CI-0131 Diseño de Experimentos
Ciclo I-2025
Profesor: Ignacio Díaz Oreiro

Laboratorio #2: Análisis Exploratorio de Datos y gráficos depurados

Introducción

El análisis exploratorio de datos (EDA, por sus siglas en inglés) es un proceso que permite la identificación de las tendencias dentro de los conjuntos de los datos. Durante el desarrollo de un experimento es importante que se logre identificar la presencia de algunas de estas tendencias o medidas como la existencia de valores faltantes, valores extremos, medidas de valores centrales o de dispersión. Esta información valiosa nos da acceso a relevante conocimiento acerca de nuestro conjunto de datos.

Por ejemplo, la presencia de valores faltantes nos puede indicar o permitirnos pensar que durante la recolección de datos algo no fue tan riguroso como se planteó. También que exista algún sesgo en el conjunto de datos, lo que potencialmente podamos evidenciar mediante un análisis visual.

La presencia de valores extremos podría ser un indicador de posibles errores de muestreo, de problemas en la consecución de datos o inclusive de un problema causado por las personas que llevaron a cabo el muestreo. La implementación de una mala estrategia de recolección de datos en un muestreo por parte de un analista puede llevar a un sesgo.

Mediante un análisis exploratorio de datos podemos identificar anomalías y solicitar al ente encargado de ejecutar el estudio que a) rellene los datos faltantes (solicite a los participantes estos datos) o b) brinde la información faltante si fue problema de transcripción. Como vemos el análisis exploratorio nos permite tomar acciones correctivas con respecto a la validez del diseño del experimento, la fase de recolección de datos.

En este laboratorio vamos a trabajar con el conjunto de datos de Kaggle de Pokemon. Puede acceder al siguiente enlace para ver documentación del significado de las columnas:

https://www.kaggle.com/datasets/rounakbanik/pokemon

Trabajo a realizar

Debe crear un reporte con evidencia de que ejecutó todo el Laboratorio. El reporte es un documento con un formato uniforme y coherente, en el que se incluye texto explicativo del trabajo que se va realizando, los segmentos de código R y los resultados R que se van aportando como parte del laboratorio. El reporte debe poder leerlo y entenderlo una persona que no cuenta con el enunciado del laboratorio.

Para los gráficos en los que se le proveyó de código, debe incluir el grafico en el reporte, debidamente rotulado con la información que presenta.

Para los gráficos donde usted debe crear las instrucciones en R (que inician con “Construya…”) debe presentar tanto el grafico resultante como el código R que utilizó para crearlo.

Cada gráfico y resultado debe estar debidamente identificado. Esto debe hacerse uno por uno, no es válido copiar todo el código R y luego todos los resultados y gráficos. Debe indicar también qué acción está realizando e identificar claramente los resultados que presenta R.

El laboratorio se divide en dos partes: en la primera se realizará un análisis exploratorio de datos, con resultados y gráficos sencillos (pero que deben contar con toda la información pertinente), mientras que en la segunda parte se realizarán algunos gráficos con formato más depurado. Para la primera parte se utilizarán gráficos y facilidades que provee R, mientras que para la segunda parte es necesario cargar algunas bibliotecas.

A. Primera parte

Todos los gráficos que se muestran o solicitan generar deben incluirse en el Reporte de Laboratorio. Para los gráficos para los que no se provee el código (identificados con la palabra ***Construya***), usted debe incluir en el reporte el código R que usted creó.

Además, todos los resultados que se generen a partir de código en R (se despliegan en la consola) deben incluirse en el reporte.

A.1. Información general

Lea el CSV que se le proveyó (pokemon.csv) con la información de los Pokemon:

```
# Al usar choose() usted puede elegir el archivo sin necesidad de definir la ruta
# definir header y encoding le ayudan a read.csv a identificar cabeceras y el formato del texto

df= (read.csv(file.choose(), header=T, encoding = "UTF-8"))
attach(df)

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

Se puede obtener información básica del dataframe con los comandos str() y glimpse().

NO es necesario que copie la salida de estos dos comandos en el reporte!

