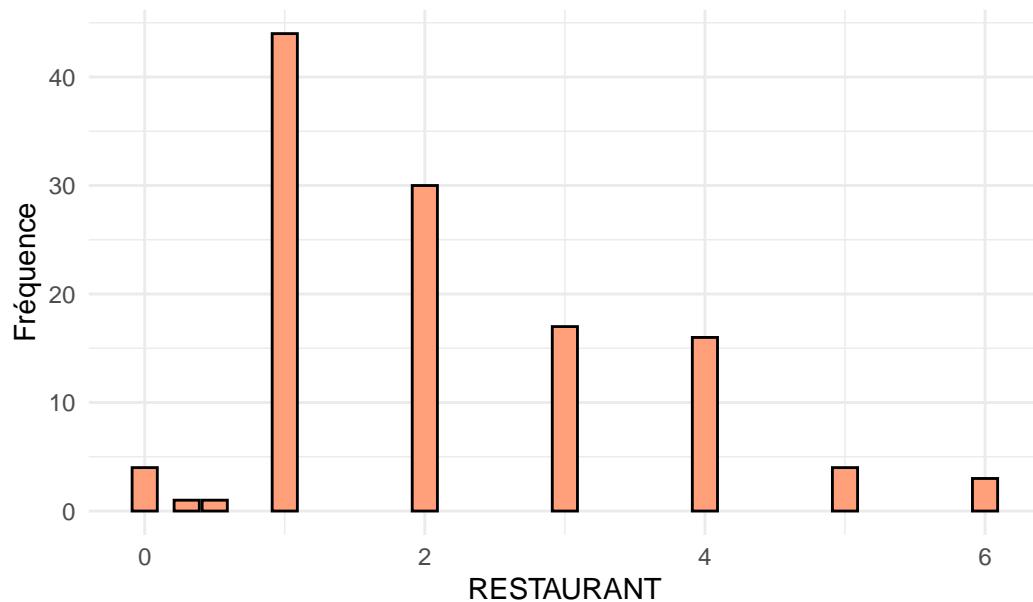
```
# Graphique pour ASSIDUITE
ggplot(Budget2, aes(x = ASSIDUITE)) +
  geom_bar(fill = couleurs[3], color = "black") +
  labs(title = "", x = "ASSIDUITE", y = "Fréquence") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_count()`).



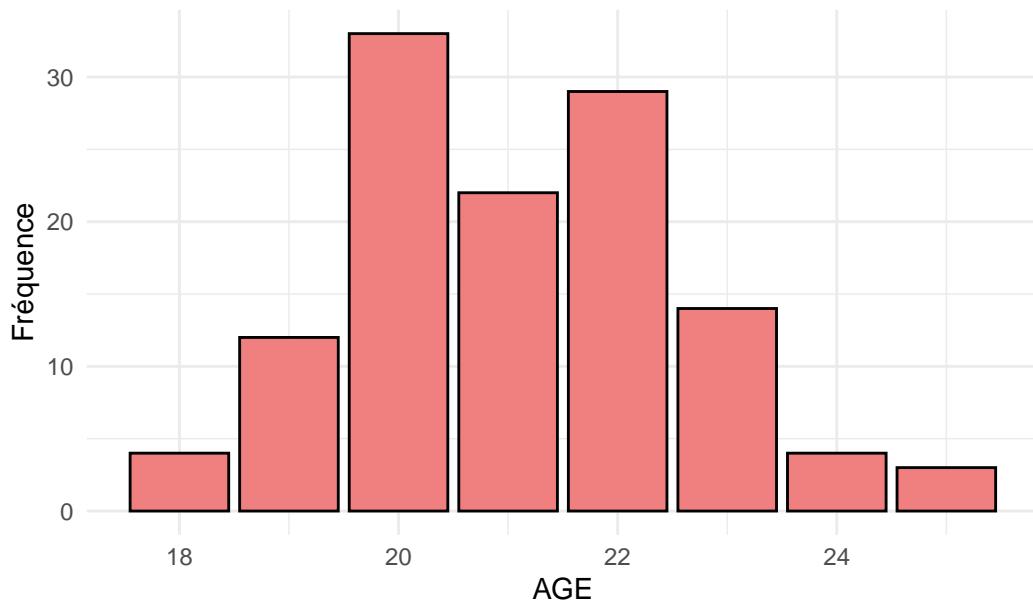
```
# Graphique pour RESTAURANT
ggplot(Budget2, aes(x = RESTAURANT)) +
  geom_bar(fill = couleurs[4], color = "black") +
  labs(title = "", x = "RESTAURANT", y = "Fréquence") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_count()`).



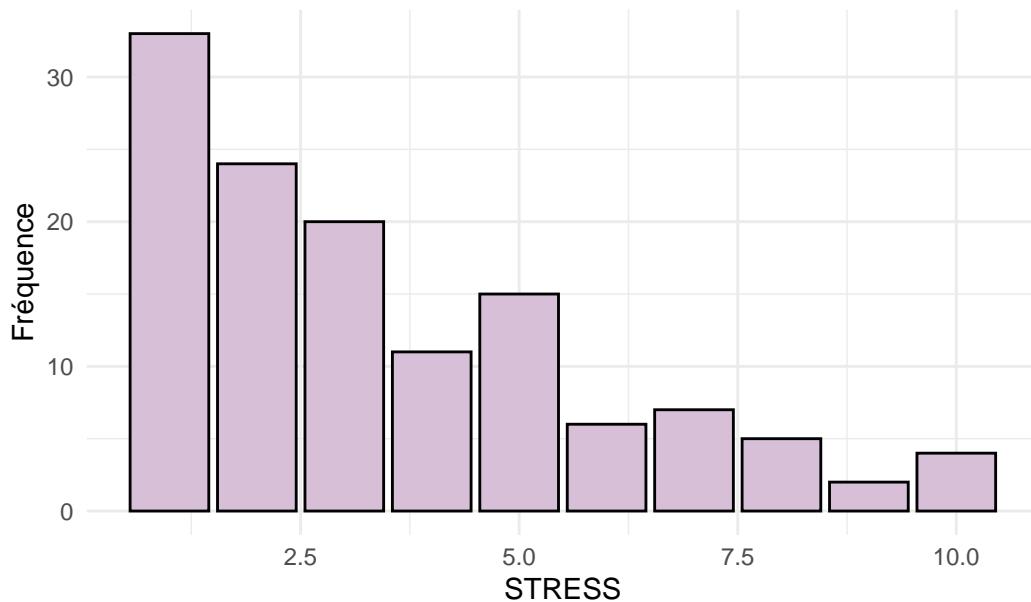
```
# Graphique pour AGE
ggplot(Budget2, aes(x = AGE)) +
  geom_bar(fill = couleurs[5], color = "black") +
  labs(title = "", x = "AGE", y = "Fréquence") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_count()`).



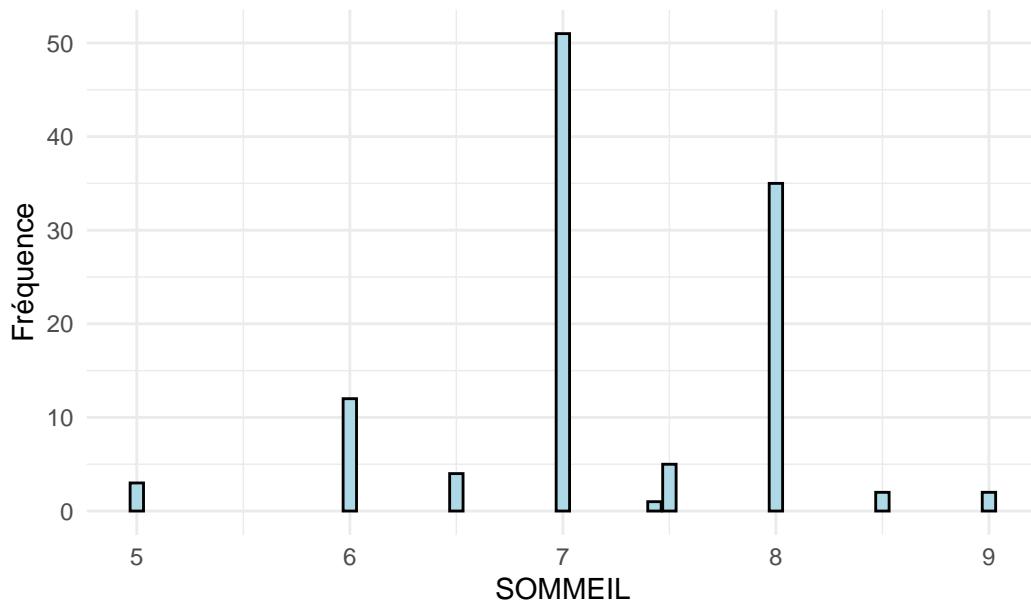
```
# Graphique pour STRESS
ggplot(Budget2, aes(x = STRESS)) +
  geom_bar(fill = couleurs[6], color = "black") +
  labs(title = "", x = "STRESS", y = "Fréquence") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_count()`).



```
# Graphique pour SOMMEIL
ggplot(Budget2, aes(x = SOMMEIL)) +
  geom_bar(fill = couleurs[7], color = "black") +
  labs(title = "", x = "SOMMEIL", y = "Fréquence") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Warning: Removed 20 rows containing non-finite outside the scale range
`stat_count()`).



```
# Boites à moustaches des 4 premières variables quantitatives

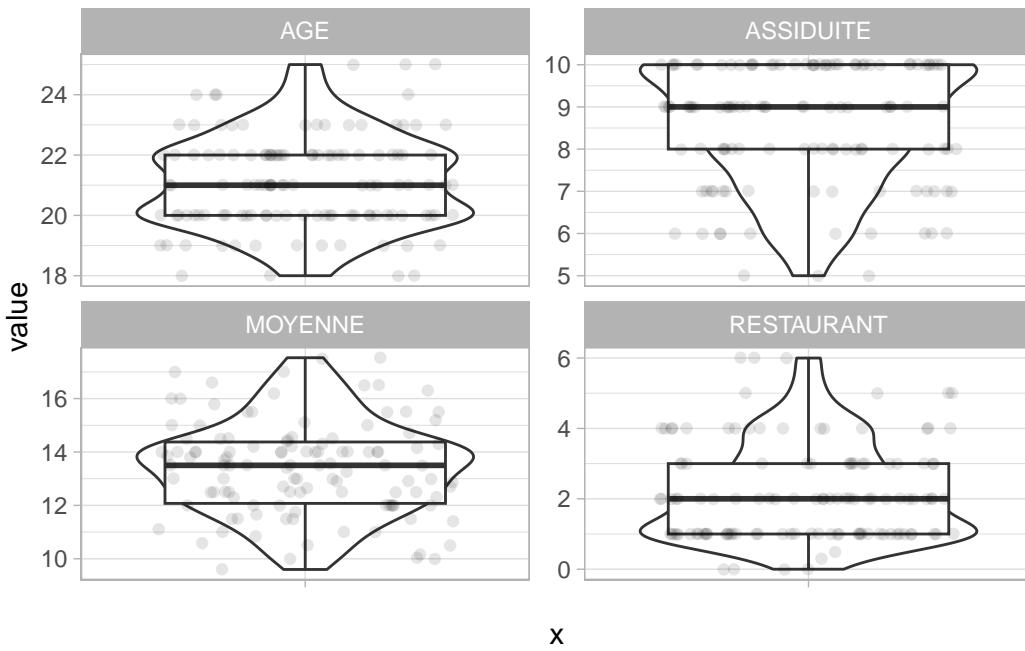
quantis1 <- quantis[1:4]

Budget2 |>
  pivot_longer(
    cols = all_of(quantis1)
  ) |>
  ggplot() +
  aes(y = value, x = "") +
  facet_wrap(~ name, scales = "free_y") +
  geom_violin() +
  geom_boxplot() +
  geom_jitter(alpha = 0.1) +
  theme_light()
```

Warning: Removed 62 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 62 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 62 rows containing missing values or values outside the scale range (`geom_point()`).



```
# Boites à moustaches sur les 4 premières variables quantitatives

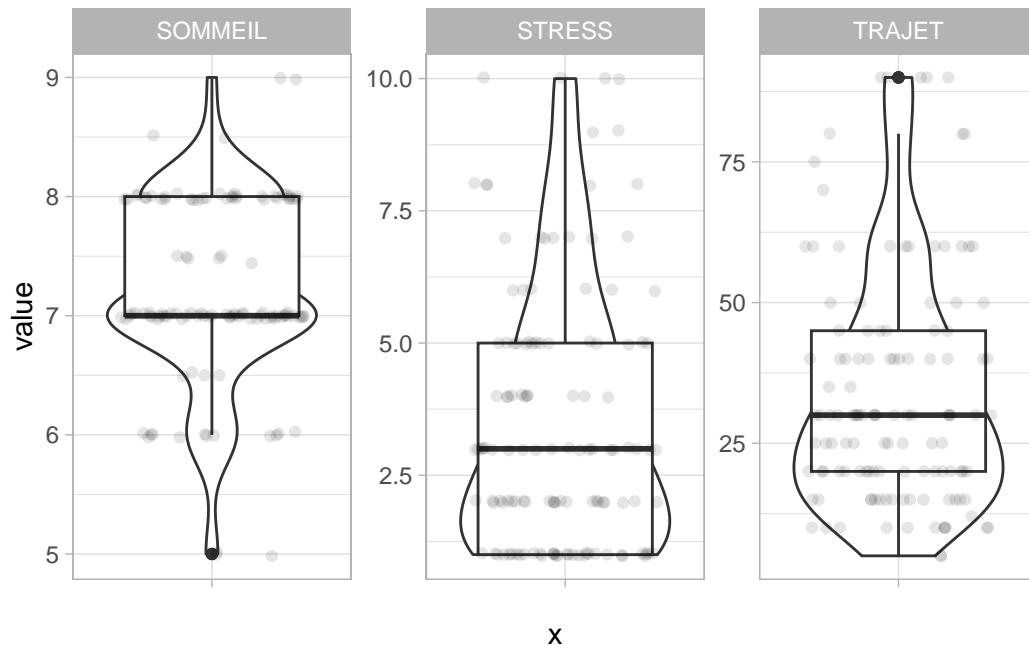
quantis2 <- quantis[5:7]

Budget2 |>
  pivot_longer(
    cols = all_of(quantis2)
  ) |>
  ggplot() +
  aes(y = value, x = "") +
  facet_wrap(~ name, scales = "free_y") +
  geom_violin() +
  geom_boxplot() +
  geom_jitter(alpha = 0.1) +
  theme_light()
```

Warning: Removed 39 rows containing non-finite outside the scale range (`stat_ydensity()`).

```
Warning: Removed 39 rows containing non-finite outside the scale range  
(`stat_boxplot()`).
```

```
Warning: Removed 39 rows containing missing values or values outside the scale range  
(`geom_point()`).
```



Grâce à ces graphiques, nous pouvons visualiser les éventuelles valeurs atypiques (ou outliers) dans nos variables quantitatives. Cependant, d'après les distributions observées, il n'y a pas de points qui semblent clairement atypiques.

2. Variables qualitatives

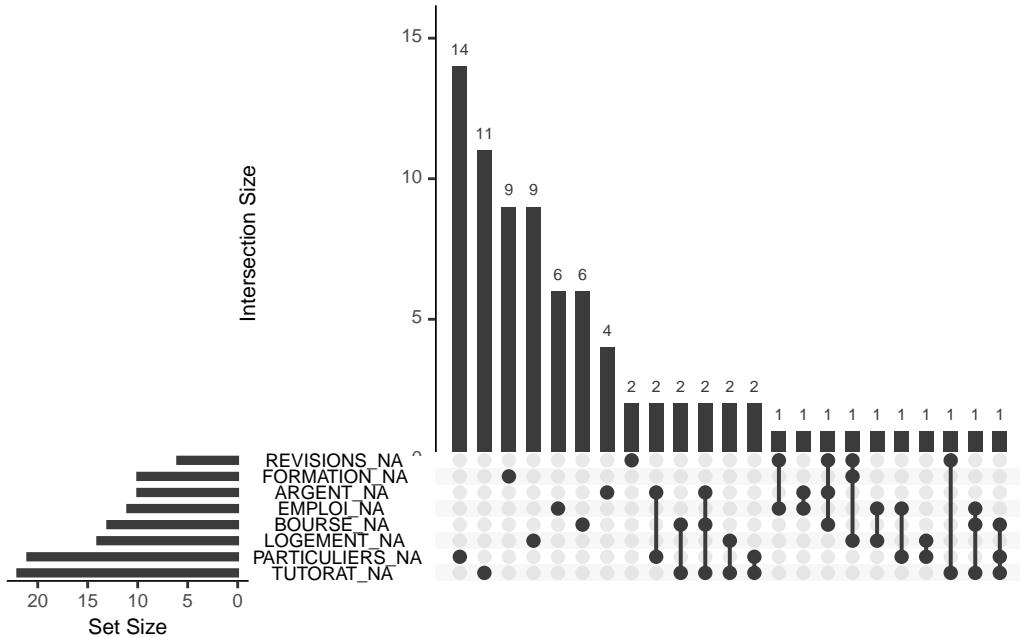
```
# Statistiques  
  
summary(Budget2[, qualis])
```

	FORMATION	REVISIONS	PARTICULIERS	TUTORAT	BOURSE
Economie et gestion:	105	2h et plus:30	Oui : 5	Oui :31	Non :86
Autres	: 20	0 - 1h :52	Non :109	Non :82	Oui :36

NA's	:	10	1 - 2h	: 47	NA's:	21	NA's: 22	NA's: 13
			NA's		NA's			
EMPLOI	LOGEMENT	ARGENT		DEPENSES	CAF		TRANSPORT	
Non : 86	Non : 59	Oui : 55	100 € et plus: 49	Non : 58	30 € et plus: 42			
Oui : 38	Oui : 62	Non : 70	0 - 100 €	Oui : 68	0 - 15 €	: 27		
NA's: 11	NA's: 14	NA's: 10	NA's	NA's: 12	NA's: 9	15 - 30 €	: 55	
						NA's		: 11
GENRE		STATUT		SANTE		STRUCTURE		
Homme: 56		Classe aisée	: 13	Non : 101	Université: 111			
Femme: 65		Classe moyenne	: 95	Oui : 21	Autres : 9			
NA's : 14		Classe populaire: 12		NA's: 13	NA's	: 15		
		NA's	: 15					

```
# Graphique des données manquantes pour la première série de 8 variables qualitatives
```

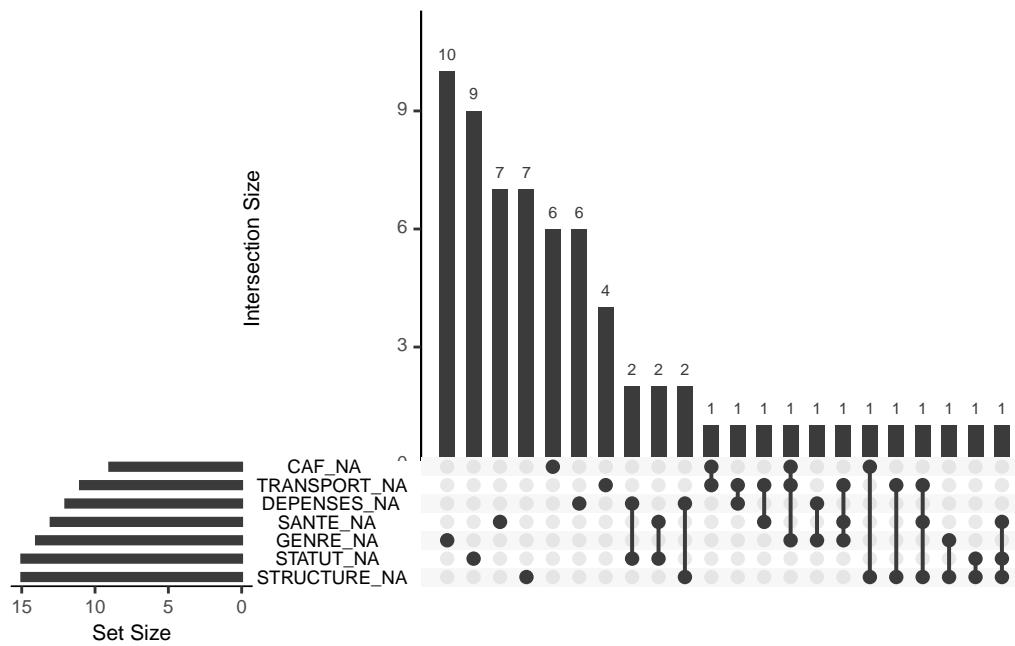
```
naniar::gg_miss_upset(Budget2[, qualis[1:8]],  
nsets = 8,  
nintersects = 150)
```



```
# Graphique des données manquantes pour la deuxième série de 7 variables qualitatives
```

```
naniar::gg_miss_upset(Budget2[, qualis[9:15]],
```

```
nsets = 8,
nintersects = 150)
```



Concernant les variables qualitatives, nous constatons qu'il y a un total de 196 valeurs manquantes.

Les statistiques et les graphiques révèlent les valeurs manquantes suivantes :

- 10 valeurs manquantes pour la variable FORMATION.
- 6 valeurs manquantes pour la variable REVISIONS.
- 21 valeurs manquantes pour la variable PARTICULIERS.
- 22 valeurs manquantes pour la variable TUTORAT.
- 13 valeurs manquantes pour la variable BOURSE.
- 11 valeurs manquantes pour la variable EMPLOI.
- 14 valeurs manquantes pour la variable LOGEMENT.
- 10 valeurs manquantes pour la variable ARGENT.
- 12 valeurs manquantes pour la variable DEPENSES.
- 9 valeurs manquantes pour la variable CAF.

- 11 valeurs manquantes pour la variable TRANSPORT.
- 14 valeurs manquantes pour la variable GENRE.
- 15 valeurs manquantes pour la variable STATUT.
- 13 valeurs manquantes pour la variable SANTE.
- 15 valeurs manquantes pour la variable STRUCTURE.

```
# Visualisation des lignes ayant des valeurs manquantes

indice_na <- which(is.na(Budget2), arr.ind = TRUE)
indice <- indice_na[,1]
Budget2[indice,]

# A tibble: 297 x 22
  FORMATION ASSIDUITE REVISIONS PARTICULIERS TUTORAT MOYENNE BOURSE EMPLOI
  <fct>      <dbl> <fct>      <fct>      <fct>      <dbl> <fct>      <fct>
1 <NA>          9 0 - 1h    Non        Non       12   Oui      Non
2 <NA>          9 0 - 1h    Non        Non       12   Non      Non
3 <NA>          NA 2h et plus Non        Oui      12.7  Non      Non
4 <NA>          10 1 - 2h   Non        Non      13.3  Non      Oui
5 <NA>          6 2h et plus Non        Non      12.7  Oui      Oui
6 <NA>          NA 1 - 2h   Non        Oui      12.5  Oui      Non
7 <NA>          9 0 - 1h    Non        Non      15.5  Non      Non
8 <NA>          9 <NA>     Non        Non      14.5  Non      Non
9 <NA>          10 0 - 1h   Non        Non      14.4  Non      Non
10 <NA>         10 1 - 2h   Non        Non      11.4  Non      Non
# i 287 more rows
# i 14 more variables: LOGEMENT <fct>, ARGENT <fct>, RESTAURANT <dbl>,
# DEPENSES <fct>, CAF <fct>, TRANSPORT <fct>, GENRE <fct>, AGE <dbl>,
# STRESS <dbl>, STATUT <fct>, SOMMEIL <dbl>, SANTE <fct>, STRUCTURE <fct>,
# TRAJET <dbl>
```

A l'aide de la fonction count, nous pouvons observer les statistiques descriptives des variables qualitatives.

```
for (i in qualis) {
  cat("Variable", i)
  print(count(Budget2, Budget2[[i]]))
}
```

```

Variable FORMATION# A tibble: 3 x 2
  `Budget2[[i]]`      n
  <fct>            <int>
1 Economie et gestion    105
2 Autres                  20
3 <NA>                   10

Variable REVISIONS# A tibble: 4 x 2
  `Budget2[[i]]`      n
  <fct>            <int>
1 2h et plus        30
2 0 - 1h           52
3 1 - 2h           47
4 <NA>              6

Variable PARTICULIERS# A tibble: 3 x 2
  `Budget2[[i]]`      n
  <fct>            <int>
1 Oui                 5
2 Non                109
3 <NA>               21

Variable TUTORAT# A tibble: 3 x 2
  `Budget2[[i]]`      n
  <fct>            <int>
1 Oui                 31
2 Non                82
3 <NA>               22

Variable BOURSE# A tibble: 3 x 2
  `Budget2[[i]]`      n
  <fct>            <int>
1 Non                86
2 Oui                 36
3 <NA>               13

Variable EMPLOI# A tibble: 3 x 2
  `Budget2[[i]]`      n
  <fct>            <int>
1 Non                86
2 Oui                 38
3 <NA>               11

Variable LOGEMENT# A tibble: 3 x 2
  `Budget2[[i]]`      n
  <fct>            <int>
1 Non                 59

```

```

2 Oui           62
3 <NA>          14
Variable ARGENT# A tibble: 3 x 2
  `Budget2[[i]]`    n
  <fct>            <int>
1 Oui             55
2 Non             70
3 <NA>            10
Variable DEPENSES# A tibble: 3 x 2
  `Budget2[[i]]`    n
  <fct>            <int>
1 100 € et plus   49
2 0 - 100 €        74
3 <NA>             12
Variable CAF# A tibble: 3 x 2
  `Budget2[[i]]`    n
  <fct>            <int>
1 Non              58
2 Oui              68
3 <NA>              9
Variable TRANSPORT# A tibble: 4 x 2
  `Budget2[[i]]`    n
  <fct>            <int>
1 30 € et plus    42
2 0 - 15 €         27
3 15 - 30 €        55
4 <NA>             11
Variable GENRE# A tibble: 3 x 2
  `Budget2[[i]]`    n
  <fct>            <int>
1 Homme            56
2 Femme             65
3 <NA>              14
Variable STATUT# A tibble: 4 x 2
  `Budget2[[i]]`    n
  <fct>            <int>
1 Classe aisée     13
2 Classe moyenne   95
3 Classe populaire 12
4 <NA>              15
Variable SANTE# A tibble: 3 x 2

```

```

`Budget2[[i]]`      n
<fct>            <int>
1 Non              101
2 Oui              21
3 <NA>             13
Variable STRUCTURE# A tibble: 3 x 2
`Budget2[[i]]`      n
<fct>            <int>
1 Université       111
2 Autres            9
3 <NA>              15

```

Nous remarquons que certaines variables sont dominées par des modalités. Ainsi, la variable FORMATION est dominée par la modalité “Economie et gestion” avec 105 réponses sur 135, tandis que la modalité “Autres” représente seulement 20 réponses. En ce qui concerne la variable PARTICULIERS, la modalité “Non” est largement majoritaire avec 109 réponses, contre seulement 5 réponses pour la modalité “Oui”. La variable STATUT est dominée par la modalité “Classe moyenne” avec 95 réponses, suivie de “Classe aisée” (13 réponses) et “Classe populaire” (12 réponses). Enfin, la variable SANTE est dominée par la modalité “Non” avec 101 réponses, tandis que “Oui” enregistre 21 réponses. Pour la variable STRUCTURE, “Université” prédomine avec 111 réponses, tandis que “Autres” ne compte que 9 réponses.

Certaines catégories sont donc beaucoup plus fréquentes que les autres. Cela peut créer un déséquilibre dans les données, introduire un biais dans l’analyse et limiter la représentativité des données car les catégories sous-représentées ne sont pas suffisamment prises en compte.

C. Analyse descriptive bivariée

Nous abordons maintenant l’analyse bivariée, qui consiste à explorer les variations d’une variable en fonction d’une autre afin de comprendre leur relation. Nous réaliserons ainsi trois analyses en fonction de la nature de la variable : une entre les variables quantitatives (1), une deuxième entre les variables qualitatives (2) et une dernière entre les variables quantitatives et qualitatives (3).

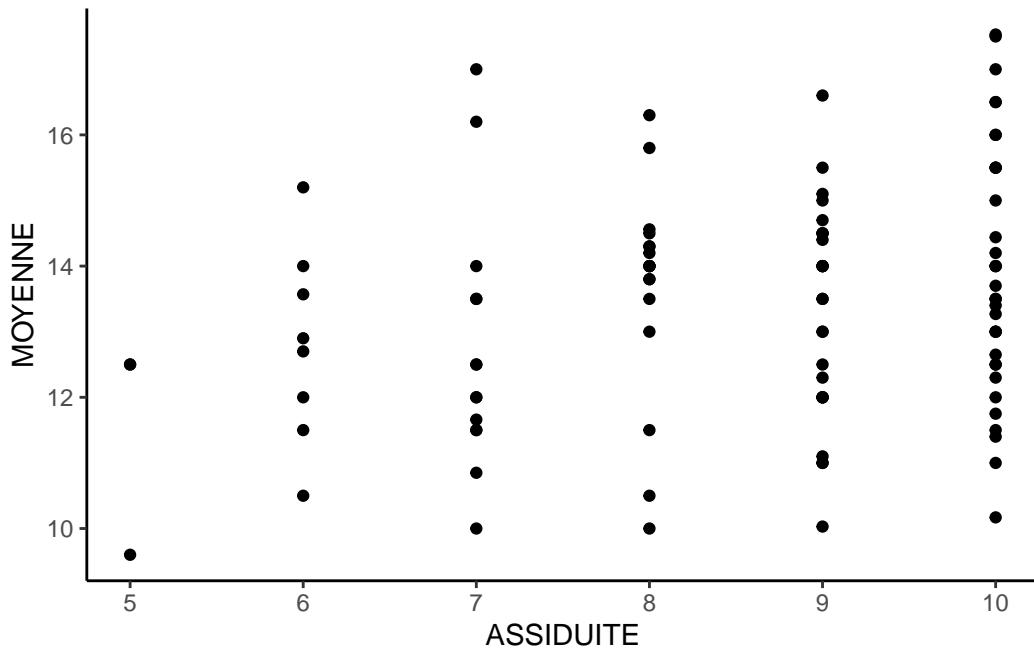
1. Variables quantitatives - quantitatives

```
# Nuage de points pour chaque paire de variables quantitatives

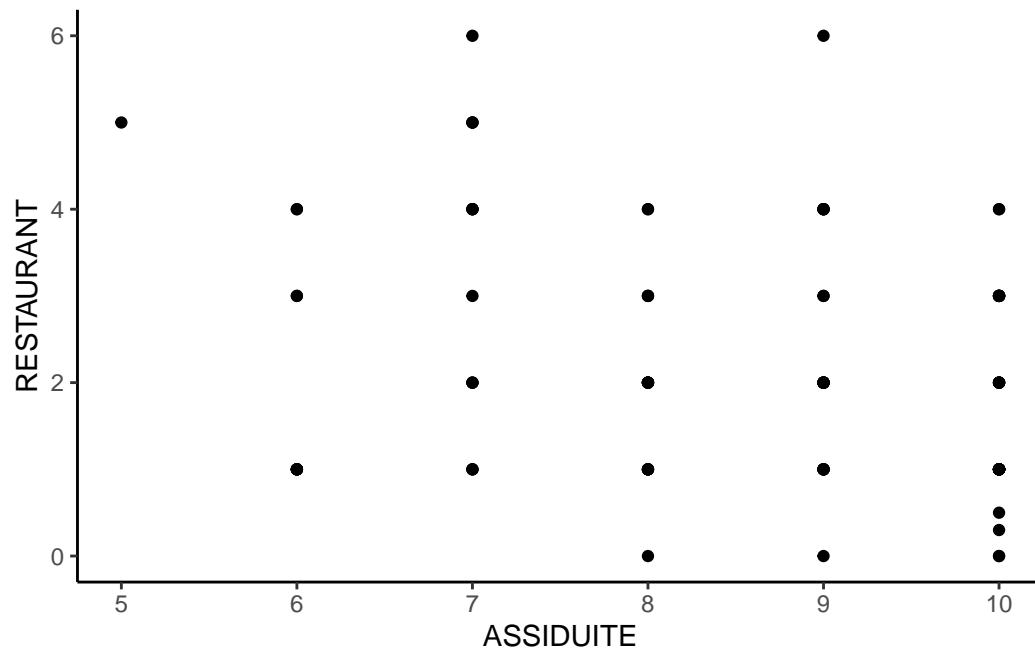
for (i in quantis) {
  for (j in quantis) {
    if (i != j) {
      p <- Budget2 |>
        ggplot() +
        aes(x = .data[[i]], y = .data[[j]]) +
        geom_point() +
        theme_classic()

