# ProjetR WaterStress

Yosra Hammoudi et Rihab Chaari M1DSSD

2023-12-30

*Au cours de ce projet nous allons analyser la variation du stress hydrique (La variable stress hydrique fait référence à une condition dans laquelle une plante subit un manque d'eau suffisant pour affecter son métabolisme et son développement. En d'autres termes, le stress hydrique se produit lorsque la quantité d'eau disponible pour une plante est insuffisante pour répondre à ses besoins physiologiques normaux.)*

*Nous avons tiré la base de données depuis AQUASTAT qui est un Système d'information mondial de la FAO sur l'eau et l'agriculture.*

## 1- IMPORTATION DES DONNEES

```
# Charger le package readxl pour la lecture des fichiers Excel
library(readxl)
```

```
## Warning: le package 'readxl' a été compilé avec la version R 4.3.2
```

```
# Spécifier le chemin du fichier Excel de notre base de données
chemin_excel <- "C:/Users/rihab/Desktop/ProjetR/base/AQUASTAT-Dissemination-System.xlsx"

# Lire le fichier Excel et stocker les données dans un objet appelé donnees_excel
donnees_excel <- read_excel(chemin_excel)

# Afficher les premières lignes des données pour examiner la structure initiale
head(donnees_excel)
```

```
## # A tibble: 6 x 6
##   Country     Variable                                  Unit  Symbol `2000` `2020`
##   <chr>       <chr>                                     <chr> <chr>  <chr>  <chr>
## 1 Afghanistan Agricultural water withdrawal as % of ~ %     I      30.61  30.61
## 2 Afghanistan MDG 7.5. Freshwater withdrawal as % of~ %     I      31.05  31.05
## 3 Afghanistan SDG 6.4.1. Industrial Water Use Effici~ US$/~ E      9.33   12.67
## 4 Afghanistan SDG 6.4.1. Irrigated Agriculture Water~ US$/~ E      0.06   0.12
## 5 Afghanistan SDG 6.4.1. Services Water Use Efficien~ US$/~ E      6.92   57.21
## 6 Afghanistan SDG 6.4.1. Water Use Efficiency         US$/~ I      0.18   0.8
```

