

Livrable 1 - Pokemon

Musset Aurélien
Arulraj Sinthujan
Schummer Lucas
Khuu Sophie

2025-04-20

1. Visualisation des différents combos de types de Pokémon

Objectif du graphique :

Ici, nous allons répondre à la question suivante : **Quel est le nombre de Pokémon par combinaison de types ?** En effet, les pokémons possèdent un type principal (type1) et un type secondaire (type2). Le but est de voir ici, si ces types sont équitablement répartis, où si on peut observer des tendances. Nous allons donc créer une heatmap permettant de visualiser les combinaisons de types de Pokémon.

Présentation :

Pour réaliser ce graphique, nous avons utilisé les données du dataset “Pokemon_data”. Nous avons filtré les Pokémon qui n’ont pas de type secondaire (type2 = “None”). Nous avons ensuite compté le nombre de Pokémon par combinaison de types, puis nous avons ajouté des totaux pour chaque type principal et secondaire.

Code et Visualisation

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(tidyr)

#Ouverture du dataset
df <- read.csv("data/Pokemon_data.csv", sep = ",")

#On ne garde que les colonnes nécessaires : name, type1 et type2
df <- df %>%
```

```

select(name, type1, type2) %>%
mutate(type1 = as.character(type1), type2 = as.character(type2))

# Comptage
df_filtered <- df %>%
  filter(type2 != "None") %>%
  count(type1, type2)

# Totaux
row_totals <- df_filtered %>%
  group_by(type1) %>%
  summarise(n = sum(n)) %>%
  mutate(type2 = "Total")

col_totals <- df_filtered %>%
  group_by(type2) %>%
  summarise(n = sum(n)) %>%
  mutate(type1 = "Total")

# Fusion
df_augmented <- bind_rows(df_filtered, row_totals, col_totals)

# Marquage des cellules de total
df_augmented <- df_augmented %>%
  mutate(
    is_total = type1 == "Total" | type2 == "Total",
    fill_value = ifelse(is_total, NA, n),
    text_color = ifelse(is_total, "black", "gray10"),
    border_color = ifelse(is_total, "black", "white")
  )

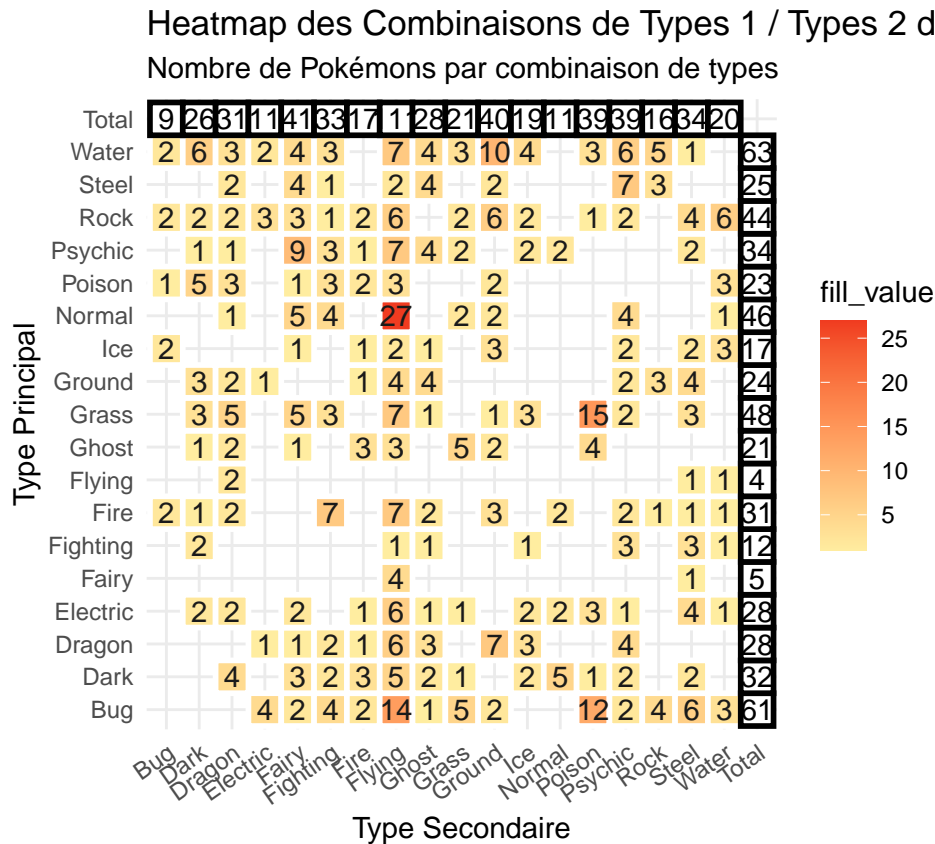
# Forcer l'ordre pour mettre "Total" à droite / en haut
type1_levels <- c(sort(unique(df_augmented$type1[df_augmented$type1 != "Total"])), "Total")
type2_levels <- c(sort(unique(df_augmented$type2[df_augmented$type2 != "Total"])), "Total")

