

# Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ciencias de la Computación e Informática

Diseño de Experimentos

Laboratorio 2 Análisis Exploratorio de Datos y gráficos depurados

2025



## **Autores**

Brandon Trigueros Lara C17899

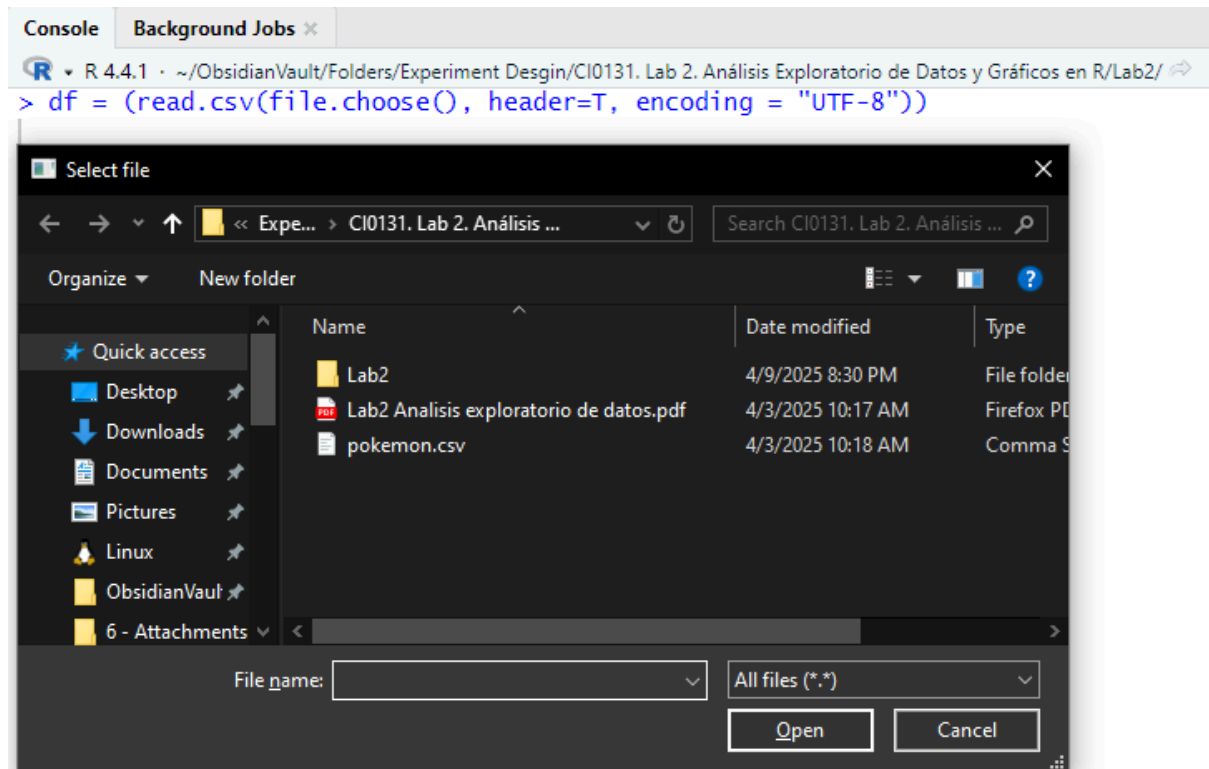
David González Villanueva C13388

## A. Primera parte

### A.1 Información general

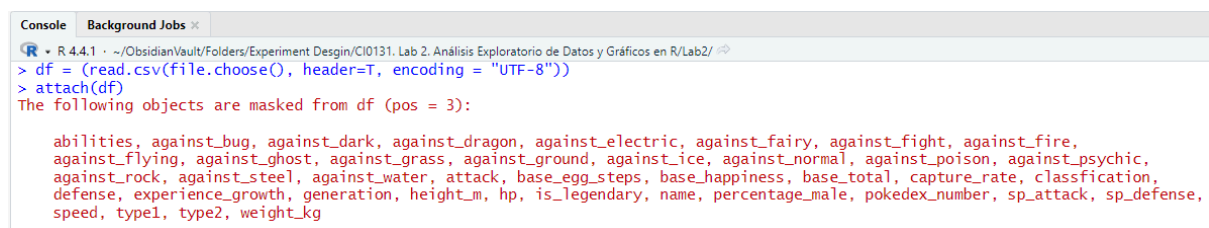
Comenzamos leyendo un archivo csv y guardándolo en el objeto df

```
7 # Se lee un archivo csv y se guarda en el objeto df
8 df = (read.csv(file.choose(), header=T, encoding = "UTF-8"))
```



Luego se convierte las columnas de df en variables disponibles directamente en el entorno global. Esto permite referenciarlas sin necesidad de escribir `df$nombre_columna`.

```
13 attach(df)
```



Se genera un resumen estadístico informativo para cada columna del data frame df dependiendo de su tipo de dato.

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

```
Console Background Jobs x
R 4.4.1 · ~/ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/
> summary(df)
abilities      against_bug      against_dark      against_dragon      against_electric      against_fairy      against_fight      against_fire
Length:801    Min.   :0.2500    Min.   :0.250    Min.   :0.0000    Min.   :0.000    Min.   :0.250    Min.   :0.000    Min.   :0.250
Class :character 1st Qu.:0.5000    1st Qu.:1.000    1st Qu.:1.0000    1st Qu.:0.500    1st Qu.:1.000    1st Qu.:0.500    1st Qu.:0.500
Mode :character Median :1.0000    Median :1.000    Median :1.0000    Median :1.000    Median :1.000    Median :1.000    Median :1.000
Mean :0.9963    Mean :1.057    Mean :0.9688    Mean :1.074    Mean :1.069    Mean :1.066    Mean :1.135
3rd Qu.:1.0000    3rd Qu.:1.000    3rd Qu.:1.0000    3rd Qu.:1.000    3rd Qu.:1.000    3rd Qu.:1.000    3rd Qu.:2.000
Max.   :4.0000    Max.   :4.000    Max.   :2.0000    Max.   :4.000    Max.   :4.000    Max.   :4.000    Max.   :4.000

against_flying against_ghost against_grass against_ground against_ice against_normal against_poison against_psychic against_rock
Min.   :0.250    Min.   :0.000    Min.   :0.250    Min.   :0.000    Min.   :0.250    Min.   :0.000    Min.   :0.0000    Min.   :0.000    Min.   :0.25
1st Qu.:1.000    1st Qu.:1.000    1st Qu.:0.500    1st Qu.:1.000    1st Qu.:0.500    1st Qu.:1.000    1st Qu.:0.5000    1st Qu.:1.000    1st Qu.:1.00
Median :1.000    Median :1.000    Median :1.000    Median :1.000    Median :1.000    Median :1.000    Median :1.0000    Median :1.000    Median :1.00
Mean :1.193    Mean :0.985    Mean :1.034    Mean :1.098    Mean :1.208    Mean :0.887    Mean :0.9753    Mean :1.005    Mean :1.25
3rd Qu.:1.000    3rd Qu.:1.000    3rd Qu.:1.000    3rd Qu.:1.000    3rd Qu.:2.000    3rd Qu.:1.000    3rd Qu.:1.0000    3rd Qu.:1.000    3rd Qu.:2.00
Max.   :4.000    Max.   :4.000    Max.   :4.000    Max.   :4.000    Max.   :4.000    Max.   :1.000    Max.   :4.0000    Max.   :4.000    Max.   :4.00

against_steel against_water attack base_egg_steps base_happiness base_total capture_rate classification
Min.   :0.2500    Min.   :0.250    Min.   : 5.00    Min.   :1280    Min.   : 0.00    Min.   :180.0    Length:801    Length:801
1st Qu.:0.5000    1st Qu.:0.500    1st Qu.:55.00    1st Qu.:5120    1st Qu.:70.00    1st Qu.:320.0    Class:character Class:character
Median :1.0000    Median :1.000    Median :75.00    Median :5120    Median :70.00    Median :435.0    Mode:character  Mode:character
Mean :0.9835    Mean :1.058    Mean :77.86    Mean :7191    Mean :65.36    Mean :428.4
3rd Qu.:1.0000    3rd Qu.:1.000    3rd Qu.:100.00    3rd Qu.:6400    3rd Qu.:70.00    3rd Qu.:505.0
Max.   :4.0000    Max.   :4.000    Max.   :185.00    Max.   :30720    Max.   :140.00    Max.   :780.0
```

Se muestra la estructura interna del objeto, revelando los tipos de cada columna y una vista resumida de los datos. Salida no requerida

```
20 str(df)
```

Se muestra una vista similar pero en un formato más compacto. Útil cuando se trabaja con data frames grandes. Salida no requerida

```
23 glimpse(df)
```

Se consultan los datos de las variables de interés

```
30 summary(attack)
```

```
Console Background Jobs x
R 4.4.1 · ~/ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/
> summary(attack)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00  55.00   75.00   77.86 100.00   185.00
```

```
32 summary(defense)
```

```
Console Background Jobs x
R 4.4.1 · ~/ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/
> summary(defense)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00  50.00   70.00   73.01  90.00   230.00
```

```
34 summary(hp)
```

```

Console Background Jobs x
R 4.4.1 ~./ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/
> summary(hp)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00   50.00   65.00   68.96   80.00  255.00

```

36 `summary(weight_kg)`

```

Console Background Jobs x
R 4.4.1 ~./ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/
> summary(weight_kg)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
   0.10    9.00   27.30   61.38   64.80  999.90    20

```

38 `summary(height_m)`

```

Console Background Jobs x
R 4.4.1 ~./ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/
> summary(height_m)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
   0.100   0.600   1.000   1.164   1.500   14.500    20

```

Se utilizan la función `table()` para calcular y mostrar las frecuencias de cada categoría en las variables categóricas indicadas

