Introduction sur les données manquantes

Marie Vaugoyeau

20 February 2024

Table of contents

1 Définitions **1**

2 Type de données manquantes **2**

3 Conséquences des valeurs manquantes **2**

4 Identifier les valeurs manquantes **6**

5 L'analyse descriptive **11**

6 Traitement des valeurs manquantes **13**

7 En savoir un peu plus sur moi **17**

1 Définitions

Les données manquantes sont les données qui ne sont pas présentes.
La donnée peut-être remplacée dans le tableau par :

- `NA`
- Une autre valeur dépendante des données ou de la personne qui s'en ai occupée : `0`, `NO`, `999`...

> **Note**
>
> Quelques soit le cas, il existent plusieurs origines aux données manquantes.

2 Type de données manquantes

Les données manquantes, représentées par `NA` ou autre peuvent avoir plusieurs origines :

- La donnée **n'est pas compatible**. *Par exemple, une personne rentre du texte au lieu d'un numéro de téléphone.* Dans ce cas le **système ne prends pas en charge** la réponse et la qualifie en `NA` pour `Not Applicable`

- La donnée **n'existe pas**. *Par exemple la personne n'a pas de numéro de téléphone*, dans ce cas, le système la qualifie de `NA` pour `Not Available`

- La donnée **existe mais n'a pas été communiquées**. *Par exemple la personne a refusé de donner son numéro*, dans ce cas, le système la qualifie de `NA` pour `Not Answer`

Dans tous les cas, la **seule information transmise** est que la **données n'est pas disponible**.
Il n'est pas toujours possible de cerner l'origine du problème mais cela n'empêche pas d'agir.
Il faut commencer par se demander ce que **signifie cette absence** et **comment elle va impacter** notre système.

3 Conséquences des valeurs manquantes

- **Perte d'information** : Si la donnée peut-être retrouvée ou remplacée, pourquoi s'en empêcher ?

- **Erreur dans la généralisation** : Si beaucoup de données sont manquantes et que les conclusions se basent uniquement sur celles présentent, **est-ce que cela représente vraiment la réalité ?**

- Comportement de certains modèles stats

```
library(tidyverse)
library(missMDA)
data("snorena")

# régression logistique
regression_logistique <- glm(
  snore ~ age + weight + size + alcohol,
  family = binomial,
  data = snorena
  )
```

```
regression_logistique
```

```
Call:  glm(formula = snore ~ age + weight + size + alcohol, family = binomial,
    data = snorena)

Coefficients:
(Intercept)          age       weight         size      alcohol
  -4.221694     0.061238     0.001180    -0.001735     0.157680

Degrees of Freedom: 72 Total (i.e. Null);  68 Residual
  (27 observations effacées parce que manquantes)
Null Deviance:      93.83
Residual Deviance: 84.72    AIC: 94.72
```

```
summary(regression_logistique)
```

```
Call:
glm(formula = snore ~ age + weight + size + alcohol, family = binomial,
    data = snorena)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.221694   6.845567  -0.617   0.5374
age          0.061238   0.025278   2.423   0.0154 *
weight       0.001180   0.040353   0.029   0.9767
size        -0.001735   0.055573  -0.031   0.9751
alcohol      0.157680   0.080148   1.967   0.0491 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93.828  on 72  degrees of freedom
Residual deviance: 84.724  on 68  degrees of freedom
  (27 observations effacées parce que manquantes)
AIC: 94.724

Number of Fisher Scoring iterations: 4
```