```
# Informaciónbásica
str(df)
```

El comando summary() nos brinda información de las variables, como cuartiles y datos máximos y mínimos.

NO es necesario que copie la salida de estes comando en el reporte!

Ahora bien, sí es importante que incluya los datos para las variables que más nos interesan en este laboratorio, a saber: **attack, defense, hp, weight_kg y height_m.**

Para cada una de ellas **construya**, ejecute y copie la salida en el reporte.

```
# Por ejemplo:
summary(df$attack)
```

Para variables categóricas se puede obtener información utilizando el comando table().

```
 # Encuentre las cantidades de Pokémon por tipo:

table(df$type1)

table(df$type2)
```

A.2. Histogramas

Los histogramas nos brindan información sobre variables continuas.

```
# Por ejemplo, el siguiente código construye un histograma para la variable hp.

hist(df$hp)

# Se puede hacer un poco más claro agregando título y etiquetas:
hist(df$hp, main="Distribución de variable hp", xlab="HP", col="lightblue", border="black")
```

Construya histogramas similares para las variables attack, defense, weight_kg y height_m.

Pueden ser histogramas **simples**, dado que estamos en la fase de exploración. No son los que eventualmente se incluirán en reportes o artículos.

A.3. Boxplots

Podemos construir boxplots para comparar valores de una variable. Por ejemplo, graficar los valores de attack por tipo de Pokémon.

```
#
boxplot(attack ~ type1, data=df, main="Ataque por Tipo de Pokémon", xlab="Tipo", ylab="Ataque",
las=2, col="lightgreen")
```

Construya boxplots similares para la variable *defense* por *tipo* de Pokémon y para la variable *hp* por *generación* de Pokémon.

A.4. Gráficos de dispersión

Podemos usar estos gráficos para ver cómo se comportan dos variables, si por ejemplo se sospecha que puede haber algún tipo de correlación.

```
# Por ejemplo, el siguiente código construye un gráfico de dispersión entre attack y defense.

plot(df$attack, df$defense)
```

```
# Se puede hacer un poco más claro agregando título y etiquetas:

plot(df$attack, df$defense, main="Ataque vs Defensa", xlab="Ataque", ylab="Defensa", col="darkred")
```

Construya diagramas de dispersión para similares para las variables attack y hp, y para defense y hp.

También es posible generara varios gráficos en uno solo, para tener una vista general de varios, y luego estudiar los más interesantes.

```
# Matrices de dispersión dos a dos para hp, attack, defense y speed.

pairs(df[, c("hp", "attack", "defense", "speed")], main="Matrices de Dispersión")
```

B. Segunda parte

Todos los gráficos que se muestran o solicitan generar deben incluirse en el Reporte de Laboratorio. Para los gráficos para los que no se provee el código (identificados con la palabra ***Construya***), usted debe incluir en el reporte el código R que usted creó.

Instalar los paquetes:

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

Cargar los paquetes:

```
library(dplyr)
library(ggplot2)
library(gridExtra)
library(tidyr)
library(reshape2)
library(RColorBrewer)
library(ggrepel)
```

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

Como se puede ver el data frame es bastante extenso entonces para esta EDA se van a seleccionar algunas columnas para realizar las exploración y visualizaciones.

```
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)
```

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

Un diagrama de densidad visualiza la distribución de datos en un intervalo continuo. Este gráfico es una variación de un histograma que utiliza el suavizado del núcleo para trazar valores, lo que permite distribuciones más uniformes al suavizar el ruido. Los picos de un gráfico de densidad ayudan a mostrar dónde se concentran los valores durante el intervalo.