33 `table(df$type1)`

```

Console Background Jobs x
R 4.4.1 ~./ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/
> table(df$type1)
  bug    dark  dragon electric  fairy fighting  fire  flying  ghost  grass  ground  ice  normal  poison  psychic
  72    29    27    39        18    28        52    3    27    78    32    23    105    32    53
  rock  steel  water
  45    24    114

```

35 `table(df$type2)`

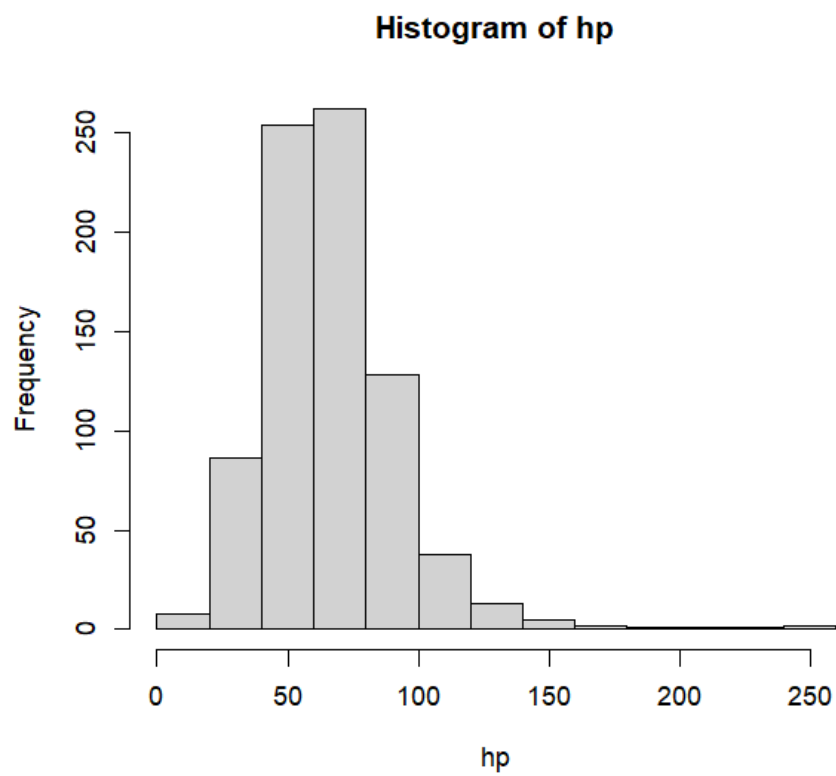
```

Console Background Jobs x
R 4.4.1 ~./ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/
> table(df$type2)
  384    bug    dark  dragon electric  fairy fighting  fire  flying  ghost  grass  ground  ice  normal  poison
  psychic  rock  steel  water
   29    14    22    17

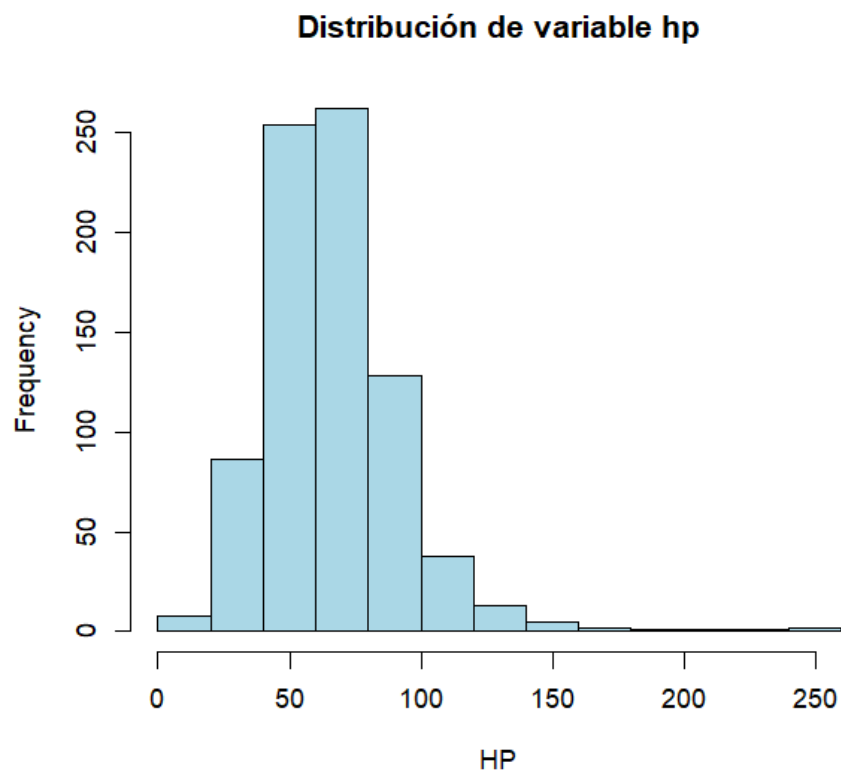
```

## A.2 Histogramas

Histograma para la variable `hp`.



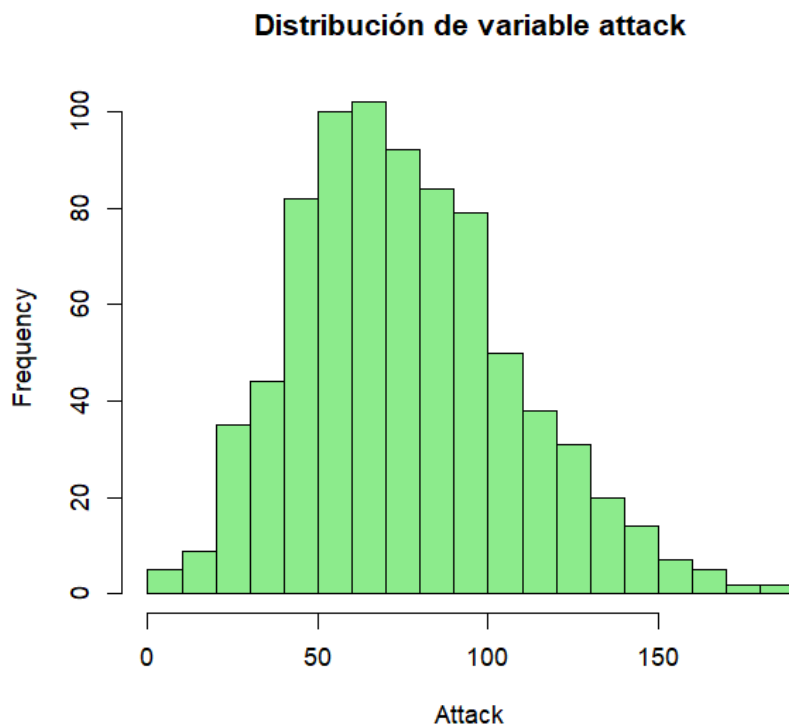
Histograma mejorado con título y etiquetas



Ahora creamos histogramas diferentes para el resto de las variables de interés:

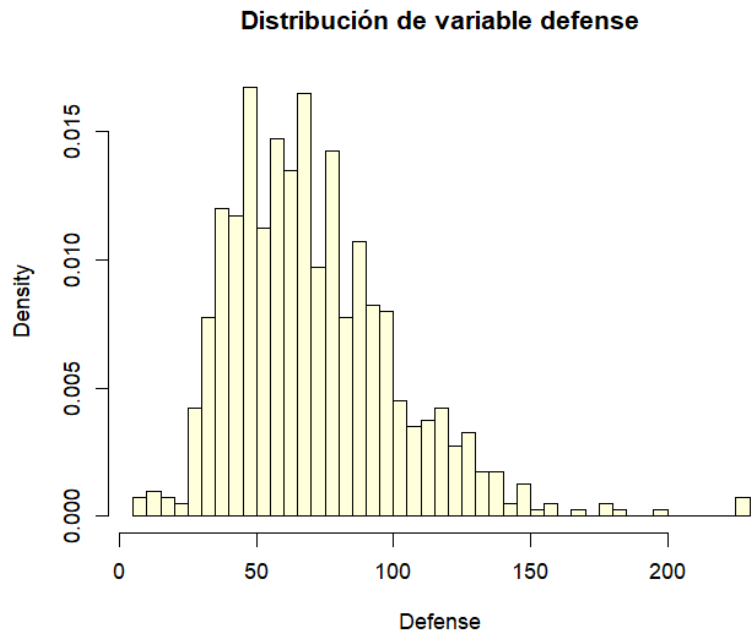
Histograma de attack

```
50 hist(attack,  
51     main = "Distribución de variable attack",  
52     xlab = "Attack",  
53     col = "lightgreen",  
54     border = "black",  
55     breaks = 15)
```



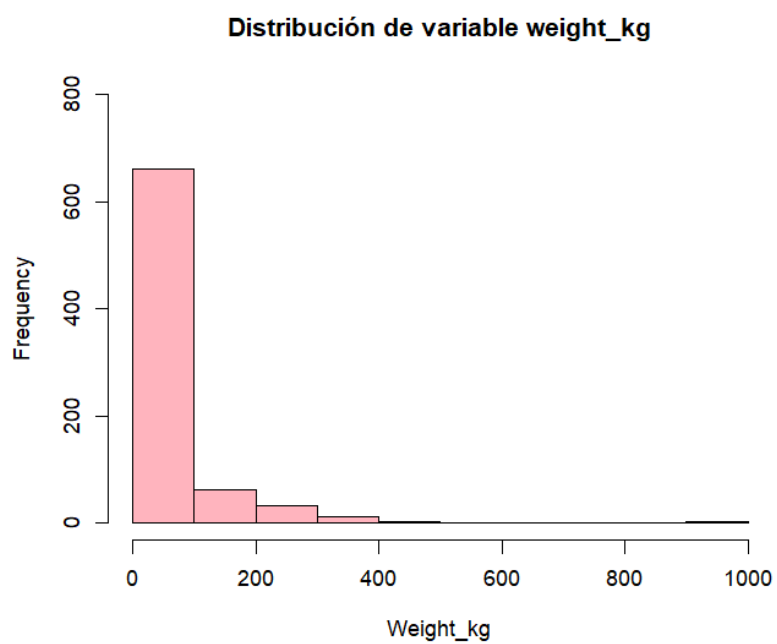
Histograma de defense

```
57 hist(defense,  
58     main = "Distribución de variable defense",  
59     xlab = "Defense",  
60     col = "lightyellow",  
61     border = "black",  
62     breaks = seq(min(defense), max(defense), by = 5),  
63     probability = TRUE)
```