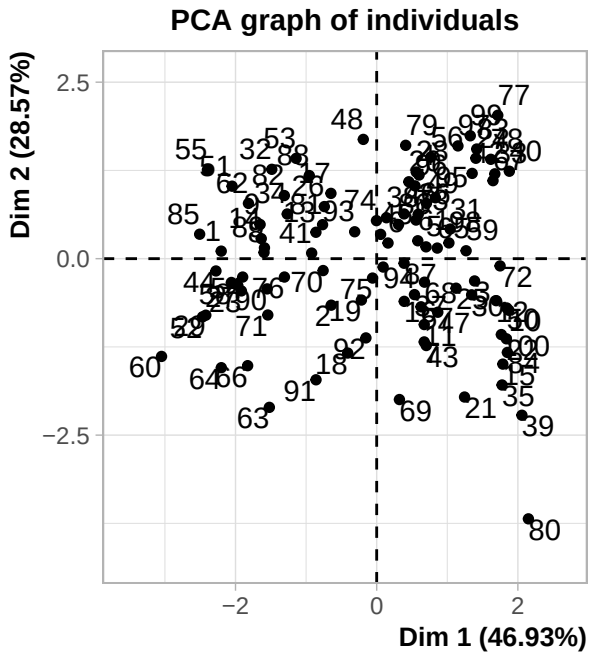
```r
regression_logistique_2 <- glm(
  snore ~ age + alcohol,
  family = binomial,
  data = snorena
  )

AIC(regression_logistique, regression_logistique_2)
```

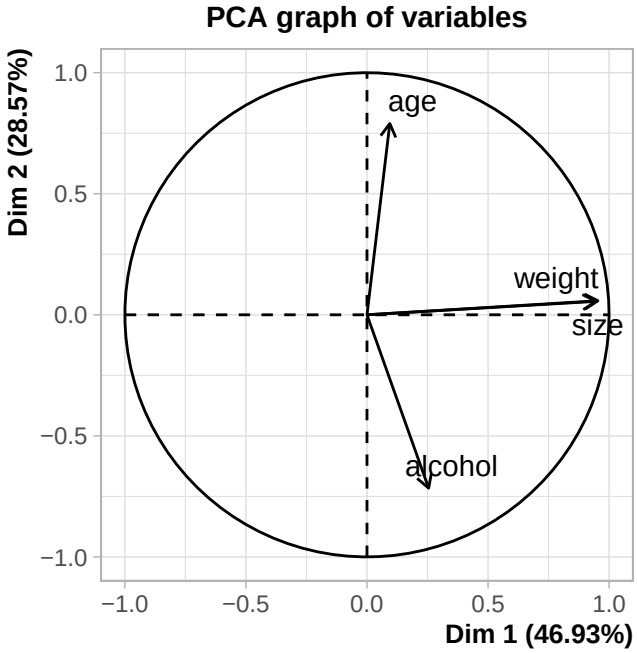
```
                        df       AIC
regression_logistique    5  94.72355
regression_logistique_2  3 104.50932
```

```r
summary(regression_logistique_2)
```

```
Call:
glm(formula = snore ~ age + alcohol, family = binomial, data = snorena)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.20428    1.50198  -3.465  0.00053 ***
age          0.07353    0.02483   2.961  0.00307 **
alcohol      0.20418    0.07716   2.646  0.00814 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 113.321  on 86  degrees of freedom
Residual deviance:  98.509  on 84  degrees of freedom
  (13 observations effacées parce que manquantes)
AIC: 104.51

Number of Fisher Scoring iterations: 4
```

```r
# ACP
library(FactoMineR)

acp <- PCA(snorena |> select(where(is.numeric)))
```


PCA graph of individuals
Dim 2 (28.57%)
Dim 1 (46.93%)
2.5
0.0
−2.5
−2
0
2


PCA graph of variables
Dim 2 (28.57%)
Dim 1 (46.93%)
age
weight
size
alcohol
1.0
0.5
0.0
−0.5
−1.0

4 Identifier les valeurs manquantes

Pour savoir comment agir, il faut commencer par quantifier et localiser les valeurs manquantes.

Une réalisation simple est l'utilisation de la fonciton `summary()` du package `{base}`.