      print(p)
    }
  }
}
```

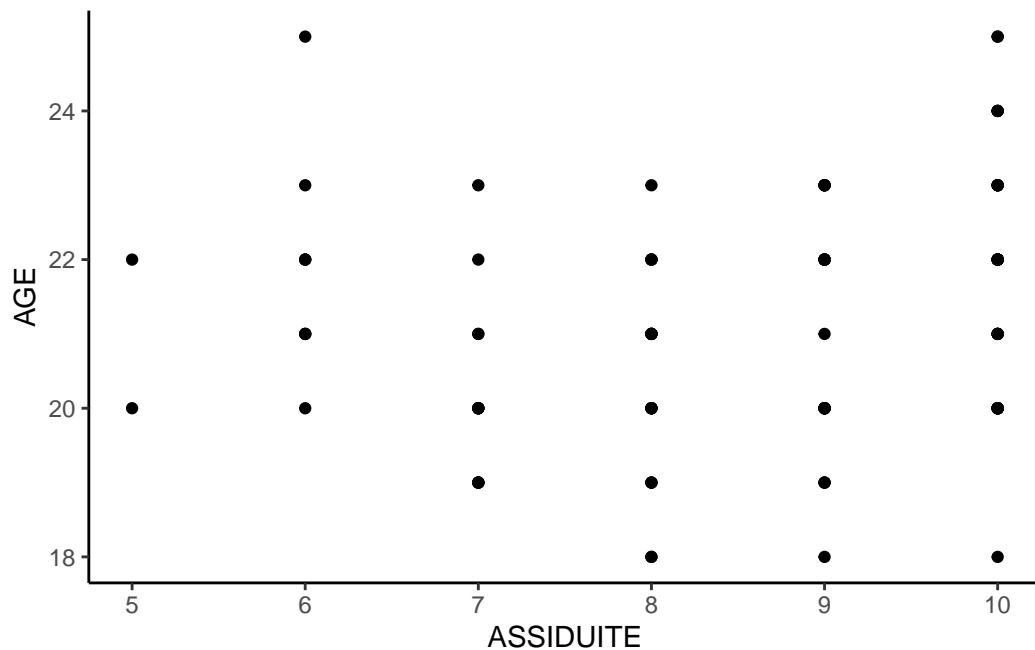
Warning: Removed 31 rows containing missing values or values outside the scale range
(`geom_point()`).



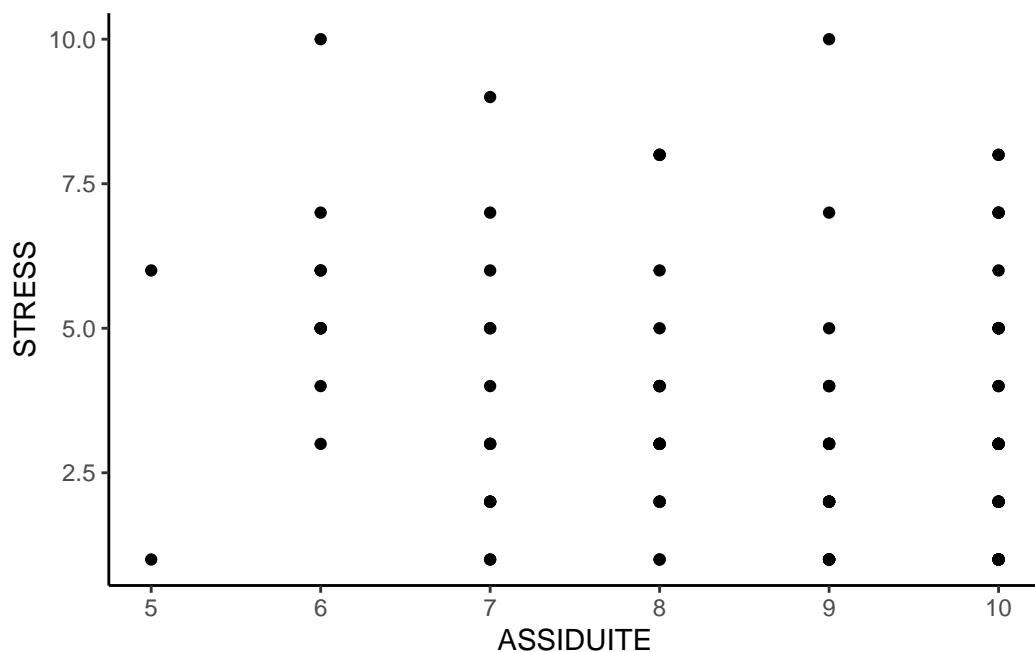
Warning: Removed 29 rows containing missing values or values outside the scale range (`geom_point()`).



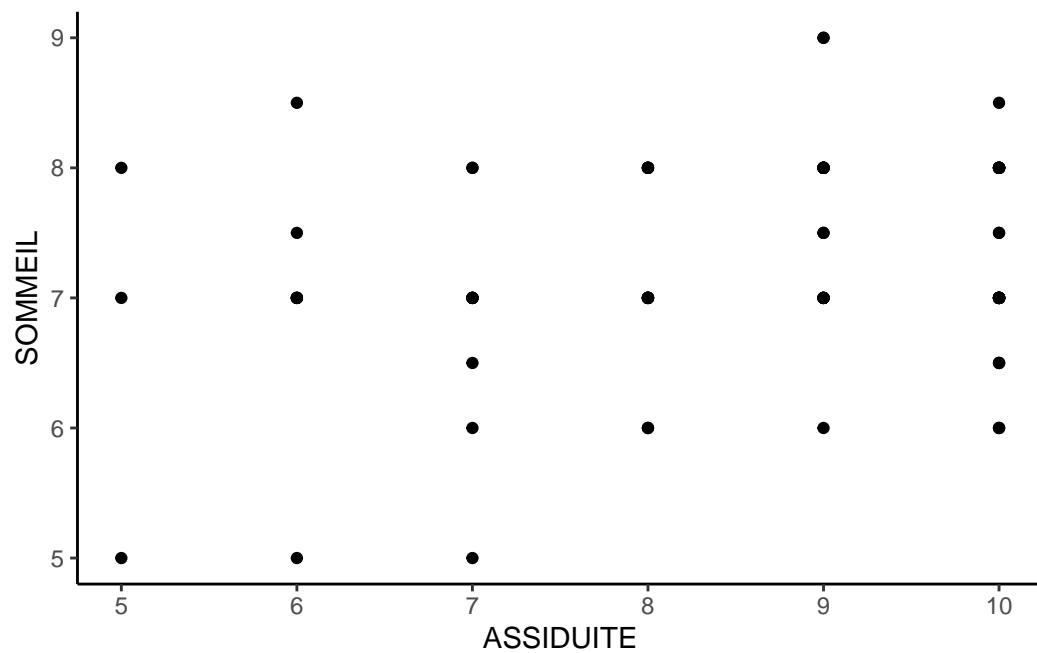
Warning: Removed 29 rows containing missing values or values outside the scale range (`geom_point()`).



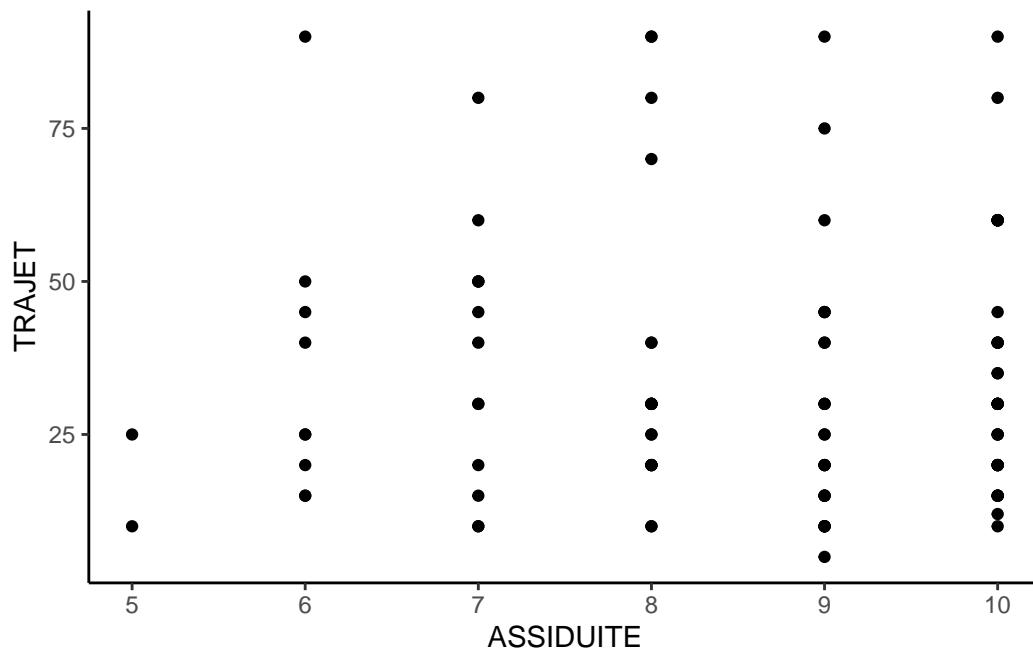
Warning: Removed 24 rows containing missing values or values outside the scale range (`geom_point()`).



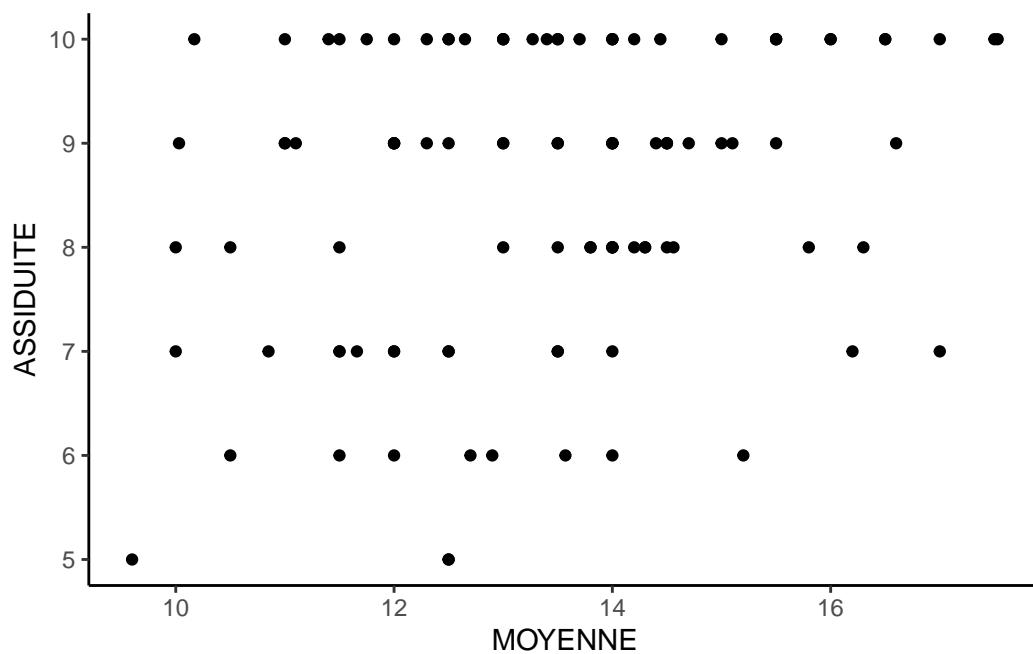
Warning: Removed 35 rows containing missing values or values outside the scale range (`geom_point()`).



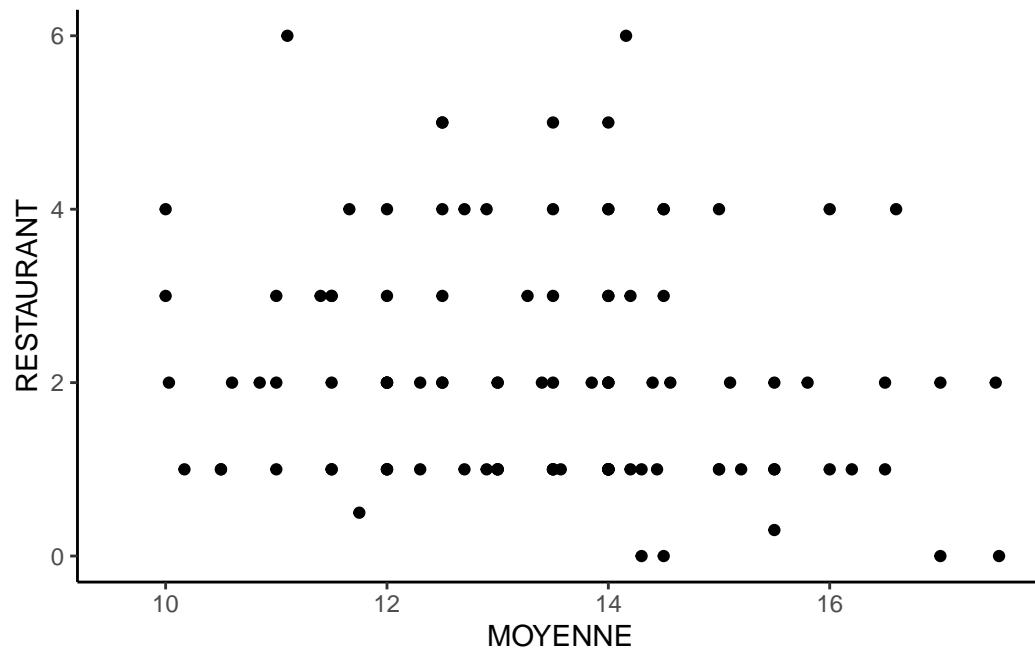
Warning: Removed 24 rows containing missing values or values outside the scale range (`geom_point()`).



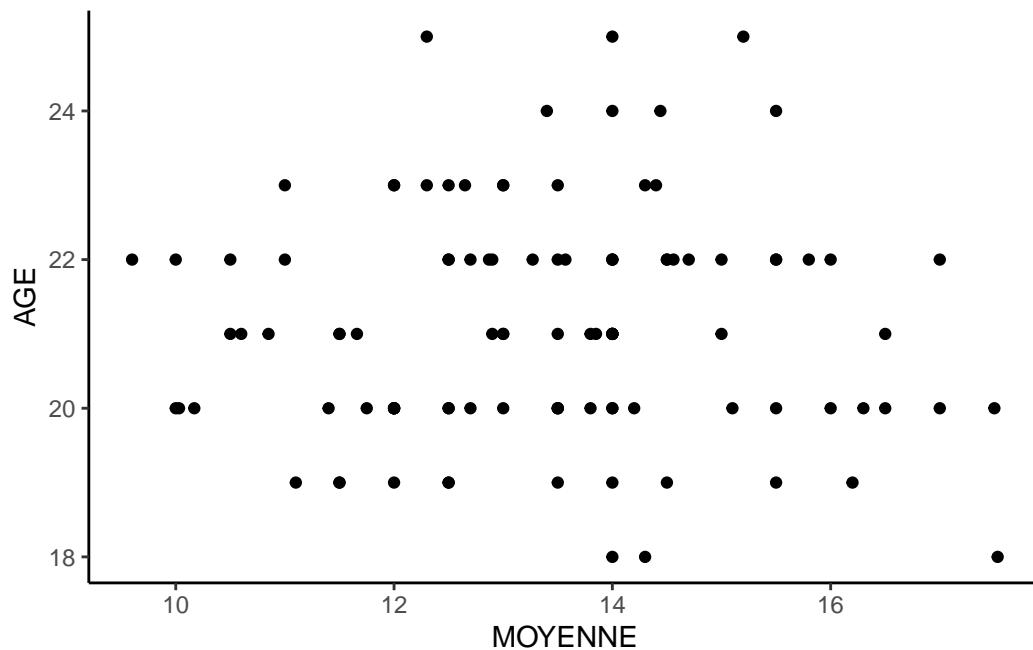
Warning: Removed 31 rows containing missing values or values outside the scale range (`geom_point()`).



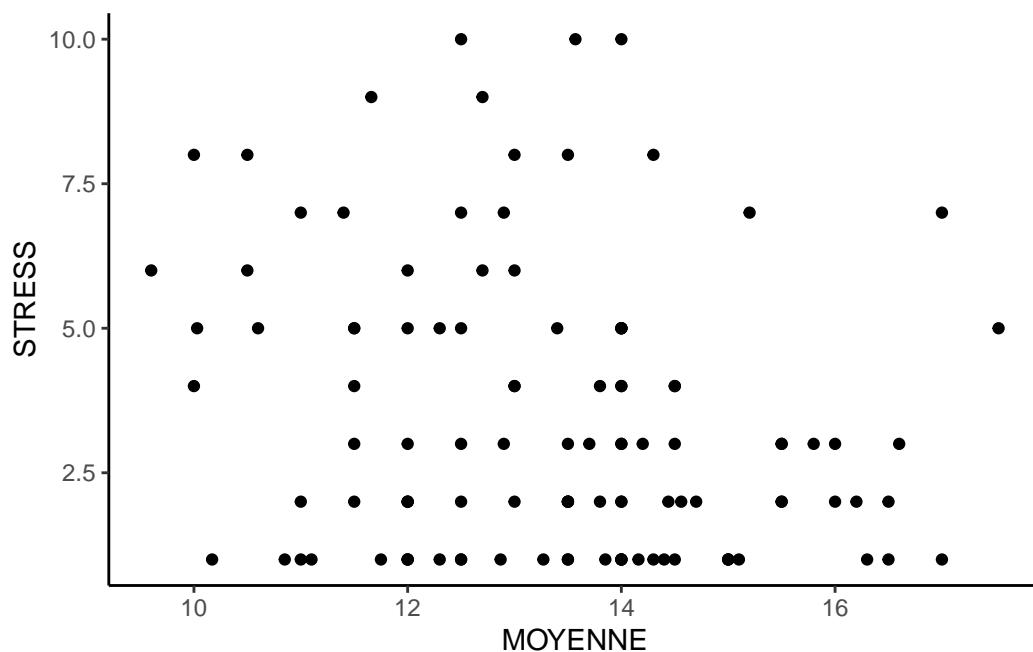
Warning: Removed 31 rows containing missing values or values outside the scale range
(`geom_point()`).



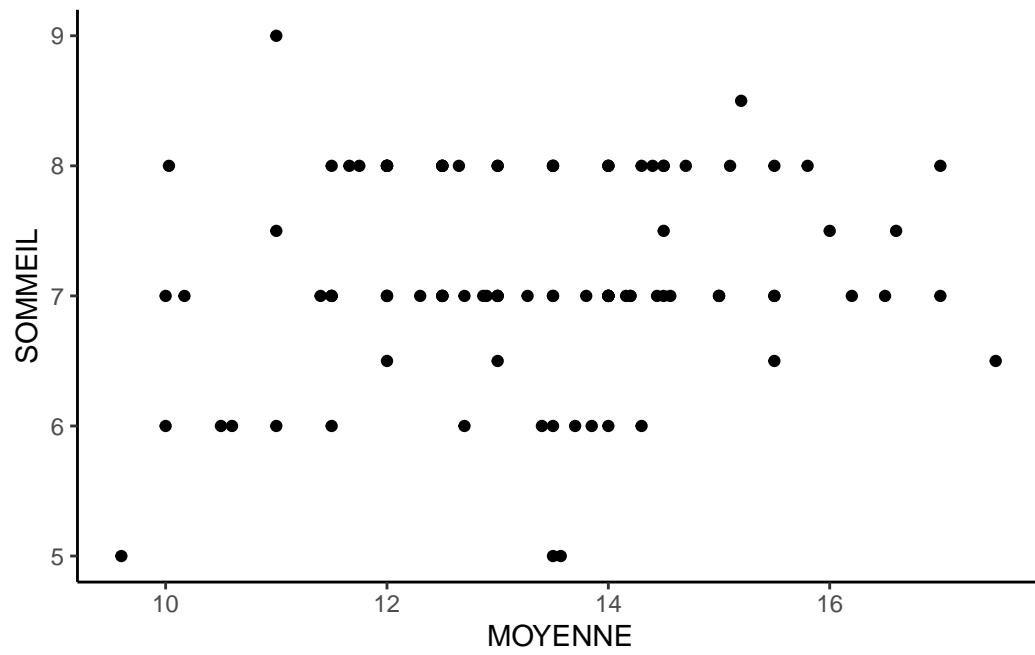
Warning: Removed 31 rows containing missing values or values outside the scale range
(`geom_point()`).



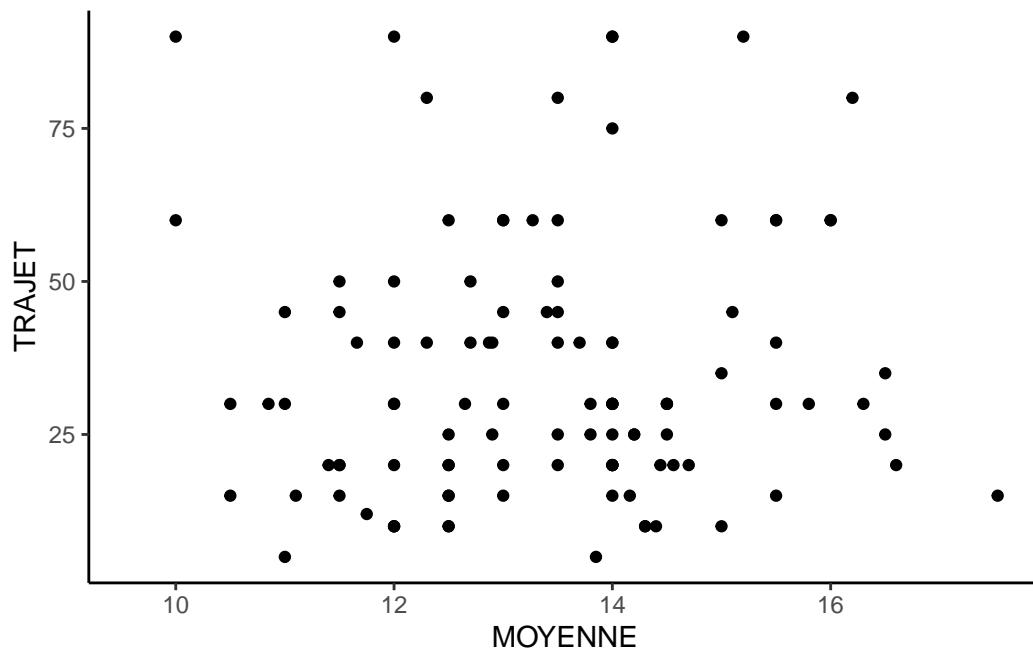
Warning: Removed 24 rows containing missing values or values outside the scale range (`geom_point()`).



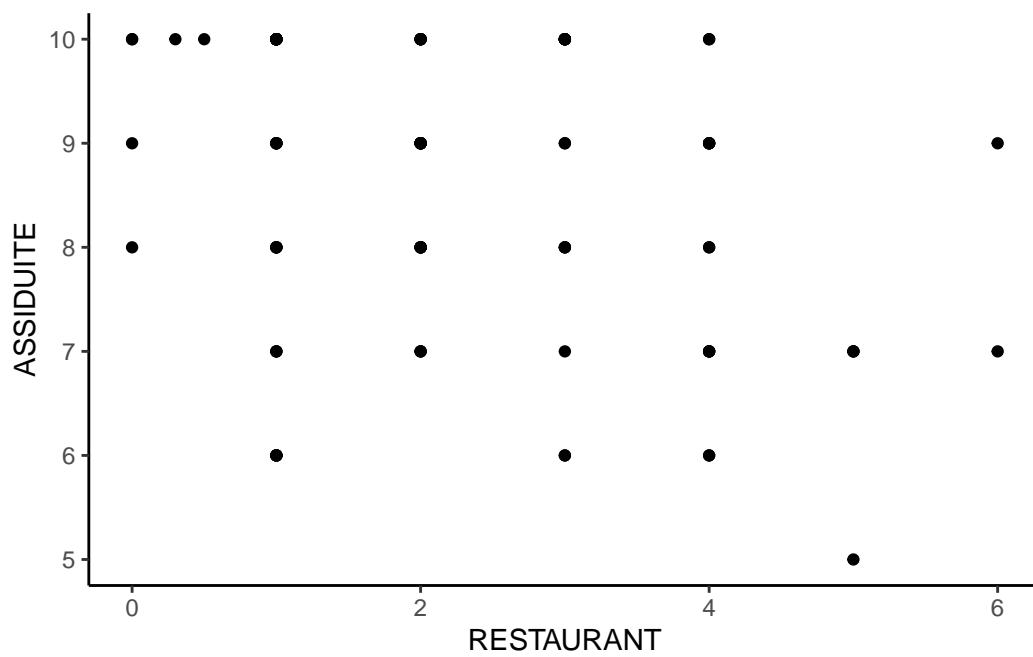
Warning: Removed 34 rows containing missing values or values outside the scale range (`geom_point()`).



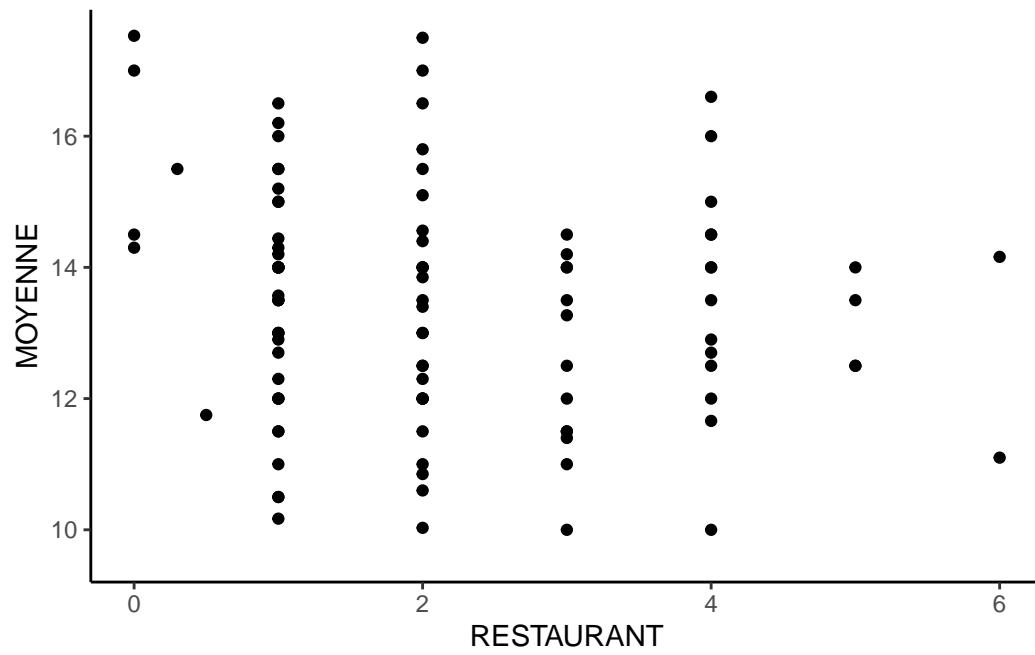
Warning: Removed 27 rows containing missing values or values outside the scale range (`geom_point()`).



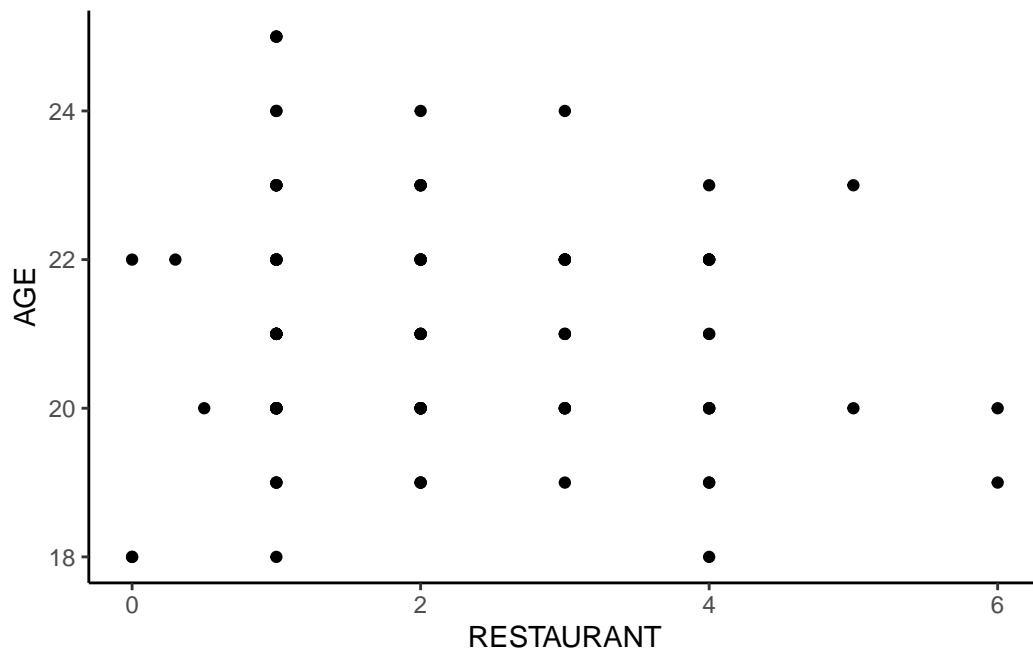
Warning: Removed 29 rows containing missing values or values outside the scale range (`geom_point()`).



Warning: Removed 31 rows containing missing values or values outside the scale range
(`geom_point()`).



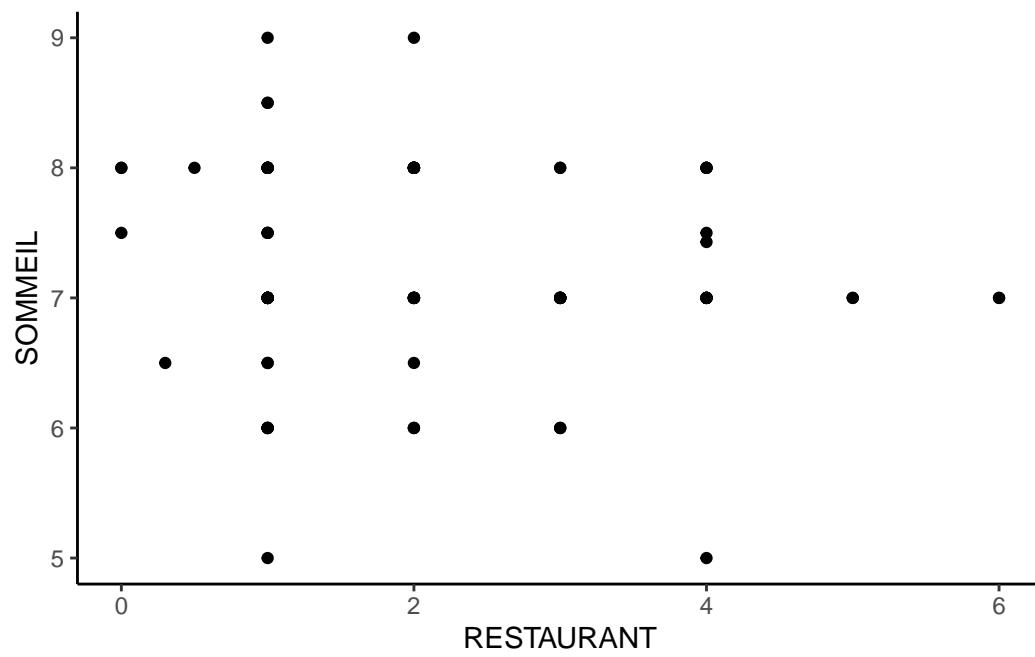
Warning: Removed 28 rows containing missing values or values outside the scale range
(`geom_point()`).



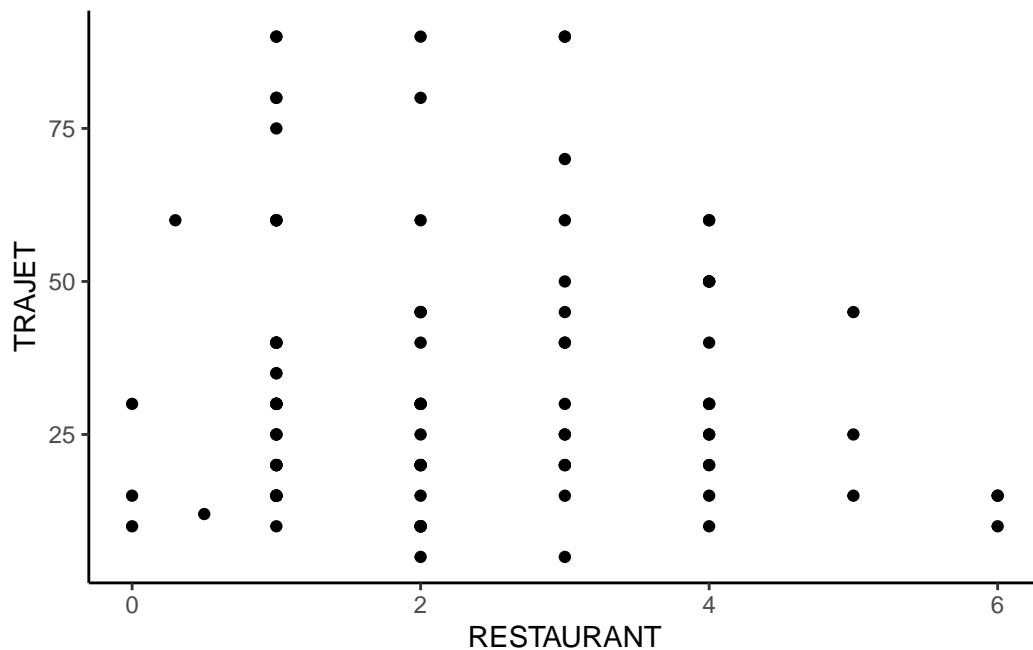
Warning: Removed 21 rows containing missing values or values outside the scale range (`geom_point()`).



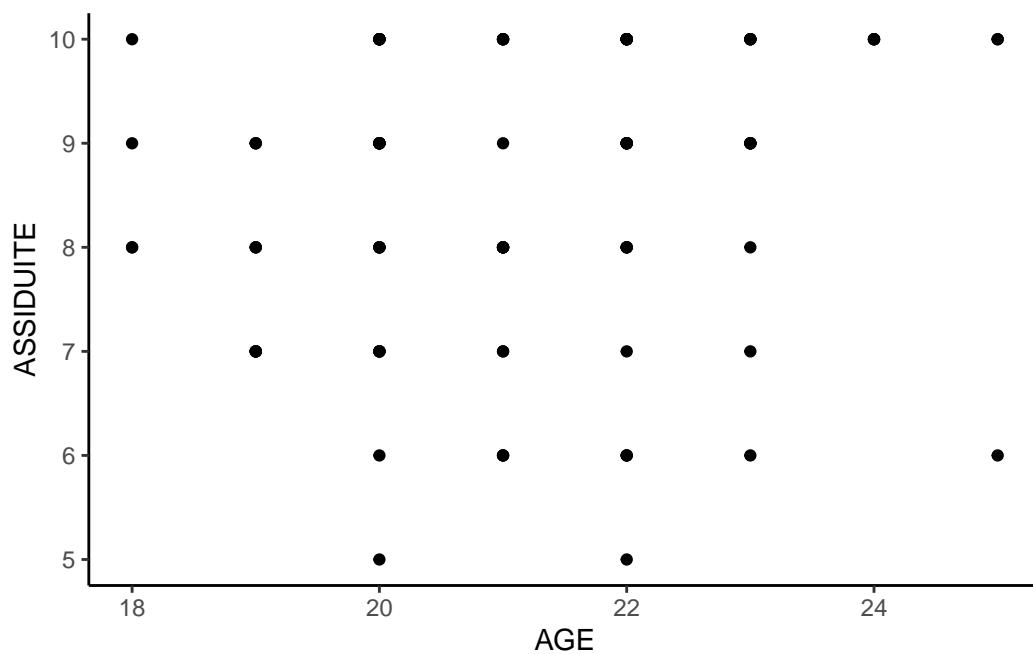
Warning: Removed 33 rows containing missing values or values outside the scale range (`geom_point()`).



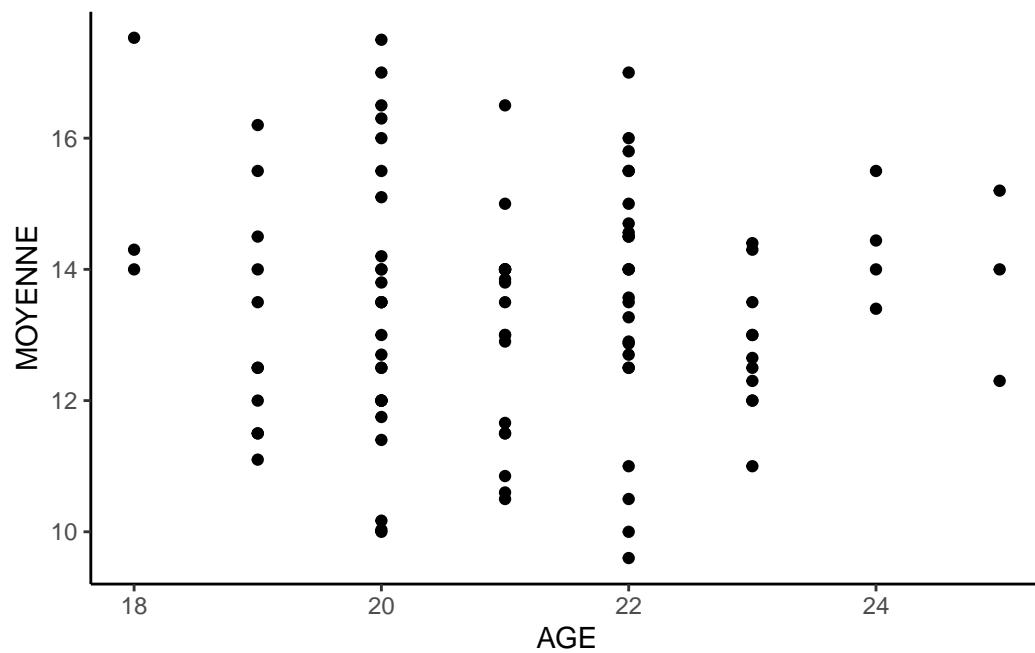
Warning: Removed 25 rows containing missing values or values outside the scale range (`geom_point()`).



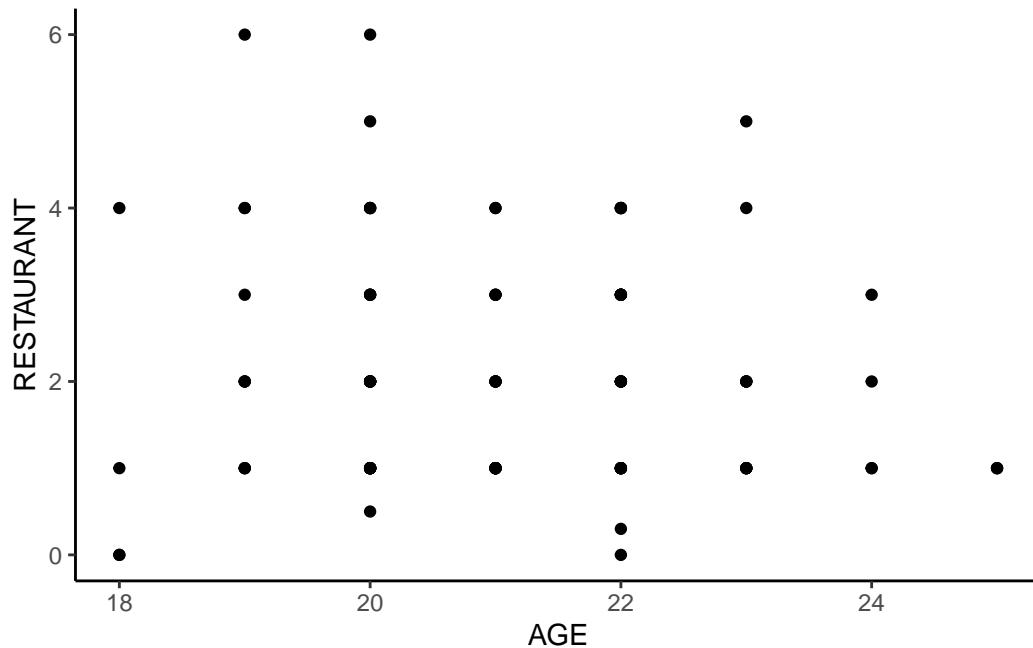
Warning: Removed 29 rows containing missing values or values outside the scale range (`geom_point()`).



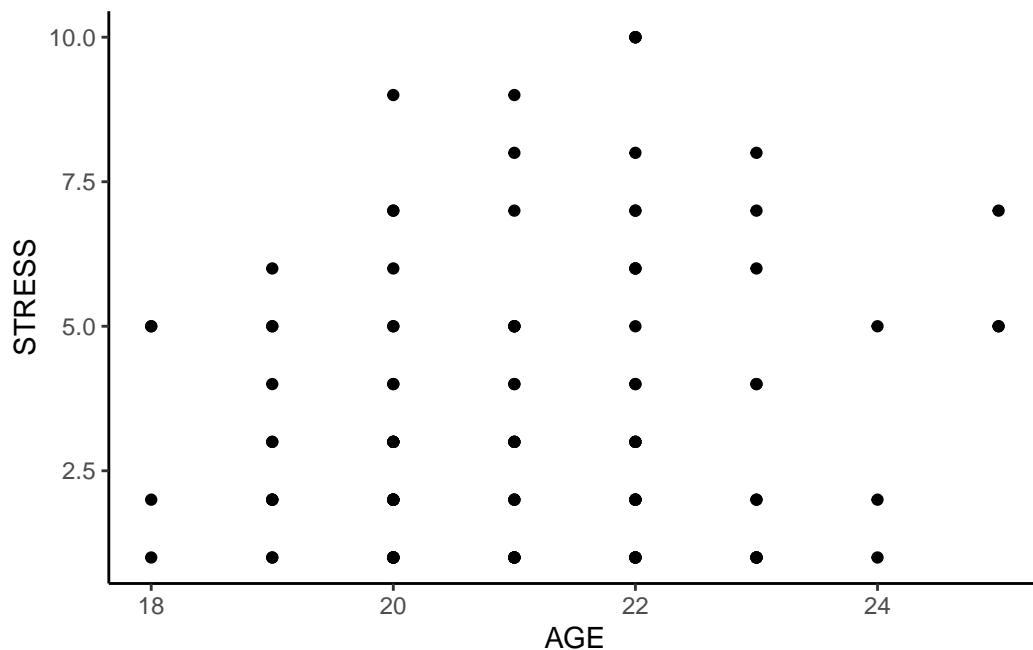
Warning: Removed 31 rows containing missing values or values outside the scale range
(`geom_point()`).



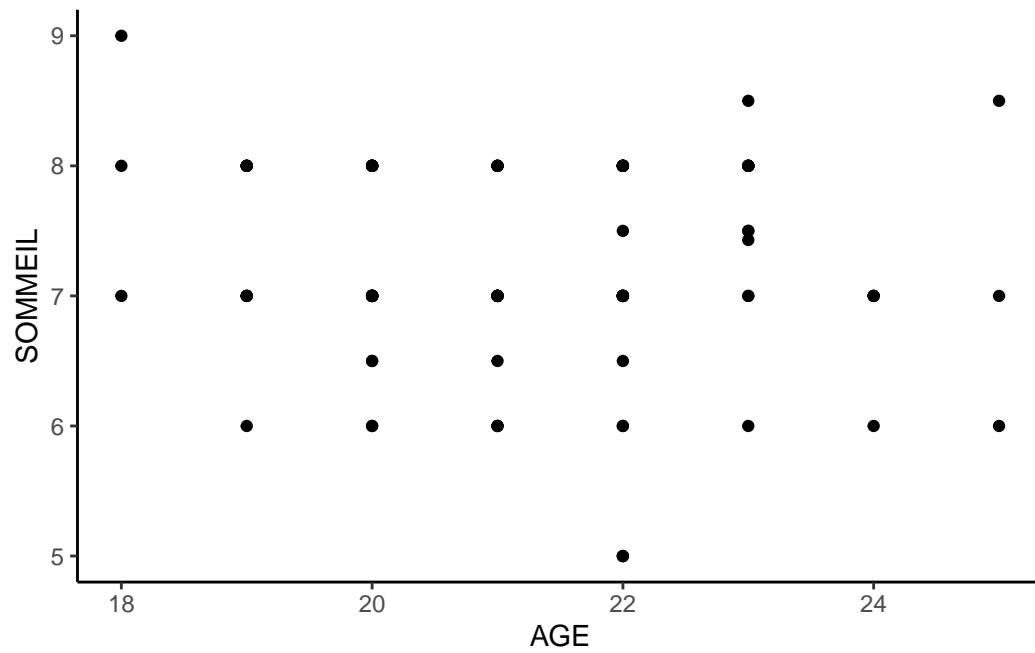
Warning: Removed 28 rows containing missing values or values outside the scale range
(`geom_point()`).



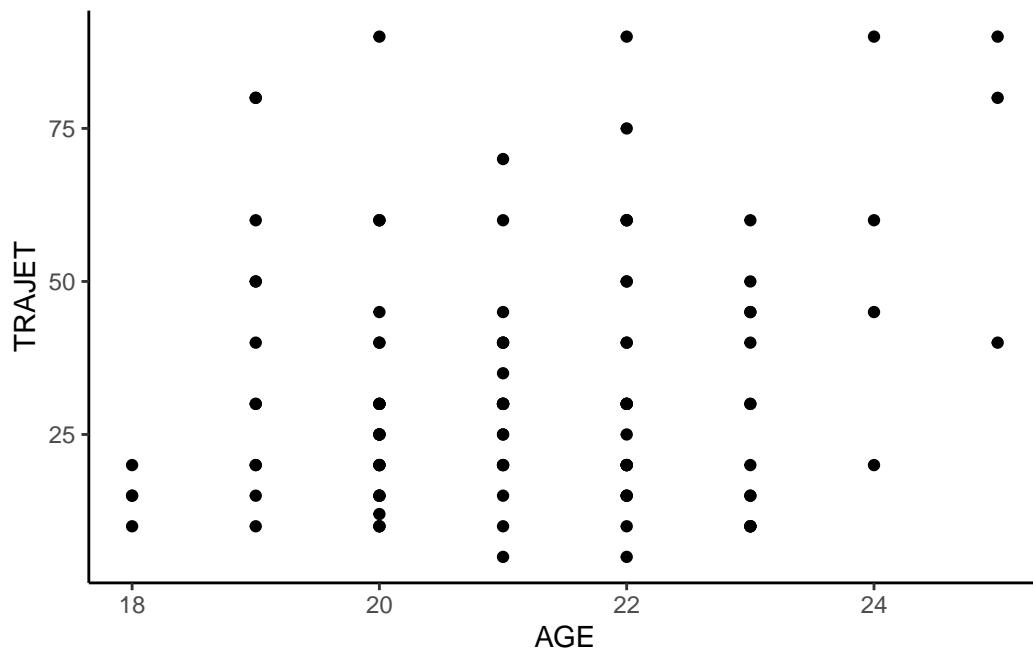
Warning: Removed 21 rows containing missing values or values outside the scale range (`geom_point()`).



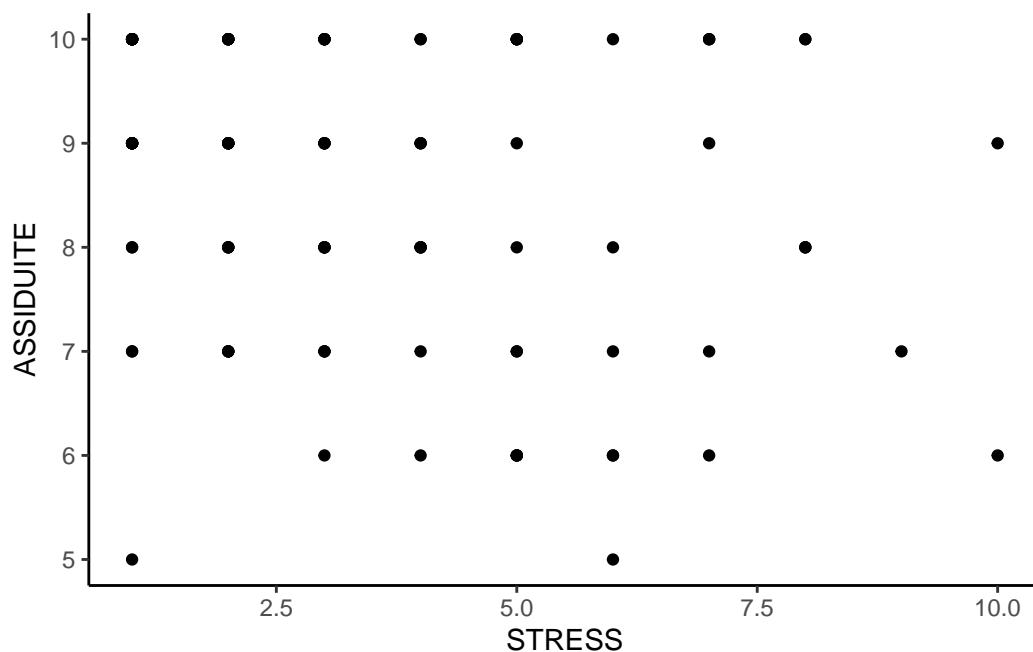
Warning: Removed 32 rows containing missing values or values outside the scale range (`geom_point()`).



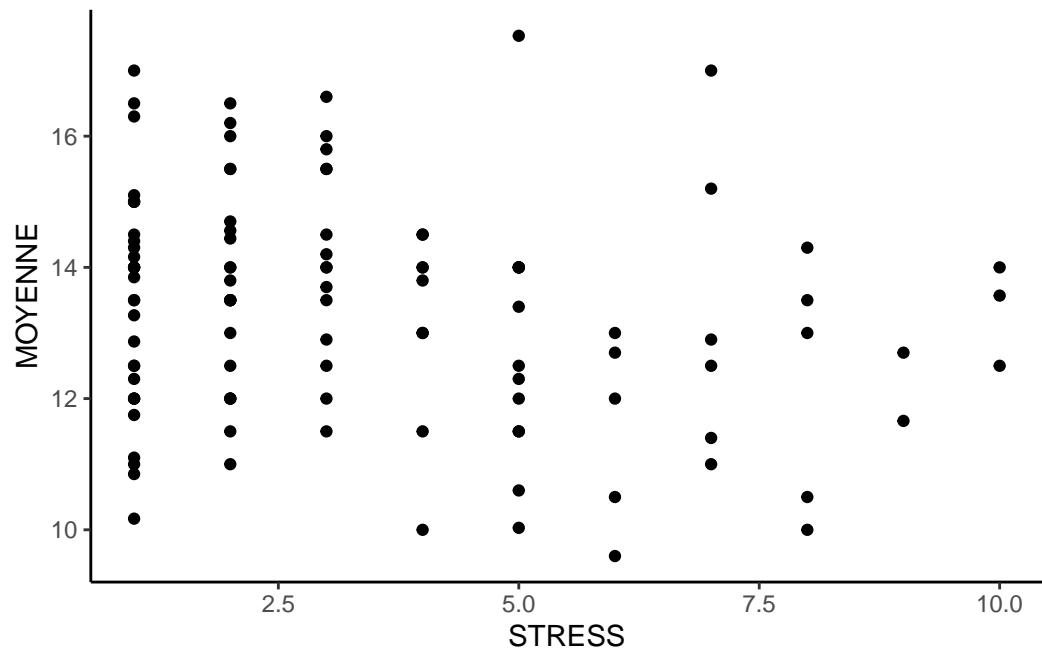
Warning: Removed 24 rows containing missing values or values outside the scale range (`geom_point()`).



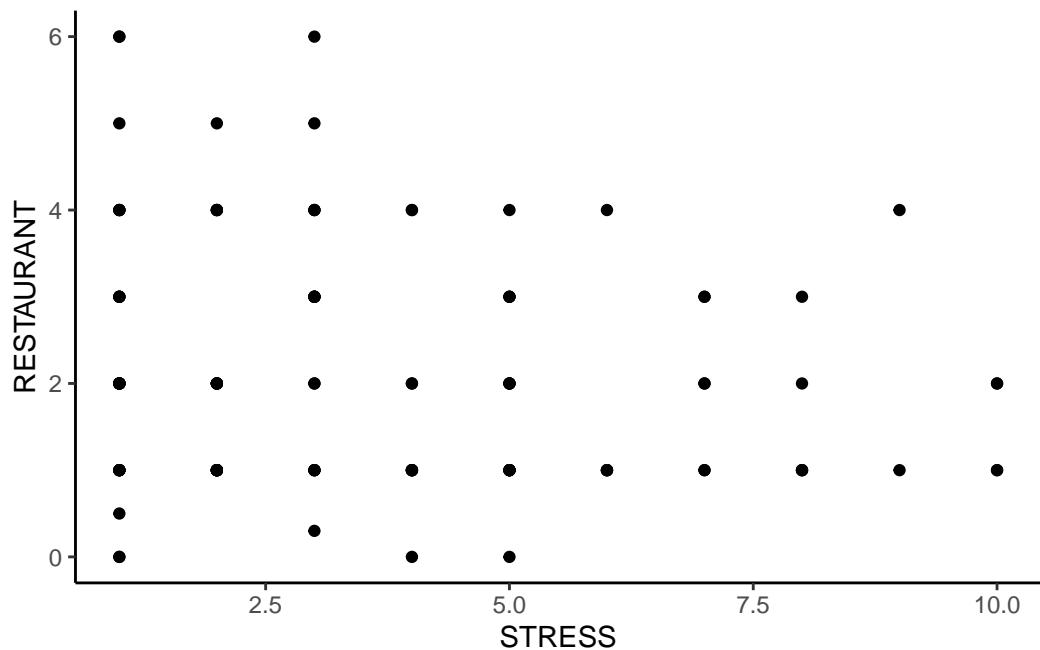
Warning: Removed 24 rows containing missing values or values outside the scale range (`geom_point()`).



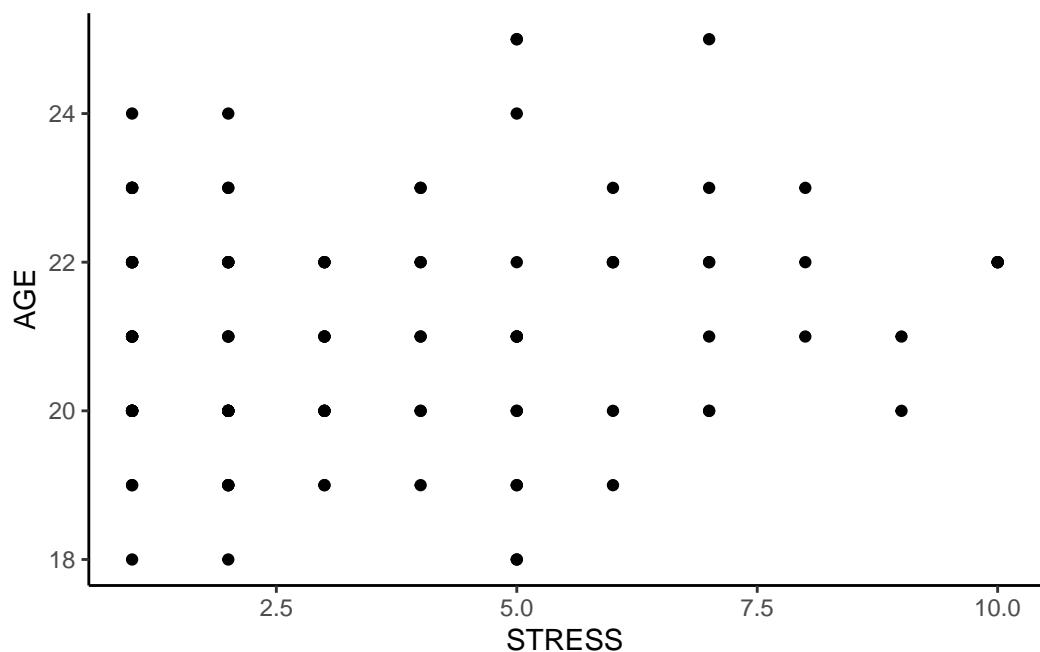
Warning: Removed 24 rows containing missing values or values outside the scale range
(`geom_point()`).



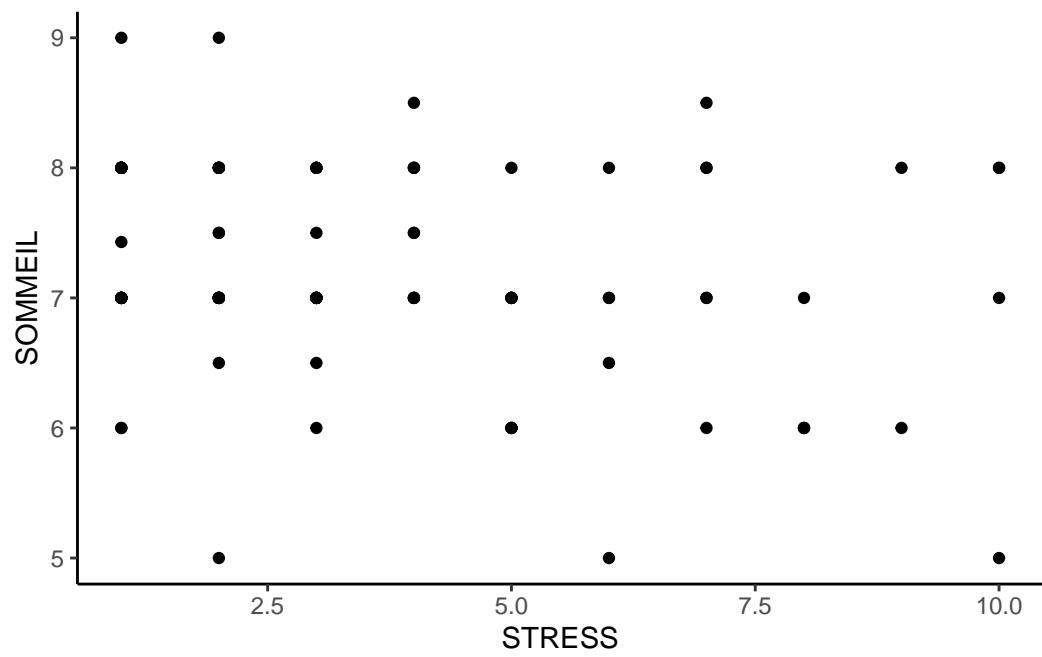
Warning: Removed 21 rows containing missing values or values outside the scale range
(`geom_point()`).



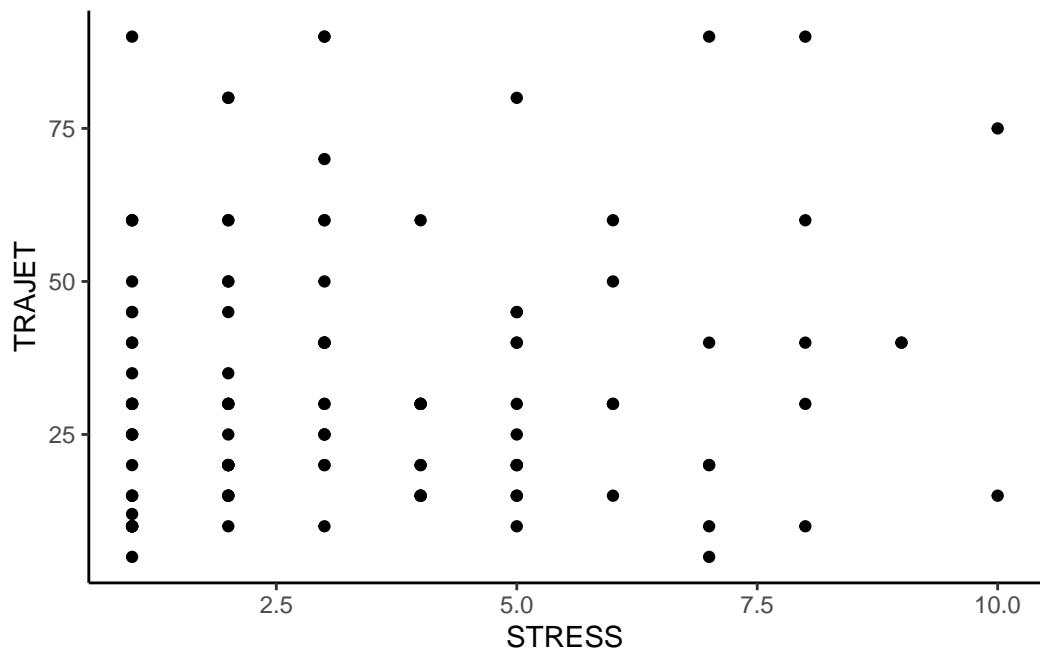
Warning: Removed 21 rows containing missing values or values outside the scale range
(`geom_point()`).



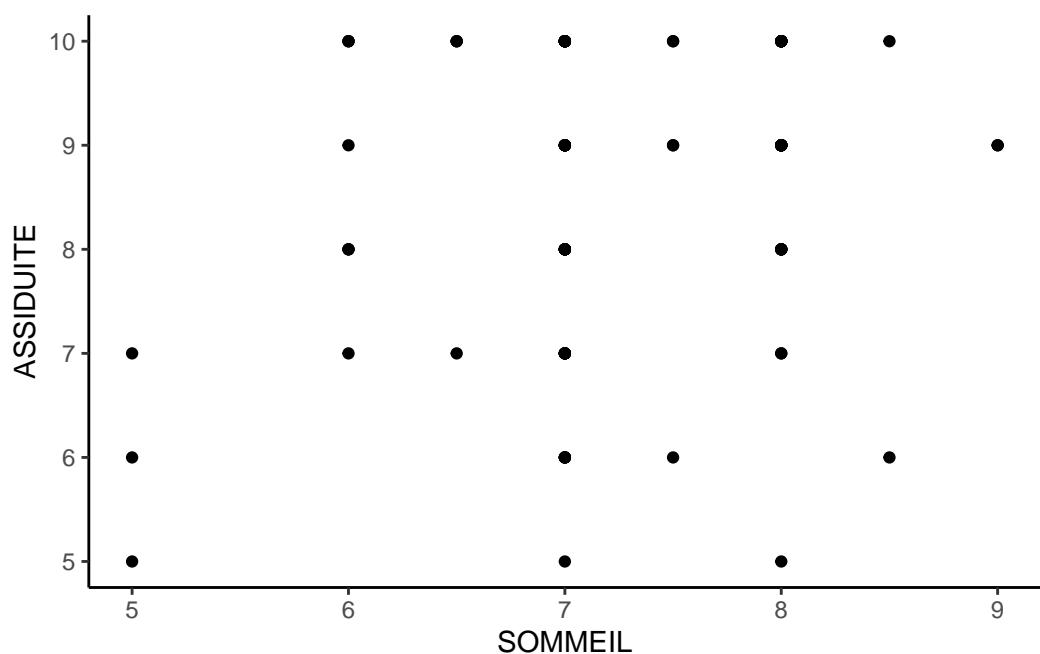
Warning: Removed 27 rows containing missing values or values outside the scale range (`geom_point()`).



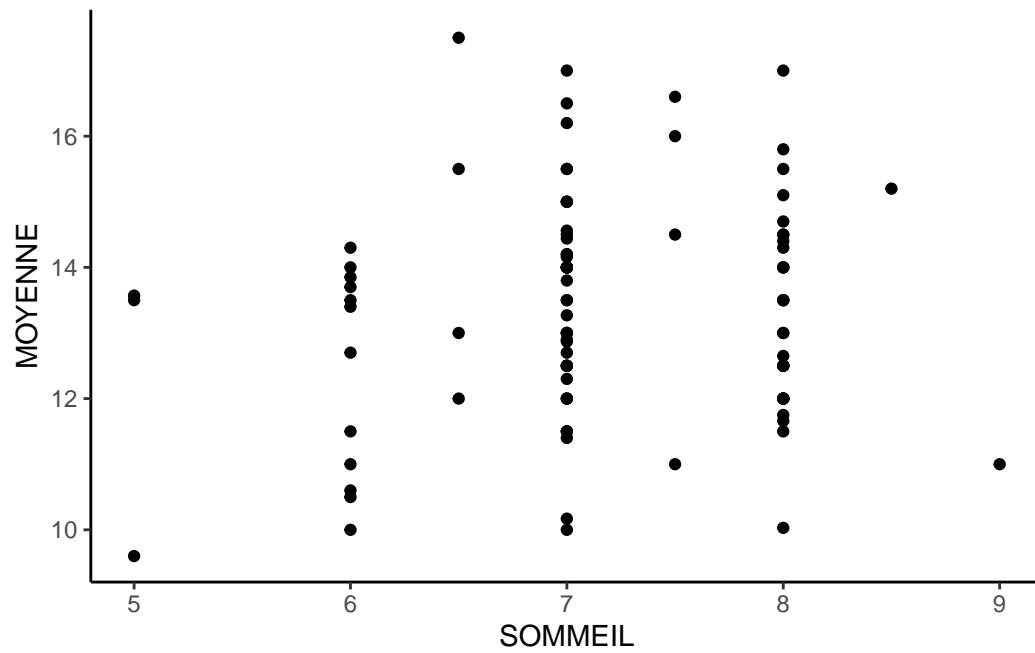
Warning: Removed 18 rows containing missing values or values outside the scale range (`geom_point()`).



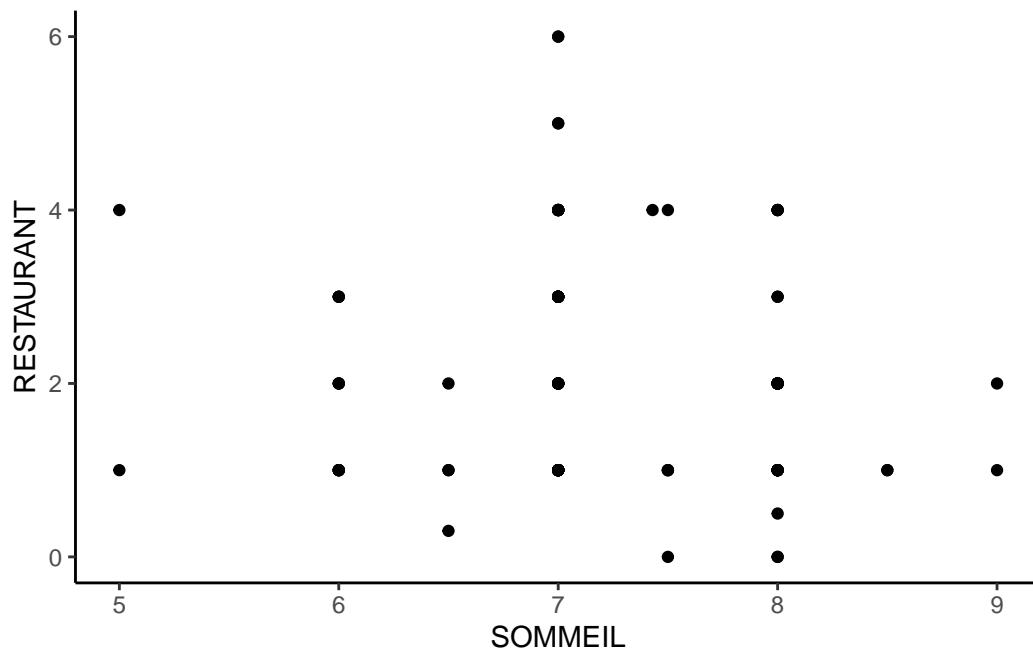
Warning: Removed 35 rows containing missing values or values outside the scale range (`geom_point()`).



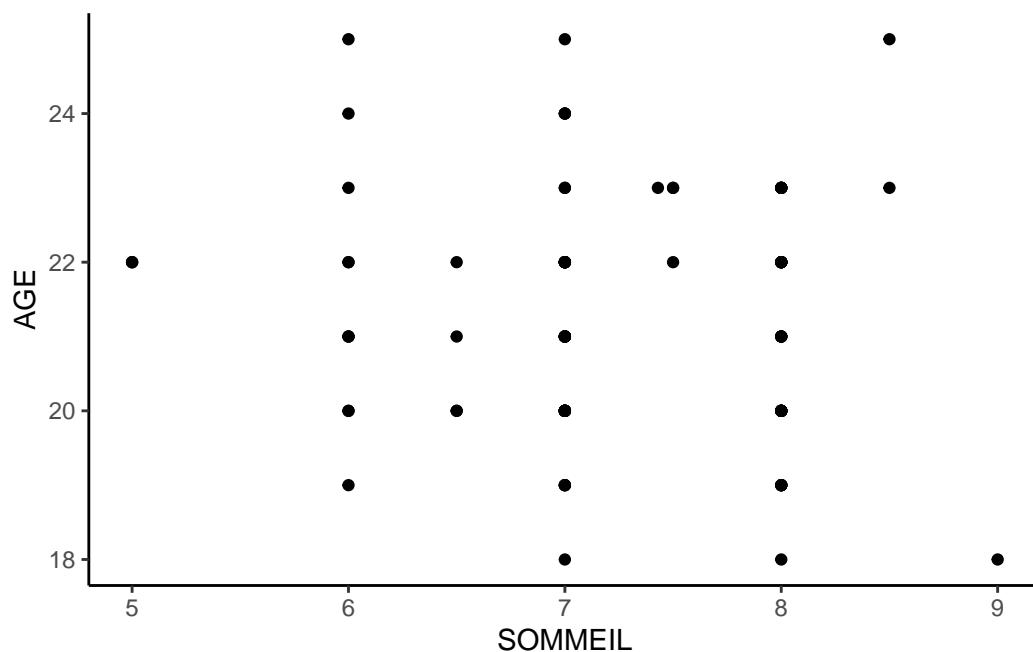
Warning: Removed 34 rows containing missing values or values outside the scale range
(`geom_point()`).



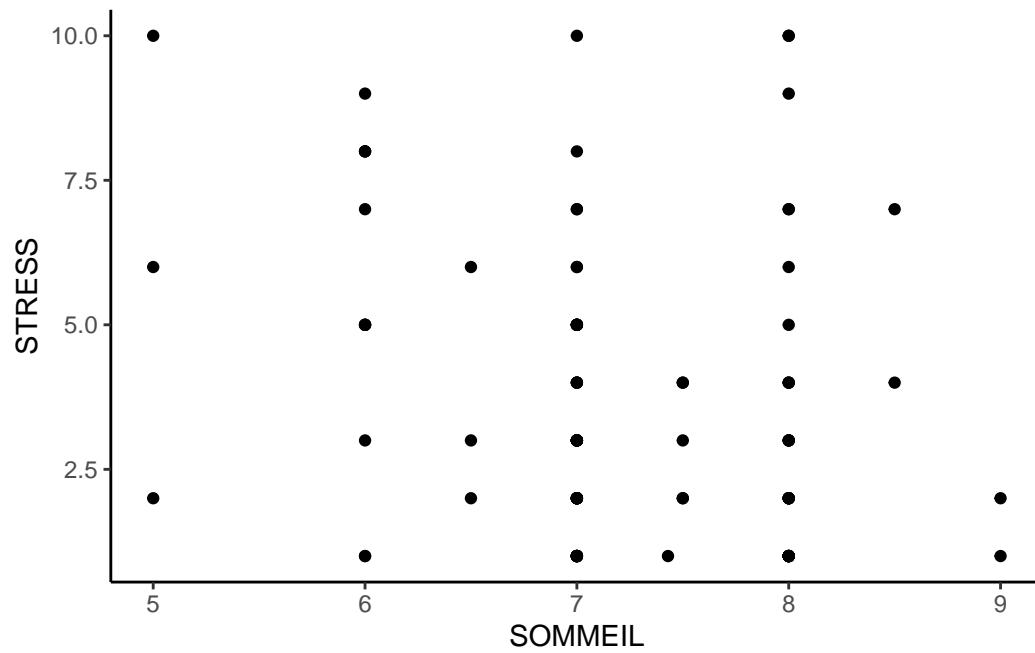
Warning: Removed 33 rows containing missing values or values outside the scale range
(`geom_point()`).



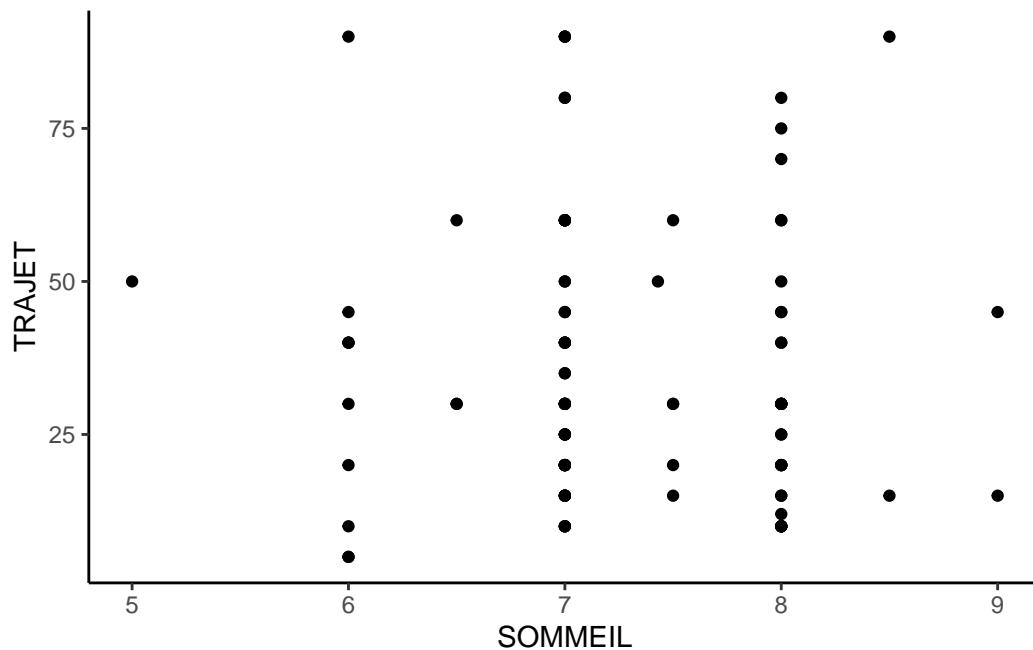
Warning: Removed 32 rows containing missing values or values outside the scale range (`geom_point()`).



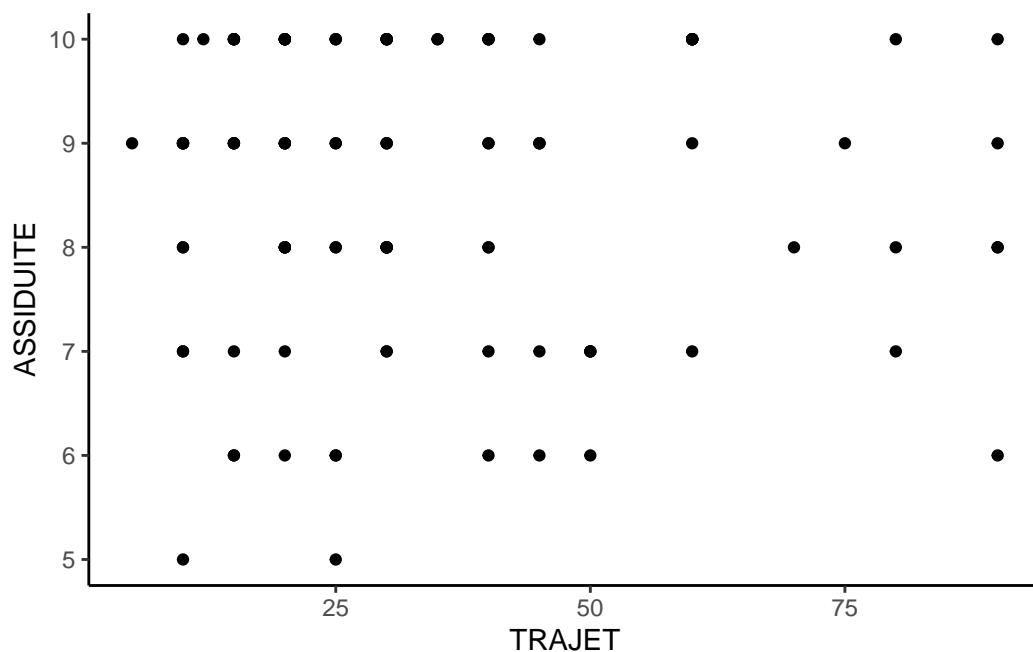
Warning: Removed 27 rows containing missing values or values outside the scale range (`geom_point()`).



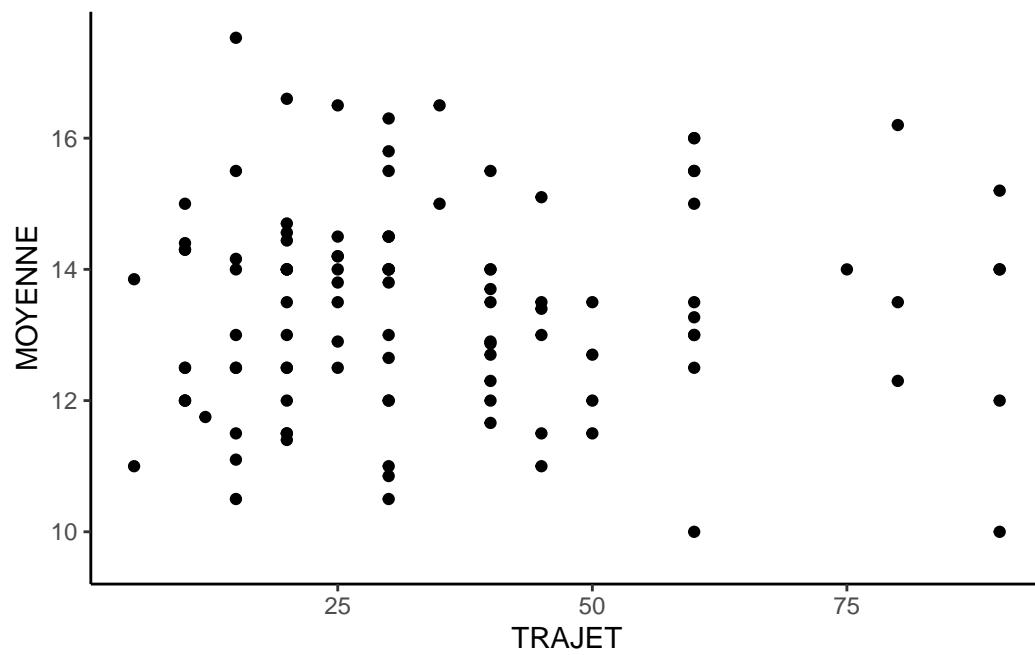
Warning: Removed 30 rows containing missing values or values outside the scale range (`geom_point()`).



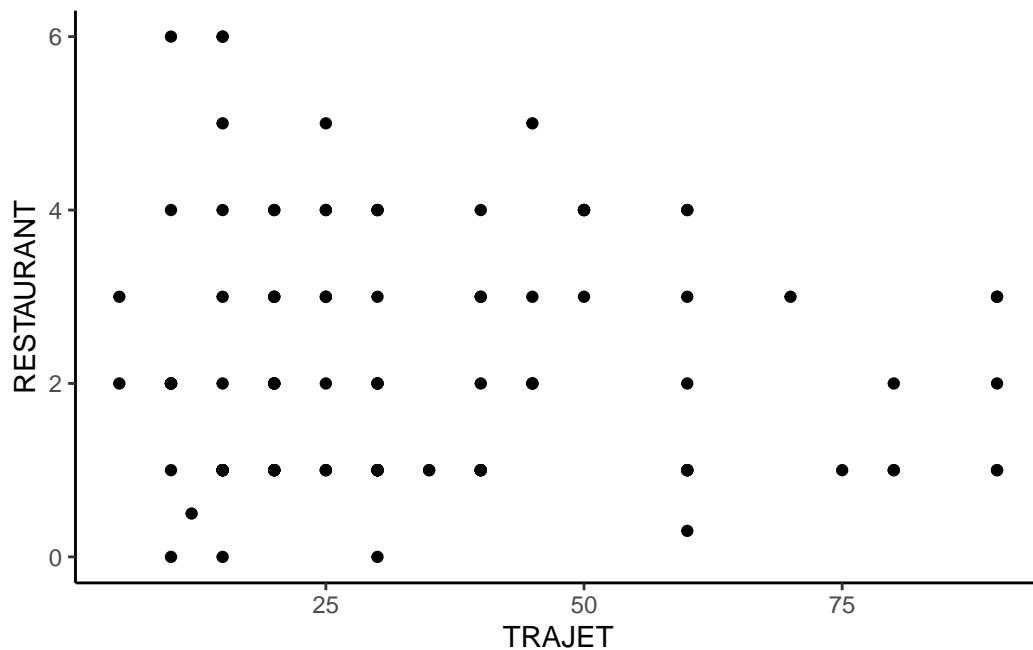
Warning: Removed 24 rows containing missing values or values outside the scale range (`geom_point()`).



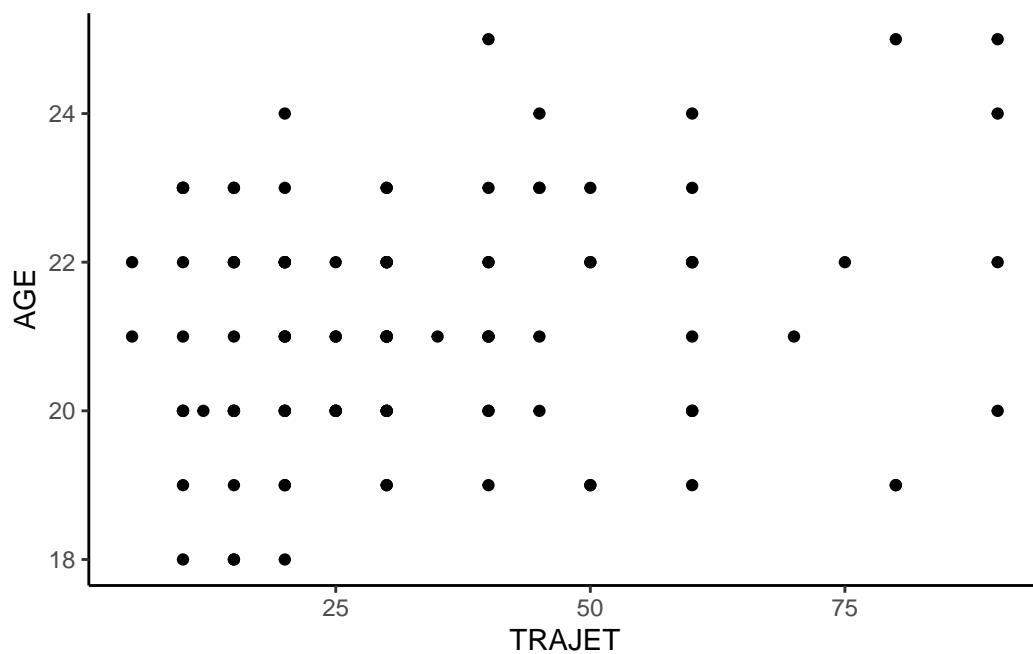
Warning: Removed 27 rows containing missing values or values outside the scale range (`geom_point()`).



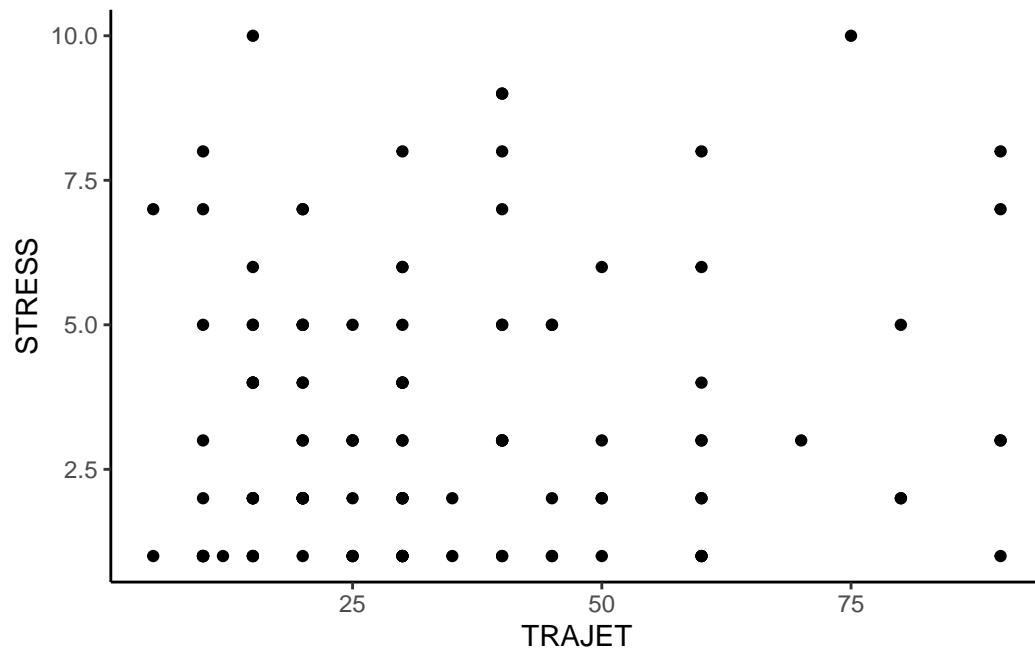
Warning: Removed 25 rows containing missing values or values outside the scale range (`geom_point()`).



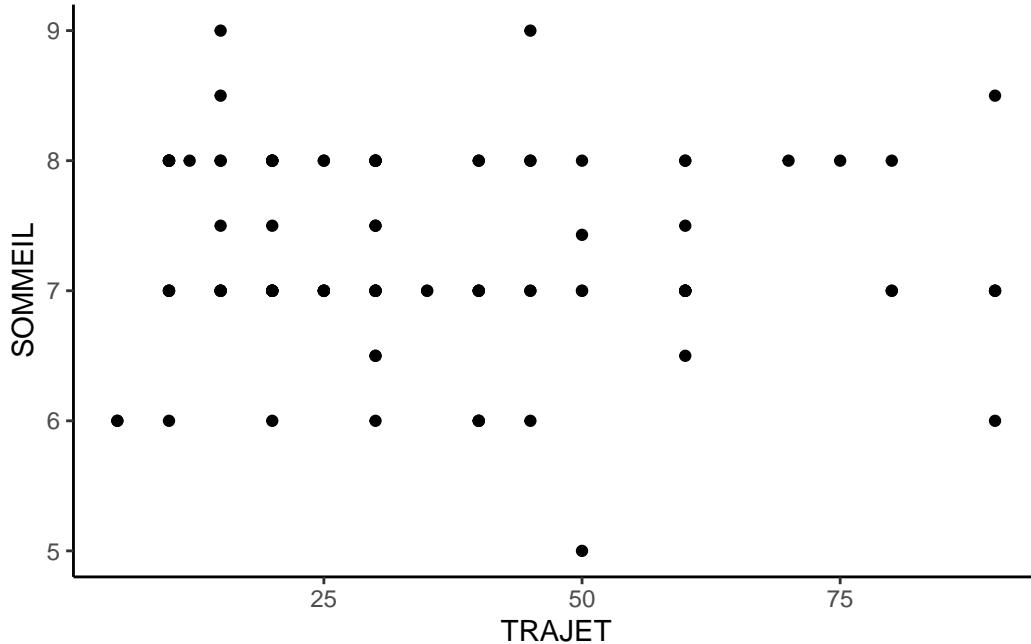
Warning: Removed 24 rows containing missing values or values outside the scale range (`geom_point()`).



Warning: Removed 18 rows containing missing values or values outside the scale range (`geom_point()`).



Warning: Removed 30 rows containing missing values or values outside the scale range (`geom_point()`).



Cette série de graphiques représente les nuages de points pour chaque combinaison de deux variables quantitatives distinctes. Cela permet d'explorer visuellement les relations entre chaque paire de variables.

Vérifions désormais la corrélation entre nos variables quantitatives.

```
# Matrice de corrélation de Pearson

cor(Budget2[, quantis],
use="complete.obs")
```

	ASSIDUITE	MOYENNE	RESTAURANT	AGE	STRESS
ASSIDUITE	1.00000000	0.15158716	-0.33348878	0.2904578	-0.31972504
MOYENNE	0.15158716	1.00000000	-0.15560171	0.1580872	-0.29544956
RESTAURANT	-0.33348878	-0.15560171	1.00000000	-0.1106026	-0.04461918
AGE	0.29045777	0.15808718	-0.11060258	1.0000000	0.11761857
STRESS	-0.31972504	-0.29544956	-0.04461918	0.1176186	1.00000000
SOMMEIL	0.06499619	0.21185326	-0.08916240	-0.1434130	-0.24910496
TRAJET	-0.06885282	0.06623986	-0.01764513	0.2104741	0.19606971
	SOMMEIL	TRAJET			
ASSIDUITE	0.064996193	-0.068852819			
MOYENNE	0.211853259	0.066239860			

```
RESTAURANT -0.089162398 -0.017645132
AGE         -0.143413012  0.210474054
STRESS      -0.249104959  0.196069711
SOMMEIL     1.000000000  0.002950475
TRAJET       0.002950475  1.000000000
```

```
# Test de Shapiro-Wilk sur chaque variable quantitative

for (i in seq(1, length(quantis))) {
  res <- shapiro.test(Budget2[[quantis[i]]])
  cat(quantis[i])
  print(res)
}
```

ASSIDUITE

Shapiro-Wilk normality test

```
data: Budget2[[quantis[i]]]
W = 0.86394, p-value = 4.624e-09
```

MOYENNE

Shapiro-Wilk normality test

```
data: Budget2[[quantis[i]]]
W = 0.98762, p-value = 0.3594
```

RESTAURANT

Shapiro-Wilk normality test

```
data: Budget2[[quantis[i]]]
W = 0.8829, p-value = 2.879e-08
```

AGE

Shapiro-Wilk normality test

```
data: Budget2[[quantis[i]]]
W = 0.9503, p-value = 0.0002128
```

STRESS

Shapiro-Wilk normality test

```
data: Budget2[[quantis[i]]]
W = 0.87215, p-value = 4.518e-09
```

SOMMEIL

Shapiro-Wilk normality test

```
data: Budget2[[quantis[i]]]
W = 0.88526, p-value = 6.18e-08
```

TRAJET

Shapiro-Wilk normality test

```
data: Budget2[[quantis[i]]]
W = 0.89104, p-value = 4.742e-08
```

Hormis MOYENNE avec une p-value supérieure à 0,05, toutes les autres variables ne suivent pas la loi normale. La corrélation de Spearman étant robuste, c'est à dire indépendante de la distribution des données, il faut calculer le coefficient de corrélation de Spearman.