## 2- PREPROCESSING

```
#Effectuer un filtrage des données en ciblant spécifiquement les lignes associées
#au "Water Stress". La colonne "Variable" de notre base de données contient
#des informations diverses, et nous utilisons la fonction subset pour extraire
#uniquement les lignes où la variable est définie comme "SDG 6.4.2. Water Stress".
```

```r
# Filtrer les lignes avec "Water Stress" dans la colonne "Variable"
donnees_water_stress <- subset(donnees_excel, Variable == "SDG 6.4.2. Water Stress")

# Afficher les données filtrées
View(donnees_water_stress)
print(donnees_water_stress)
```

```
## # A tibble: 190 x 6
##    Country             Variable                Unit  Symbol `2000` `2020`
##    <chr>               <chr>                   <chr> <chr>  <chr>  <chr>
##  1 Afghanistan         SDG 6.4.2. Water Stress %     E      54.76  54.76
##  2 Albania             SDG 6.4.2. Water Stress %     E      11.04  4.72
##  3 Algeria             SDG 6.4.2. Water Stress %     E      79.26  137.92
##  4 Angola              SDG 6.4.2. Water Stress %     E      1.7    1.87
##  5 Antigua and Barbuda SDG 6.4.2. Water Stress %     I      7.63   8.46
##  6 Argentina           SDG 6.4.2. Water Stress %     E      8.42   10.46
##  7 Armenia             SDG 6.4.2. Water Stress %     E      37.74  57.09
##  8 Australia           SDG 6.4.2. Water Stress %     E      8.71   3.47
##  9 Austria             SDG 6.4.2. Water Stress %     E      10.05  9.64
## 10 Azerbaijan          SDG 6.4.2. Water Stress %     E      48.53  55.6
## # i 180 more rows
```

```r
# vérification  de valeurs manquantes (NA) dans les colonnes "2000" et "2020"
# de notre ensemble de données filtrées sur "Water Stress".
#La fonction is.na() est utilisée pour identifier les valeurs NA,
#et la fonction any() est utilisée pour déterminer si au moins une valeur manquante
#est présente dans la colonne.

# Vérifier s'il y a des valeurs NA dans la colonne "2000"
presence_na_2000 <- any(is.na(donnees_water_stress$`2000`))

# Afficher le résultat "boolean"
print(presence_na_2000)
```

```
## [1] TRUE
```

```r
# Vérifier s'il y a des valeurs NA dans la colonne "2000"
presence_na_2020 <- any(is.na(donnees_water_stress$`2020`))

# Afficher le résultat "boolean"
print(presence_na_2020)
```

```
## [1] TRUE
```

```r
#Effectuer le comptage du nombre de valeurs manquantes (NA) dans les colonnes
#"2000" et "2020" de notre ensemble de données filtrées sur "Water Stress".
# Compter le nombre de valeurs NA dans la colonne "2000"
nb_na_2000 <- sum(is.na(donnees_water_stress$`2000`))

# Afficher le nombre de valeurs NA
print(nb_na_2000)
```

```
## [1] 34
```

```
# Compter le nombre de valeurs NA dans la colonne "2000"
nb_na_2020 <- sum(is.na(donnees_water_stress$`2020`))

# Afficher le nombre de valeurs NA
print(nb_na_2020)
```

```
## [1] 12
```

```
# filtre les lignes de la base de données  et conserve uniquement celles
#qui contiennent au moins une valeur manquante (NA) dans l'une de leurs colonnes.
data_null_values <- subset(donnees_water_stress, apply(donnees_water_stress, 1,
                                                       function(row) any(is.na(row))))
# Affichez le résultat
print(data_null_values)
```

```
## # A tibble: 46 x 6
##    Country    Variable                Unit  Symbol `2000` `2020`
##    <chr>      <chr>                   <chr> <chr>  <chr>  <chr>
##  1 Bahrain    SDG 6.4.2. Water Stress %     E      248.91 <NA>
##  2 Bahrain    SDG 6.4.2. Water Stress %     I      <NA>   133.71
##  3 Bangladesh SDG 6.4.2. Water Stress %     E      <NA>   5.72
##  4 Barbados   SDG 6.4.2. Water Stress %     E      101.25 <NA>
##  5 Barbados   SDG 6.4.2. Water Stress %     I      <NA>   87.5
##  6 Bhutan     SDG 6.4.2. Water Stress %     E      <NA>   1.41
##  7 Cabo Verde SDG 6.4.2. Water Stress %     E      <NA>   8.43
##  8 Cambodia   SDG 6.4.2. Water Stress %     E      <NA>   1.04
##  9 Costa Rica SDG 6.4.2. Water Stress %     E      <NA>   5.35
## 10 Croatia    SDG 6.4.2. Water Stress %     E      <NA>   1.48
## # i 36 more rows
```

```
# Identifier les doublons basés sur la colonne "country"
doublons_country <- data_null_values[duplicated(data_null_values$Country)
                                     | duplicated(data_null_values$Country, fromLast = TRUE), ]

# Afficher les doublons
View(doublons_country)
print(doublons_country)
```

```
## # A tibble: 24 x 6
##    Country  Variable                Unit  Symbol `2000` `2020`
##    <chr>    <chr>                   <chr> <chr>  <chr>  <chr>
##  1 Bahrain  SDG 6.4.2. Water Stress %     E      248.91 <NA>
##  2 Bahrain  SDG 6.4.2. Water Stress %     I      <NA>   133.71
##  3 Barbados SDG 6.4.2. Water Stress %     E      101.25 <NA>
##  4 Barbados SDG 6.4.2. Water Stress %     I      <NA>   87.5
##  5 Djibouti SDG 6.4.2. Water Stress %     E      6.33   <NA>
##  6 Djibouti SDG 6.4.2. Water Stress %     I      <NA>   6.33
##  7 Dominica SDG 6.4.2. Water Stress %     E      8.3    <NA>
##  8 Dominica SDG 6.4.2. Water Stress %     I      <NA>   10
##  9 Fiji     SDG 6.4.2. Water Stress %     E      0.29   <NA>
## 10 Fiji     SDG 6.4.2. Water Stress %     I      <NA>   0.3
## # i 14 more rows
```

```r
# Utiliser le package dplyr pour combiner les doublons basés sur la colonne "Country".

library(dplyr)
```

```
## Warning: le package 'dplyr' a été compilé avec la version R 4.3.2
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
donnees_combine <- doublons_country %>%
  group_by(Country) %>%
  summarise(across(c("2000", "2020"), ~na.omit(.)[1]))

# Afficher les données combinées
View(donnees_combine)
print(donnees_combine)
```

```
## # A tibble: 12 x 3
##    Country      `2000` `2020`
##    <chr>        <chr>  <chr>
##  1 Bahrain      248.91 133.71
##  2 Barbados     101.25 87.5
##  3 Djibouti     6.33   6.33
##  4 Dominica     8.3    10
##  5 Fiji         0.29   0.3
##  6 Libya        615.43 817.14
##  7 Maldives     17.67  15.67
##  8 Malta        61.27  81.86
##  9 Oman         92.86  116.71
## 10 Qatar        282.59 431.03
## 11 Saudi Arabia 819.76 974.17
## 12 Yemen        161.14 169.76
```

```r
# Utiliser le package dplyr pour combiner les données filtrées sur "Water Stress"
#en se basant sur la colonne "Country".
library(dplyr)

donnees_combine_waterStress <- donnees_water_stress %>%
  group_by(Country) %>%
  summarise(across(c("2000", "2020"), ~na.omit(.)[1]))

# Afficher les données combinées
View(donnees_combine_waterStress)
print(donnees_combine_waterStress)
```

```
## # A tibble: 178 x 3
##    Country             `2000` `2020`
##    <chr>               <chr>  <chr>
##  1 Afghanistan         54.76  54.76
##  2 Albania             11.04  4.72
##  3 Algeria             79.26  137.92
##  4 Angola              1.7    1.87
##  5 Antigua and Barbuda 7.63   8.46
##  6 Argentina           8.42   10.46
##  7 Armenia             37.74  57.09
##  8 Australia           8.71   3.47
##  9 Austria             10.05  9.64
## 10 Azerbaijan          48.53  55.6
## # i 168 more rows
```

```
#Compter le nombre de valeurs manquantes (NA) dans les colonnes "2000" et "2020"
#après la combinaison des données filtrées sur "water Stress".

missing_values_2000 <- sum(is.na(donnees_combine_waterStress$`2000`) )
print(missing_values_2000)
```

```
## [1] 22
```

```
missing_values_2020 <- sum(is.na(donnees_combine_waterStress$`2020`) )
print(missing_values_2020)
```

```
## [1] 0
```

```
#Utiliser la fonction na.omit() pour supprimer les lignes qui contiennent des valeurs
#manquantes dans les données combinées sur "water Stress".
donnees_combine_waterStress_propre<- na.omit(donnees_combine_waterStress)
print(donnees_combine_waterStress_propre)
```

```
## # A tibble: 156 x 3
##    Country             `2000` `2020`
##    <chr>               <chr>  <chr>
##  1 Afghanistan         54.76  54.76
##  2 Albania             11.04  4.72
##  3 Algeria             79.26  137.92
##  4 Angola              1.7    1.87
##  5 Antigua and Barbuda 7.63   8.46
##  6 Argentina           8.42   10.46
##  7 Armenia             37.74  57.09
##  8 Australia           8.71   3.47
##  9 Austria             10.05  9.64
## 10 Azerbaijan          48.53  55.6
## # i 146 more rows
```

```
View(donnees_combine_waterStress_propre)
```

```r
#Compter nombre de valeurs manquantes (NA) pour vérifier qu'aucune valeur NaN ne subsiste dans notre ba
missing_values_2000 <- sum(is.na(donnees_combine_waterStress_propre$`2000`) )
print(missing_values_2000)
```

```
## [1] 0
```

```r
missing_values_2020 <- sum(is.na(donnees_combine_waterStress_propre$`2020`) )
print(missing_values_2020)
```

```
## [1] 0
```

```r
# Charger le package
library(dplyr)

# Ajouter la colonne "Variable" avec la valeur "SDG 6.4.2. Water Stress"
data_null_values_proper <- donnees_combine_waterStress_propre %>%
  mutate(Variable = "SDG 6.4.2. Water Stress")

# Afficher les données avec la nouvelle colonne
View(data_null_values_proper)
print(data_null_values_proper)
```

```
## # A tibble: 156 x 4
##    Country             `2000` `2020` Variable
##    <chr>               <chr>  <chr>  <chr>
##  1 Afghanistan         54.76  54.76  SDG 6.4.2. Water Stress
##  2 Albania             11.04  4.72   SDG 6.4.2. Water Stress
##  3 Algeria             79.26  137.92 SDG 6.4.2. Water Stress
##  4 Angola              1.7    1.87   SDG 6.4.2. Water Stress
##  5 Antigua and Barbuda 7.63   8.46   SDG 6.4.2. Water Stress
##  6 Argentina           8.42   10.46  SDG 6.4.2. Water Stress
##  7 Armenia             37.74  57.09  SDG 6.4.2. Water Stress
##  8 Australia           8.71   3.47   SDG 6.4.2. Water Stress
##  9 Austria             10.05  9.64   SDG 6.4.2. Water Stress
## 10 Azerbaijan          48.53  55.6   SDG 6.4.2. Water Stress
## # i 146 more rows
```