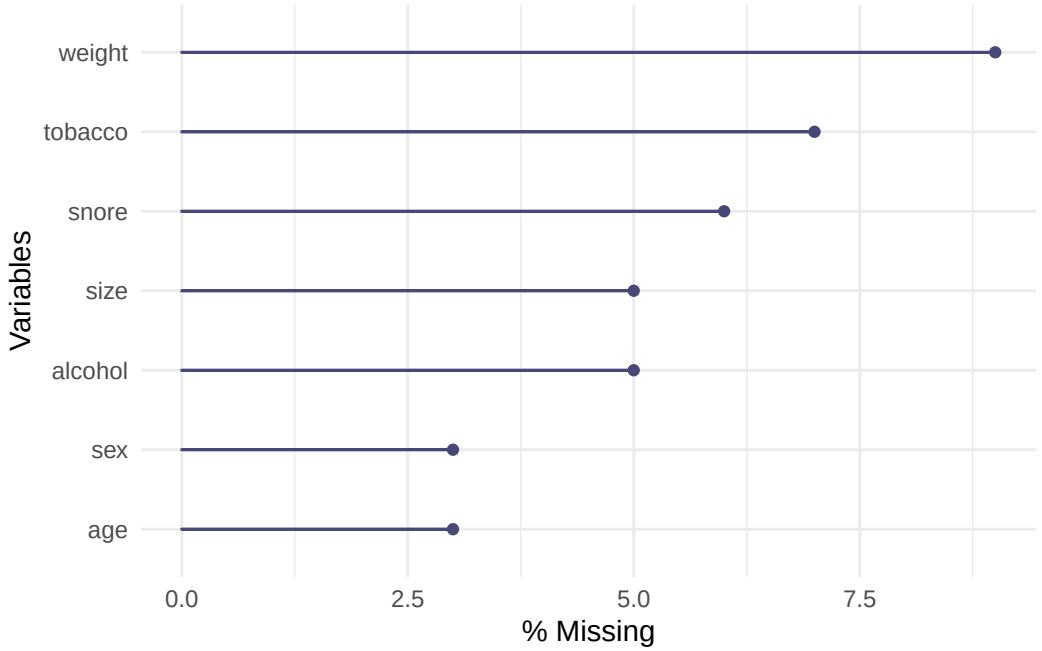
```
summary(snorena)
```

```
      age            weight           size          alcohol          sex
 Min.   :23.00   Min.   : 42.0   Min.   :158.0   Min.   : 0.000   M   :75
 1st Qu.:43.00   1st Qu.: 77.0   1st Qu.:166.0   1st Qu.: 0.000   W   :22
 Median :51.00   Median : 94.0   Median :186.0   Median : 2.000   NA's: 3
 Mean   :52.16   Mean   : 90.4   Mean   :181.1   Mean   : 2.905
 3rd Qu.:63.00   3rd Qu.:104.5   3rd Qu.:194.0   3rd Qu.: 4.000
 Max.   :74.00   Max.   :120.0   Max.   :208.0   Max.   :15.000
 NA's   :3       NA's   :9       NA's   :5       NA's   :5
  snore    tobacco
 N   :62   N   :32
 Y   :32   Y   :61
 NA's: 6   NA's: 7
```

Le package `{naniar}` est spécialement adapté à la visualisation des données manquantes.

