Control 2, respuestas

Web Scraping y acceso a datos desde la web

Cristián Ayala

Ponderación 20% de la nota final del curso

Formato Desarrollar esta tarea con Quarto o Rmarkdown generando un `.pdf`, agregando comentarios cuando sea necesario.

1 Objetivo:

Interesa indagar sobre el cine chileno. Queremos saber la evolución del número de películas chilenas estrenadas por año y su calificación según la nota dada por IMDb.

Para ello usaremos el sitio web IMDb para filtrar películas chilenas realizadas en Chile. En total son **315**[1] según se muestra en esta búsqueda:

https://www.imdb.com/search/title/?title_type=feature&countries=cl&locations=chile

2 Tareas:

2.1 Captura de datos

1) Desde esa página web capturar los siguientes datos de esas 315 películas:

Los objetos están dentro de `<div>` de nombre `#main` con clase `lister-item-content`.

- Título: `.lister-item-header a`
- Año de estreno: `.lister-item-header .lister-item-year`
- Puntaje IMDb: `.ratings-imdb-rating strong`
- Géneros: `.genre`

Cada página muestra 50 películas y utiliza el parámetro `start=NUMERO` para mostrar desde la película número `NUMERO` las 50 siguientes.

```
url_1 <- 'https://www.imdb.com/search/title/?title_type=feature&countries=cl&locations=chile'
url_parse_1 <- parse_url(url_1)
```

[1] Número de películas al momento de diseñar este control.

```
l_pelicula_1_html <- read_html(build_url(url_parse_1))
```

¿Cuántas son las películas totales que están presente en la búsqueda?

```
n_peliculas <- l_pelicula_1_html |>
  html_elements('.nav div.desc') |>
  html_text2()

n_peliculas <- n_peliculas |>
  str_extract(' of (\\d+)', group = 1) |>
  as.integer()

n_peliculas
```

```
[1] 315
```

Creamos ahora los intervalos de búsqueda

```
n_pel_por_pagina <- 50

intervalos <- seq(1,
                  ceiling(n_peliculas/n_pel_por_pagina) * n_pel_por_pagina,
                  n_pel_por_pagina)

intervalos
```

```
[1]   1  51 101 151 201 251 301
```

Saco la página 1 porque ya la tengo capturada

```
intervalos <- intervalos[-1]
```

Construyo los links para cada una de las páginas de búsqueda.

```
querys <- map(intervalos,
              \(x) c(url_parse_1$query, 'start' = x))

f_urls <- function(.query){
  url_parse_1['query'] <- list(.query)

  url_parse_1 |>
    build_url()
}

l_urls <- map(querys, f_urls)
```

```
  l_urls
```

```
[[1]]
[1] "https://www.imdb.com/search/title/?title_type=feature&countries=cl&locations=chile&start=51"

[[2]]
[1] "https://www.imdb.com/search/title/?title_type=feature&countries=cl&locations=chile&start=101"

[[3]]
[1] "https://www.imdb.com/search/title/?title_type=feature&countries=cl&locations=chile&start=151"

[[4]]
[1] "https://www.imdb.com/search/title/?title_type=feature&countries=cl&locations=chile&start=201"

[[5]]
[1] "https://www.imdb.com/search/title/?title_type=feature&countries=cl&locations=chile&start=251"

[[6]]
[1] "https://www.imdb.com/search/title/?title_type=feature&countries=cl&locations=chile&start=301"
```

Lectura de cada hoja

```
  l_peliculas_html <- map(l_urls, read_html)
```

Agrego la primera hoja ya capturada. Tengo un total de 6 hojas

```
  l_peliculas_html <- append(list(l_pelicula_1_html), l_peliculas_html)

  length(l_peliculas_html)
```

```
[1] 7
```

