

Rapport Final statistique en grande dimension

Amadou BAH ~ Frederic AKADJE

1. Chargement et exploration du jeu de données

Chargement du jeu de données

```
# Exemple : jeu de données simulé
set.seed(123456789)
n <- 72
p <- 1000
X <- matrix(rnorm(n * p), n, p)
colnames(X) <- paste0("V", 1:p)
y <- factor(sample(c("ALL", "AML"), n, replace = TRUE))
#
# Vérifier les dimensions
dim(X)
```

```
[1] 72 1000
```

Exploration des données

Résumé des dimensions du jeu de données : Compte tenu du nombre important de variables ici, une disposition particulière du jeu de données sera adoptée pour la réalisation de la description du jeu de données.

```
library(tidyverse)
library(gt)
# Transformation du jeu de données en dataFrame
df <- bind_cols(X, Y=y)
# Nombre de valeurs manquantes par variable
na_<- df |>
```

```

select(-Y)|> # Retrait de la colonne cible
summarise_all(list(na = ~sum(is.na(.)))) |> # Aggrégation sur chaque colonne
t() # Transposition (format vecteur)

# Moyenne des variables
mean_ <- df |>
  select(-Y)|>
  summarise_all(list(mean = ~mean(.x,na.rm = TRUE))) |>
  t()

# Maximum par colonne
max_ <- df |>
  select(-Y)|>
  summarise_all(list(max = ~max(.x))) |>
  t()

# Médiabe par colonne
median_ <- df |>
  select(-Y)|>
  summarise_all(list(max = ~max(.x))) |>
  t()

# Minimum par colonne
min_ <- df |>
  select(-Y)|>
  summarise_all(list(min = ~min(.x))) |>
  t()

# Ecart-type par colonne
sd_ <- df |>
  select(-Y)|>
  summarise_all(list(sd = ~sd(.x))) |>
  t()

# Tableau de restitution pour la description
restitution <- bind_cols(Variables = paste0("V", 1:p), # Concaténation de toutes les agrégat.
  NA_ =na_ ,
  Max_ = max_,
  Min_ = min_,
  Mean_ = mean_,
  Median_ = median_,
  SD_ = sd_ )

```

```
restitution |>
  head() # Affichage des 5 premières lignes
```

```
# A tibble: 6 x 7
  Variables NA_[,1] Max_[,1] Min_[,1] Mean_[,1] Median_[,1] SD_[,1]
  <chr>      <int>    <dbl>    <dbl>    <dbl>      <dbl>    <dbl>
1 V1         0      1.99     -2.28    -0.0642     1.99     0.995
2 V2         0      2.25     -1.47     0.234      2.25     0.887
3 V3         0      2.19     -2.17     0.147      2.19     1.03
4 V4         0      3.47     -2.20    -0.123      3.47     1.20
5 V5         0      1.96     -3.21    -0.102      1.96     0.937
6 V6         0      2.48     -2.36     0.236      2.48     1.08
```

```
# gt() # Pour format plus esthétiques
```