## 3- MANIPILATION DES DONNEES

### a- Creation d'une carte pour montrer la valeurs de stress hydrique dans chaque pays en 2000

```r
library(sf)
```

```
## Warning: le package 'sf' a été compilé avec la version R 4.3.2
```

```
## Linking to GEOS 3.11.2, GDAL 3.7.2, PROJ 9.3.0; sf_use_s2() is TRUE
```

```r
library(ggplot2)
```

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.3.2
```

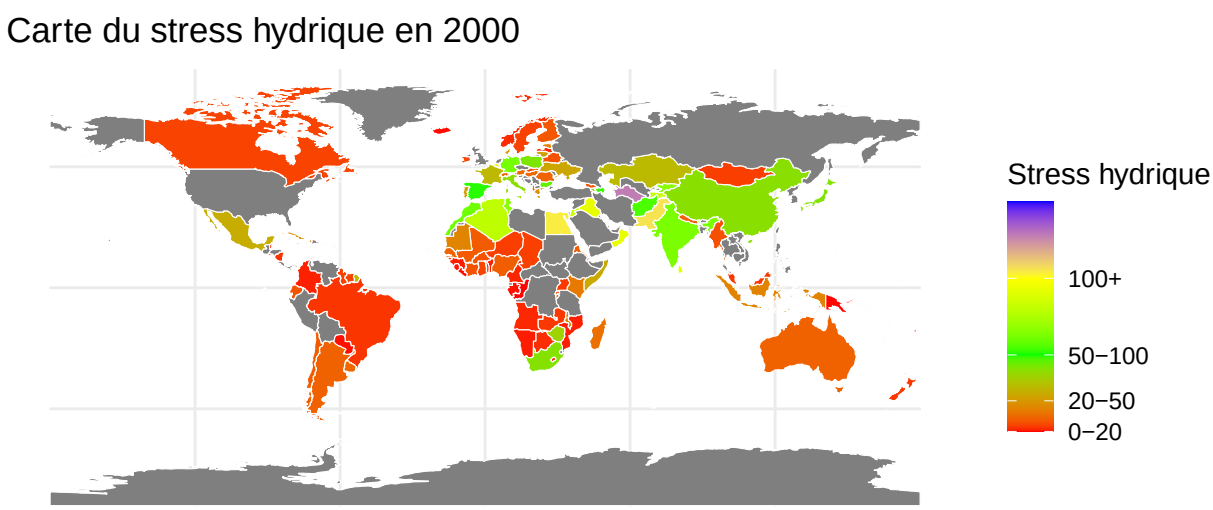
```r
library(tidyr)
```