df_augmented <- df_augmented %>%
  mutate(
    type1 = factor(type1, levels = type1_levels),
    type2 = factor(type2, levels = type2_levels)
  )

# Heatmap finale
ggplot(df_augmented, aes(x = type2, y = type1)) +
  geom_tile(aes(fill = fill_value), color = df_augmented$border_color, size = 1) +
  geom_text(aes(label = n, color = text_color), show.legend = FALSE) +
  scale_fill_gradient(low = "#ffeda0", high = "#f03b20", na.value = "white") +
  scale_color_identity() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 35, hjust = 1)) +
  labs(
    title = "Heatmap des Combinaisons de Types 1 / Types 2 des Pokémon avec un Type Secondaire",
    subtitle = "Nombre de Pokémon par combinaison de types",
    x = "Type Secondaire",

```

```
y = "Type Principal"
) +
coord_fixed()
```



Analyse :

Nous pouvons observer que la majorité des Pokémon possèdent un type principal de type “Eau” (Water) et un type secondaire de type “Vol” (Flying). Nous pouvons également observer que les Pokémon de type “Normal” (Normal) possèdent le plus fréquemment un type secondaire de type “Vol” (Flying) et que les Pokémon de type “Plante” (Grass) possèdent souvent un type secondaire de type “Poison” (Poison). **Ce sont les combos les plus fréquents.** On peut également remarquer que de nombreux combos n'existent pas, comme par exemple “Poison” (Poison) et “Acier” (Steel), ou encore “Fée” (Fairy) et “Combat” (Fighting). Cela peut sembler logique pour certaines combinaisons (par exemple “Feu” (Fire) et “Eau” (Water)), mais il est intéressant de le noter. Malgré tout, on peut remarquer que la plupart des combinaisons sont représentées par la même échelle de couleur, ce qui signifie qu'il n'y a pas de combo de type qui se démarque vraiment des autres (hormis ceux notés au début).

2. Visualisation de l'évolution du nombre de ventes

Objectif du graphique :

Ici, nous allons répondre à la question suivante : **Y a-t-il une corrélation entre le nombre de ventes et l'année de sortie du jeu ?**

Présentation :

Pour réaliser ce graphique, j'ai utilisé les données du dataset “vgsales”. Ce graphique présente l'évolution

des ventes des jeux Pokémon (> 6 Millions de ventes) au cours du temps. Il présente l'évolution des ventes des jeux Pokémon global et également pour chaque région.

L'axe des abscisses représente le temps en année de 1996 à 2014.

L'axe des ordonnées représente le nombre de ventes en millions de 0 à plus de 30 millions.

Code et Visualisation

Charger les packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v lubridate 1.9.4     v stringr  1.5.1
## v purrr     1.0.2     v tibble   3.2.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Charger les données

```
df <- read.csv("data/vgsales.csv")
```

Filtrer les jeux Pokémon avec >6M ventes

```
pokemon_sales <- df %>%
  filter(grepl("Pokemon", Name, ignore.case = TRUE), Global_Sales > 6) %>%
  filter(!is.na(Year)) %>%
  mutate(Year = as.integer(as.character(Year)))
```

Agréger ventes par année