```
# Matrice de corrélation de Spearman

cor(Budget2[, quantis],
  use="complete.obs", method = c("spearman"))
```

	ASSIDUITE	MOYENNE	RESTAURANT	AGE	STRESS
ASSIDUITE	1.00000000	0.13845147	-0.34479683	0.32390662	-0.36976643
MOYENNE	0.13845147	1.00000000	-0.13078867	0.19745682	-0.27571166
RESTAURANT	-0.34479683	-0.13078867	1.00000000	-0.07571144	-0.04366170
AGE	0.32390662	0.19745682	-0.07571144	1.00000000	0.04366002
STRESS	-0.36976643	-0.27571166	-0.04366170	0.04366002	1.00000000
SOMMEIL	0.04500434	0.21461055	-0.05330807	-0.12785551	-0.32663970
TRAJET	-0.04042760	0.07616135	0.01976602	0.13788131	0.20309171
	SOMMEIL	TRAJET			
ASSIDUITE	0.045004340	-0.040427601			
MOYENNE	0.214610546	0.076161350			
RESTAURANT	-0.053308069	0.019766025			
AGE	-0.127855507	0.137881315			
STRESS	-0.326639699	0.203091708			

```
SOMMEIL      1.000000000 -0.006951733
TRAJET       -0.006951733  1.000000000
```

```
# Diagramme pour visualiser la matrice de corrélation
```

```
Budget2 |>
  select(quantis) |>
  drop_na() |>
  cor(method = "spearman") |>
  corrplot::corrplot.mixed()
```

Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.

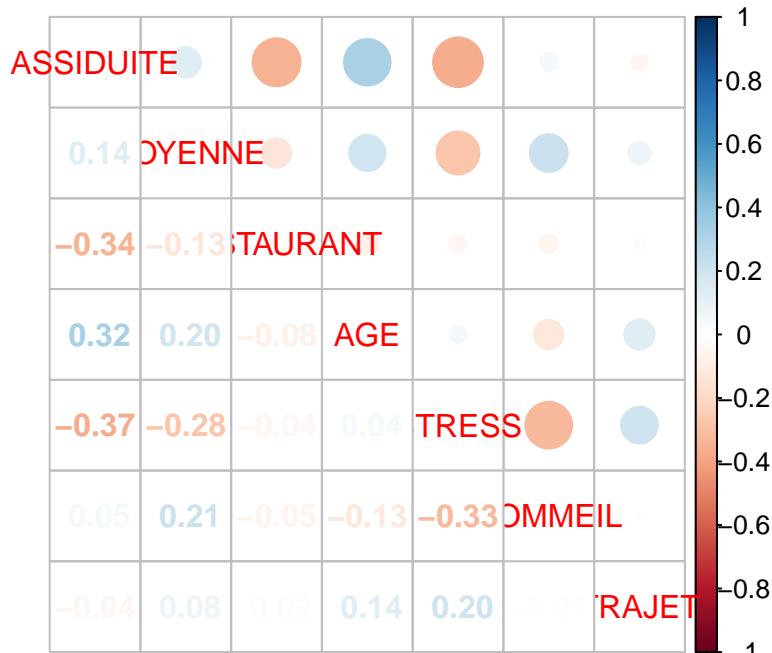
Was:

```
data %>% select(quantis)
```

Now:

```
data %>% select(all_of(quantis))
```

See <<https://tidyselect.r-lib.org/reference/faq-external-vector.html>>.



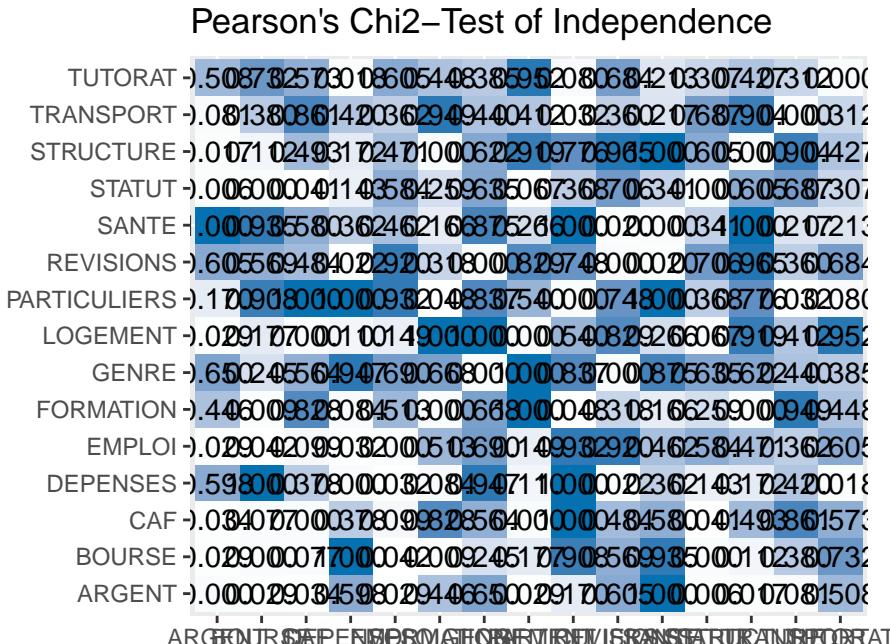
Chaque coefficient est bien inférieur à la valeur absolue de 0,6. Il ne semble pas y avoir de corrélation entre ces variables.

2. Variables qualitatives - qualitatives

Pour effectuer cette analyse, nous générerons une matrice évaluant l'indépendance entre chaque paire de variables catégorielles à l'aide du test du chi-deux.

```
# Matrice d'indépendance des variables qualitatives

ind_qualis = as.data.frame(Budget2[,qualis])
sjp.chi2(ind_qualis, show.legend = TRUE)
```

L'analyse de la matrice d'indépendance des variables qualitatives est primordiale pour vérifier s'il existe des liens significatifs entre les variables qualitatives. Le coefficient varie entre 0 et 1. Une valeur proche de 0 indiquera une forte association entre les deux variables, entraînant une dépendance plus élevée. Réciproquement, une valeur proche de 1 signifiera une faible association.

Nous pouvons constater que plusieurs intensités sont proches de 0, voire égale 0. Cela révèle une certaine dépendance entre les variables qualitatives.

3. Variables quantitatives - qualitatives

Étudions à présent comment les variables quantitatives varient en fonction de chaque modalité des variables qualitatives.

```
# Visualisation des relations des quatre premières variables quantitatives avec chaque variable qualitative
```

```
for (qt in quantis[1:4]) {
  for (ql in qualis) {
    p <- Budget2 |>
      ggplot() +
      aes_string(x = qt, y = ql, color = ql) +
      geom_violin() +
      geom_boxplot()
```

```

    geom_boxplot(width = 0.3, alpha = 0.5) +
    geom_jitter(alpha = 0.3) +
    theme_minimal() +
    labs(title = paste("Distribution de", qt, "en fonction de", ql))

  print(p)
}
}

```

Warning: `aes_string()` was deprecated in ggplot2 3.0.0.

i Please use tidy evaluation idioms with `aes()``.

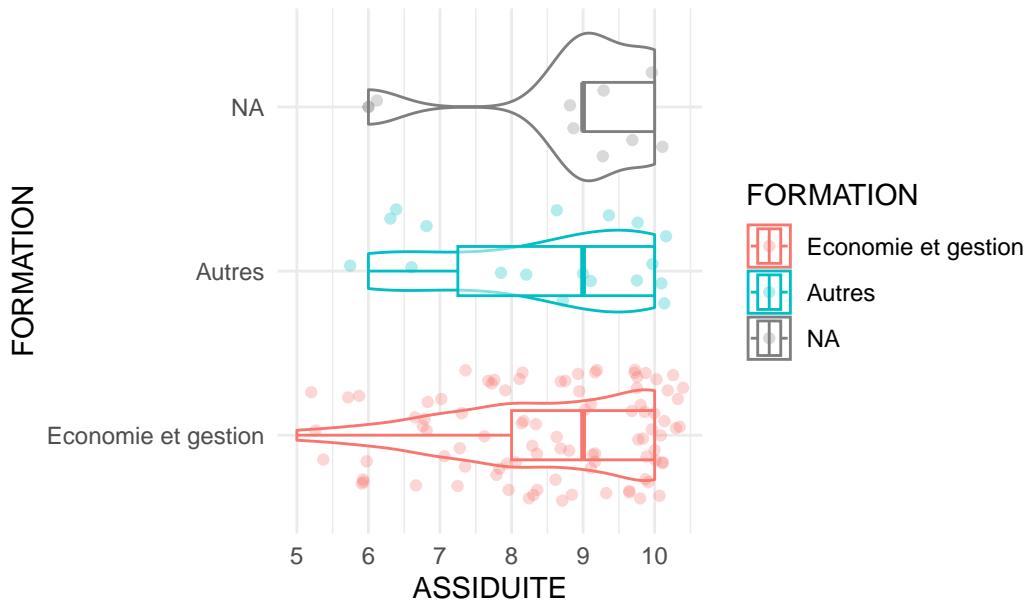
i See also `vignette("ggplot2-in-packages")` for more information.

Warning: Removed 16 rows containing non-finite outside the scale range (`stat_ydensity()``).

Warning: Removed 16 rows containing non-finite outside the scale range (`stat_boxplot()``).

Warning: Removed 16 rows containing missing values or values outside the scale range (`geom_point()``).

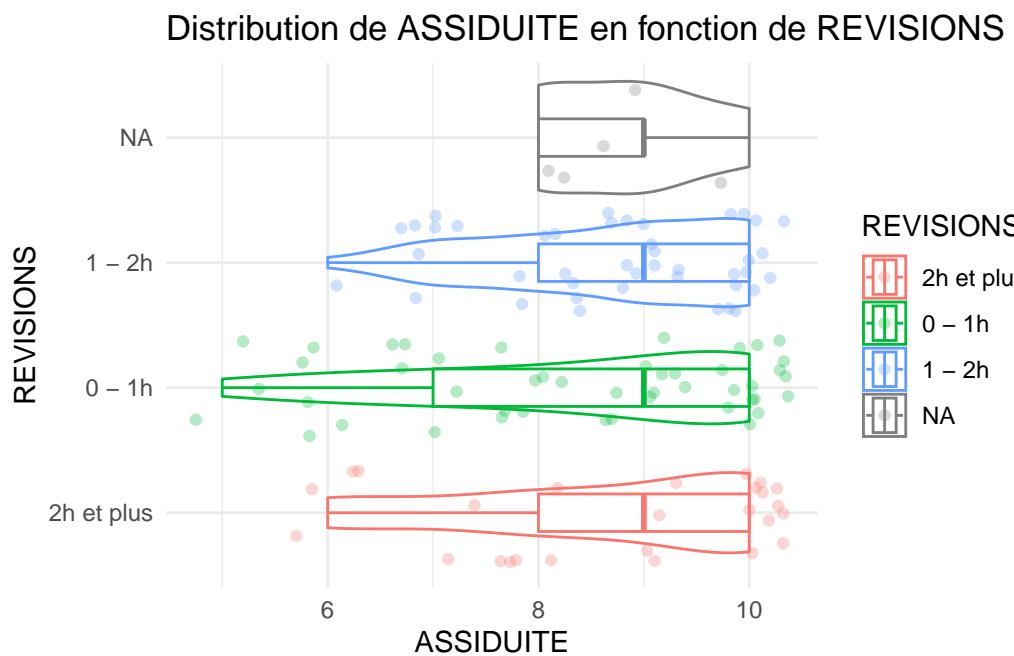
Distribution de ASSIDUITE en fonction de FORMA



Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

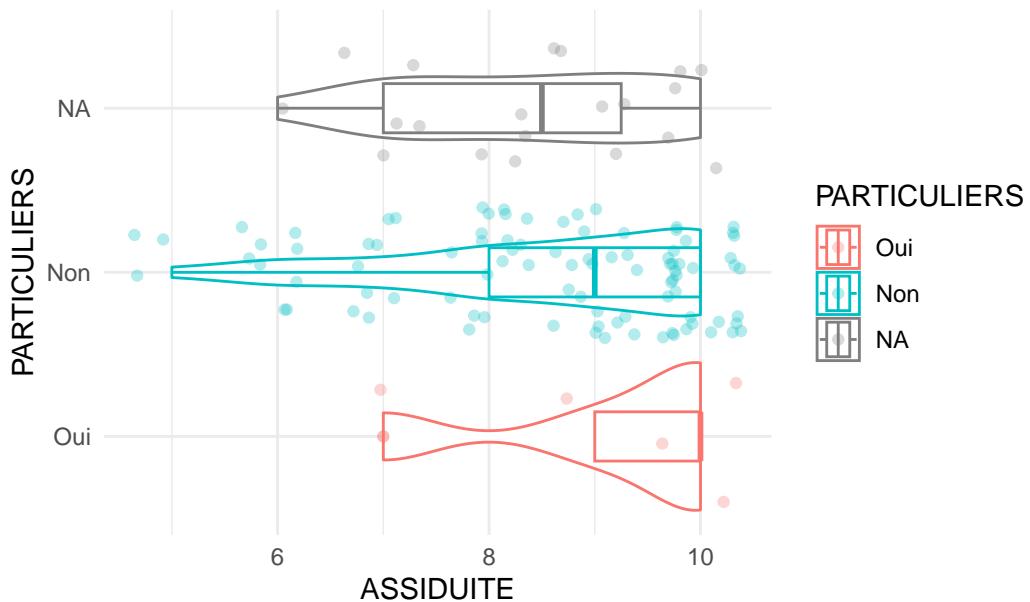


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de PARTICULARIERS

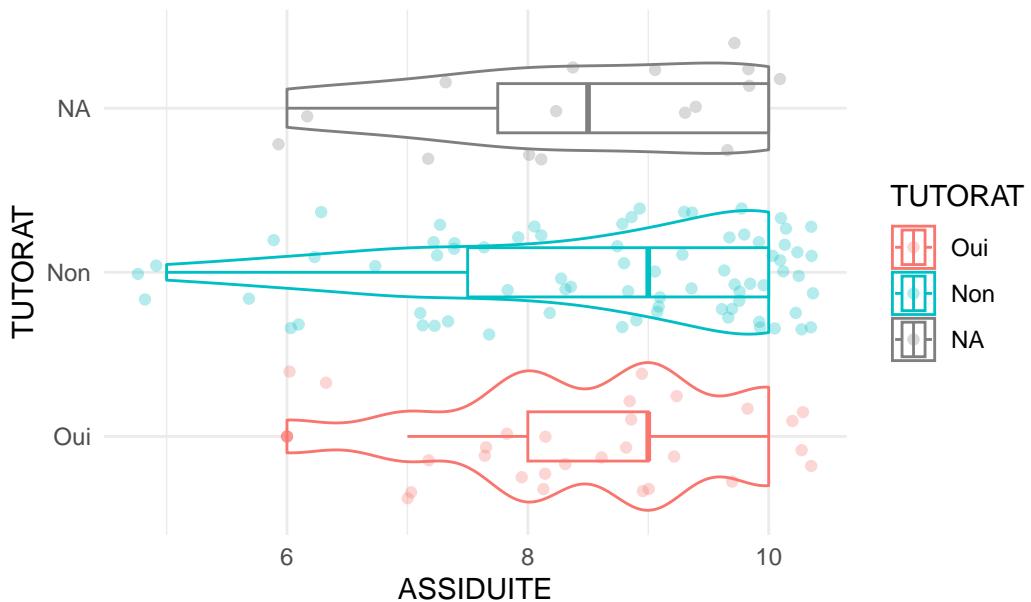


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de TUTORAT

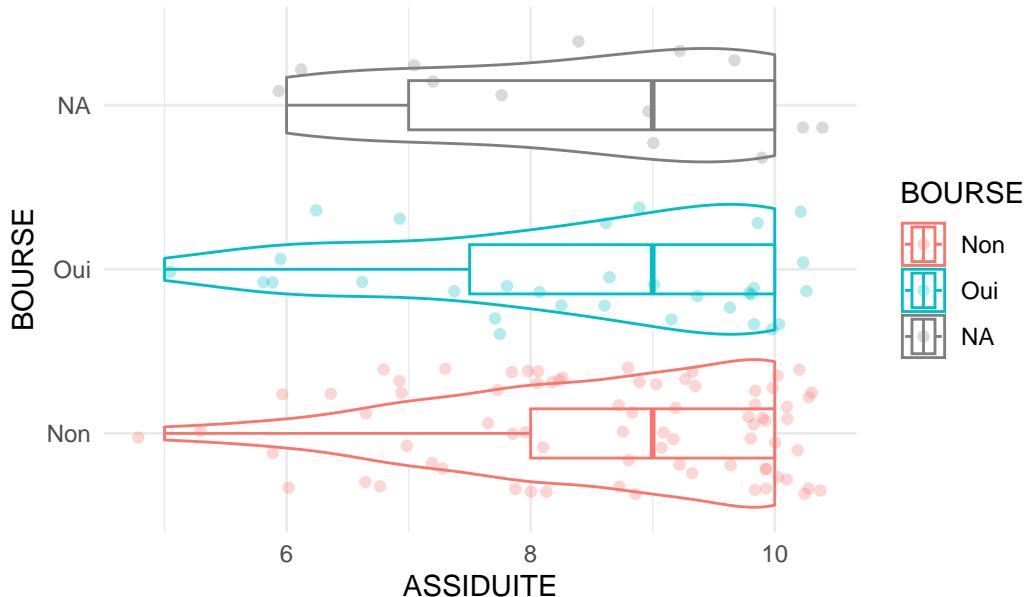


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de BOURSE

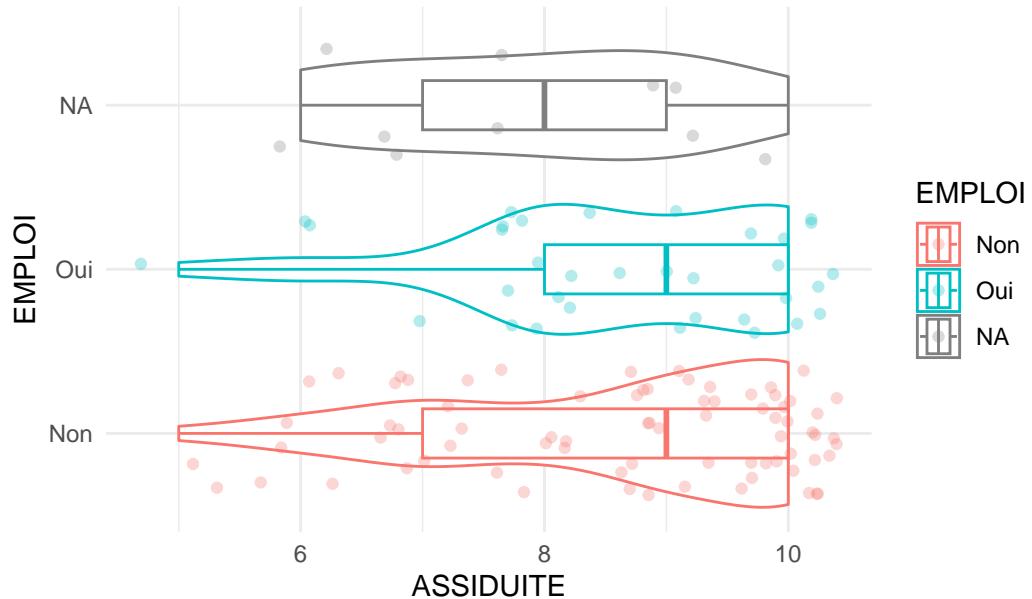


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de EMPLOI

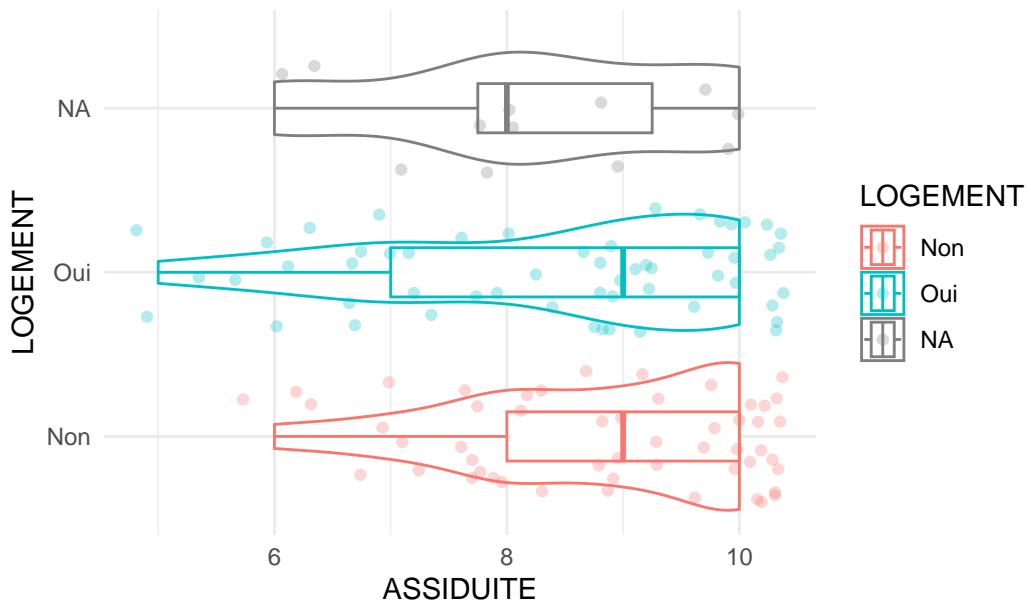


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de LOGEMENT

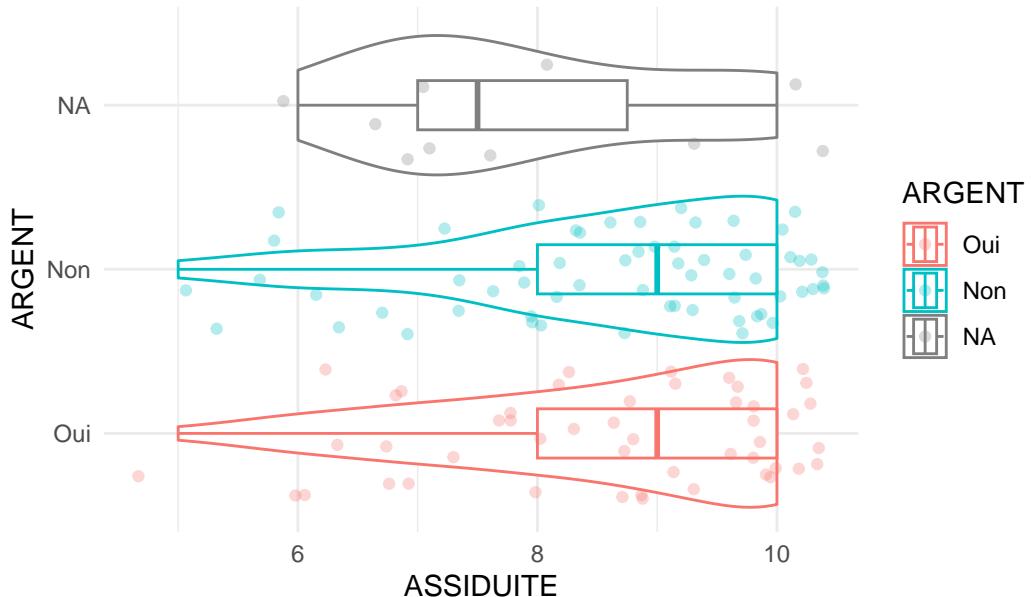


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de ARGENT

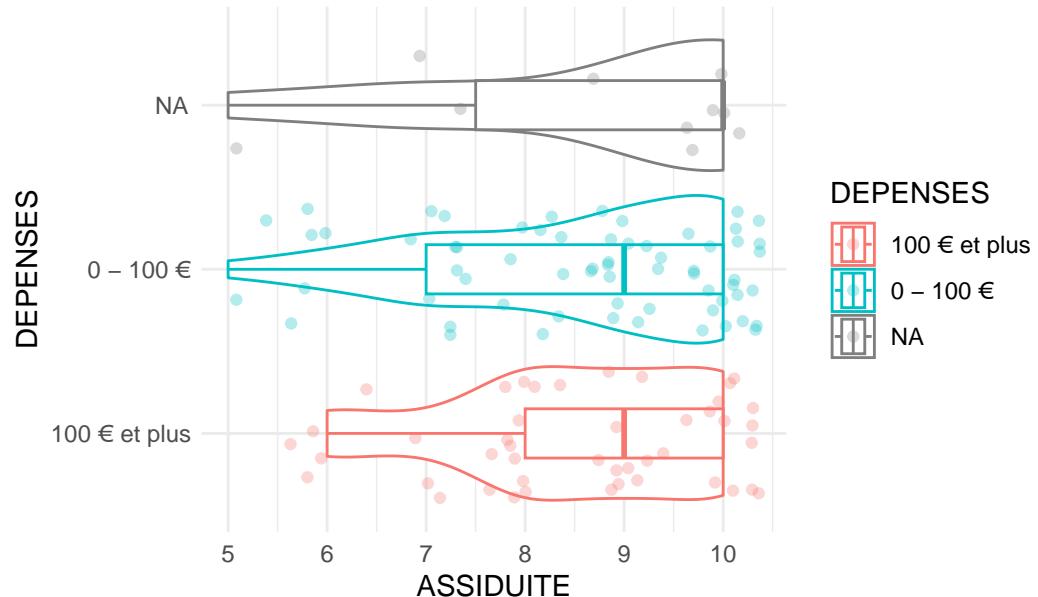


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de DEPENSES

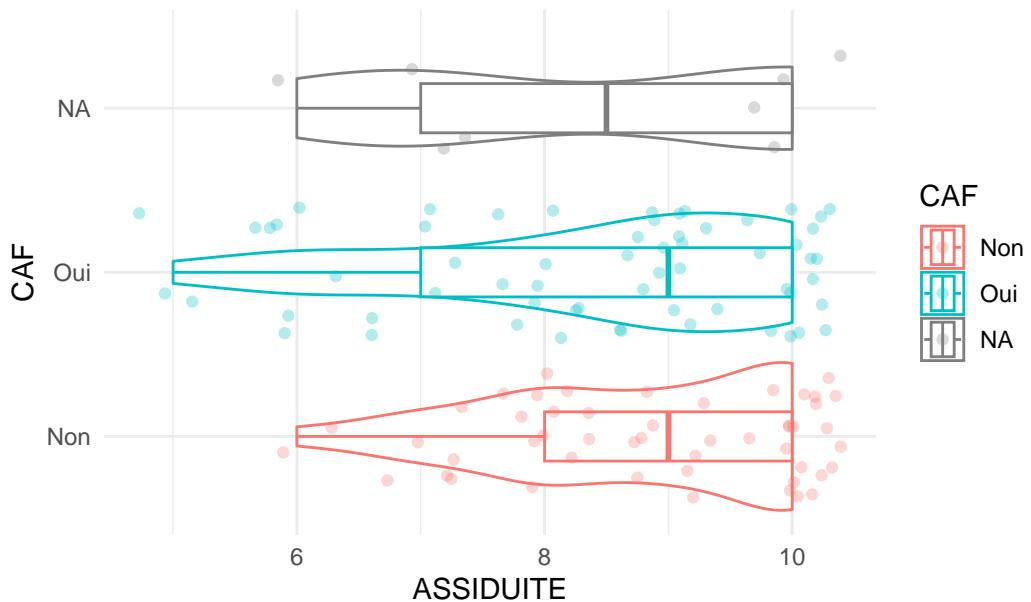


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de CAF

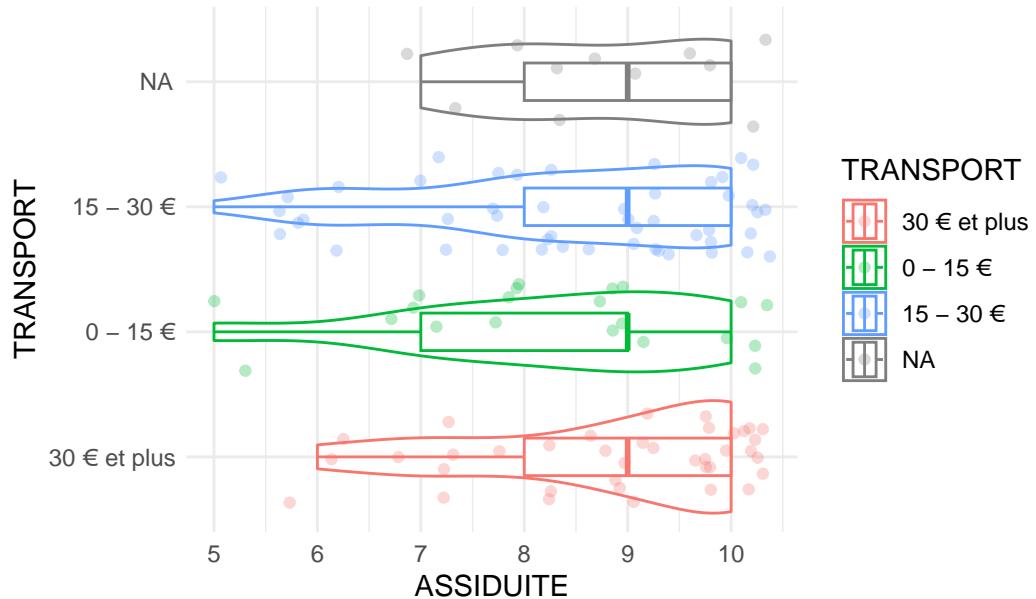


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de TRANSPORT

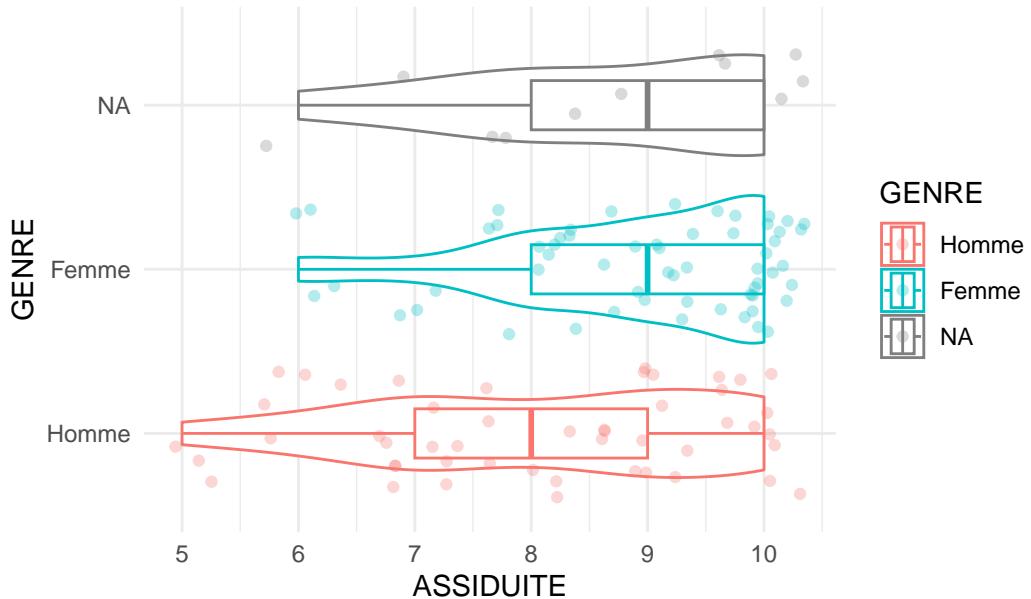


```
Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).
```

```
Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```
Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Distribution de ASSIDUITE en fonction de GENRE

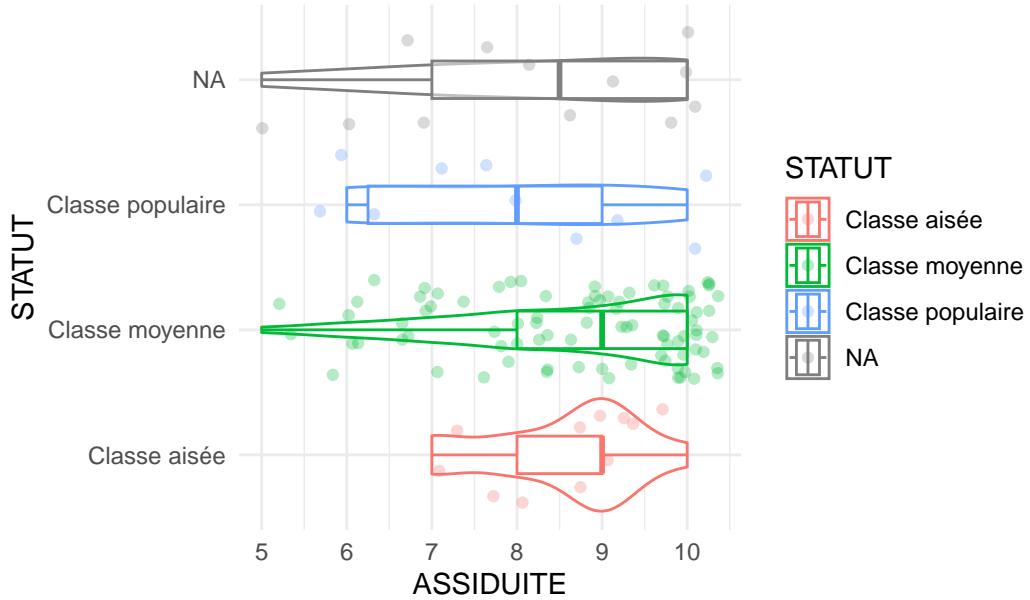


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de STATUT

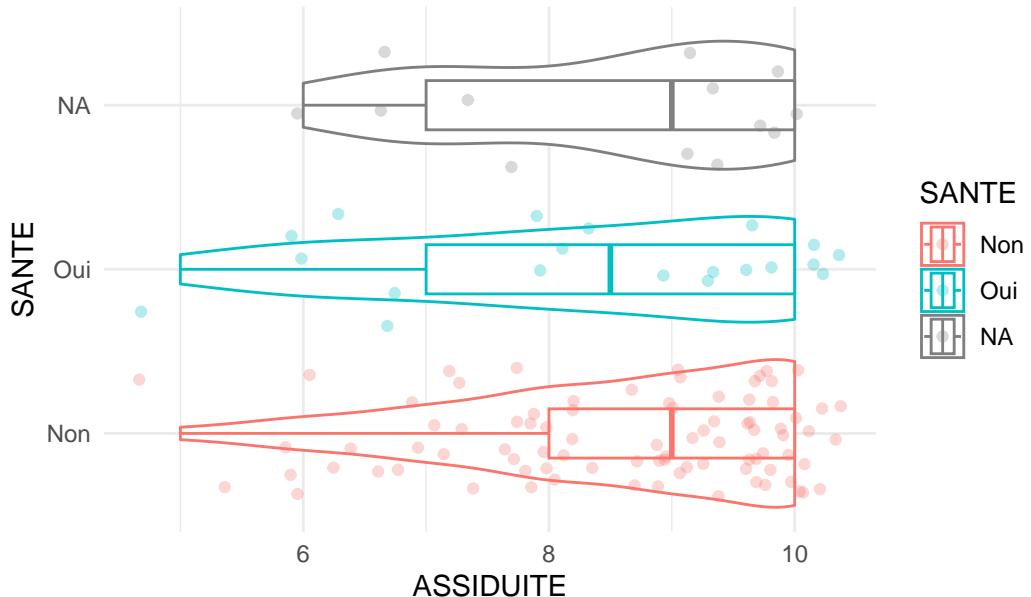


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de SANTE

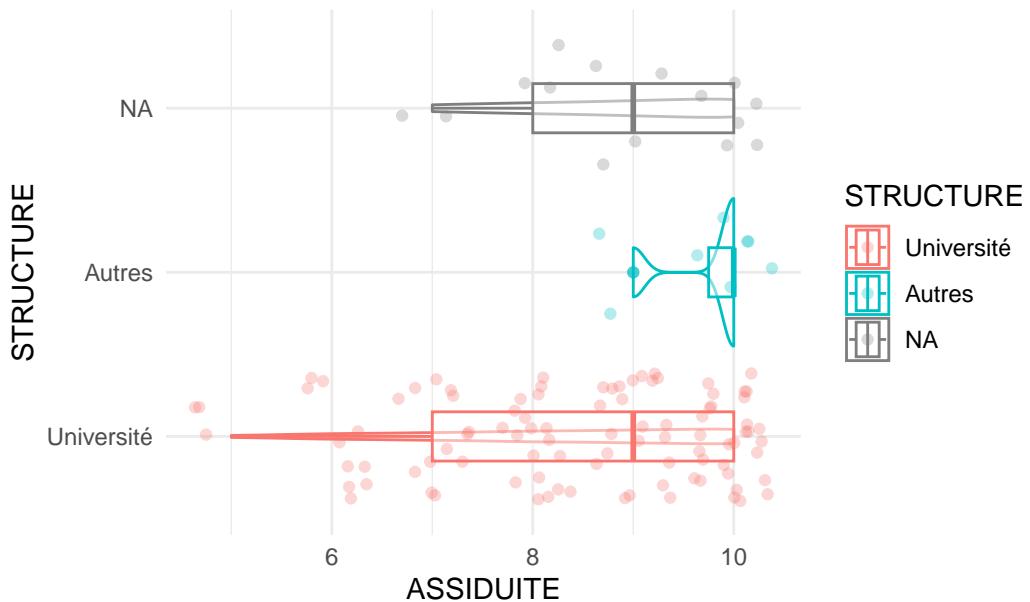


Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 16 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de ASSIDUITE en fonction de STRUCTURE