```
## Warning: le package 'tidyr' a été compilé avec la version R 4.3.2
```

```r
library(rnaturalearth)
```

```
## Warning: le package 'rnaturalearth' a été compilé avec la version R 4.3.2
```

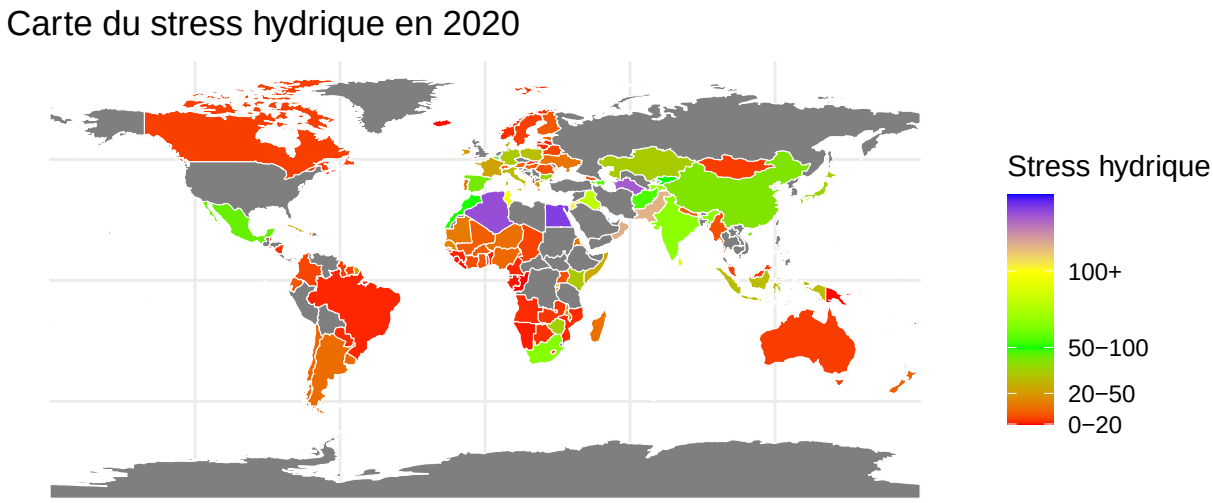
```r
# charger la carte du monde
world_map <- ne_countries(scale = "medium", returnclass = "sf")

# Sélectionner les colonnes nécessaires pour la carte
selected_cols <- c("Country", "2000")
data_for_map <- data_null_values_proper[selected_cols]

# Convertir les valeurs "2000" en numérique
data_for_map$`2000` <- as.numeric(data_for_map$`2000`)

# Fusionner les données avec la carte mondiale
merged_data <- merge(world_map, data_for_map, by.x = "name", by.y = "Country", all.x = TRUE)

# Créer la carte avec ggplot2 et une échelle de couleur continue
ggplot() +
  geom_sf(data = merged_data, aes(fill = `2000`), color = "white", lwd = 0.2) +
  scale_fill_gradientn(colors = c("red", "green", "yellow", "blue"),
                       breaks = c(0, 20, 50, 100),
                       labels = c("0-20", "20-50", "50-100", "100+"),
                       limits = c(0, 150),
                       name = "Stress hydrique") +
  theme_minimal() +
  labs(title = "Carte du stress hydrique en 2000")
```




b- Creation d'une carte pour montrer la valeurs de stress hydrique dans chaque pays en 2020

```
library(sf)
library(ggplot2)
library(tidyr)
library(rnaturalearth)

#charger la carte mondiale
world_map <- ne_countries(scale = "medium", returnclass = "sf")

# Sélectionner les colonnes nécessaires pour la carte
selected_cols <- c("Country", "2020")
data_for_map <- data_null_values_proper[selected_cols]

# Convertir les valeurs "2020" en numérique
data_for_map$`2020` <- as.numeric(data_for_map$`2020`)

# Fusionner les données avec la carte mondiale
merged_data <- merge(world_map, data_for_map, by.x = "name", by.y = "Country", all.x = TRUE)