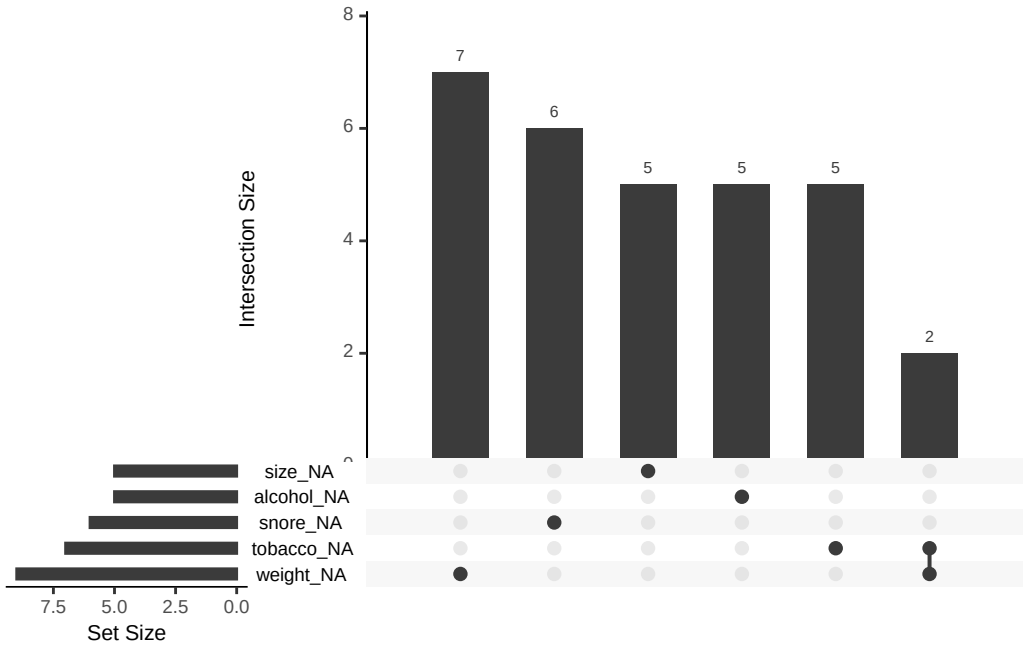
Visualisation du nombre ou de la proportion de données manquantes grâce aux fonctions `gg_miss_var()`.

```
library(naniar)

gg_miss_var(snorena, show_pct = TRUE)
```



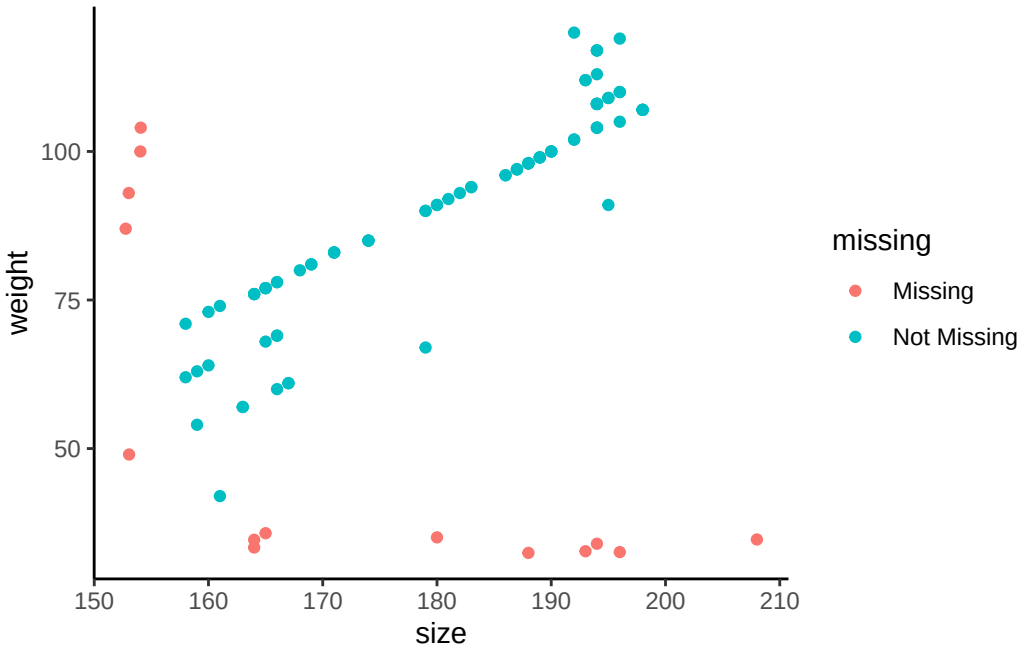
La fonction `gg_miss_upset()` permet de représenter sur un graphique les variables qui ont des données manquantes et le lien entre les colonnes.

```
gg_miss_upset(snorena)
```



La fonction `geom_miss_point()` permet de visualiser les valeurs manquantes sur les nuage de poin.

