Laboratorio 2 análisis exploratorio de datos

Estudiantes: A. Badilla Olivas B80874 - Brandon Mora Umaña C15179

---

0. Instalar paquetes

Solo es necesario instalar los paquetes una vez, si ya los tiene instalados puede omitir este paso.

```
install.packages(c(
  "dplyr",
  "ggplot2",
  "gridExtra",
  "tidyr",
  "reshape2",
  "RColorBrewer",
  "ggrepel"
))
```

```
## Installing packages into '/home/abo/R/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

Ahora cargamos los paquetes necesarios para el análisis.

```
library(dplyr)
library(ggplot2)
library(gridExtra)
library(tidyr)
library(reshape2)
library(RColorBrewer)
library(ggrepel)
library(rmarkdown)
options(width = 80)
```

En la siguiente sección se carga el archivo de datos.

```
# Al usar choose() usted puede elegir el archivo sin necesidad de definir
# la ubicación
# definir header y encodinmg le ayudan a read.csv a
# identificar cabeceras y el formato del texto
df <- (read.csv("pokemon.csv", header = TRUE, encoding = "UTF-8"))
attach(df)
```

```
## The following objects are masked from df (pos = 3):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
```

```
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 4):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 5):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 6):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 7):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg
```

```
## The following objects are masked from df (pos = 8):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 9):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 10):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 12):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 13):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
```

```
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 14):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg
```

```
df$capture_rate <- as.numeric(df$capture_rate)
```

```
## Warning: NAs introduced by coercion
```

```
# Resumen informativo de los datos - tendencias
summary(df)
```

```
##   abilities          against_bug      against_dark   against_dragon
##  Length:801         Min.   :0.2500   Min.   :0.250   Min.   :0.0000
##  Class :character   1st Qu.:0.5000   1st Qu.:1.000   1st Qu.:1.0000
##  Mode  :character   Median :1.0000   Median :1.000   Median :1.0000
##                     Mean   :0.9963   Mean   :1.057   Mean   :0.9688
##                     3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000
##                     Max.   :4.0000   Max.   :4.000   Max.   :2.0000
##
##  against_electric against_fairy   against_fight    against_fire
##  Min.   :0.000    Min.   :0.250   Min.   :0.000   Min.   :0.250
##  1st Qu.:0.500    1st Qu.:1.000   1st Qu.:0.500   1st Qu.:0.500
##  Median :1.000    Median :1.000   Median :1.000   Median :1.000
##  Mean   :1.074    Mean   :1.069   Mean   :1.066   Mean   :1.135
##  3rd Qu.:1.000    3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:2.000
##  Max.   :4.000    Max.   :4.000   Max.   :4.000   Max.   :4.000
##
##  against_flying  against_ghost   against_grass   against_ground
##  Min.   :0.250   Min.   :0.000   Min.   :0.250   Min.   :0.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.500   1st Qu.:1.000
##  Median :1.000   Median :1.000   Median :1.000   Median :1.000
##  Mean   :1.193   Mean   :0.985   Mean   :1.034   Mean   :1.098
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :4.000
##
##   against_ice    against_normal  against_poison   against_psychic
##  Min.   :0.250   Min.   :0.000   Min.   :0.0000   Min.   :0.000
```

```
##  1st Qu.:0.500   1st Qu.:1.000   1st Qu.:0.5000   1st Qu.:1.000
##  Median :1.000   Median :1.000   Median :1.0000   Median :1.000
##  Mean   :1.208   Mean   :0.887   Mean   :0.9753   Mean   :1.005
##  3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :4.000   Max.   :1.000   Max.   :4.0000   Max.   :4.000
##
##   against_rock  against_steel    against_water       attack
##  Min.   :0.25   Min.   :0.2500   Min.   :0.250   Min.   :  5.00
##  1st Qu.:1.00   1st Qu.:0.5000   1st Qu.:0.500   1st Qu.: 55.00
##  Median :1.00   Median :1.0000   Median :1.000   Median : 75.00
##  Mean   :1.25   Mean   :0.9835   Mean   :1.058   Mean   : 77.86
##  3rd Qu.:2.00   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:100.00
##  Max.   :4.00   Max.   :4.0000   Max.   :4.000   Max.   :185.00
##
##  base_egg_steps  base_happiness     base_total     capture_rate
##  Min.   : 1280   Min.   :  0.00   Min.   :180.0   Min.   :  3.00
##  1st Qu.: 5120   1st Qu.: 70.00   1st Qu.:320.0   1st Qu.: 45.00
##  Median : 5120   Median : 70.00   Median :435.0   Median : 60.00
##  Mean   : 7191   Mean   : 65.36   Mean   :428.4   Mean   : 98.76
##  3rd Qu.: 6400   3rd Qu.: 70.00   3rd Qu.:505.0   3rd Qu.:170.00
##  Max.   :30720   Max.   :140.00   Max.   :780.0   Max.   :255.00
##                                                   NA's   :1
##  classfication         defense       experience_growth    height_m
##  Length:801         Min.   :  5.00   Min.   : 600000   Min.   : 0.100
##  Class :character   1st Qu.: 50.00   1st Qu.:1000000   1st Qu.: 0.600
##  Mode  :character   Median : 70.00   Median :1000000   Median : 1.000
##                     Mean   : 73.01   Mean   :1054996   Mean   : 1.164
##                     3rd Qu.: 90.00   3rd Qu.:1059860   3rd Qu.: 1.500
##                     Max.   :230.00   Max.   :1640000   Max.   :14.500
##                                                        NA's   :20
##        hp             name           percentage_male  pokedex_number
##  Min.   :  1.00   Length:801         Min.   :  0.00   Min.   :  1
##  1st Qu.: 50.00   Class :character   1st Qu.: 50.00   1st Qu.:201
##  Median : 65.00   Mode  :character   Median : 50.00   Median :401
##  Mean   : 68.96                      Mean   : 55.16   Mean   :401
##  3rd Qu.: 80.00                      3rd Qu.: 50.00   3rd Qu.:601
##  Max.   :255.00                      Max.   :100.00   Max.   :801
##                                      NA's   :98
##    sp_attack        sp_defense         speed            type1
##  Min.   : 10.00   Min.   : 20.00   Min.   :  5.00   Length:801
##  1st Qu.: 45.00   1st Qu.: 50.00   1st Qu.: 45.00   Class :character
##  Median : 65.00   Median : 66.00   Median : 65.00   Mode  :character
##  Mean   : 71.31   Mean   : 70.91   Mean   : 66.33
##  3rd Qu.: 91.00   3rd Qu.: 90.00   3rd Qu.: 85.00
##  Max.   :194.00   Max.   :230.00   Max.   :180.00
##
##     type2            weight_kg        generation    is_legendary
##  Length:801         Min.   :  0.10   Min.   :1.00   Min.   :0.00000
##  Class :character   1st Qu.:  9.00   1st Qu.:2.00   1st Qu.:0.00000
##  Mode  :character   Median : 27.30   Median :4.00   Median :0.00000
##                     Mean   : 61.38   Mean   :3.69   Mean   :0.08739
##                     3rd Qu.: 64.80   3rd Qu.:5.00   3rd Qu.:0.00000
##                     Max.   :999.90   Max.   :7.00   Max.   :1.00000
##                     NA's   :20
```

```r
# definimos el conjunto como una tibble que es una data frame simplificada
# las tibbles son versiones de dataframes con algunas facilidades
# de impresión y uso.
df <- tibble::as_tibble(df)
colnames(df)[25] <- "classification"

head(df)
```

```
## # A tibble: 6 x 40
##   abilities             against_bug against_dark against_dragon against_electric
##   <chr>                       <dbl>        <dbl>          <dbl>            <dbl>
## 1 ['Overgrow', 'Chloro~        1               1              1              0.5
## 2 ['Overgrow', 'Chloro~        1               1              1              0.5
## 3 ['Overgrow', 'Chloro~        1               1              1              0.5
## 4 ['Blaze', 'Solar Pow~        0.5             1              1              1
## 5 ['Blaze', 'Solar Pow~        0.5             1              1              1
## 6 ['Blaze', 'Solar Pow~        0.25            1              1              2
## # i 35 more variables: against_fairy <dbl>, against_fight <dbl>,
## #   against_fire <dbl>, against_flying <dbl>, against_ghost <dbl>,
## #   against_grass <dbl>, against_ground <dbl>, against_ice <dbl>,
## #   against_normal <dbl>, against_poison <dbl>, against_psychic <dbl>,
## #   against_rock <dbl>, against_steel <dbl>, against_water <dbl>, attack <int>,
## #   base_egg_steps <int>, base_happiness <int>, base_total <int>,
## #   capture_rate <dbl>, classification <chr>, defense <int>, ...
```

```r
df <- select(
  df,
  name,
  classification,
  hp, weight_kg,
  height_m, speed,
  attack, defense,
  sp_attack,
  sp_defense,
  type1,
  type2,
  abilities,
  generation,
  is_legendary,
  capture_rate
)
head(df)
```

```
## # A tibble: 6 x 16
##   name    classification    hp weight_kg height_m speed attack defense sp_attack
##   <chr>   <chr>          <int>     <dbl>    <dbl> <int>  <int>   <int>     <int>
## 1 Bulbas~ Seed Pokémon      45       6.9      0.7    45     49      49        65
## 2 Ivysaur Seed Pokémon      60      13        1      60     62      63        80
## 3 Venusa~ Seed Pokémon      80     100        2      80    100     123       122
## 4 Charma~ Lizard Pokémon    39       8.5      0.6    65     52      43        60
## 5 Charme~ Flame Pokémon     58      19        1.1    80     64      58        80
## 6 Chariz~ Flame Pokémon     78      90.5      1.7   100    104      78       159
## # i 7 more variables: sp_defense <int>, type1 <chr>, type2 <chr>,
```

```
## #   abilities <chr>, generation <int>, is_legendary <int>, capture_rate <dbl>
```

