

# Control 1, respuestas

## Web Scraping y acceso a datos desde la web

Cristián Ayala

Ponderación: 20% de la nota final del curso

### 1 Tareas:

#### 1.1 Identificar selectores

- 1) Desde la página web <https://www.scrapethissite.com/pages/simple/> a capturar, identificar el nombre de la clase de css para:

La clase de cada uno de estos elementos son:

- Nombre de país: `country-name`
- Nombre de capital: `country-capital`
- Población: `country-population`
- Superficie: `country-area`

#### 1.2 Captura de datos

- 2) Capturar la información de todos los países para pasarla a una `data.frame`.

```
url <- 'https://www.scrapethissite.com/pages/simple/'
html <- read_html(url)

# Listado con los 4 selectores de interés.
l_css_selectors <- c('pais'      = '.country-name',
                    'capital'   = '.country-capital',
                    'poblacion' = '.country-population',
                    'superficie' = '.country-area')

# Extraigo los datos correspondientes para cada selector.
l_paises <- map(l_css_selectors,
               \(css_sel) html_elements(html, css_sel) |>
               html_text() |>
```

```

str_squish()

df_paises <- as_tibble(l_paises) # Convierto la lista a tibble.

# Corrijo el tipo de variable de las variables capturadas.
df_paises <- readr::type_convert(df_paises)

head(df_paises)

```

```

# A tibble: 6 x 4
  pais                capital          poblacion superficie
<chr>                <chr>          <dbl>         <dbl>
1 Andorra             Andorra la Vella  84000          468
2 United Arab Emirates Abu Dhabi        4975593        82880
3 Afghanistan        Kabul            29121286       647500
4 Antigua and Barbuda St. John's       86754          443
5 Anguilla            The Valley       13254          102
6 Albania             Tirana           2986952        28748

```

### 1.3 Listado de países ordenados según población

3) Listar los nombres de países desde el con menor población al con mayor población.

```

# Usando dplyr
df_paises |>
  arrange(poblacion) |>
  pull(pais) |>
  head(10)

```

```

[1] "Antarctica"
[2] "Bouvet Island"
[3] "Heard Island and McDonald Islands"
[4] "U.S. Minor Outlying Islands"
[5] "South Georgia and the South Sandwich Islands"
[6] "Pitcairn Islands"
[7] "French Southern Territories"
[8] "Cocos [Keeling] Islands"
[9] "Vatican City"
[10] "Tokelau"

```

```

# Usando R base
df_paises[order(df_paises$poblacion), 'pais', drop = TRUE] |>
  head(10)

```

```
[1] "Antarctica"
[2] "Bouvet Island"
[3] "Heard Island and McDonald Islands"
[4] "U.S. Minor Outlying Islands"
[5] "South Georgia and the South Sandwich Islands"
[6] "Pitcairn Islands"
[7] "French Southern Territories"
[8] "Cocos [Keeling] Islands"
[9] "Vatican City"
[10] "Tokelau"
```

## 1.4 Agregar variable continente

- 4) Agregar a la base de datos el dato `continente` para cada país según la base de datos `countryName_continent.csv` disponible en el repositorio.

```
df_continente <- read_csv('c_1_files/countryName_continent.csv',
                          col_types = cols(col_character())) # Asigno las variables como chr

sum(is.na(df_continente$continent))
```

```
[1] 41
```

41 países no tienen continente asignado en la base de continentes `df_continente`.

```
df_paises <- left_join(df_paises,
                      df_continente,
                      by = c('pais' = 'countryName'))

table(df_paises$continent, useNA = 'ifany')
```

```
AF  AN  AS  EU  OC  SA <NA>
53   3  49  51  26  14   54
```

54 países en la base de población y superficie quedaron sin un continente asignado. Les daré un valor explícito: *sin dato*.

```
df_paises$continent <- fct_na_value_to_level(df_paises$continent, 'sin dato')
```

## 1.5 Graficar relación entre superficie y población

- 5) Graficar la relación entre superficie (eje x) y población (eje y) coloreando cada país según el continente al que pertenezca según la base de datos continente.

El gráfico se muestra en Figura 1.

```
ggplot(df_paises,
       aes(x = superficie,
           y = poblacion,
           colour = continent)) +
ggforce::geom_mark_ellipse(aes(fill = continent),
                           alpha = 0.1,
                           linetype = 0,
                           show.legend = FALSE) +
geom_point() +
scale_x_log10('log Superficie (miles de km<sup>2</sup>)',
              labels = ~scales::number(., scale = 0.001)) +
scale_y_log10('log Población (miles)',
              labels = ~scales::number(., scale = 0.001)) +
labs(title = 'Relacion entre superficie y población',
     caption = 'Web Scraping y acceso a datos desde la web',
     colour = 'Continentes') +
theme_minimal() +
theme(axis.title.x = ggtext::element_markdown())
```

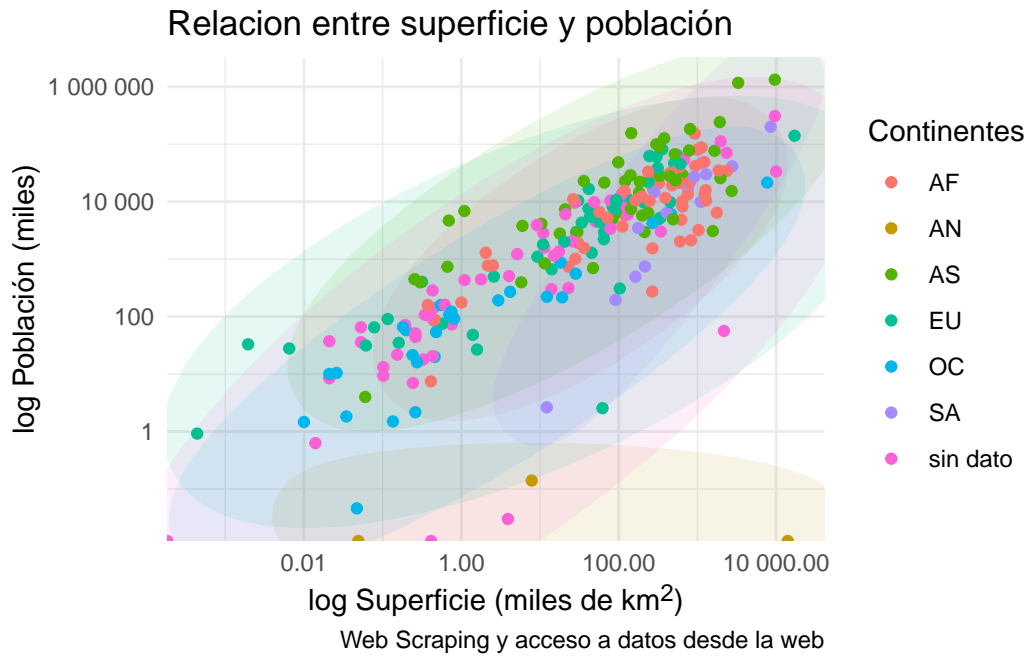


Figura 1: Relación entre superficie y población