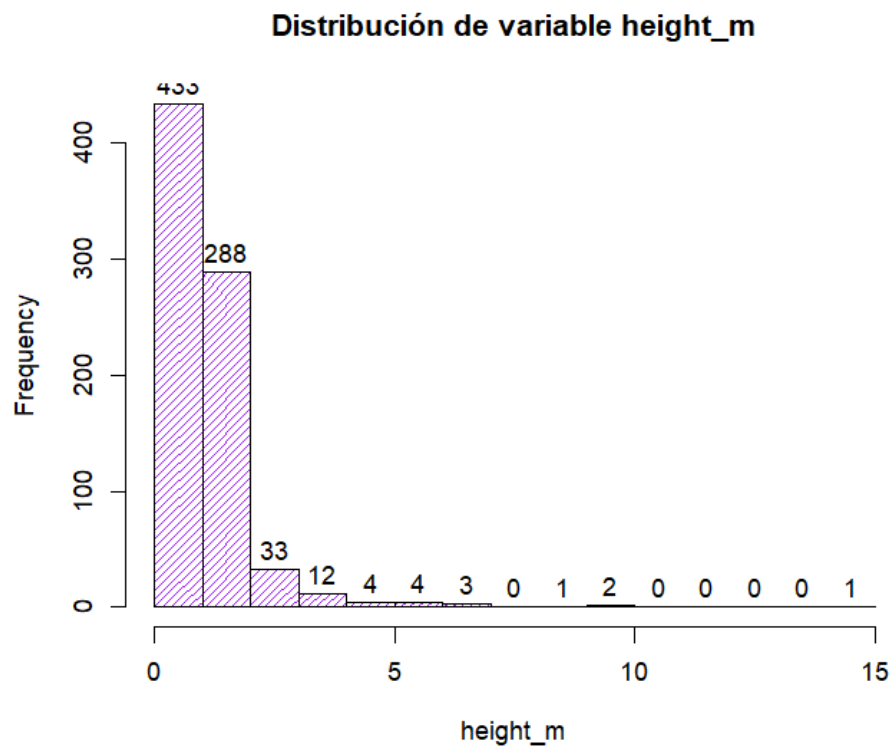
Histograma de peso en kilogramos

```
65 hist(weight_kg,  
66      main = "Distribución de variable weight_kg",  
67      xlab = "Weight_kg",  
68      col = "lightpink",  
69      border = "black",  
70      xlim = c(0, 1000),  
71      ylim = c(0, 800))
```



## Histograma de altura en metros

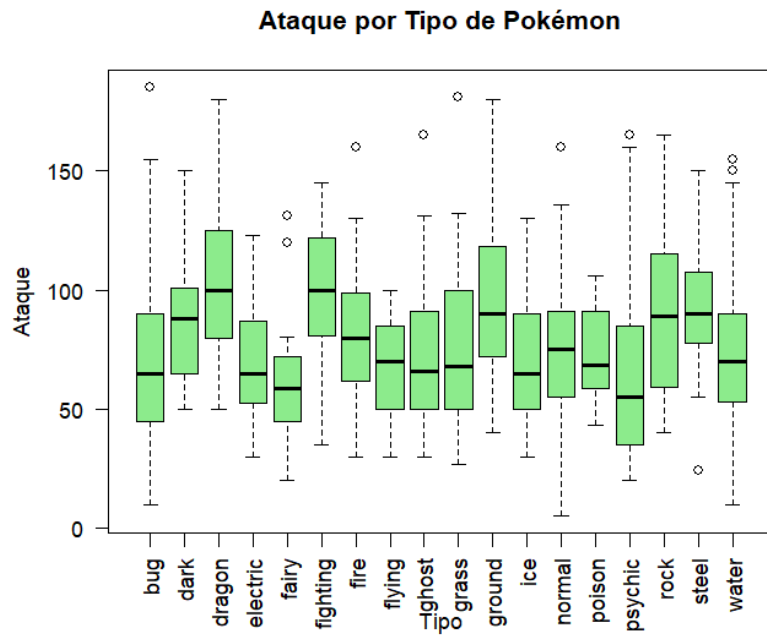
```
73 hist(height_m,  
74       main = "Distribución de variable height_m",  
75       xlab = "height_m",  
76       col = "purple",  
77       border = "black",  
78       labels = TRUE,  
79       density = 25,  
80       angle = 45,)
```



### A.3 Boxplots

Boxplot de la variable attack por type1



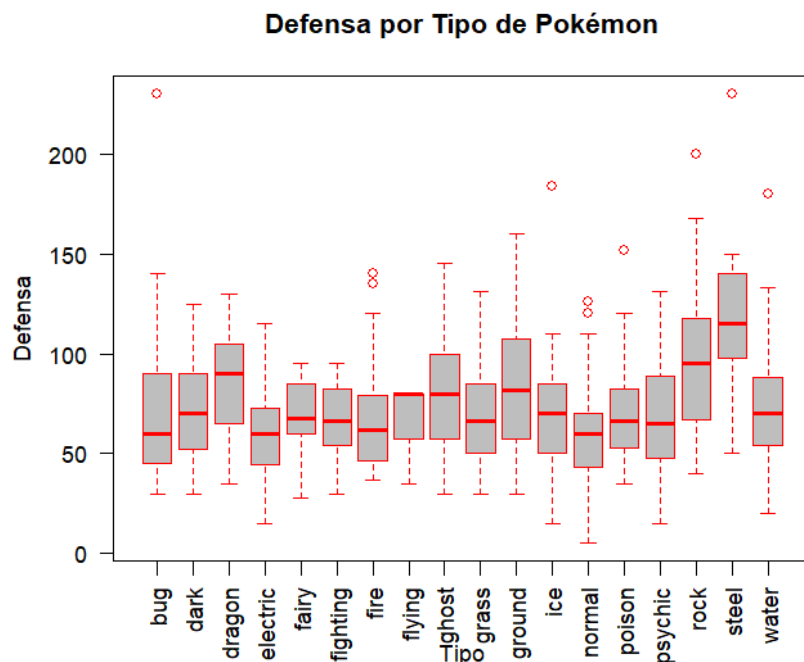


Para uno similar con defensa podemos hacer

```

93 boxplot(defense ~ type1,
94         data=df,
95         main="Defensa por Tipo de Pokémon",
96         xlab="Tipo",
97         ylab="Defensa",
98         las=2,
99         col="gray",
100         border = "red")
101

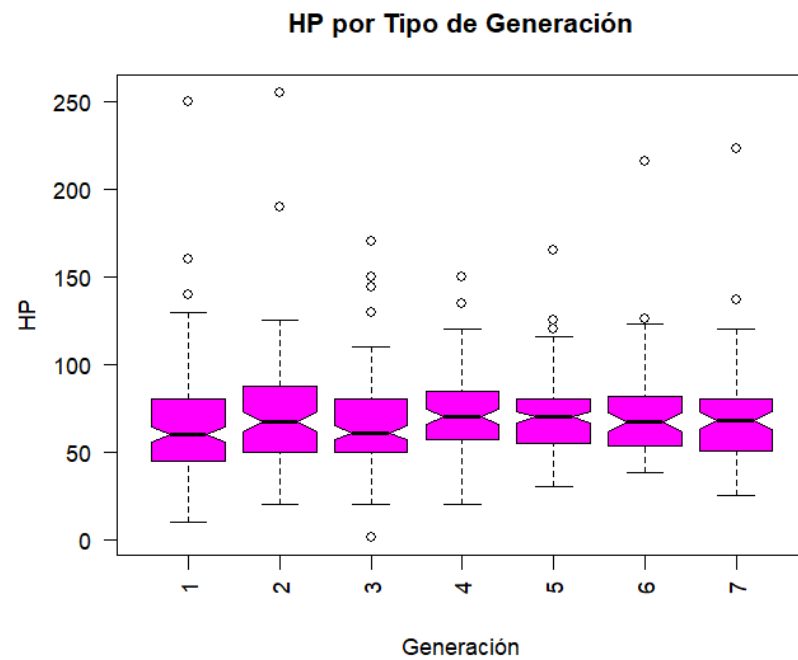
```