```r
density_plot <- function(
    data, column, fill_color, color,
    transparency, title, x_label) {
  ggplot(data = data, aes(column)) +
    geom_density(col = color, fill = fill_color, alpha = transparency) +
    ggtitle(title) +
    labs(x = x_label, y = "Densidad")
}
```

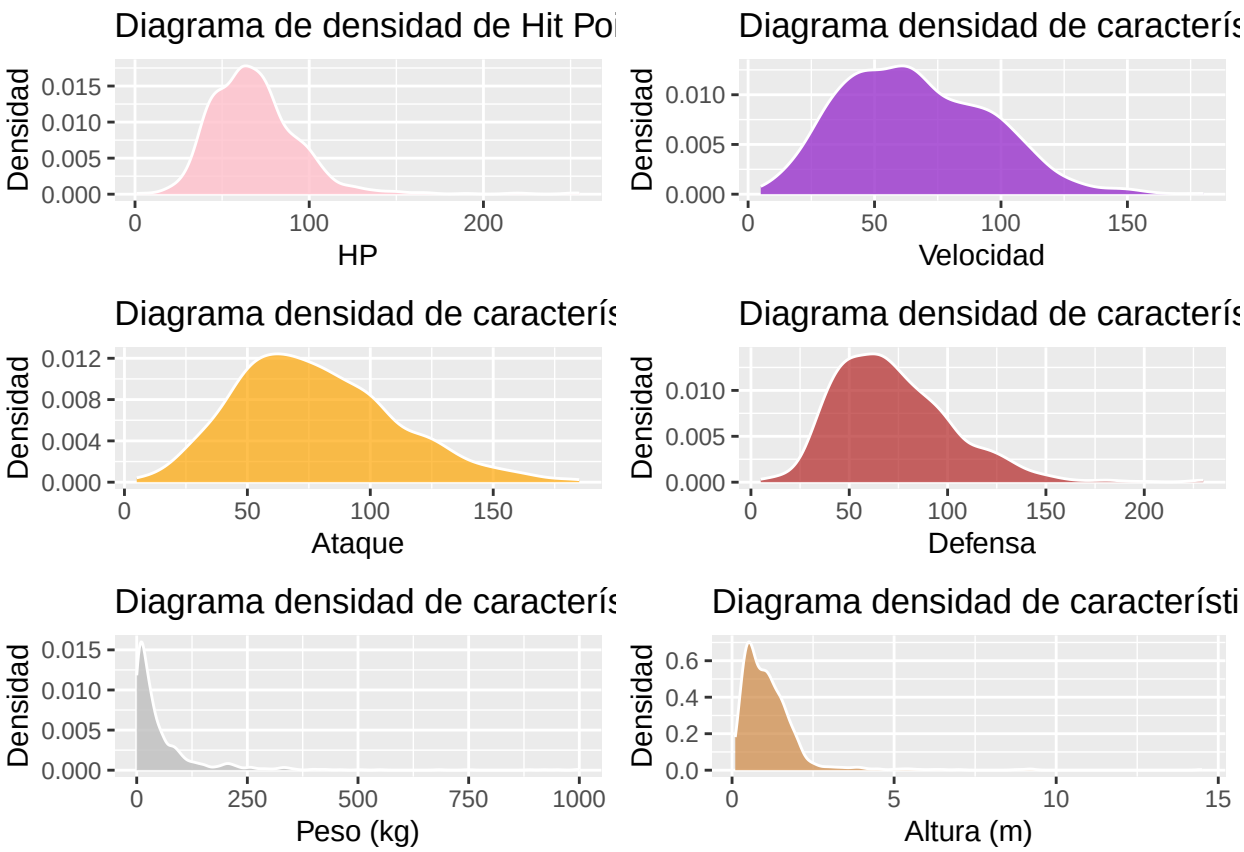
---

1. Diagramas de densidad

```r
# el mas sirve para agregarle parametros al plot.

# Diagrama de densidad de Hit Points o Vida
density_hp <- ggplot(data = df, aes(hp)) +
  geom_density(col = "white", fill = "pink", alpha = 0.8) +
  ggtitle("Diagrama de densidad de Hit Points o Vida") +
  labs(x = "HP", y = "Densidad")

# Diagrama densidad de características de velocidad
density_speed <- ggplot(data = df, aes(speed)) +
  geom_density(col = "white", fill = "darkorchid", alpha = 0.8) +
  ggtitle("Diagrama densidad de características de velocidad") +
  labs(x = "Velocidad", y = "Densidad")

# Diagrama densidad de características ofensivas
density_attack <- ggplot(data = df, aes(attack)) +
  geom_density(col = "white", fill = "orange", alpha = 0.7) +
  ggtitle("Diagrama densidad de características ofensivas") +
  labs(x = "Ataque", y = "Densidad")

# Diagrama densidad de características defensivas
density_defense <- ggplot(data = df, aes(defense)) +
  geom_density(col = "white", fill = "firebrick", alpha = 0.7) +
  ggtitle("Diagrama densidad de características defensivas") +
  labs(x = "Defensa", y = "Densidad")

# Diagrama densidad de características de altura
density_height <- ggplot(data = df, aes(height_m)) +
  geom_density(col = "white", fill = "tan3", alpha = 0.7) +
  ggtitle("Diagrama densidad de características de altura") +
  labs(x = "Altura (m)", y = "Densidad")

# Diagrama densidad de características de peso
density_weight <- density_plot(
  data = df,
  column = weight_kg,
```

```r
  fill_color = "gray",
  color = "white",
  transparency = 0.8,
  title = "Diagrama densidad de características peso",
  x_label = "Peso (kg)"
)

# Organizar los gráficos en una cuadrícula
grid.arrange(density_hp, density_speed, density_attack,
  density_defense, density_weight, density_height,
  ncol = 2
)
```

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

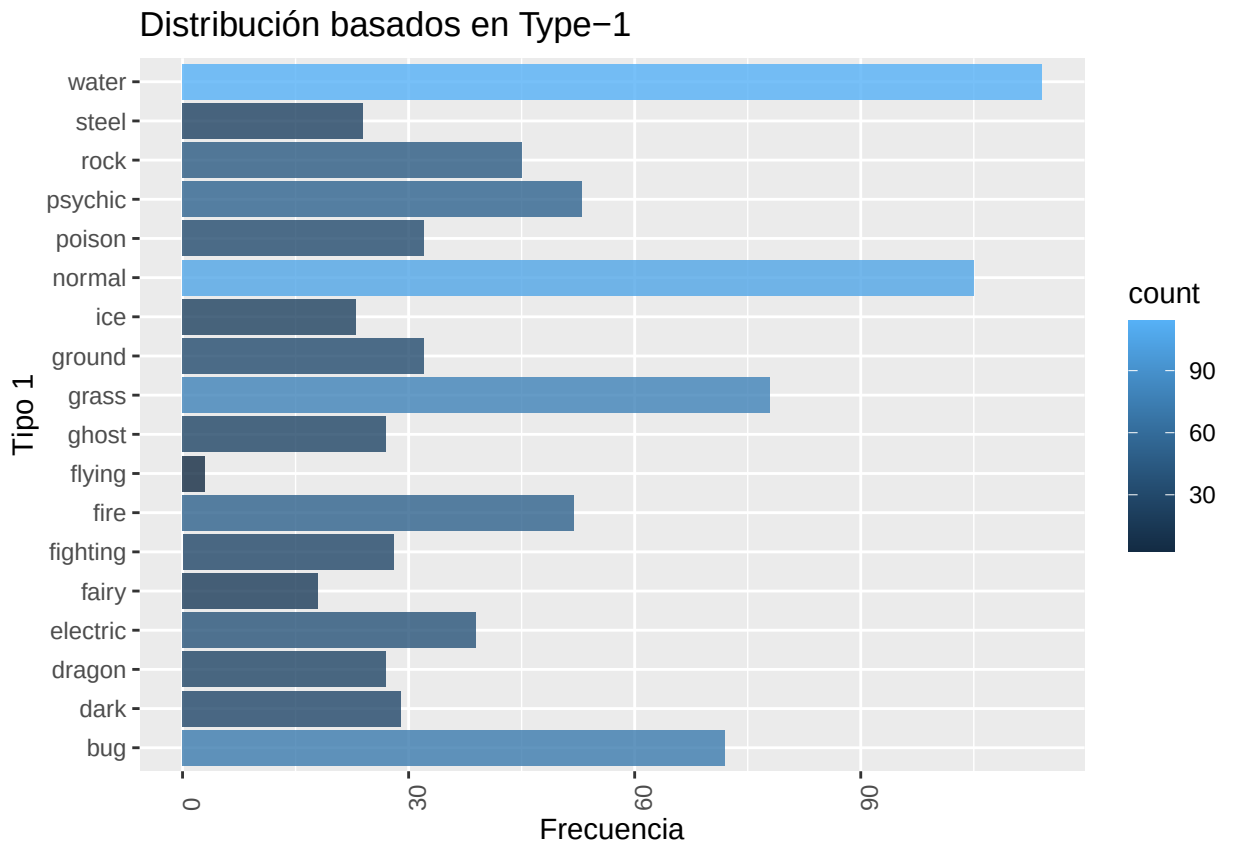


---

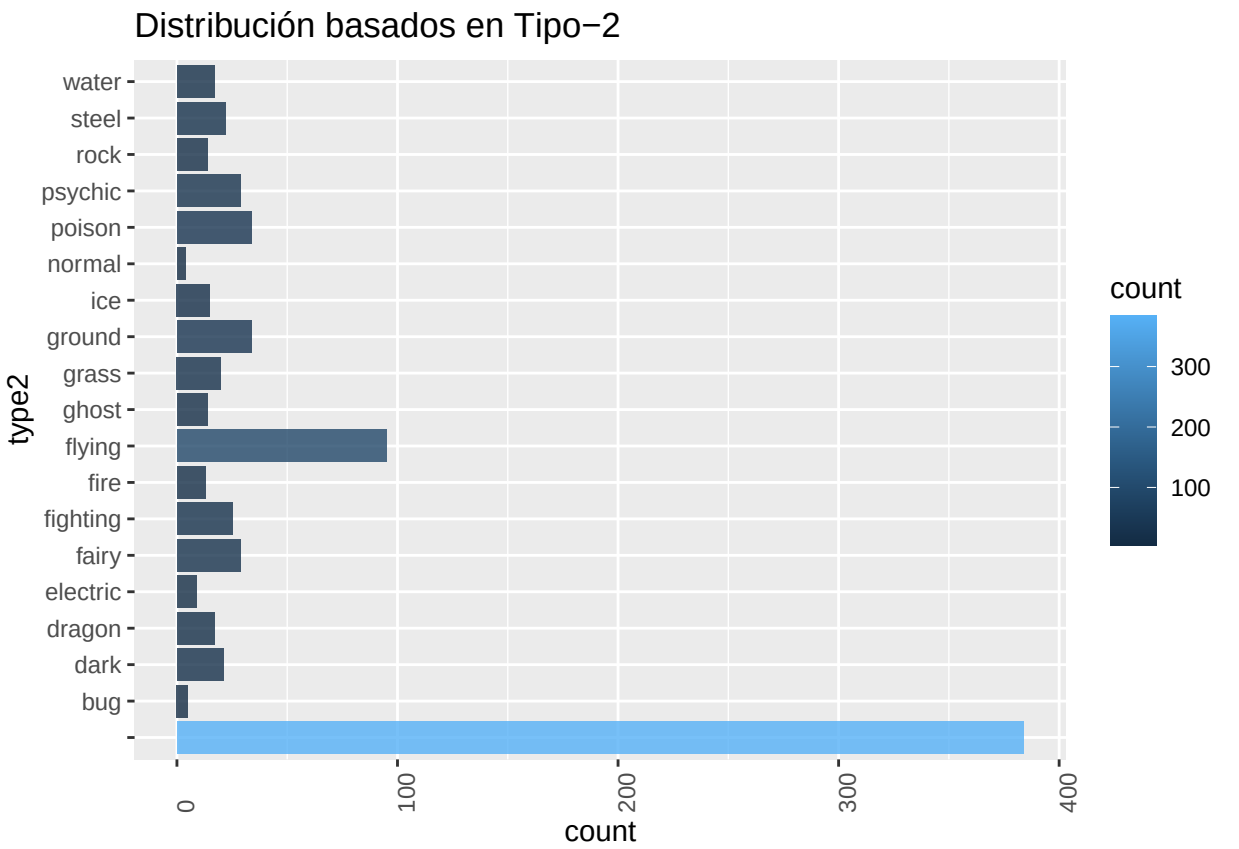
2. Gráficos de barras