```
pokemon_region_year <- pokemon_sales %>%
  group_by(Year) %>%
  summarise(
    North_A = sum(NA_Sales, na.rm = TRUE),
    EU = sum(EU_Sales, na.rm = TRUE),
    JP = sum(JP_Sales, na.rm = TRUE),
    Other = sum(Other_Sales, na.rm = TRUE),
    Total = sum(Global_Sales, na.rm = TRUE)
  ) %>%
  pivot_longer(cols = c(North_A, EU, JP, Other, Total), names_to = "Region", values_to = "Sales")
```

Graphique esthétique

```
ggplot(pokemon_region_year, aes(x = Year, y = Sales, color = Region, group = Region)) +
  geom_line(aes(size = Region == "Total", linetype = Region == "Total")) +
  geom_point(size = 2.5) +
  scale_size_manual(values = c("TRUE" = 1.6, "FALSE" = 1), guide = "none") +
  scale_linetype_manual(values = c("TRUE" = "solid", "FALSE" = "dashed"), guide = "none") +
  scale_color_brewer(palette = "Set1") +
  scale_y_continuous(labels = scales::label_number(suffix = "M", scale = 1)) +
  scale_x_continuous(breaks = unique(pokemon_region_year$Year)) +
  labs(
    title = "Évolution des ventes des jeux Pokémon (>6M de ventes)",
    subtitle = "Répartition par région et total global (en millions d'unités)",
    x = "Année",
```

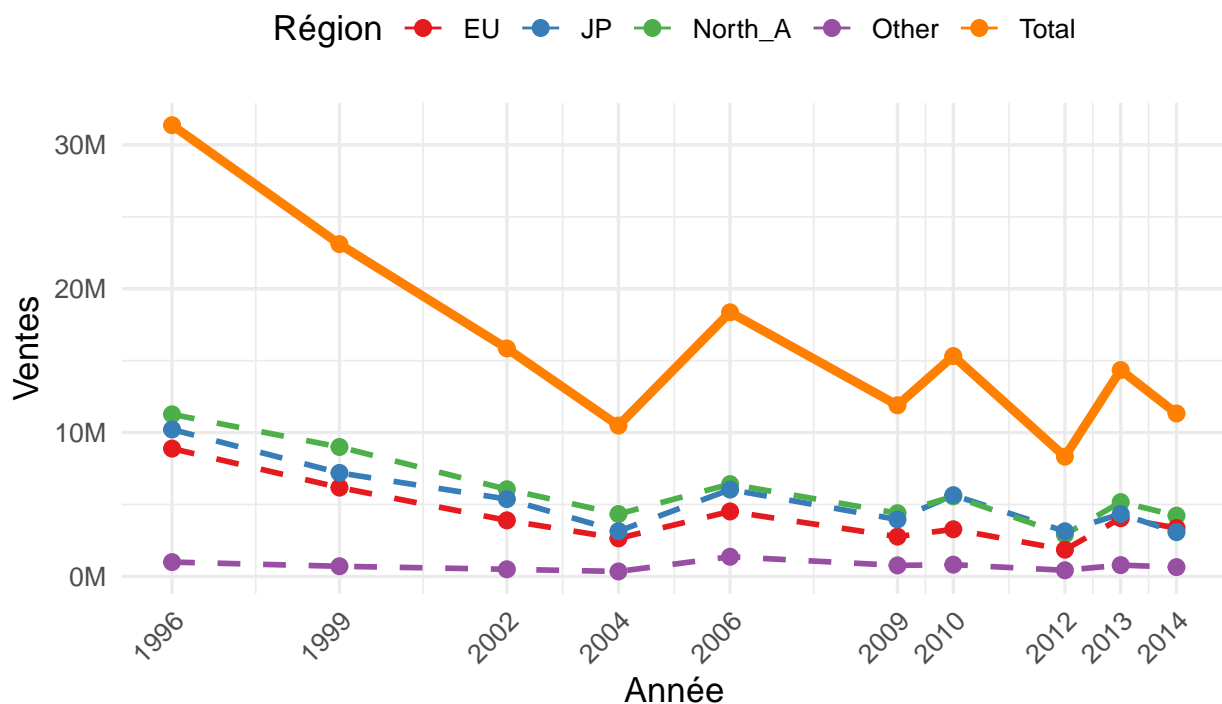
```