Selección de datos de interés

```
  # Obtener lista de nodo de películas
  selectores <- c(index  = '.lister-item-index',
                  titulo = '.lister-item-header a',
                  anio   = '.lister-item-header .lister-item-year',
                  rating = '.ratings-imdb-rating strong',
                  genero = '.genre')

  f_capturar_elementos <- function(.html, .selector, .names_sel){

    links <- NULL # Objeto solo para links en el caso de estar capturando el título

    html <- .html |>
```

```
    html_elements('#main .lister-item-content')

  # Captura general del elemento de interés.
  data <- html |>
    html_element(.selector) |>
    html_text() |>
    str_squish()

  # Captura de link a la película solo si estoy viendo elemento nominado título
  if (.names_sel == 'titulo'){
    links <- html |>
      html_element(.selector) |>
      html_attr('href')
  }

  # Devuelvo los datos capturados: un vector con texto y links.
  setNames(list(data, links),
           nm = c(.names_sel, 'link'))
}

# Itero todos los selectores en todas las páginas de películas que capturamos

l_peliculas <- map(l_peliculas_html,
                   \(l_pel){
                     map2(selectores, names(selectores),
                          \(selector, names_sel){
                            f_capturar_elementos(l_pel, selector, names_sel)
                          }
                     )
                   }
)

datos_a_df <- function(.datos){
  list_flatten(.datos) |> # quita un nivel
    discard(is.null) |>  # elimina variables vecias
    as_tibble() # transforma listas a df
}

# Lista de tibbles de cada página
l_peliculas_df <- l_peliculas |>
  map(datos_a_df)

# Creación de tibble única
df_peliculas <- l_peliculas_df |>
  list_rbind()

df_peliculas |> dim()
```

```
[1] 315   6
```

Mejora de nombres de las columnas de `df_peliculas`.

```
names(df_peliculas) <- str_replace(names(df_peliculas), "^(.*)_\\1$", "\\1")

df_peliculas |> names()
```

```
[1] "index"       "titulo"      "titulo_link" "anio"        "rating"
[6] "genero"
```

2) Guardar esa información en un data.frame

```
head(df_peliculas)
```

```
# A tibble: 6 x 6
  index titulo                       titulo_link             anio  rating genero
  <chr> <chr>                        <chr>                   <chr> <chr>  <chr>
1 1.    Knock Knock: Seducción Fatal /title/tt3605418/?ref_~ (I) ~ 4.9    Crime~
2 2.    Caníbales                    /title/tt2403021/?ref_~ (201~ 5.3    Adven~
3 3.    Trauma                       /title/tt6705640/?ref_~ (II)~ 4.9    Actio~
4 4.    Los 33                       /title/tt2006295/?ref_~ (201~ 6.9    Biogr~
5 5.    Diarios de motocicleta       /title/tt0318462/?ref_~ (200~ 7.7    Adven~
6 6.    El Príncipe                  /title/tt7945236/?ref_~ (201~ 6.4    Drama
```

Corregiremos alguna de las variables extraídas para el análisis siguiente.

```
df_peliculas <- df_peliculas |>
  mutate(
    # Remover punto final en index
    index = str_remove(index, '\\.'),
    # Extraer los números de la variable anio
    anio = str_extract(anio, '\\d+'),
    # Separar un solo string de género en distintas palabras
    genero = str_split(genero, ', ?')
    )

df_peliculas <- df_peliculas |>
  mutate(across(c(index), as.integer),
         across(c(rating), as.double),
         anio = as.Date(paste0(anio, '-01-01', '%Y-%M-$d')))

head(df_peliculas)
```