```r
# ---- Primer gráfico ----
```

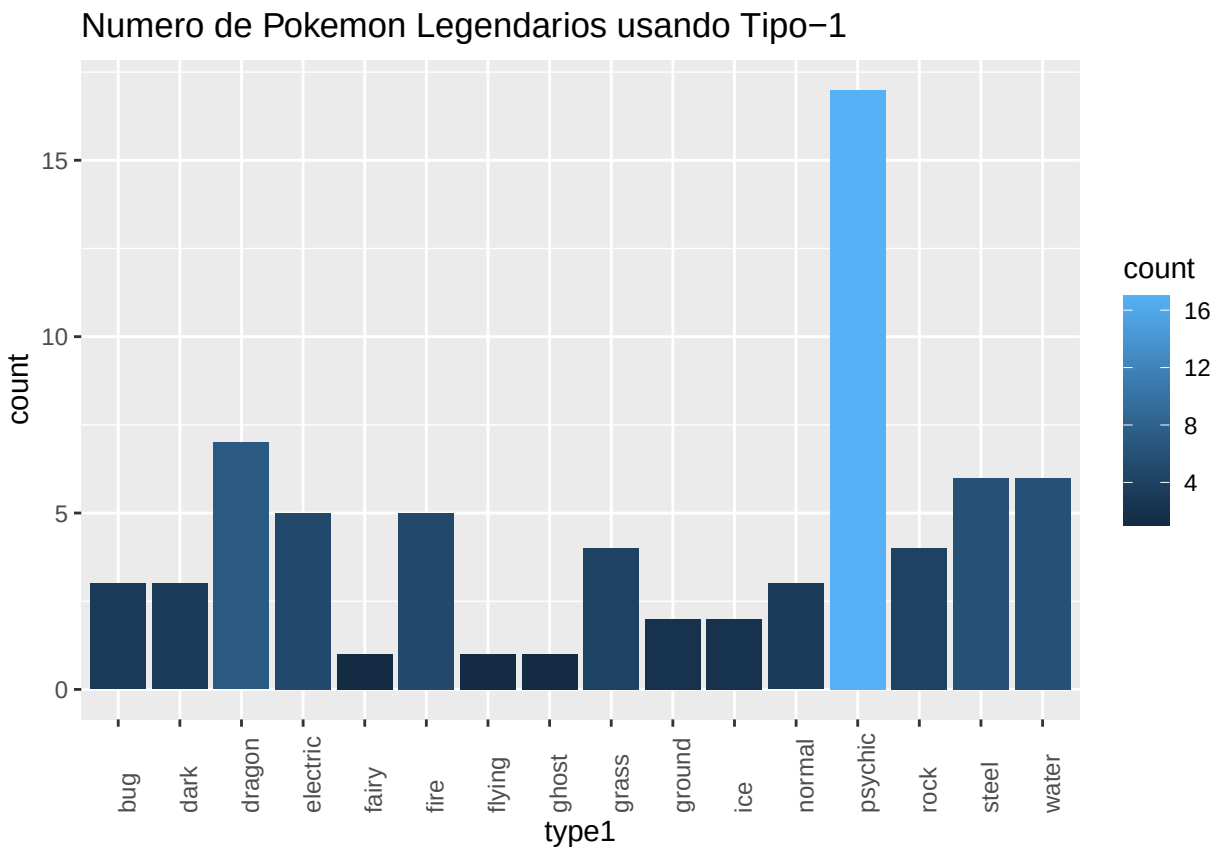
```r
# Gráfico de barras para la distribución de Type-1
ggplot(data = df, aes(type1)) +
  geom_bar(aes(fill = ..count..), alpha = 0.8) +
  theme(axis.text.x = element_text(angle = 90, hjust = 0)) +
  ggtitle("Distribución basados en Type-1") +
  coord_flip() + # Intercambia los ejes x e y
  labs(x = "Tipo 1", y = "Frecuencia")
```




```r
# ---- Segundo gráfico ----

# Gráfico de barras para la distribución de Type-2
ggplot(data = df, aes(type2)) +
  geom_bar(aes(fill = ..count..), alpha = 0.8) +
  theme(axis.text.x = element_text(angle = 90, hjust = 0)) +
  ggtitle("Distribución basados en Tipo-2") +
  coord_flip()
```




```
df %>%
  filter(is_legendary == 1) %>%
  ggplot(aes(type1)) +
  geom_bar(aes(fill = ..count..)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 0)) +
  ggtitle("Numero de Pokemon Legendarios usando Tipo-1")
```



Comentarios sobre los gráficos de barras

Podemos observar que los pokemon tipo agua son los más frequentes del tipo 1 en normales. Para los legendarios es psíquico.

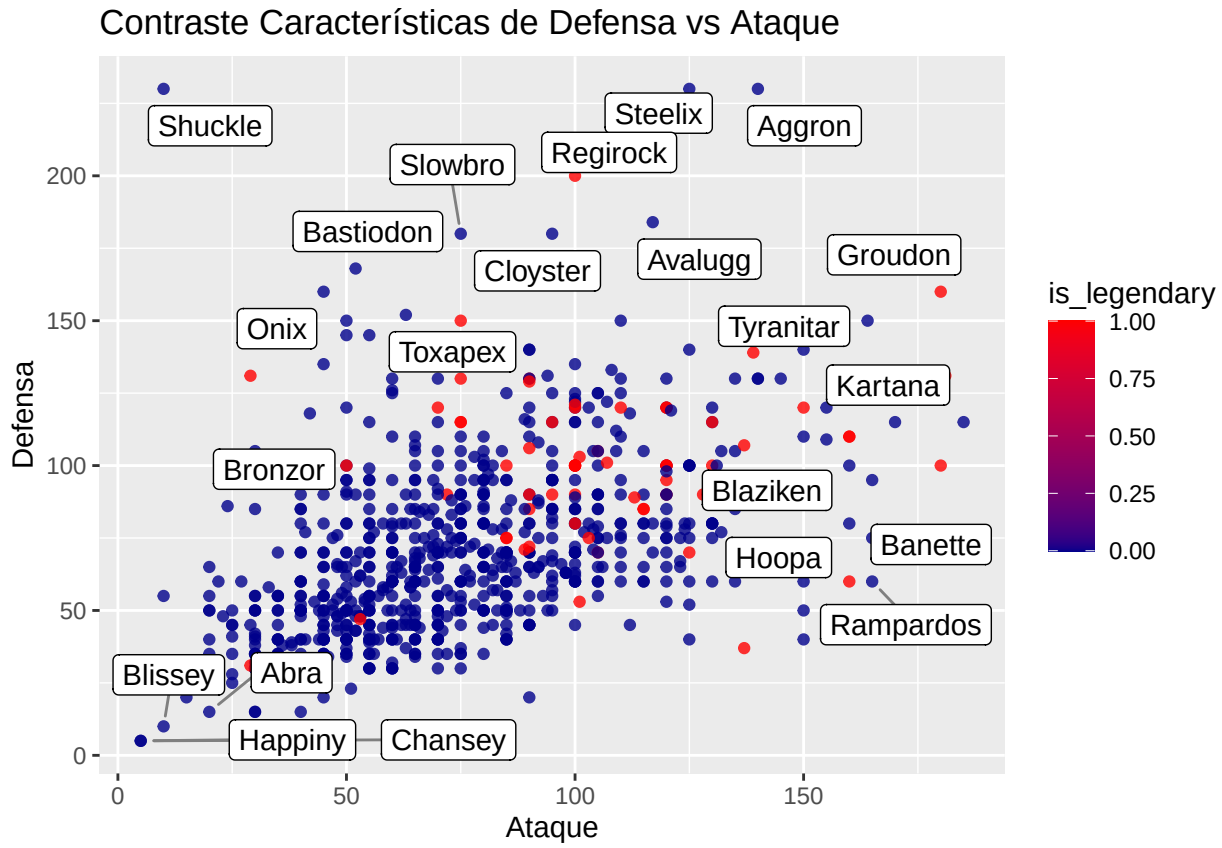
---

3. Gráficos de dispersión