# Créer la carte avec ggplot2 et une échelle de couleur continue
ggplot() +
  geom_sf(data = merged_data, aes(fill = `2020`), color = "white", lwd = 0.2) +
  scale_fill_gradientn(colors = c("red", "green", "yellow", "blue"),
                       breaks = c(0, 20, 50, 100),
                       labels = c("0-20", "20-50", "50-100", "100+"),
```

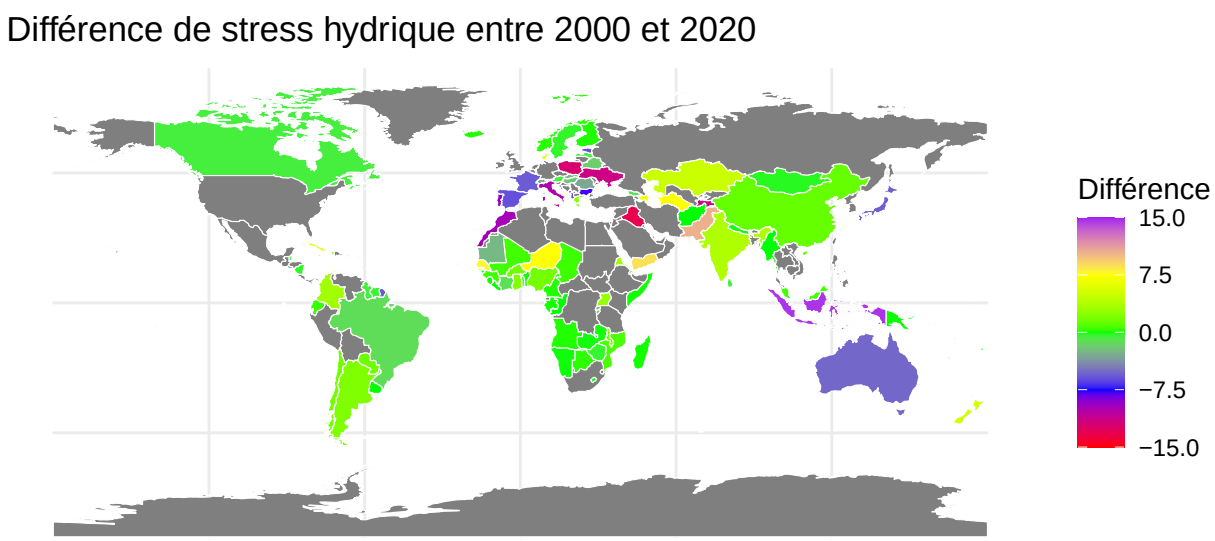
```
                      limits = c(0, 150),  # Ajuster les limites en fonction de vos données
                      name = "Stress hydrique") +
  theme_minimal() +
  labs(title = "Carte du stress hydrique en 2020")
```




c- Creer une carte pour montrer la difference entre 2000 et 2020

```
library(sf)
library(ggplot2)
library(tidyr)
library(rnaturalearth)

# Charger les données de la carte mondiale
world_map <- ne_countries(scale = "medium", returnclass = "sf")

# Sélectionner les colonnes nécessaires pour la carte
selected_cols <- c("Country", "2000", "2020")
data_for_map <- data_null_values_proper [selected_cols]

# Convertir les valeurs "2000" et "2020" en numérique
data_for_map$`2000` <- as.numeric(data_for_map$`2000`)
data_for_map$`2020` <- as.numeric(data_for_map$`2020`)

# Ajouter une colonne représentant la différence entre les deux années
data_for_map$Difference <- data_for_map$`2020` - data_for_map$`2000`
```

```r
# Fusionner les données avec la carte mondiale
merged_data_diff <- merge(world_map, data_for_map, by.x = "name", by.y = "Country", all.x = TRUE)

# Créer la carte de différence avec ggplot2
ggplot() +
  geom_sf(data = merged_data_diff, aes(fill = Difference), color = "white", lwd = 0.2) +
  scale_fill_gradientn(colors = c("red", "blue", "green", "yellow", "purple"),
                       breaks = c(-15, -7.5, 0, 7.5, 15),  # Ajustez les valeurs
                                                           #en fonction de vos données
                       limits = c(-15, 15),  # Ajustez la plage de valeurs
                       name = "Différence") +
  theme_minimal() +
  labs(title = "Différence de stress hydrique entre 2000 et 2020")
```




d- Creer un tableau contenant les pays selon la catégorie de différence

```r
# Charger la bibliothèque knitr
library(knitr)

# Calculer la différence de stress hydrique
data_null_values_proper$Difference<-
                              as.numeric(data_null_values_proper$`2020`)-
                              as.numeric(data_null_values_proper$`2000`)

# Créer des intervalles
breaks <- c(-Inf, -15, -7.5, 0, 7.5, 15, Inf)
```

```r
labels <- c("<-15", "-15 à -7.5", "-7.5 à 0", "0 à 7.5", "7.5 à 15", ">15")

# Catégoriser les différences
data_null_values_proper$Category <- cut(data_null_values_proper$Difference,
                                        breaks = breaks, labels = labels, include.lowest = TRUE)

# Préparer une liste pour chaque catégorie
categories_list <- setNames(vector("list", length(labels)), labels)

# Remplir la liste avec les noms des pays
for (label in labels) {
    categories_list[[label]] <- data_null_values_proper$Country[
                                            data_null_values_proper$Category == label]
}

# Trouver le nombre maximum de pays dans une catégorie
max_length <- max(sapply(categories_list, length))

# Standardiser la longueur de chaque vecteur dans la liste
categories_list <- lapply(categories_list, function(x) c(x, rep(NA, max_length - length(x))))

# Convertir la liste en data frame sans modifier les noms de colonne
categories_table <- setNames(data.frame(categories_list, check.names = FALSE), labels)

