R 프로그래밍 기초다지기

9강 - 웹 스크래핑을 통한 나만의 데이터 만들기

슬기로운통계생활

Issac Lee

데이터 스크래핑(Scraping)

데이터 스크래핑이란

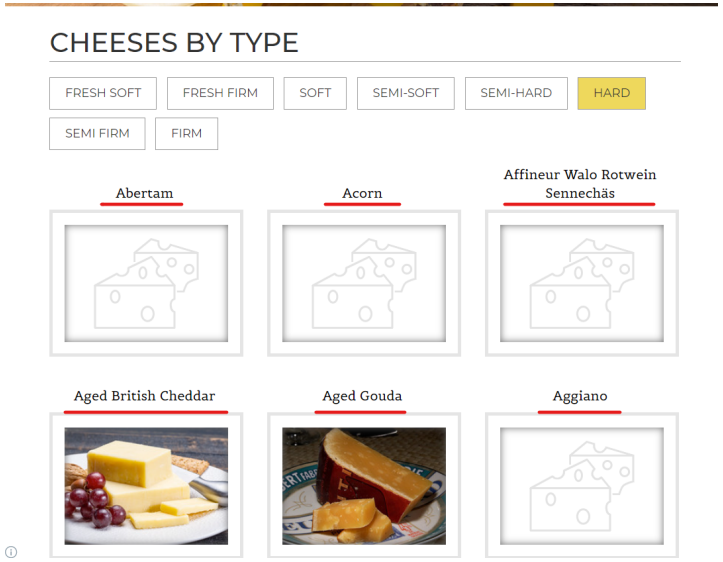
정보의 가공 및 추출

- 여러 형태의 자료에서 원하는 정보만을 **쏙쏙 빼와서** 새로운 형태의 데이터를 만드는 과정
- 크롤링이랑 **다른** 개념
  - 크롤링이란 엄청나게 큰 네트워크를 어떻게 효율적으로 전부 다 수집할 것인지에 대한 이야기

웹 스크래핑의 예

제품 이름만 쏙 골라 넣기

- 제품 정보를 추출 후 데이터 프레임으로 만들고 싶을 때



cheese
Abertam
Acorn
Affineur
Aged British Cheddar
Aged Gouda
Aggiano

주의할 점

지적 재산권 침해 요소

- 내가 스크랩하는 정보가 다른 사람의 재산은 아닌가?
  - 웹에는 특정 정보를 사용해서 사업을 하는 경우가 많음
  - 숙박업소, 제품 판매 사이트 (가격 정보)
- 스크랩이 허용된 정보인가?
  - `robots.txt` 확인
- 나의 스크랩 활동이 사이트의 트래픽에 영향을 주지는 않는가?

비장의 무기 (a.k.a. 준비물)

크롬 확장 프로그램

- SelectorGadget
  - 웹 사이트에서 특정 부분의 내용이 어떤 태그에 물려있는지 알려주는 도구
  - `rvest` 패키지와 궁합이 너무 좋음.

사용 패키지

`rvest` 패키지

```
# install.packages("rvest")
library(rvest)
```

- Easily Harvest (Scrape) Web Pages
- 여러 페이지를 스크랩 할 때 `polite` 패키지를 꼭 같이 사용할 것.

`rvest` 사용법

주요 함수들

- `read_html()`: 웹 페이지 읽어오기
- `html_elements()`: 특정 요소에 해당하는 내용 추출하기
- `html_attr()`: 특정 태그에 해당하는 값 추출하기
- `html_text()`: 추출한 내용 텍스트로 바꾸기

`issaclee.com` 웹 페이지 접근

기초 통계 사이트 목차 가져오기

- https://www.theissaclee.com/ko/courses/rstat101/
- 오른쪽 클릭 > 페이지 소스보기

```
url <- "https://www.theissaclee.com/ko/courses/rstat101/"
web_page <- read_html(url)
print(web_page)
```

```
## {html_document}
## <html lang="ko">
## [1] <head>\n<meta http-equiv="Content-Type" content="text/html; ch
## [2] <body id="top" data-spy="scroll" data-offset="70" data-target=
```

목차 태그 접근

- 셀렉터 가젯을 통해서 태그 추출하기
    - 원하는 부분 클릭 후, 원하지 않는 부분 클릭으로 빨간색 처리

```
chapter_name <- web_page |>
    html_elements(".docs-sidenav a") |>
    html_text()
head(chapter_name)
```

```
## [1] "Week 1 - R 기초 및 데이터 불러오기" "Week 2 - 데이터 시각화"
## [3] "Week 3 - 분포를 나타내는 지표"      "Week 4 - 데이터 다루기"
## [5] "Week 5 - 상관계수"                  "Week 6 - 회귀분석 기초"
```

`transfermarkt.com` 스크랩핑

스크랩이 가능한가?

- `google.com/robots.txt` 내용을 확인해보자.
- `transfermarkt.com/robots.txt` 주소로 접근

  스크랩 가능 확인

  - User-agent: *
  - Allow: /

선수 이름 따오기

Most valuable player 페이지

- 선수 이름과 연관된 태그 선택

```
url <- "https://www.transfermarkt.com/spieler-statistik/wertvollstesp
web_page <- read_html(url)

player_name <- web_page |>
    html_elements("#yw1 .spielprofil_tooltip") |>
    html_text()
head(player_name)
```

```
## [1] "Kylian Mbappe"   "Erling Haaland" "Harry Kane"     "Jadon Sanc
## [5] "Mohamed Salah"   "Romelu Lukaku"
```