```
# Creamos la base del gráfico
ggplot(data = df, aes(x = attack, y = defense)) +

  # Añadimos puntos con color según legendario o no
  geom_point(aes(color = is_legendary), alpha = 0.8) +

  # Creamos una escala de color degradado
  scale_color_gradient(low = "darkblue", high = "red") +

  # Título del gráfico
  ggtitle("Contraste Características de Defensa vs Ataque") +

  # Añadimos etiquetas de algunos Pokémon
  geom_label_repel(
```

```
    data = subset(df, attack > 150 | defense > 150 | attack < 25),
    aes(label = name),
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50"
  ) +
  labs(x = "Ataque", y = "Defensa")
```

```
## Warning: ggrepel: 20 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



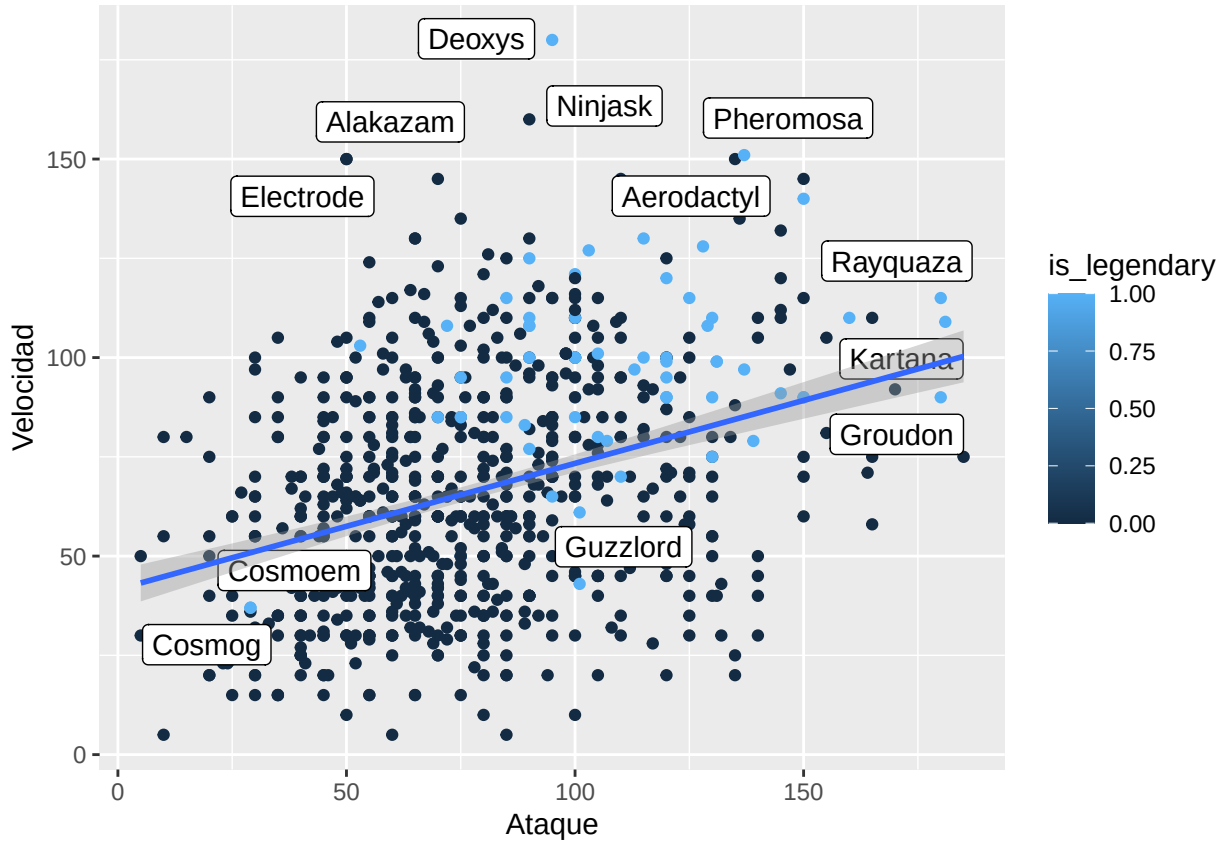
Comentarios sobre el gráfico de dispersión

Podemos observar que los Pokémon legendarios tienden a tener valores más altos en las características de ataque y defensa. A continuación, se presentan gráficos de dispersión que comparan las características de ataque con velocidad, peso, altura y HP.

```
## Gráfico: Ataque vs Velocidad
ggplot(df, aes(attack, speed)) +
  # Los puntos representan Pokémon, coloreados según sean legendarios o no
  geom_point(aes(color = is_legendary)) +
  # Etiquetas para destacar Pokémon específicos
  geom_label_repel(
```

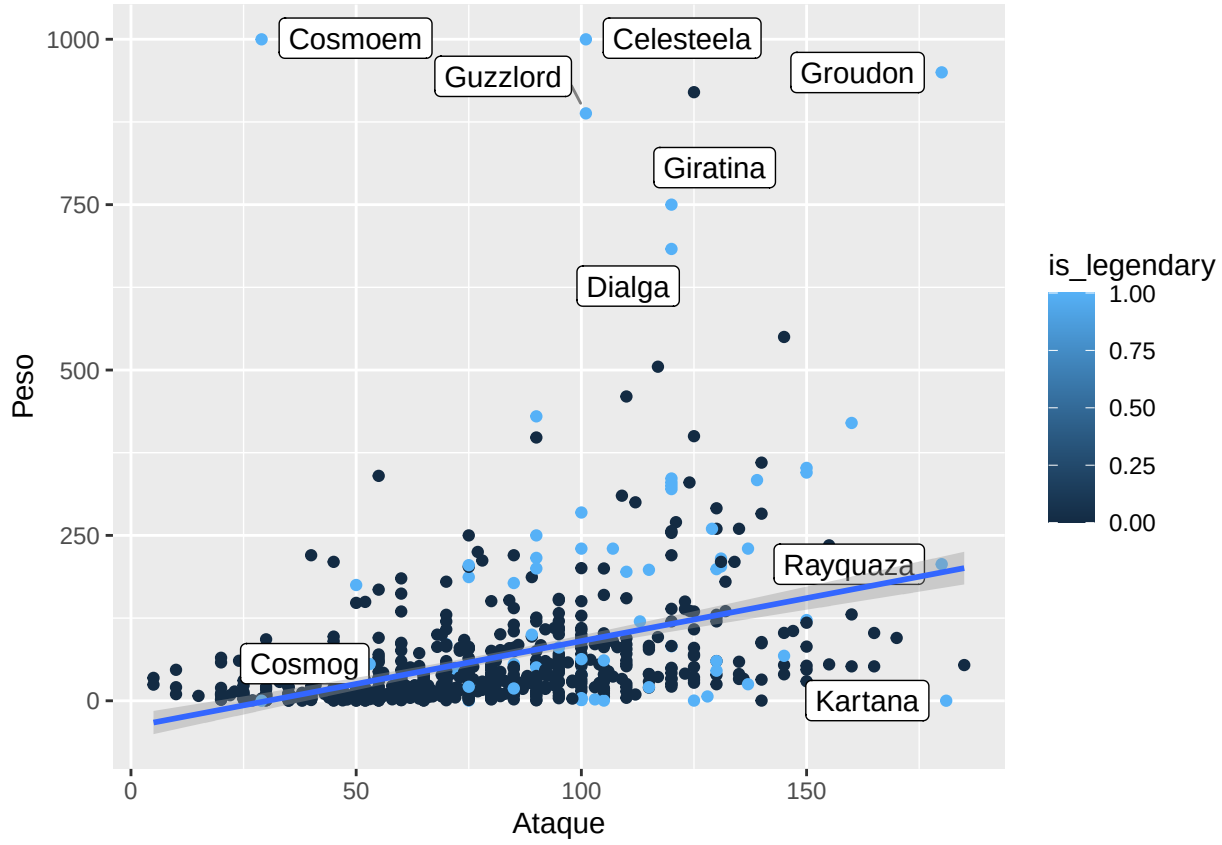
```
    data = subset(df, (attack > 170 | attack < 50 & speed > 150 | speed < 50) & is_legendary == 1 | spe
    aes(label = name),
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50"
  ) +
  # Línea de tendencia
  geom_smooth(method = "lm") +
  #  Etiquetas de los ejes
  labs(x = "Ataque", y = "Velocidad")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



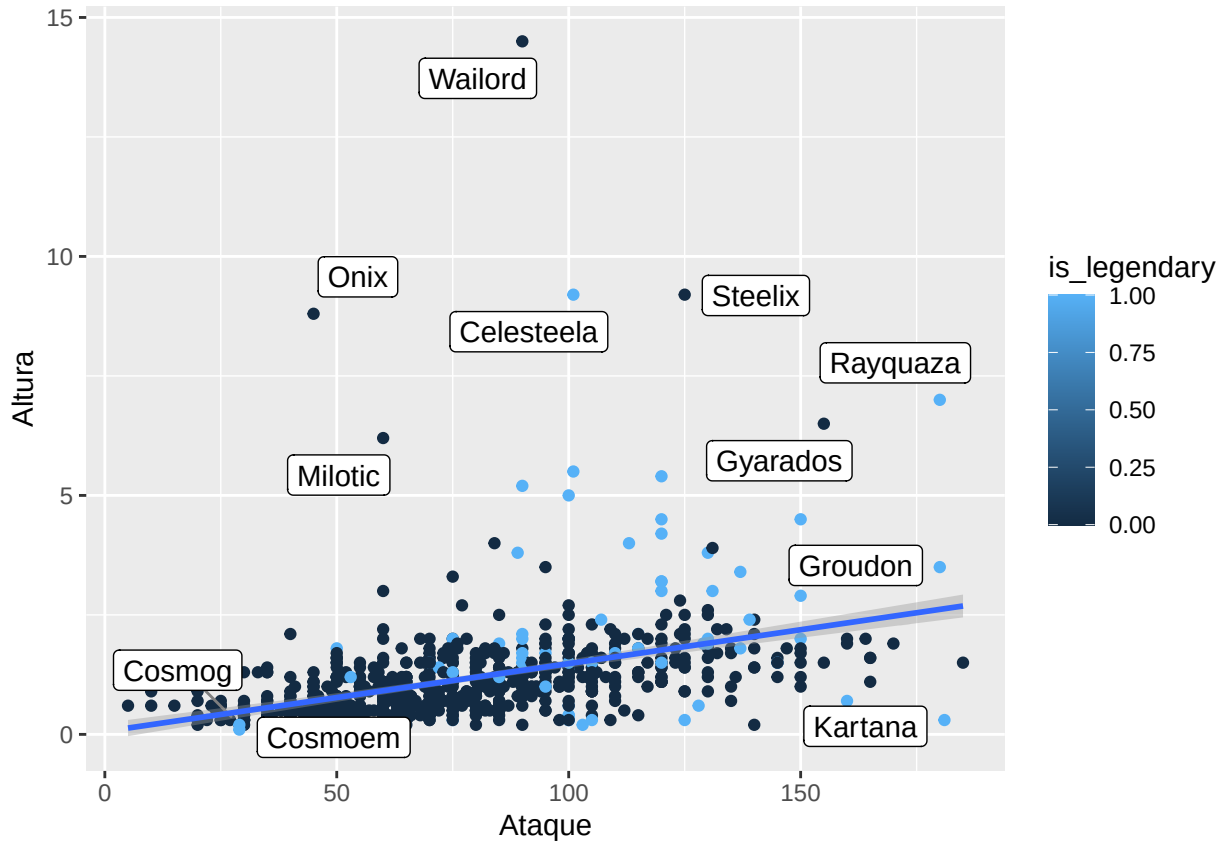