# Afficher le tableau
kable(categories_table, "markdown")
```

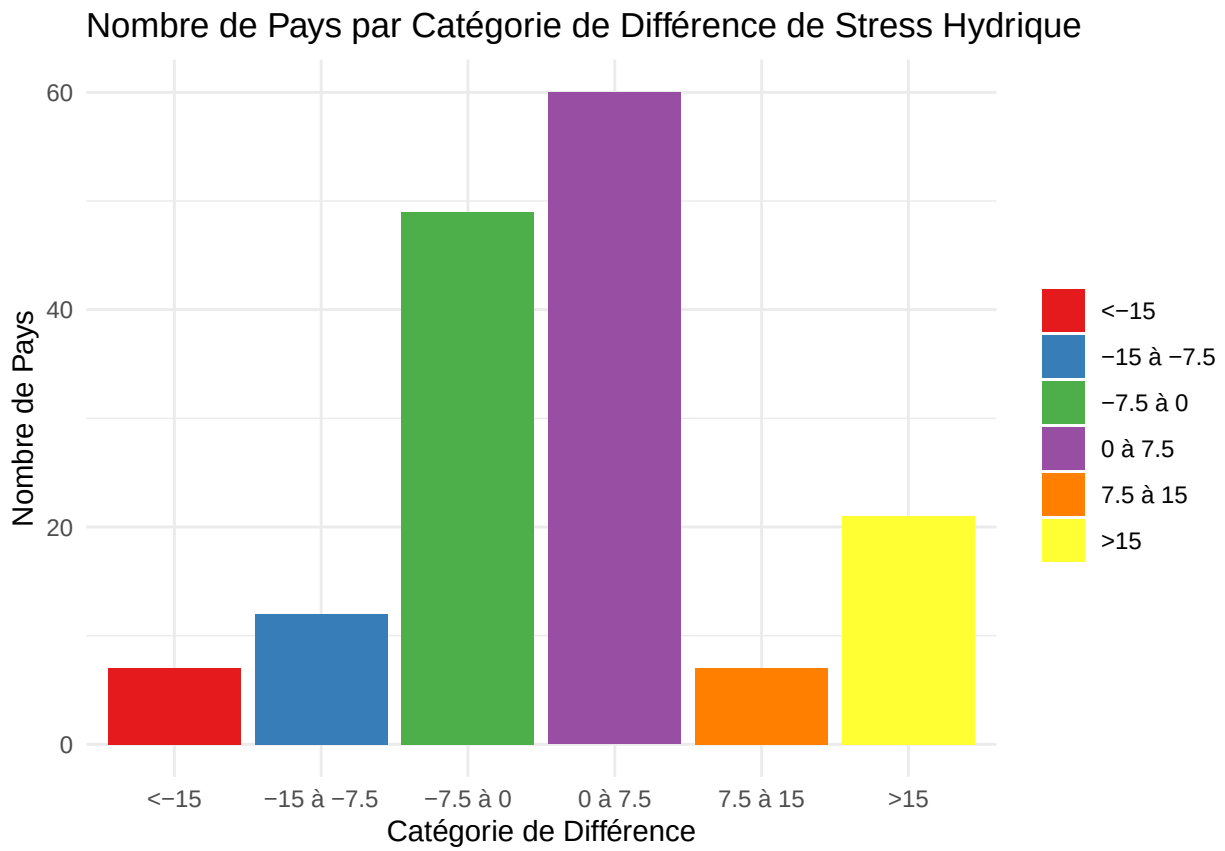
| <-15 | -15 à -7.5 | -7.5 à 0 | 0 à 7.5 | 7.5 à 15 | >15 |
|---|---|---|---|---|---|
| Bahrain | Barbados | Afghanistan | Angola | Dominican Republic | Algeria |
| Belgium | Bulgaria | Albania | Antigua and Barbuda | Indonesia | Armenia |
| Germany | Czechia | Australia | Argentina | Pakistan | Cyprus |
| Israel | Iraq | Austria | Azerbaijan | Palestine | Egypt |
| Kyrgyzstan | Italy | Belarus | Bolivia (Plurinational State of) | Senegal | Ireland |
| Lithuania | Morocco | Belize | Botswana | Türkiye | Jordan |
| Singapore | Poland | Benin | Burkina Faso | Yemen | Kenya |
| NA | Portugal | Bosnia and Herzegovina | Cameroon | NA | Kuwait |
| NA | Republic of Moldova | Brazil | Central African Republic | NA | Lebanon |
| NA | Syrian Arab Republic | Brunei Darussalam | Chad | NA | Libya |
| NA | Tajikistan | Burundi | Chile | NA | Malta |
| NA | Ukraine | Canada | China | NA | Mexico |
| NA | NA | Comoros | Colombia | NA | North Macedonia |
| NA | NA | Congo | Cuba | NA | Oman |
| NA | NA | Côte d'Ivoire | Democratic People's Republic of Korea | NA | Qatar |
| NA | NA | Djibouti | Democratic Republic of the Congo | NA | Rwanda |

| <-15 | -15 à -7.5 | -7.5 à 0 | 0 à 7.5 | 7.5 à 15 | >15 |
|---|---|---|---|---|---|
| NA | NA | El Salvador | Denmark | NA | Saudi Arabia |
| NA | NA | Estonia | Dominica | NA | South Africa |
| NA | NA | France | Ecuador | NA | Tunisia |
| NA | NA | Georgia | Equatorial Guinea | NA | United Arab Emirates |
| NA | NA | Guinea-Bissau | Eritrea | NA | Uzbekistan |
| NA | NA | Guyana | Eswatini | NA | NA |
| NA | NA | Hungary | Fiji | NA | NA |
| NA | NA | Japan | Finland | NA | NA |
| NA | NA | Latvia | Gabon | NA | NA |
| NA | NA | Lesotho | Gambia | NA | NA |
| NA | NA | Luxembourg | Ghana | NA | NA |
| NA | NA | Maldives | Greece | NA | NA |
| NA | NA | Mauritania | Guinea | NA | NA |
| NA | NA | Mauritius | Haiti | NA | NA |
| NA | NA | Mongolia | Iceland | NA | NA |
| NA | NA | Myanmar | India | NA | NA |
| NA | NA | Namibia | Iran (Islamic Republic of) | NA | NA |
| NA | NA | Nepal | Jamaica | NA | NA |
| NA | NA | Nicaragua | Kazakhstan | NA | NA |
| NA | NA | Romania | Liberia | NA | NA |
| NA | NA | Russian Federation | Madagascar | NA | NA |
| NA | NA | Saint Vincent and the Grenadines | Malawi | NA | NA |
| NA | NA | Slovakia | Malaysia | NA | NA |
| NA | NA | Spain | Mali | NA | NA |
| NA | NA | Sri Lanka | Mozambique | NA | NA |
| NA | NA | Suriname | Netherlands (Kingdom of the) | NA | NA |
| NA | NA | Sweden | New Zealand | NA | NA |
| NA | NA | Switzerland | Niger | NA | NA |
| NA | NA | United Kingdom of Great Britain and Northern Ireland | Nigeria | NA | NA |
| NA | NA | United States of America | Norway | NA | NA |
| NA | NA | Uruguay | Papua New Guinea | NA | NA |
| NA | NA | Zambia | Paraguay | NA | NA |
| NA | NA | Zimbabwe | Puerto Rico | NA | NA |
| NA | NA | NA | Republic of Korea | NA | NA |
| NA | NA | NA | Sao Tome and Principe | NA | NA |
| NA | NA | NA | Sierra Leone | NA | NA |
| NA | NA | NA | Somalia | NA | NA |
| NA | NA | NA | Togo | NA | NA |
| NA | NA | NA | Trinidad and Tobago | NA | NA |
| NA | NA | NA | Turkmenistan | NA | NA |
| NA | NA | NA | Uganda | NA | NA |
| NA | NA | NA | United Republic of Tanzania | NA | NA |

| <-15 | -15 à -7.5 | -7.5 à 0 | 0 à 7.5 | 7.5 à 15 | >15 |
|---|---|---|---|---|---|
| NA | NA | NA | Venezuela (Bolivarian Republic of) | NA | NA |
| NA | NA | NA | Viet Nam | NA | NA |

e- Visualiser les nombres de pays par catégories de différence avec un histogramme