```
density_hp <- ggplot(data=df, aes(hp)) + geom_density(col="white",fill="pink", alpha=0.8) + ggtitle("Densidad de Hit Points o Vida")
density_speed <- ggplot(data=df, aes(speed)) + geom_density(col="white", fill="darkorchid", alpha=0.8) + ggtitle("Densidad de velocidad")
density_attack <- ggplot(data=df, aes(attack)) + geom_density(col="white", fill="orange", alpha=0.7) + ggtitle("Densidad caracteristicas ofensivas")
density_defense <- ggplot(data=df, aes(defense)) + geom_density(col="white", fill="firebrick", alpha=0.7) + ggtitle("Densidad caracteristicas defensivas")
grid.arrange(density_hp, density_speed, density_attack, density_defense, ncol=2)
```

Construya un nuevo diagrama:

Modifique el grid anterior para incluir dos diagramas de densidad más:

a. El diagrama de densidad basado en la altura (m) que utiliza la variable height_m
b. El diagrama de densidad de peso (kg) que utiliza la variable weight_kg

El nuevo grid contará ahora con 6 diagramas. Asegúrese que el grid se presenta ahora en 3 columnas.

B.2. Diagramas de Barras

Un gráfico de barras es quizás la visualización de datos estadísticos más común utilizada por los medios. Un gráfico de barras desglosa los datos categóricos (lo que R conoce como factores) por grupo y representa estas cantidades usando barras de diferentes longitudes. Vamos a trazar el número de individuos en cada grupo (también llamado frecuencia).

Número de Pokémon basado en su tipo primario (type1) y secundario (type2)

Observamos que los tipos de Pokémon más comunes son Water, Normal, Grass y Bug.

```
ggplot(data=df, aes(type1)) +
 geom_bar(aes(fill=..count..), alpha=0.85) +
 scale_fill_gradient(low = "lightblue", high = "darkblue") +  # Gradiente de colores
 theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
 ggtitle("Distribucion basados en Tipo 1") +
 coord_flip()
```

El tipo de Pokémon secundario más común es Flying. Hay que tomar en cuenta que hay una gran cantidad de Pokémon que no tiene tipo secundario, al menos en este conjunto de datos.

Construya un gráfico de barras similar al anterior, pero para el tipo secundario (type2). Modifique los colores del gráfico anterior y asegúrese de que el ángulo de las etiquetas está en 90 grados.

Número de Pokemon legendarios según su tipo primario (type1)

Podemos observar que hay más de 15 Pokémon **legendarios** del tipo Psíquico. Los tipos de Hada (Fairy), voladores (Fly) y fantasmas (Ghost) tienen la menor cantidad de Pokémon legendarios. Probablemente este sea el caso porque los Pokémon Psíquicos son "misteriosos".

```
df %>%
 filter(is_legendary==1) %>%
 ggplot(aes(type1)) +
 geom_bar(aes(fill= ..count..)) +
 scale_fill_gradient(low = "lightblue", high = "darkblue") +  # Gradiente de colores
 theme(axis.text.x = element_text(angle=90, hjust=0)) +
 ggtitle("Numero de Pokemon Legendarios del Type-1")
```

Construya un gráfico de barras similar al anterior, pero filtrando por los no legendarios. Modifique los colores del gráfico anterior para que los colores estén entre naranja y rojo.

B.3. Gráfico de dispersión – *Scatterplots*

Un diagrama o gráfico de dispersión es una visualización de datos bidimensional que utiliza puntos para representar los valores obtenidos para dos variables (usualmente referidos como X y Y): una trazada a lo largo del eje x, y la otra trazada a lo largo del eje y.

Los diagramas de dispersión son útiles para interpretar tendencias en datos estadísticos y se usan cuando desea mostrar la relación entre dos variables. Los gráficos de dispersión a veces se llaman gráficos de correlación porque, se utilizan para mostrar la correlación entre las variables X y Y.

Características de Defensa vs Ataque del Pokémon

Aquí, los Pokémon legendarios son de color verde, mientras que los Pokémon no legendarios son de color azul.

Para los Pokémon no legendarios, parece existir una relación lineal positiva entre los atributos de ataque y defensa. Lo que significa que los Pokémon que tienen un atributo de ataque más alto, tienden a tener un atributo de defensa más alto también.

Aunque hay algunas anomalías, los Pokémon legendarios, en general, tienen mayores atributos de defensa y ataque. Por ejemplo, tenemos Pokemon como Groudon y Gyarados (que podemos identificar arriba a la derecha) los cuales parecen equilibrados tanto en ataque como en defensa. Por otro, lado tenemos algunos elementos como Magikarp y Abra los cuales poseen características bajas de defensa y ataque.