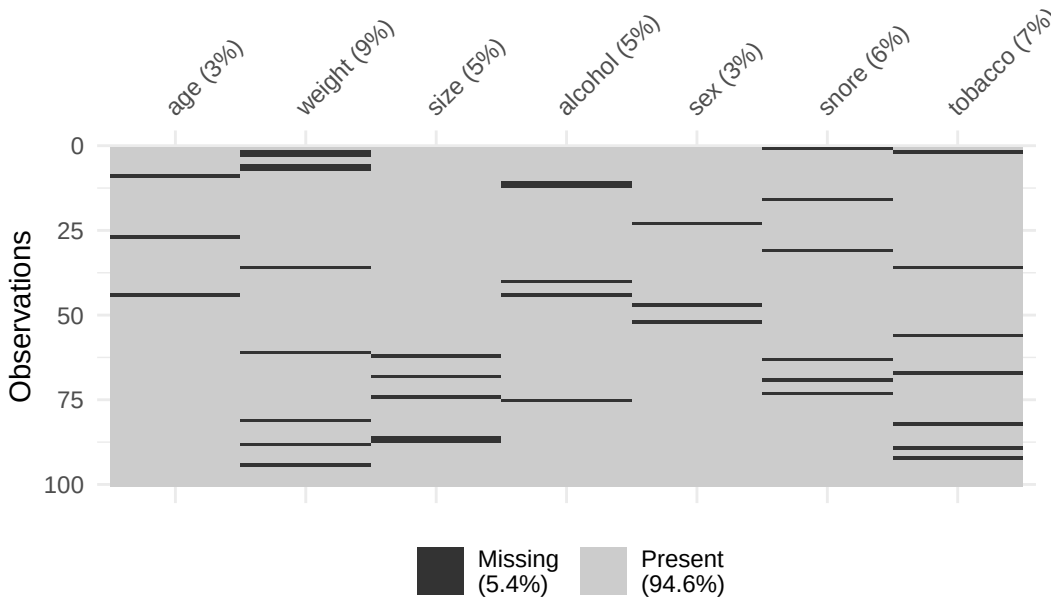
```
ggplot(snorena) +
  aes(x = size, y = weight) +
  geom_miss_point() +
  theme_classic()
```



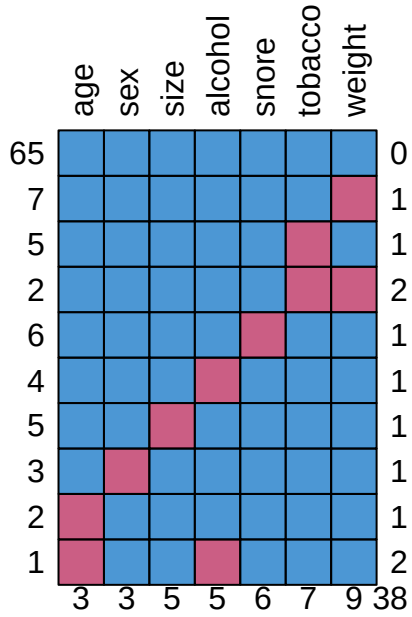
Et il existe d'autres fonctions :

- `vis_miss()` du package `{visdata}`
- `md.pattern()` du package `{mice}`

```
visdat::vis_miss(snorena)
```




```
mice::md.pattern(snorena, rotate.names = TRUE)
```




```
age sex size alcohol snore tobacco weight
```

```
65    1    1    1        1      1        1       1   0
7     1    1    1        1      1        1       0   1
5     1    1    1        1      1        0       1   1
2     1    1    1        1      1        0       0   2
6     1    1    1        1      0        1       1   1
4     1    1    1        0      1        1       1   1
5     1    1    0        1      1        1       1   1
3     1    0    1        1      1        1       1   1
2     0    1    1        1      1        1       1   1
1     0    1    1        0      1        1       1   2
      3    3    5        5      6        7       9  38
```

> ⚠ Détection des valeurs manquantes
>
> Les données manquantes peuvent avoir été remplacées par d'autres. Il est possible de les détecter grâce à l'**analyse descriptive**.