```
## Gráfico: Ataque vs Peso
ggplot(df, aes(attack, weight_kg)) +
  geom_point(aes(color = is_legendary)) +
  geom_label_repel(
    data = subset(df, (attack > 170 | attack < 50 | weight_kg > 650) & (is_legendary == 1)),
    aes(label = name),
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50"
  ) +
  geom_smooth(method = "lm") +
  labs(x = "Ataque", y = "Peso")
```

```
## `geom_smooth()` using formula = 'y ~ x'

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

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



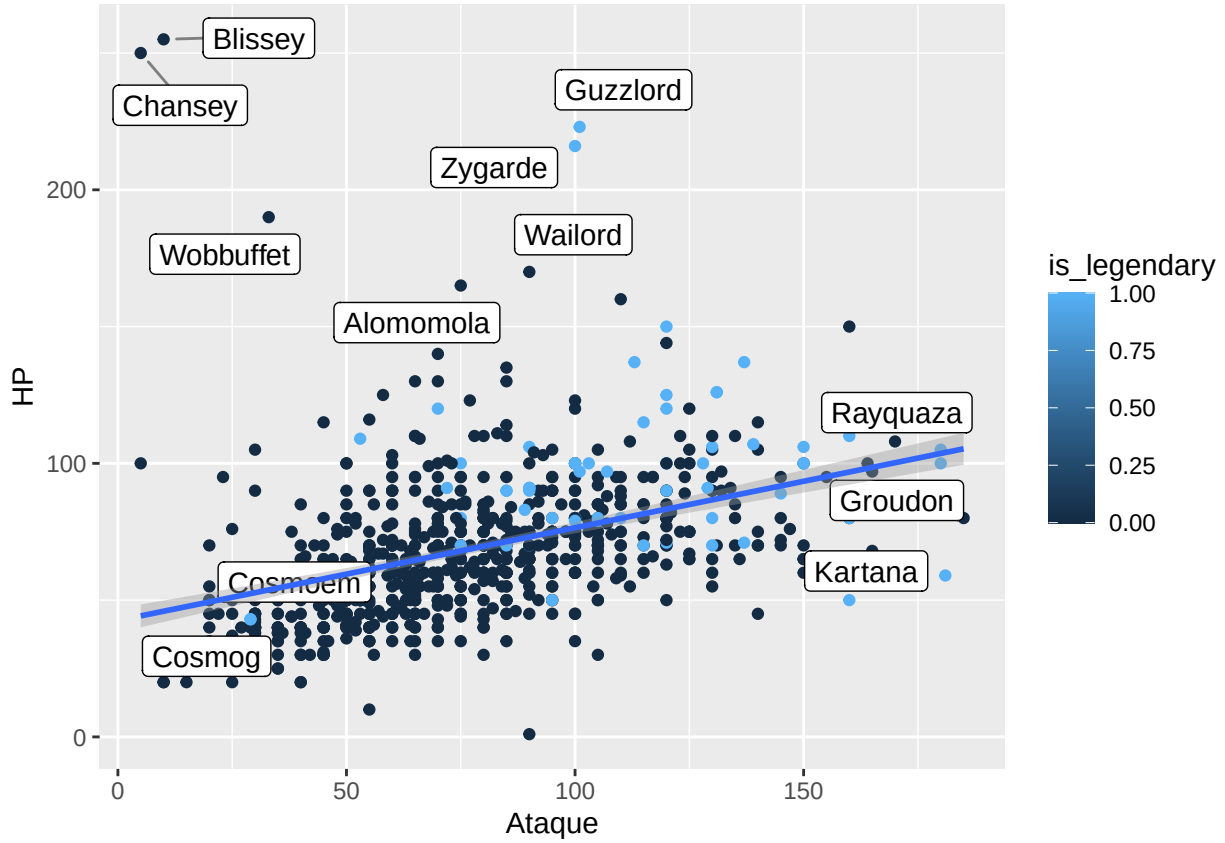

```
## Gráfico: Ataque vs Altura
ggplot(df, aes(attack, height_m)) +
  geom_point(aes(color = is_legendary)) +
  geom_label_repel(
    data = subset(df, ((attack > 170 | attack < 50 | height_m > 7.5) & is_legendary == 1) | height_m > 5
    aes(label = name),
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50"
  ) +
  geom_smooth(method = "lm") +
  labs(x = "Ataque", y = "Altura")
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 20 rows containing non-finite outside the scale range (`stat_smooth()`).
## Removed 20 rows containing missing values or values outside the scale range (`geom_point()`).
```




```
## Gráfico: Ataque vs HP
ggplot(df, aes(attack, hp)) +
  geom_point(aes(color = is_legendary)) +
  geom_label_repel(
    data = subset(df, ((attack > 170 | hp > 190 | attack < 50) & is_legendary == 1) | hp > 160),
    aes(label = name),
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50"
  ) +
  geom_smooth(method = "lm") +
  labs(x = "Ataque", y = "HP")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Comentarios sobre los gráficos de dispersión

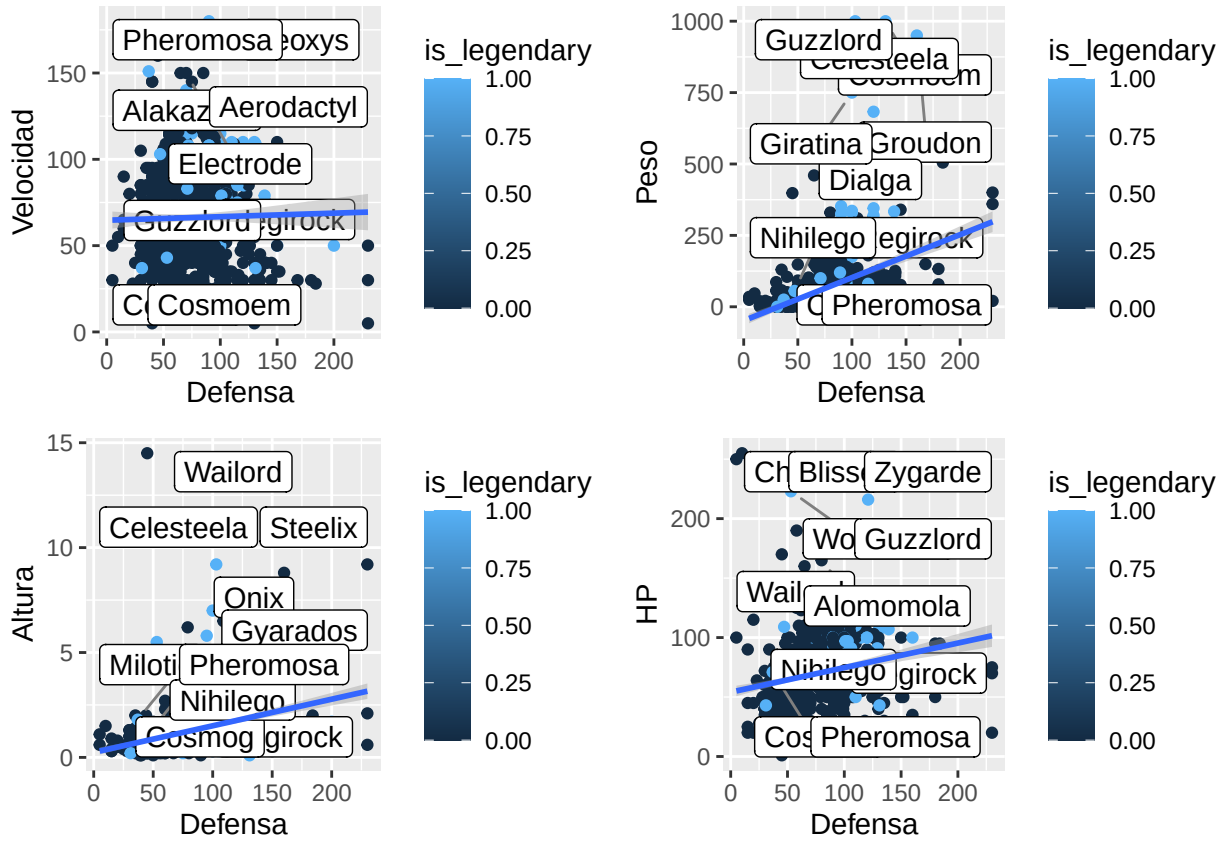
Podemos observar que a medida que aumenta el ataque, aumentan la velocidad, el peso y la altura. Tambien Cosmog y Cosmoem tienen un peso, altura y HP muy bajos en comparacion con otros.

