

Juntar R y Python con RStudio

Ramon Ceballos

11/2/2021

A. Pokemon (Py -> R)

1. Generalidades para el uso inicial de Python en R

Para poder trabajar con R y Python se emplea la librería “reticulate”.

Para ver que versión de Python se está utilizando, pon en la consola “Sys.which(“python”)” y te indicará el directorio de uso. Si quisiera emplear Python de anaconda u otra versión tendrá que especificar en la primer chunk del RMarkdown la expresión “use_python”(“directorio/ananconda3/bin/python”).

```
#Para ver si funciona  
import pandas as pd
```

Ahora vamos a llamar a Python desde R para lo cuál empleamos “py()” de reticulate.

2. Carga de dataset de Pokemon y modelado del dataset (selección) en Python

Cargamos el csv de pokemon con pd.read_csv de pandas. Exploramos si se ha cargado bien el dataset.

Vamos a filtrar el dataset con la siguiente chunk.

Vamos a filtrar por los pokemon de la primera generación y luego con las columnas Type 1, Type 2 y Speed. Limpiamos los Na con la función de python dropna().

```
pokemon = pd.read_csv("../../data/Pokemon.csv")  
print(pokemon.shape)
```

```
## (800, 12)
```

```
print(pokemon.head (4))
```

```
##           Name Type 1  Type 2  Total  HP  Attack  Defense  Sp_Atk  \  
## 0      Bulbasaur  Grass  Poison   318  45     49      49      65  
## 1       Ivysaur  Grass  Poison   405  60     62      63      80  
## 2       Venusaur  Grass  Poison   525  80     82      83     100  
## 3 VenusaurMega Venusaur  Grass  Poison   625  80    100     123     122  
##  
##      Sp_Def  Speed  Generation  Legendary  
## 0         65     45           1       False
```

```
## 1      80      60      1      False
## 2     100      80      1      False
## 3     120      80      1      False
```

```
pokemon = pokemon[pokemon['Generation'] == 1]
print(pokemon.shape)
```

```
## (166, 12)
```

```
pokemon = pokemon[["Type 1", "Type 2", "Speed"]]
print(pokemon.shape)
```

```
## (166, 3)
```

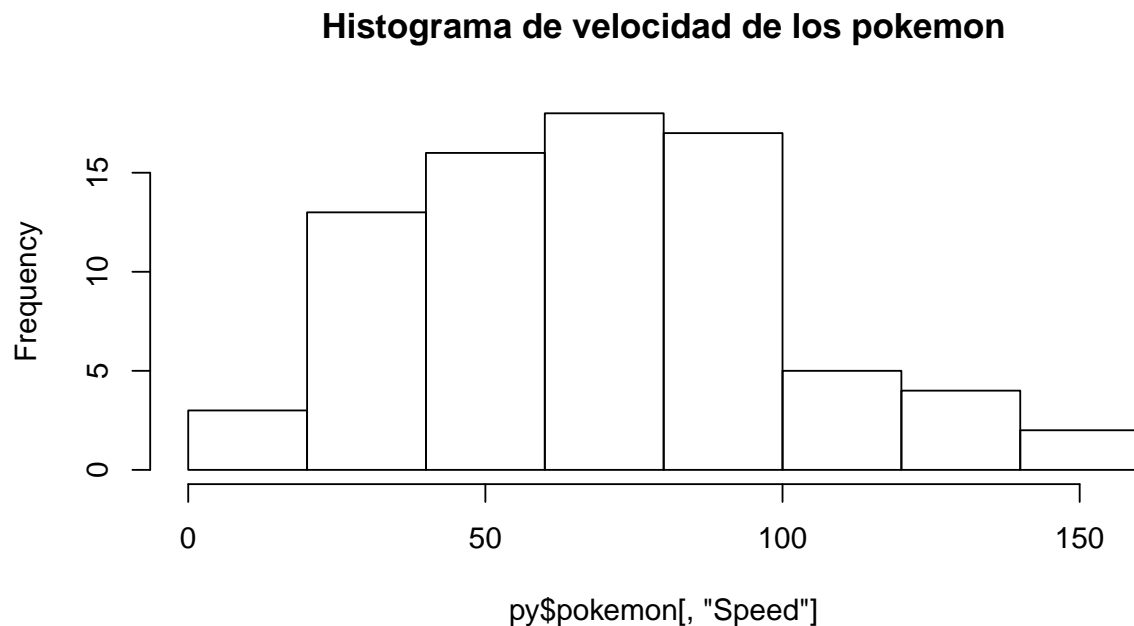
```
pokemon = pokemon.dropna()
print(pokemon.shape)
```

```
## (78, 3)
```

3. Representar gráficas de los datos de Python en R

Accedemos desde R a la variable creada en python empleando la expresión “py\$objeto_python”. Representamos con un histograma de R la columna de “Speed” de pokemon.

```
hist(py$pokemon[, "Speed"],
     breaks=9,
     main="Histograma de velocidad de los pokemon")
```



B. Pokemon (R -> Py)

1. Carga de dataset de Pokemon y modelado del dataset (selección) en R

Ahora vamos a importar datos en R y representaremos dichos datos a través de Python (lo contrario al apartado A).

```
pokemon2 <- read.csv("../../data/Pokemon.csv", header=TRUE)

#Cargamos libreria tidyverse
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.3      v purrr  0.3.4
## v tibble  3.0.4      v dplyr  1.0.2
## v tidyr   1.1.2      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

#Filtramos el dataset
pokemon2 <- pokemon2 %>%
  #Filtro
  filter(Generation==1) %>%
  #Selecciono las columnas
  select(Type.1, Type.2, Speed) %>%
  #Quito los NA
  na.omit()

#Ver estadísticos de cada variable
summary(pokemon2)

##      Type.1      Type.2      Speed
## Water  :31      :88   Min.   : 15.00
## Normal :24   Flying :23   1st Qu.: 50.00
## Bug    :14   Poison :22   Median : 70.00
## Fire   :14   Psychic: 7    Mean    : 72.58
## Poison :14   Ground : 6    3rd Qu.: 92.25
## Grass  :13   Water  : 4    Max.    :150.00
## (Other):56   (Other):16
```

2. Representar gráficas de los datos de R en Python

Transmitir datos de R a Python con la expresión “r.dataset_R”.

```
print(r.pokemon2.head())
```

##	Type.1	Type.2	Speed
## 0	Grass	Poison	45
## 1	Grass	Poison	60
## 2	Grass	Poison	80
## 3	Grass	Poison	80
## 4	Fire		65