5 L'analyse descriptive

L'analyse descriptive a pour but d'analyser les variables pour connaître la nature des données mais aussi identifier les valeurs extrêmes (à ne pas confondre avec aberrantes).

Utilisation de fonctions rapide comme :

- `skim()` du package `{skimr}`
- `dfSummary()` du package `{summarytools}`
- `create_report()` du package `{DataExplorer}` : `DataExplorer::create_report(snorena)`

```
skimr::skim(snorena)
```

Table 1: Data summary

Name	snorena
Number of rows	100
Number of columns	7
Column type frequency:	
factor	3
numeric	4

Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
sex	3	0.97	FALSE	2	M: 75, W: 22
snore	6	0.94	FALSE	2	N: 62, Y: 32
tobacco	7	0.93	FALSE	2	Y: 61, N: 32

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
age	3	0.97	52.16	11.52	23	43	51	63.0	74	
weight	9	0.91	90.40	18.08	42	77	94	104.5	120	
size	5	0.95	181.09	13.50	158	166	186	194.0	208	
alcohol	5	0.95	2.91	3.36	0	0	2	4.0	15	

```
summarytools::dfSummary(snorena)
```

```
Data Frame Summary
snorena
Dimensions: 100 x 7
Duplicates: 0

-------------------------------------------------------------------------------------------
No   Variable    Stats / Values             Freqs (% of Valid)   Graph                Valid
---- ----------- -------------------------- -------------------- -------------------- ------
1    age         Mean (sd) : 52.2 (11.5)    40 distinct values         . .   :   .     97
     [integer]   min < med < max:                                      : :   :   :     (97.0%
                 23 < 51 < 74                                    : : : . : . : .
                 IQR (CV) : 20 (0.2)                             : : : : : : : :
                                                               . : : : : : : : : :

2    weight      Mean (sd) : 90.4 (18.1)    45 distinct values             : .         91
     [integer]   min < med < max:                                          : :         (91.0%
                 42 < 94 < 120                                       . . : :
                 IQR (CV) : 27.5 (0.2)                             : : : : : .
```

```
                                                          . : : : : : : : :

3    size        Mean (sd) : 181.1 (13.5)   30 distinct values                :       95
     [integer]   min < med < max:                           .                 :       (95.0%
                 158 < 186 < 208                            :               . :
                 IQR (CV) : 28 (0.1)                    : : :     : : : :
                                                        : : : . : : : : :

4    alcohol     Mean (sd) : 2.9 (3.4)      12 distinct values   :                    95
     [integer]   min < med < max:                                :                    (95.0%
                 0 < 2 < 15                                      : .
                 IQR (CV) : 4 (1.2)                              : :
                                                                 : : : : .

5    sex         1. M                       75 (77.3%)           IIIIIIIIIIIIIII      97
     [factor]    2. W                       22 (22.7%)           IIII                 (97.0%

6    snore       1. N                       62 (66.0%)           IIIIIIIIIIIII        94
     [factor]    2. Y                       32 (34.0%)           IIIIII               (94.0%

7    tobacco     1. N                       32 (34.4%)           IIIIII               93
     [factor]    2. Y                       61 (65.6%)           IIIIIIIIIIII         (93.0%
-------------------------------------------------------------------------------------------
```