```
speed_defense_legendary <- ggplot(na.omit(df), aes(defense, speed)) +
  # Base del gráfico: Defensa en el eje x, Velocidad en el eje y
  geom_point(aes(color = is_legendary)) +
  # Puntos que representan cada Pokémon, coloreados según sean o no legendarios
  geom_label_repel(
    data = subset(df, (defense > 170 | defense < 50 & speed > 150 | speed < 50) &
      is_legendary == 1 | speed > 145),
    # Filtrado: Pokémon con defensa alta o baja, velocidad extrema y legendarios
    # además de incluir algunos con sólo alta velocidad.
    aes(label = name),
    # Usa la columna 'name' como etiqueta
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50" # Ajuste de espaciado y color de segmentos
  ) +
  geom_smooth(method = "lm") + # Agrega una línea de tendencia lineal
  labs(x = "Defensa", y = "Velocidad") # Etiquetas de ejes específicas para este gráfico.

weight_defense_legendary <- ggplot(na.omit(df), aes(defense, weight_kg)) +
  geom_point(aes(color = is_legendary)) +
```

```
  geom_label_repel(
    data = subset(df, (defense > 170 | defense < 50 | weight_kg > 650) &
      (is_legendary == 1)),
    # Filtra: Defensa extrema, peso muy alto, y sólo legendarios
    aes(label = name),
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50"
  ) +
  geom_smooth(method = "lm") +
  labs(x = "Defensa", y = "Peso") # Etiquetas de ejes específicas para este gráfico.


height_defense_legendary <- ggplot(na.omit(df), aes(defense, height_m)) +
  geom_point(aes(color = is_legendary)) +
  geom_label_repel(
    data = subset(df, ((defense > 170 | defense < 50 | height_m > 7.5) &
      is_legendary == 1) |
      (height_m > 5 & is_legendary == 0)),
    # Filtra: Defensa extrema, altura extrema y legendarios, o altura alta y no legendarios
    aes(label = name),
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50"
  ) +
  geom_smooth(method = "lm") +
  labs(x = "Defensa", y = "Altura")

hp_defense_legendary <- ggplot(na.omit(df), aes(defense, hp)) +
  geom_point(aes(color = is_legendary)) +
  geom_label_repel(
    data = subset(df, ((defense > 170 | hp > 190 | defense < 50) &
      is_legendary == 1) |
      (hp > 160)),
    # Filtra: Defensa extrema, HP alto y legendarios, o HP alto y no legendarios
    aes(label = name),
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = "grey50"
  ) +
  geom_smooth(method = "lm") +
  labs(x = "Defensa", y = "HP")


grid.arrange(speed_defense_legendary, weight_defense_legendary,
  height_defense_legendary, hp_defense_legendary,
  ncol = 2
)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
# Muestra los gráficos en una cuadrícula de dos columnas
```

Comentarios sobre los gráficos de dispersión

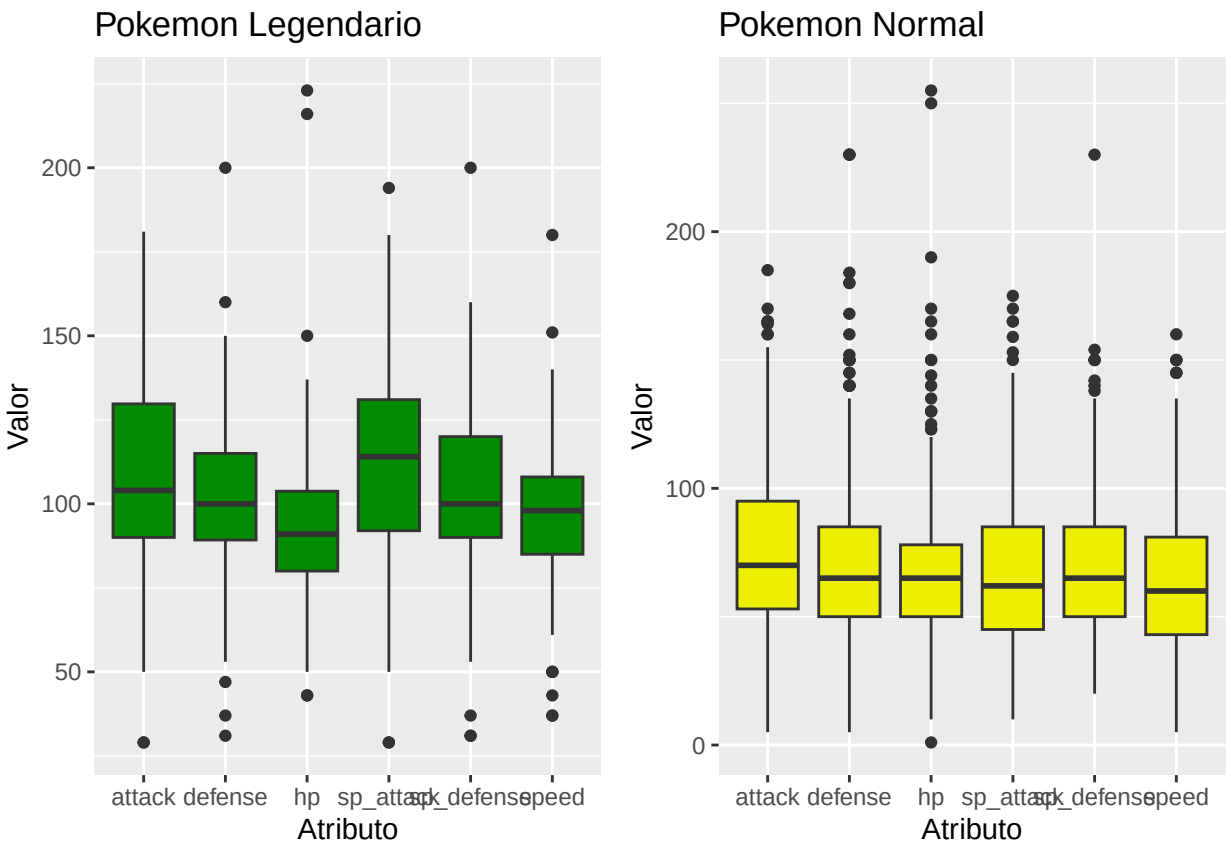
Podemos observar que tanto peso como altura estan ligeramente correlacionadas con defensa, mientras que HP y velocidad no parecen estarlo.

---

4. Gráficos de caja

```
# Seleccionamos las columnas que nos interesan
box_plot_attr <- select(
  df, type1, is_legendary, hp, defense, attack, sp_attack, sp_defense,
  speed
)
# Filtramos los datos para obtener solo los legendarios
box_plot_attr_leg <- filter(box_plot_attr, is_legendary == 1)
```

```
box_plot_attr_nor <- filter(box_plot_attr, is_legendary == 0)
box_plot_attr_leg_long <- gather(box_plot_attr_leg, attribute, value, -c(
  type1,
  is_legendary
))

box_plot_attr_nor_long <- gather(box_plot_attr_nor, attribute, value, -c(
  type1,
  is_legendary
))
# Creamos el gráfico de caja para los no legendarios y legendarios
bp_leg <- ggplot(data = box_plot_attr_leg_long, aes(attribute, value)) +
  geom_boxplot(fill = "green4") +
  ggtitle("Pokemon Legendario") +
  labs(x = "Atributo", y = "Valor")
bp_nor <- ggplot(data = box_plot_attr_nor_long, aes(attribute, value)) +
  geom_boxplot(
    fill = "yellow2"
  ) +
  ggtitle("Pokemon Normal") +
  labs(x = "Atributo", y = "Valor")
# Organizamos los gráficos en una cuadrícula
grid.arrange(bp_leg, bp_nor, ncol = 2)
```



Comentarios sobre los gráficos de caja

Podemos observar que los Pokémon legendarios tienen menos valores atípicos en comparación con los Pokémon normales. Además, los Pokémon legendarios tienden a tener valores más altos en las características de ataque, defensa, ataque especial, defensa especial y velocidad.

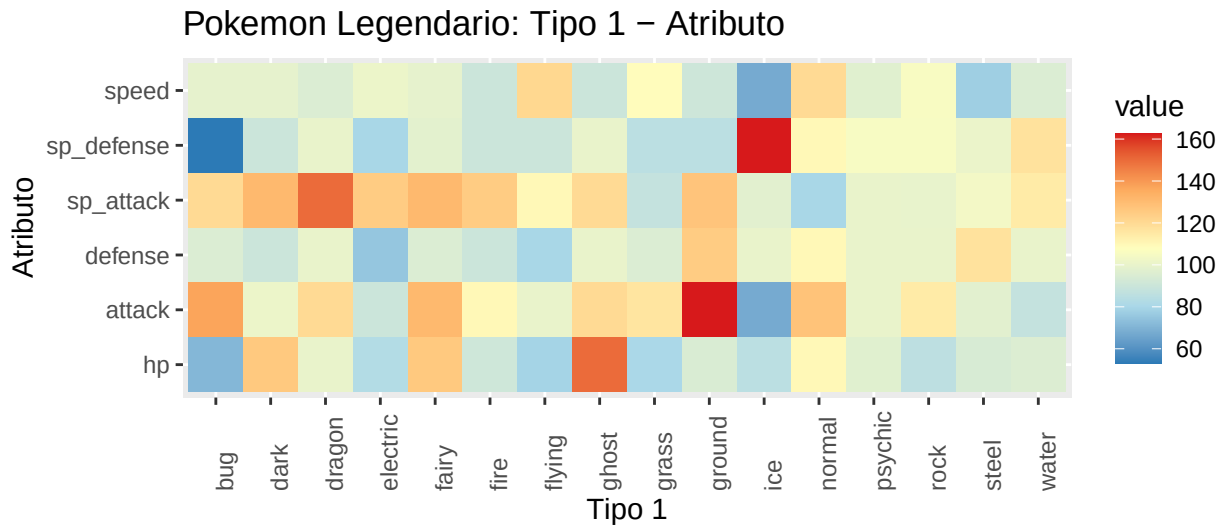
---

5. Mapas de calor