```
ggplot(data=df, aes(attack, defense)) + geom_point(aes(color=is_legendary), alpha=0.8) +
scale_color_gradient(low="green", high="blue") + ggtitle("Contraste  Defensa vs Ataque") +
  geom_label_repel(data=subset(df,attack > 150 | defense > 160 | attack < 25 | defense < 25),
aes(label=name),
                box.padding = 0.35, point.padding = 0.5, size = 3,
                segment.color = 'grey50'))
```

Otros Scatterplots

Como veremos más adelante en este curso cuando veamos regresión, el diagrama de dispersión o scatterplot, nos ayuda a encontrar relaciones entre X y Y. Estas relaciones, en su mayoría son lineales. Asimismo, estas relaciones nos permiten identificar las asociaciones entre las variables, pero también permiten identificar si en nuestro modelo existen variables ocultas (*lurking*). Algo que debemos recordar siempre es que cualquier asociación que identifiquemos será eso, una asociación o correlación, pero no una causalidad directa.

Pero, regresando a nuestro análisis de Pokémon, vamos a enfocarnos en las siguientes variables:
velocidad, altura, peso y HP (*hit points* o vida) vs atributo de ataque. Esto porque queremos ver si existe algunas relaciones entre estos atributos. La idea es ver si quizás, a mayor velocidad menos HP, o a mayor peso, más HP.

De hecho, y a manera de avance (*spoiler alert!*), a medida que aumenta el ataque en los Pokemon, los atributos de Velocidad, Altura, Peso y HP del Pokémon aumentan también. Así pues, vemos que existe una relación positiva débil entre:

a. la velocidad y los atributos de ataque.
b. una relación lineal positiva más fuerte entre altura, peso y HP con el atributo de ataque.

En el diagrama de dispersión, con el siguiente código (ver abajo), se hace un etiquetado de los Pokémon legendarios.
Primero vamos a crear el gráfico de dispersión entre attack y speed para los Pokémon legendarios. Etiquetaremos aquellos con attack > 170 ó speed > 150.

```
speed_attack_legendary <- ggplot(na.omit(df), aes(attack, speed)) +
  geom_point(aes(color=is_legendary)) +
  geom_label_repel(data=subset(df, (attack > 170 | speed > 150) & is_legendary == 1), aes(label=name),
            box.padding = 0.35, point.padding = 0.5, segment.color = 'grey50') +
  geom_smooth(method = "lm")

print(speed_attack_legendary)
```

Ahora **construye** un gráfico equivalente para attack vrs weight_kg en los Pokémon legendarios. Etiqueta los Pokémon con attack > 150 ó weight_kg > 650.

También **construye** un gráfico equivalente para attack vrs height_m en los Pokémon legendarios. Etiqueta los Pokémon con attack > 150 ó height_m > 7.5.

Finalmente, **construye** un gráfico equivalente para attack vrs hp en los Pokémon legendarios. Etiqueta los Pokémon con attack > 170 ó hp > 190.

B.4. Diagramas de Caja o Boxplots

Un diagrama de caja o box plot es una forma de mostrar la dispersión y los centros de un conjunto de datos.

Diagrama de caja de atributos de Pokémon
En los boxplots se puede observar que la mediana de todas las características de Pokémon legendarios se encuentran por encima de aquellas de sus contrapartes no legendarias. Por otra parte, el boxplot nos permite ver que para el atributo HP podemos observar que existen muchos más valores atípicos para los Pokemon no legendarios cuando lo comparamos con los otros atributos. Finalmente, al comparar los 2 grupos de análisis (legendarios y no legendarios) podemos ver que:

Hay más valores atípicos (en cuanto a atributos) en Pokémon no legendarios en comparación con los Pokémon legendarios.

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

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))

bp_leg <- ggplot(data = box_plot_attr_leg_long, aes(attribute, value)) + geom_boxplot(fill="green4") + ggtitle("Pokemon Legendario")

bp_nor <- ggplot(data = box_plot_attr_nor_long, aes(attribute, value)) + geom_boxplot(fill = "yellow2") + ggtitle("Pokemon No Legendario")

grid.arrange(bp_leg, bp_nor,ncol=2)
```

B.5. Mapas de calor

Un mapa de calor se puede definir como tablas con colores en lugar de números, donde una escala colorimétrica representa un rango de valores continuos (escala de grises en lugar de valores de 0 a 1). Con base en esto podemos entender entonces que por ejemplo el valor más bajo se establece en azul oscuro, el valor más alto en rojo y los valores de rango medio en blanco, con una transición (o gradiente) correspondiente entre estos extremos.