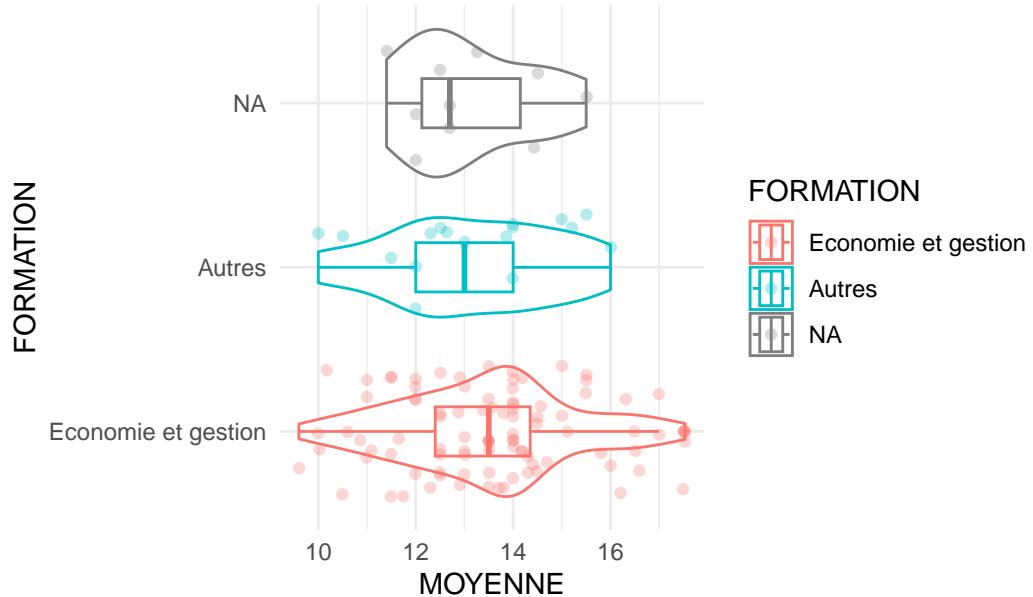


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de FORMAT

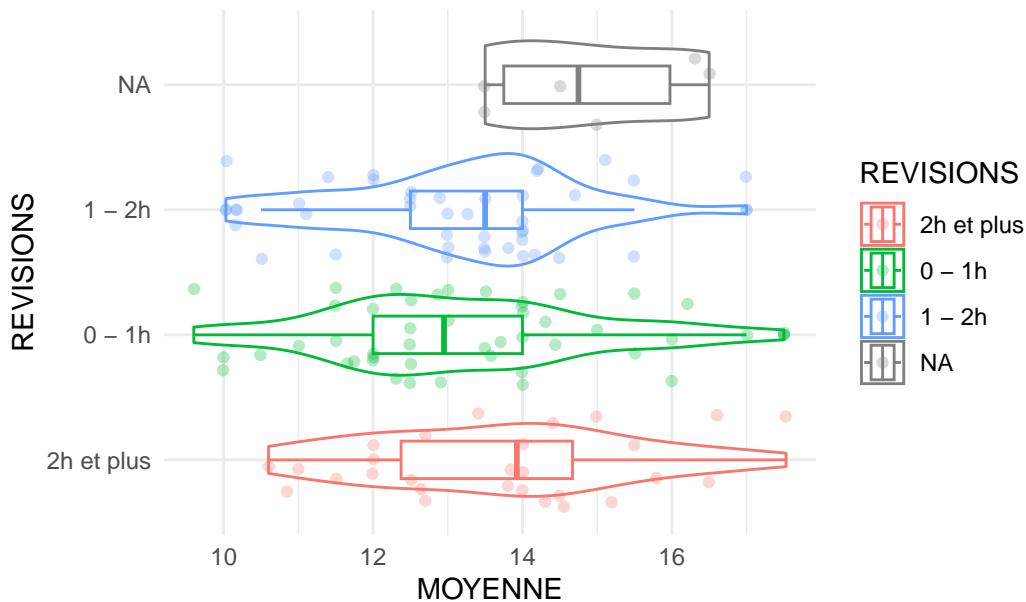


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de REVISIONS

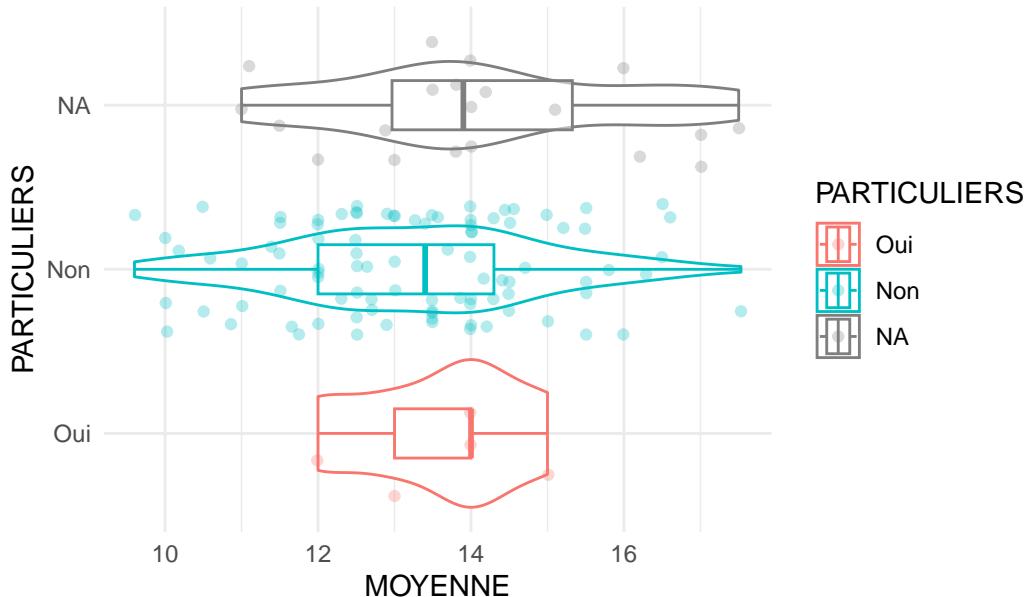


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de PARTICULARIERS

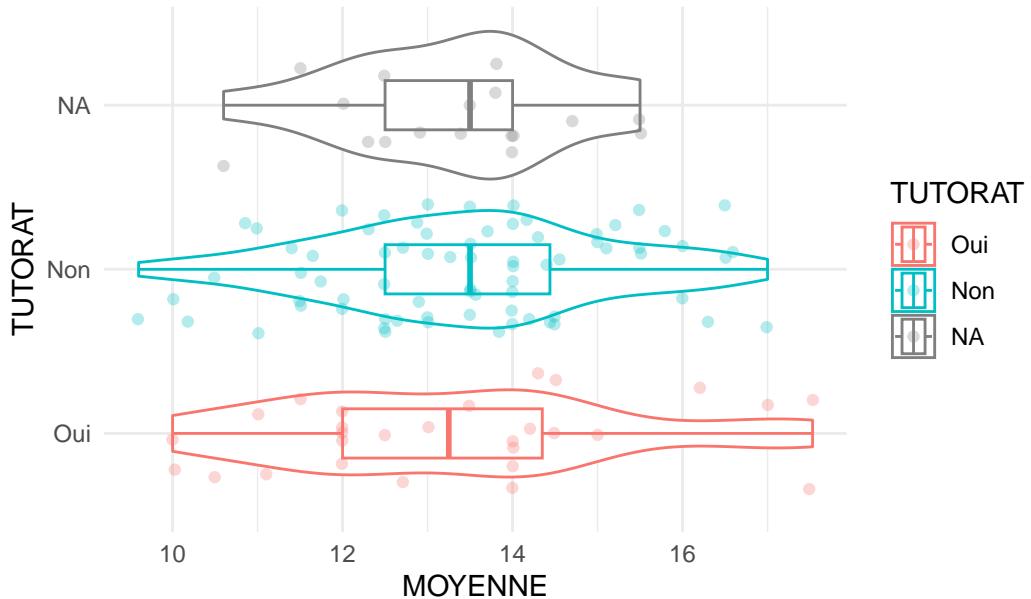


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de TUTORAT

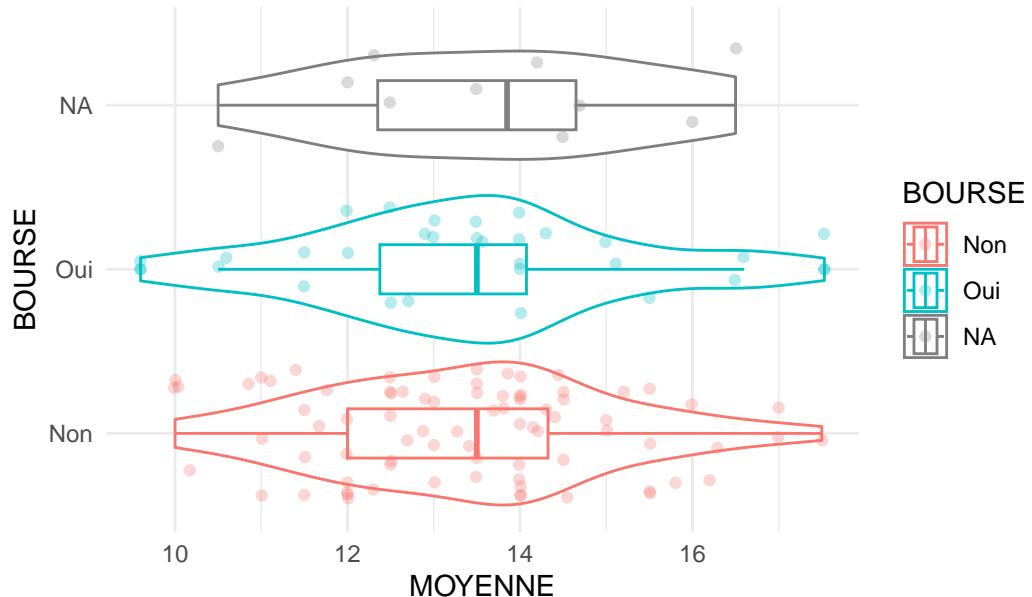


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de BOURSE

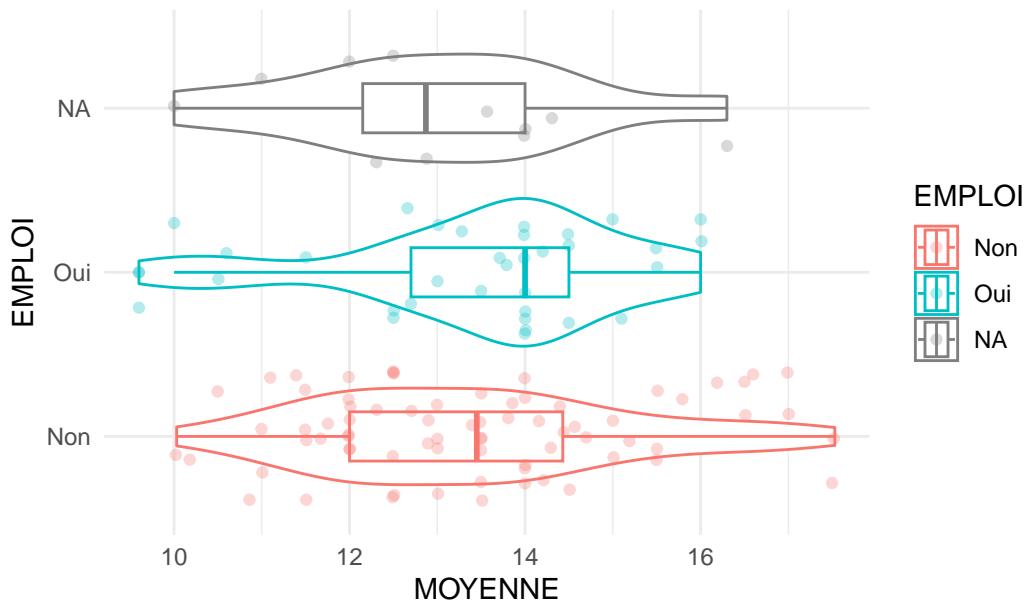


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de EMPLOI

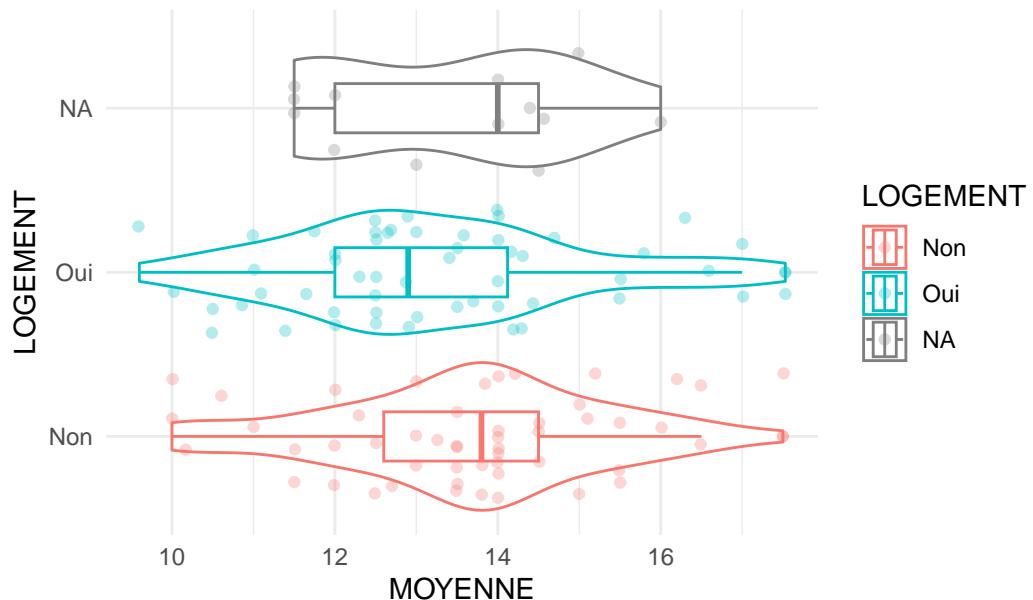


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de LOGEMENT

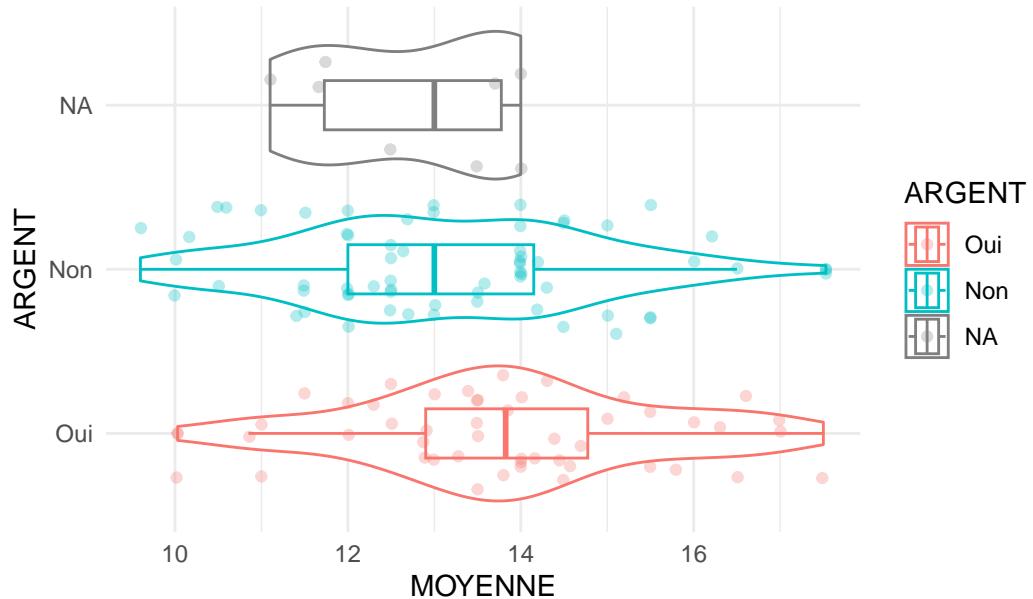


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de ARGENT

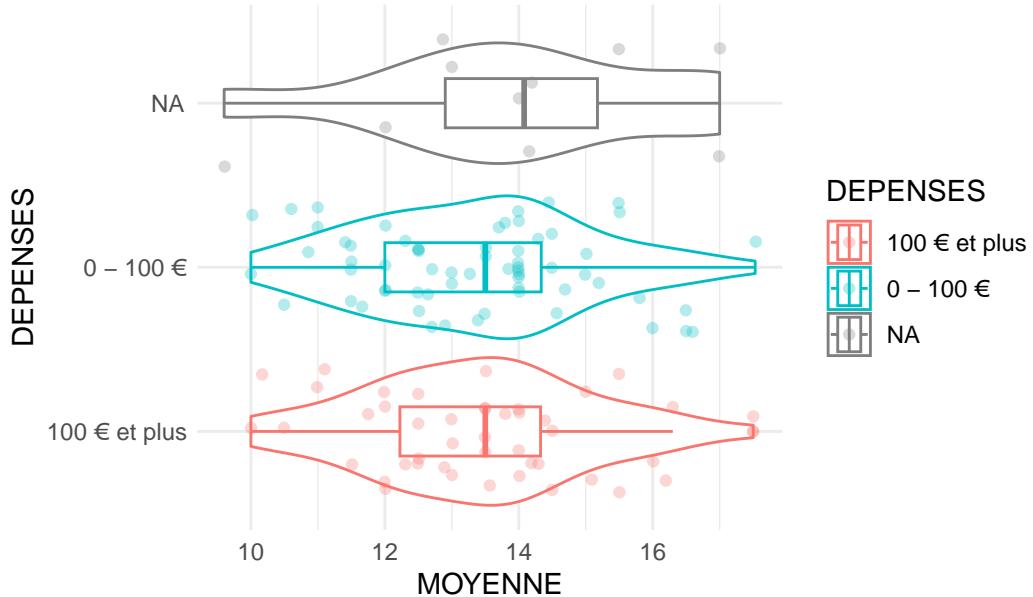


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de DEPENSES

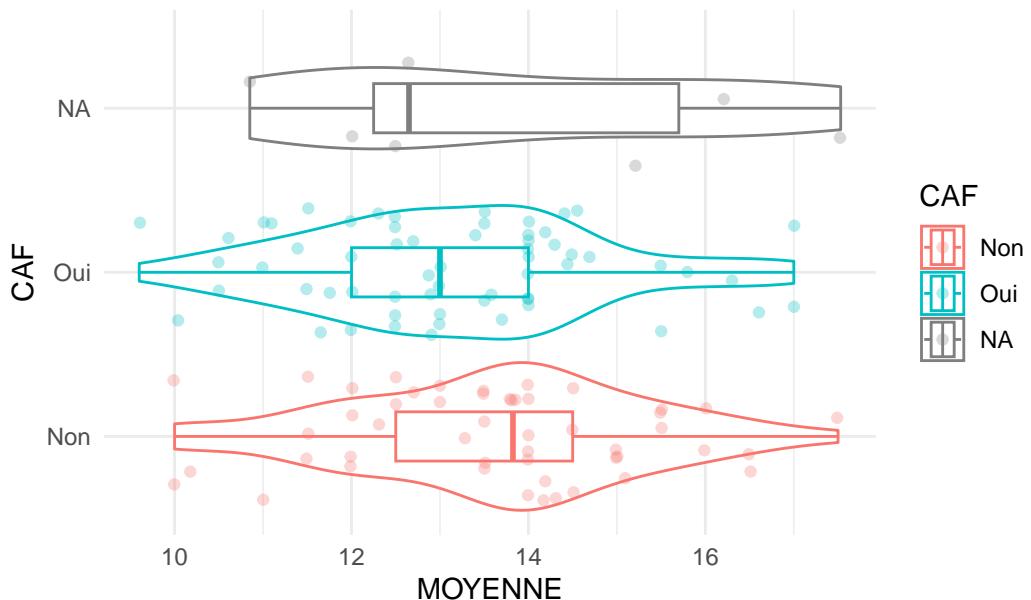


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de CAF

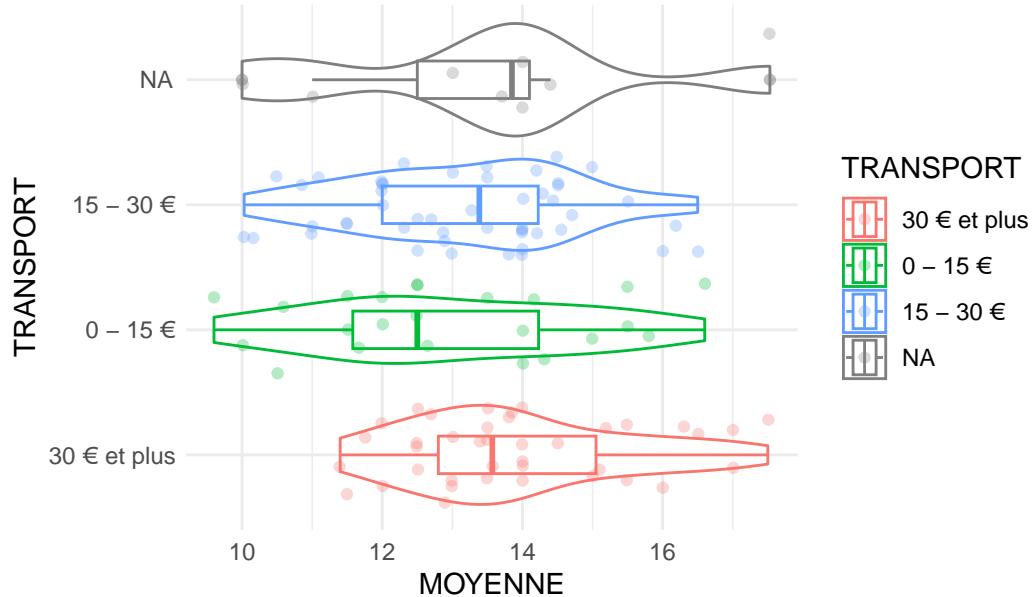


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de TRANSPORT

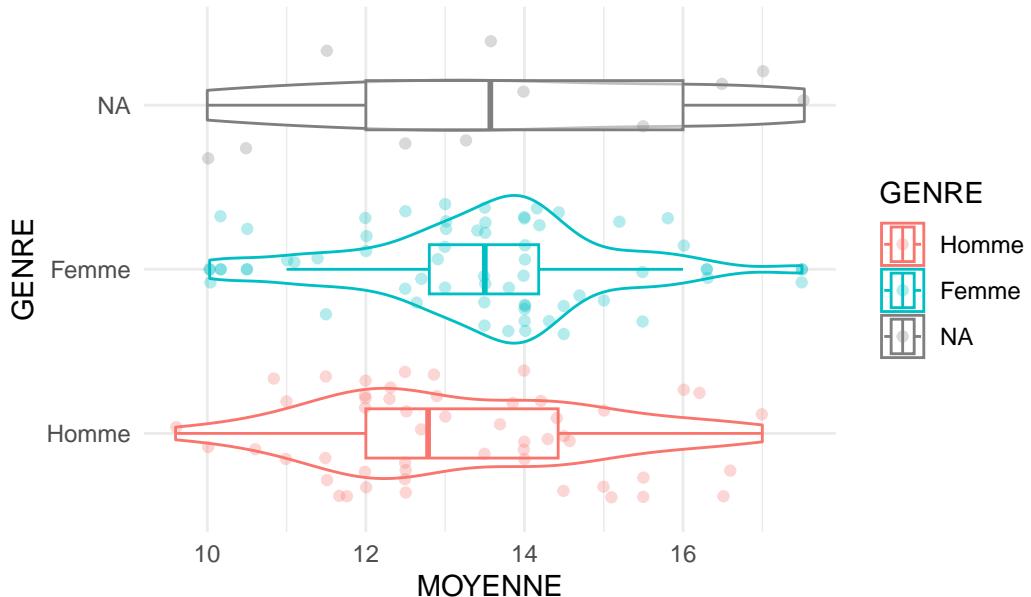


```
Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).
```

```
Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```
Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Distribution de MOYENNE en fonction de GENRE

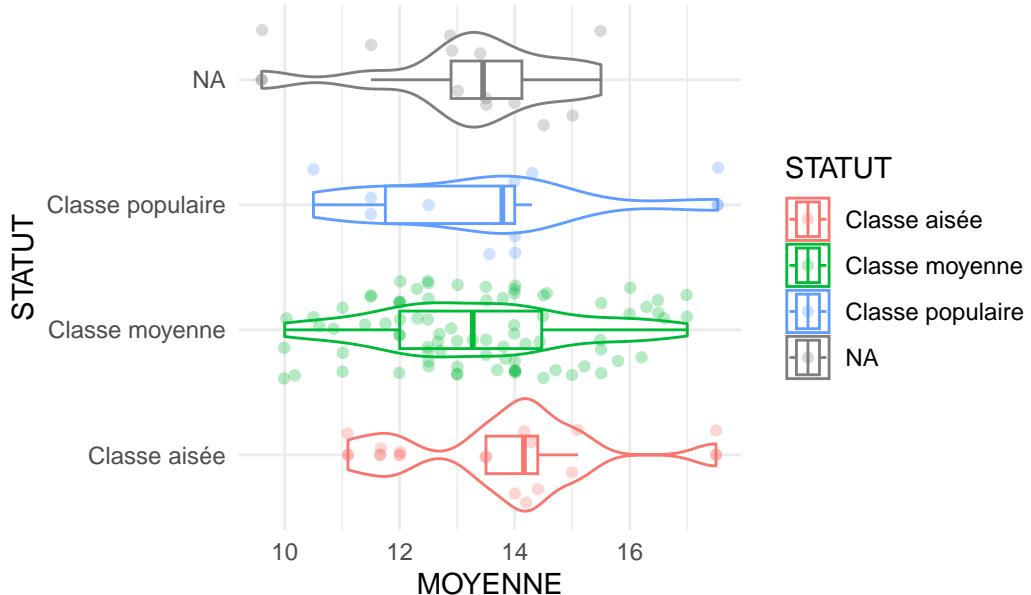


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de STATUT

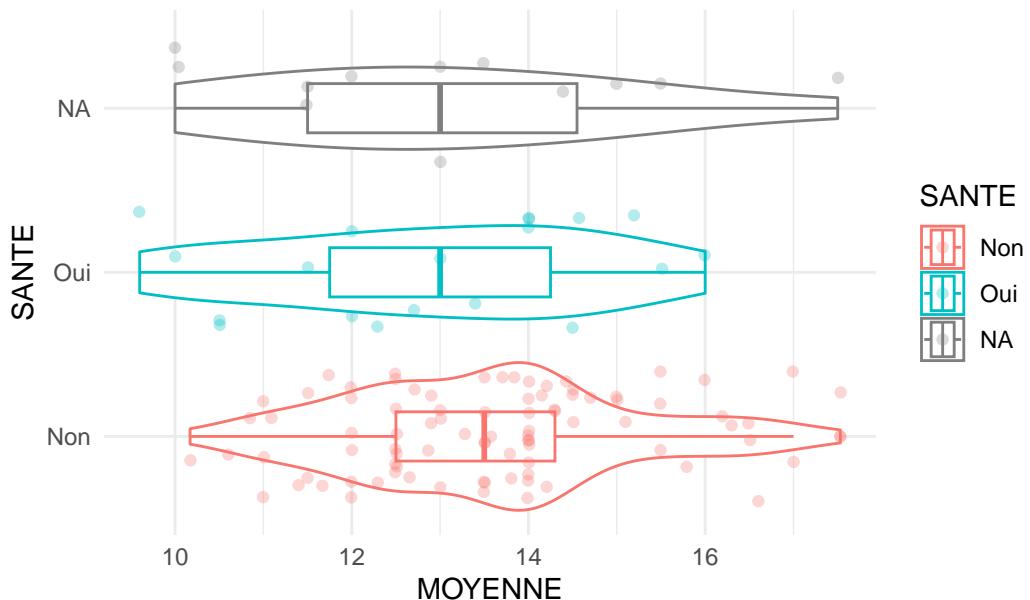


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de SANTE

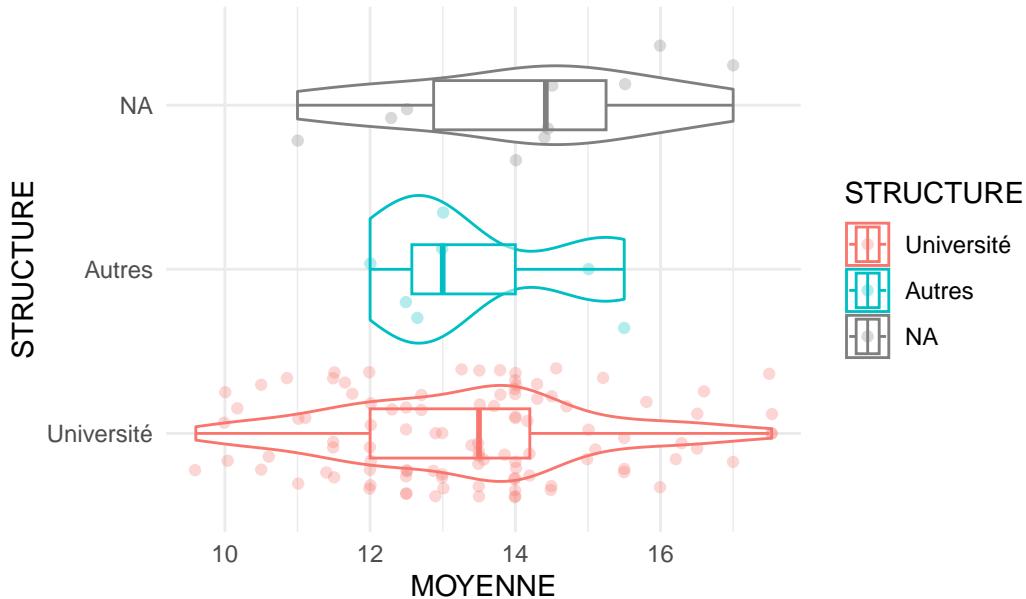


Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 17 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de MOYENNE en fonction de STRUCTURE

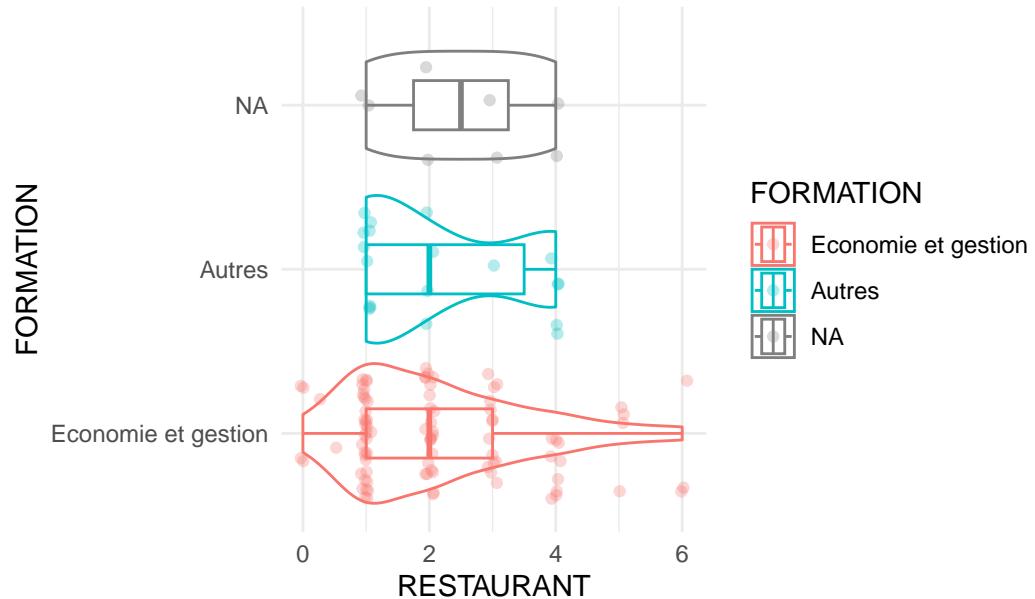


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de FORI

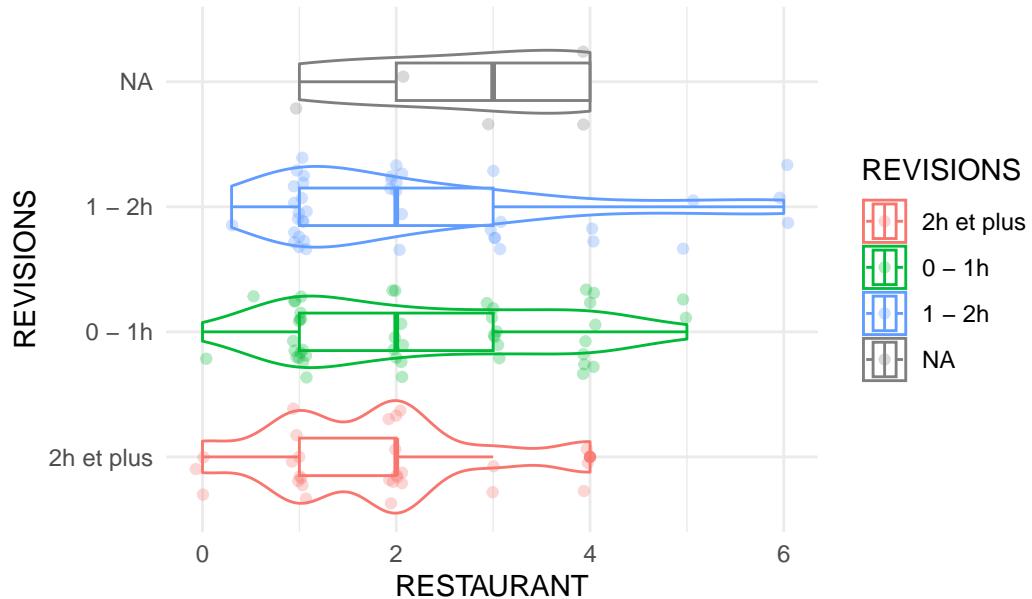


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de REVISIONS

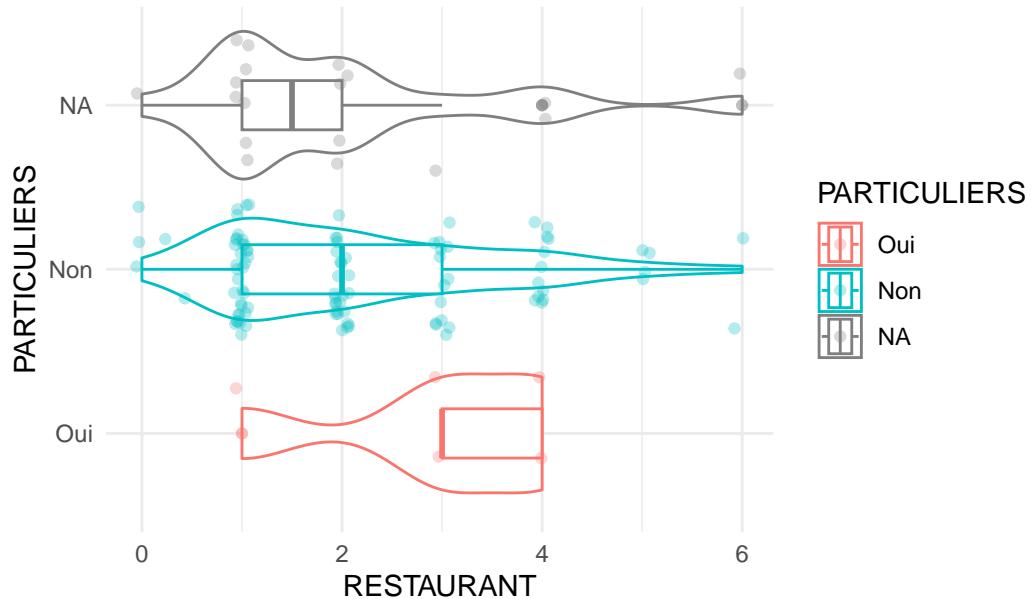


```
Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).
```

```
Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```
Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Distribution de RESTAURANT en fonction de PARTICULIERS

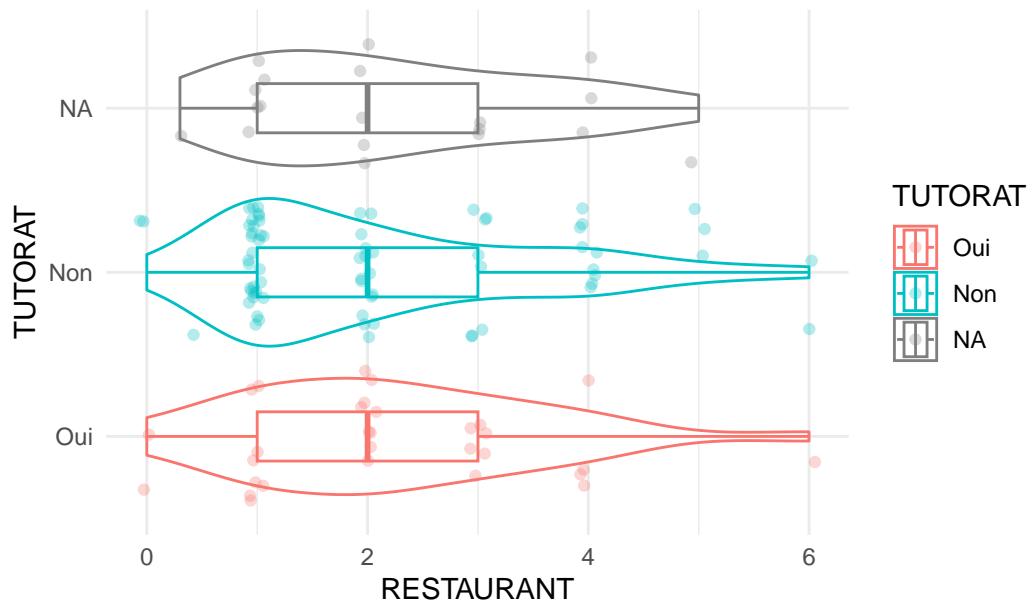


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de TUTORAT

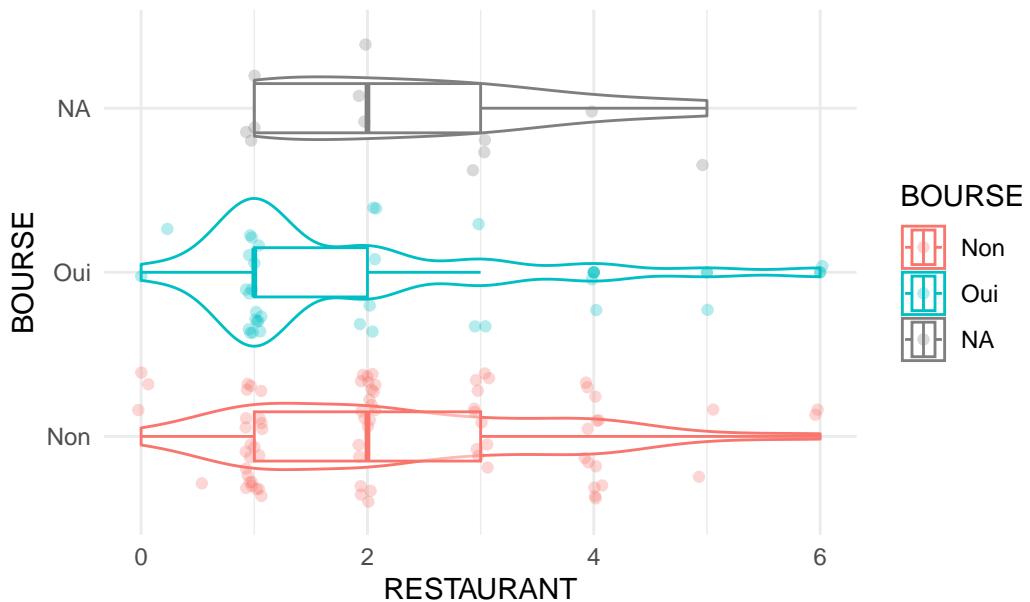


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de BOURSE

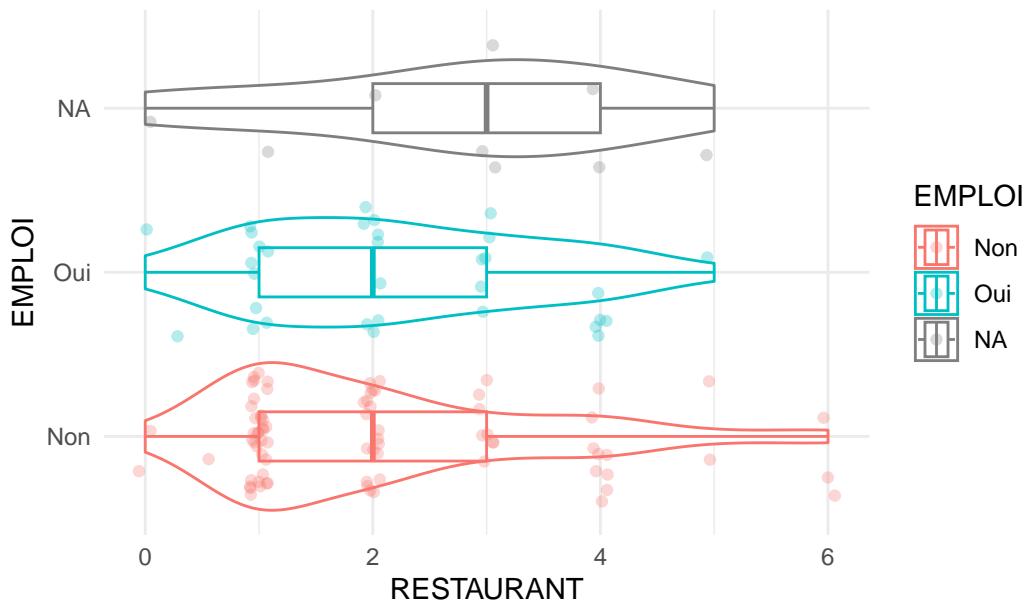


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de EMPLOI

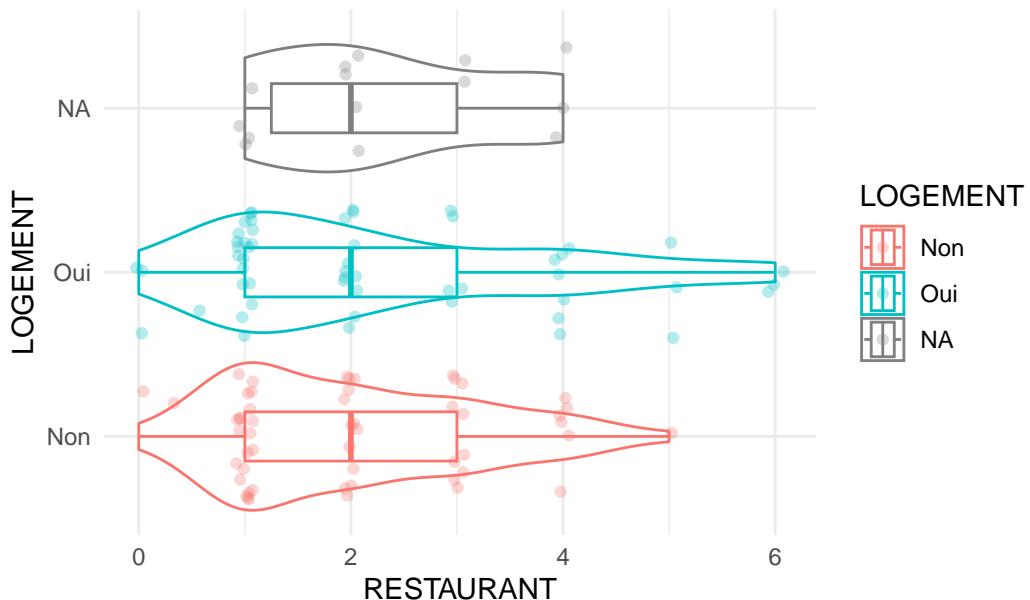


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de LOGEMENT

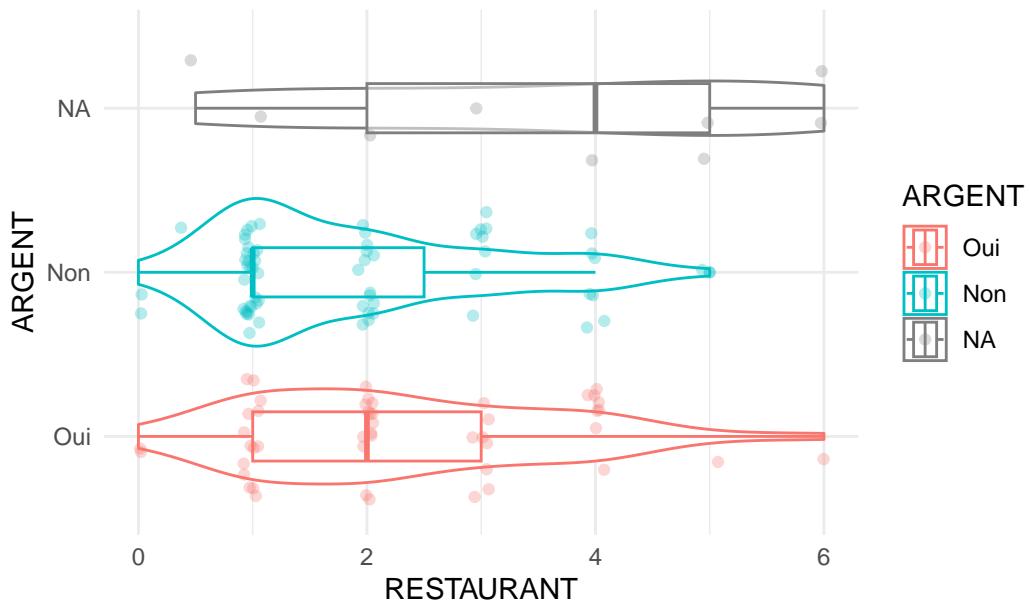


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de ARGENT

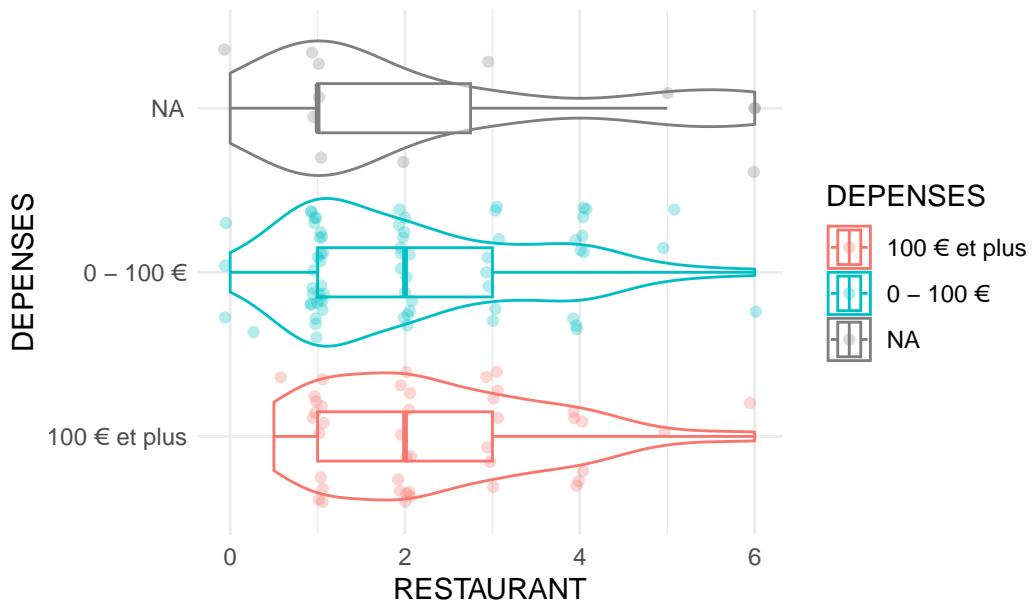


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de DEPENSE

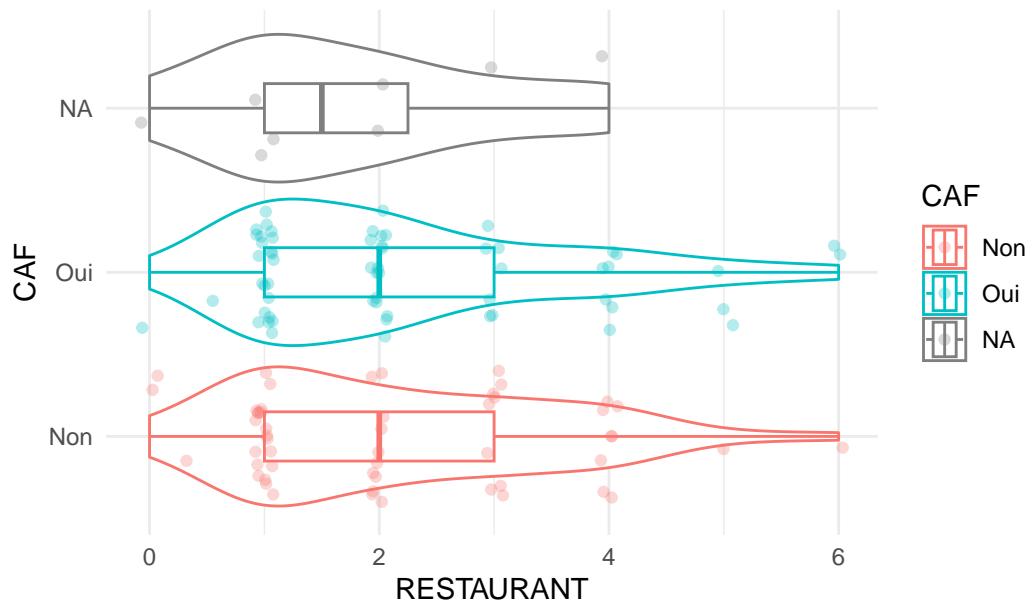


```
Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).
```

```
Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```
Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Distribution de RESTAURANT en fonction de CAF

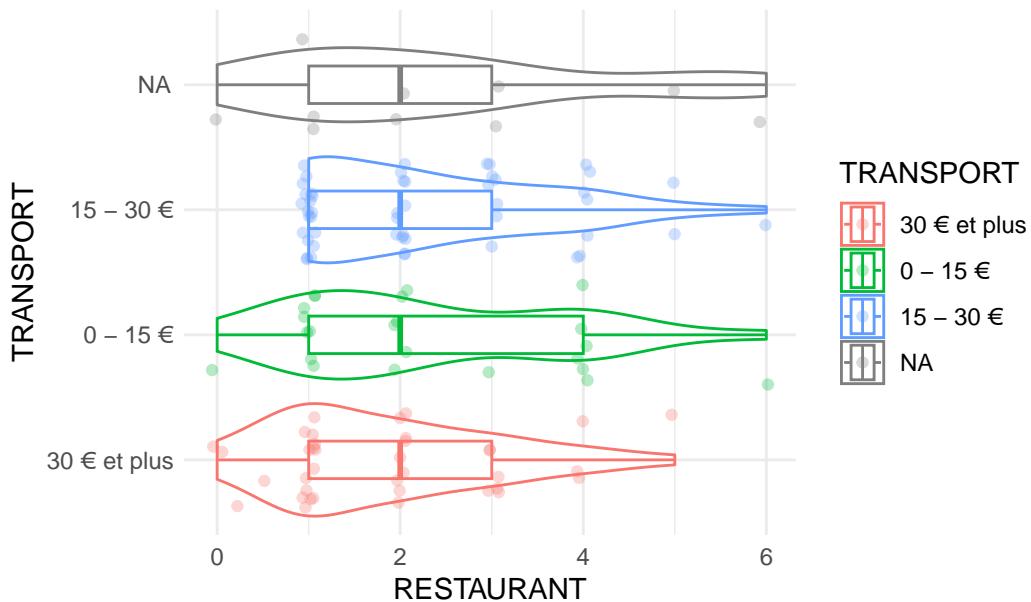


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de TRANSPORT

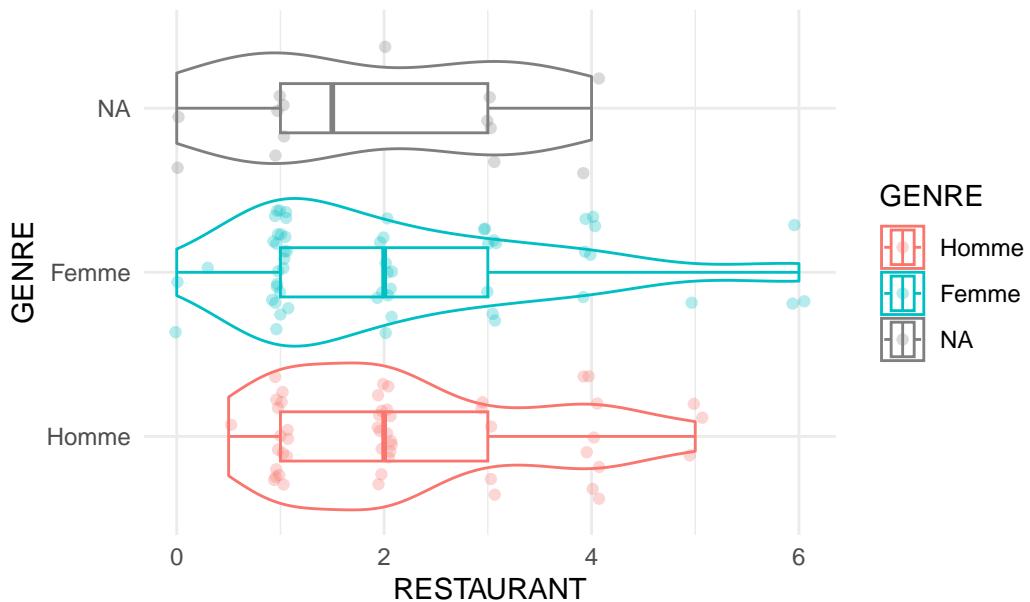


```
Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).
```

```
Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```
Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Distribution de RESTAURANT en fonction de GENRE

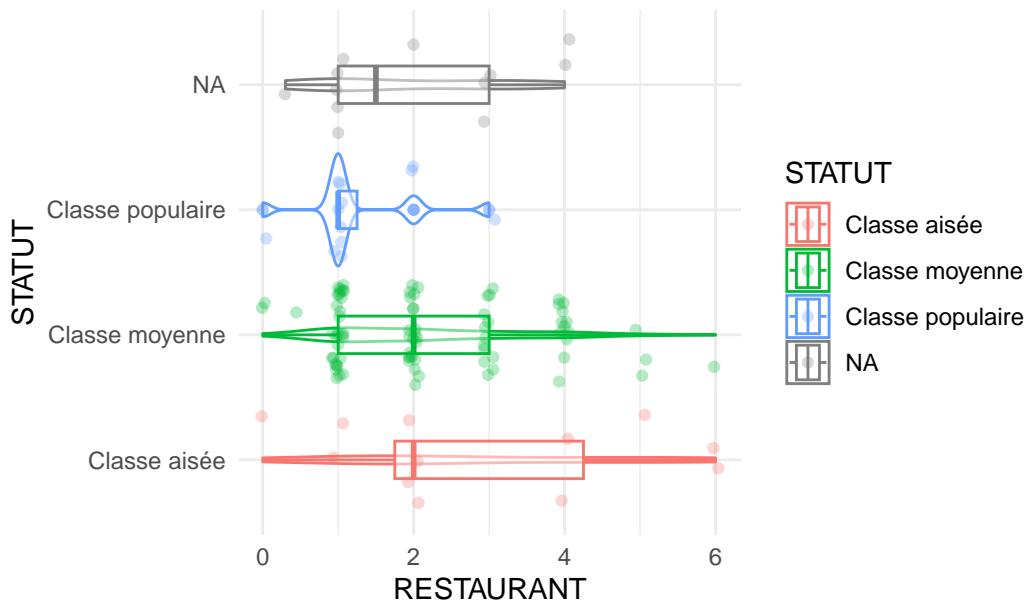


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de STATUT

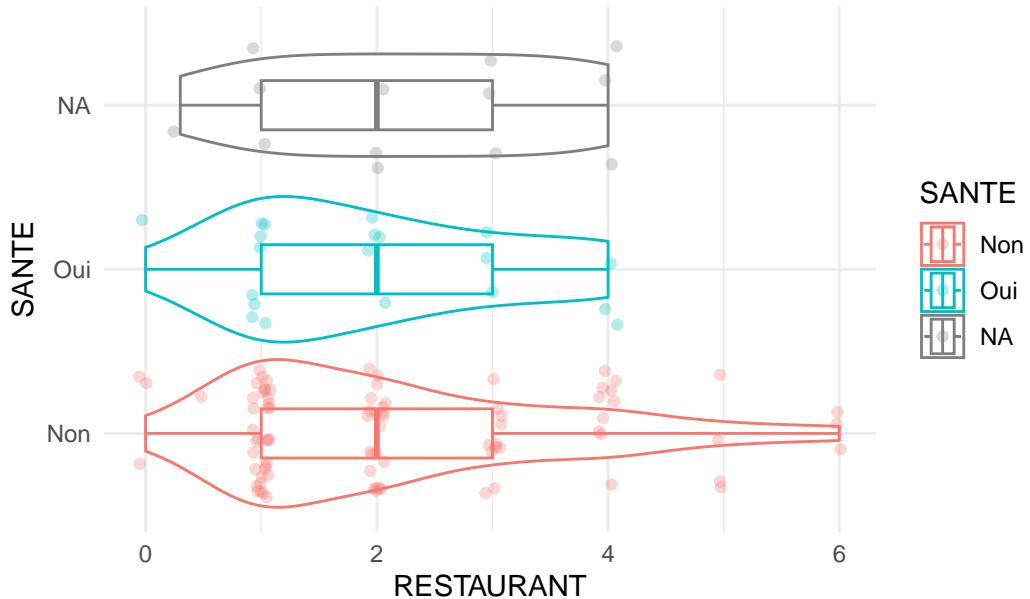


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de SANTE

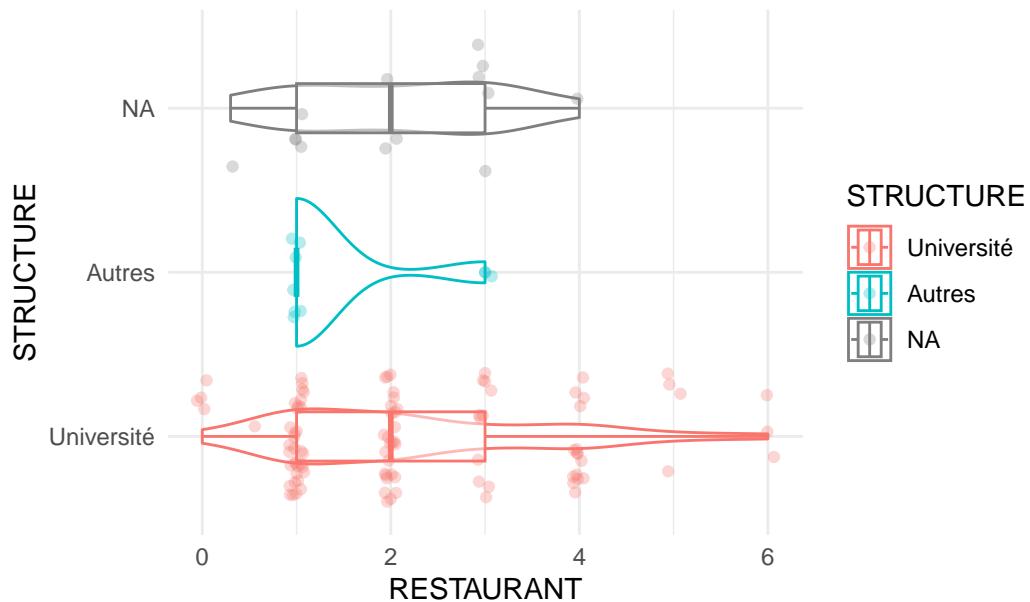


Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 15 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 15 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de RESTAURANT en fonction de STRUCTUR

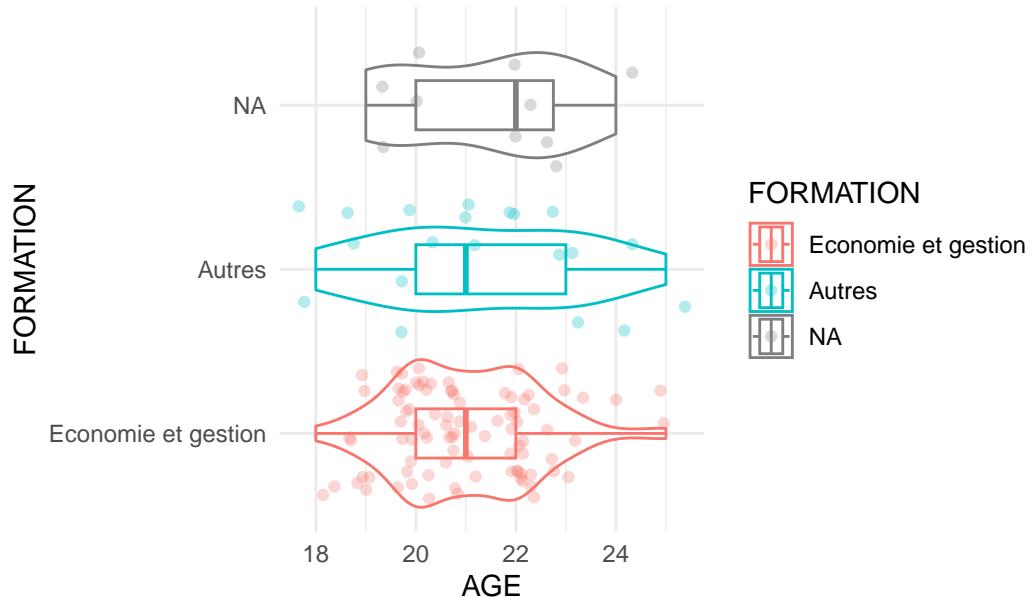


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de FORMATION

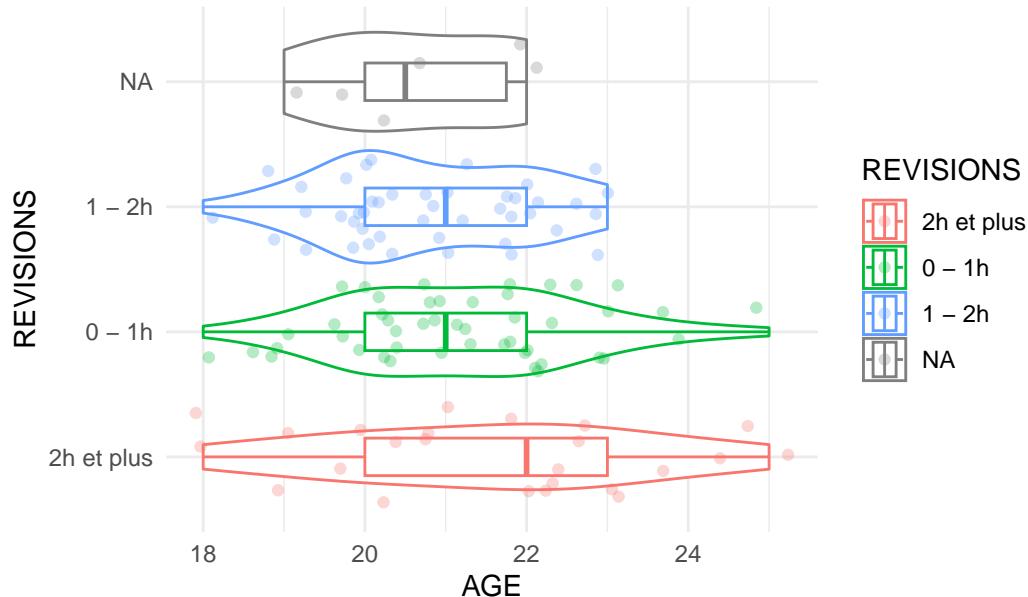


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de REVISIONS

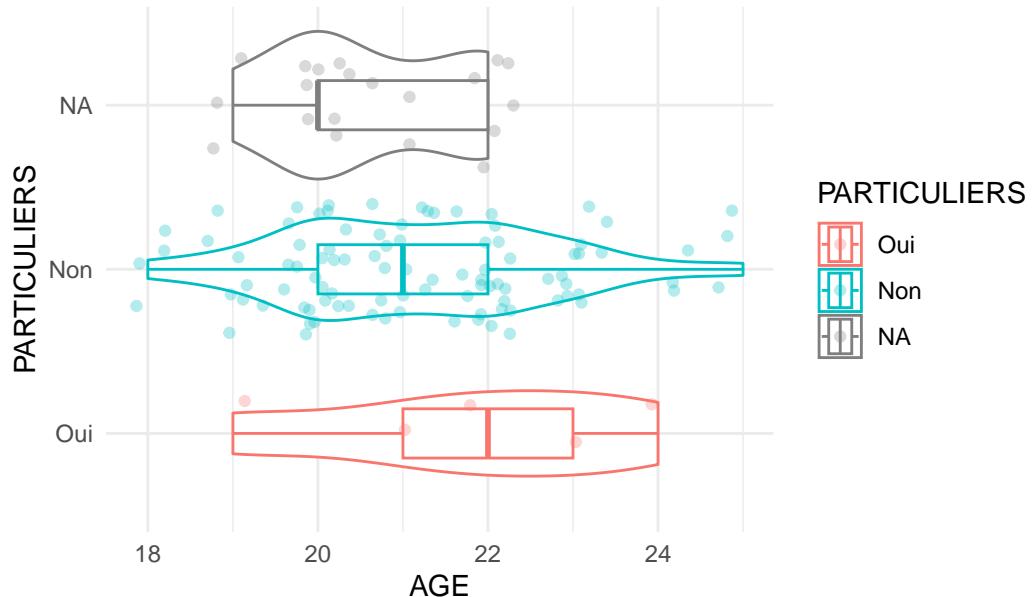


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de PARTICULARIERS

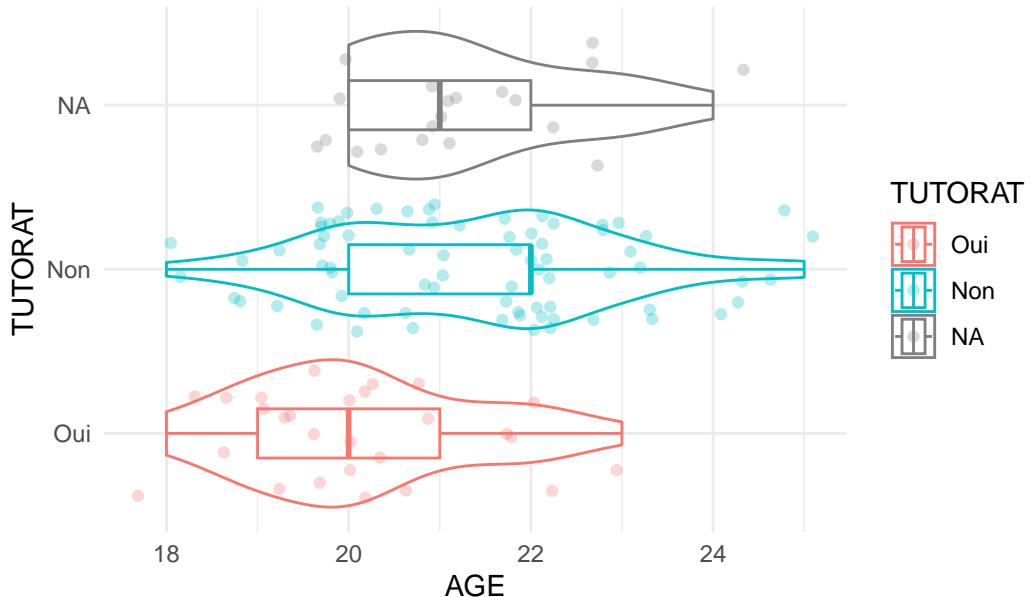


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de TUTORAT

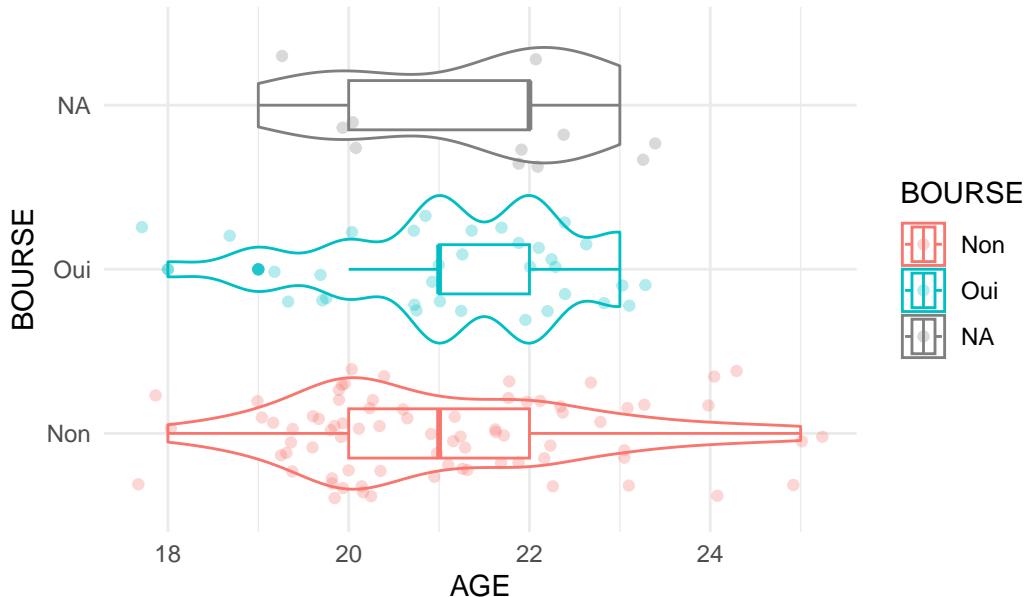


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de BOURSE

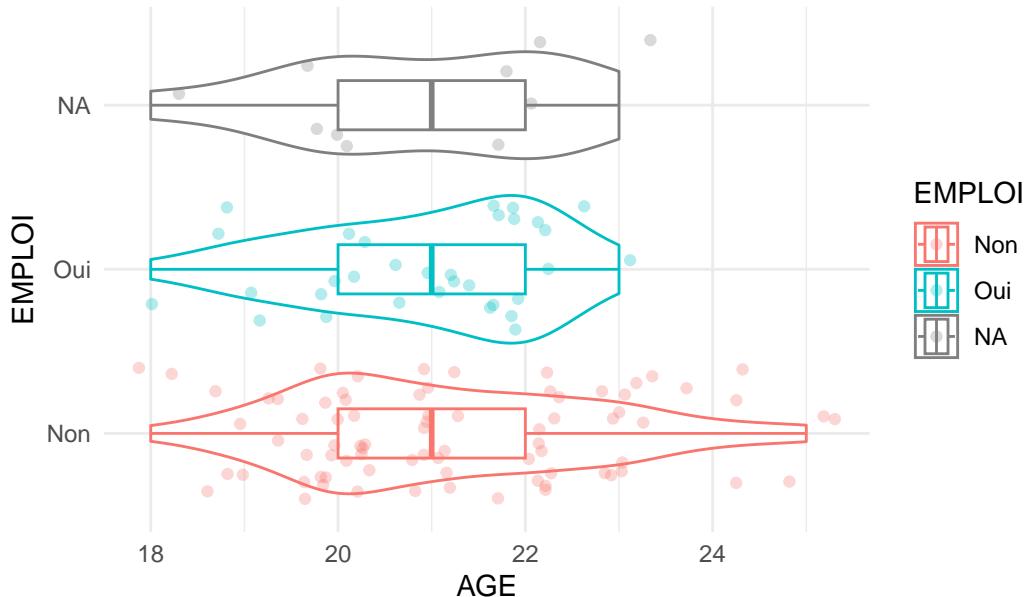


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de EMPLOI

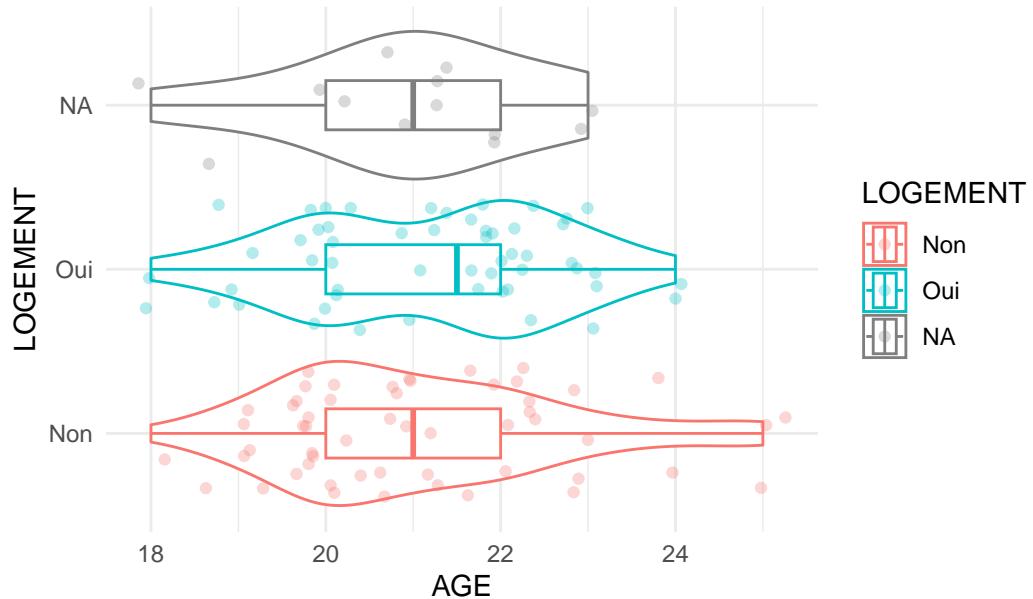


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de LOGEMENT

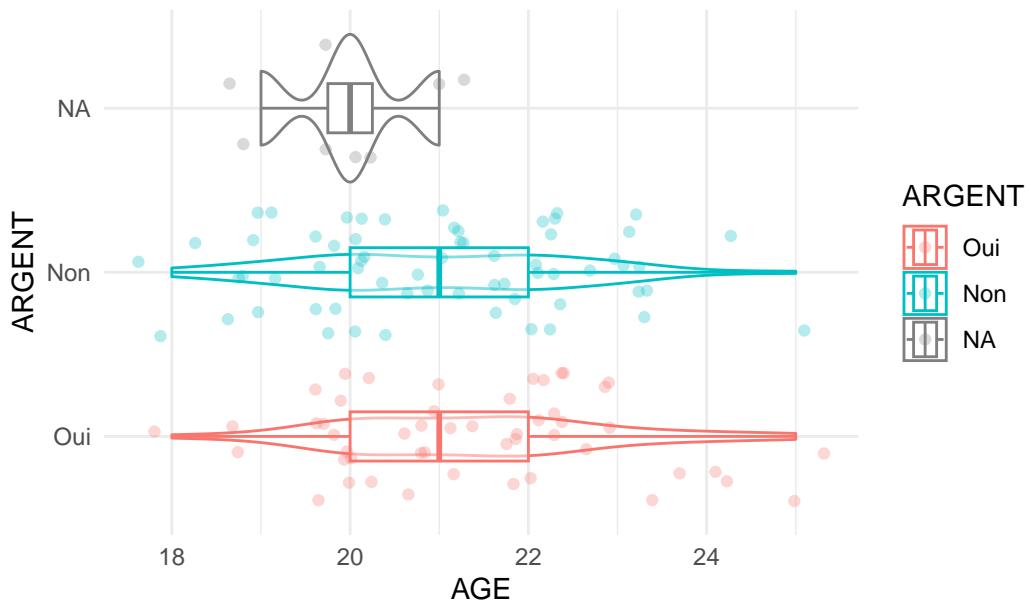


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de ARGENT

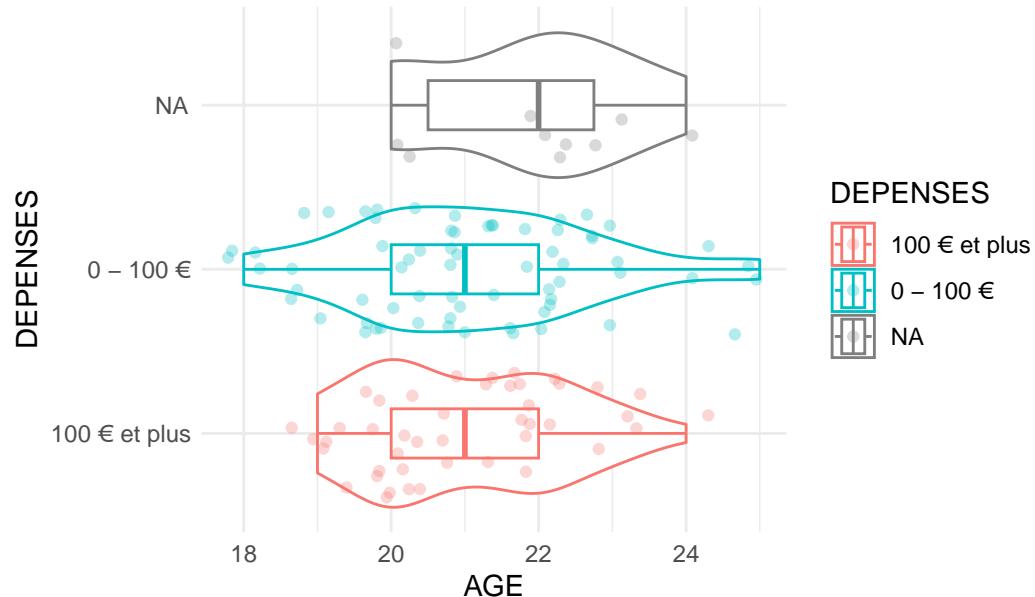


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de DEPENSES

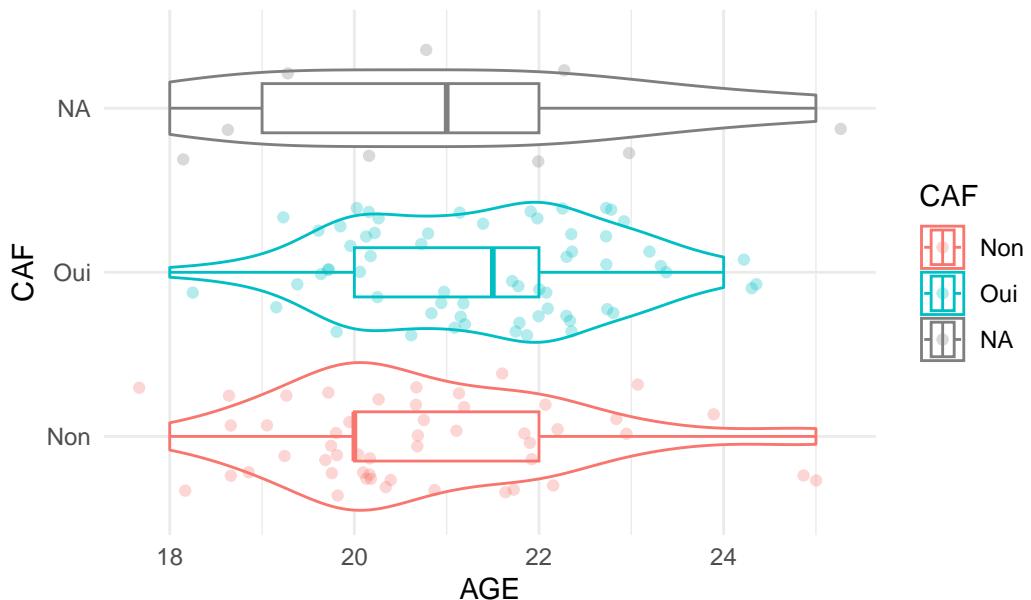


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de CAF

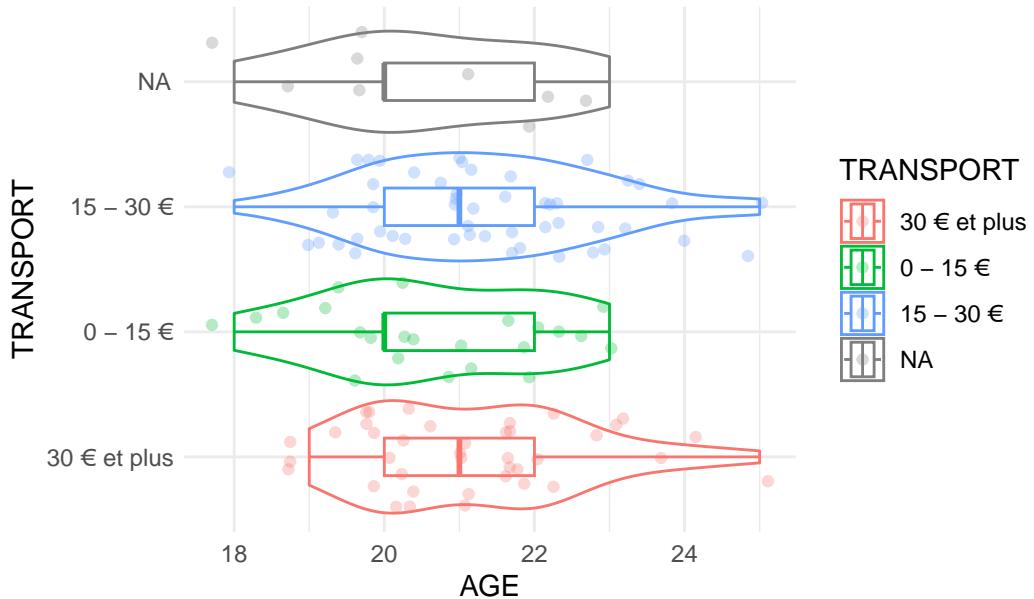


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de TRANSPORT

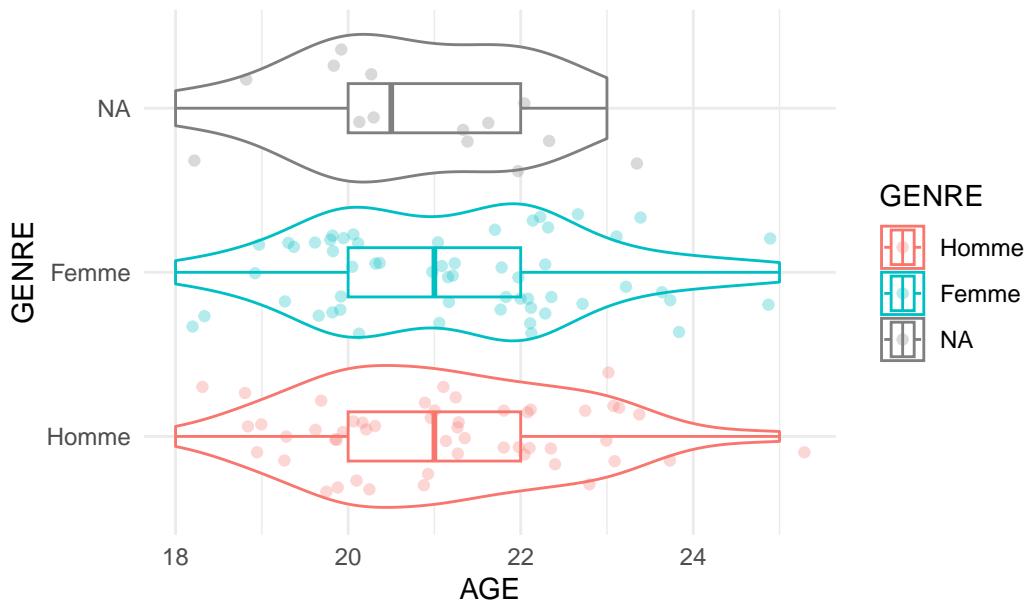


```
Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).
```

```
Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```
Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Distribution de AGE en fonction de GENRE

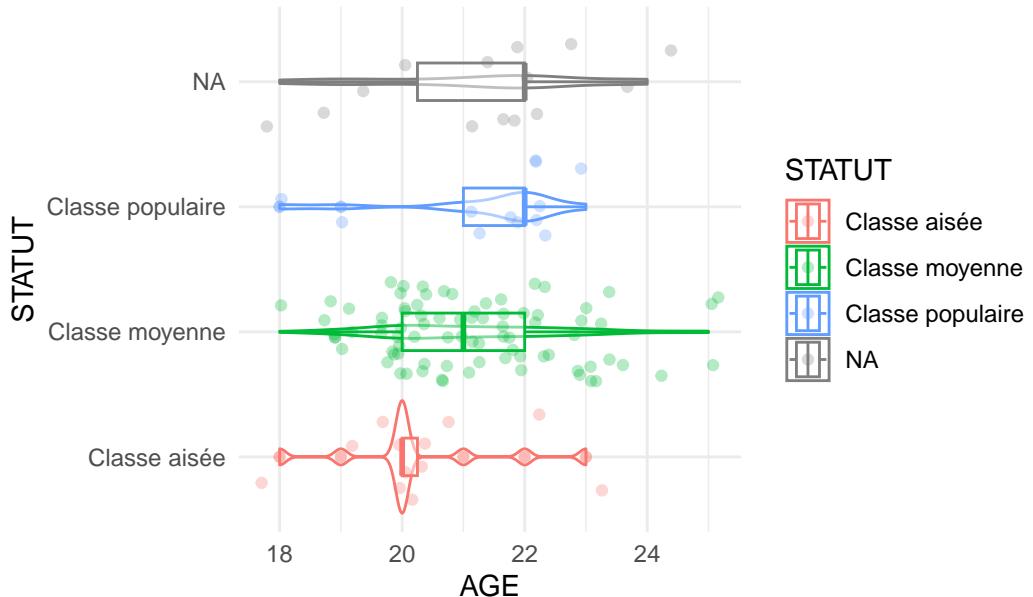


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de STATUT

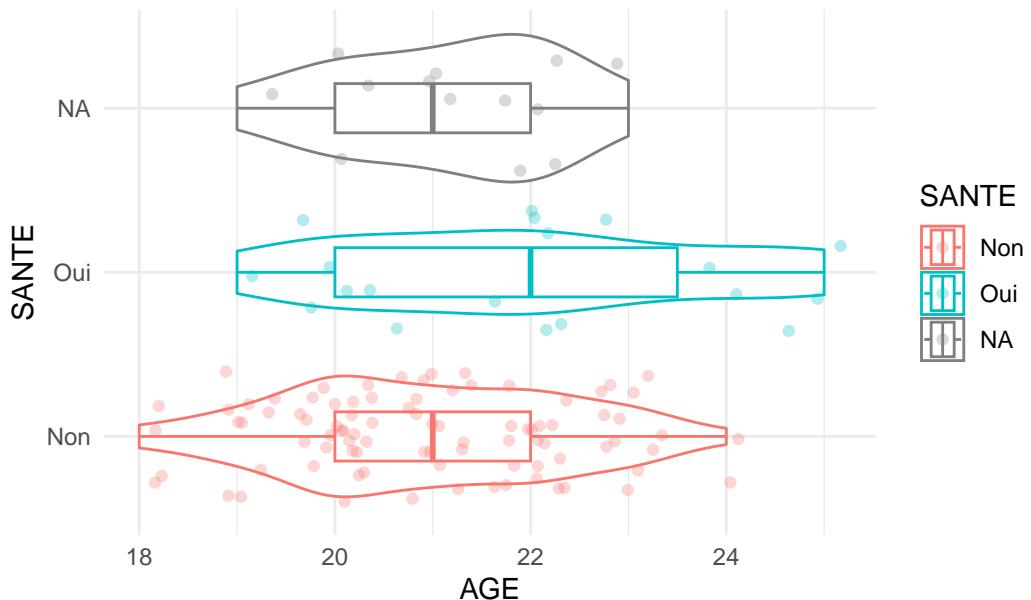


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de SANTE

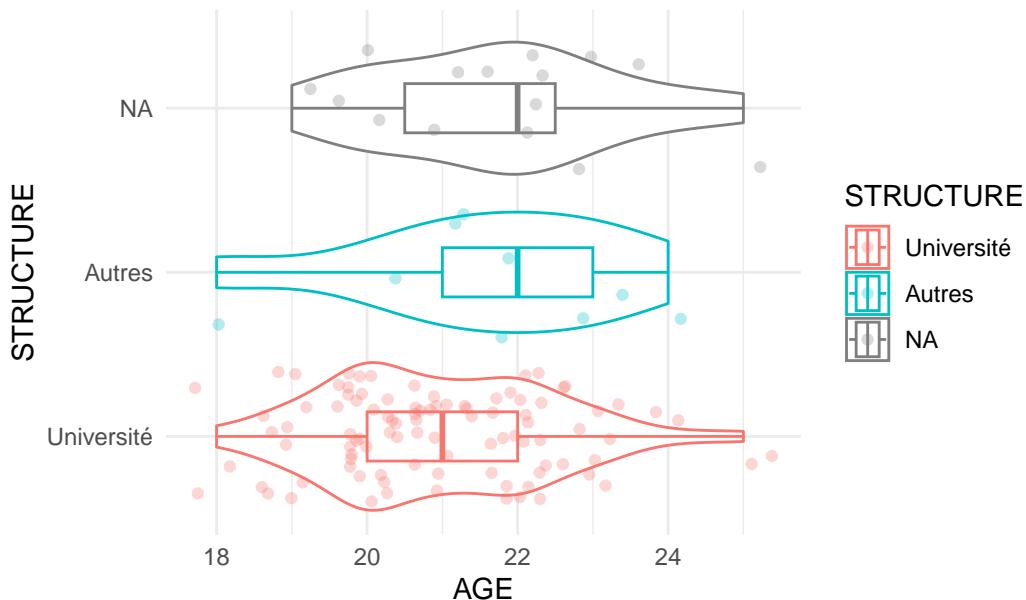


Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de AGE en fonction de STRUCTURE