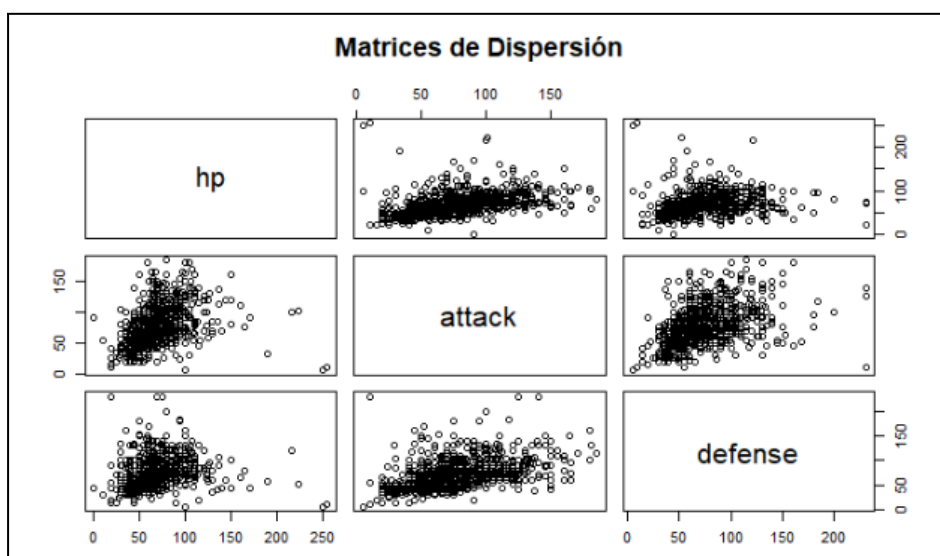
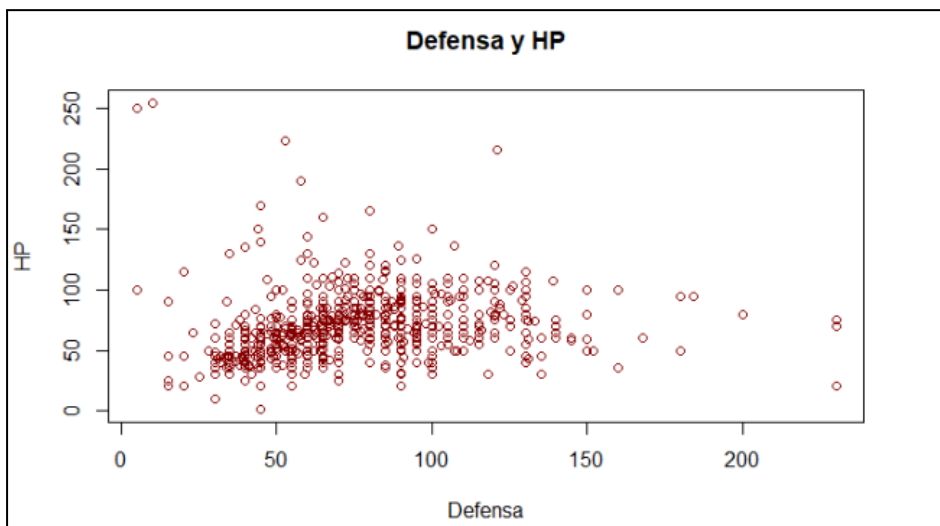
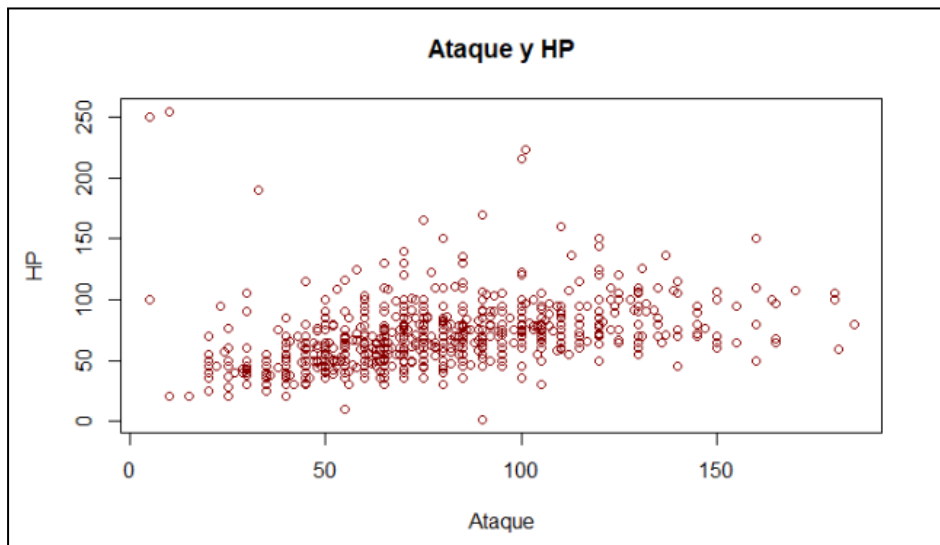
```

102 boxplot(hp ~ generation,
103         data=df,
104         main="HP por Tipo de Generación",
105         xlab="Generación",
106         ylab="HP",
107         las=2,
108         col="magenta",
109         notch = TRUE)

```



## A.4. Gráficos de dispersión



## B. Segunda parte

### Instalar paquetes

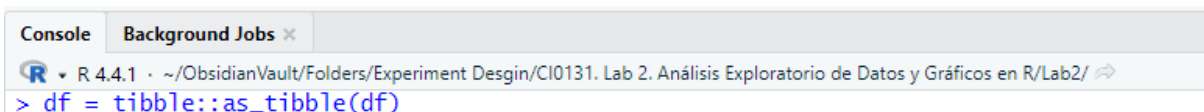
```
114 install.packages(c("dplyr",  
115   "ggplot2",  
116   "gridExtra",  
117   "tidyr",  
118   "reshape2",  
119   "RColorBrewer",  
120   "ggrepel"))
```

### Cargar paquetes

```
122 library(dplyr)  
123 library(ggplot2)  
124 library(gridExtra)  
125 library(tidyr)  
126 library(reshape2)  
127 library(RColorBrewer)  
128 library(ggrepel)
```

Una tibble es una data frame simplificada. Las tibbles son versiones de dataframes con algunas facilidades de impresión y uso

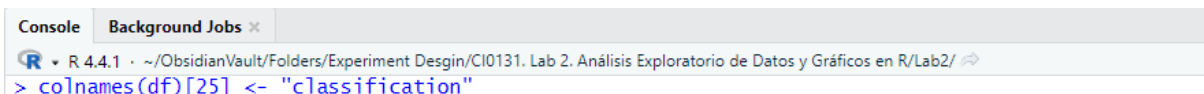
```
130 df = tibble::as_tibble(df)
```



Console Background Jobs x  
R 4.4.1 · ~/ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/ ↗  
> df = tibble::as\_tibble(df)

### Actualizar el nombre de una columna

```
132 colnames(df)[25] <- "classification"
```



Console Background Jobs x  
R 4.4.1 · ~/ObsidianVault/Folders/Experiment Desgin/CI0131. Lab 2. Análisis Exploratorio de Datos y Gráficos en R/Lab2/ ↗  
> colnames(df)[25] <- "classification"

### Convierte la columna capture\_rate de df a tipo numérico

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

```
> df$capture_rate <- as.numeric(df$capture_rate)  
Warning message:  
NAs introduced by coercion
```

Se muestran las primeras 6 filas del data frame

```
> head(df)
# A tibble: 6 x 40
# abilities
  against_bug against_dark against_dragon against_electric against_fairy against_fight against_fire against_flying against_ghost against_grass
  <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
1 ['Overgrow', 'Chlo... 1             1             1             0.5         0.5         0.5         2             2             1             0.25
2 ['Overgrow', 'Chlo... 1             1             1             0.5         0.5         0.5         2             2             1             0.25
3 ['Overgrow', 'Chlo... 1             1             1             0.5         0.5         0.5         2             2             1             0.25
4 ['Blaze', 'Solar P... 0.5           1             1             1             0.5         1             0.5         1             1             0.5
5 ['Blaze', 'Solar P... 0.5           1             1             1             0.5         1             0.5         1             1             0.5
6 ['Blaze', 'Solar P... 0.25          1             1             2             0.5         0.5         0.5         1             1             0.25
# i 29 more variables: 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>, experience_growth <int>, height_m <dbl>, hp <int>, name <chr>, percentage_male <dbl>, pokedex_number <int>,
# sp_attack <int>, sp_defense <int>, speed <int>, type1 <chr>, type2 <chr>, weight_kg <dbl>, generation <int>, is_legendary <int>
```

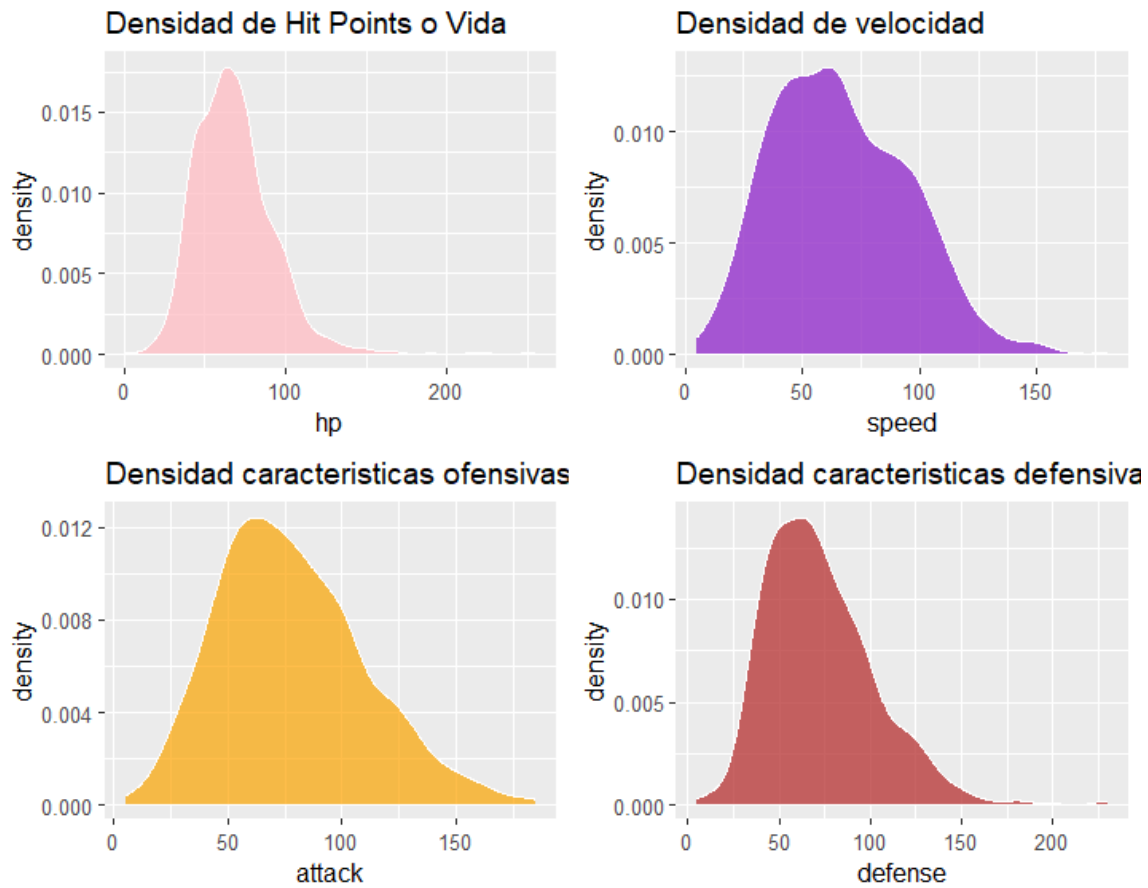
Se seleccionan algunas columnas para realizar las exploración y visualizaciones.

```
> head(df)
# A tibble: 6 x 16
# name
  name      classification  hp weight_kg height_m speed attack defense sp_attack sp_defense type1 type2 abilities generation is_legendary
  <chr>      <chr>          <int> <dbl>   <dbl> <int> <int> <int> <int> <int> <chr> <chr> <chr> <int> <int>
1 Bulbasaur Seed Pokémon    45    6.9     0.7    45    49    49    65    65 grass "pois..." ['Overgr... 1 0
2 Ivysaur   Seed Pokémon    60    13     1     60    62    63    80    80 grass "pois..." ['Overgr... 1 0
3 Venusaur  Seed Pokémon    80   100     2     80   100   123   122   120 grass "pois..." ['Overgr... 1 0
4 Charmander Lizard Pokémon    39    8.5     0.6    65    52    43    60    50 fire "" ['Blaze"... 1 0
5 Charmeleon Flame Pokémon    58    19     1.1    80    64    58    80    65 fire "" ['Blaze"... 1 0
6 Charizard Flame Pokémon    78   90.5    1.7   100   104    78   159   115 fire "flyi..." ['Blaze"... 1 0
# i 1 more variable: capture_rate <dbl>
```