y = "Ventes",
color = "Région"
) +
theme_minimal(base_size = 13) +
theme(
  plot.title = element_text(face = "bold", hjust = 0.5),
  plot.subtitle = element_text(hjust = 0.5),
  legend.position = "top",
  axis.text.x = element_text(angle = 45, hjust = 1)
)

```

Évolution des ventes des jeux Pokémon (>6M de ventes)

Répartition par région et total global (en millions d'unités)



Analyse :

On peut remarquer que les jeux récents se vendent moins bien que les précédents. Le jeu Pokémon le plus vendu de l'histoire est sorti en 1996 avec plus de 30 millions de ventes, tandis que les plus récents peinent à atteindre les 20 millions de ventes.

Cela peut être surprenant, si on pense à la popularité des jeux Pokémon dans les années 2000 et 2010 et à l'essor des jeux vidéo et des consoles, qui occupaient une place centrale dans les loisirs des jeunes à cette époque.

3. Popularité des type de Pokémon

Objectif du graphique :

Ici, nous allons répondre à la question suivante : **Quels sont les types de Pokémon préférés de la communauté ?**

Présentation :

Pour réaliser ce graphique, nous avons combiné deux jeux de données :

- **The complete Pokemon dataset** : Statistiques complètes des Pokémons des 8 premières générations (notamment le type)
- **Sondage de popularité** : Sondage réalisé en 2019 sur un forum de la communauté Pokémon recueillant le Pokémon préféré de 52000 votants

Nous considérons ici seulement le premier type de chaque Pokémon (certains Pokémons ont un deuxième type, mais on peut considérer qu'il est moins caractérisant que le premier).

On affiche ici, pour chacun des types, le pourcentage de vote reçu par l'ensemble des Pokémons de ce type. Pour mettre en perspective ces résultats, on affiche également la distribution des Pokémons par type. Les Pokémons dessinés sont les plus populaires pour chacun des types.

Code et Visualisation

```
library(tidyverse)
library(ggimage)

# Load data
popularity <- tibble(read.table("data/popularity.txt", header = TRUE, sep = ","))
pok_data <- tibble(read.csv2("data/Pokemon_data.csv", header = TRUE, sep=';'))

# Join datasets by Pokemon name
merged <- popularity %>% left_join(pok_data %>% select(name, type1), by = c("Pokemon" = "name"))

# Create aggregated data by Pokemon type
type_summary <- merged %>%
  filter(!is.na(type1)) %>%
  group_by(type1) %>%
  summarise(TotalVotes = sum(Number.of.votes), Count = n()) %>%
  mutate(AvgVotes = TotalVotes / Count) %>%
  arrange(desc(AvgVotes)) %>%
  left_join(
    # Select the most popular Pokemon for each type
    merged %>%
    filter(!is.na(type1)) %>%
    group_by(type1) %>%
    slice_max(Number.of.votes, n = 1) %>%
    select(type1, Pokemon, Number.of.votes),
    by = "type1"
  ) %>%
  mutate(img_path = paste0("data/images/", tolower(Pokemon), ".png"))

# Define colors of each type for graphical purpose
type_colors <-c(
  'Normal' = '#A8A77A',
  'Fire' = '#EE8130',
  'Water' = '#6390F0',
  'Electric' = '#F7D02C',
  'Grass' = '#7AC74C',
  'Ice' = '#96D9D6',
  'Fighting' = '#C22E28',
  'Poison' = '#A33EA1',
```

```

'Ground' = '#E2BF65',
'Flying' = '#A98FF3',
'Psychic' = '#F95587',
'Bug' = '#A6B91A',
'Rock' = '#B6A136',
'Ghost' = '#735797',
'Dragon' = '#6F35FC',
'Dark' = '#705746',
'Steel' = '#B7B7CE',
'Fairy' = '#D685AD'
)

