Librería reticulate

J Ramajo

Aplicación básica de *reticulate*

A través de este ejemplo se verá fácilmente cómo interactúan R y Python a través de la librería `reticulate`. Se pueden encontrar todos los detalles en la página web siguiente: https://rstudio.github.io/reticulate/

Configuración de RStudio para permitir la interacción R-Python

Simplemente hay que leer la librería `reticulate` para activar ambos lenguajes.

```
library(reticulate)
use_python("/home/econometria/.local/share/r-miniconda/bin/python")
```

Desde Python hacia R

Se usa Python para leer y manipular los datos

```
import pandas
flights = pandas.read_csv("data/flights.csv")
flights = flights[flights['dest'] == "ORD"]
flights = flights[['carrier', 'dep_delay', 'arr_delay']]
flights = flights.dropna()
flights
```

```
      carrier  dep_delay  arr_delay
5          UA       -4.0       12.0
9          AA       -2.0        8.0
25         MQ        8.0       32.0
38         AA       -1.0       14.0
```

```
57          AA       -4.0        4.0
...        ...        ...        ...
336645      AA      -12.0      -37.0
336669      UA       -7.0      -13.0
336675      MQ       -7.0      -11.0
336696      B6       -5.0      -23.0
336709      AA      -13.0      -38.0

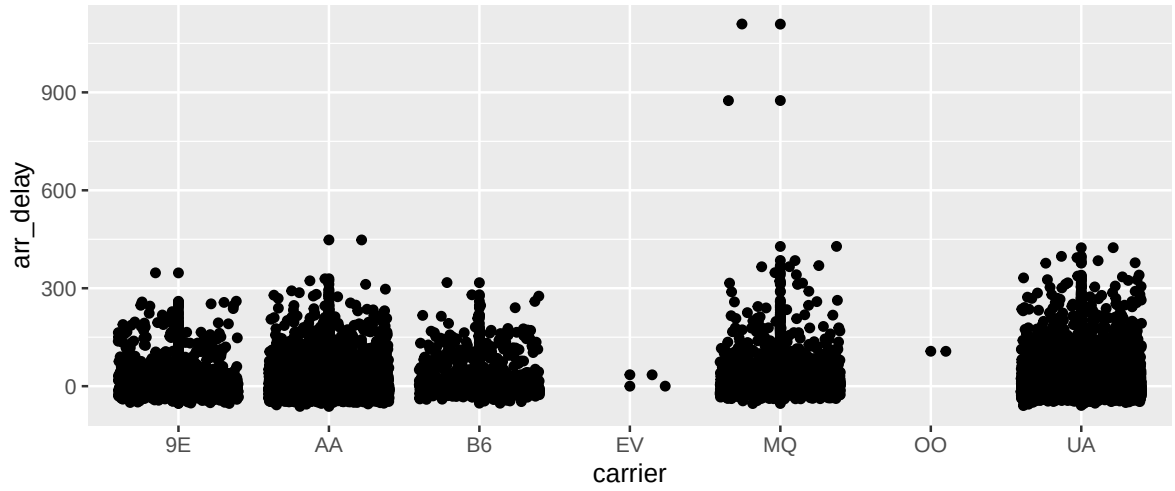
[16566 rows x 3 columns]
```

```
flights.describe()
```

```
          dep_delay     arr_delay
count  16566.000000  16566.000000
mean      13.432814      5.876615
std       42.708837     48.008944
min      -20.000000    -62.000000
25%       -5.000000    -20.000000
50%       -2.000000     -8.000000
75%       11.000000     13.000000
max     1126.000000   1109.000000
```

Ahora se usa R para visualizar los datos guardados en Python (en formato Pandas DataFrame)

```
library(ggplot2)
ggplot(py$flights, aes(carrier, arr_delay)) + geom_point() + geom_jitter()
```



Desde R hacia Python

Se usa R para leer y manipular los datos

```
library(tidyverse)
flights <- read_csv("data/flights.csv") %>%
  filter(dest == "ORD") %>%
  select(carrier, dep_delay, arr_delay) %>%
  na.omit()
head(flights)
```

```
# A tibble: 6 x 3
  carrier dep_delay arr_delay
  <chr>       <dbl>     <dbl>
1 UA             -4        12
2 AA             -2         8
3 MQ              8        32
4 AA             -1        14
5 AA             -4         4
6 UA              9        20
```

```
tail(flights)
```

```
# A tibble: 6 x 3
  carrier dep_delay arr_delay
  <chr>       <dbl>     <dbl>
1 9E              4        -2
2 AA            -12       -37
3 UA             -7       -13
4 MQ             -7       -11
5 B6             -5       -23
6 AA            -13       -38
```

```
summary(flights)
```

```
   carrier             dep_delay          arr_delay       
 Length:16566       Min.   : -20.00   Min.   : -62.000  
 Class :character   1st Qu.:  -5.00   1st Qu.: -20.000  
 Mode  :character   Median :  -2.00   Median :  -8.000  
                    Mean   :  13.43   Mean   :   5.877  
                    3rd Qu.:  11.00   3rd Qu.:  13.000  
                    Max.   :1126.00   Max.   :1109.000  
```

Ahora se usa Python para visualizar los datos guardados en R (en formato R DataFrame)

```
from plotnine import *
(ggplot(r.flights, aes(x="carrier", y="arr_delay")) + geom_point() + geom_jitter())
```

```
<plotnine.ggplot.ggplot object at 0x7f04ea1a67e0>
```