## B.1. Gráficos de densidad de varios atributos de Pokémon.

Gráficos de densidad para HP, velocidad, ataque y defensa almacenados en variables. Me presenta los gráficos que cree en un grid

```
161 grid.arrange(density_hp, density_speed, density_attack, density_defense, ncol=2)
```

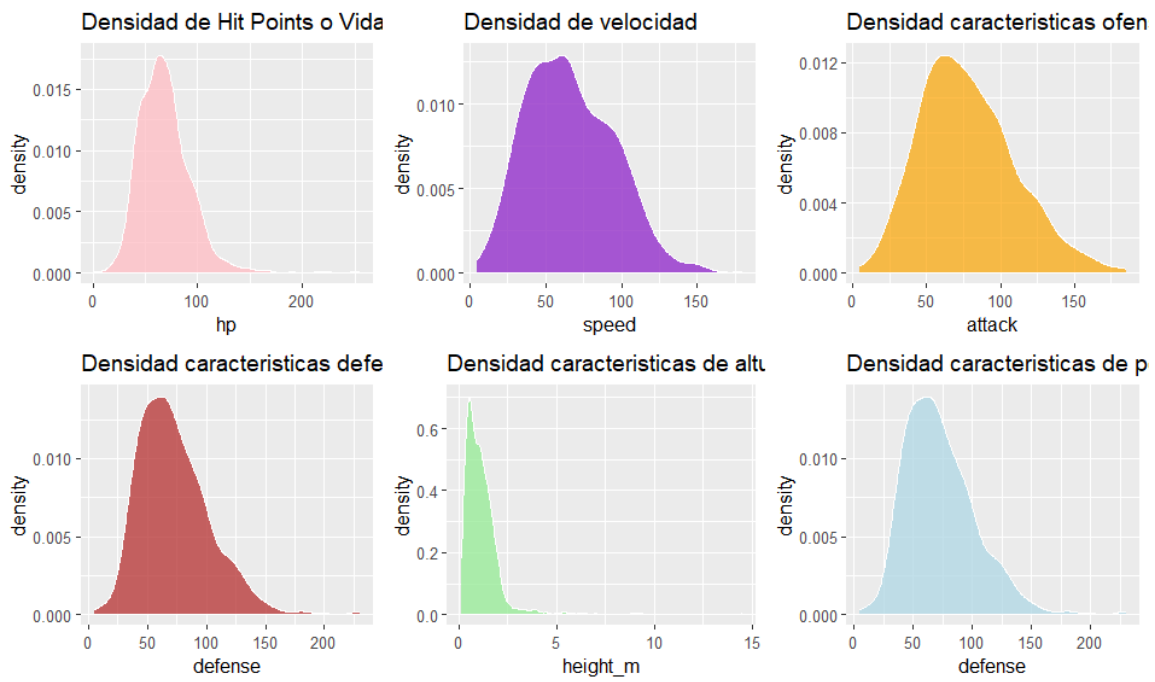


## Creación de dos nuevos gráficos de densidad

```

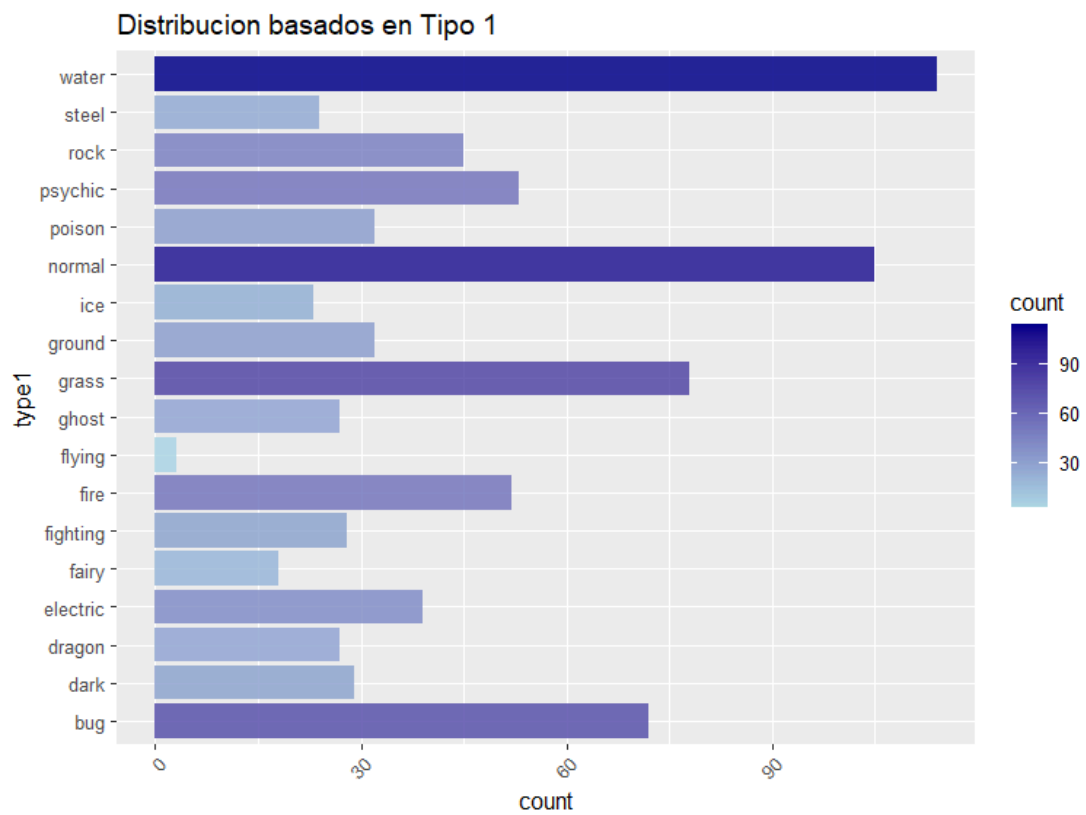
163 density_height_m <- ggplot(data=df, aes(height_m)) + geom_density(col="white", fill="lightgreen", alpha=0.7) +
164 ggtitle("Densidad características de altura")
165
166 density_weight_kg <- ggplot(data=df, aes(defense)) + geom_density(col="white", fill="lightblue", alpha=0.7) +
167 ggtitle("Densidad características de peso")

```



## B2. Diagrama de barras

Diagrama de barras del número basado en type1



Ahora creamos uno similar para el Tipo 2

```

180 ggplot(data=df, aes(type2)) +
181   geom_bar(aes(fill=..count..), alpha=0.85) +
182   scale_fill_gradient(low = "lightblue", high = "darkgreen") + # Gradiente de colores
183   theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
184   ggtitle("Distribucion basados en Tipo 1") +
185   coord_flip()

```

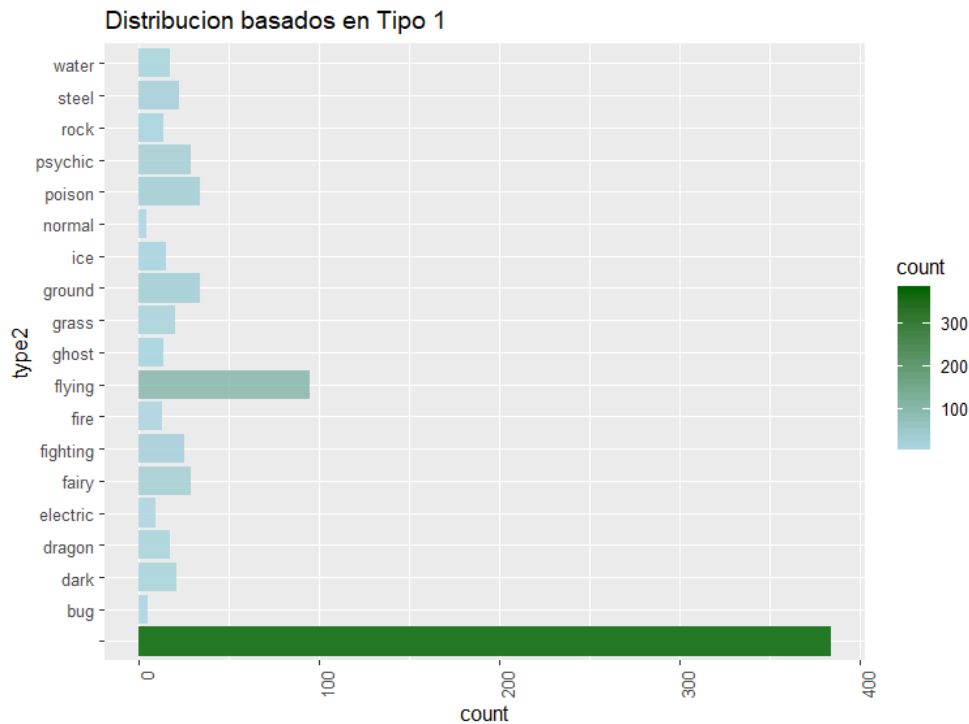


Gráfico para el número de Pokémon legendarios según su tipo primario (type1)