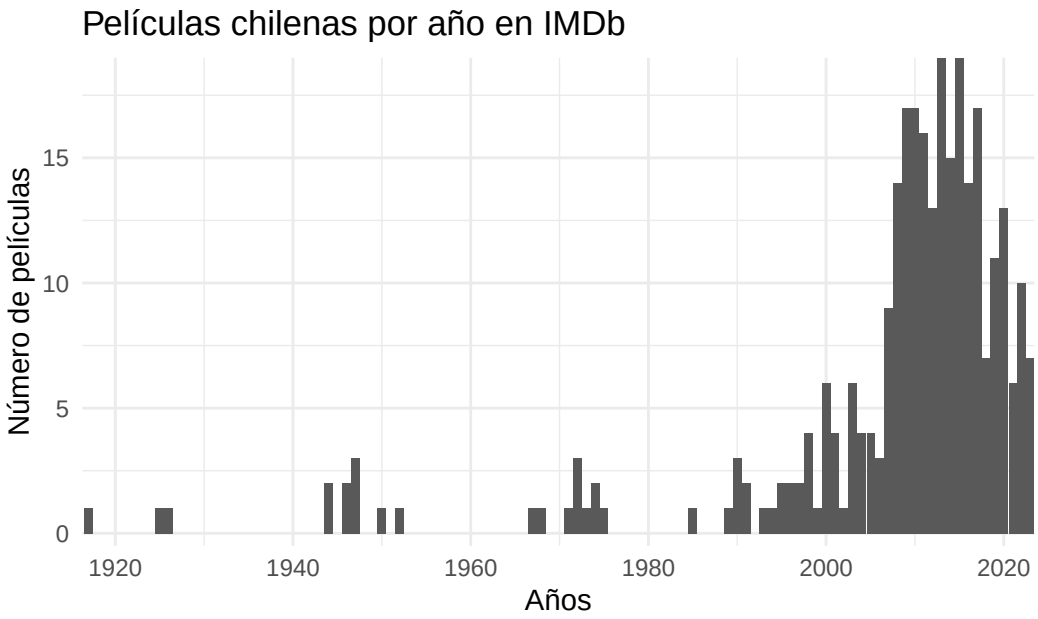
```
# A tibble: 6 x 6
  index titulo                      titulo_link        anio       rating genero
  <int> <chr>                       <chr>              <date>      <dbl> <list>
1     1 Knock Knock: Seducción Fatal /title/tt3605418/~ 2015-01-01    4.9 <chr>
2     2 Caníbales                   /title/tt2403021/~ 2013-01-01    5.3 <chr>
3     3 Trauma                      /title/tt6705640/~ 2017-01-01    4.9 <chr>
4     4 Los 33                      /title/tt2006295/~ 2015-01-01    6.9 <chr>
5     5 Diarios de motocicleta      /title/tt0318462/~ 2004-01-01    7.7 <chr>
6     6 El Príncipe                 /title/tt7945236/~ 2019-01-01    6.4 <chr>
```

2.2 Análisis

3) Graficar la evolución del el *número de películas* (eje y) estrenadas por *año* (eje x).

```r
df_peliculas_anio <- df_peliculas |>
  count(anio, name = 'n_peliculas')

df_peliculas_anio |>
  ggplot(aes(x = anio, y = n_peliculas)) +
  geom_col() +
  scale_x_date('Años', expand = expansion(add = c(100, 0))) +
  scale_y_continuous(expand = expansion(add = c(0.5, 0))) +
  labs(title = 'Películas chilenas por año en IMDb',
       caption = 'Fuente: IMDb.com. Web Scraping y acceso a datos desde la web',
       y = 'Número de películas') +
  theme_minimal()
```



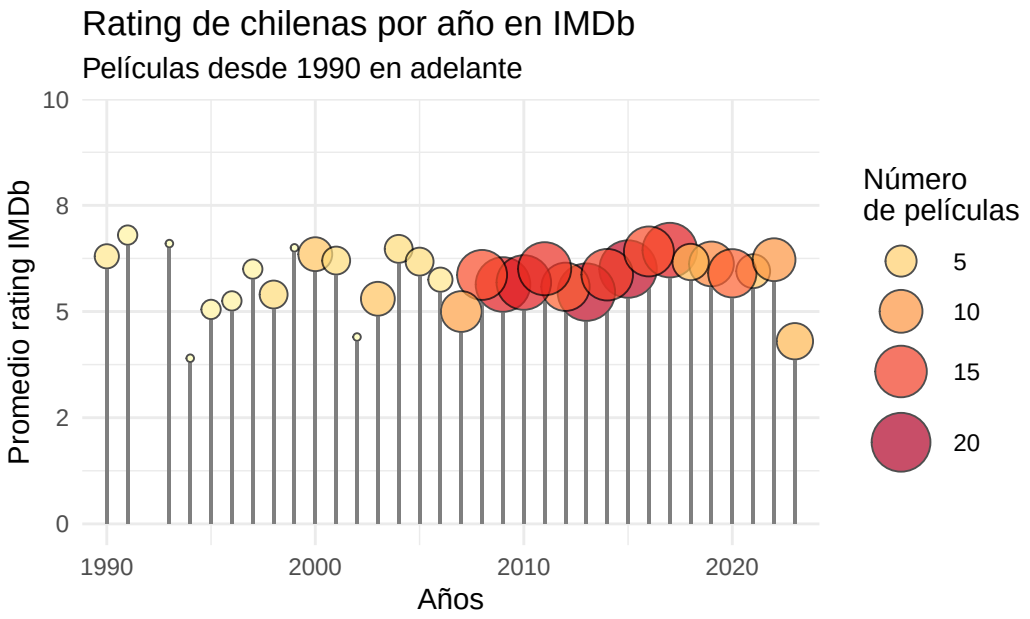
Fuente: IMDb.com. Web Scraping y acceso a datos desde la web