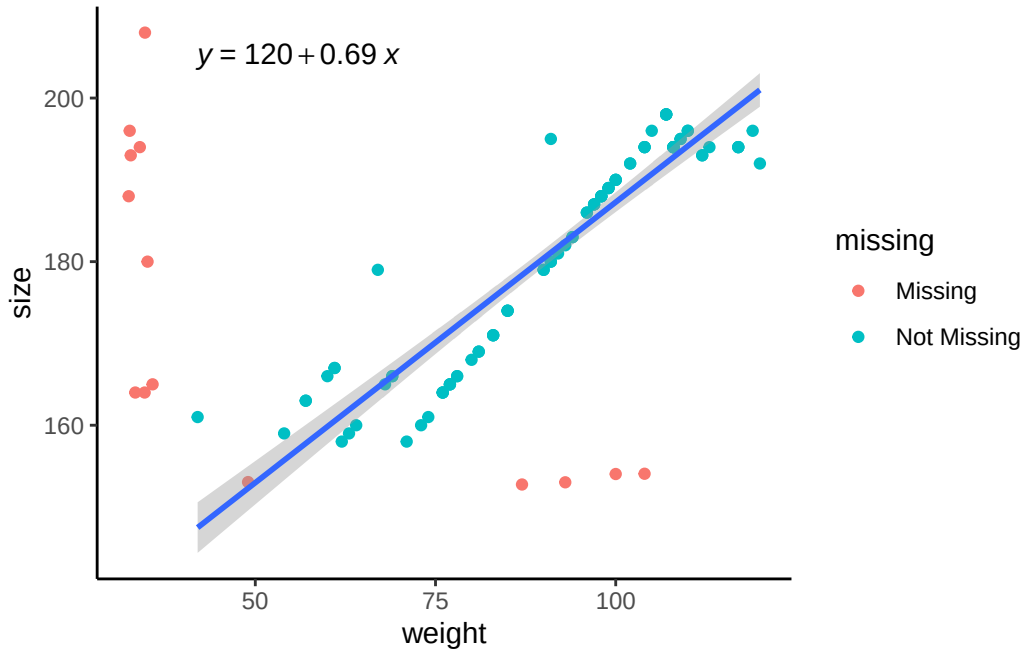
6 Traitement des valeurs manquantes

- Remplacer la donnée manquante par :
  - La vraie valeur s'il est possible de la retrouver.
  - Une valeur de remplacement :
    - Déterminée à partir des autres données de la variables : *moyenne*, *médiane*, *minimum*, *maximum*...
    - Modélisée à partir des autres variables grâce à *une régression linéaire*, *une ACP*...
- Ne pas remplacer la données mais garder le `NA`
- Supprimer la ligne ou la colonne concernée. **Cette solution est la moins envisageable et ne doit être mise en place que si les deux autres ne sont pas possibles.**

```r
# remplacement par régression logistique
snore_pred <- predict(
  regression_logistique_2,
  newdata = snorena |> filter(is.na(snore))
  )

snorena_mod <- snorena |>
  mutate(
    snore =
      case_when(
        is.na(snore) ~ "N",
        TRUE ~ snore
      ) |>
      as.factor()
  )

summary(snorena_mod)
```

```
      age            weight           size          alcohol         sex
 Min.   :23.00   Min.   : 42.0   Min.   :158.0   Min.   : 0.000   M   :75
 1st Qu.:43.00   1st Qu.: 77.0   1st Qu.:166.0   1st Qu.: 0.000   W   :22
 Median :51.00   Median : 94.0   Median :186.0   Median : 2.000   NA's: 3
 Mean   :52.16   Mean   : 90.4   Mean   :181.1   Mean   : 2.905
 3rd Qu.:63.00   3rd Qu.:104.5   3rd Qu.:194.0   3rd Qu.: 4.000
 Max.   :74.00   Max.   :120.0   Max.   :208.0   Max.   :15.000
 NA's   :3       NA's   :9       NA's   :5       NA's   :5
 snore  tobacco
 N:68   N   :32
 Y:32   Y   :61
        NA's: 7
```

```r
# régression linéaire
ggplot(snorena_mod) +
  aes(x = weight, y = size) +
  geom_miss_point() +
  geom_smooth(method = "lm") +
  ggpubr::stat_regline_equation() +
  theme_classic()
```




```
snorena_mod <- snorena_mod |>
  mutate(
    size =
      case_when(
        is.na(size) ~ 120 + 0.69 * weight,
        TRUE ~ size
      )
  )

summary(snorena_mod)
```

```
      age            weight          size          alcohol          sex
 Min.   :23.00   Min.   : 42.0   Min.   :153.8   Min.   : 0.000   M   :75
 1st Qu.:43.00   1st Qu.: 77.0   1st Qu.:166.0   1st Qu.: 0.000   W   :22
 Median :51.00   Median : 94.0   Median :186.0   Median : 2.000   NA's: 3
 Mean   :52.16   Mean   : 90.4   Mean   :181.0   Mean   : 2.905
 3rd Qu.:63.00   3rd Qu.:104.5   3rd Qu.:194.0   3rd Qu.: 4.000
 Max.   :74.00   Max.   :120.0   Max.   :208.0   Max.   :15.000
 NA's   :3       NA's   :9                       NA's   :5
 snore  tobacco
 N:68   N   :32