```r
# Assurez-vous que ggplot2 est chargé
library(ggplot2)

# Calculer le nombre de pays par catégorie
country_counts <- table(data_null_values_proper$Category)

# Transformer en data frame
country_counts_df <- as.data.frame(country_counts)
names(country_counts_df) <- c("Category", "Number_of_Countries")

# Créer l'histogramme avec des couleurs
ggplot(country_counts_df, aes(x = Category, y = Number_of_Countries, fill = Category)) +
  geom_bar(stat = "identity") +  # Utiliser geom_bar pour un histogramme
  scale_fill_brewer(palette = "Set1") +  # Utiliser une palette de couleurs prédéfinie
  theme_minimal() +
  labs(title = "Nombre de Pays par Catégorie de Différence de Stress Hydrique",
       x = "Catégorie de Différence",
       y = "Nombre de Pays") +
  theme(legend.title = element_blank())  # Supprimer le titre de la légende
```




f- Cocher la nature d'evolution de stress hydrique par pays

```
# Sélectionner les colonnes nécessaires
selected_cols <- c("Country", "2000", "2020")

# Créer une nouvelle colonne indiquant si la valeur a augmenté
data_null_values_proper$augmentation <- ifelse(data_null_values_proper$`2020`
                                               > data_null_values_proper$`2000`, "X", "")

# Créer une nouvelle colonne indiquant si la valeur a diminué
data_null_values_proper$diminution <- ifelse(data_null_values_proper$`2020`
                                             < data_null_values_proper$`2000`, "X", "")

# Créer une nouvelle colonne indiquant si la valeur est stable
data_null_values_proper$stable <- ifelse(data_null_values_proper$`2020`
                                         == data_null_values_proper$`2000`, "X", "")

# Sélectionner les colonnes nécessaires pour le tableau final
result_table <- data_null_values_proper[, c("Country", "augmentation", "diminution", "stable")]