```
hmap_attr <- select(
  df, type1, is_legendary, hp,
  defense, attack, sp_attack, sp_defense, speed
)

hmap_attr_leg <- filter(hmap_attr, is_legendary == 1)

hmap_attr_leg <- group_by(hmap_attr_leg, type1)

hmap_attr_leg <- summarise(hmap_attr_leg,
  hp = median(hp), attack = median(attack),
  defense = median(defense), sp_attack = median(sp_attack),
  sp_defense = median(sp_defense),
  speed = median(speed)
)

hmap_attr_leg_m <- melt(hmap_attr_leg)
```

```
## Using type1 as id variables
```

```
hm_palette <- colorRampPalette(rev(brewer.pal(5, "RdYlBu")), space = "Lab")

ggplot(data = hmap_attr_leg_m, aes(type1, variable)) +
  geom_tile(aes(fill = value)) +
  ggtitle("Pokemon Legendario: Tipo 1 - Atributo") +
  scale_fill_gradientn(
    colours =
      hm_palette(100)
  ) +
  theme(axis.text.x = element_text(angle = 90, hjust = 0)) +
  coord_equal() +
  labs(x = "Tipo 1", y = "Atributo")
```



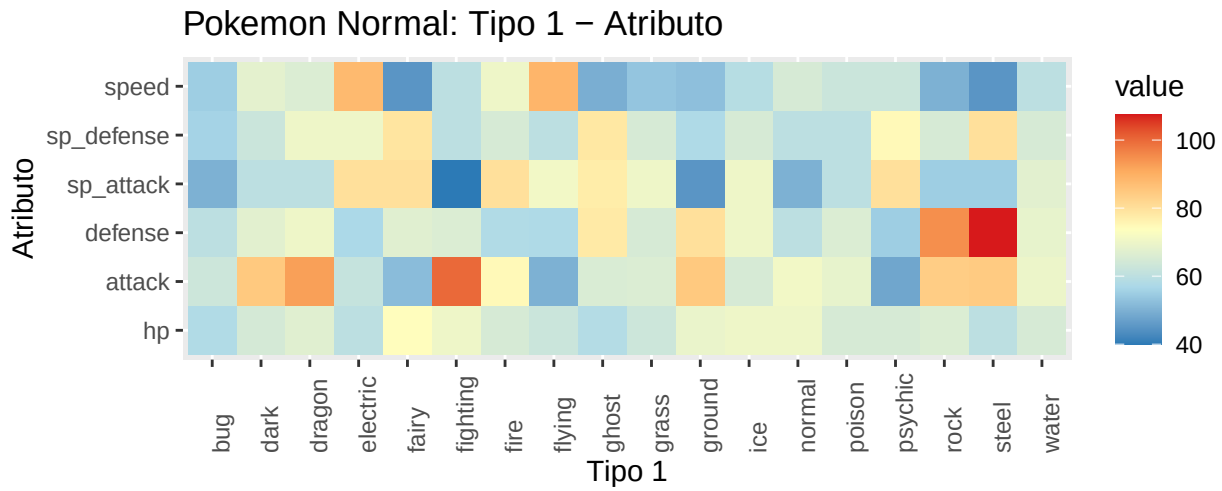
Podemos observar que los pokemon legendarios de tipo tierra tiene valores de ataque altos, los hielo tienen defensa especial alta.

```
hmap_attr <- select(
  df, type1, is_legendary, hp, defense, attack,
  sp_attack, sp_defense, speed
)

hmap_attr_nor <- filter(hmap_attr, is_legendary == 0)

hmap_attr_nor <- group_by(hmap_attr_nor, type1)

hmap_attr_nor <- summarise(hmap_attr_nor,
  hp = median(hp), attack = median(attack),
  defense = median(defense), sp_attack = median(sp_attack),
  sp_defense = median(sp_defense),
  speed = median(speed)
)

hmap_attr_nor_m <- melt(hmap_attr_nor)
```

```
## Using type1 as id variables
```

```
hm_palette <- colorRampPalette(rev(brewer.pal(5, "RdYlBu")), space = "Lab")

ggplot(data = hmap_attr_nor_m, aes(type1, variable)) +
```

```
  geom_tile(aes(fill = value)) +
  ggtitle("Pokemon Normal: Tipo 1 - Atributo") +
  scale_fill_gradientn(
    colours =
      hm_palette(100)
  ) +
  theme(axis.text.x = element_text(angle = 90, hjust = 0)) +
  coord_equal() +
  labs(x = "Tipo 1", y = "Atributo")
```

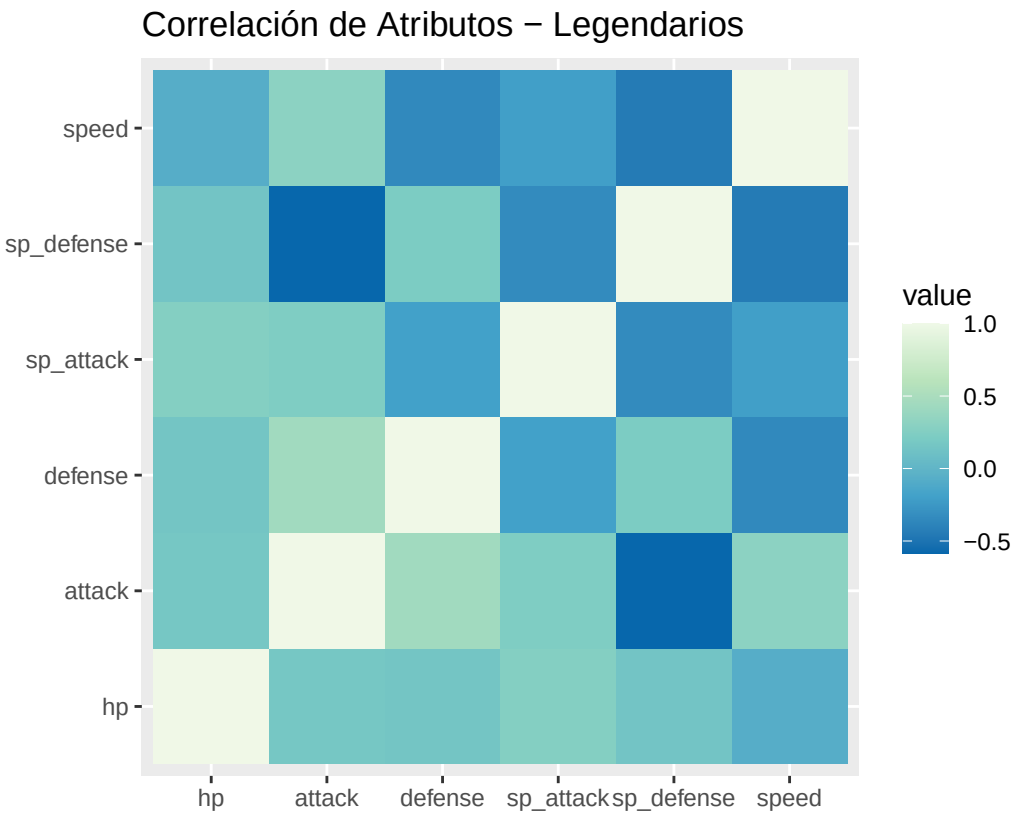



```
hmap_attr <- select(
  df, type1, is_legendary, hp, defense,
  attack, sp_attack, sp_defense, speed
)

hmap_attr_leg <- filter(hmap_attr, is_legendary == 1)

hmap_attr_leg <- group_by(hmap_attr_leg, type1)

hmap_attr_leg <- summarise(hmap_attr_leg,
  hp = median(hp), attack = median(attack),
  defense = median(defense), sp_attack = median(sp_attack),
  sp_defense = median(sp_defense),
  speed = median(speed)
)
```

```r
row.names(hmap_attr_leg) <- hmap_attr_leg$type1
```

```
## Warning: Setting row names on a tibble is deprecated.
```

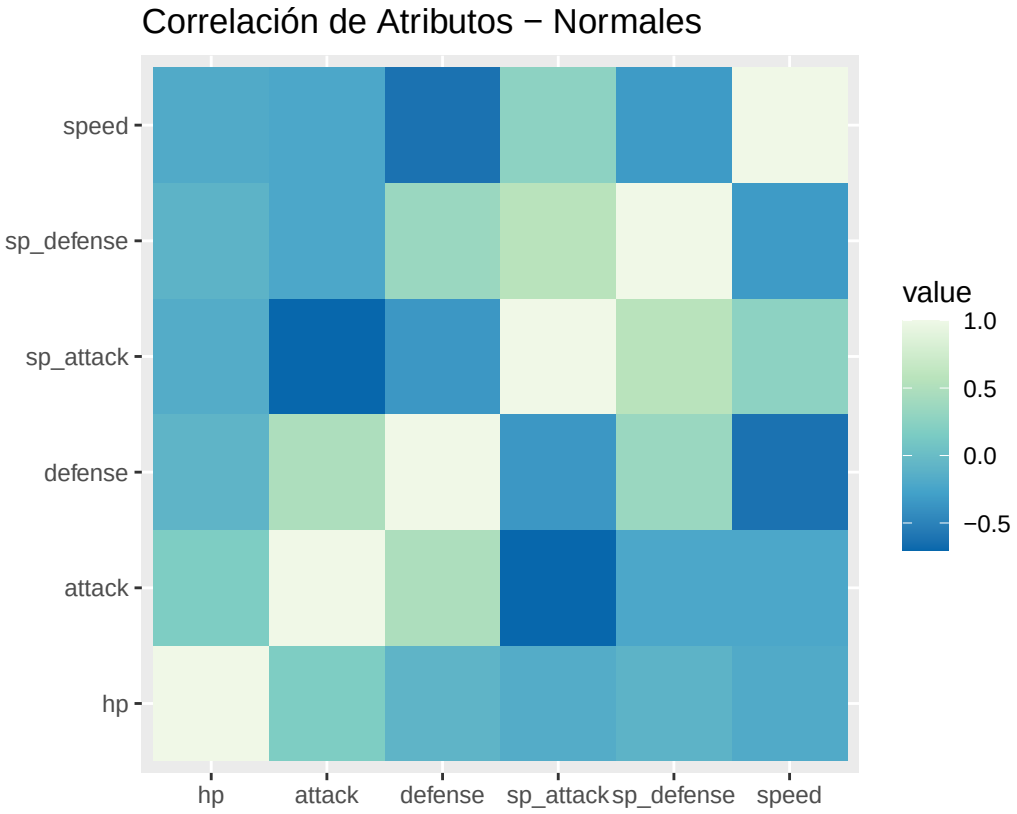
```r
hmap_attr_leg$type1 <- NULL