Los mapas de calor son muy usados para visualizar cantidades de datos multidimensionales. Esta característica los hace parte del conjunto de herramientas que permiten explorar e identificar tendencias relevantes en un conjunto de datos. Por ejemplo, a través de los mapas de calor podemos identificar grupos de filas con valores similares, ya que se muestran como áreas de color similar. Esto permite identificar variables que poseen una relación.

A continuación, veremos cómo los atributos de varios tipos de Pokémon se comparan entre sí. Para esto, calcularemos la mediana de estos atributos y trazaremos un mapa de calor para Pokémon legendarios y no legendarios.

Mapa de calor de tipo primario vs atributo - Pokémon legendarios

En este momento utilizaremos mapas de calor para analizar los Pokemon tipo hielo. Este análisis nos permite ver que los Pokemon legendarios de tipo hielo o Ice tienen un valor muy alto de defensa especial (sp_defense), pero un valor bajo de velocidad o Speed. Asimismo, podemos apreciar que los Pokémon legendarios terrestres tienen un atributo de ataque muy alto, y aquellos Pokémon legendarios de tipo bug o Insecto tienen un nivel muy bajo de sp_defense o defensa especial.

```
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)
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("Pokémon Legendaio: Type1 - Atributo") + scale_fill_gradientn(colours = hm.palette(100)) + theme(axis.text.x = element_text(angle=90, hjust=0)) + coord_equal()
```

Mapa de calor de tipo primario vs atributo - Pokémon no legendarios

Construya un mapa de calor equivalente, pero para Pokémon no legendarios.

En este escenario podemos ver que:

a. Los Pokémon tipo acero tienen un alto valor de defensa, pero muy baja velocidad.
b. Los Pokémon de tipo lucha (fighting) tienen un alto valor de ataque, pero un valor muy bajo de sp_attack.
c. Los Pokémon terrestres, de hielo y normales tienen niveles de HP similares
d. Los Pokémon de tipo veneno, psíquico y de roca tienen niveles de HP similar.

B.6. Matriz de Correlación

Una matriz de correlación es una tabla que muestra los coeficientes de correlación (coeficiente de correlación de Pearson) entre conjuntos de variables que le permiten a las personas analistas ver qué pares o parejas de atributos tienen la correlación más alta. El coeficiente de correlación puede variar en valor de −1 a +1. Cuanto mayor sea el valor absoluto del coeficiente, más fuerte será la relación entre las variables.

En los experimentos nos interesa identificar las relaciones entre las variables dependientes e independientes. Esto es muy importante porque nos facilita saber cuáles interacciones o relaciones lineales pueden existir y que vale la pena analizar con mayor profundidad. Adicionalmente, 2 variables pueden no tener

correlaciones fuertes con Y, pero si hacemos ingeniería de variables la nueva variable puede tener una correlación alta. Esto es algo muy importante y justifica la necesidad de comprender el contexto del experimento.

Para la correlación de Pearson, un valor absoluto de 1 indica una relación lineal perfecta.

Mapa de calor de correlación para los atributos de Pokemon legendarios

Interesantemente, y como vimos antes, la matriz de correlación existe en un rango de -1 a +1. No es entonces de extrañarse que podamos convertirla en un mapa de calor. Esto es algo que realmente ocurre en muchas ocasiones, y facilita mucho el análisis de datos. Es más fácil ver las relaciones en términos de colores que en una tabla. El valor más bajo en el mapa de calor se establece en azul oscuro, el valor más alto en blanco y los valores de rango medio en verde, con una transición (o gradiente) correspondiente entre estos extremos. Esta es una decisión arbitraria y cosmética. Cada uno decide como usarla, y es claro que pueden usar otros esquemas, como escala de grises, café-anaranjado, etc.