```
# Visualisation des relations des trois dernières variables quantitatives avec chaque

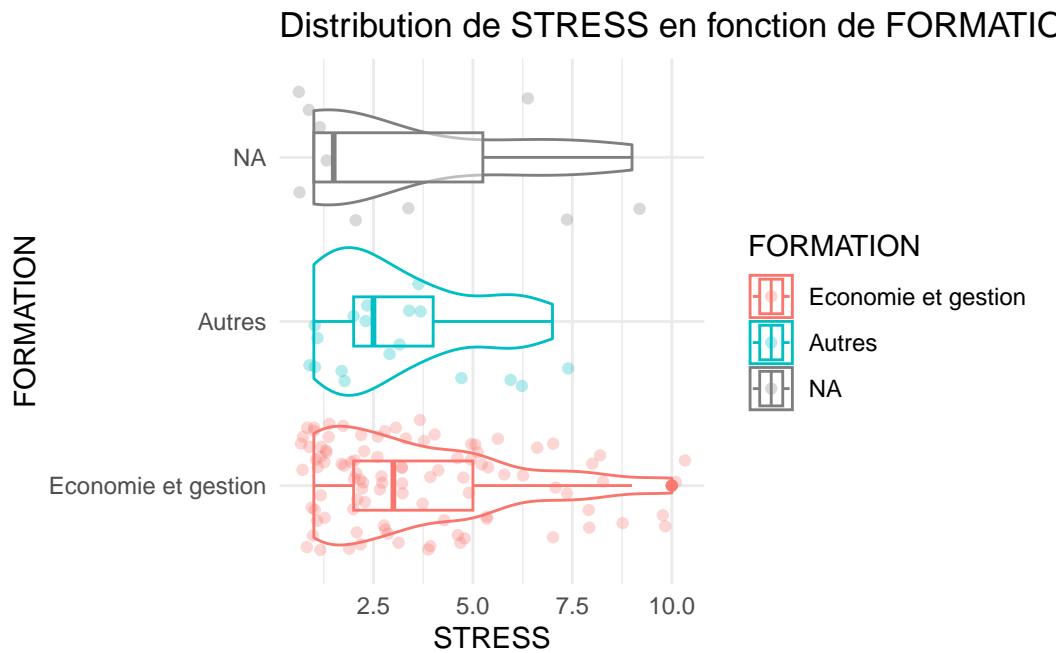
for (qt in quantis[5:7]) {
  for (ql in qualis) {
    p <- Budget2 |>
      ggplot() +
      aes_string(x = qt, y = ql, color = ql) +
      geom_violin() +
      geom_boxplot(width = 0.3, alpha = 0.5) +
      geom_jitter(alpha = 0.3) +
      theme_minimal() +
      labs(title = paste("Distribution de", qt, "en fonction de", ql))

    print(p)
  }
}
```

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range (`geom_point()`).

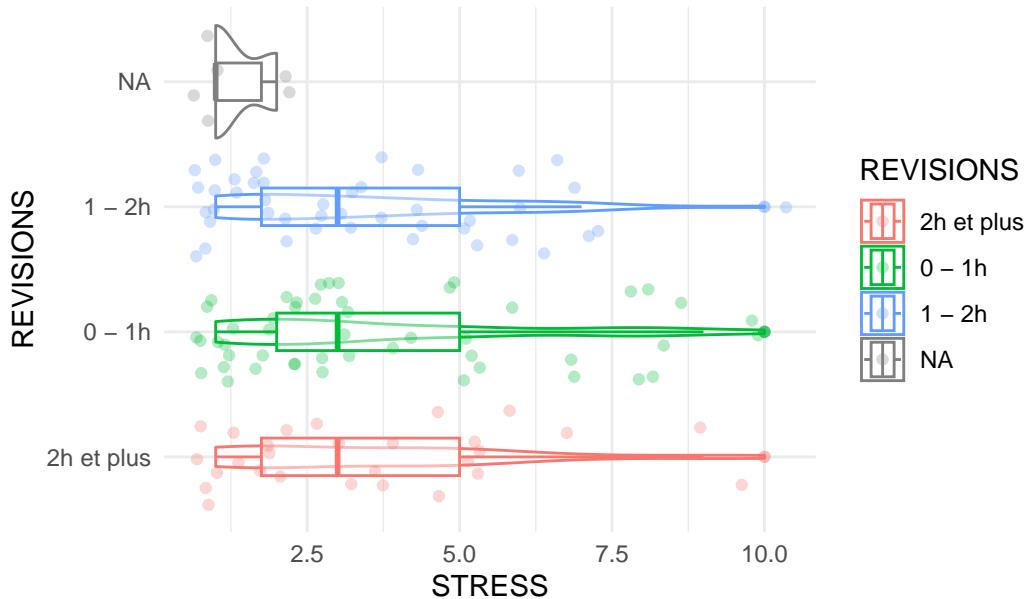


Warning: Removed 8 rows containing non-finite outside the scale range (`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range (`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range (`geom_point()`).

Distribution de STRESS en fonction de REVISIONS

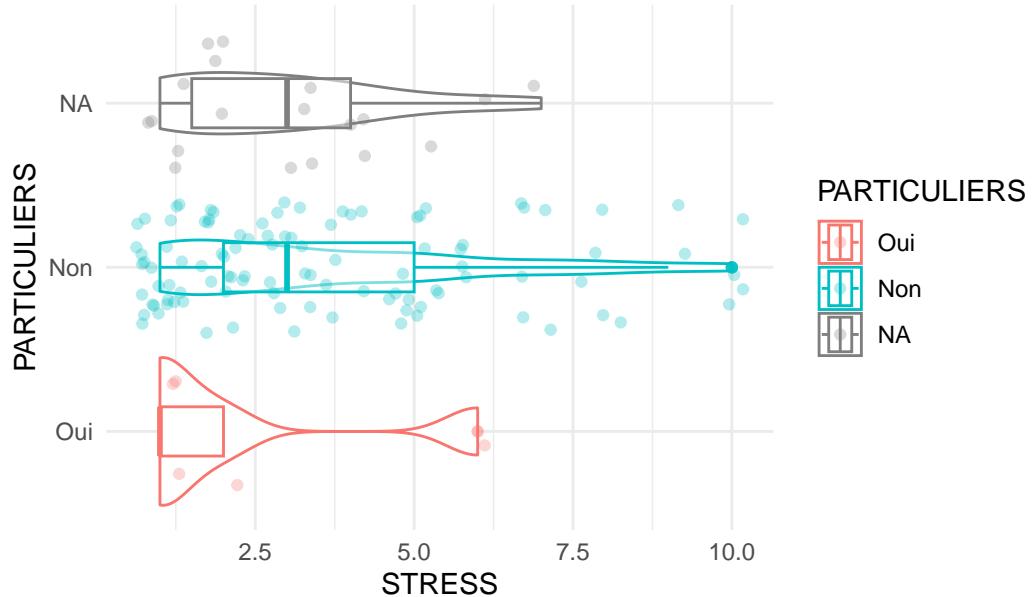


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de PARTICULARIERS

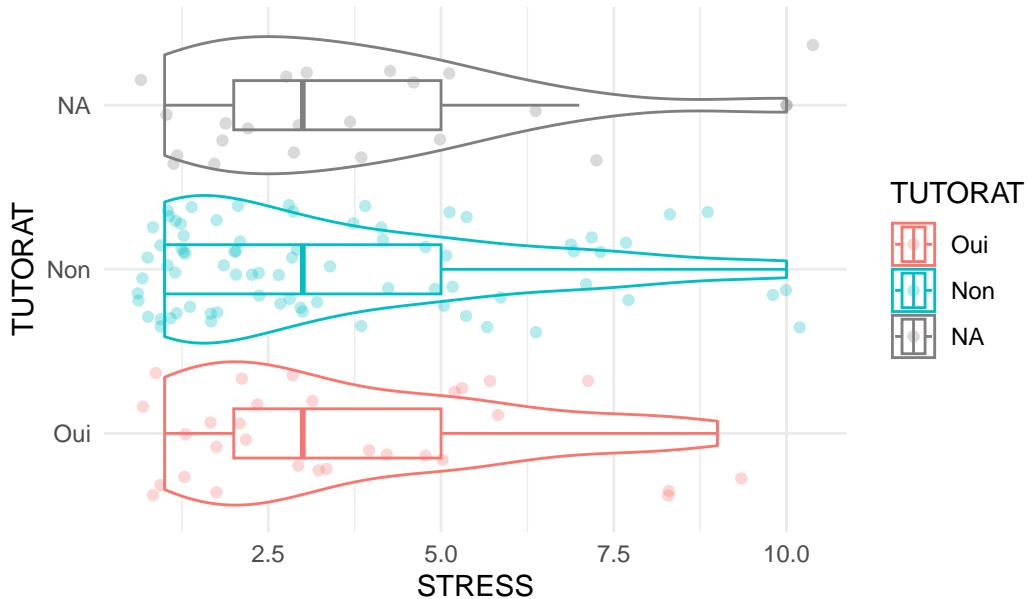


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de TUTORAT

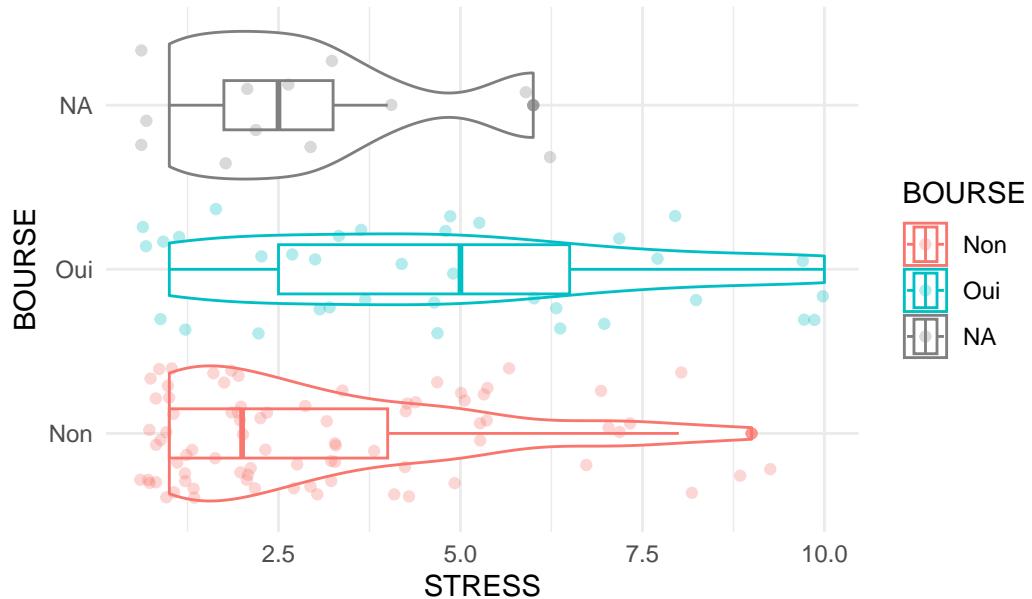


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de BOURSE

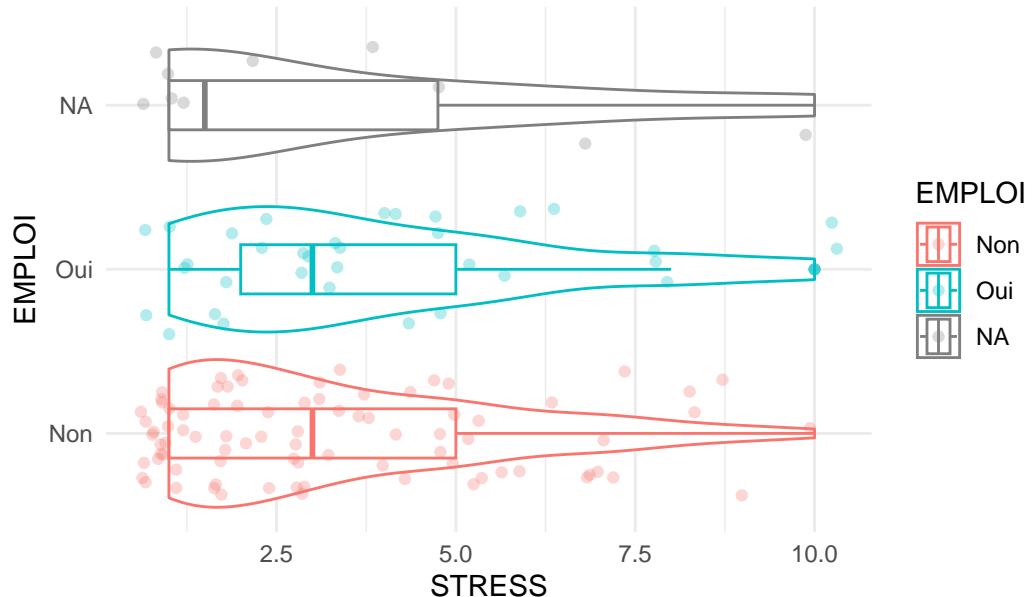


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de EMPLOI

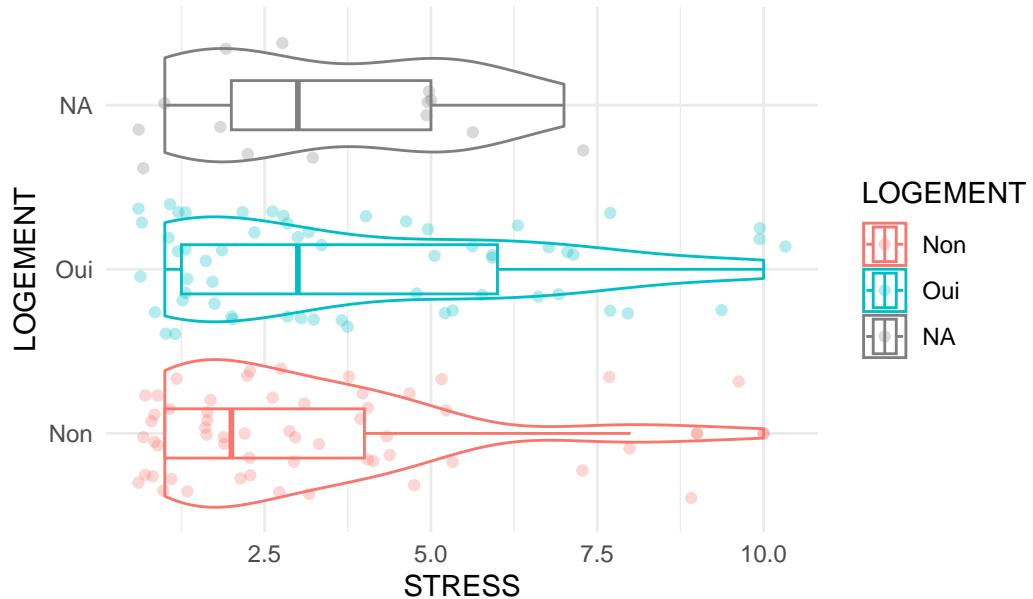


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de LOGEMENT

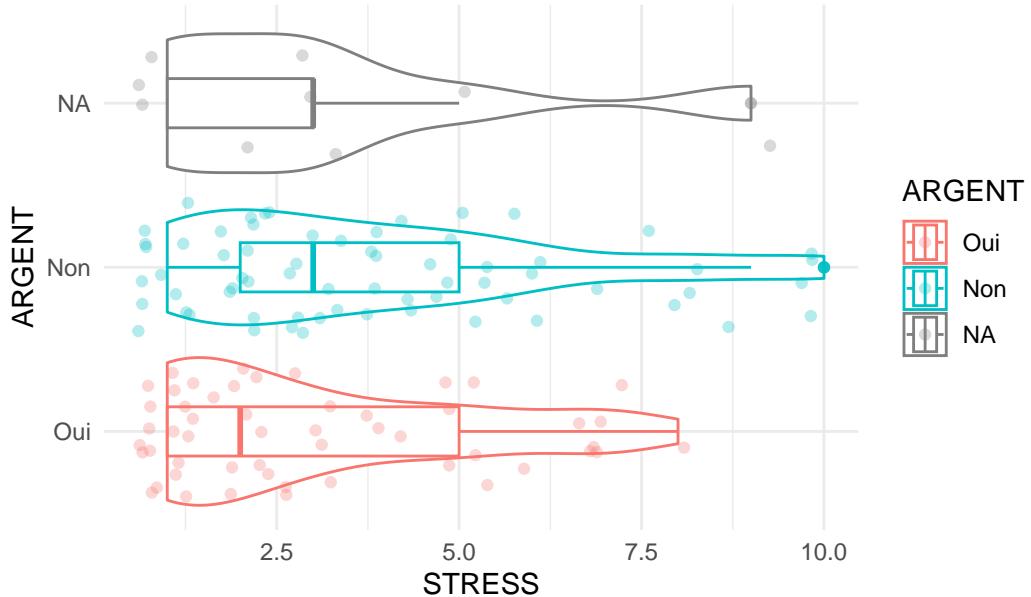


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de ARGENT

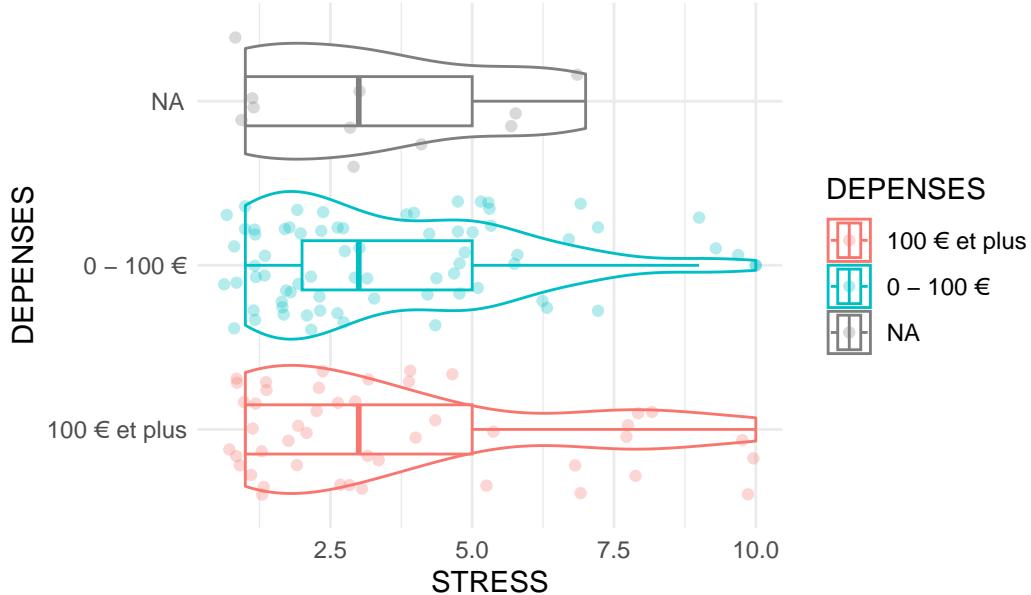


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de DEPENSES

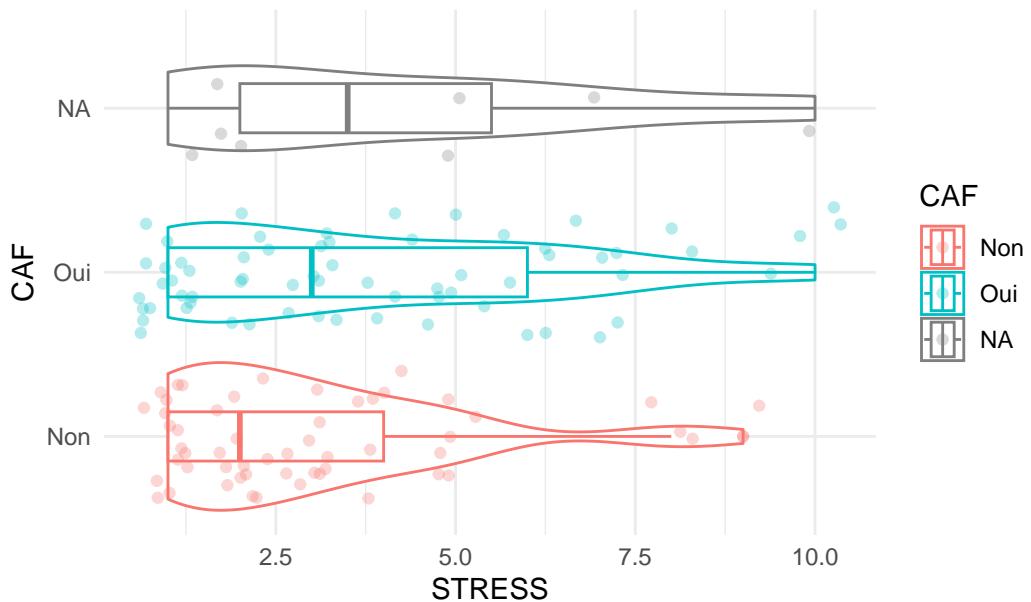


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de CAF

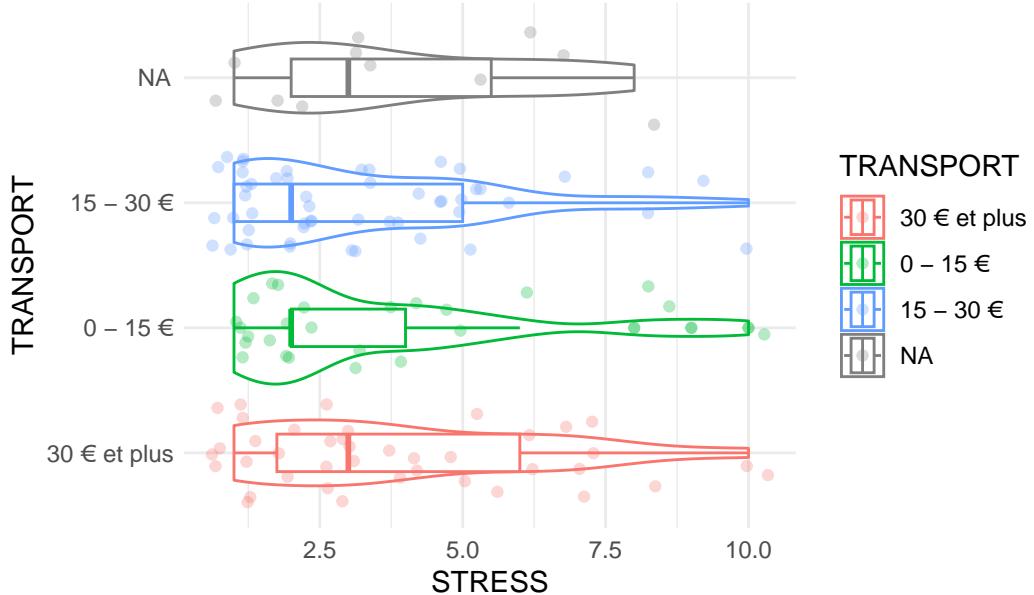


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de TRANSPORT

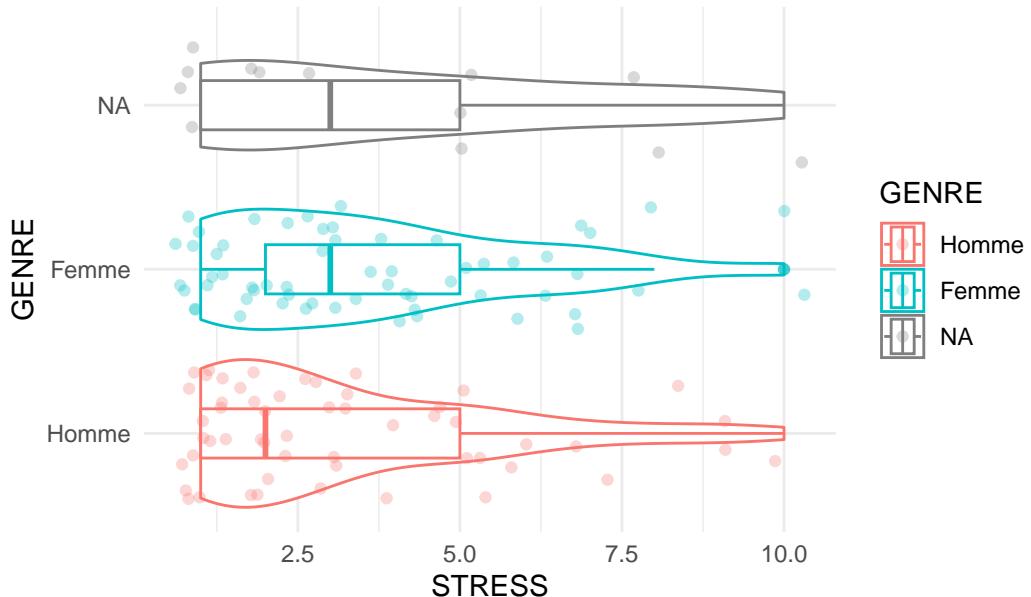


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de GENRE

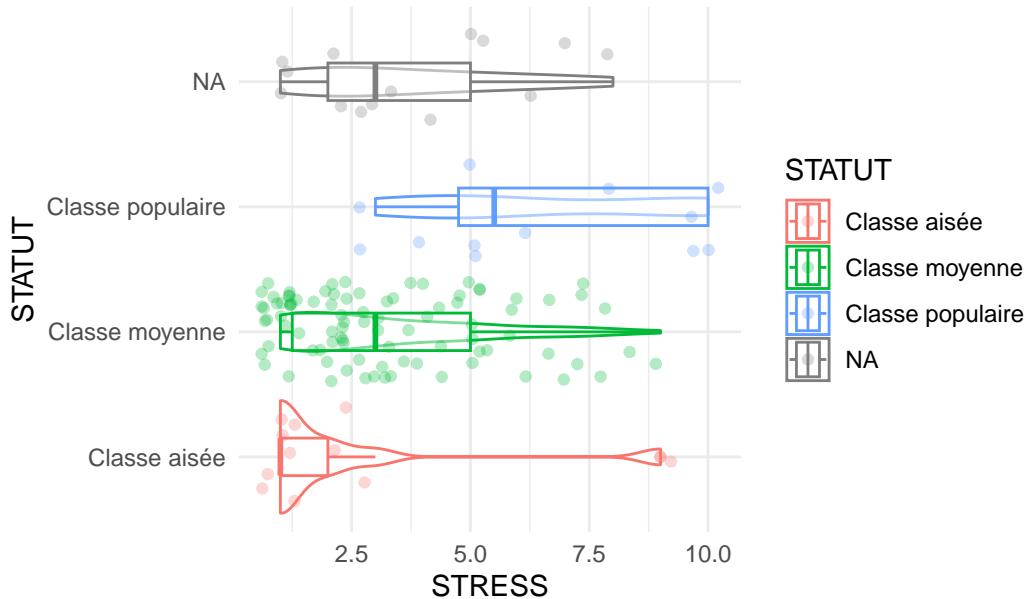


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de STATUT

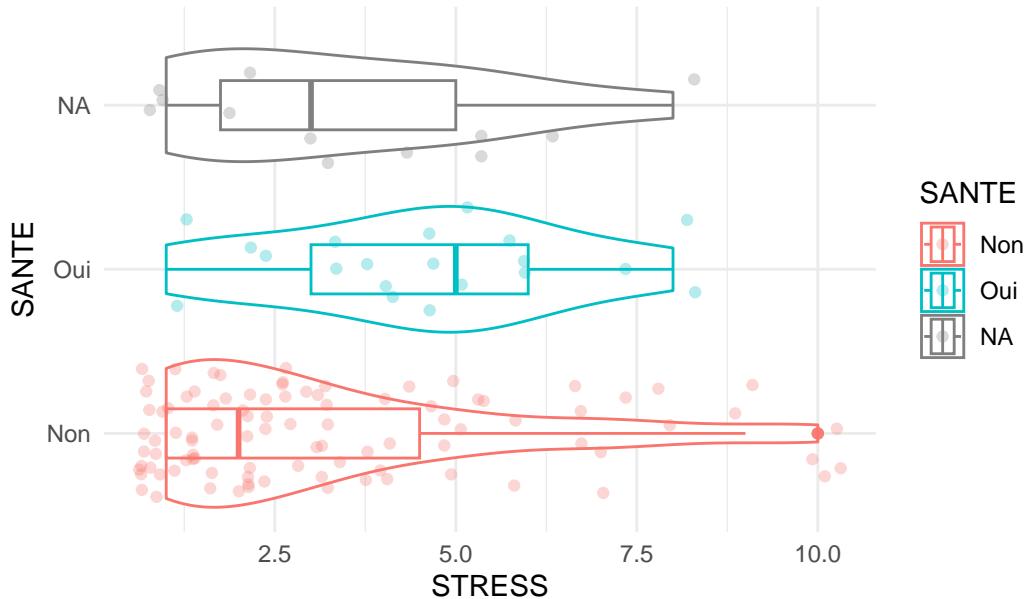


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de SANTE

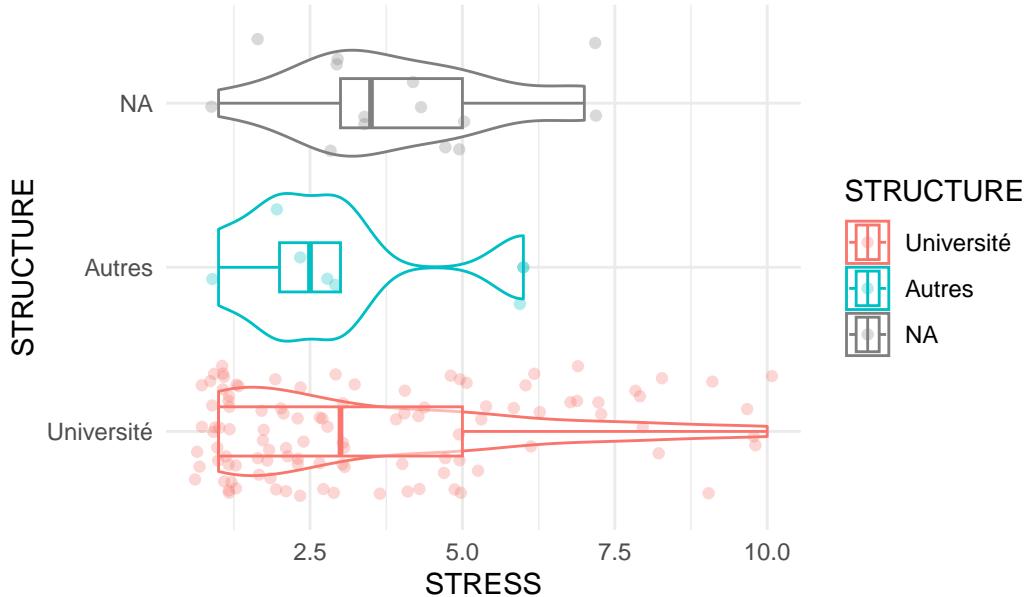


Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 8 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 8 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de STRESS en fonction de STRUCTURE

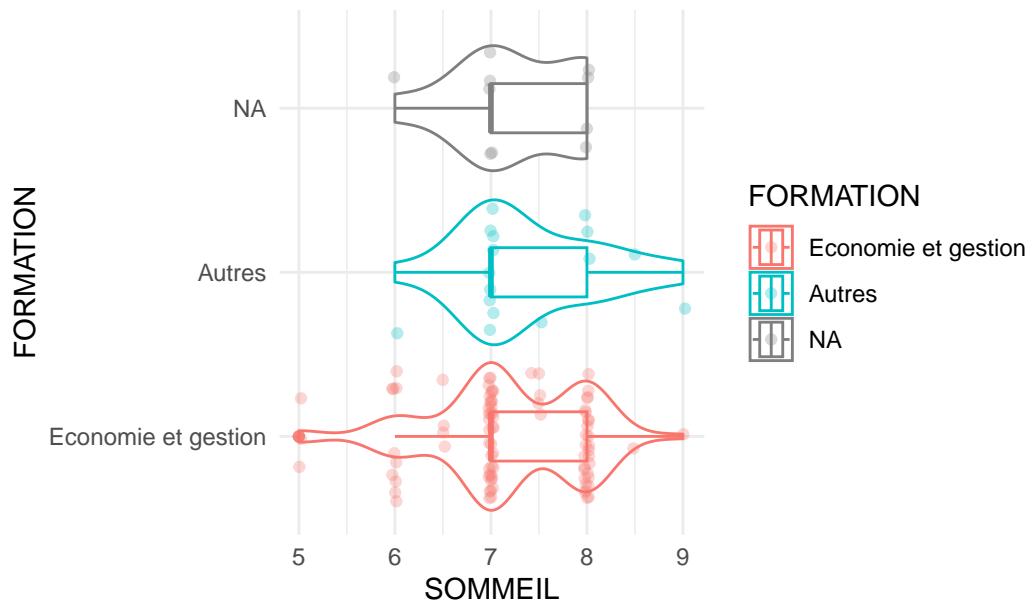


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de FORMATION

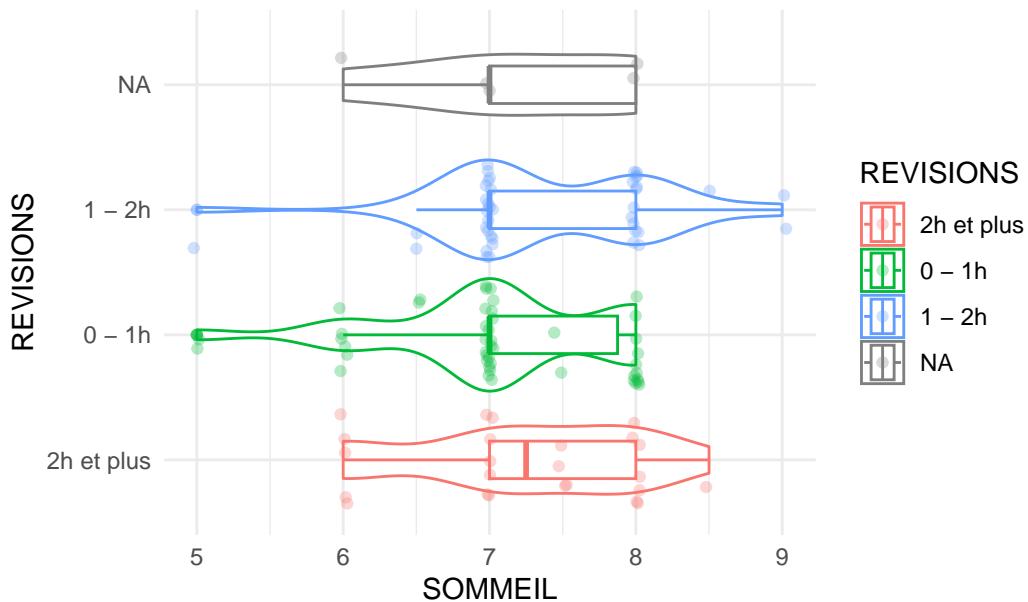


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de REVISIONS

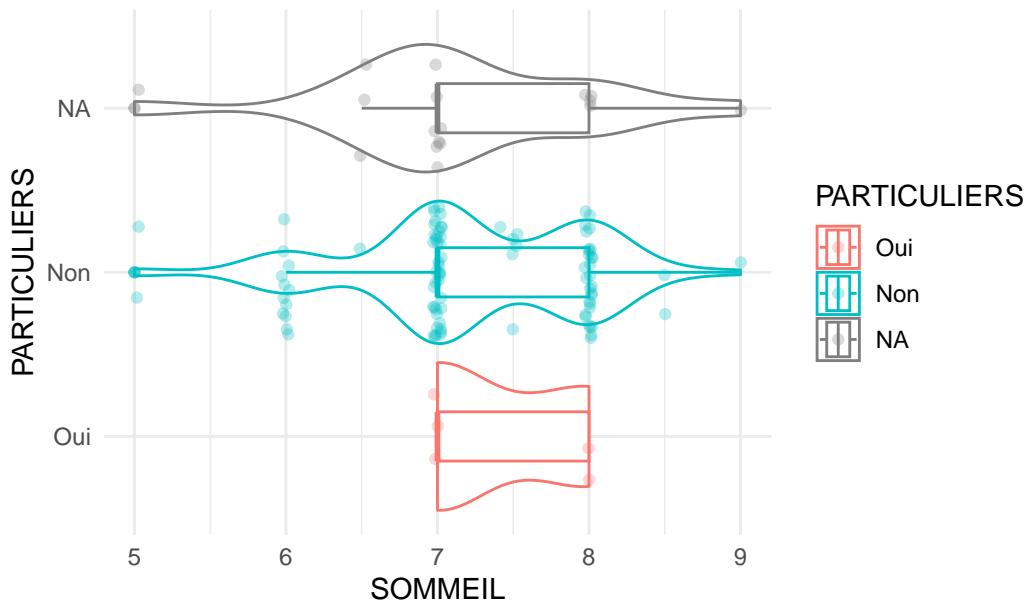


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de PARTICULIERS

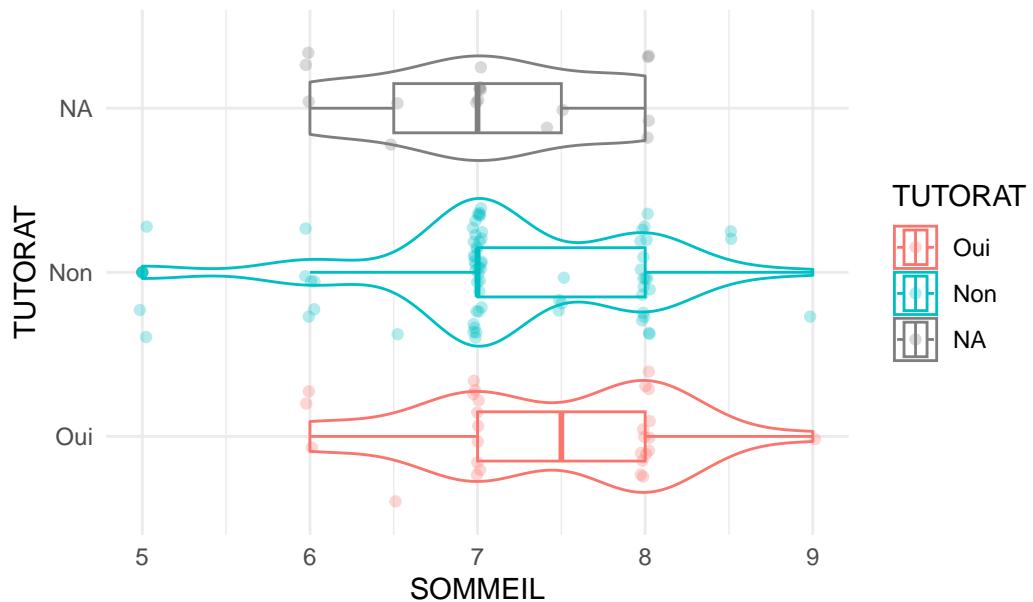


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de TUTORAT

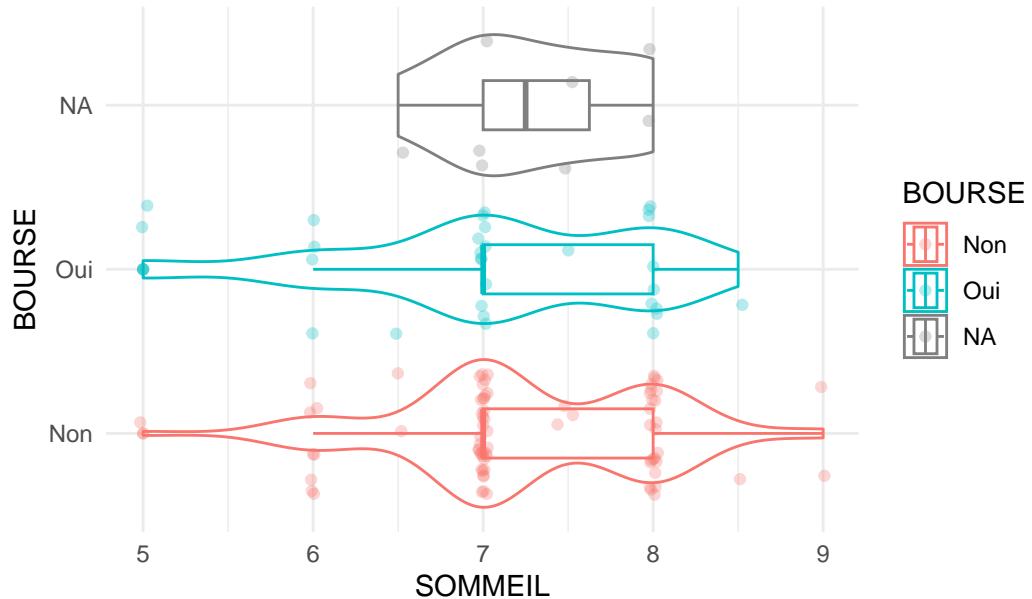


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de BOURSE

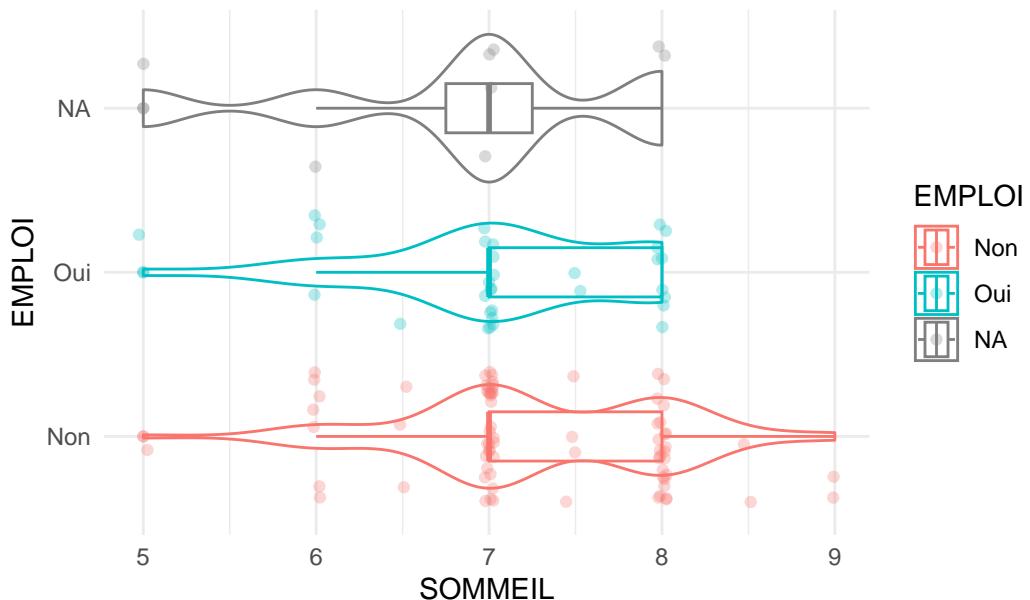


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de EMPLOI

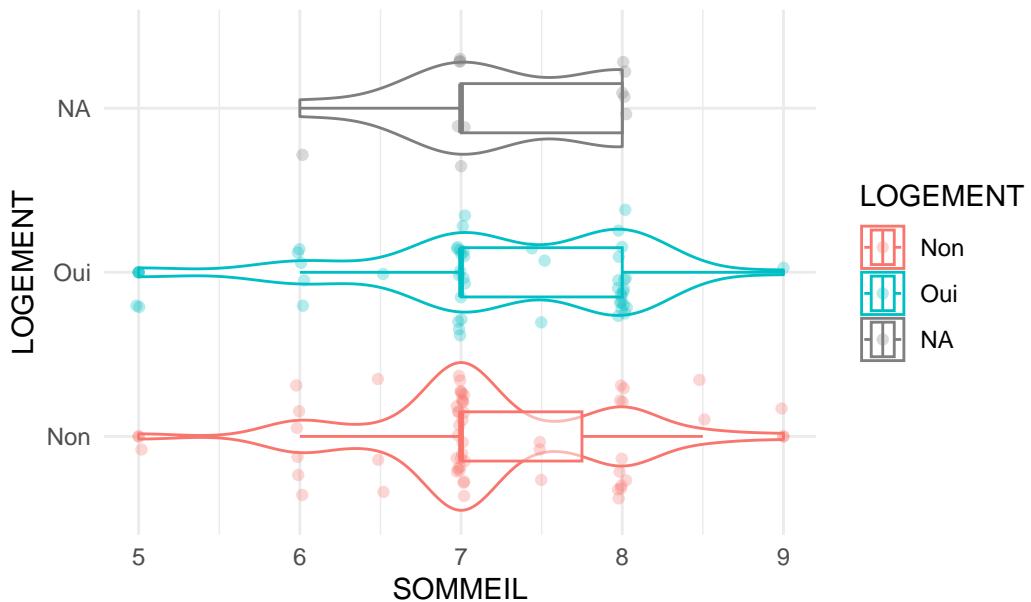


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de LOGEMENT

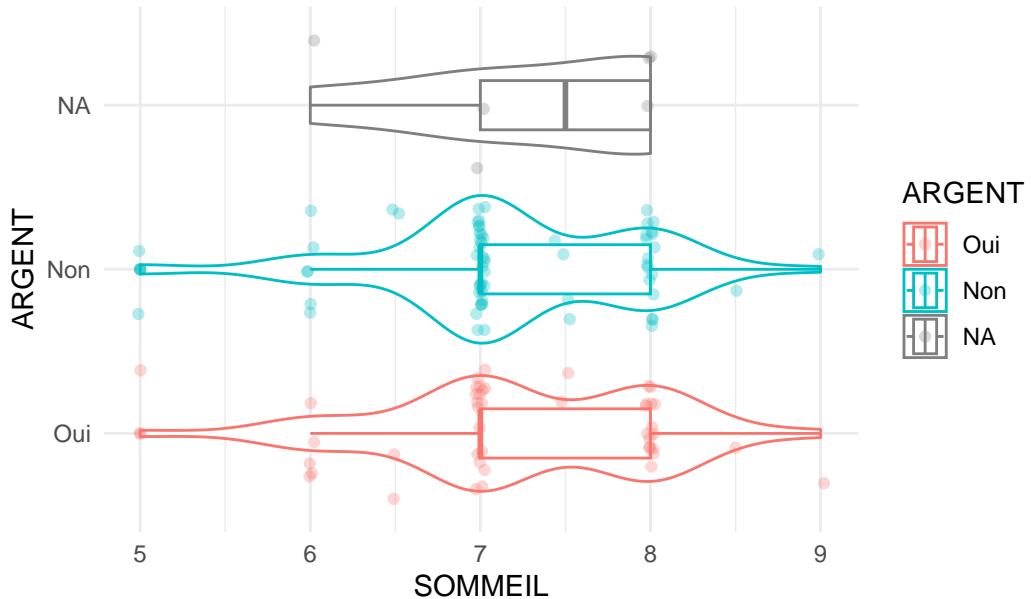


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de ARGENT

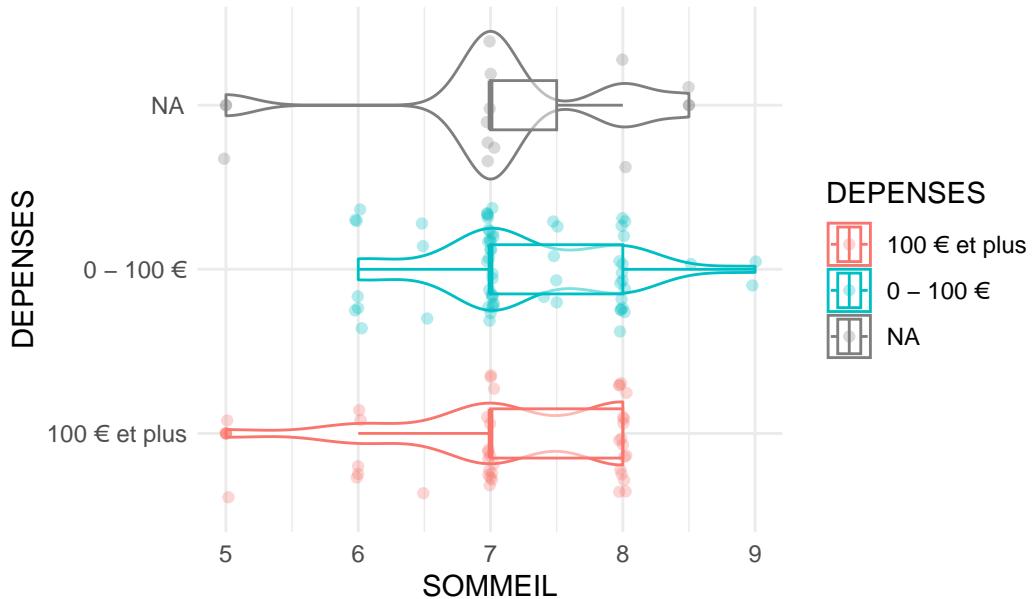


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de DEPENSES

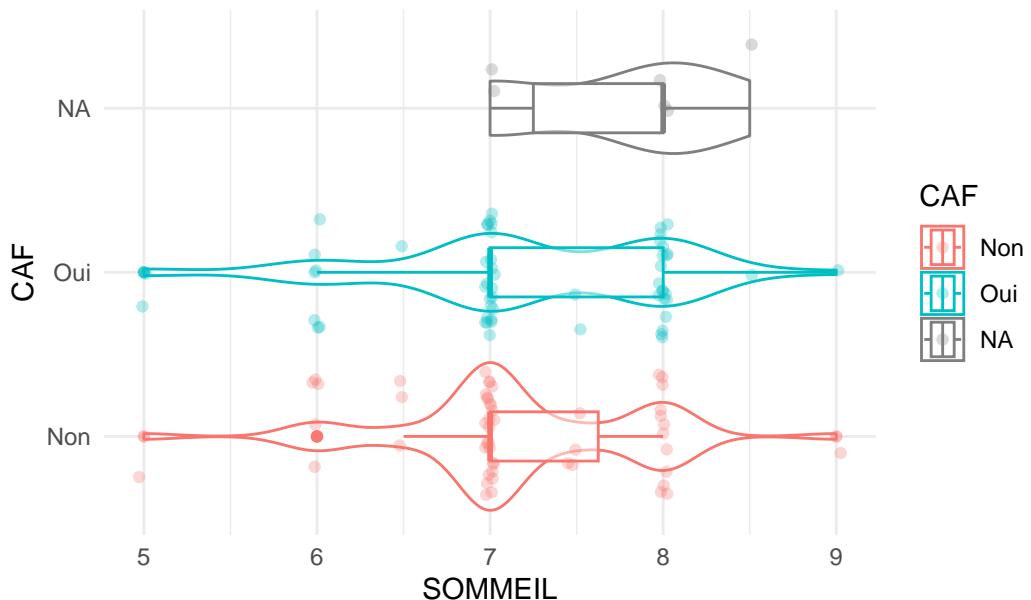


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de CAF

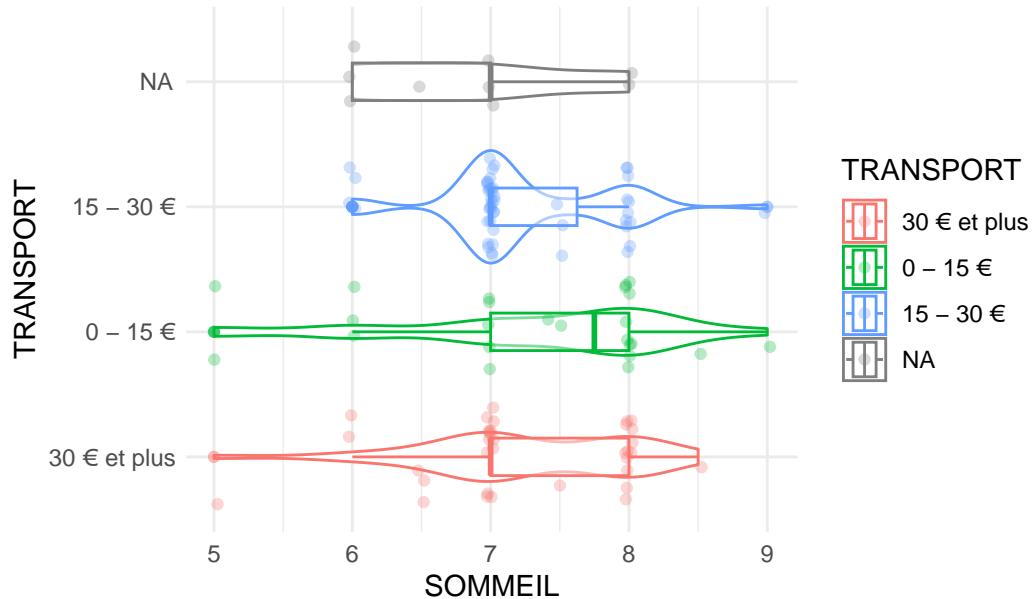


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de TRANSPORT

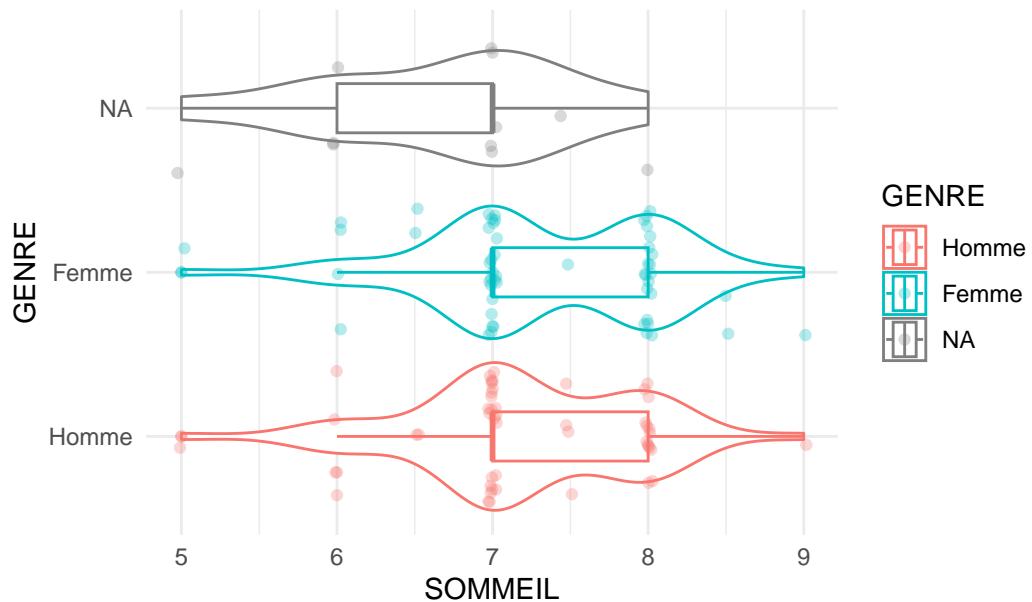


```
Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).
```

```
Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```

```
Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).
```

Distribution de SOMMEIL en fonction de GENRE

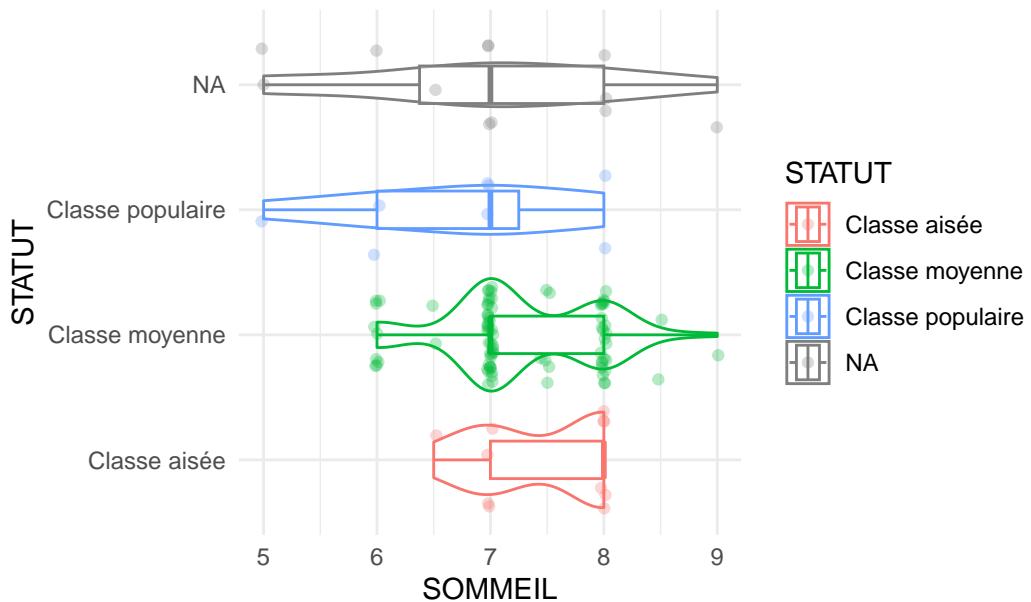


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de STATUT

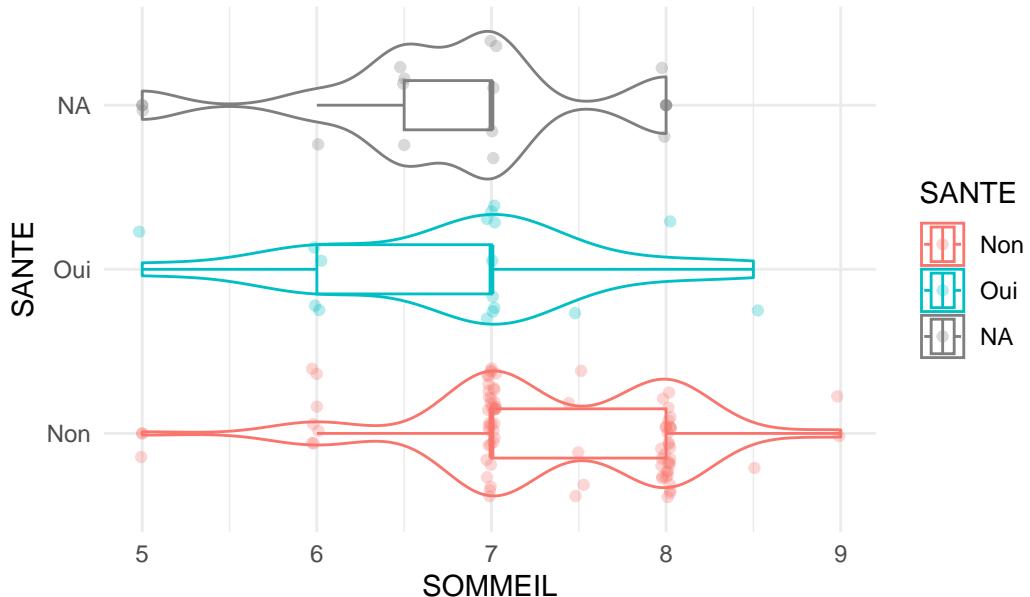


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de SANTE

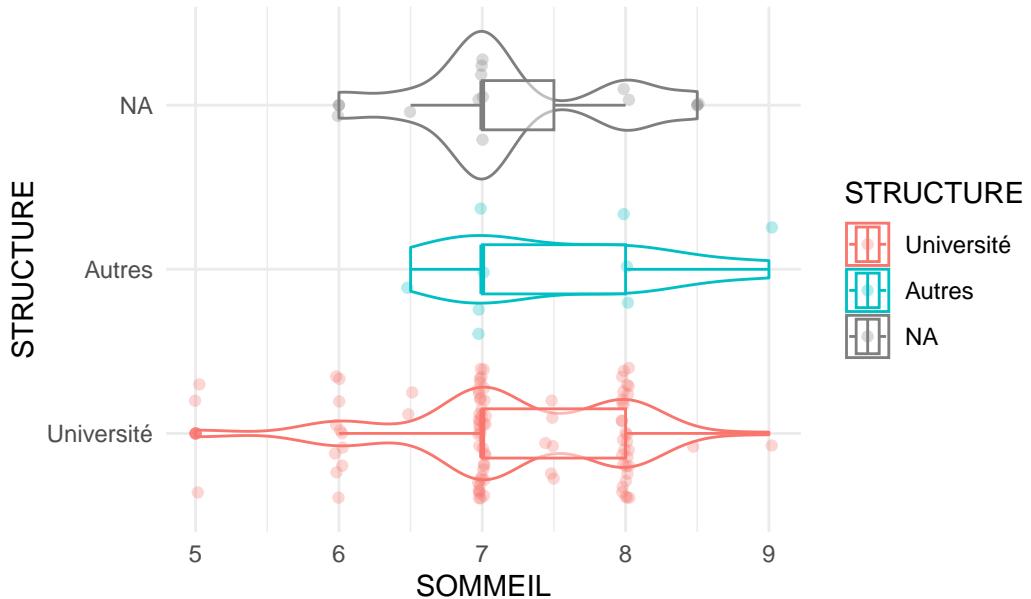


Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 20 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 20 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de SOMMEIL en fonction de STRUCTURE