hmap_attr_leg$is_legendary <- NULL

hmap_attr_leg_cor <- cor(hmap_attr_leg)

hmap_attr_leg_cor_m <- melt(hmap_attr_leg_cor)

hm_palette <- colorRampPalette(rev(brewer.pal(5, "GnBu")), space = "Lab")

ggplot(data = hmap_attr_leg_cor_m, aes(Var1, Var2)) +
  geom_tile(aes(fill = value)) +
  ggtitle("Correlación de Atributos - Legendarios") +
  scale_fill_gradientn(
    colours =
      hm_palette(100)
  ) +
  coord_equal() +
  labs(x = "", y = "")
```



```r
hmap_attr <- select(
  df, type1, is_legendary, hp,
  defense, attack, sp_attack, sp_defense, speed
)

hmap_attr_nor <- filter(hmap_attr, is_legendary == 0)

hmap_attr_nor <- group_by(hmap_attr_nor, type1)

hmap_attr_nor <- summarise(hmap_attr_nor,
  hp = median(hp), attack = median(attack),
  defense = median(defense), sp_attack = median(sp_attack),
  sp_defense = median(sp_defense),
  speed = median(speed)
)

row.names(hmap_attr_nor) <- hmap_attr_nor$type1
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```r
hmap_attr_nor$type1 <- NULL

hmap_attr_nor$is_legendary <- NULL

hmap_attr_nor_cor <- cor(hmap_attr_nor)

hmap_attr_nor_cor_m <- melt(hmap_attr_nor_cor)

hm_palette <- colorRampPalette(rev(brewer.pal(5, "GnBu")), space = "Lab")

ggplot(data = hmap_attr_nor_cor_m, aes(Var1, Var2)) +
  geom_tile(aes(fill = value)) +
  ggtitle("Correlación de Atributos - Normales") +
  scale_fill_gradientn(
    colours =
      hm_palette(100)
  ) +
  coord_equal() +
  labs(x = "", y = "")
```




```
df <- (read.csv("pokemon.csv", header = TRUE, encoding = "UTF-8"))
attach(df)
```

```
## The following objects are masked from df (pos = 3):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 4):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg
```

```
## The following objects are masked from df (pos = 5):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 6):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 7):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 8):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 9):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
```

```
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 10):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 11):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 13):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg

## The following objects are masked from df (pos = 14):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg
```

```
## The following objects are masked from df (pos = 15):
##
##     abilities, against_bug, against_dark, against_dragon,
##     against_electric, against_fairy, against_fight, against_fire,
##     against_flying, against_ghost, against_grass, against_ground,
##     against_ice, against_normal, against_poison, against_psychic,
##     against_rock, against_steel, against_water, attack, base_egg_steps,
##     base_happiness, base_total, capture_rate, classfication, defense,
##     experience_growth, generation, height_m, hp, is_legendary, name,
##     percentage_male, pokedex_number, sp_attack, sp_defense, speed,
##     type1, type2, weight_kg
```

```
df <- tibble::as_tibble(df)
colnames(df)[25] <- "classification"
df$capture_rate <- as.numeric(df$capture_rate)
```

```
## Warning: NAs introduced by coercion
```

```
df_fight_against <- select(df, type1, against_bug:against_water)
head(df_fight_against)
```

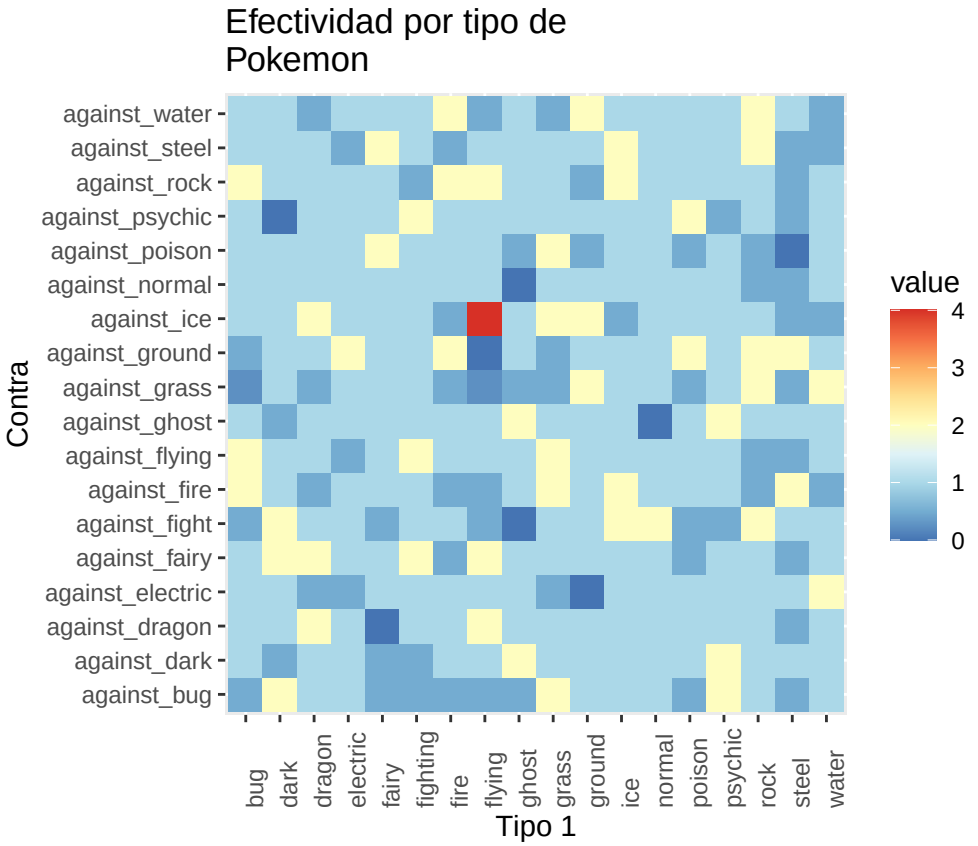
```
## # A tibble: 6 x 19
##   type1 against_bug against_dark against_dragon against_electric against_fairy
##   <chr>       <dbl>        <dbl>          <dbl>            <dbl>         <dbl>
## 1 grass         1              1              1              0.5           0.5
## 2 grass         1              1              1              0.5           0.5
## 3 grass         1              1              1              0.5           0.5
## 4 fire          0.5            1              1              1             0.5
## 5 fire          0.5            1              1              1             0.5
## 6 fire          0.25           1              1              2             0.5
## # i 13 more variables: against_fight <dbl>, against_fire <dbl>,
## #   against_flying <dbl>, against_ghost <dbl>, against_grass <dbl>,
## #   against_ground <dbl>, against_ice <dbl>, against_normal <dbl>,
## #   against_poison <dbl>, against_psychic <dbl>, against_rock <dbl>,
## #   against_steel <dbl>, against_water <dbl>
```

```
df_fight_against_g <- group_by(df_fight_against, type1)
df_fight_against_summ <- summarise(df_fight_against_g,
  against_bug = median(against_bug),
  against_dark = median(against_dark),
  against_dragon = median(against_dragon),
  against_electric = median(against_electric),
  against_fairy = median(against_fairy),
  against_fight = median(against_fight),
  against_fire = median(against_fire),
  against_flying = median(against_flying),
  against_ghost = median(against_ghost),
  against_grass = median(against_grass),
  against_ground = median(against_ground),
  against_ice = median(against_ice),
  against_normal = median(against_normal),
  against_poison = median(against_poison),
```

```
  against_psychic = median(against_psychic),
  against_rock = median(against_rock),
  against_steel = median(against_steel),
  against_water = median(against_water)
)

df_fight_against_long <- melt(df_fight_against_summ)
```

```
## Using type1 as id variables
```

```
hm_palette <- colorRampPalette(rev(brewer.pal(9, "RdYlBu")), space = "Lab")
ggplot(data = df_fight_against_long, aes(type1, variable)) +
  geom_tile(aes(fill = value)) +
  scale_fill_gradientn(colours = hm_palette(100)) +
  coord_equal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 0)) +
  ggtitle("Efectividad por tipo de
Pokemon") +
  labs(x = "Tipo 1", y = "Contra")
```



Comentarios sobre los mapas de calor

En este mapa de calor de la correlación de los atributos de los pokemon legendarios podemos ver que la defensa especial esta negativamente correlacionada con la velocidad y el ataque.

En el mapa de calor de la correlación de los atributos de los pokemon normales podemos ver que la defensa especial esta negativamente correlacionada con la velocidad y el defensa, tambien una muy fuerte negativa entre ataque especial y ataque. En el caso de los pokemon normales podemos observar que los de tipo hierro tienen mucha defensa, los de tipo luchador tienen mucho ataque y los de tipo fantasma tienen suficiente defensa especial.

Se puede observar que los Pokemon de tipo hielo son muy efectivos contra los de tipo volador.