5) Graficar la evolución del el *rating IMDb* promedio (eje y) *estrenadas desde 1990* a la fecha (eje x).

```
df_peliculas_rank <- df_peliculas |>
  filter(anio >= as.Date("1990-01-01")) |>
  summarise(n_peliculas = n(),
            rating = mean(rating, na.rm = TRUE),
            .by = anio)


df_peliculas_rank |>
  ggplot(aes(x = anio, y = rating,
             fill = n_peliculas,
             size = n_peliculas)) +
  geom_col(width = 60, fill = 'gray50',
           show.legend = F) +
  geom_point(colour = 'white') +
  geom_point(shape = 21,
             alpha = .7) +
  scale_x_date('Años',
               expand = expansion(add = c(400, 400))) +
  scale_y_continuous(limits = c(0, 10),
                     expand = expansion(add = c(.5, 0)),
                     labels = round) +
  scale_fill_distiller('Número\nde películas',
```

```
                      palette = 'YlOrRd',
                      direction = 1,
                      limits = c(1, 20),
                      breaks = scales::pretty_breaks(4)) +
  scale_size_continuous('Número\nde películas',
                        range = c(1, 10),
                        limits = c(1, 20),
                        breaks = scales::pretty_breaks(4)) +
  guides(fill = guide_legend(),
         size = guide_legend()) +
  labs(title = 'Rating de chilenas por año en IMDb',
       subtitle = 'Películas desde 1990 en adelante',
       caption = 'Fuente: IMDb.com. Web Scraping y acceso a datos desde la web',
       y = 'Promedio rating IMDb') +
  theme_minimal()
```



Fuente: IMDb.com. Web Scraping y acceso a datos desde la web

6) ¿Cuál es el *género* que tienen el *mejor puntaje promedio* considerando películas estrenadas desde 1990 a la fecha?

Modificar base para que la unidad de análisis sea `genero`.

```
df_genero <- df_peliculas |>
  filter(anio >= as.Date("1990-01-01")) |>
  select(index, rating, genero) |>
  unnest_longer(col = genero)
```

```r
df_genero <- df_genero |>
  summarise(n_peliculas = n(),
            n_peliculas_con_rating = sum(!is.na(rating)),
            rating = mean(rating, na.rm = TRUE),
            .by = genero) |>
  arrange(-rating)

head(df_genero)
```

```
# A tibble: 6 x 4
  genero    n_peliculas n_peliculas_con_rating rating
  <chr>           <int>                  <int>  <dbl>
1 Biography          11                      8   7.24
2 Music               7                      5   7.22
3 <NA>                4                      1   7.2
4 History             7                      7   7.03
5 Animation           3                      3   6.9
6 Drama             174                    145   6.17
```

El género con mejor puntaje promedio desde 1990 es **Biography**.

```r
df_genero |>
  arrange(-n_peliculas) |>
  head()
```

```
# A tibble: 6 x 4
  genero   n_peliculas n_peliculas_con_rating rating
  <chr>          <int>                  <int>  <dbl>
1 Drama            174                    145   6.17
2 Comedy            67                     54   5.42
3 Thriller          30                     27   5.67
4 Romance           28                     26   6.09
5 Action            27                     25   5.31
6 Horror            25                     23   4.78
```