# Afficher le tableau final
print(result_table)
```

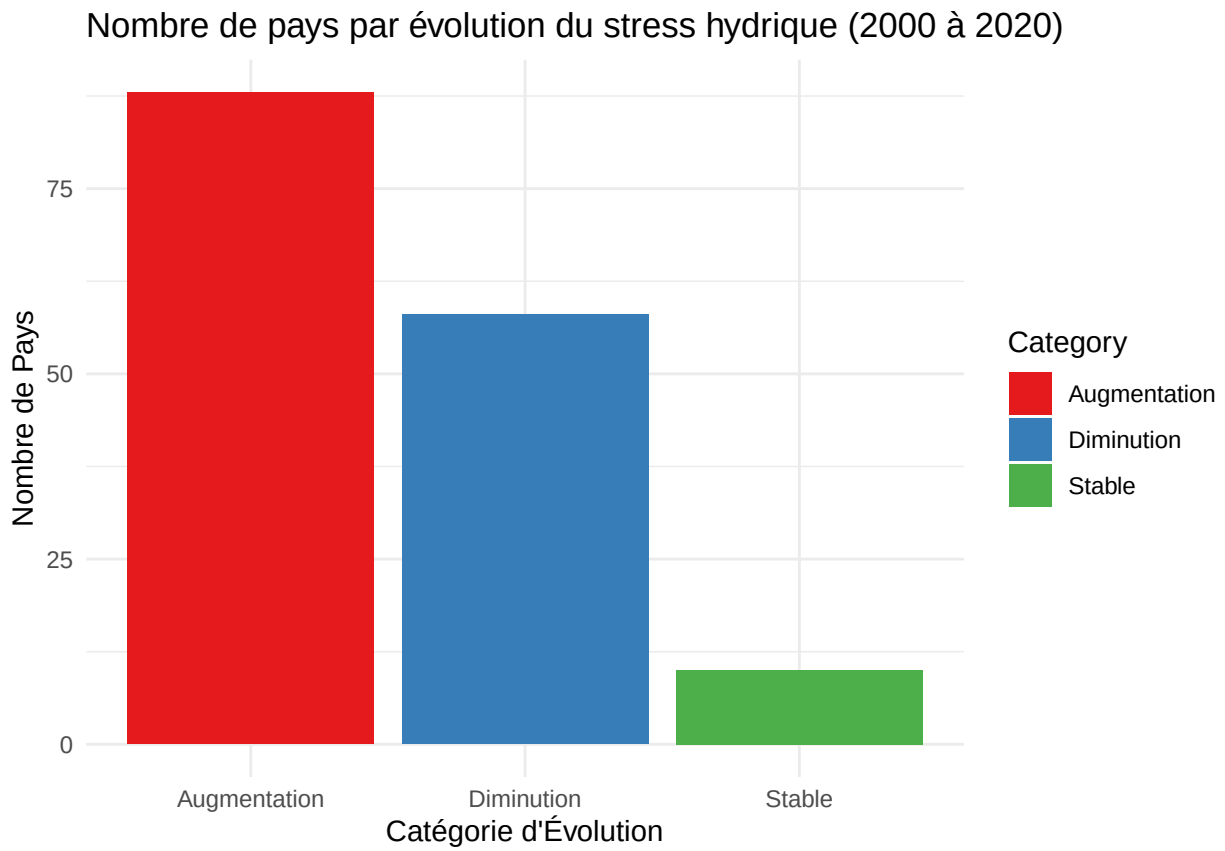
```
## # A tibble: 156 x 4
##    Country             augmentation diminution stable
##    <chr>               <chr>        <chr>      <chr>
```

```
##  1 Afghanistan          ""           ""          "X"
##  2 Albania              "X"          ""          ""
##  3 Algeria              ""           "X"         ""
##  4 Angola               "X"          ""          ""
##  5 Antigua and Barbuda "X"           ""          ""
##  6 Argentina            ""           "X"         ""
##  7 Armenia              "X"          ""          ""
##  8 Australia            ""           "X"         ""
##  9 Austria              "X"          ""          ""
## 10 Azerbaijan           "X"          ""          ""
## # i 146 more rows
```

```
View(result_table)
```

g- Faire un Histogramme qui visualise l'évolution du stress hydrique entre 2000 et 2020

```
library(ggplot2)

# Ajouter une colonne pour la classification
data_null_values_proper$Evolution <- with(data_null_values_proper,
                                          ifelse(Difference > 0, "Augmentation",
                                        ifelse(Difference < 0, "Diminution", "Stable")))

# Compter le nombre de pays dans chaque catégorie
evolution_count <- table(data_null_values_proper$Evolution)

# Transformer en data frame pour ggplot
evolution_df <- as.data.frame(evolution_count)
names(evolution_df) <- c("Category", "Count")

# Créer l'histogramme
ggplot(evolution_df, aes(x = Category, y = Count, fill = Category)) +
  geom_bar(stat = "identity") +
  labs(title = "Nombre de pays par évolution du stress hydrique (2000 à 2020)",
       x = "Catégorie d'Évolution", y = "Nombre de Pays") +
  scale_fill_brewer(palette = "Set1") + # Ajouter des couleurs
  theme_minimal()
```




## 4- CONCLUSION

*Notre analyse approfondie des tendances du stress hydrique dans un échantillon de 157 pays au cours des deux dernières décennies révèle une réalité alarmante : dans plus de 87 de ces pays, soit environ 55des ressources en eau.*

*Ce que ces chiffres indiquent clairement, c'est que nous sommes à un point critique. La croissance démographique, les pratiques industrielles et agricoles non durables, ainsi que l'impact accru du changement climatique contribuent tous à cette augmentation rapide du stress hydrique. Nos analyses suggèrent que cette tendance à la hausse ne va pas seulement se poursuivre, mais va probablement s'accélérer dans les années à venir. Cela implique que, sans interventions significatives et des changements radicaux dans notre manière de gérer les ressources en eau, nous pourrions nous diriger vers une crise mondiale de l'eau.*

*'En somme, les données parlent d'elles-mêmes : nous sommes à un tournant décisif dans notre gestion des ressources en eau. Si nous ne prenons pas des mesures immédiates et efficaces, nous risquons de faire face à des pénuries d'eau généralisées, avec des conséquences désastreuses sur tous les aspects de la vie humaine et environnementale. Il est impératif d'agir maintenant pour inverser ces tendances inquiétantes et garantir un avenir où l'eau, cette ressource vitale, est disponible et accessible pour tous.*