 Y:32   Y   :61
```

```
        NA's: 7
```

```
# regression linéaire
reg_lin <- lm(weight ~ size, data = snorena)
reg_lin$coefficients
```

```
(Intercept)        size
 -136.42657     1.25551
```

```
snorena_mod <- snorena_mod |>
  mutate(
    weight =
      case_when(
        is.na(weight) ~
          reg_lin$coefficients[1] + reg_lin$coefficients[2] * size,
        TRUE ~ weight
      )
  )

summary(snorena_mod)
```

```
      age            weight           size          alcohol         sex
 Min.   :23.00   Min.   : 42.00   Min.   :153.8   Min.   : 0.000   M   :75
 1st Qu.:43.00   1st Qu.: 77.00   1st Qu.:166.0   1st Qu.: 0.000   W   :22
 Median :51.00   Median : 95.00   Median :186.0   Median : 2.000   NA's: 3
 Mean   :52.16   Mean   : 90.72   Mean   :181.0   Mean   : 2.905
 3rd Qu.:63.00   3rd Qu.:106.17   3rd Qu.:194.0   3rd Qu.: 4.000
 Max.   :74.00   Max.   :124.72   Max.   :208.0   Max.   :15.000
 NA's   :3                                        NA's   :5
 snore  tobacco
 N:68   N   :32
 Y:32   Y   :61
        NA's: 7
```

`replace_na()` du package `{tidyr}`

Lors du live j'ai oublié de présenter la fonction `replace_na()` du package `{tidyr}` ! Cette fonction permet de remplacer les valeurs manquantes d'une colonne ou plusieurs colonnes par une valeur spécifique.

7 En savoir un peu plus sur moi

Bonjour,

Je suis Marie Vaugoyeau et je suis disponible pour des **missions en freelance** d'**accompagnement à la formation** à R et à l'analyse de données et/ou en **programmation** (reprise de scripts, bonnes pratiques de codage, développement de package). Ayant un **bagage recherche en écologie**, j'ai accompagné plusieurs chercheuses en biologie dans leurs analyses de données mais je suis ouverte à d'autres domaines.

Vous pouvez retrouver mes offres ici.

En plus de mes missions de consulting je diffuse mes savoirs en R et analyse de données sur plusieurs plateformes :

- J'ai écrit un **livre** aux éditions ENI
- Tous les mois je fais un **live sur Twitch** pour parler d'un package de R, d'une analyse
- Je rédige une **newsletter** de manière irrégulière pour parler de mes **inspirations** et transmettre **des trucs et astuces sur R**. Pour s'y inscrire, c'est par là. J'ai aussi un **blog** sur lequel vous pourrez retrouver une version de cet article.

Pour en savoir encore un peu plus sur moi, il y a LinkedIn et pour retrouver tous ces liens et plus encore, c'est ici

N'hésitez pas à me contacter sur marie.vaugoyeau@gmail.com !

Bonne journée

Marie