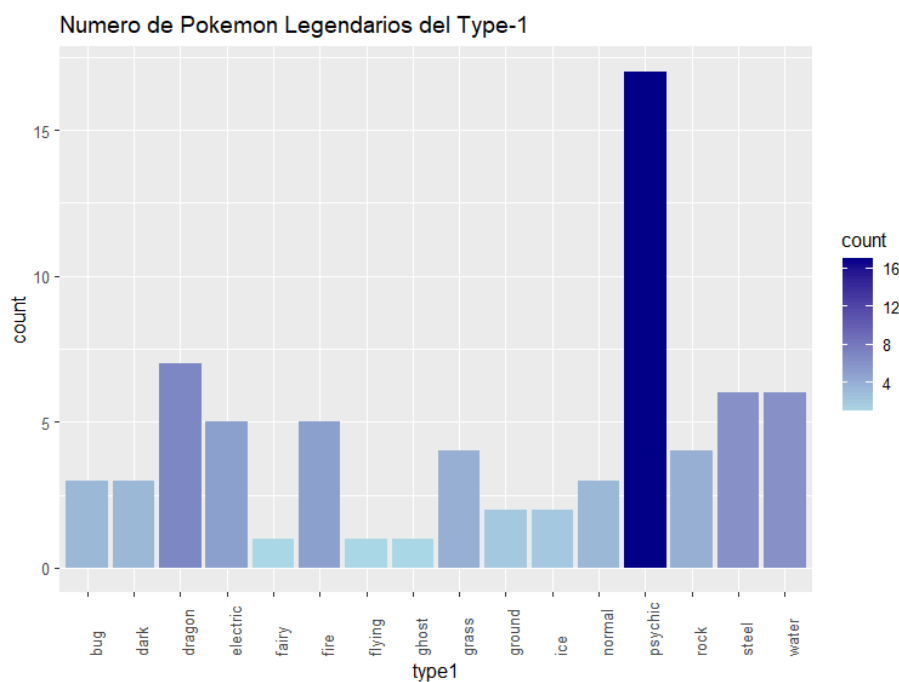


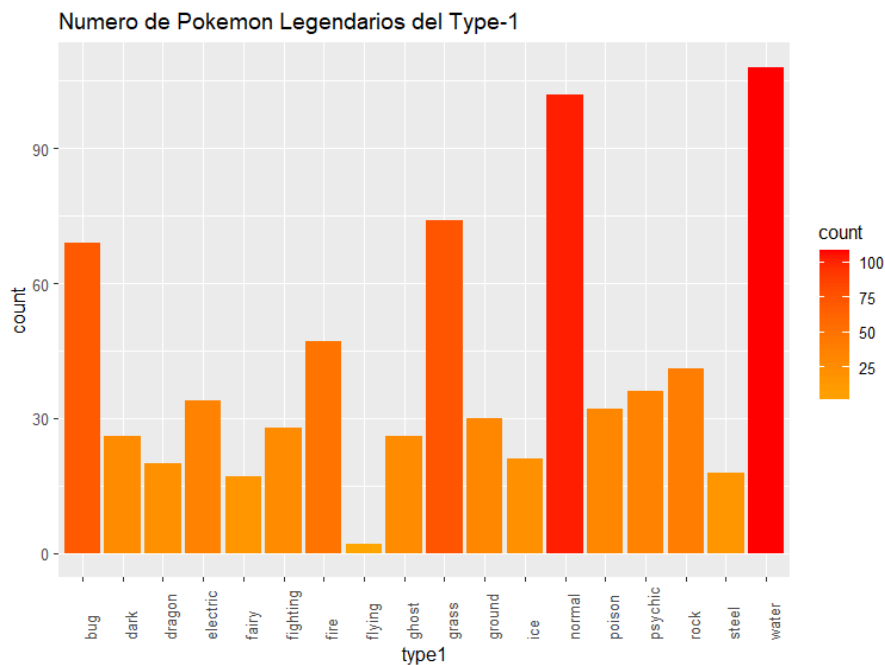
Gráfico similar al anterior pero para los no legendarios y con otros colores



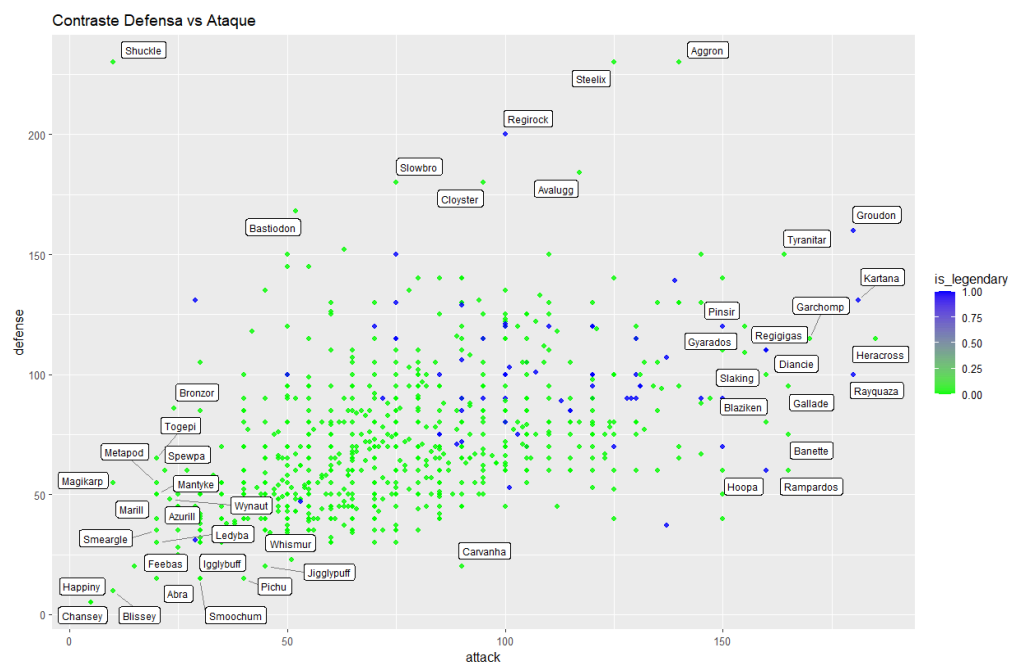
```

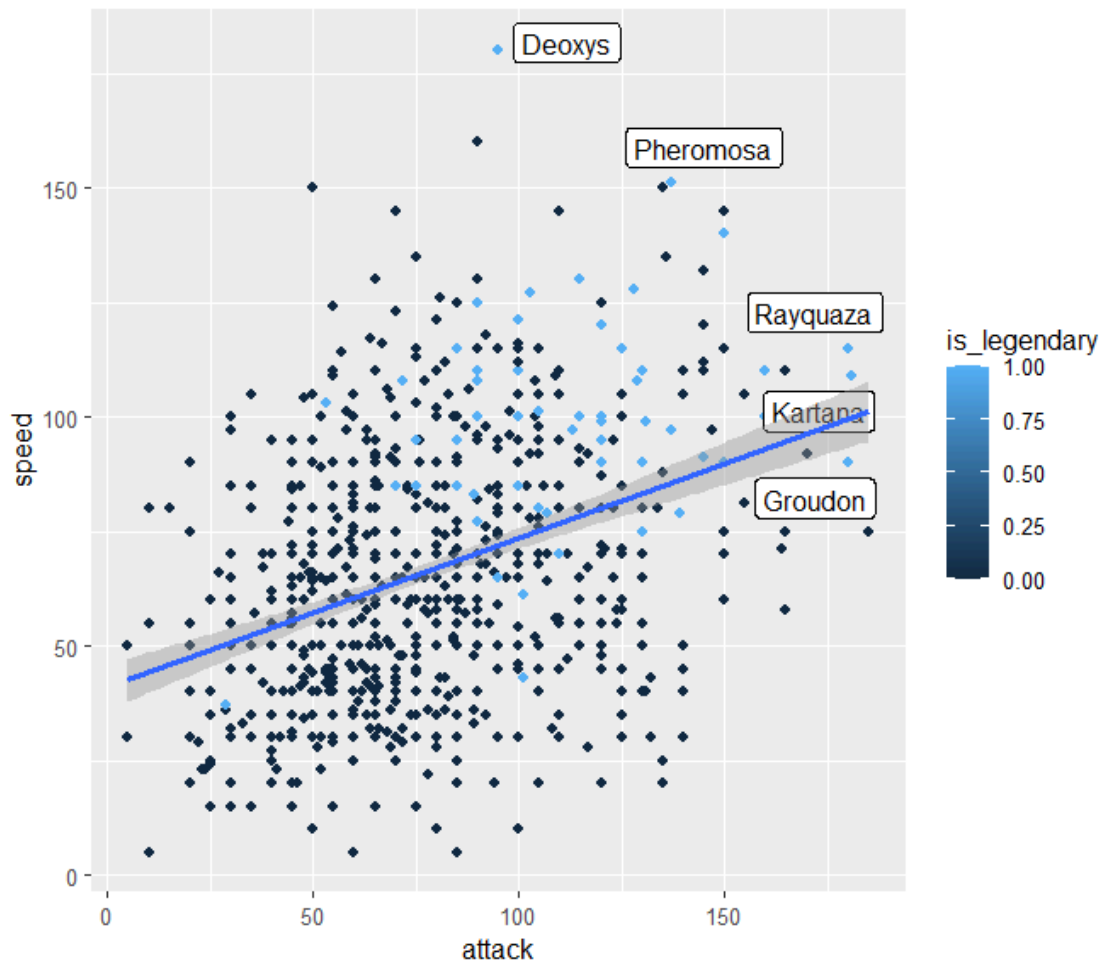
195 df %>%
196   filter(is_legendary==0) %>%
197   ggplot(aes(type1)) +
198   geom_bar(aes(fill= ..count..)) +
199   scale_fill_gradient(low = "orange", high = "red") + # Gradiente de colores
200   theme(axis.text.x = element_text(angle=90, hjust=0)) +
201   ggtitle("Numero de Pokemon Legendarios del Type-1")