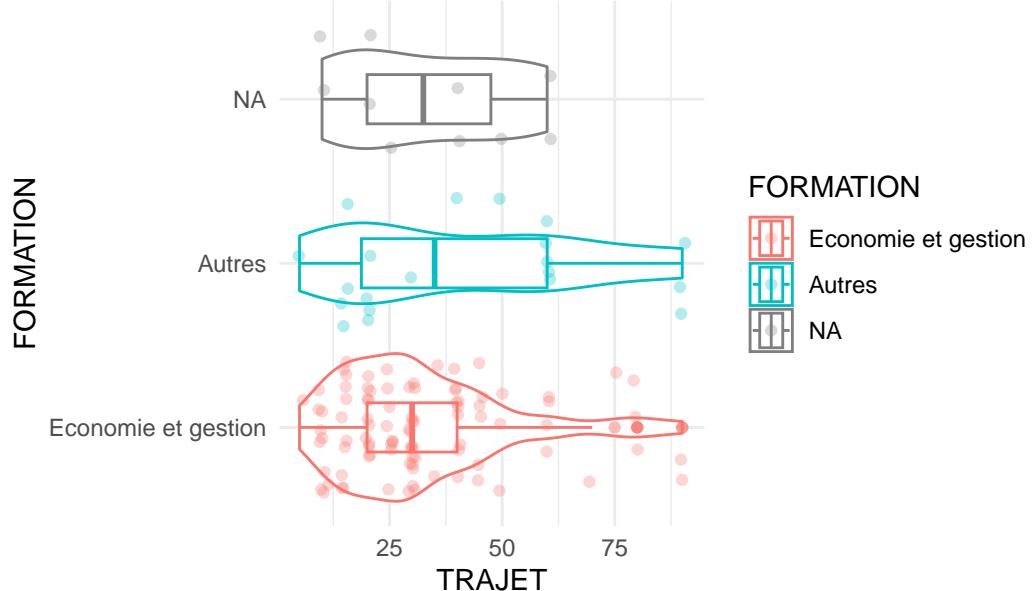


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de FORMATIO

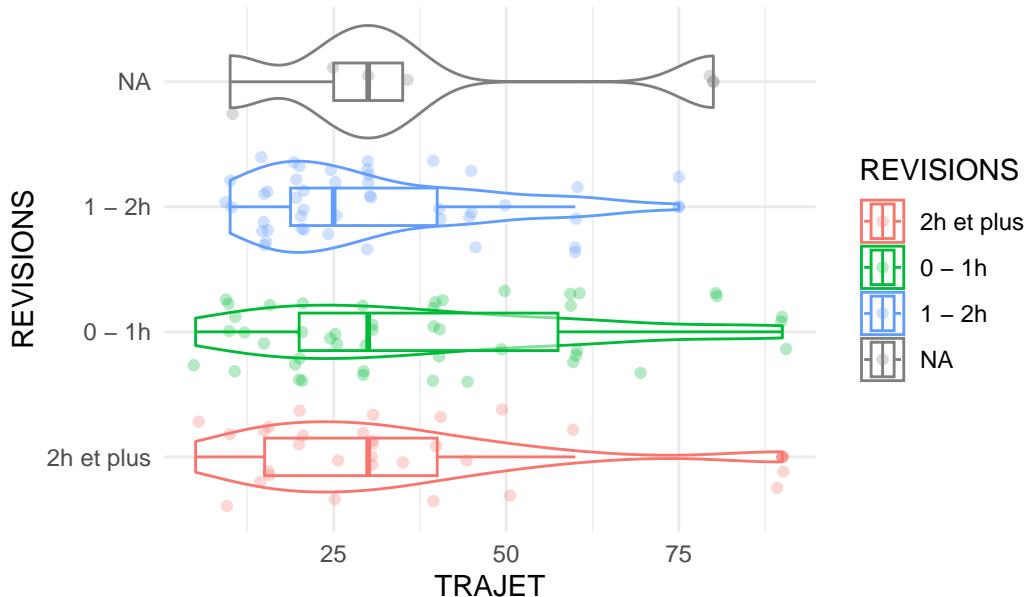


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de REVISIONS

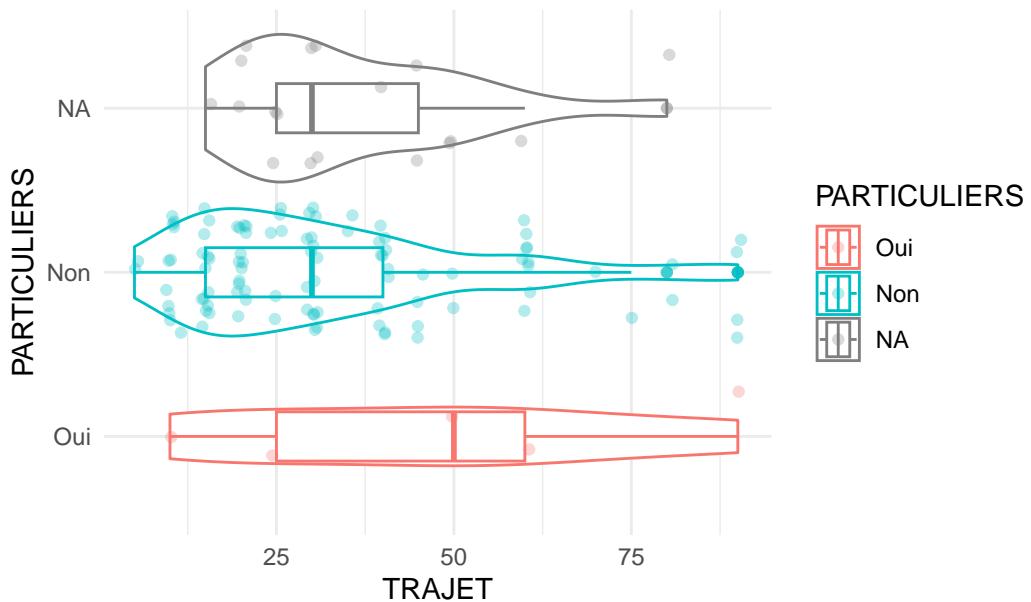


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de PARTICULIERS

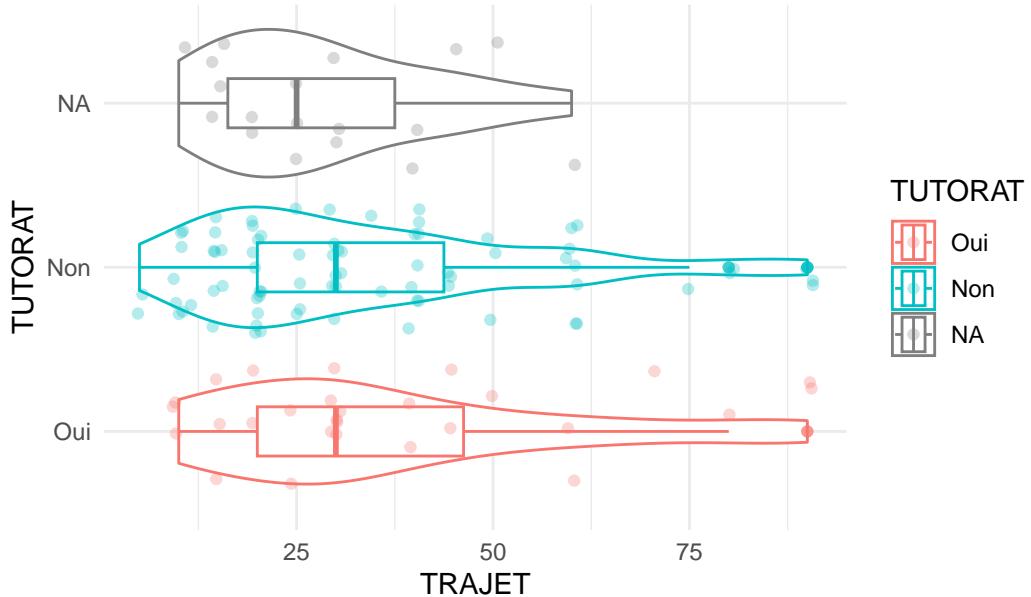


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de TUTORAT

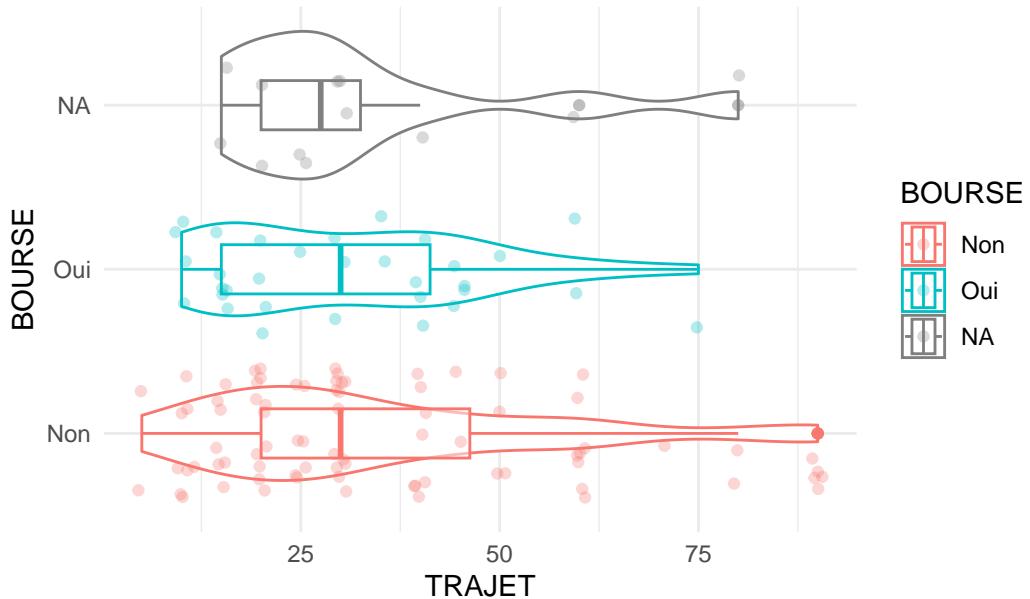


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de BOURSE

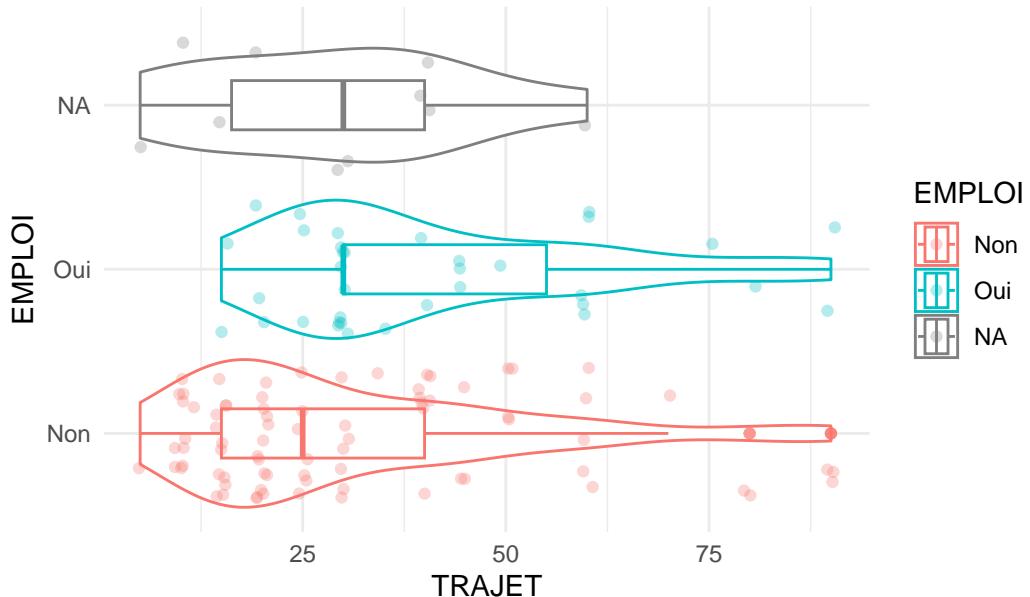


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de EMPLOI

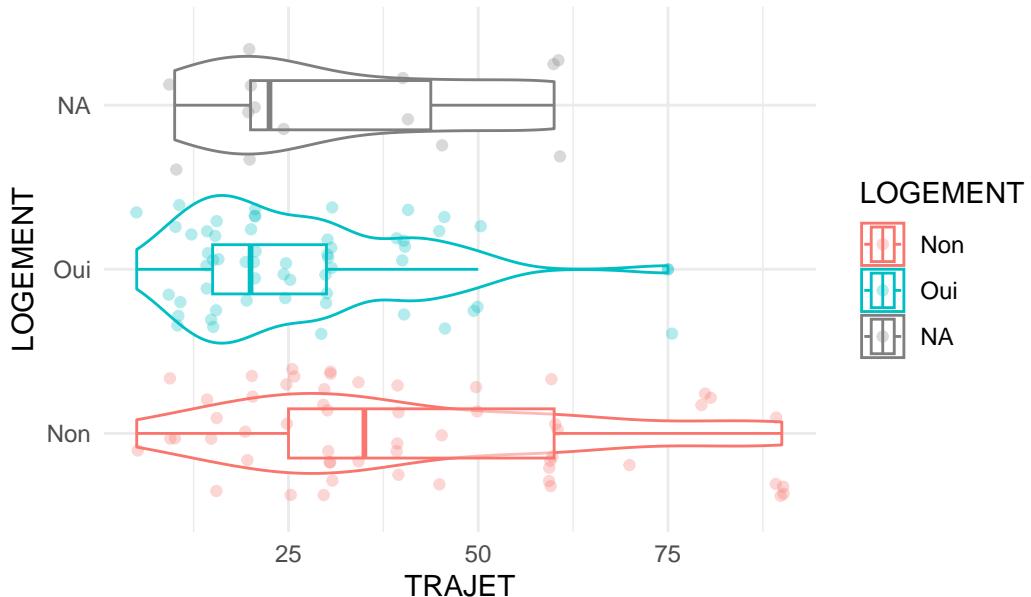


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de LOGEMENT

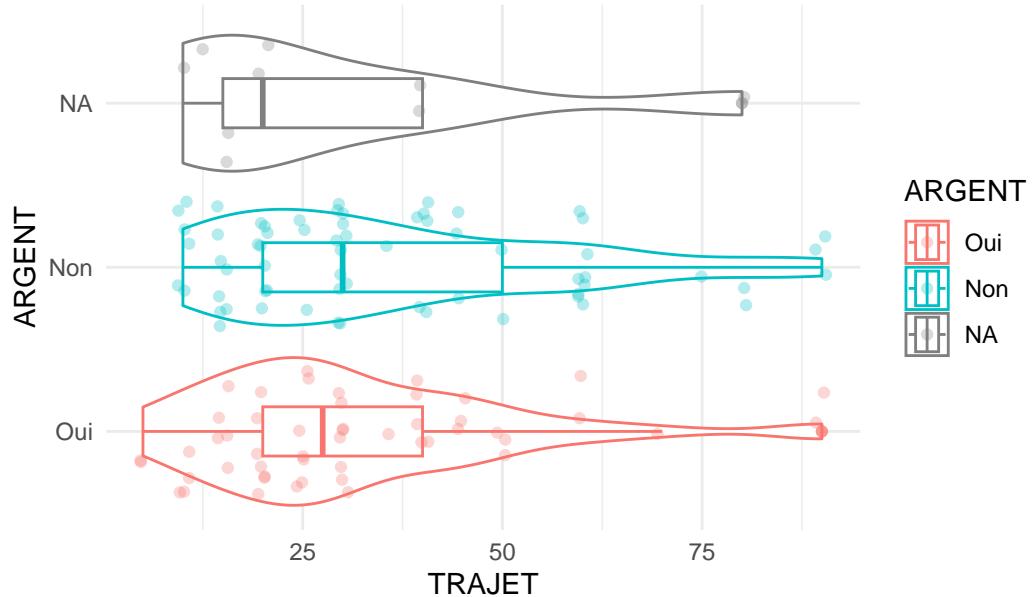


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de ARGENT

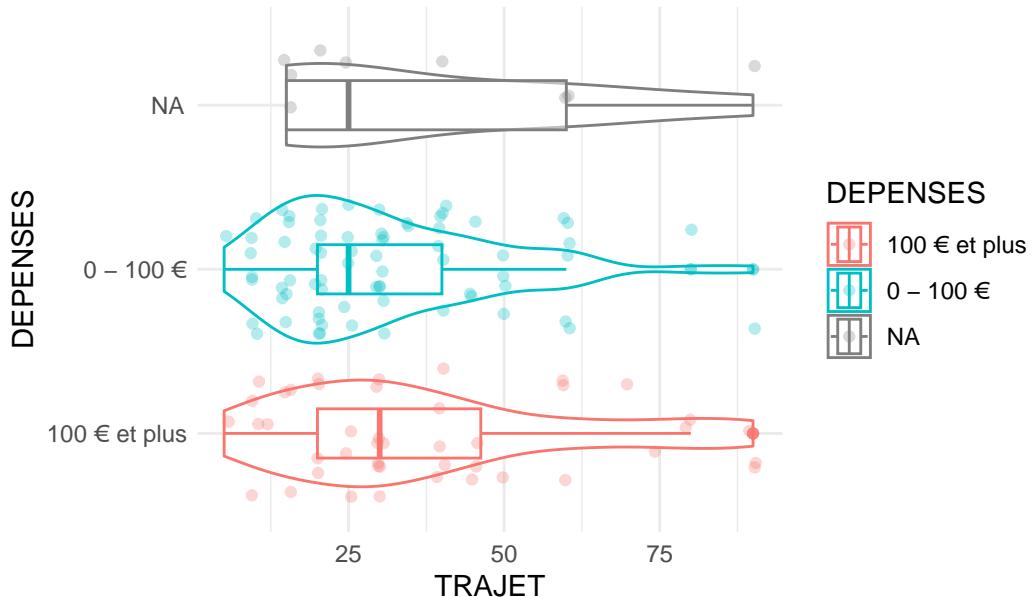


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de DEPENSES

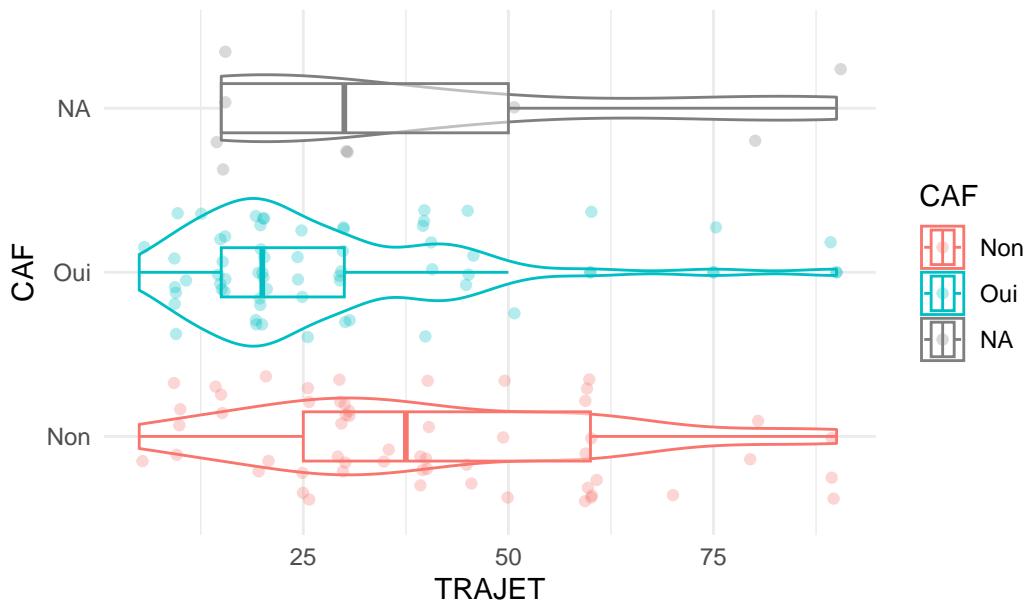


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de CAF

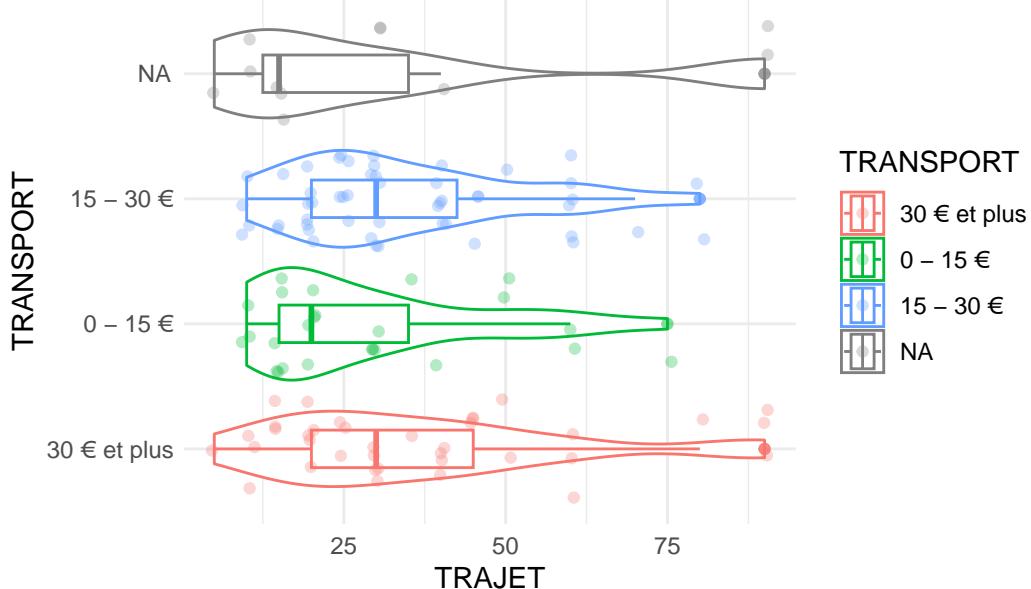


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de TRANSPORT

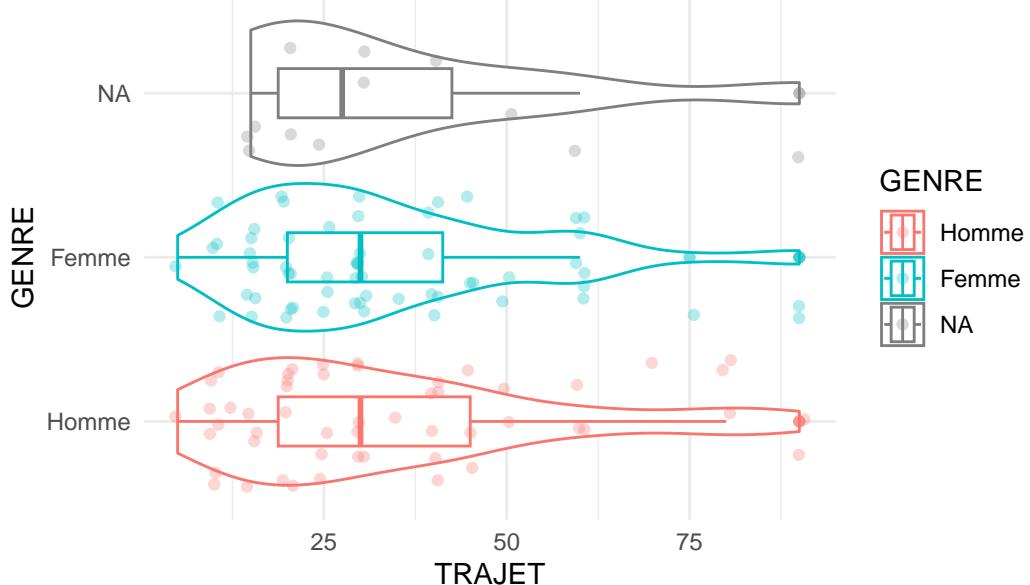


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de GENRE

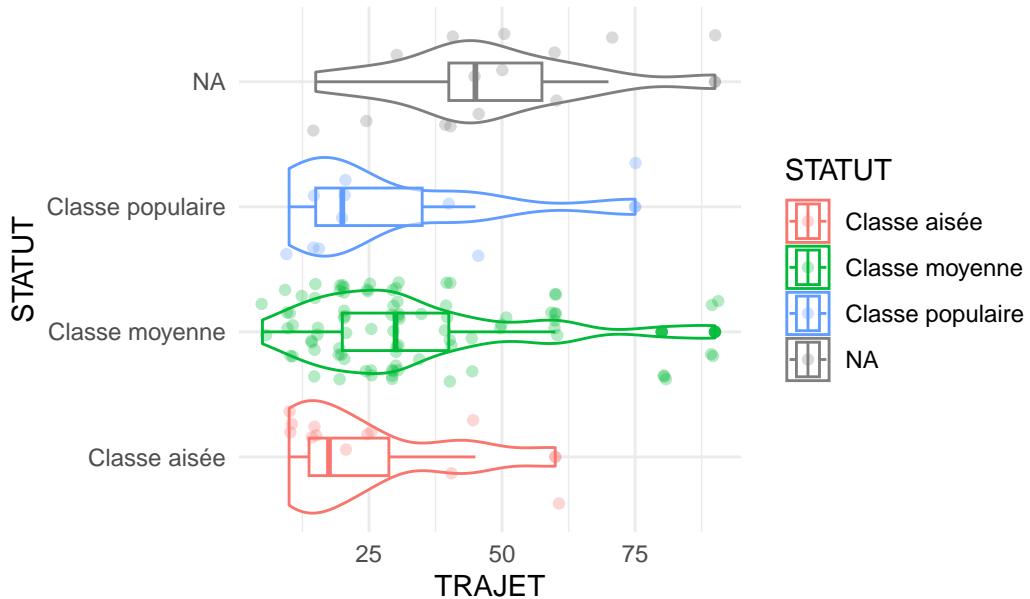


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de STATUT

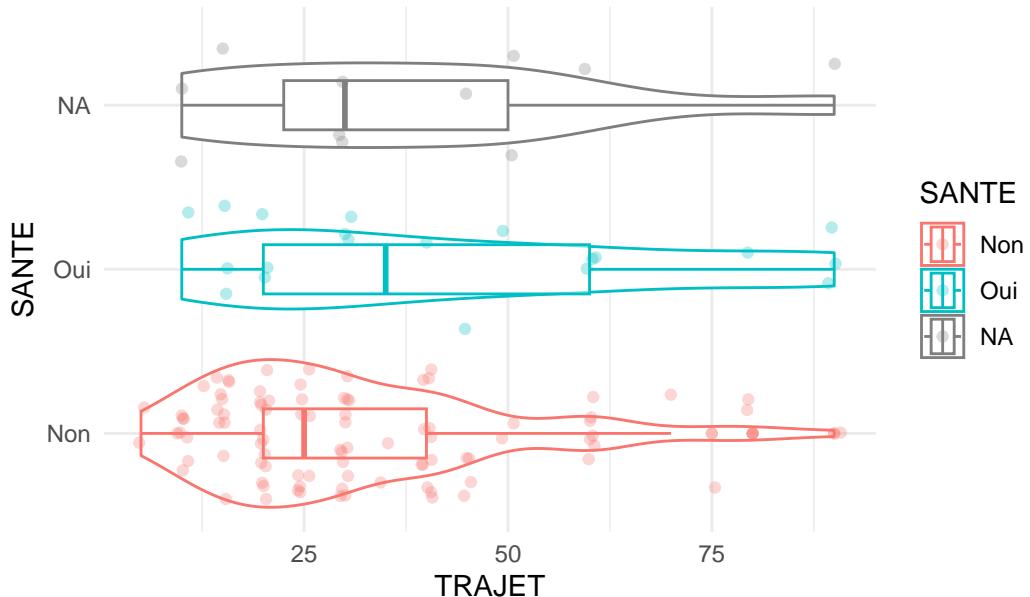


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de SANTE

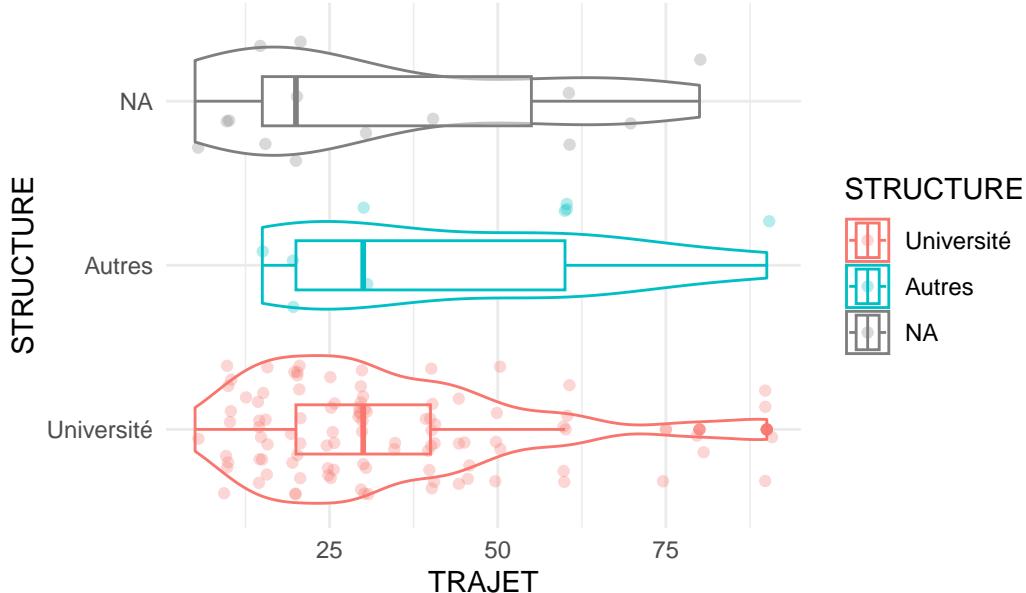


Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 11 rows containing missing values or values outside the scale range
(`geom_point()`).

Distribution de TRAJET en fonction de STRUCTURE



Les graphiques générés permettent d'explorer la distribution des variables quantitatives pour chaque modalité des variables qualitatives, afin de comparer ces distributions avec celle des valeurs manquantes. Les boîtes à moustaches montrent la dispersion de la variable quantitative à travers la médiane, les quartiles et les valeurs aberrantes, tandis que les violons illustrent la densité de la distribution de la variable quantitative selon chaque catégorie de la variable qualitative. Ces éléments permettent d'observer les différences de variabilité entre les modalités et de les comparer à la catégorie des valeurs manquantes.

VII. Imputation des valeurs manquantes : trois techniques utilisées

Cette partie se concentre sur l'étude de trois méthodes pour remplacer les valeurs manquantes : nous commencerons par la régression logistique binaire, suivie de l'imputation par la moyenne, et nous terminerons par l'imputation multiple grâce au package mice. L'objectif est de minimiser l'impact des données manquantes sur les résultats en sélectionnant les méthodes d'imputation les plus appropriées pour estimer ces valeurs.

A. Régression linéaire binaire

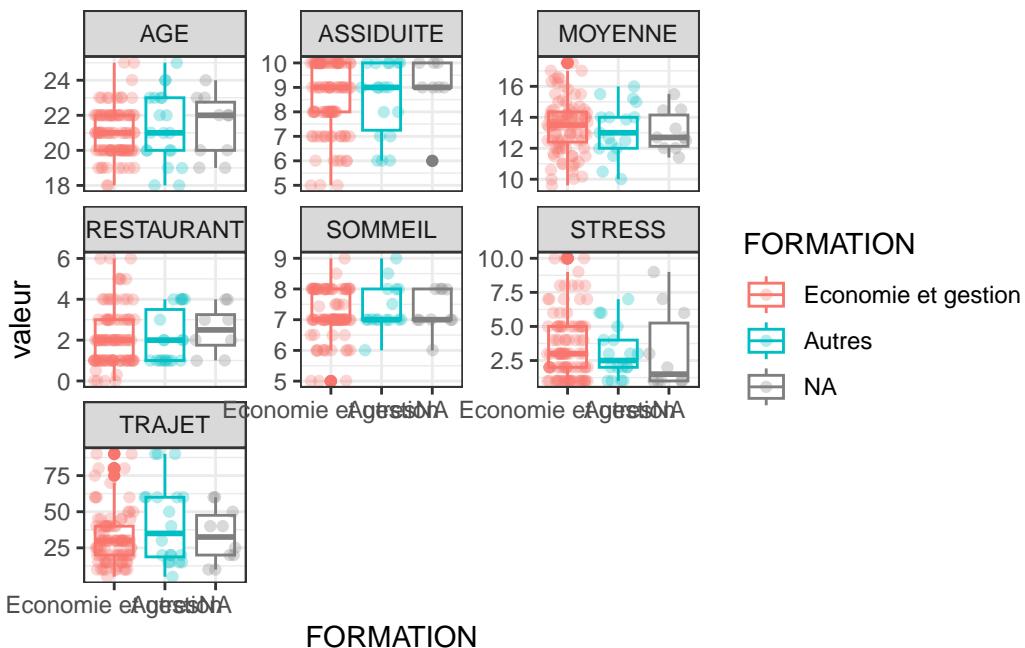
1. Visualisation des données avec des boîtes à moustaches pour FORMATION, PARTICULIERS, TUTORAT, BOURSE, EMPLOI, LOGEMENT, ARGENT, CAF ET GENRE

a. FORMATION

```
# Boites à moustaches selon les modalités de FORMATION pour chaque variable quantitati
Budget2 |>
pivot_longer(
  cols = where(is.numeric),
  names_to = "mesure",
  values_to = "valeur" ) |>
ggplot() +
aes(y = valeur, x = FORMATION, color = FORMATION) +
geom_boxplot() +
geom_jitter(alpha = 0.3) +
facet_wrap(~ mesure, scales = "free_y") +
theme_bw()
```

Warning: Removed 101 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 101 rows containing missing values or values outside the scale range
(`geom_point()`).



Nous constatons que les boites à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité **Economie et gestion**.

b. PARTICULIERS

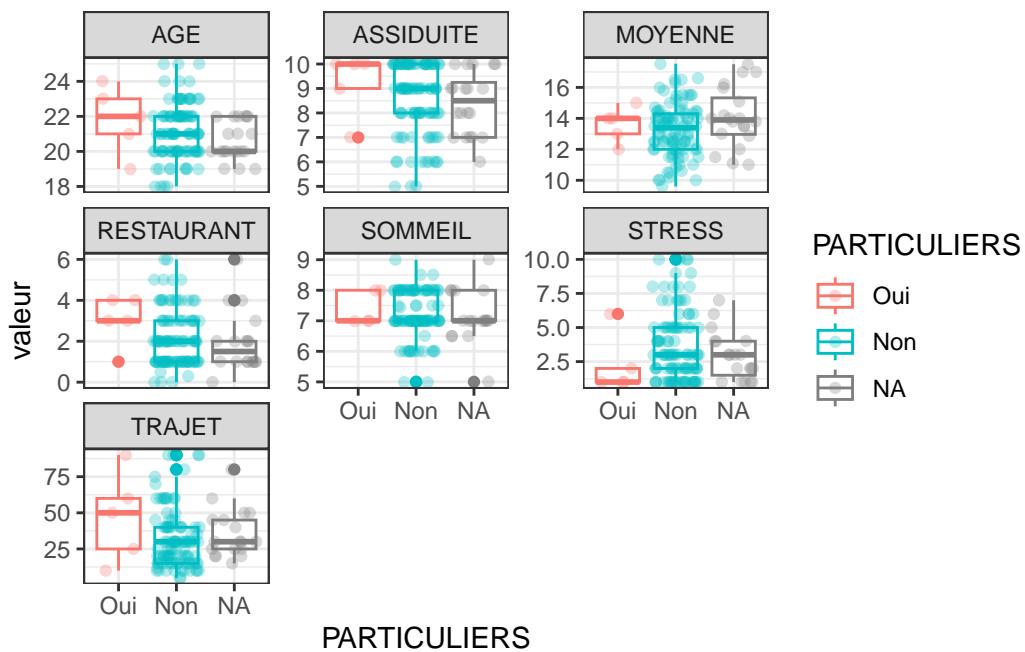
```
# Boites à moustaches selon les modalités de PARTICULIERS pour chaque variable quantit
```

```
Budget2 |>
pivot_longer(
  cols = where(is.numeric),
  names_to = "mesure",
  values_to = "valeur" ) |>
ggplot() +
  aes(y = valeur, x = PARTICULIERS, color = PARTICULIERS) +
  geom_boxplot() +
  geom_jitter(alpha = 0.3) +
  facet_wrap(~ mesure, scales = "free_y") +
  theme_bw()
```

Warning: Removed 101 rows containing non-finite outside the scale range

```
(`stat_boxplot()`).
```

```
Warning: Removed 101 rows containing missing values or values outside the scale range  
(`geom_point()`).
```



Nous constatons que les boites à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité **Non**.

c. TUTORAT

```
# Boites à moustaches selon les modalités de TUTORAT pour chaque variable quantitative  
  
Budget2 |>  
pivot_longer(  
  cols = where(is.numeric),  
  names_to = "mesure",  
  values_to = "valeur" ) |>  
ggplot() +  
aes(y = valeur, x = TUTORAT, color = TUTORAT) +  
geom_boxplot()
```

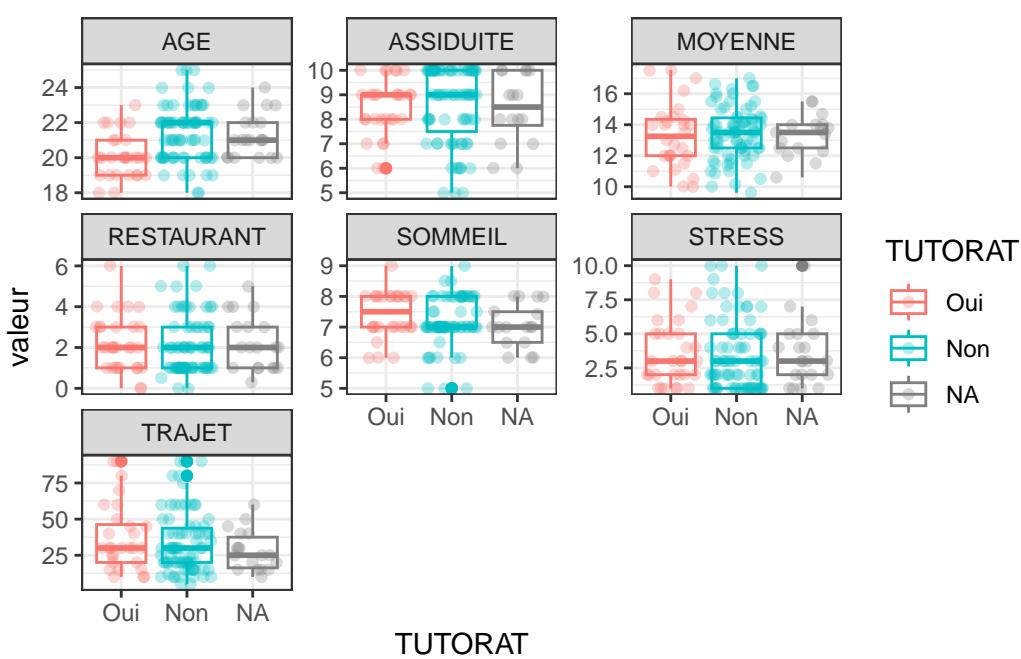
```

geom_jitter(alpha = 0.3) +
facet_wrap(~ mesure, scales = "free_y") +
theme_bw()

```

Warning: Removed 101 rows containing non-finite outside the scale range (`stat_boxplot()`).

Warning: Removed 101 rows containing missing values or values outside the scale range (`geom_point()`).



Nous constatons que les boîtes à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité **Non**.

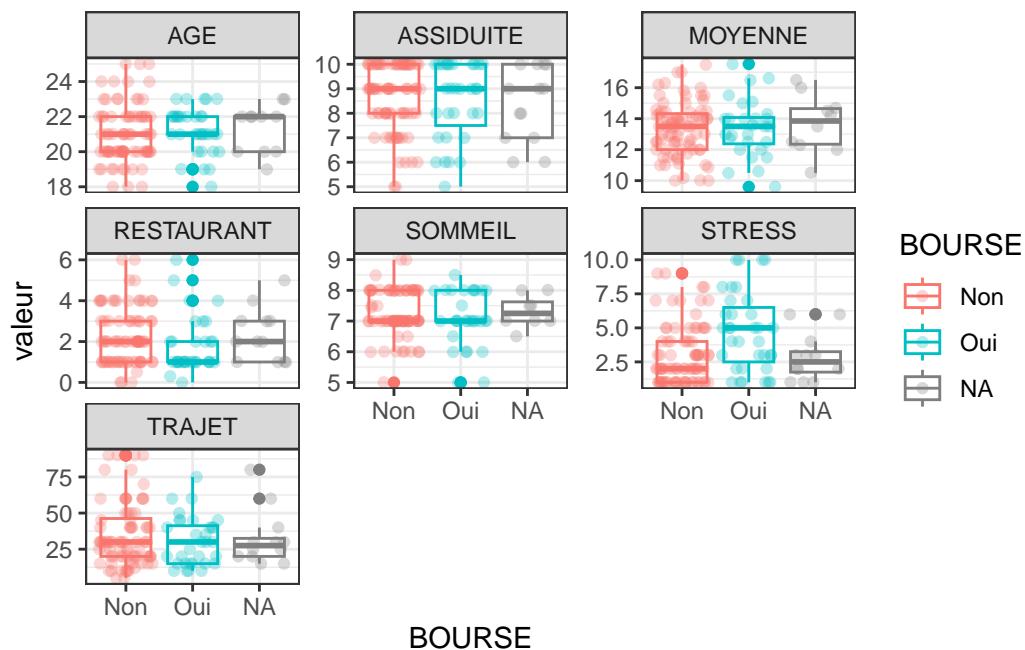
d. BOURSE

```
# Boites à moustaches selon les modalités de BOURSE pour chaque variable quantitative

Budget2 |>
pivot_longer(
  cols = where(is.numeric),
  names_to = "mesure",
  values_to = "valeur" ) |>
ggplot() +
aes(y = valeur, x = BOURSE, color = BOURSE) +
geom_boxplot() +
geom_jitter(alpha = 0.3) +
facet_wrap(~ mesure, scales = "free_y") +
theme_bw()
```

Warning: Removed 101 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 101 rows containing missing values or values outside the scale range
(`geom_point()`).



Nous constatons que les boites à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité **Oui**.

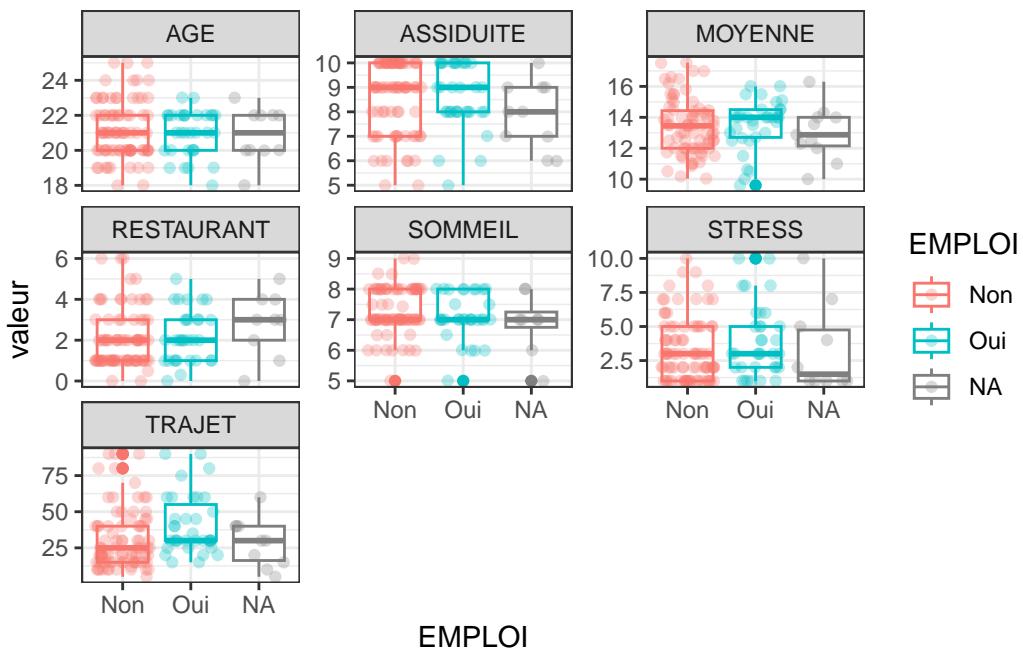
e. EMPLOI

```
# Boites à moustaches selon les modalités de EMPLOI pour chaque variable quantitative

Budget2 |>
pivot_longer(
  cols = where(is.numeric),
  names_to = "mesure",
  values_to = "valeur" ) |>
ggplot() +
aes(y = valeur, x = EMPLOI, color = EMPLOI) +
geom_boxplot() +
geom_jitter(alpha = 0.3) +
facet_wrap(~ mesure, scales = "free_y") +
theme_bw()
```

Warning: Removed 101 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 101 rows containing missing values or values outside the scale range
(`geom_point()`).



Nous constatons que les boites à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité **Oui**.

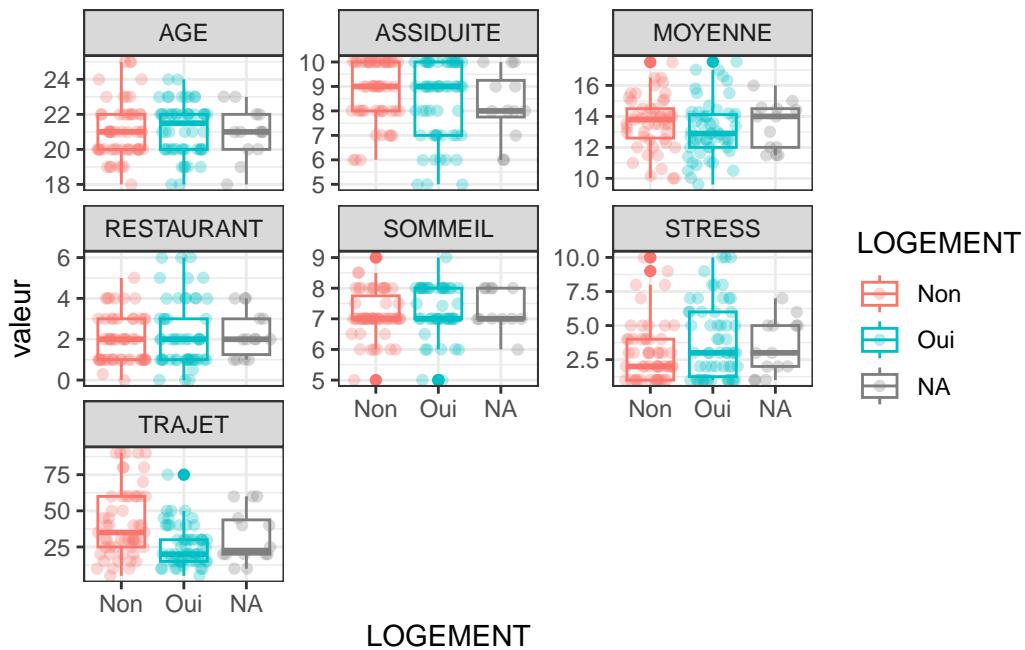
f. LOGEMENT

```
# Boites à moustaches selon les modalités de LOGEMENT pour chaque variable quantitativ
Budget2 |>
pivot_longer(
  cols = where(is.numeric),
  names_to = "mesure",
  values_to = "valeur" ) |>
ggplot() +
  aes(y = valeur, x = LOGEMENT, color = LOGEMENT) +
  geom_boxplot() +
  geom_jitter(alpha = 0.3) +
  facet_wrap(~ mesure, scales = "free_y") +
  theme_bw()
```

Warning: Removed 101 rows containing non-finite outside the scale range

```
(`stat_boxplot()`).
```

```
Warning: Removed 101 rows containing missing values or values outside the scale range  
(`geom_point()`).
```



Nous constatons que les boites à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité **Oui**.

g. ARGENT

```
# Boites à moustaches selon les modalités de ARGENT pour chaque variable quantitative  
  
Budget2 |>  
pivot_longer(  
  cols = where(is.numeric),  
  names_to = "mesure",  
  values_to = "valeur" ) |>  
ggplot() +  
aes(y = valeur, x = ARGENT, color = ARGENT) +  
geom_boxplot()
```

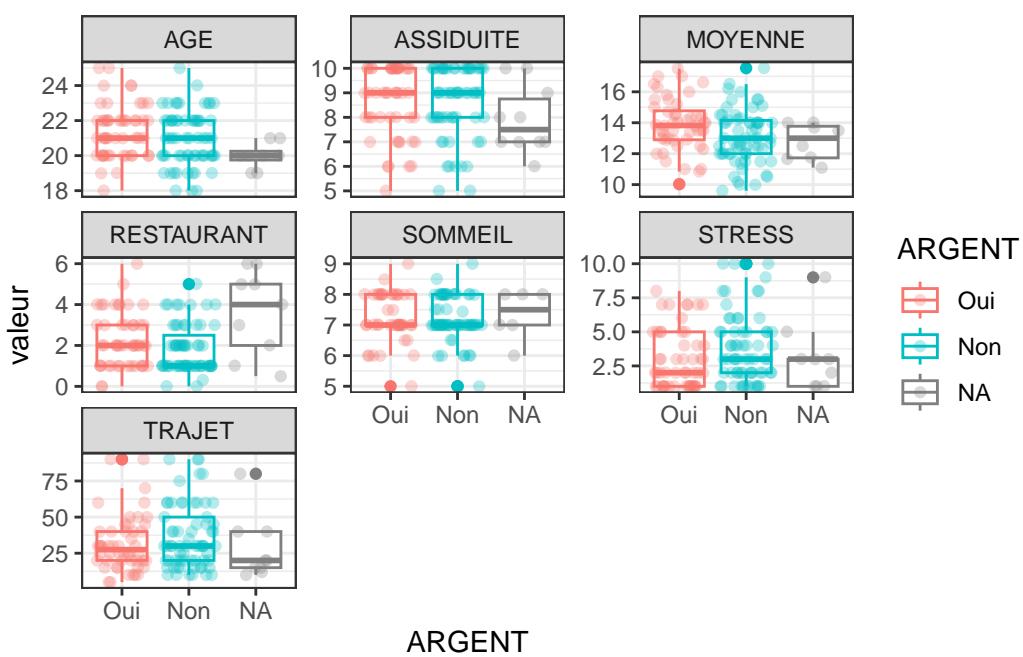
```

geom_jitter(alpha = 0.3) +
facet_wrap(~ mesure, scales = "free_y") +
theme_bw()

```

Warning: Removed 101 rows containing non-finite outside the scale range (`stat_boxplot()`).

Warning: Removed 101 rows containing missing values or values outside the scale range (`geom_point()`).



Nous constatons que les boîtes à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité Oui.

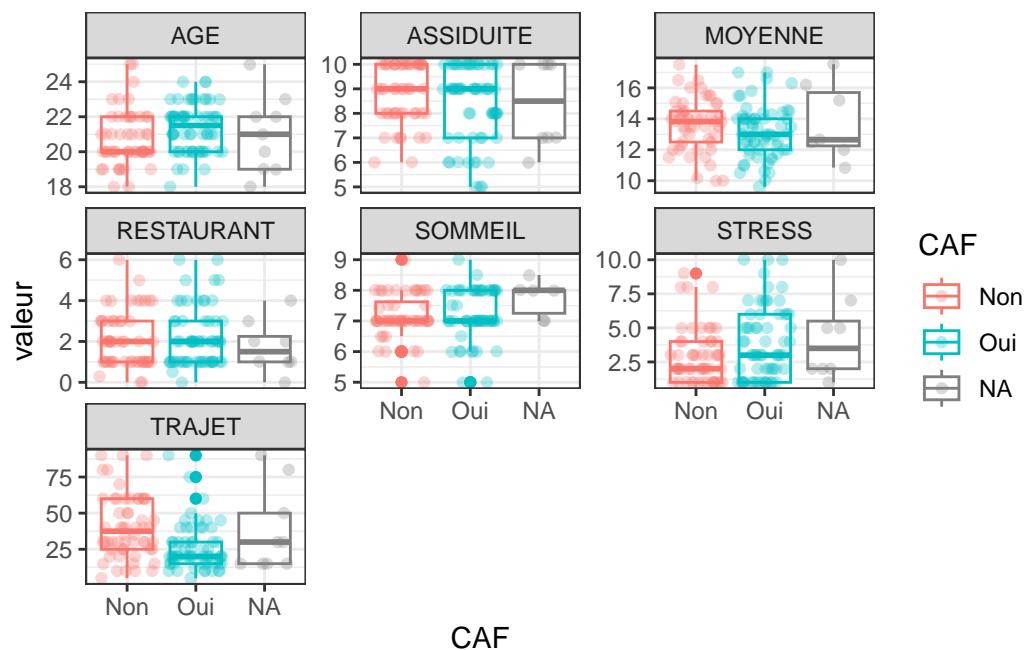
h. CAF

```
# Boites à moustaches selon les modalités de CAF pour chaque variable quantitative

Budget2 |>
pivot_longer(
  cols = where(is.numeric),
  names_to = "mesure",
  values_to = "valeur" ) |>
ggplot() +
aes(y = valeur, x = CAF, color = CAF) +
geom_boxplot() +
geom_jitter(alpha = 0.3) +
facet_wrap(~ mesure, scales = "free_y") +
theme_bw()
```

Warning: Removed 101 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 101 rows containing missing values or values outside the scale range
(`geom_point()`).



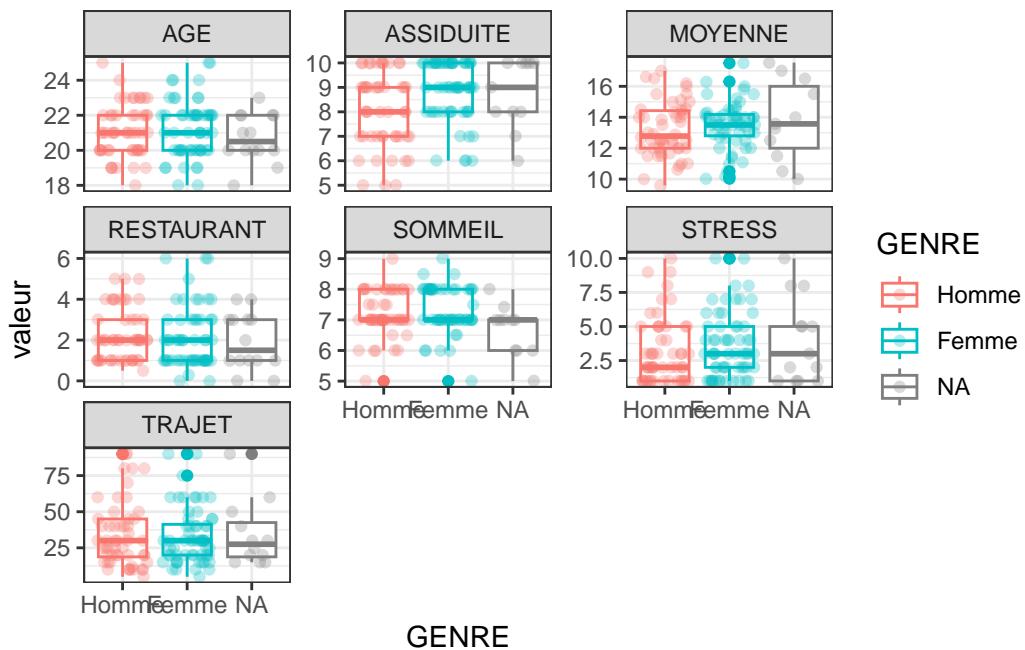
Nous constatons que les boites à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité **Oui**.

i. GENRE

```
# Boites à moustaches selon les modalités de GENRE pour chaque variable quantitative  
  
Budget2 |>  
pivot_longer(  
  cols = where(is.numeric),  
  names_to = "mesure",  
  values_to = "valeur" ) |>  
ggplot() +  
aes(y = valeur, x = GENRE, color = GENRE) +  
geom_boxplot() +  
geom_jitter(alpha = 0.3) +  
  
facet_wrap(~ mesure, scales = "free_y") +  
theme_bw()
```

Warning: Removed 101 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 101 rows containing missing values or values outside the scale range
(`geom_point()`).



Nous constatons que les boites à moustaches correspondant aux valeurs manquantes présentent des caractéristiques qui se rapprochent davantage de celles de la modalité **Femmes**.

2. Imputation

a. FORMATION

Avant d'effectuer une régression logistique binaire, il est nécessaire de binariser la variable FORMATION.

```
# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(For = case_when(
    FORMATION == "Autres" ~ 0,
    FORMATION == "Economie et gestion" ~ 1
  ))
```

Ensuite, nous créons deux jeux de données : un premier qui permettra d'entrainer notre modèle et un second qui nous servira à vérifier la qualité de prédiction du modèle.

```

set.seed(123)

# Entraînement sur 80 % des données non manquantes.

formation_entrainement <- Budget2 |>
  filter(!is.na(For)) |>
  slice_sample(prop = 0.8)

# Vérification sur les 20 % des données non manquantes restantes

formation_verification <- anti_join(Budget2, formation_entrainement) |>
  filter(!is.na(For))

```

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS, TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF, TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET, For)`

A présent, nous pouvons passer à la régression logistique binaire.

```

# Création du modèle de régression logistique binaire

regression_logistique1 <- glm(
  For ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = formation_entrainement,
  family = binomial
)

# Test d'Anova pour le modèle

car::Anova(regression_logistique1)

```

Analysis of Deviance Table (Type II tests)

Response: For

	LR	Chisq	Df	Pr(>Chisq)
AGE	4.6147	1	0.031700	*
ASSIDUITE	6.7053	1	0.009612	**
MOYENNE	2.2167	1	0.136527	
RESTAURANT	9.8572	1	0.001692	**

```

SOMMEIL      3.0054  1   0.082989 .
STRESS       0.5136  1   0.473585
TRAJET       10.5193  1   0.001181 **

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

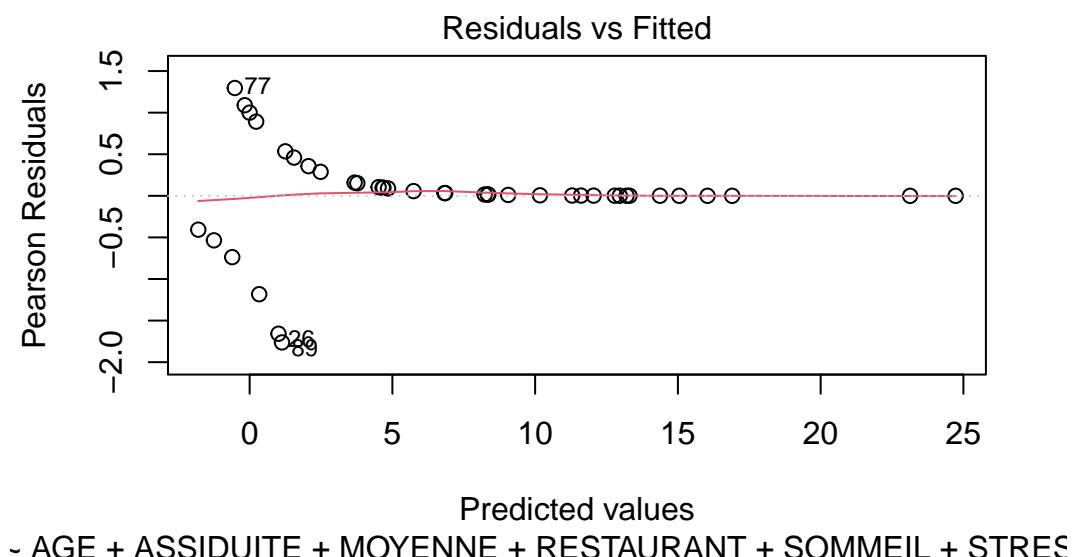
Dans cette régression, trois variables sont significatives au seuil de 1 % : ASSIDUITE, RESTAURANT et TRAJET. Nous avons la variable AGE significative au seuil de 5 %. Enfin, notre variable SOMMEIL est significative au seuil de 10 %.

En essayant de supprimer les variables dont les p-values étaient les plus élevées, notre modèle devenait moins pertinent. Nous gardons donc notre modèle initial.

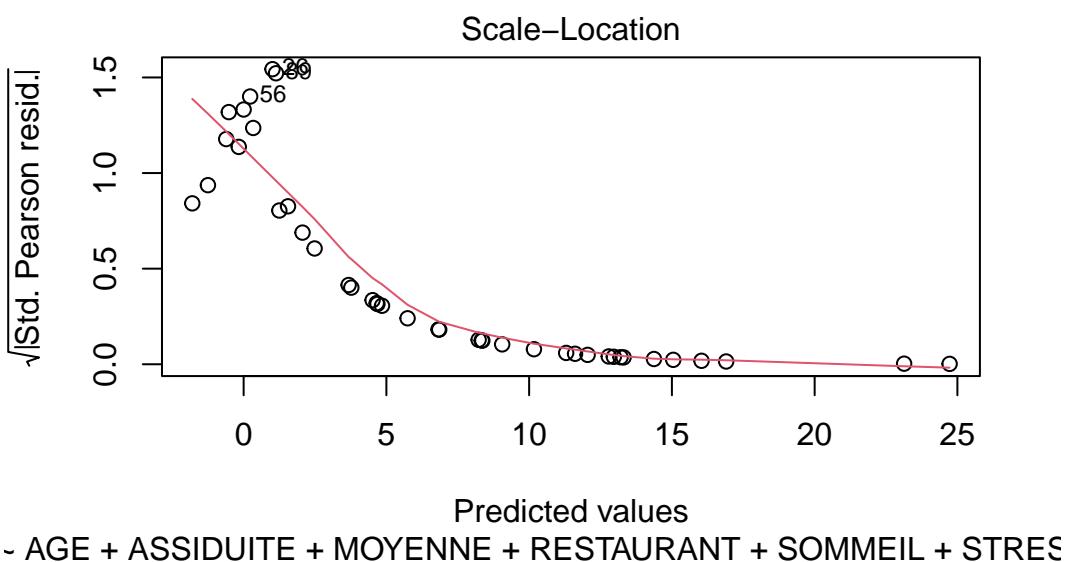
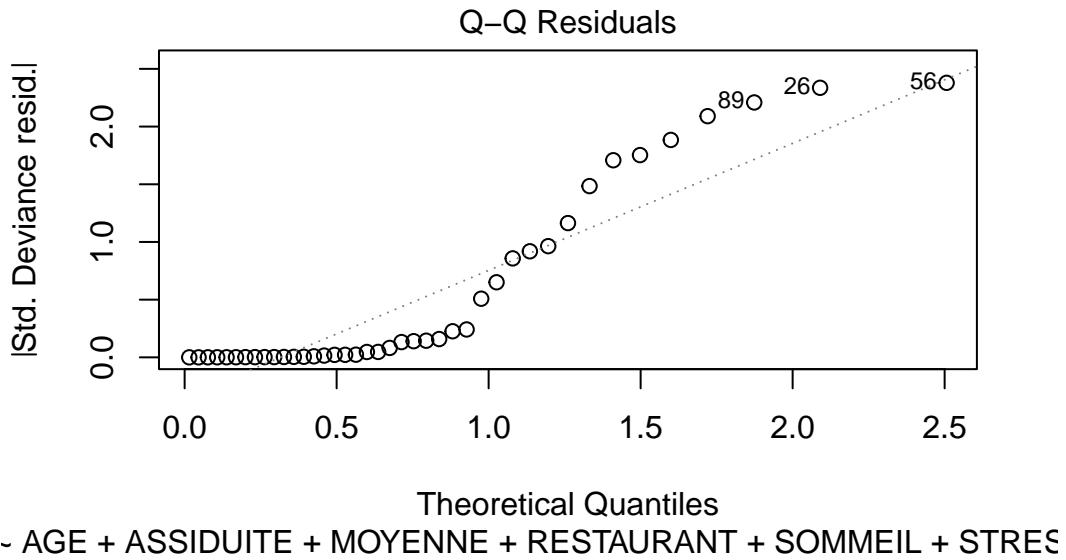
```

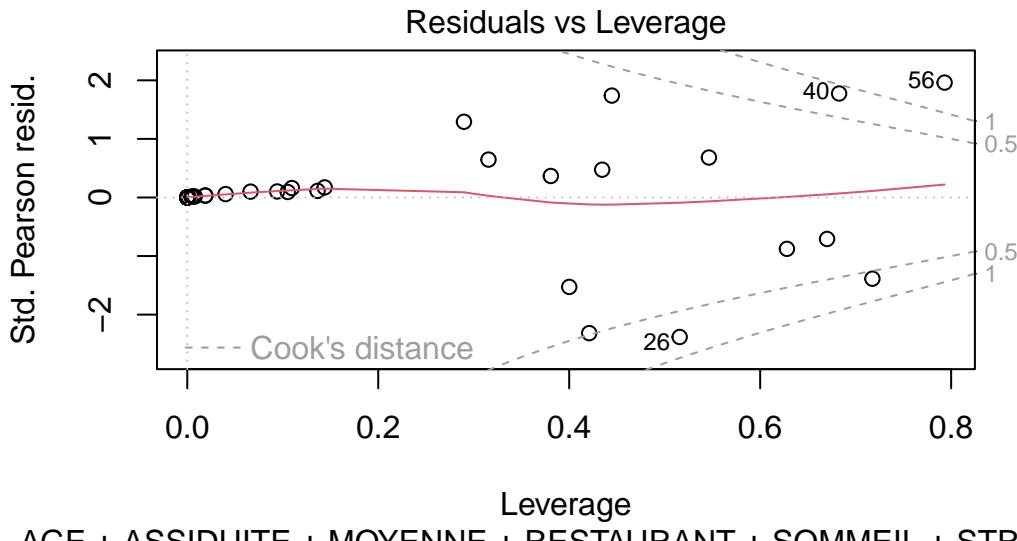
# Graphiques évaluant la qualité et la validité du modèle
plot(regression_logistique1)

```



- AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRES





Le premier graphique, nous permet de vérifier les résidus de ce modèle, ces individus qui ressortent signifiant que les résidus sont différents de leur valeur d'origine. Cependant, l'intervalle était bien compris entre -2 et 2 donc cela n'affecte en rien notre analyse. La même conclusion peut être faite pour les trois autres graphiques.

Nous allons ensuite tester notre modèle de régression logistique binaire sur 20 % de nos données.