국가 정보 가져오기

태그를 이용한 정보 추출

- 태그 정보에 국가 정보가 들어있는 것을 확인하자.
    - 태그 정보를 꺼내올 땐 `html_attr()`

```
national <- web_page |>
    html_elements(".flaggenrahmen") |>
    html_attr("title")

head(national)
```

```
## [1] "France"  "Norway"  "England" "England" "Egypt"   "Belgium"
```

```
print(national[6], width = 1000)
```

클럽정보 가져오기

```
club_name <- web_page |>
    html_elements("#yw1 .vereinprofil_tooltip") |>
    html_children() |>
    html_attr("alt")
head(club_name)
```

```
## [1] "Paris Saint-Germain" "Borussia Dortmund"   "Tottenham Hotspur
## [4] "Manchester United"   "Liverpool FC"        "Chelsea FC"
```

선수 나이 가져오기

```
player_age <- web_page |>
    html_elements("#yw1 .zentriert:nth-child(3)") |>
    html_text()
player_age <- player_age[-1] |> as.integer()
head(player_age)
```

```
## [1] 22 21 28 21 29 28
```

포지션 정보 가져오기

```
position <- web_page |>
    html_elements(".inline-table tr+ tr td") |>
    html_text()
head(position)
```

```
## [1] "Centre-Forward" "Centre-Forward" "Centre-Forward" "Right Wing
## [5] "Right Winger"   "Centre-Forward"
```

Market value 가져오기

정규 표현식의 파워

```
# install.packages("stringr")
library("stringr")
market_value <- web_page |>
    html_elements("#yw1 b") |>
    html_text() |>
    str_extract("\\d+[.]\\d\\d")
head(market_value)
```

```
## [1] "160.00" "130.00" "120.00" "100.00" "100.00" "100.00"
```

```
length(market_value)
```

```
## [1] 25
```

데이터 프레임으로 만들어 저장하기

- `national` 변수: 중복 국적 처리
- 데이터 프레임으로 만든 후 `write.csv()` 함수를 사용하여 저장

```
soccer_data <- data.frame(
    name = player_name,
    age = player_age,
    position = position,
    nationality = national[-c(7, 13, 22)],
    club = club_name,
    market_value = market_value
)
write.csv(soccer_data,
          file = "./data/soccer.csv",
          row.names = FALSE,
          fileEncoding = "UTF-8")
```

중간 데이터 점검

```
head(soccer_data)
```

```
##              name age       position nationality                clu
## 1  Kylian Mbappe  22 Centre-Forward      France Paris Saint-Germai
## 2 Erling Haaland  21 Centre-Forward      Norway   Borussia Dortmun
## 3     Harry Kane  28 Centre-Forward     England   Tottenham Hotspu
## 4   Jadon Sancho  21   Right Winger     England   Manchester Unite
## 5  Mohamed Salah  29   Right Winger       Egypt        Liverpool F
## 6  Romelu Lukaku  28 Centre-Forward     Belgium          Chelsea F
##   market_value
## 1       160.00
## 2       130.00
## 3       120.00
## 4       100.00
## 5       100.00
## 6       100.00
```

중복데이터는 어떻게 할까?

- 테이블로 선택자를 잡아서 내용이 몇개가 있는지 세어보자.

```
national2 <- web_page |>
    html_elements("#yw1 .zentriert:nth-child(4)")
national2 <- national2[-1]

html_children(national2[6]) |> length()
```

```
## [1] 3
```

```
count_dup <- sapply(national2,
                    \(x) length(html_children(x)))
which(count_dup == 3)
```

```
## [1]  6 12 21
```

childeren와 attr 콤보를 사용한 추출

```
result <- sapply(national2,
       \(x) html_attr(html_children(x)[1], "alt"))
head(result)
```

```
## [1] "France"  "Norway"  "England" "England" "Egypt"   "Belgium"
```

```
length(result)
```

```
## [1] 25
```

여러 페이지 추출하기

2페이지 주소 알아내기

- 오른쪽 클릭 > 링크 주소 복사
- 주소 + "?page=number" 구조

```
base_url <- "https://www.transfermarkt.com/spieler-statistik/wertvoll
url <- paste0(base_url, 1:3)
url |> substr(50, nchar(url))
```

```
## [1] "ertvollstespieler/marktwertetop?page=1"
## [2] "ertvollstespieler/marktwertetop?page=2"
## [3] "ertvollstespieler/marktwertetop?page=3"
```

`lapply()`를 사용한 여러 페이지 스크래핑

- `lapply()` 후에 `unlist()`로 벡터로 받음.

```
twopage_scraping <- function(url
    Sys.sleep(1)
    web_page <- read_html(url)

    player_name <- web_page |>
        html_elements("#yw1 .spi
        html_text()
    player_name
}
```

```
result <- lapply(url, twopage_sc
player_name <- unlist(result)
length(player_name)
```

```
## [1] 75
```

```
head(player_name)
```

```
## [1] "Kylian Mbappe"  "Erling
## [5] "Mohamed Salah"  "Romelu
```

완강!

기초 R 프로그래밍 완강을 축하드립니다!

참고자료 및 사용교재

[1] The art of R programming

- R 공부하시는 분이면 꼭 한번 보셔야 하는 책입니다.
- 위 교재의 한글 번역본 빅데이터 분석 도구 R 프로그래밍도 있습니다. 도서 제목 클릭하셔서 구매하시면 저의 사리사욕을 충당하는데 도움이 됩니다.

[2] Web Scraping in R: Get Text from ANY Website

- SelectorGadget을 소개시켜준 고마운 유튜버

코스 홍보

[1] 클래스101 기초 통계 강의

- 제가 하는 기초 통계강의. 통계를 대하는 여러분의 생각을 바꿔주는 기초 강의!