```



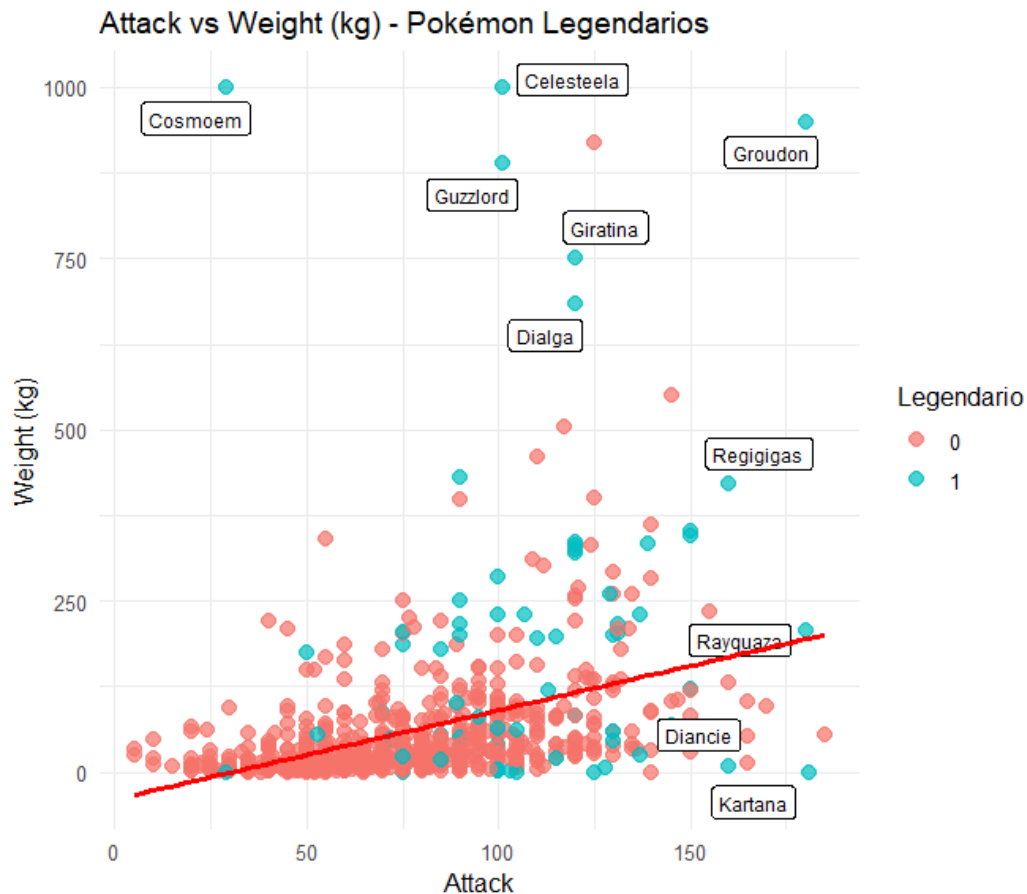
### B.3. Gráfico de dispersión – Scatterplots





Se etiqueta a aquellos legendarios que tienen *attack* mayor a 150 o *weight\_kg* mayor a 650.

```
220 attack_weight_legendary <- ggplot(na.omit(df), aes(x = attack, y = weight_kg)) +
221   geom_point(aes(color = factor(is_legendary)), size = 3, alpha = 0.7) +
222   geom_label_repel(
223     data = subset(df, (attack > 150 | weight_kg > 650) & is_legendary == 1),
224     aes(label = name),
225     box.padding = 0.35, point.padding = 0.5, segment.color = "grey50",
226     size = 3
227   ) +
228   geom_smooth(method = "lm", se = FALSE, color = "red") +
229   labs(
230     title = "Attack vs Weight (kg) - Pokémon Legendarios",
231     x = "Attack",
232     y = "Weight (kg)",
233     color = "Legendario"
234   ) +
235   theme_minimal()
236 print(attack_weight_legendary)
```

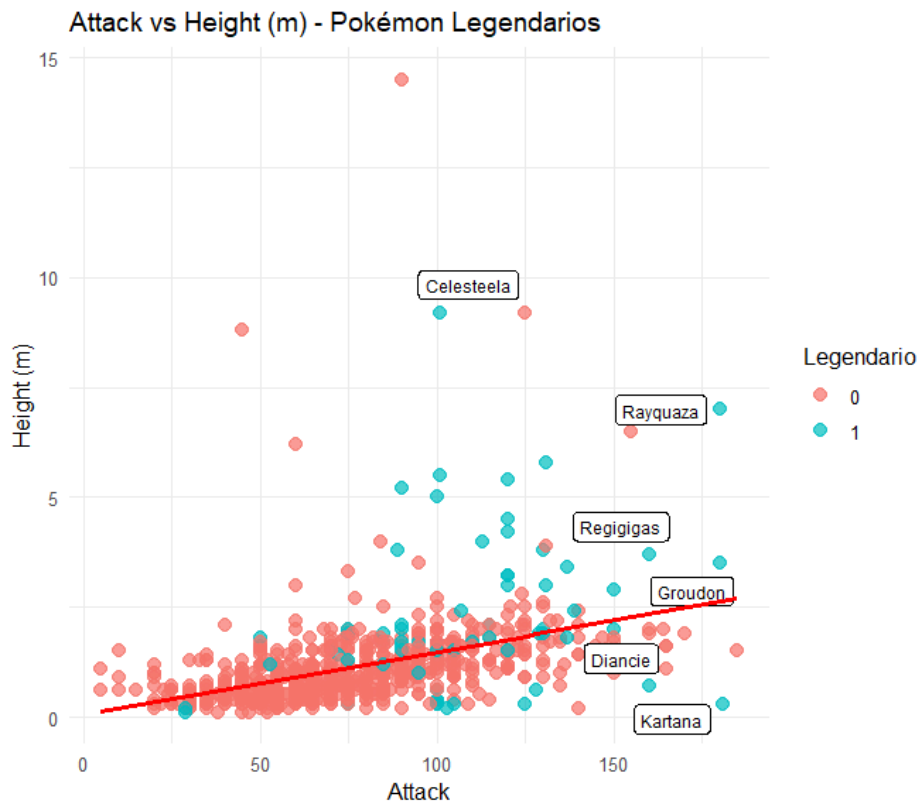


Se etiqueta a aquellos legendarios que tienen *attack* mayor a 150 o *height\_m* mayor a 7.5.

```

238 attack_height_legendary <- ggplot(na.omit(df), aes(x = attack, y = height_m)) +
239   geom_point(aes(color = factor(is_legendary)), size = 3, alpha = 0.7) +
240   geom_label_repel(
241     data = subset(df, (attack > 150 | height_m > 7.5) & is_legendary == 1),
242     aes(label = name),
243     box.padding = 0.35, point.padding = 0.5, segment.color = "grey50",
244     size = 3
245   ) +
246   geom_smooth(method = "lm", se = FALSE, color = "red") +
247   labs(
248     title = "Attack vs Height (m) - Pokémon Legendarios",
249     x = "Attack",
250     y = "Height (m)",
251     color = "Legendario"
252   ) +
253   theme_minimal()
254 print(attack_height_legendary)

```

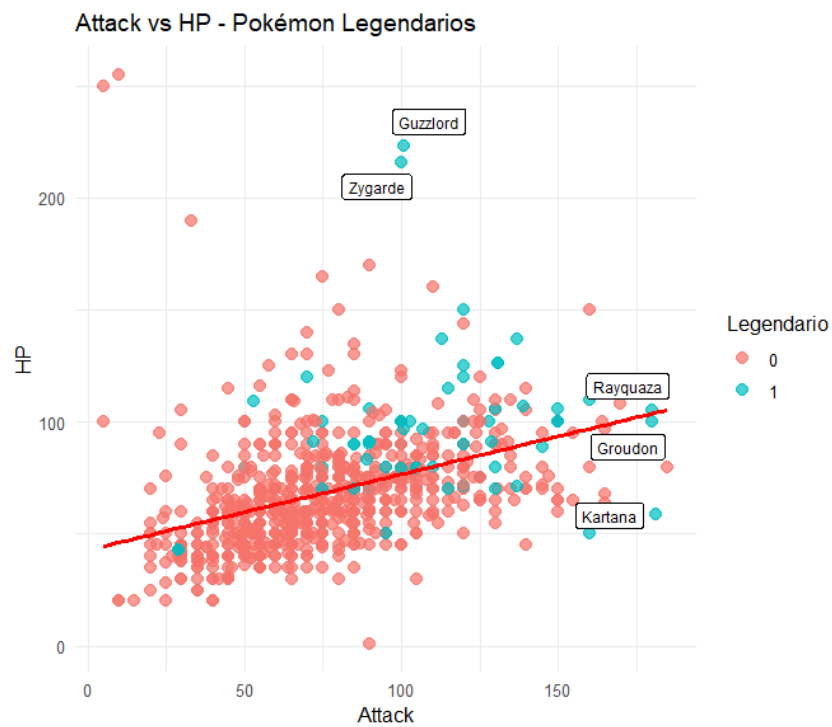


Se etiqueta a aquellos legendarios que tienen *attack* mayor a 170 o *hp* mayor a 190.