# Compute the average number of votes
avg_votes <- mean(type_summary$TotalVotes) / sum(type_summary$TotalVotes) * 100

# Draw graph
type_summary %>%
  ggplot(aes(x = reorder(type1, -TotalVotes),
              y = TotalVotes / sum(TotalVotes) * 100,
              fill=type1)) +
  # Colored bars
  geom_col() +
  geom_hline(yintercept = avg_votes, linetype = "dashed", color = "red", linewidth = 1, alpha=.5) +

  # Black bars
  geom_col(aes(y = Count / sum(Count) * 100, fill = "Nombre de Pokémons"), alpha = 0.3, width = 0.6) +

  # Fake invisible layer to create "Nombre de Votes" legend
  geom_col(aes(y = 0, fill = "Nombre de Votes"), show.legend = TRUE) +

  scale_fill_manual(
    name = NULL,
    values = c(type_colors, "Nombre de Pokémons" = "black", "Nombre de Votes" = "#6390F0"),
    breaks = c("Nombre de Votes", "Nombre de Pokémons")
  ) +
  guides(
    fill = guide_legend(override.aes = list(alpha = c(1, 0.3)))
  ) +

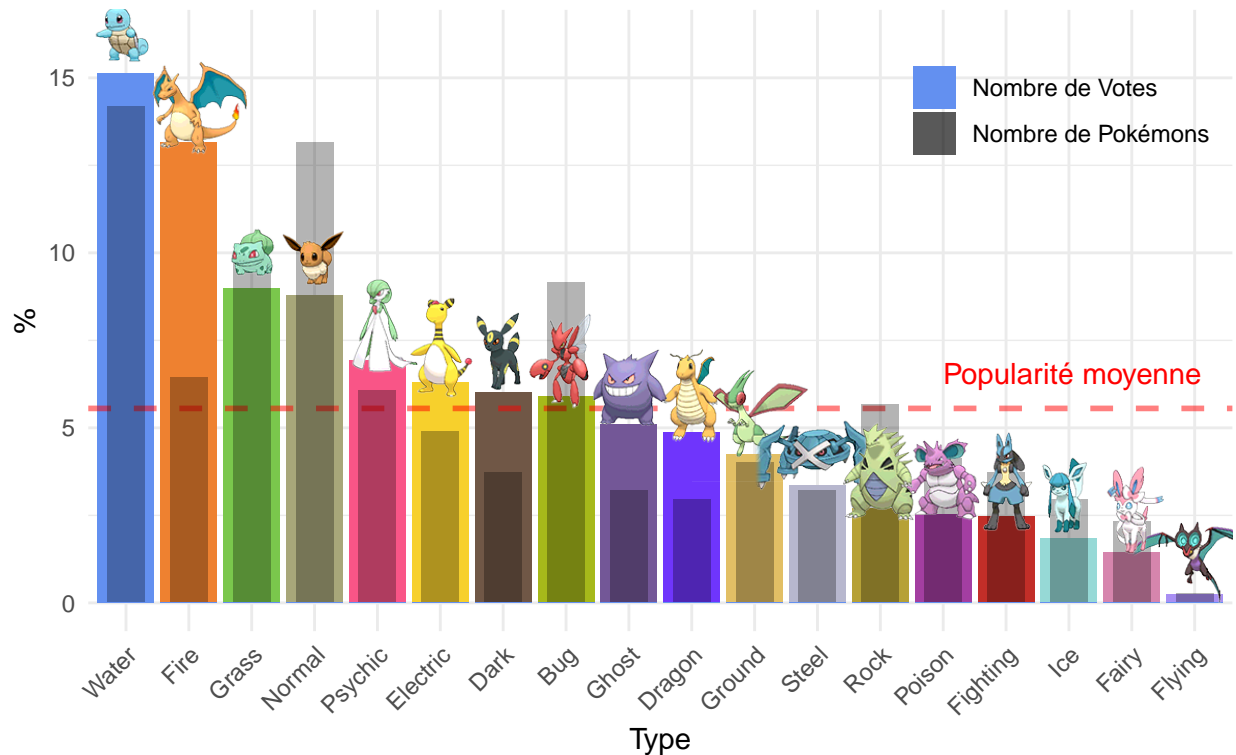
  annotate("text", x = 14, y = avg_votes + 1, label = "Popularité moyenne", color = "red", size = 4, hjust = 1) +
  labs(title = "Popularité des types de Pokemon", subtitle = "Nombre de votes par type", x = "Type", y = "Votes") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = c(0.85, 0.85)) +