Es habitual y muy recomendable evitar la escala rojo-verde más tradicional. Durante mucho tiempo fue la escala de color por defecto, pero en la actualidad, se ha dejado de utilizar. Esta es una consideración de accesibilidad para individuos que padecen daltonismo (color blindness). Es muy importante que cuando construyan sus visualizaciones tomen en consideración esto. Un estudio identificó que es común encontrar alrededor del 10% de las personas en un auditorio pueden padecer algún tipo de discapacidad visual, como la falta de percepción de color. Estos incluyen: deuteranopia, protanopia y tritanopia. Consideren esto, para presentación de datos o para una entrevista de trabajo, o cualquier escenario profesional. Eviten ser esa persona – el que no hizo el gráfico que el jefe pudo ver porque no mostró empatía.

Ahora procedemos a hacer una serie de conjeturas a partir del conocimiento que tenemos de nuestros contextos, por ejemplo:

a. Es posible que exista una relación entre HP y defensa: cuanta más HP tendrá más defensa y así soporta más daño.
b. Es cierto que existe una relación entre ataque y velocidad, es decir mientras más veloz un Pokémon, más aumenta su atributo de ataque.
c. Existe una relación entre velocidad y defensa, entonces a mayor defensa se hacen más lentos (relación negativa)

Entonces esto lo podemos verificar con el heatmap:

```
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))

row.names(hmap_attr_leg) <- hmap_attr_leg$type1
hmap_attr_leg$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(4, 'RdBu')), 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()
```

Análisis de la efectividad de los Pokémon basados en su tipo de ataque primario.

La idea de esto es determinar si podemos determinar cuál tipo de Pokémon es más efectivo contra otro a la hora de batallar. Esto tiene mucho sentido porque si queremos tener un conjunto balanceado y capaz de vencer a nuestros oponentes debemos buscar cual grupo de Pokémon posee la mayor efectividad en combate.

```
#  Hay que volver a cargar los datos, dado que df se modificó

df= (read.csv(file.choose(), header=T, encoding = "UTF-8"))
attach(df)

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

Para esto seleccionamos un sub-conjunto de los datos:

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

Se deben encontrar la mediana de todas las columnas against_type

```
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))
```

Construimos el heatmap

df_fight_against_long <- melt(df_fight_against_summ)

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")

Como mencionamos anteriormente, el valor más bajo en el mapa de calor fue definido como azul oscuro, y el valor más alto en rojo. Los valores de rango intermedio son blancos, con una transición (o gradiente) correspondiente entre estos extremos.

Instrucciones de la entrega:

El trabajo se entrega en grupos de 2 personas.

Las entregas se realizarán en formato PDF. La letra debe ser Times New Roman, Arial o Cambria. Los tamaños permitidos son 10-12.

Debe incluir una portada de página completa en la que incluya el nombre del curso, del laboratorio, así como los nombres completos de los estudiantes. Esta página será utilizada por el profesor para apuntar la nota y también para incluir comentarios en caso de tener que incluir generales del informe.

El reporte se debe entregar en Mediación Virtual, en un archivo .pdf que pueda ser leído en programas comerciales de uso habitual. Debe verificar que el .pdf que subió a Mediación Virtual contiene los ejercicios resueltos y que el archivo puede abrirse correctamente. En caso de problemas con el archivo .pdf (no abre correctamente, está corrupto, etc.) se considerará que no entregó la tarea.

Las imágenes deben ser visibles y contar con todos los elementos necesarios para su entendimiendo, por ejemplo, que las escalas estén completas: se recomienda ampliar la imagen en RStudio antes de copiarla en el informe.

En los casos que deba incluir código R en el reporte, cópielo como un pantallazo (screenshot) de forma que no se pierda el formato original.

Las entregas tardías se penalizarán con un 10% de la nota luego de vencida la fecha y hora de entrega, más un 10% adicional por cada hora de retraso.