```
# Prédiction du modèle sur les 20 % des données non manquantes restantes

prediction1 = predict(regression_logistique1,
                      newdata = formation_verification)

# Ajout de la prédiction parmi les 20 % de données en attribuant à chaque observation

formation_verification <- formation_verification |>
  mutate(prediction1) |>
  mutate(For_predi = case_when(prediction1 < 0 ~ "Autres",
                               TRUE ~ "Economie et gestion"))

# Comparaison des prédictions du modèle avec les vraies valeurs de la variable FORMATI
```

```

formation_verification |>
  count(FORMATION, For_predi)

# A tibble: 3 x 3
  FORMATION      For_predi     n
  <fct>          <chr>       <int>
1 Economie et gestion Autres        1
2 Economie et gestion Economie et gestion    21
3 Autres          Economie et gestion      3

# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction2 = predict(regression_logistique1,
                      newdata = filter(Budget2, is.na(FORMATION)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(FORMATION)) |>
    mutate(prediction2) |>
    mutate(FORMATION = case_when(prediction2 < 0 ~ "Autres",
                                 TRUE ~ "Economie et gestion")) |>
    select(- For),
  Budget2 |> filter(!is.na(FORMATION)))

```

b. PARTICULARIERS

Concentrons-nous ensuite sur la variable PARTICULARIERS.

```

# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(Par = case_when(
    PARTICULARIERS == "Non" ~ 0,
    PARTICULARIERS == "Oui" ~ 1
  ))
  set.seed(123)

```

```

# Entraînement sur 80 % des données non manquantes.

particuliers_entrainement <- Budget2 |>
  filter(!is.na(Par)) |>
  slice_sample(prop = 0.8)

# Vérification sur les 20 % des données non manquantes restantes

particuliers_verification <- anti_join(Budget2, particuliers_entrainement) |>
  filter(!is.na(Par))

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS,
TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF,
TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET,
prediction2, For, Par)`
```

A présent, nous pouvons passer à la régression logistique binaire.

```

# Création du modèle de régression logistique binaire

regression_logistique2 <- glm(
  Par ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = particuliers_entrainement,
  family = binomial
)

# Test d'Anova pour le modèle

car::Anova(regression_logistique2)
```

Analysis of Deviance Table (Type II tests)

```

Response: Par
      LR Chisq Df Pr(>Chisq)
AGE      0.1338  1   0.71457
ASSIDUITE 4.5405  1   0.03310 *
MOYENNE  0.9237  1   0.33650
RESTAURANT 5.0160  1   0.02511 *
```

```

SOMMEIL      3.8076  1    0.05102 .
STRESS       0.0012  1    0.97185
TRAJET        3.6359  1    0.05655 .

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Nous enlevons :

- STRESS

```

# Modèle le plus performant

regression_logistique3 <- glm(
  Par ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + TRAJET,
  data = particuliers_entrainement,
  family = binomial
)

# Test d'Anova pour ce modèle

car::Anova(regression_logistique3)

```

Analysis of Deviance Table (Type II tests)

Response: Par

	LR	Chisq	Df	Pr(>Chisq)
AGE	0.1789	1		0.67234
ASSIDUITE	5.3327	1		0.02093 *
MOYENNE	0.2645	1		0.60703
RESTAURANT	4.9154	1		0.02662 *
SOMMEIL	2.6896	1		0.10101
TRAJET	4.1289	1		0.04216 *

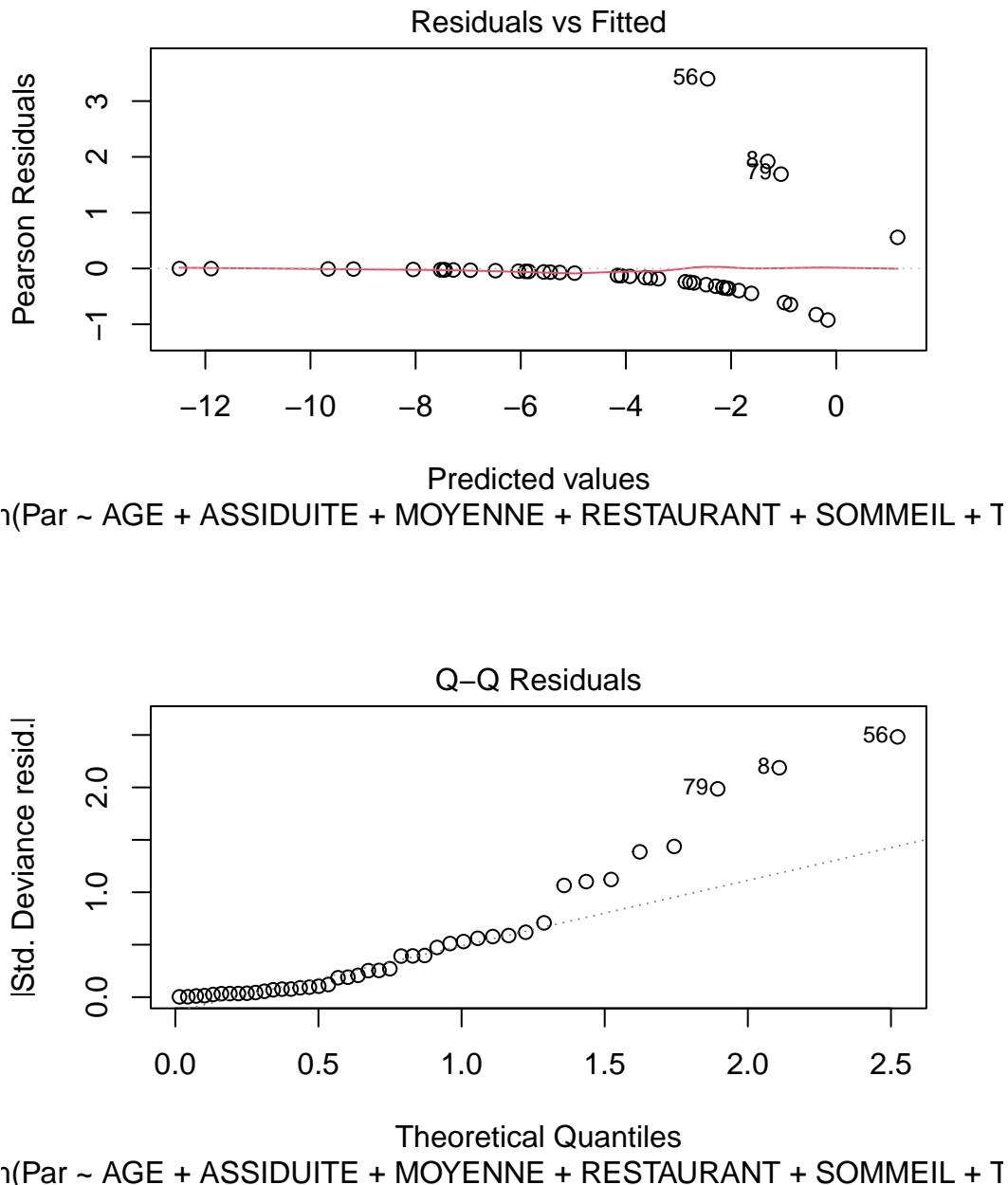
```

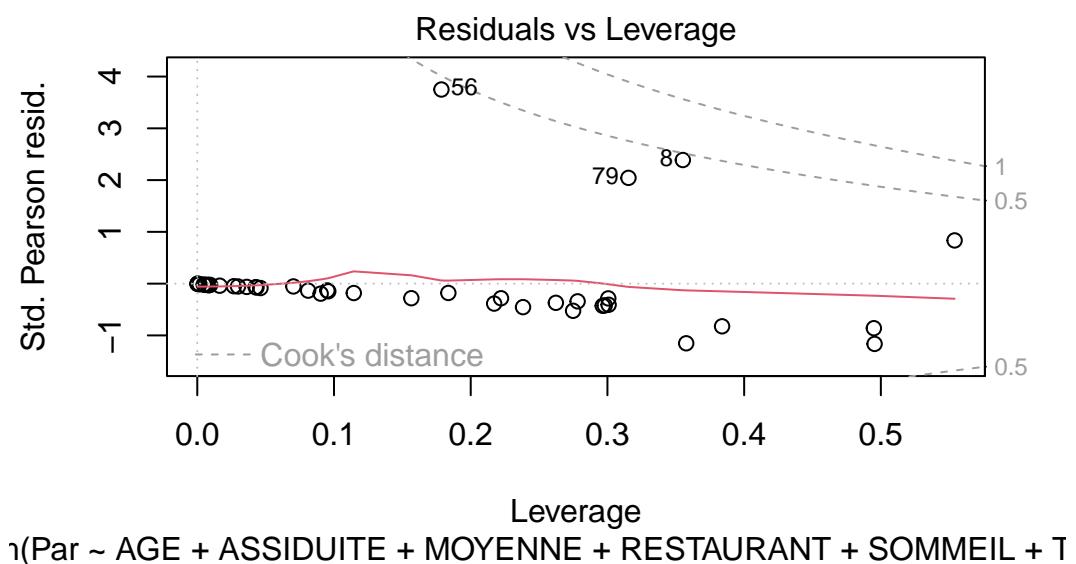
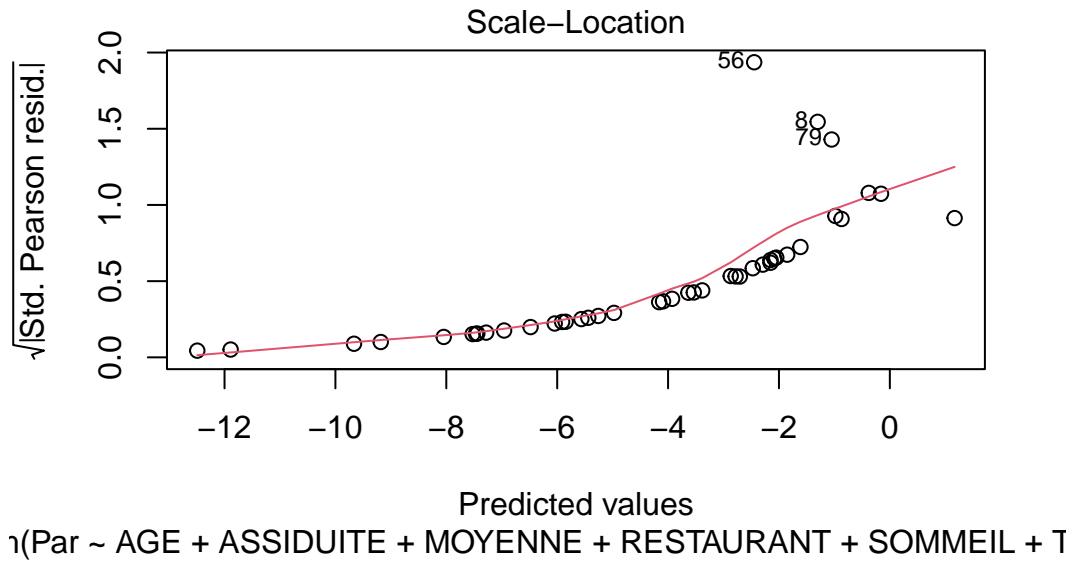
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Nous avons 3 variables significatives au seuil de 10 % : ASSIDUITE, SOMMEIL et TRAJET.

```
# Graphiques évaluant la qualité et la validité du modèle
plot(regression_logistique3)
```





```
# Prédition du modèle sur les 20 % des données non manquantes restantes

prediction3 = predict(regression_logistique3,
                      newdata = particuliers_verification)
```

```

# Ajout de la prédition parmi les 20 % de données en attribuant à chaque observation

particuliers_verification <- particuliers_verification |>
  mutate(prediction3) |>
  mutate(Par_predi = case_when(prediction3 < 0 ~ "Non",
                               TRUE ~ "Oui"))

# Comparaison des prédictions du modèle avec les vraies valeurs de la variable PARTICULARIERS

particuliers_verification |>
  count(PARTICULIERS, Par_predi)

# A tibble: 3 x 3
#   PARTICULARIERS Par_predi     n
#   <fct>          <chr>      <int>
# 1 Oui            Non         1
# 2 Non            Non         8
# 3 Non            Oui        14

# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction4 = predict(regression_logistique3,
                      newdata = filter(Budget2, is.na(PARTICULIERS)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(PARTICULIERS)) |>
    mutate(prediction4) |>
    mutate(PARTICULIERS = case_when(prediction4 < 0 ~ "Non",
                                    TRUE ~ "Oui")) |>
    select(- Par),
  Budget2 |> filter(!is.na(PARTICULIERS)))

```

c. TUTORAT

Concentrons-nous ensuite sur la variable TUTORAT.

```

# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(Tut = case_when(
    TUTORAT == "Non" ~ 0,
    TUTORAT == "Oui" ~ 1
  ))
  
set.seed(123)

# Entraînement sur 80 % des données non manquantes.

tutorat_entrainement <- Budget2 |>
  filter(!is.na(Tut)) |>
  slice_sample(prop = 0.8)

# Vérification sur les 20 % des données non manquantes restantes

tutorat_verification <- anti_join(Budget2, tutorat_entrainement) |>
  filter(!is.na(Tut))

```

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS, TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF, TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET, prediction2, For, prediction4, Par, Tut)`

A présent, nous pouvons passer à la régression logistique binaire.

```

# Création du modèle de régression logistique binaire

regression_logistique4 <- glm(
  Tut ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = tutorat_entrainement,
  family = binomial
)

# Test d'Anova pour le modèle

car::Anova(regression_logistique4)

```

Analysis of Deviance Table (Type II tests)

Response: Tut

	LR	Chisq	Df	Pr(>Chisq)
AGE	3.9752	1	0.04618	*
ASSIDUITE	0.2769	1	0.59875	
MOYENNE	0.3478	1	0.55537	
RESTAURANT	0.7638	1	0.38215	
SOMMEIL	1.3467	1	0.24586	
STRESS	1.0750	1	0.29981	
TRAJET	0.3363	1	0.56200	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

Nous enlevons :

- RESTAURANT
- MOYENNE
- ASSIDUITE
- TRAJET

```
# Modèle le plus performant

regression_logistique5 <- glm(
  Tut ~ AGE + SOMMEIL + STRESS,
  data = tutorat_entrainement,
  family = binomial
)

# Test d'Anova pour ce modèle

car::Anova(regression_logistique5)
```

Analysis of Deviance Table (Type II tests)

Response: Tut

	LR	Chisq	Df	Pr(>Chisq)
AGE	7.7910	1	0.005251	**

```

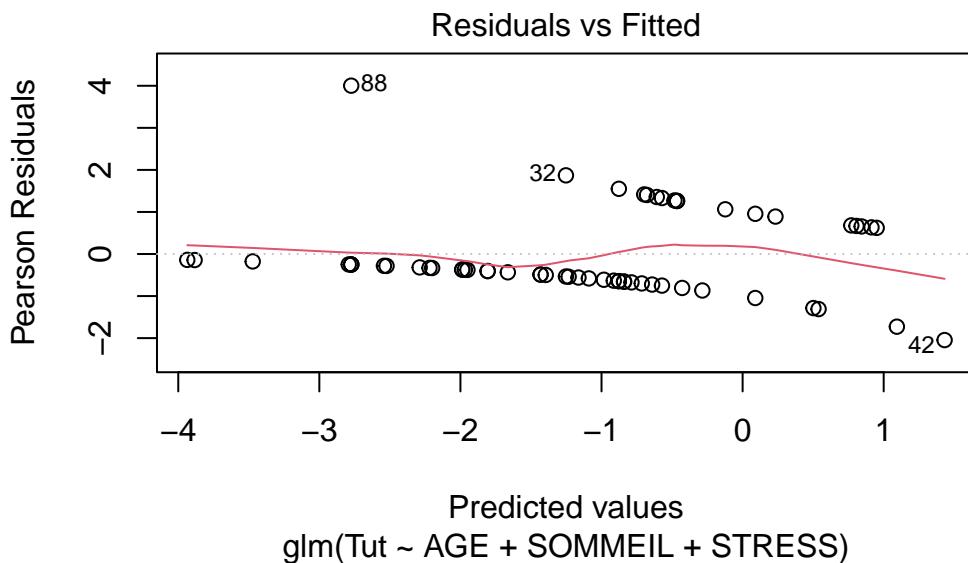
SOMMEIL    2.7938   1    0.094627 .
STRESS     6.6002   1    0.010197 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

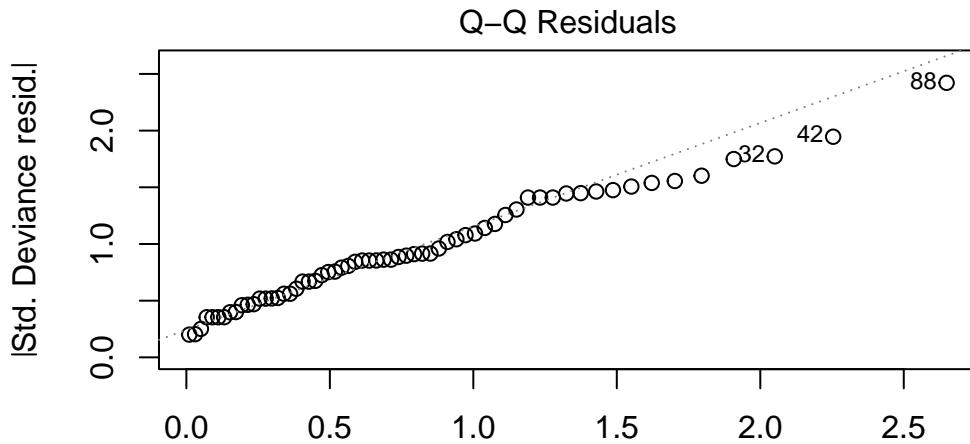
```

Nous avons 3 variables significatives : la variable AGE au seuil de 1 %, STRESS au seuil de 5 % et SOMMEIL au seuil de 10 %.

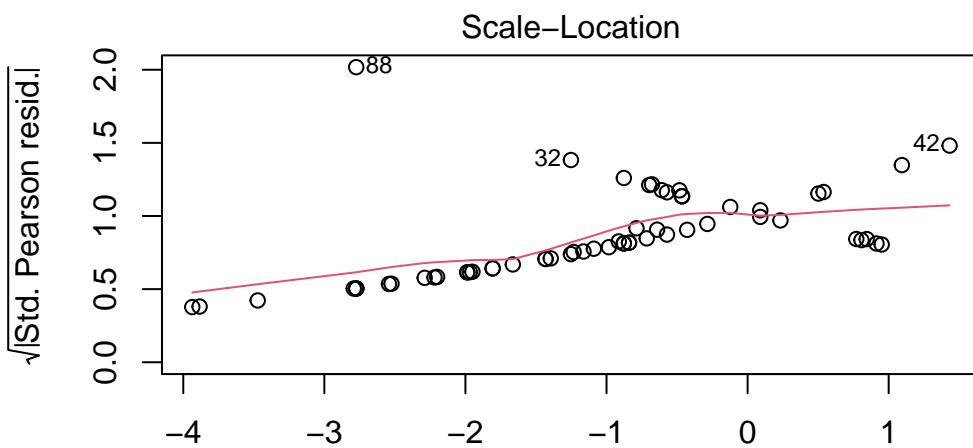
```
# Graphiques évaluant la qualité et la validité du modèle
```

```
plot(regression_logistique5)
```

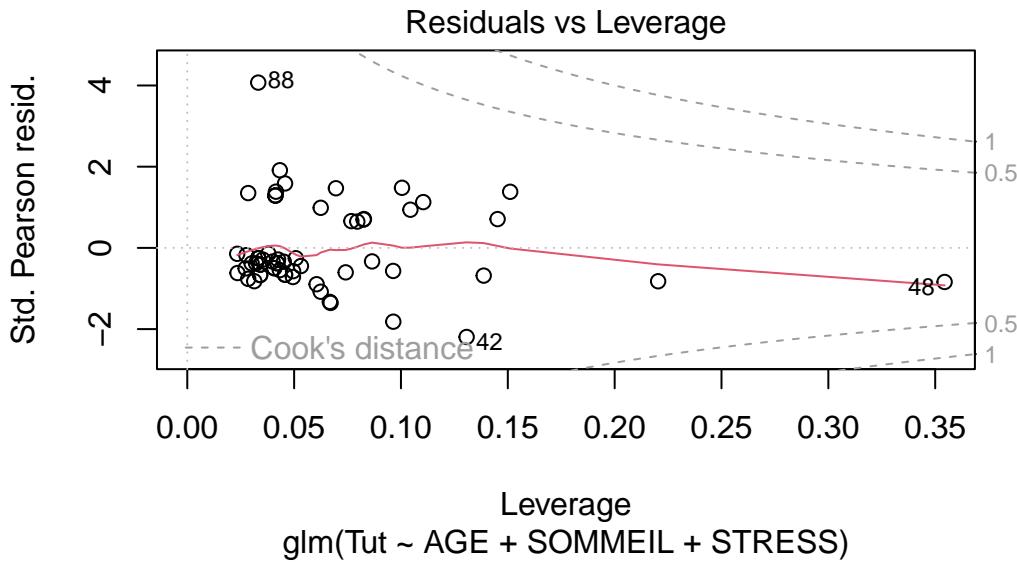




Theoretical Quantiles
glm(Tut ~ AGE + SOMMEIL + STRESS)



Predicted values
glm(Tut ~ AGE + SOMMEIL + STRESS)



```
# Prédition du modèle sur les 20 % des données non manquantes restantes

prediction5 = predict(regression_logistique5,
                      newdata = tutorat_verification)

# Ajout de la prédition parmi les 20 % de données en attribuant à chaque observation

tutorat_verification <- tutorat_verification |>
  mutate(prediction5) |>
  mutate(Tut_predi = case_when(prediction5 < 0 ~ "Non",
                               TRUE ~ "Oui"))

# Comparaison des prédictions du modèle avec les vraies valeurs de la variable TUTORAT

tutorat_verification |>
  count(TUTORAT, Tut_predi)

# A tibble: 4 x 3
  TUTORAT Tut_predi     n
  <fct>    <chr>     <int>
1 Oui      Non          4
```

2	Oui	Oui	1
3	Non	Non	11
4	Non	Oui	7

```
# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction6 = predict(regression_logistique5,
                      newdata = filter(Budget2, is.na(TUTORAT)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(TUTORAT)) |>
    mutate(prediction6) |>
    mutate(TUTORAT = case_when(prediction6 < 0 ~ "Non",
                               TRUE ~ "Oui")) |>
    select(- Tut),
  Budget2 |> filter(!is.na(TUTORAT)))
```

d. BOURSE

Concentrons-nous ensuite sur la variable BOURSE.

```
# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(Bou = case_when(
    TUTORAT == "Non" ~ 0,
    TUTORAT == "Oui" ~ 1
  ))
  set.seed(123)

# Entraînement sur 80 % des données non manquantes.

bourse_entrainement <- Budget2 |>
  filter(!is.na(Bou)) |>
  slice_sample(prop = 0.8)
```

```
# Vérification sur les 20 % des données non manquantes restantes

bourse_verification <- anti_join(Budget2, bourse_entrainement) |>
  filter(!is.na(Bou))
```

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS, TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF, TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET, prediction2, For, prediction4, Par, prediction6, Tut, Bou)`

A présent, nous pouvons passer à la régression logistique binaire.

```
# Création du modèle de régression logistique binaire

regression_logistique6 <- glm(
  Bou ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = bourse_entrainement,
  family = binomial
)

# Test d'Anova pour le modèle

car::Anova(regression_logistique6)
```

Analysis of Deviance Table (Type II tests)

Response: Bou

	LR	Chisq	Df	Pr(>Chisq)
AGE	5.0880	1		0.02409 *
ASSIDUITE	0.0000	1		0.99868
MOYENNE	0.1743	1		0.67632
RESTAURANT	0.0150	1		0.90262
SOMMEIL	3.8047	1		0.05111 .
STRESS	3.4979	1		0.06145 .
TRAJET	0.1586	1		0.69045

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

Nous avons décider d'enlever les variables suivantes :

- ASSIDUITE
- RESTAURANT
- STRESS

```
# Modèle le plus performant

regression_logistique7 <- glm(
  Bou ~ AGE + MOYENNE + SOMMEIL + TRAJET,
  data = bourse_entrainement,
  family = binomial
)

# Test d'Anova pour ce modèle

car::Anova(regression_logistique7)
```

Analysis of Deviance Table (Type II tests)

Response: Bou

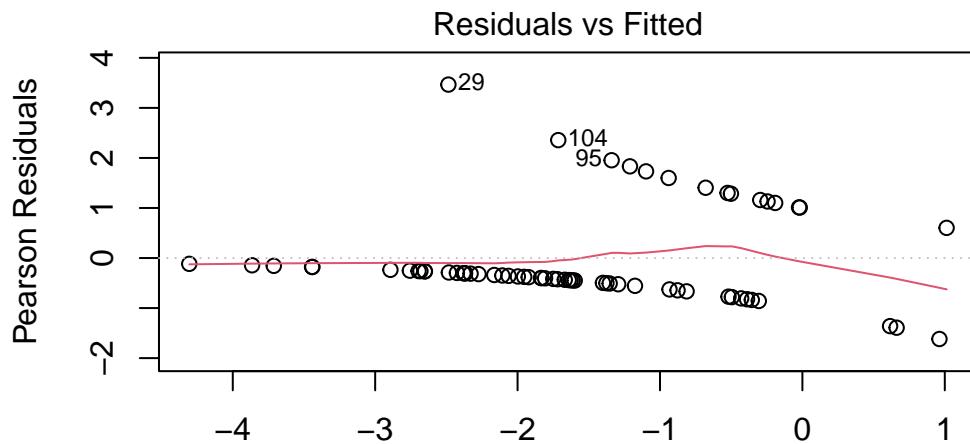
	LR	Chisq	Df	Pr(>Chisq)							
AGE	7.5296	1	0.006069	**							
MOYENNE	2.9893	1	0.083817	.							
SOMMEIL	1.3790	1	0.240276								
TRAJET	1.4656	1	0.226044								

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

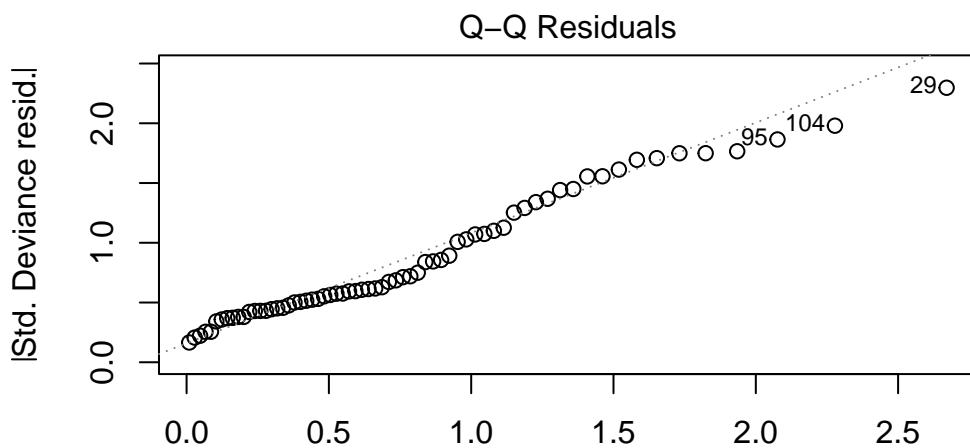
Les variables AGE et MOYENNE sont significatives au seuil de 5 % et la variable TRAJET est significative au seuil de 10 %.

```
# Graphiques évaluant la qualité et la validité du modèle

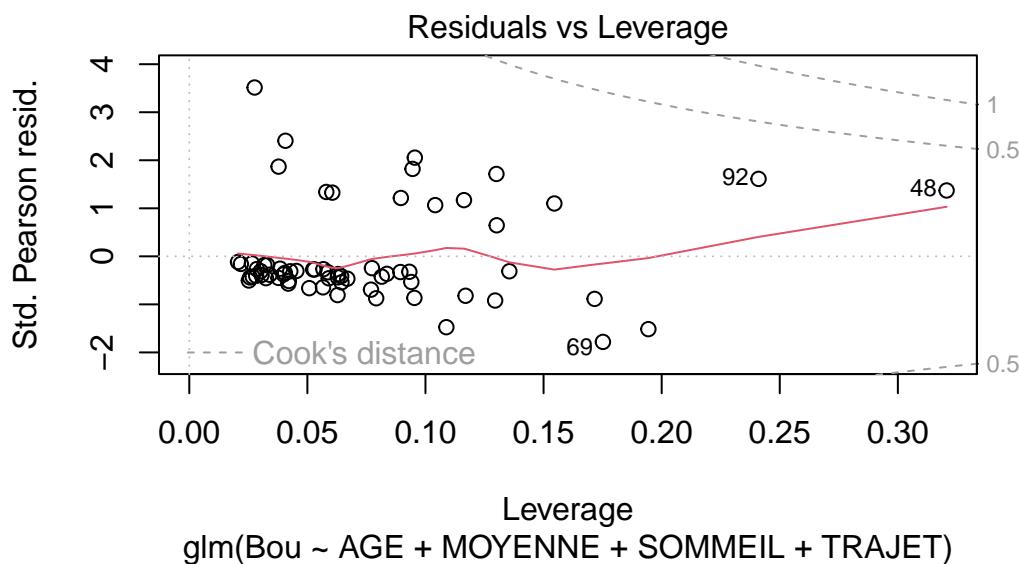
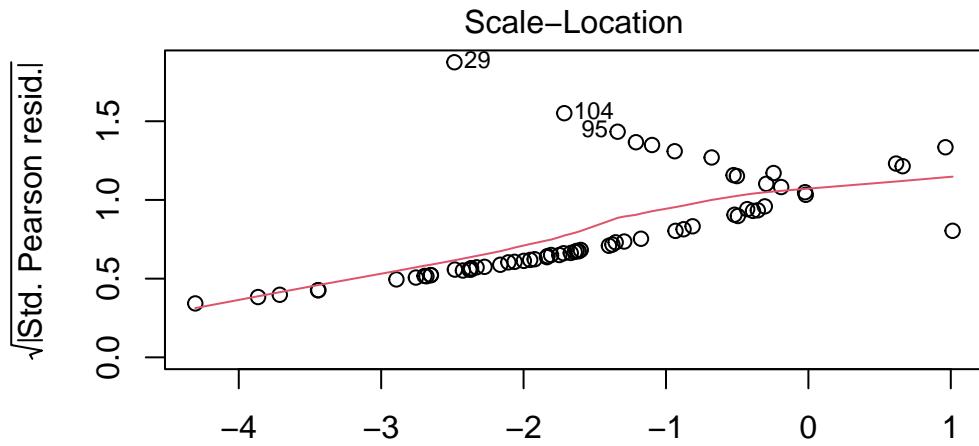
plot(regression_logistique7)
```



Predicted values
glm(Bou ~ AGE + MOYENNE + SOMMEIL + TRAJET)



Theoretical Quantiles
glm(Bou ~ AGE + MOYENNE + SOMMEIL + TRAJET)



```
# Prédition du modèle sur les 20 % des données non manquantes restantes

prediction7 = predict(regression_logistique7,
                      newdata = bourse_verification)
```

```

# Ajout de la prédition parmi les 20 % de données en attribuant à chaque observation

bourse_verification <- bourse_verification |>
  mutate(prediction7) |>
  mutate(Bou_predi = case_when(prediction7 < 0 ~ "Non",
                               TRUE ~ "Oui"))

# Comparaison des prédictions du modèle avec les vraies valeurs de la variable BOURSE

bourse_verification |>
  count(BOURSE, Bou_predi)

# A tibble: 6 x 3
#   BOURSE Bou_predi     n
#   <fct>  <chr>     <int>
# 1 Non      Non         8
# 2 Non      Oui        10
# 3 Oui      Non         2
# 4 Oui      Oui         5
# 5 <NA>     Non         1
# 6 <NA>     Oui         1

# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction8 = predict(regression_logistique7,
                      newdata = filter(Budget2, is.na(BOURSE)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(BOURSE)) |>
    mutate(prediction8) |>
    mutate(BOURSE = case_when(prediction8 < 0 ~ "Non",
                               TRUE ~ "Oui")) |>
    select(- Bou),
  Budget2 |> filter(!is.na(BOURSE)))

```

e. EMPLOI

Concentrons-nous ensuite sur la variable EMPLOI.

```
# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(Emp = case_when(
    EMPLOI == "Non" ~ 0,
    EMPLOI == "Oui" ~ 1
  ))

set.seed(123)

# Entraînement sur 80 % des données non manquantes.

emploi_entrainement <- Budget2 |>
  filter(!is.na(Emp)) |>
  slice_sample(prop = 0.8)

# Vérification sur les 20 % des données non manquantes restantes

emploi_verification <- anti_join(Budget2, emploi_entrainement) |>
  filter(!is.na(Emp))
```

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS, TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF, TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET, prediction2, For, prediction4, Par, prediction6, Tut, prediction8, Bou, Emp)`

A présent, nous pouvons passer à la régression logistique binaire.

```
# Créeation du modèle de régression logistique binaire

regression_logistique10 <- glm(
  Emp ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = emploi_entrainement,
  family = binomial
)
```

```
# Test d'Anova pour le modèle  
car::Anova(regression_logistique10)
```

Analysis of Deviance Table (Type II tests)

Response: Emp

	LR	Chisq	Df	Pr(>Chisq)
AGE	3.5087	1		0.06105 .
ASSIDUITE	2.0813	1		0.14911
MOYENNE	5.2170	1		0.02237 *
RESTAURANT	5.5217	1		0.01878 *
SOMMEIL	0.0225	1		0.88069
STRESS	4.0598	1		0.04391 *
TRAJET	5.2475	1		0.02198 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

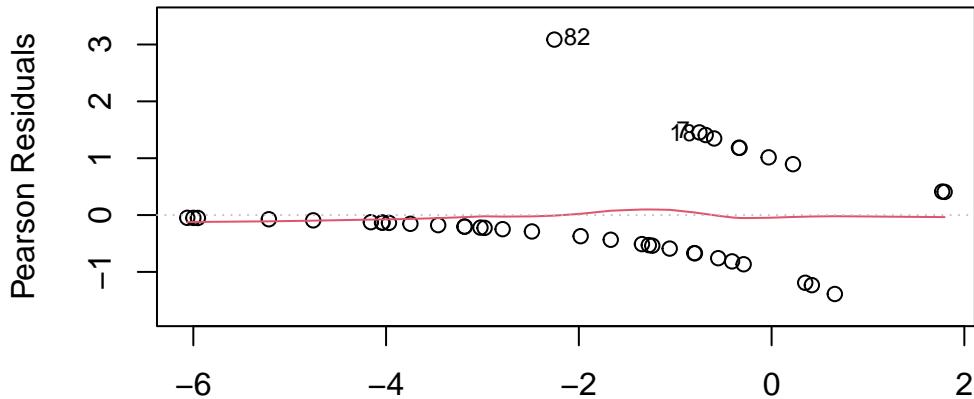
Les variables RESTAURANT et STRESS sont significatives au seuil de 5 %, et la variable TRAJET au seuil de 10 %.

En essayant de supprimer les variables dont les p-values étaient les plus élevées, notre modèle devenait moins pertinent. Nous gardons donc notre modèle initial.

```
# Graphiques évaluant la qualité et la validité du modèle
```

```
plot(regression_logistique10)
```

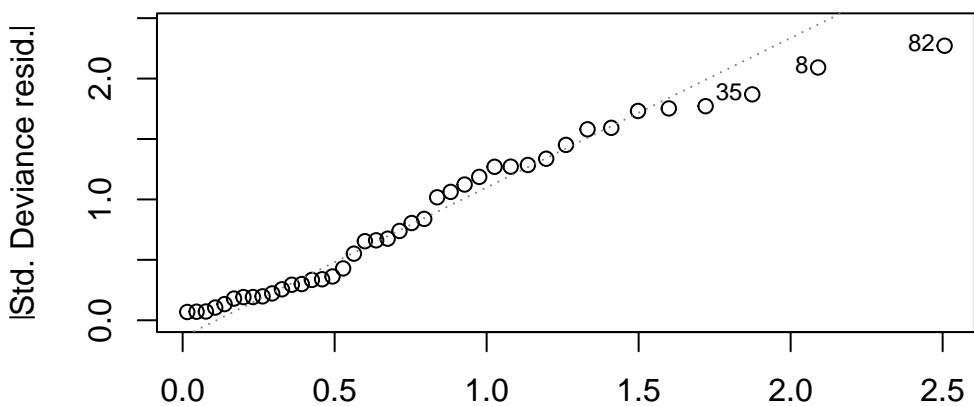
Residuals vs Fitted



Predicted values

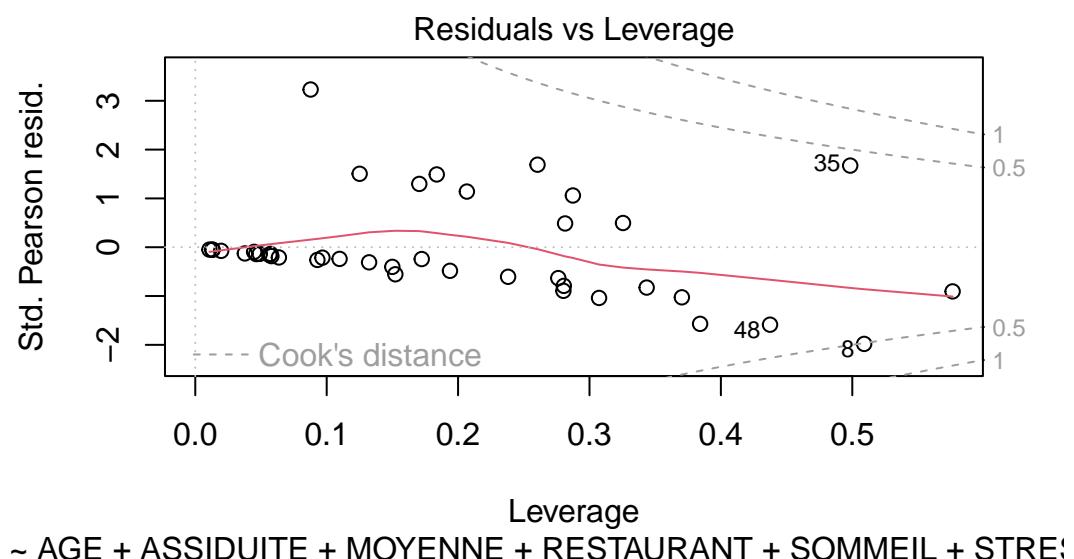
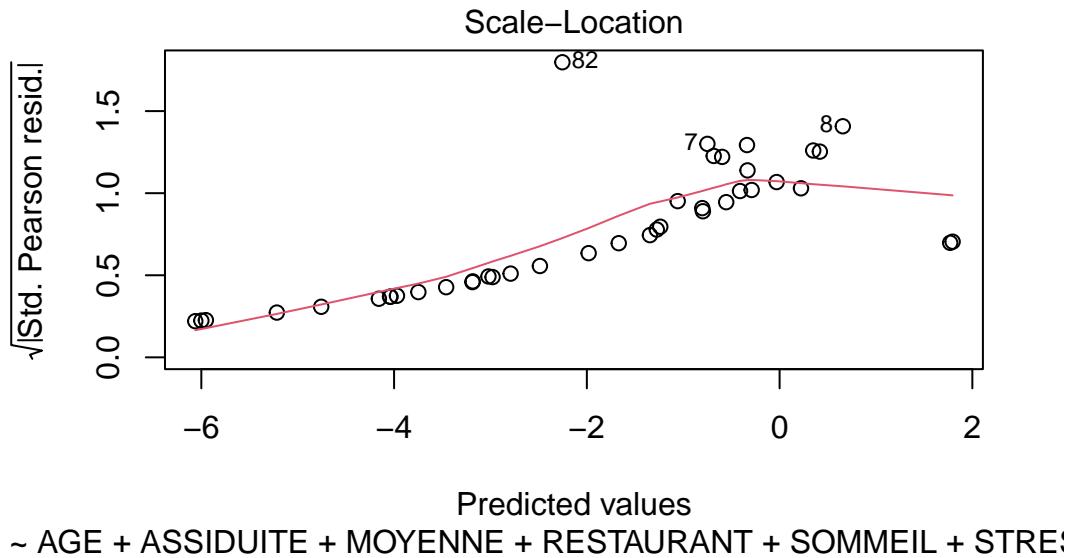
~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS

Q-Q Residuals



Theoretical Quantiles

~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS



```
# Prédition du modèle sur les 20 % des données non manquantes restantes

prediction11 = predict(regression_logistique10,
                      newdata = emploi_verification)
```

```

# Ajout de la prédition parmi les 20 % de données en attribuant à chaque observation

emploi_verification <- emploi_verification |>
  mutate(prediction11) |>
  mutate(Emp_predi = case_when(prediction11 < 0 ~ "Non",
                               TRUE ~ "Oui"))

# Comparaison des prédictions du modèle avec les vraies valeurs de la variable EMPLOI

emploi_verification |>
  count(EMPLOI, Emp_predi)

# A tibble: 4 x 3
#>   EMPLOI Emp_predi     n
#>   <fct>   <chr>    <int>
#> 1 Non      Non        9
#> 2 Non      Oui        8
#> 3 Oui      Non        4
#> 4 Oui      Oui        4

# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction12 = predict(regression_logistique10,
                      newdata = filter(Budget2, is.na(EMPLOI)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(EMPLOI)) |>
    mutate(prediction12) |>
    mutate(TUTORAT = case_when(prediction12 < 0 ~ "Non",
                               TRUE ~ "Oui")) |>
    select(- Emp),
  Budget2 |> filter(!is.na(EMPLOI)))

```

f. LOGEMENT

Concentrons-nous ensuite sur la variable LOGEMENT.

```

# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(Log = case_when(
    LOGEMENT == "Non" ~ 0,
    LOGEMENT == "Oui" ~ 1
  ))
  
set.seed(123)

# Entraînement sur 80 % des données non manquantes.

logement_entrainement <- Budget2 |>
  filter(!is.na(Log)) |>
  slice_sample(prop = 0.8)

# Vérification sur les 20 % des données non manquantes restantes

logement_verification <- anti_join(Budget2, logement_entrainement) |>
  filter(!is.na(Log))

```

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS, TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF, TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET, prediction2, For, prediction4, Par, prediction6, Tut, prediction8, Bou, prediction12, Emp, Log)`

A présent, nous pouvons passer à la régression logistique binaire.

```

# Création du modèle de régression logistique binaire

regression_logistique8 <- glm(
  Log ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = logement_entrainement,
  family = binomial
)

# Test d'Anova pour le modèle

car::Anova(regression_logistique8)

```

```
Analysis of Deviance Table (Type II tests)
```

```
Response: Log
```

	LR	Chisq	Df	Pr(>Chisq)
AGE	0.1798	1	0.6715333	
ASSIDUITE	0.2015	1	0.6535187	
MOYENNE	0.7770	1	0.3780706	
RESTAURANT	0.1206	1	0.7283709	
SOMMEIL	1.7983	1	0.1799122	
STRESS	7.6679	1	0.0056211	**
TRAJET	12.4779	1	0.0004118	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

Nous enlevons :

- AGE
- ASSIDUITE

```
# Modèle le plus performant

regression_logistique9 <- glm(
  Log ~ MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = logement_entrainement,
  family = binomial
)

# Test d'Anova pour ce modèle

car::Anova(regression_logistique9)
```

```
Analysis of Deviance Table (Type II tests)
```

```
Response: Log
```

	LR	Chisq	Df	Pr(>Chisq)
MOYENNE	0.0065	1	0.9359306	
RESTAURANT	1.2823	1	0.2574804	
SOMMEIL	1.9199	1	0.1658648	
STRESS	4.8267	1	0.0280224	*

```

TRAJET      11.5952  1  0.0006612 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

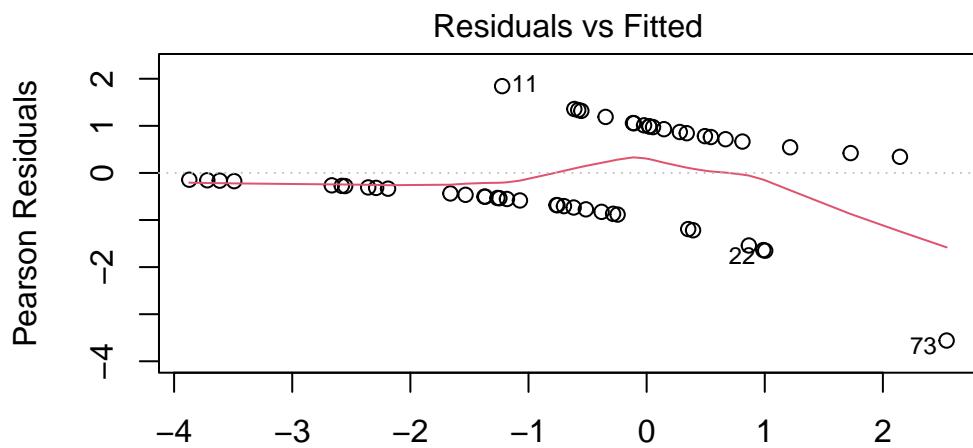
```

Nous obtenons trois variables significatives : SOMMEIL au seuil de 10 %, STRESS au seuil de 5 % et TRAJET au seuil de 1 %.

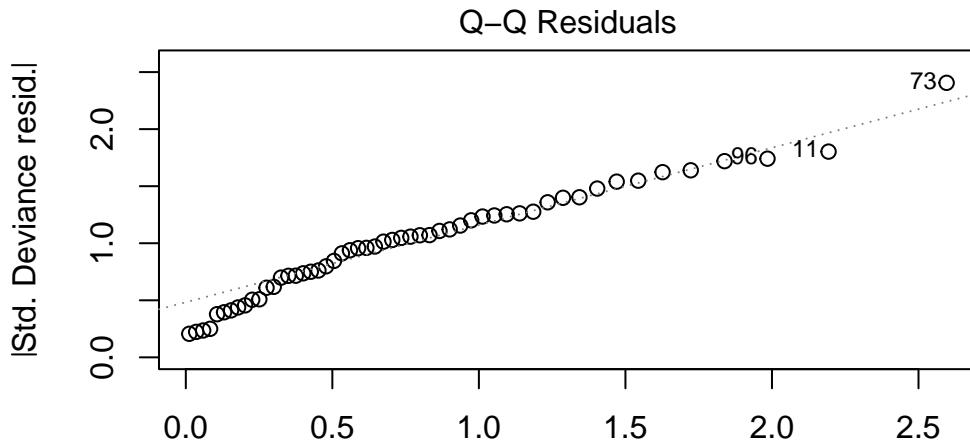
```

# Graphiques évaluant la qualité et la validité du modèle
plot(regression_logistique9)

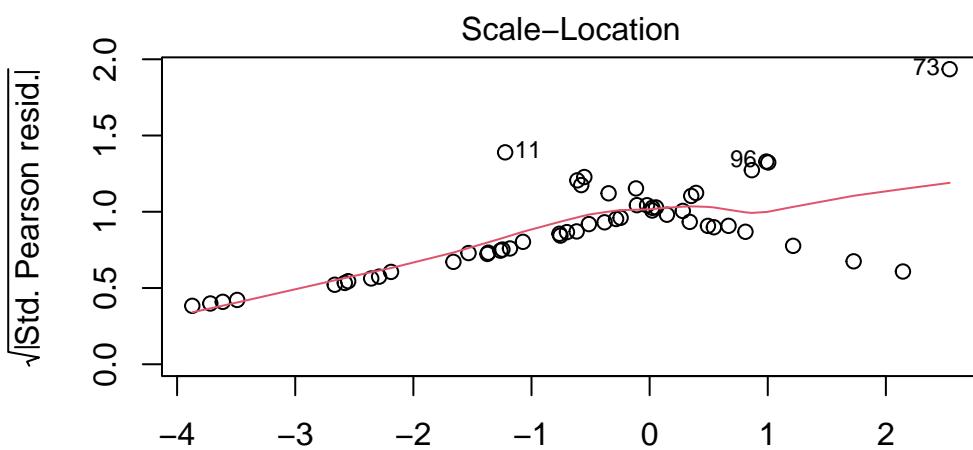
```



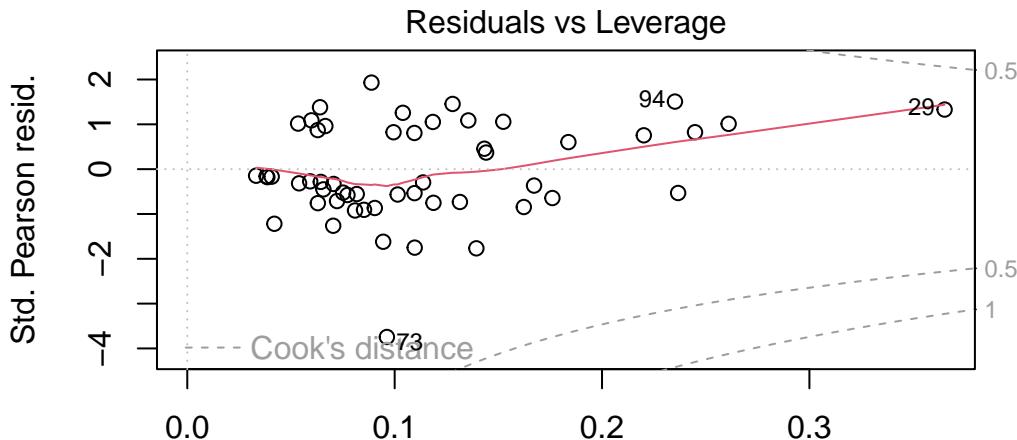
Predicted values
 $\text{glm}(\text{Log} \sim \text{MOYENNE} + \text{RESTAURANT} + \text{SOMMEIL} + \text{STRESS} + \text{TRAJ})$



Theoretical Quantiles
glm(Log ~ MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJ)



Predicted values
glm(Log ~ MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJ)



Leverage
 $\text{glm}(\text{Log} \sim \text{MOYENNE} + \text{RESTAURANT} + \text{SOMMEIL} + \text{STRESS} + \text{TRAJ|}$

```
# Prédition du modèle sur les 20 % des données non manquantes restantes

prediction9 = predict(regression_logistique9,
                      newdata = logement_verification)

# Ajout de la prédition parmi les 20 % de données en attribuant à chaque observation

logement_verification <- logement_verification |>
  mutate(prediction9) |>
  mutate(Log_predi = case_when(prediction9 < 0 ~ "Non",
                                TRUE ~ "Oui"))
```

```
# Comparaison des prédictions du modèle avec les vraies valeurs de la variable LOGEMENT

logement_verification |>
  count(LOGEMENT, Log_predi)
```

```
# A tibble: 4 x 3
  LOGEMENT Log_predi     n
  <fct>    <chr>      <int>
  1 Non      Non          4
```

2	Non	Oui	8
3	Oui	Non	2
4	Oui	Oui	11

```
# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction10 = predict(regression_logistique9,
                       newdata = filter(Budget2, is.na(LOGEMENT)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(LOGEMENT)) |>
    mutate(prediction10) |>
    mutate(LOGEMENT = case_when(prediction10 < 0 ~ "Non",
                                 TRUE ~ "Oui")) |>
    select(- Log),
  Budget2 |> filter(!is.na(LOGEMENT)))
```

g. ARGENT

Concentrons-nous ensuite sur la variable ARGENT.

```
# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(Arg = case_when(
    ARGENT == "Non" ~ 0,
    ARGENT == "Oui" ~ 1
  ))

set.seed(123)

# Entraînement sur 80 % des données non manquantes.

argent_entrainement <- Budget2 |>
  filter(!is.na(Arg)) |>
  slice_sample(prop = 0.8)
```

```
# Vérification sur les 20 % des données non manquantes restantes

argent_verification <- anti_join(Budget2, argent_entrainement) |>
  filter(!is.na(Arg))
```

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS, TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF, TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET, prediction2, For, prediction4, Par, prediction6, Tut, prediction8, Bou, prediction12, Emp, prediction10, Log, Arg)`

A présent, nous pouvons passer à la régression logistique binaire.