  geom_image(aes(image=img_path, y = TotalVotes / sum(TotalVotes) * 100 + 1), size=.2)

```

Popularité des types de Pokemon

Nombre de votes par type



Analyse :

Tout d'abord, il est intéressant d'observer que les 3 types ayant reçu le plus grand nombre de vote sont les 3 types de base (**Eau**, **Feu** et **Plante**). Presque 40% des votes se sont tournés vers ces types, notamment grâce aux Pokémons **starters** (Pokémons de départ) qui sont très populaires. Bien que le type **Eau** arrive en tête du trio, les Pokémons **Eau** étant assez largement les plus représentés, ils sont, en proportion, bien moins appréciés que les Pokémons **Feu**.

De même, le type **Normal** arrive en quatrième position, mais cela semble surtout s'expliquer par le nombre très important de Pokémons de ce type. Les types **Insecte** et **Roche** ont également reçu un nombre de vote très faible en comparaison de leur nombre.

A l'inverse, les types **Ténèbres**, **Spectre** et **Dragon** ont été beaucoup cités malgré le nombre de Pokémons assez modeste.

Le grand gagnant de ce sondage apparaît assez clairement : il s'agit du type **Feu**

4. Comparaison des base stats des Pokémons légendaires et des Pokémons normaux

Objectif du graphique : Ici, nous allons répondre à la question suivante : **Les Pokémons légendaires sont-ils les plus forts ?**

Parmi tous les Pokémon existants, certains Pokémon sont considérés comme étant spéciaux : les Pokémon légendaires, iconiques à chaque jeu, accompagnés des Pokémon fabuleux. Introduits dans les jeux Pokémon X et Y, les méga-évolutions constituent également une catégorie à part.

Présentation : Afin de savoir si ces Pokémon sont réellement plus puissants que les autres, nous allons observer la répartition des totaux des statistiques de base (abrégé “base stats”) pour tous les Pokémon, en différenciant les légendaires des autres.

Nous allons utiliser le dataset “The Complete Pokemon Dataset”.

Les conditions des Pokémon légendaires, mythiques (“mythical” en anglais) et méga-évolutions sont dispersées dans 3 colonnes différentes que nous allons fusionner en une seule colonne “group”. Enfin, nous affichons l’histogramme du total de base stats, en différenciant les Pokémon spéciaux par la couleur.

Code et Visualisation