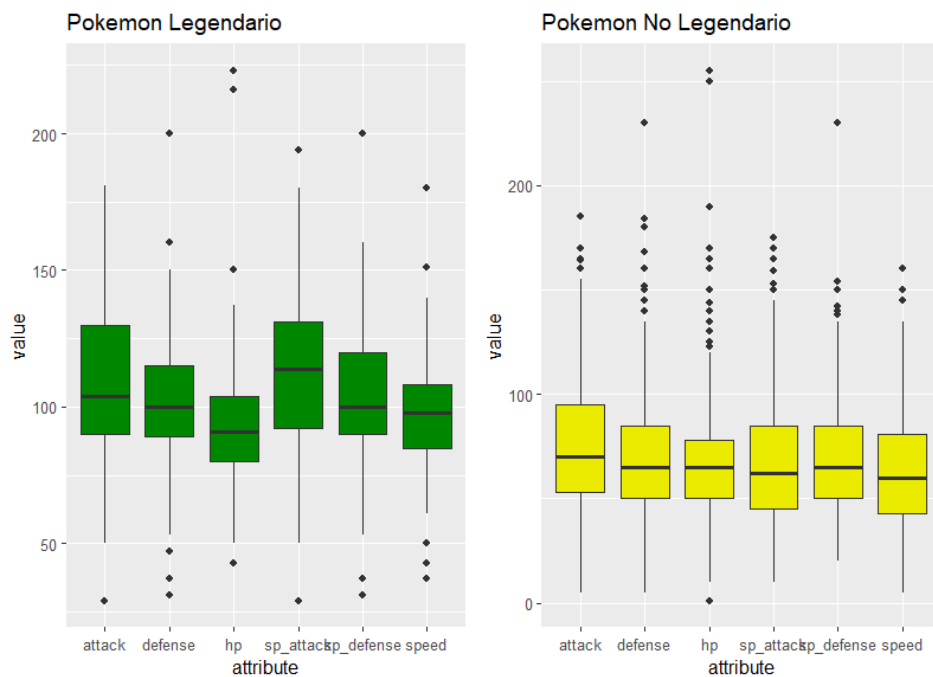
```

256 attack_hp_legendary <- ggplot(na.omit(df), aes(x = attack, y = hp)) +
257   geom_point(aes(color = factor(is_legendary)), size = 3, alpha = 0.7) +
258   geom_label_repel(
259     data = subset(df, (attack > 170 | hp > 190) & is_legendary == 1),
260     aes(label = name),
261     box.padding = 0.35, point.padding = 0.5, segment.color = "grey50",
262     size = 3
263   ) +
264   geom_smooth(method = "lm", se = FALSE, color = "red") +
265   labs(
266     title = "Attack vs HP - Pokémon Legendarios",
267     x = "Attack",
268     y = "HP",
269     color = "Legendario"
270   ) +
271   theme_minimal()
272 print(attack_hp_legendary)

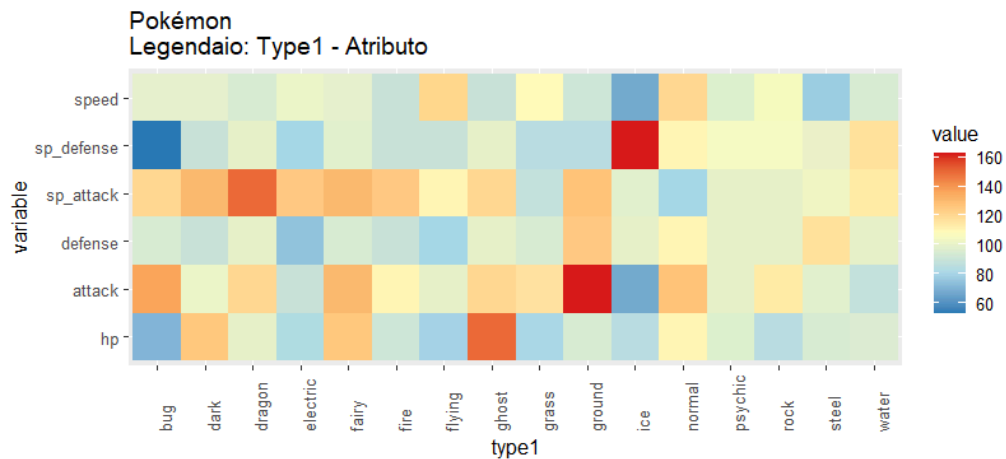
```



## B.4 Boxplots



En este momento utilizaremos mapas de calor para analizar los Pokémon tipo hielo

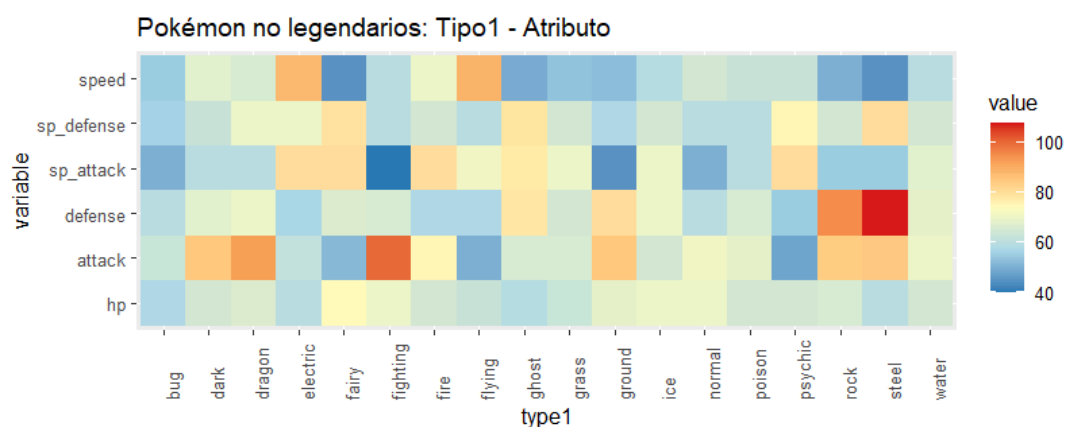


## B.5 Nuevo mapa de calor

```

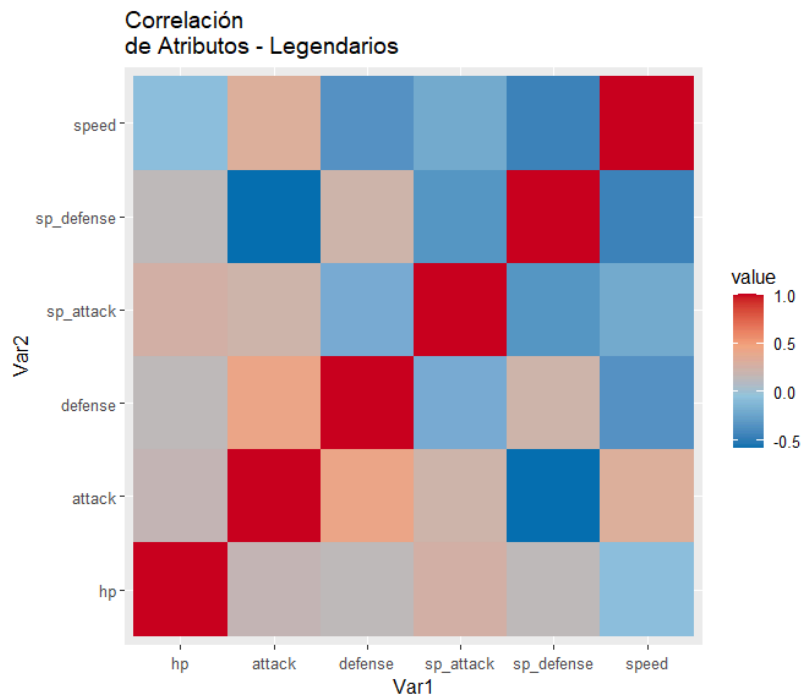
303 hmap_attr <- select(df, type1, is_legendary, hp, defense, attack, sp_attack, sp_defense, speed)
304 hmap_attr_non_leg <- filter(hmap_attr, is_legendary == 0)
305 hmap_attr_non_leg <- group_by(hmap_attr_non_leg, type1)
306 hmap_attr_non_leg <- summarise(hmap_attr_non_leg,
307   hp = median(hp),
308   attack = median(attack),
309   defense = median(defense),
310   sp_attack = median(sp_attack),
311   sp_defense = median(sp_defense),
312   speed = median(speed)
313 )
314 hmap_attr_non_leg_m <- melt(hmap_attr_non_leg)
315 hm.palette <- colorRampPalette(rev(brewer.pal(5, "RdYlBu"))), space = "Lab")
316 ggplot(data = hmap_attr_non_leg_m, aes(x = type1, y = variable)) +
317   geom_tile(aes(fill = value)) +
318   ggtitle("Pokémon no legendarios: Tipo1 - Atributo") +
319   scale_fill_gradientn(colours = hm.palette(100)) +
320   theme(axis.text.x = element_text(angle = 90, hjust = 0)) +
321   coord_equal()

```



## B.6. Matriz de Correlación

Verificar conjeturas con heatmap de correlación



Mapa de

Calor: Medianas de Efectividad (against\_type) por Tipo Primario de Pokémon