```
# Création du modèle de régression logistique binaire

regression_logistique11 <- glm(
  Arg ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = argent_entrainement,
  family = binomial
)

# Test d'Anova pour le modèle

car::Anova(regression_logistique11)
```

Analysis of Deviance Table (Type II tests)

Response: Arg

	LR	Chisq	Df	Pr(>Chisq)
AGE	0.9791	1	0.32242	
ASSIDUITE	0.2968	1	0.58588	
MOYENNE	1.1386	1	0.28595	
RESTAURANT	0.1398	1	0.70852	
SOMMEIL	1.5541	1	0.21253	
STRESS	3.2487	1	0.07148	.
TRAJET	1.0427	1	0.30718	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

On décide d'enlever :

- ASSIDUITE
- SOMMEIL
- STRESS
- RESTAURANT

```
# Modèle le plus performant

regression_logistique12 <- glm(
  Arg ~ AGE + MOYENNE + TRAJET,
  data = argent_entrainement,
  family = binomial
)

# Test d'Anova pour ce modèle

car::Anova(regression_logistique12)
```

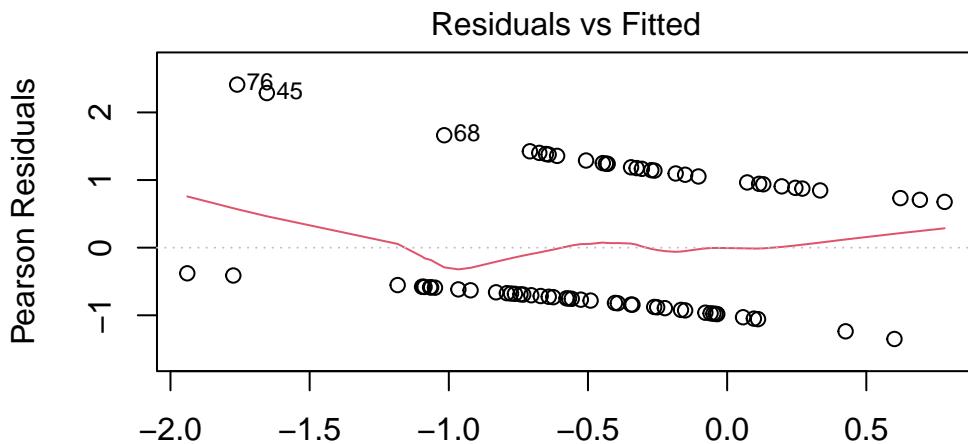
Analysis of Deviance Table (Type II tests)

```
Response: Arg
          LR Chisq Df Pr(>Chisq)
AGE      3.2137  1   0.07302 .
MOYENNE 1.8366  1   0.17535
TRAJET   1.1076  1   0.29260
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

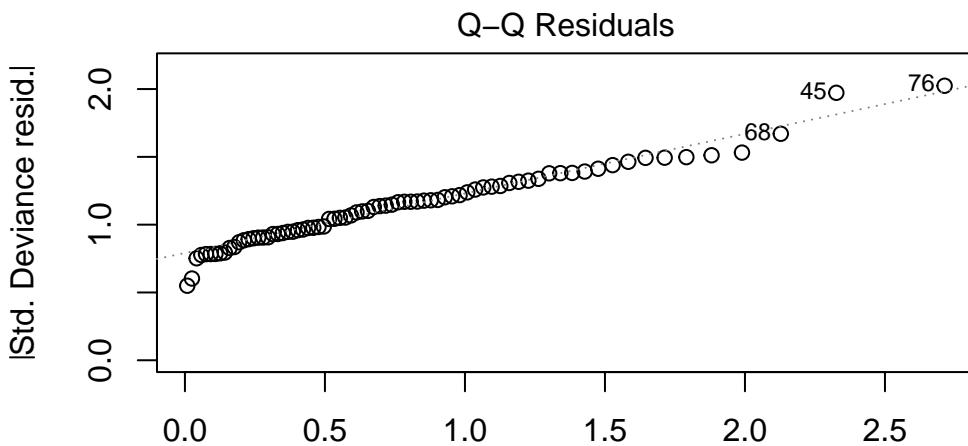
La variable MOYENNE est significative au seuil de 5 % et TRAJET au seuil de 10 %.

```
# Graphiques évaluant la qualité et la validité du modèle

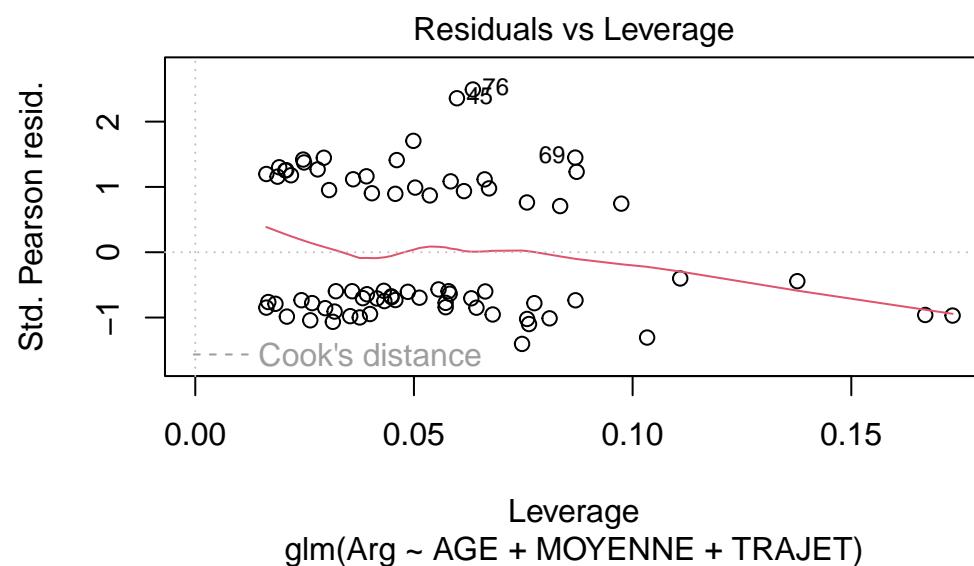
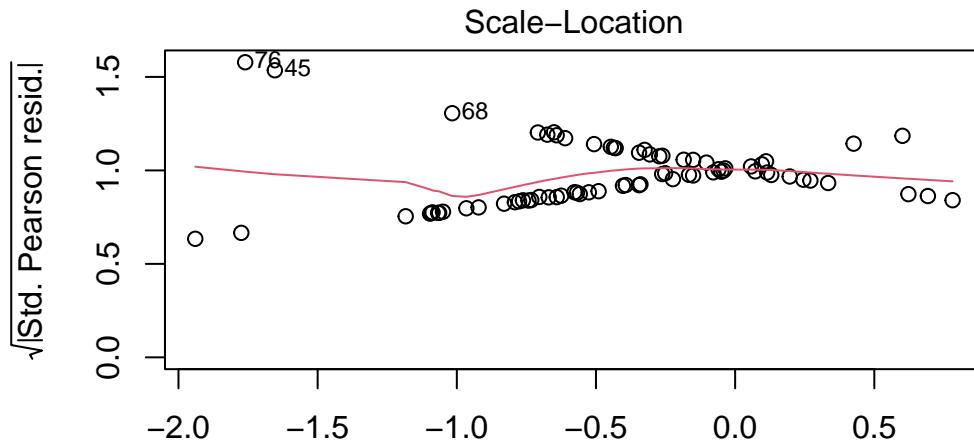
plot(regression_logistique12)
```



Predicted values
 $\text{glm}(\text{Arg} \sim \text{AGE} + \text{MOYENNE} + \text{TRAJET})$



Theoretical Quantiles
 $\text{glm}(\text{Arg} \sim \text{AGE} + \text{MOYENNE} + \text{TRAJET})$



```
# Prédition du modèle sur les 20 % des données non manquantes restantes

prediction13 = predict(regression_logistique12,
                      newdata = argent_verification)
```

```

# Ajout de la prédition parmi les 20 % de données en attribuant à chaque observation

argent_verification <- argent_verification |>
  mutate(prediction13) |>
  mutate(Arg_predi = case_when(prediction13 < 0 ~ "Non",
                               TRUE ~ "Oui"))

# Comparaison des prédictions du modèle avec les vraies valeurs de la variable ARGENT

argent_verification |>
  count(ARGENT, Arg_predi)

# A tibble: 4 x 3
#>   ARGENT Arg_predi     n
#>   <fct>   <chr>    <int>
#> 1 Oui      Non        5
#> 2 Oui      Oui        7
#> 3 Non      Non        5
#> 4 Non      Oui        8

# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction14 = predict(regression_logistique12,
                       newdata = filter(Budget2, is.na(ARGENT)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(ARGENT)) |>
    mutate(prediction14) |>
    mutate(ARGENT = case_when(prediction14 < 0 ~ "Non",
                               TRUE ~ "Oui")) |>
    select(- Arg),
  Budget2 |> filter(!is.na(ARGENT)))

```

h. CAF

Concentrons-nous ensuite sur la variable CAF.

```

# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(Caf = case_when(
    CAF == "Non" ~ 0,
    CAF == "Oui" ~ 1
  ))
  
set.seed(123)

# Entraînement sur 80 % des données non manquantes.

caf_entrainement <- Budget2 |>
  filter(!is.na(Caf)) |>
  slice_sample(prop = 0.8)

# Vérification sur les 20 % des données non manquantes restantes

caf_verification <- anti_join(Budget2, caf_entrainement) |>
  filter(!is.na(Caf))

```

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS, TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF, TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET, prediction2, For, prediction4, Par, prediction6, Tut, prediction8, Bou, prediction12, Emp, prediction10, Log, prediction14, Arg, Caf)`

A présent, nous pouvons passer à la régression logistique binaire.

```

# Création du modèle de régression logistique binaire

regression_logistique13 <- glm(
  Caf ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = caf_entrainement,
  family = binomial
)

# Test d'Anova pour le modèle

car::Anova(regression_logistique13)

```

```
Analysis of Deviance Table (Type II tests)
```

```
Response: Caf
      LR Chisq Df Pr(>Chisq)
AGE      2.0913  1   0.148142
ASSIDUITE 0.3855  1   0.534700
MOYENNE  2.0054  1   0.156741
RESTAURANT 0.0296  1   0.863488
SOMMEIL   2.2238  1   0.135898
STRESS    4.8091  1   0.028310 *
TRAJET    8.9286  1   0.002807 **

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nous enlevons les variables suivantes :

- ASSIDUITE
- RESTAURANT
- SOMMEIL
- MOYENNE

```
# Modèle le plus performant

regression_logistique14 <- glm(
  Caf ~ AGE + STRESS + TRAJET,
  data = caf_entrainement,
  family = binomial
)

# Test d'Anova pour ce modèle

car::Anova(regression_logistique14)
```

```
Analysis of Deviance Table (Type II tests)
```

```
Response: Caf
      LR Chisq Df Pr(>Chisq)
AGE      4.1354  1   0.041995 *
```

```

STRESS    8.3796   1   0.003795  **
TRAJET    18.6558   1   1.566e-05 ***

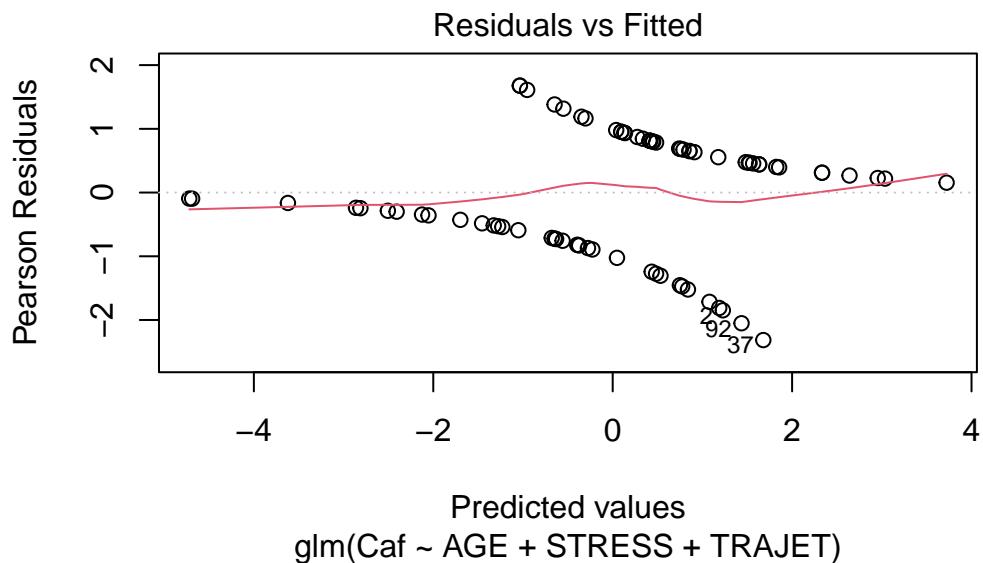
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

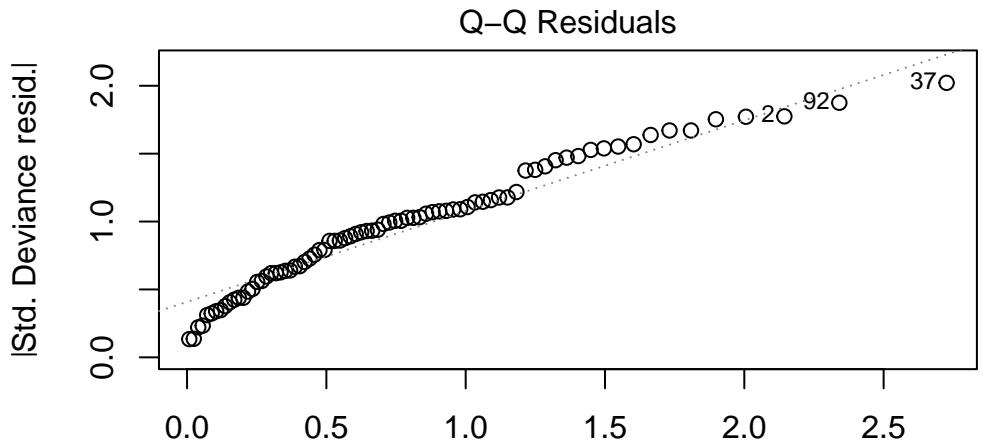
```

Nous obtenons trois variables significatives : TRAJET au seuil de 0.1 %, STRESS au seuil de 1 % et AGE au seuil de 10 %.

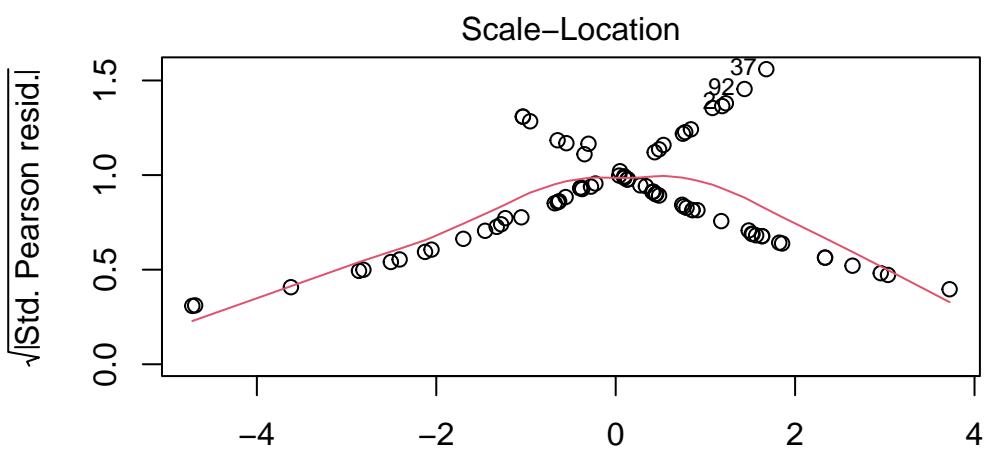
```
# Graphiques évaluant la qualité et la validité du modèle
```

```
plot(regression_logistique14)
```

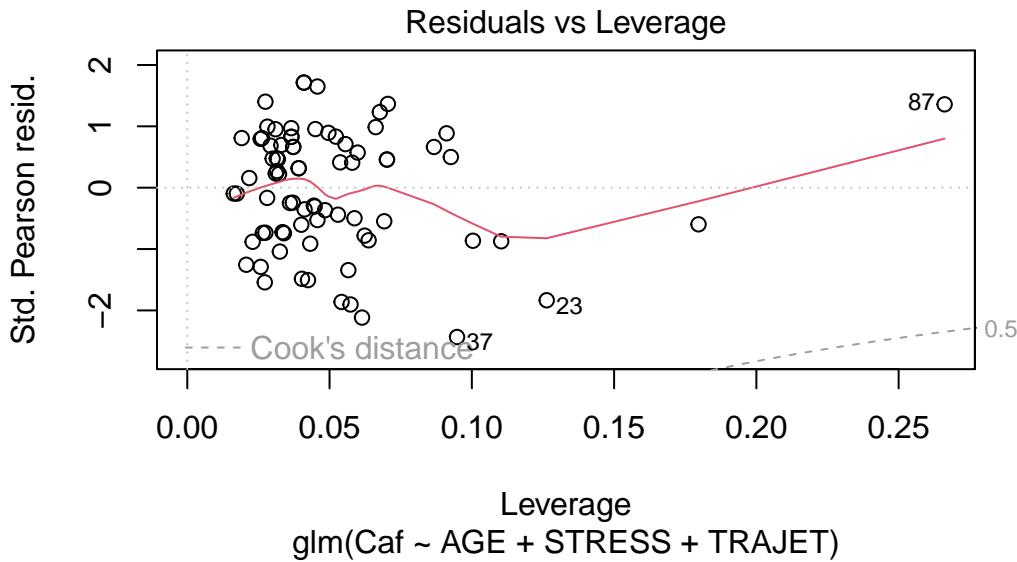




Theoretical Quantiles
glm(Caf ~ AGE + STRESS + TRAJET)



Predicted values
glm(Caf ~ AGE + STRESS + TRAJET)



```
# Prédition du modèle sur les 20 % des données non manquantes restantes

prediction15 = predict(regression_logistique14,
                      newdata = caf_verification)

# Ajout de la prédition parmi les 20 % de données en attribuant à chaque observation

caf_verification <- caf_verification |>
  mutate(prediction15) |>
  mutate(Caf_predi = case_when(prediction15 < 0 ~ "Non",
                               TRUE ~ "Oui"))

# Comparaison des prédictions du modèle avec les vraies valeurs de la variable CAF

caf_verification |>
  count(CAF, Caf_predi)

# A tibble: 4 x 3
  CAF    Caf_predi     n
  <fct> <chr>      <int>
1 Non    Non            7
```

```

2 Non    Oui      5
3 Oui    Non      4
4 Oui    Oui     10

```

```

# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction16 = predict(regression_logistique14,
                       newdata = filter(Budget2, is.na(CAF)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(CAF)) |>
    mutate(prediction16) |>
    mutate(CAF = case_when(prediction16 < 0 ~ "Non",
                           TRUE ~ "Oui")) |>
    select(- Caf),
  Budget2 |> filter(!is.na(CAF)))

```

i. GENRE

Pour terminer, concentrons-nous sur la variable GENRE.

```

# Binarisation de la variable

Budget2 <- Budget2 |>
  mutate(Gen = case_when(
    GENRE == "Femme" ~ 0,
    GENRE == "Homme" ~ 1
  ))

set.seed(123)

# Entraînement sur 80 % des données non manquantes.

genre_entrainement <- Budget2 |>
  filter(!is.na(Gen)) |>
  slice_sample(prop = 0.8)

```

```
# Vérification sur les 20 % des données non manquantes restantes

genre_verification <- anti_join(Budget2, genre_entrainement) |>
  filter(!is.na(Gen))
```

Joining with `by = join_by(FORMATION, ASSIDUITE, REVISIONS, PARTICULIERS, TUTORAT, MOYENNE, BOURSE, EMPLOI, LOGEMENT, ARGENT, RESTAURANT, DEPENSES, CAF, TRANSPORT, GENRE, AGE, STRESS, STATUT, SOMMEIL, SANTE, STRUCTURE, TRAJET, prediction2, For, prediction4, Par, prediction6, Tut, prediction8, Bou, prediction12, Emp, prediction10, Log, prediction14, Arg, prediction16, Caf, Gen)`

A présent, nous pouvons passer à la régression logistique binaire.

```
# Création du modèle de régression logistique binaire

regression_logistique15 <- glm(
  Gen ~ AGE + ASSIDUITE + MOYENNE + RESTAURANT + SOMMEIL + STRESS + TRAJET,
  data = genre_entrainement,
  family = binomial
)

# Test d'Anova pour le modèle

car::Anova(regression_logistique15)
```

Analysis of Deviance Table (Type II tests)

	LR	Chisq	Df	Pr(>Chisq)
AGE	0.14151	1		0.7068
ASSIDUITE	0.99022	1		0.3197
MOYENNE	0.32758	1		0.5671
RESTAURANT	0.02311	1		0.8792
SOMMEIL	0.28623	1		0.5926
STRESS	1.88213	1		0.1701
TRAJET	0.00692	1		0.9337

Nous enlevons :

- AGE
- RESTAURANT

```
# Modèle le plus performant

regression_logistique16 <- glm(
  Gen ~ ASSIDUITE + MOYENNE + SOMMEIL + STRESS + TRAJET,
  data = genre_entrainement,
  family = binomial
)

# Test d'Anova pour ce modèle

car::Anova(regression_logistique16)
```

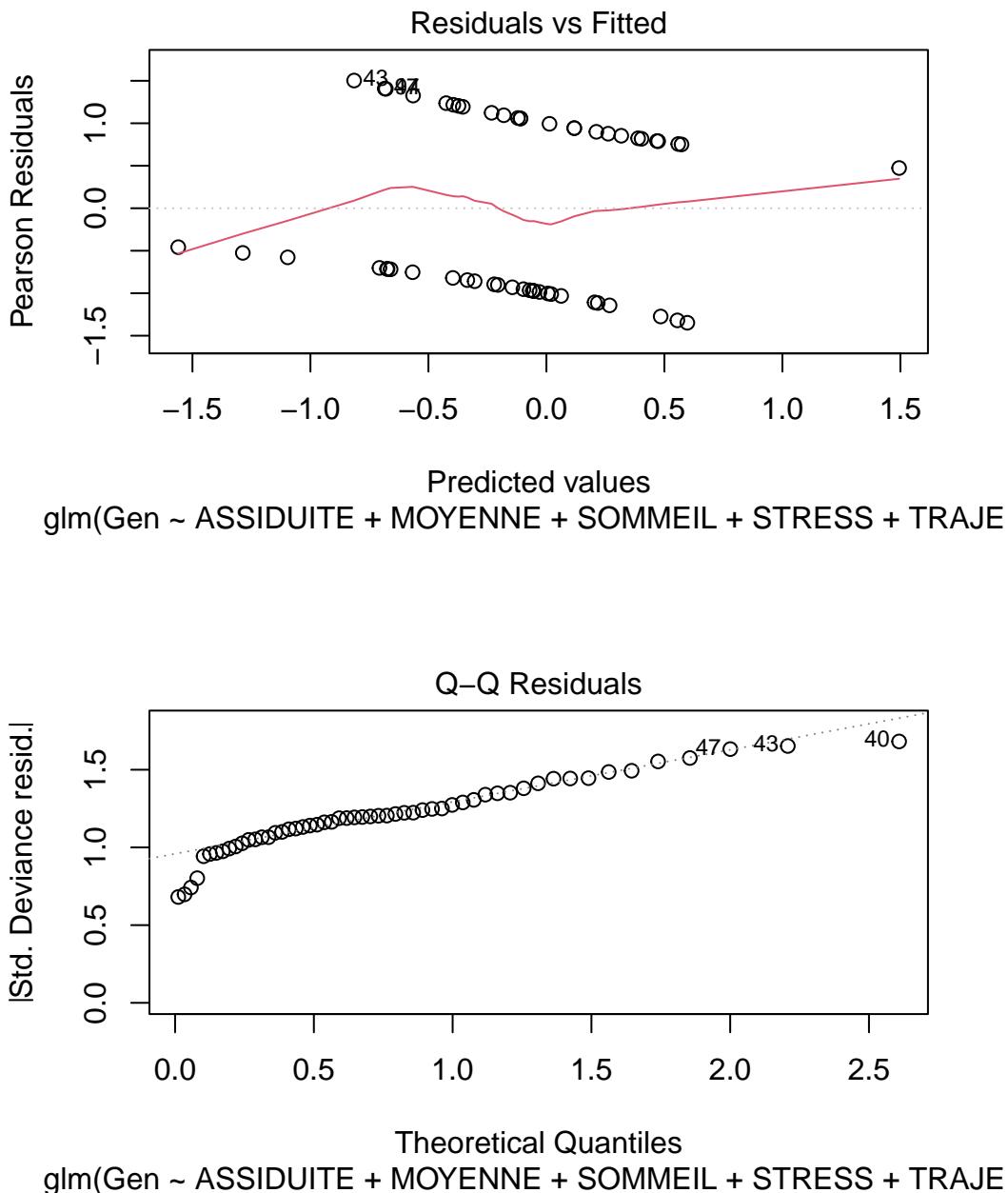
Analysis of Deviance Table (Type II tests)

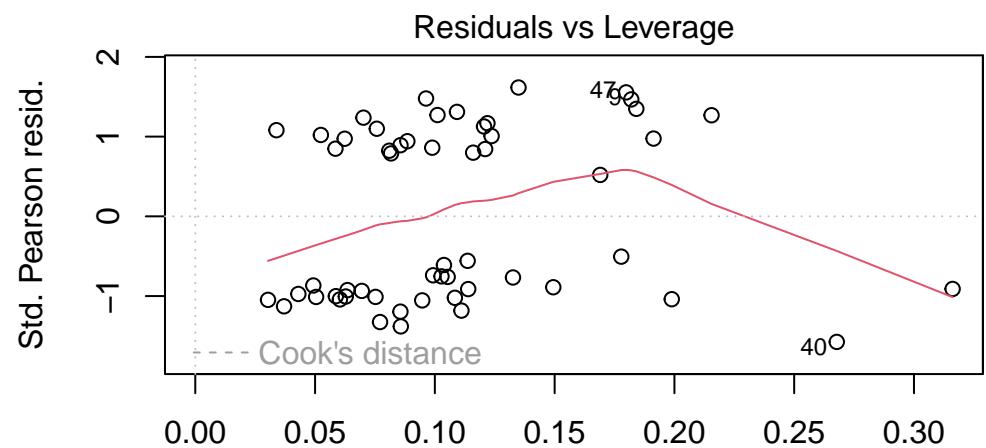
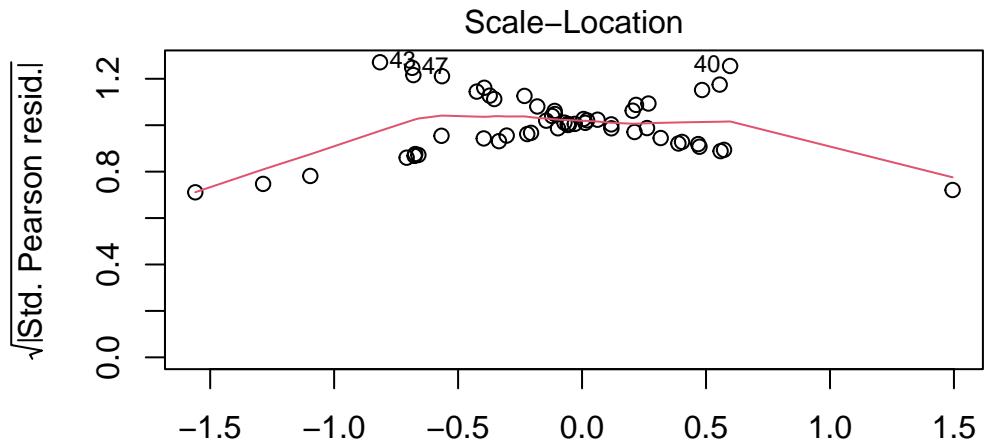
	LR	Chisq	Df	Pr(>Chisq)
ASSIDUITE	1.66105	1		0.1975
MOYENNE	0.08179	1		0.7749
SOMMEIL	0.04200	1		0.8376
STRESS	1.04725	1		0.3061
TRAJET	0.56963	1		0.4504

Nous obtenons deux variables significatives : MOYENNE au seuil de 5 % et ASSIDUITE au seuil de 1 %.

```
# Graphiques évaluant la qualité et la validité du modèle

plot(regression_logistique16)
```





```
# Prédition du modèle sur les 20 % des données non manquantes restantes

prediction17 = predict(regression_logistique16,
                      newdata = genre_verification)
```

```

# Ajout de la prédition parmi les 20 % de données en attribuant à chaque observation

genre_verification <- genre_verification |>
  mutate(prediction17) |>
  mutate(Gen_predi = case_when(prediction17 < 0 ~ "Femme",
                               TRUE ~ "Homme"))

# Comparaison des prédictions du modèle avec les vraies valeurs de la variable GENRE

genre_verification |>
  count(GENRE, Gen_predi)

# A tibble: 4 x 3
#>   GENRE Gen_predi     n
#>   <fct> <chr>      <int>
#> 1 Homme  Femme        4
#> 2 Homme  Homme        9
#> 3 Femme  Femme        3
#> 4 Femme  Homme        9

# Application du modèle de régression logistique binaire pour prédire les valeurs manquantes

prediction18 = predict(regression_logistique16,
                      newdata = filter(Budget2, is.na(GENRE)))

# Imputation de ces données manquantes dans notre base de données Budget2

Budget2 <- bind_rows(
  Budget2 |>
    filter(is.na(GENRE)) |>
    mutate(prediction18) |>
    mutate(GENRE = case_when(prediction18 < 0 ~ "Femme",
                             TRUE ~ "Homme")) |>
    select(- Gen),
  Budget2 |> filter(!is.na(GENRE)))

# Suppression des variables non nécessaires

```

```

sup <- c("For", "Par", "Tut", "Bou", "Emp", "Log", "Arg", "Caf", "Gen",
       "prediction2", "prediction4", "prediction6",
       "prediction8", "prediction10", "prediction12",
       "prediction14", "prediction16", "prediction18")

Budget2 <- Budget2 |>
  select(-all_of(sup))

```

B. Remplacement par la moyenne

1. Visualisation des données manquantes pour la variable MOYENNE

Procédons maintenant à l'imputation des valeurs manquantes de la variable MOYENNE. Pour cela, nous identifions les modalités des variables ASSIDUITE et REVISIONS associées à chaque valeur manquante. Ensuite, nous calculons la moyenne de MOYENNE pour les individus ayant les mêmes modalités. Cette moyenne est ensuite utilisée pour remplacer les valeurs manquantes correspondant à ces modalités.

En premier lieu, nous visualisons les lignes présentant les valeurs absentes.

```
# Lignes contenant des valeurs manquantes pour la variable MOYENNE.
```

```
moy_na <- which(is.na(Budget2$MOYENNE))
moy_na
```

```
[1] 2   6  12  17  22  31  33  55  56  69  79 118 120 121 126 133 135
```

Ensuite, nous déterminons les modalités des variables ASSIDUITE et REVISIONS pour les individus sans donnée pour MOYENNE.

```
# Modalités des variables ASSIDUITE ET REVISIONS pour les individus qui n'ont pas de d
```

```
Budget2 |>
  select(ASSIDUITE, REVISIONS, MOYENNE) |>
  filter(is.na(MOYENNE))
```

```
# A tibble: 17 x 3
  ASSIDUITE REVISIONS MOYENNE
  <dbl> <fct>     <dbl>
1     8 1 - 2h      NA
2     NA 0 - 1h     NA
3    10 1 - 2h     NA
4    10 1 - 2h     NA
5    10 2h et plus  NA
6     7 1 - 2h     NA
7    10 0 - 1h     NA
8    6 2h et plus  NA
9    10 0 - 1h     NA
10   NA 0 - 1h     NA
11   10 1 - 2h     NA
12   8 1 - 2h     NA
13   9 1 - 2h     NA
14   8 1 - 2h     NA
15   8 0 - 1h     NA
16   10 1 - 2h    NA
17   6 0 - 1h     NA
```

2. Imputation

Puis, nous calculons la moyenne de la variable MOYENNE pour les individus partageant les mêmes modalités et ayant une valeur définie pour MOYENNE.

```
# Moyennes des individus ayant des modalités identiques à ceux qui n'ont pas de valeur

moyenne81 <- Budget2 |>
  filter(ASSIDUITE == "8" & REVISIONS == "1 - 2h") |>
  select(where(is.numeric)) |>
  summarise(
    across(
      everything(),
      ~ mean(.x, na.rm = TRUE) |>
        round()
    )
  )

moyenne81
```

```
# A tibble: 1 x 7
  ASSIDUITE MOYENNE RESTAURANT    AGE STRESS SOMMEIL TRAJET
  <dbl>     <dbl>      <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1       8       13        2     20      4      7     27
```

```
moyenne71 <- Budget2 |>
  filter(ASSIDUITE == "7" & REVISIONS == "1 - 2h") |>
  select(where(is.numeric)) |>
  summarise(
    across(
      everything(),
      ~ mean(.x, na.rm = TRUE) |>
        round()
    )
  )
```

```
moyenne71
```

```
# A tibble: 1 x 7
  ASSIDUITE MOYENNE RESTAURANT    AGE STRESS SOMMEIL TRAJET
  <dbl>     <dbl>      <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1       7       14        4     21      4      7     27
```

```
moyenne101 <- Budget2 |>
  filter(ASSIDUITE == "10" & REVISIONS == "1 - 2h") |>
  select(where(is.numeric)) |>
  summarise(
    across(
      everything(),
      ~ mean(.x, na.rm = TRUE) |>
        round()
    )
  )
```

```
moyenne101
```

```
# A tibble: 1 x 7
  ASSIDUITE MOYENNE RESTAURANT    AGE STRESS SOMMEIL TRAJET
  <dbl>     <dbl>      <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1      10       13        2     21      3      7     32
```

```

moyenne102 <- Budget2 |>
  filter(ASSIDUITE == "10" & REVISIONS == "2h et plus") |>
  select(where(is.numeric)) |>
  summarise(
    across(
      everything(),
      ~ mean(.x, na.rm = TRUE) |>
        round()
    ) )

```

moyenne102

```

# A tibble: 1 x 7
  ASSIDUITE MOYENNE RESTAURANT   AGE STRESS SOMMEIL TRAJET
  <dbl>     <dbl>       <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1       10       14           1    22      3       7      31

```

```

moyenne0 <- Budget2 |>
  filter(REVISIONS == "0 - 1h") |>
  select(where(is.numeric)) |>
  summarise(
    across(
      everything(),
      ~ mean(.x, na.rm = TRUE) |>
        round()
    ) )

```

moyenne0

```

# A tibble: 1 x 7
  ASSIDUITE MOYENNE RESTAURANT   AGE STRESS SOMMEIL TRAJET
  <dbl>     <dbl>       <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1       8        13           2    21      4       7      38

```

```

moyenne100 <- Budget2 |>
  filter(ASSIDUITE == "10" & REVISIONS == "0 - 1h") |>
  select(where(is.numeric)) |>
  summarise(

```

```

across(
  everything(),
  ~ mean(.x, na.rm = TRUE) |>
  round()
) )

moyenne100

# A tibble: 1 x 7
ASSIDUITE MOYENNE RESTAURANT    AGE STRESS SOMMEIL TRAJET
<dbl>     <dbl>      <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1       10       14           2     22      3       7     42

moyenne62 <- Budget2 |>
filter(ASSIDUITE == "6" & REVISIONS == "2h et plus") |>
select(where(is.numeric)) |>
summarise(
  across(
    everything(),
    ~ mean(.x, na.rm = TRUE) |>
    round()
) )

moyenne62

# A tibble: 1 x 7
ASSIDUITE MOYENNE RESTAURANT    AGE STRESS SOMMEIL TRAJET
<dbl>     <dbl>      <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1       6        13           2     23      6       8     49

moyenne91 <- Budget2 |>
filter(ASSIDUITE == "9" & REVISIONS == "1 - 2h") |>
select(where(is.numeric)) |>
summarise(
  across(
    everything(),
    ~ mean(.x, na.rm = TRUE) |>
    round()
)

```

```
) )
```

```
moyenne91
```

```
# A tibble: 1 x 7
ASSIDUITE MOYENNE RESTAURANT    AGE STRESS SOMMEIL TRAJET
<dbl>     <dbl>      <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1         9       13        2     21      3       8      30
```

```
moyenne80 <- Budget2 |>
  filter(ASSIDUITE == "8" & REVISIONS == "0 - 1h") |>
  select(where(is.numeric)) |>
  summarise(
    across(
      everything(),
      ~ mean(.x, na.rm = TRUE) |>
        round()
    ) )
```

```
moyenne80
```

```
# A tibble: 1 x 7
ASSIDUITE MOYENNE RESTAURANT    AGE STRESS SOMMEIL TRAJET
<dbl>     <dbl>      <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1         8       13        2     21      6       7      41
```

```
moyenne60 <- Budget2 |>
  filter(ASSIDUITE == "6" & REVISIONS == "0 - 1h") |>
  select(where(is.numeric)) |>
  summarise(
    across(
      everything(),
      ~ mean(.x, na.rm = TRUE) |>
        round()
    ) )
```

```
moyenne60
```

```
# A tibble: 1 x 7
ASSIDUITE MOYENNE RESTAURANT AGE STRESS SOMMEIL TRAJET
<dbl>     <dbl>      <dbl> <dbl>   <dbl>   <dbl>   <dbl>
1       6      13        2     21      6      6     29
```

Enfin, nous remplaçons les valeurs manquantes de MOYENNE en fonction des modalités de ASSIDUITE et REVISIONS, en utilisant la moyenne appropriée pour chaque combinaison de modalités.

```
# Imputations par les moyennes calculées

moy <- c(moyenne81[[1, 2]], moyenne71[[1, 2]], moyenne101[[1, 2]], moyenne102[[1, 2]],
moyenne0[[1, 2]], moyenne101[[1, 2]], moyenne100[[1, 2]], moyenne62[[1, 2]],
moyenne100[[1, 2]], moyenne0[[1, 2]], moyenne101[[1, 2]], moyenne81[[1, 2]],
moyenne91[[1, 2]], moyenne81[[1, 2]], moyenne80[[1, 2]], moyenne101[[1, 2]],
moyenne60[[1, 2]])

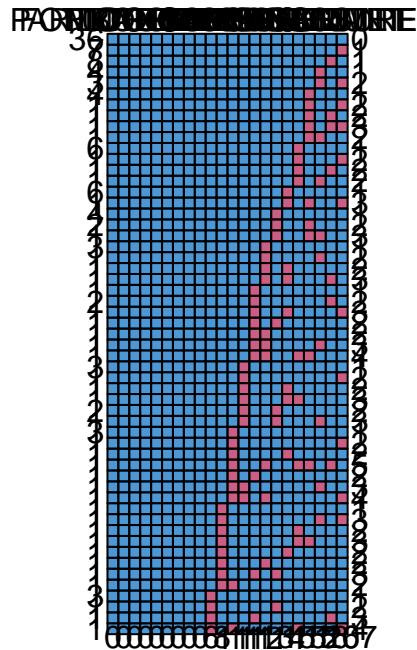
# Boucle for pour modifier toutes les valeurs
m = 0
for (i in moy_na) {
  m <- m + 1
  Budget2$MOYENNE[i] <- moy[m]
}
```

C. MICE

Il nous reste des valeurs manquantes à remplacer dans 8 variables différentes. Pour ce faire, nous allons procéder à une troisième méthode de remplacement qui est la méthode d'imputation multiple MICE.

Nous commençons par utiliser une fonction pour visualiser le modèle de valeurs manquantes. Celle-ci indique entre autres quelles colonnes présentent des données manquantes et combien d'observations sont affectées.

```
md.pattern(Budget2)
```



	FORMATION	PARTICULIERS	TUTORAT	MOYENNE	BOURSE	LOGEMENT	ARGENT	CAF	GENRE
36	1		1	1	1	1	1	1	1
7	1		1	1	1	1	1	1	1
8	1		1	1	1	1	1	1	1
4	1		1	1	1	1	1	1	1
3	1		1	1	1	1	1	1	1
4	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
6	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
6	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
4	1		1	1	1	1	1	1	1
2	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1
3	1		1	1	1	1	1	1	1
1	1		1	1	1	1	1	1	1

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0

REVISIONS STRESS EMPLOI TRANSPORT TRAJET DEPENSES SANTE AGE RESTAURANT

36	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1

4	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	0
6	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	0	1	1
2	1	1	1	1	1	1	0	1	1
1	1	1	1	1	1	1	0	1	1
3	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	0	1	1	0	1
2	1	1	1	1	0	1	1	1	1
1	1	1	1	1	0	1	1	0	1
1	1	1	1	1	0	1	0	1	1
1	1	1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	1	1	0
3	1	1	1	0	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1
1	1	1	1	0	1	1	1	0	1
1	1	1	1	0	1	1	1	0	0
2	1	1	1	0	1	1	0	1	1
1	1	1	1	0	1	1	0	1	1
3	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	0	1
1	1	1	0	1	1	0	1	1	1
1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	1	1	1	1	0
1	1	1	0	0	1	0	1	1	0
1	1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	0	1	1	1
1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	1	0	1	1	1
1	1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	0

1	1	0	1	1	1	1	1	1	0
1	1	0	1	1	1	1	1	0	1
1	1	0	1	1	1	0	1	1	1
1	1	0	1	1	0	1	0	1	1
1	1	0	0	1	1	1	1	1	1
3	0	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	1	1	0
	6	8	11	11	11	12	13	14	15
	STATUT	STRUCTURE	ASSIDUITE	SOMMEIL					
36	1	1	1	1	0				
7	1	1	1	0	1				
8	1	1	0	1	1				
4	1	0	1	1	1				
3	1	0	1	0	2				
4	0	1	1	1	1				
1	0	1	1	0	2				
1	0	1	0	1	2				
1	0	1	0	0	3				
1	0	0	1	1	2				
6	1	1	1	1	1				
1	1	1	1	0	2				
1	1	1	0	1	2				
1	1	0	1	1	2				
6	1	1	1	1	1				
1	0	1	1	0	3				
4	1	1	1	1	1				
2	0	1	1	1	2				
1	0	0	1	1	3				
3	1	1	1	1	1				
1	1	0	1	1	2				
1	1	1	1	1	2				
1	1	1	0	1	3				
1	1	1	1	1	1				
2	1	1	0	1	2				
1	1	1	1	0	3				
1	1	1	1	1	2				
1	1	1	1	1	2				
1	1	0	1	1	3				
1	0	1	1	1	4				

3	1	1	1	1	1
1	1	1	1	0	2
1	1	1	1	1	2
1	1	1	1	1	3
2	1	1	1	1	2
1	1	0	1	1	3
3	1	1	1	1	1
1	1	1	1	0	2
1	1	1	1	1	2
1	0	1	0	1	5
1	1	1	1	1	2
1	1	0	1	1	3
1	1	1	1	0	4
1	1	1	1	1	1
1	1	0	1	0	3
1	1	1	1	1	2
1	0	1	1	1	3
1	1	1	1	1	2
1	1	1	1	1	2
1	1	1	1	1	3
1	1	1	1	1	2
3	1	1	1	1	1
1	1	1	1	1	2
1	1	1	0	1	3
1	1	1	1	0	4
15	15	16	20	167	

Nous effectuons ensuite une imputation multiple sur un jeu de données contenant des valeurs manquantes.

```
budget_impute <- mice(Budget2, m = 5)
```

iter	imp	variable	ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
1	1	ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S	
1	2	ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S	
1	3	ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S	
1	4	ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S	
1	5	ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S	
2	1	ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S	

			ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
2	2		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
2	3		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
2	4		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
2	5		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
3	1		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
3	2		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
3	3		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
3	4		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
3	5		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
4	1		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
4	2		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
4	3		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
4	4		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
4	5		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
5	1		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
5	2		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
5	3		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
5	4		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S
5	5		ASSIDUITE	REVISIONS	EMPLOI	RESTAURANT	DEPENSES	TRANSPORT	AGE	STRESS	S

Warning: Number of logged events: 8

Puis, nous accédons directement aux valeurs imputées générées.

```
budget_impute$imp
```

```
$FORMATION
[1] 1 2 3 4 5
<0 rows> (or 0-length row.names)

$ASSIDUITE
     1 2 3 4 5
6    9 8 10 9 10
8   10 10 10 6 6
9    7 10 8 8 10
18   9 9 6 10 9
37   10 9 8 9 8
42   10 10 10 9 9
46   9 8 8 9 6
```

```

66   8   9   6   7   9
67   9   6   6   7   10
69   7   5   6  10   7
85   7   7   6   6   6
87  10  10   6   8   6
104 10  10  10   8  10
112 10   6   7   6  10
113  8   8   9   8   9
129 10   7   8  10   8

```

\$REVISIONS

	1	2	3	4	5
25	1 - 2h	0 - 1h	0 - 1h	2h et plus	0 - 1h
36	0 - 1h	2h et plus	0 - 1h	0 - 1h	0 - 1h
51	0 - 1h	2h et plus	1 - 2h	1 - 2h	1 - 2h
66	1 - 2h	0 - 1h	0 - 1h	0 - 1h	1 - 2h
91	1 - 2h	1 - 2h	1 - 2h	1 - 2h	1 - 2h
103	2h et plus	2h et plus	1 - 2h	2h et plus	0 - 1h

\$PARTICULIERS

```

[1] 1 2 3 4 5
<0 rows> (or 0-length row.names)

```

\$TUTORAT

```

[1] 1 2 3 4 5
<0 rows> (or 0-length row.names)

```

\$MOYENNE

```

[1] 1 2 3 4 5
<0 rows> (or 0-length row.names)

```

\$BOURSE

```

[1] 1 2 3 4 5
<0 rows> (or 0-length row.names)

```

\$EMPLOI

	1	2	3	4	5
4	Oui	Oui	Oui	Non	Oui
23	Oui	Non	Non	Non	Non
32	Oui	Oui	Oui	Oui	Non
45	Non	Non	Oui	Non	Non

```
46 Non Non Non Non Non  
47 Oui Non Non Oui Non  
48 Non Non Oui Oui Non  
49 Non Oui Non Non Non  
50 Oui Non Non Non Non  
51 Non Non Oui Non Oui  
52 Non Non Oui Oui Non
```

\$LOGEMENT

```
[1] 1 2 3 4 5  
<0 rows> (or 0-length row.names)
```

\$ARGENT

```
[1] 1 2 3 4 5  
<0 rows> (or 0-length row.names)
```

\$RESTAURANT

	1	2	3	4	5
19	0.0	1	0	1.0	1
29	0.3	1	2	4.0	1
46	4.0	5	4	2.0	5
51	4.0	2	4	2.0	3
54	2.0	1	3	1.0	1
61	2.0	2	4	1.0	3
62	1.0	1	1	2.0	2
87	5.0	0	3	3.0	5
88	2.0	2	4	1.0	1
97	4.0	5	2	5.0	2
105	3.0	4	5	6.0	2
110	5.0	3	4	3.0	1
117	1.0	1	1	2.0	1
121	2.0	1	1	3.0	6
130	1.0	1	1	0.5	1

\$DEPENSES

	1	2	3	4	5
10	0 - 100 €	100 € et plus	0 - 100 €	100 € et plus	100 € et plus
23	0 - 100 €	100 € et plus	100 € et plus	0 - 100 €	100 € et plus
39	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €
46	0 - 100 €	100 € et plus	0 - 100 €	100 € et plus	0 - 100 €
59	100 € et plus	100 € et plus	0 - 100 €	0 - 100 €	100 € et plus

75	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €	100 € et plus
79	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €
98	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €
105	0 - 100 €	0 - 100 €	100 € et plus	100 € et plus	0 - 100 €
112	100 € et plus	0 - 100 €	100 € et plus	0 - 100 €	100 € et plus
115	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €	0 - 100 €
133	0 - 100 €	0 - 100 €	0 - 100 €	100 € et plus	100 € et plus

\$CAF

```
[1] 1 2 3 4 5
<0 rows> (or 0-length row.names)
```

\$TRANSPORT

	1	2	3	4	5
1	30 € et plus				
14	30 € et plus	15 - 30 €	15 - 30 €	15 - 30 €	15 - 30 €
17	30 € et plus	15 - 30 €	30 € et plus	30 € et plus	0 - 15 €
23	0 - 15 €	0 - 15 €	0 - 15 €	0 - 15 €	30 € et plus
29	15 - 30 €	30 € et plus	15 - 30 €	0 - 15 €	15 - 30 €
31	15 - 30 €	0 - 15 €	0 - 15 €	15 - 30 €	15 - 30 €
38	30 € et plus	15 - 30 €	0 - 15 €	0 - 15 €	0 - 15 €
49	15 - 30 €	15 - 30 €	15 - 30 €	15 - 30 €	15 - 30 €
80	0 - 15 €	0 - 15 €	0 - 15 €	30 € et plus	15 - 30 €
114	15 - 30 €	0 - 15 €	30 € et plus	0 - 15 €	15 - 30 €
118	30 € et plus	30 € et plus	0 - 15 €	30 € et plus	30 € et plus

\$GENRE

```
[1] 1 2 3 4 5
<0 rows> (or 0-length row.names)
```

\$AGE

	1	2	3	4	5
24	20	23	20	20	20
29	23	20	23	20	23
32	22	23	22	20	20
59	21	20	23	22	23
64	21	21	20	22	21
78	23	21	21	22	18
92	20	20	19	20	21
94	25	25	25	25	20
95	20	20	21	20	20

100	20	20	18	20	21
108	22	20	20	24	23
112	22	20	20	20	20
114	23	22	23	20	22
123	22	22	22	24	22

\$STRESS

	1	2	3	4	5
2	2	8	4	3	1
19	2	1	3	1	3
52	1	1	1	2	1
72	1	8	7	3	1
81	1	2	3	2	1
95	2	1	3	2	5
98	1	4	4	1	5
130	6	3	5	4	6

\$STATUT

	1	2	3	4
42	Classe populaire	Classe moyenne	Classe moyenne	Classe aisée
46	Classe moyenne	Classe moyenne	Classe moyenne	Classe moyenne
63	Classe moyenne	Classe moyenne	Classe moyenne	Classe moyenne
65	Classe moyenne	Classe moyenne	Classe populaire	Classe moyenne
74	Classe moyenne	Classe moyenne	Classe moyenne	Classe moyenne
76	Classe moyenne	Classe moyenne	Classe moyenne	Classe moyenne
105	Classe moyenne	Classe moyenne	Classe populaire	Classe moyenne
120	Classe populaire	Classe aisée	Classe populaire	Classe aisée
122	Classe moyenne	Classe moyenne	Classe moyenne	Classe moyenne
123	Classe populaire	Classe moyenne	Classe populaire	Classe moyenne
126	Classe moyenne	Classe moyenne	Classe moyenne	Classe aisée
128	Classe aisée	Classe aisée	Classe moyenne	Classe moyenne
129	Classe moyenne	Classe moyenne	Classe moyenne	Classe populaire
130	Classe populaire	Classe moyenne	Classe moyenne	Classe moyenne
135	Classe moyenne	Classe moyenne	Classe populaire	Classe populaire
	5			
42	Classe moyenne			
46	Classe moyenne			
63	Classe moyenne			
65	Classe populaire			
74	Classe aisée			
76	Classe moyenne			

105 Classe moyenne
120 Classe aisée
122 Classe moyenne
123 Classe moyenne
126 Classe populaire
128 Classe populaire
129 Classe populaire
130 Classe moyenne
135 Classe moyenne

\$SOMMEIL

	1	2	3	4	5
1	8.0	7.0	7.0	8.0	9.0
2	7.0	7.0	6.0	6.0	6.5
5	7.0	8.0	7.0	8.0	9.0
21	8.0	7.0	8.0	8.0	8.0
22	7.0	7.5	5.0	7.0	7.0
23	8.0	6.5	7.0	7.0	7.0
24	7.0	6.0	7.0	7.0	8.0
26	9.0	7.0	8.0	7.0	7.0
34	8.0	6.5	6.0	6.0	7.0
35	7.0	5.0	6.0	6.0	7.0
45	8.0	7.0	7.0	7.0	7.0
51	6.0	8.0	7.0	7.0	7.0
58	7.0	6.0	8.0	7.0	8.0
62	8.0	7.0	8.0	8.0	7.0
73	8.0	5.0	8.0	7.0	8.0
99	8.0	8.0	8.0	8.0	8.0
107	8.0	8.0	7.5	8.0	8.0
123	7.5	6.0	7.0	8.0	7.0
129	8.0	7.0	7.0	7.5	6.0
135	6.0	8.0	6.0	7.0	7.0

\$SANTE

	1	2	3	4	5
14	Oui	Non	Non	Oui	Oui
38	Oui	Non	Non	Non	Non
40	Non	Non	Oui	Non	Oui
53	Non	Oui	Non	Oui	Oui
56	Oui	Non	Oui	Non	Non
65	Oui	Oui	Non	Oui	Oui

74	Oui	Oui	Non	Non	Non
76	Non	Non	Non	Oui	Oui
80	Non	Oui	Non	Non	Non
81	Non	Non	Non	Non	Non
91	Non	Non	Non	Non	Non
101	Non	Oui	Non	Non	Oui
109	Non	Non	Oui	Non	Non

\$STRUCTURE

	1	2	3	4	5
2	Université	Université	Université	Université	Université
22	Université	Université	Université	Université	Autres
35	Université	Université	Université	Université	Université
38	Université	Université	Université	Université	Autres
49	Université	Université	Université	Université	Université
57	Université	Université	Université	Université	Université
65	Autres	Université	Autres	Université	Université
73	Université	Autres	Autres	Université	Université
75	Université	Université	Université	Université	Université
89	Université	Université	Université	Autres	Autres
116	Université	Université	Université	Université	Université
121	Université	Université	Université	Université	Université
126	Université	Université	Université	Université	Autres
133	Université	Autres	Autres	Université	Université
134	Autres	Autres	Autres	Autres	Autres

\$TRAJET

	1	2	3	4	5
4	30	10	90	30	40
10	40	30	10	30	20
24	45	90	60	60	20
66	20	20	25	15	30
67	10	60	10	20	25
69	30	20	15	10	25
75	40	15	30	20	25
81	15	45	50	70	30
105	10	90	90	60	90
109	15	20	10	20	35
124	20	50	15	10	60

Nous pouvons désormais consulter pour modifier si nécessaire les méthodes d'imputation

utilisées pour chaque variable.

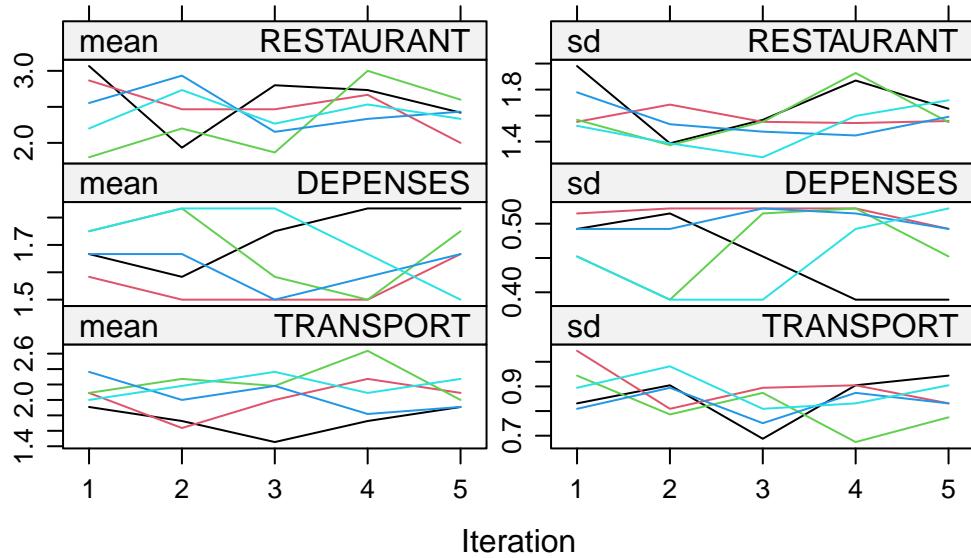
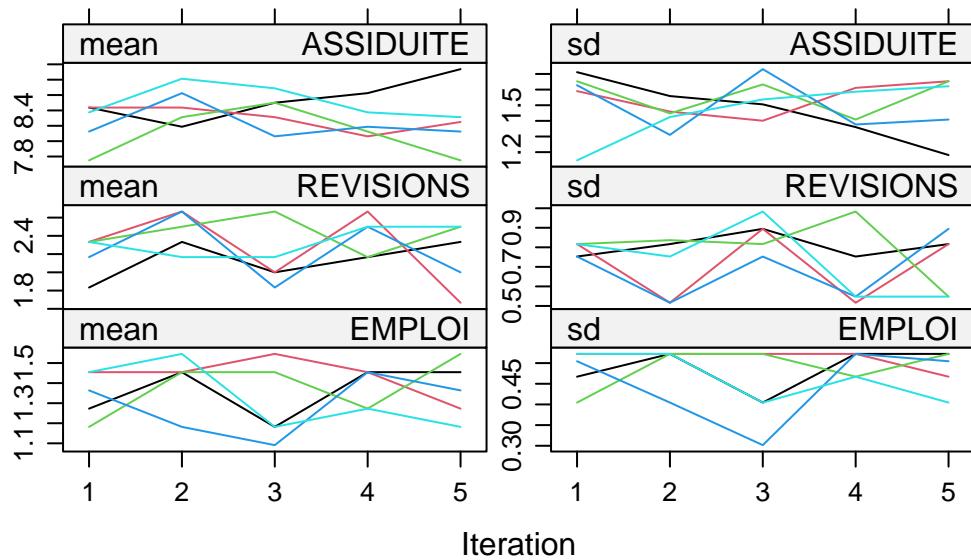
```
budget_impute$method
```

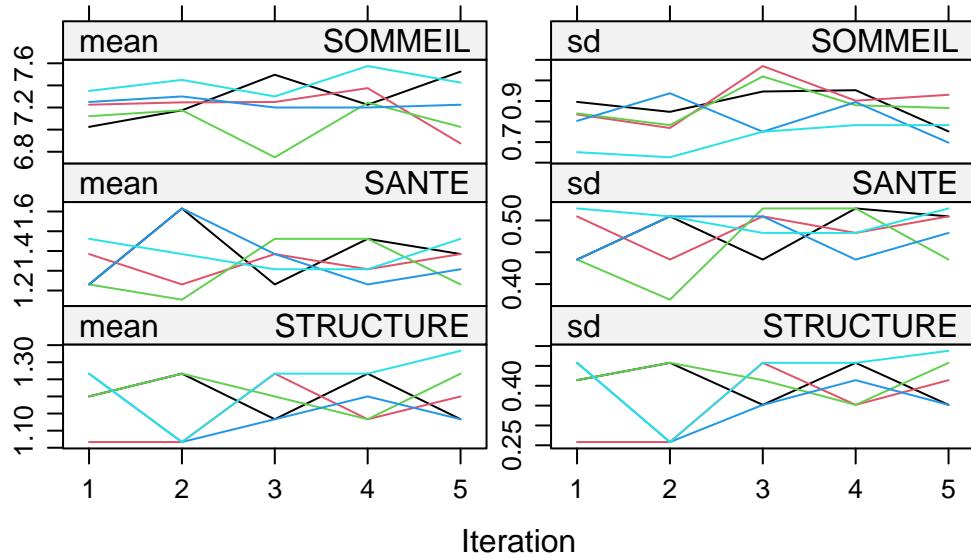
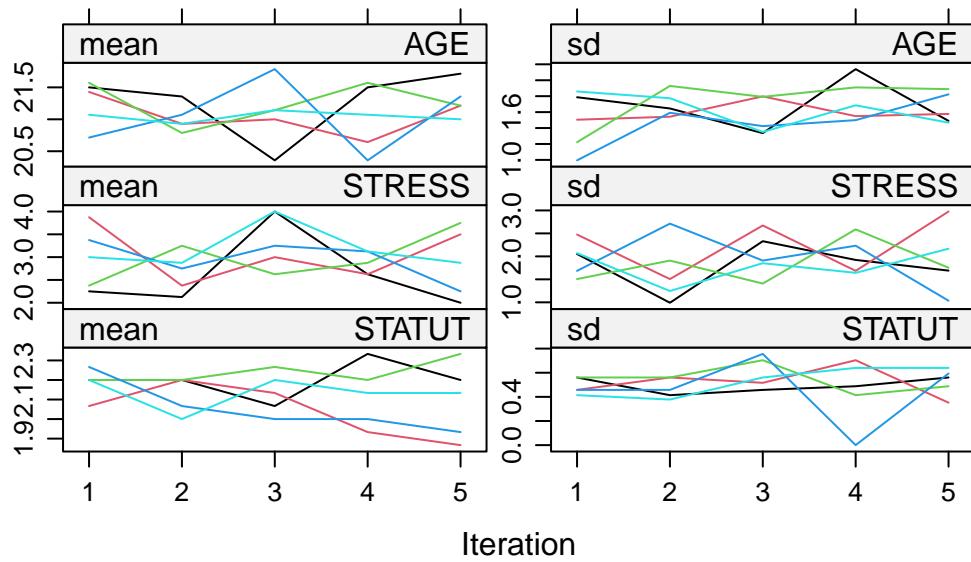
FORMATION	ASSIDUITE	REVISIONS	PARTICULIERS	TUTORAT	MOYENNE
" "	"pmm"	"polyreg"	" "	" "	" "
BOURSE	EMPLOI	LOGEMENT	ARGENT	RESTAURANT	DEPENSES
" "	"logreg"	" "	" "	"pmm"	"logreg"
CAF	TRANSPORT	GENRE	AGE	STRESS	STATUT
" "	"polyreg"	" "	"pmm"	"pmm"	"polyreg"
SOMMEIL	SANTE	STRUCTURE	TRAJET		
"pmm"	"logreg"	"logreg"	"pmm"		

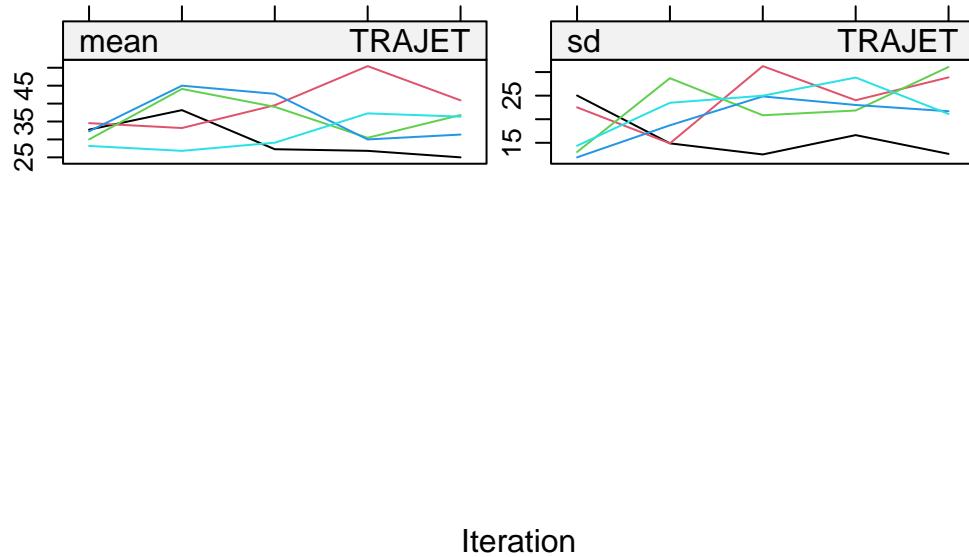
Des différentes méthodes d'imputation sont utilisées en fonction du type de variable et de la nature des données manquantes. Pour les variables continues comme ASSIDUITE, STRESS, et SOMMEIL, la méthode pmm (Predictive Mean Matching) est utilisée, ce qui permet d'imputer les valeurs manquantes en tirant des valeurs proches des prédictions des autres observations. Pour les variables binaires comme PARTICULIERS et EMPLOI, la méthode logreg (régression logistique) est appliquée, qui remplace les valeurs manquantes en fonction des probabilités d'appartenir à l'une des deux catégories possibles. Les variables catégorielles, telles que REVISIONS, bénéficient de la méthode polyreg (régression polynomiale), qui est adaptée aux variables avec plusieurs catégories. Certaines variables, comme FORMATION, MOYENNE, et LOGEMENT, ne nécessitent pas d'imputation ou sont exclues du processus, car elles ne contiennent pas de valeurs manquantes.

Nous choisissons l'imputation 3 pour continuer le MICE.

```
budget_complete_mice <- mice::complete(budget_impute, 3)  
plot(budget_impute)
```

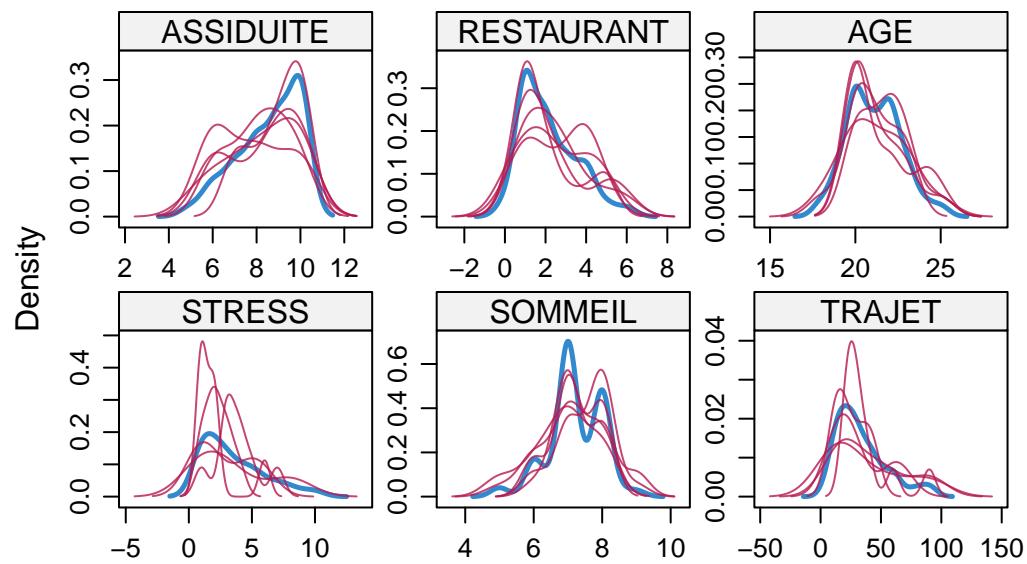






Nous utilisons maintenant la commande DensityPlot pour visualiser la distribution des valeurs imputées pour chaque variable dans notre jeu de données. Elle est particulièrement utile pour examiner la qualité et la cohérence des imputations.

```
densityplot(budget_impute)
```



Ici, nous pouvons voir que les valeurs imputées (ligne rouge) suivent, généralement, les valeurs observées (ligne bleue).

```
# Base de données finale sans valeur manquante
View(budget_complete_mice)
```

VIII. Conclusion

En conclusion, pour remplacer nos différentes valeurs manquantes nous avons procéder à 3 méthodes, cela nous a permis d'éviter une perte d'information pouvant être importante.

En comparant l'ancienne à la nouvelle base nous avons pu relever de très faibles différences.

Par exemple, pour la ligne 9, concernant TUTORAT, la valeur observée était "Non" et la valeur prédictive est "Non". Pour la ligne 16, à FORMATION, la valeur observée était "Autres" et la valeur prédictive est "Autres". Pour la ligne 18, à MOYENNE, la valeur observée était "17" et la valeur prédictive est "10,85". Cela peut être dû au fait que la valeur "17" est une valeur extrême comparée aux autres. Par exemple, à la ligne 133, la valeur prédictive est "13" et la valeur observée est "13" également. Pour la ligne 24, à ASSIDUITE, la valeur observée était "9" et la valeur prédictive est "0".

Il y a donc une bonne cohérence entre l'ancienne et la nouvelle base de données, ce qui prouve que ces méthodes sont efficaces.