```
# importations des bibliothèques tidyverse
library(ggplot2)
library(dplyr)
library(readr)
```

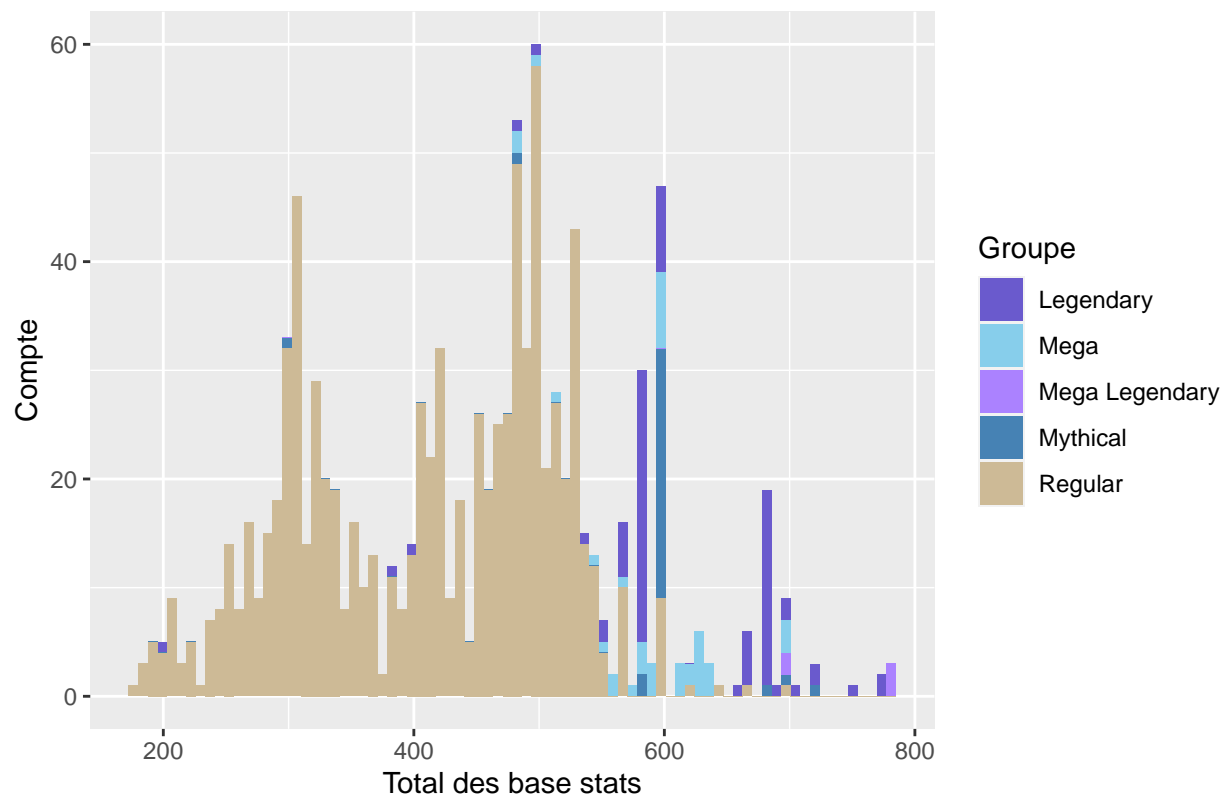
```
# ouverture du dataset
df = read_csv("data/Pokemon_data.csv")
```

```
## Rows: 1033 Columns: 40
## -- Column specification -----
## Delimiter: ","
## chr (6): name, classification, abilities, type1, type2, base_egg_steps
## dbl (34): pokedex_number, generation, height_m, weight_kg, base_total, hp, a...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
# création d'une nouvelle colonne "group", possédant comme valeur "legendary" si le pokémon est légendaire
df <- df %>% mutate(group = case_when(
  (is_legendary == TRUE & is_mega == TRUE) ~ "Mega Legendary",
  is_legendary == TRUE ~ "Legendary",
  is_mythical == TRUE ~ "Mythical",
  is_mega == TRUE ~ "Mega",
  TRUE ~ "Regular"
))
```

```
# affichage de l'histogramme
ggplot(df, aes(base_total)) +
  geom_histogram(bins=80, aes(fill = group)) +
  scale_fill_manual(values = alpha(c("slateblue", "skyblue", "mediumpurple1", "steelblue", "wheat3"))) +
  labs(
    title = "Distribution des Pokémon selon le total de leurs base stats et leur groupe",
    x = "Total des base stats",
    y = "Compte",
    fill = "Groupe")
```

Distribution des Pokémons selon le total de leurs base stats et leur groupe



Analyse :

On peut observer que les Pokémons légendaires, fabuleux et méga-évolués possèdent bien un total de statistiques plus élevé que les Pokémons normaux. Ils constituent la quasi majorité de tous les Pokémons ayant 580 ou plus de base stats. On observe également que la plupart des Pokémons spéciaux ont le même montant de base stats : 600, 580 ou 680. De plus, les Pokémons les plus “puissants”, possédant le montant le plus élevé de base stats font partie d’un groupe rare : ils sont à la fois des méga-évolutions et des Pokémons légendaires (Mega Mewtwo X, Mega Mewtwo Y et méga Rayquaza). Il existe tout de même quelques cas de Pokémons spéciaux avec des statistiques moyennes voire faibles. On peut penser à Cosmog (200 de base stats), qui est un cas particulier parmi les Pokémons légendaires car il peut évoluer : sa dernière évolution, Solgaleo, possède lui 680 de base stats.