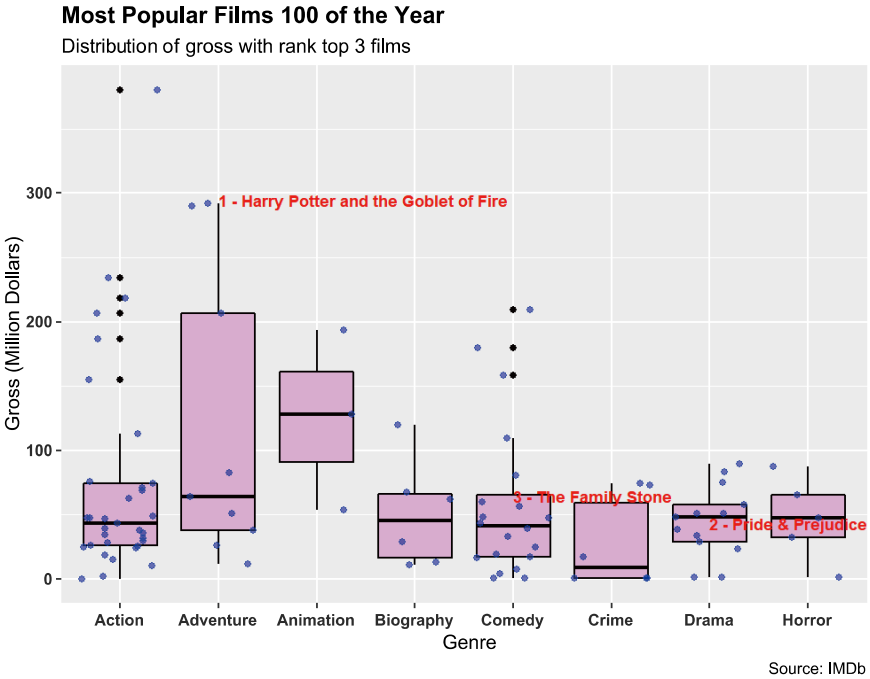
그림 4-25 **올해의 영화 100.** 장르별 흥행 수익의 분포를 확인할 수 있다. 평균적으로 가장 높은 흥행 수익을 가져오는 장르는 애니메이션이다. 어드벤처 장르는 영화에 따라 흥행 수익의 편차가 가장 크다.



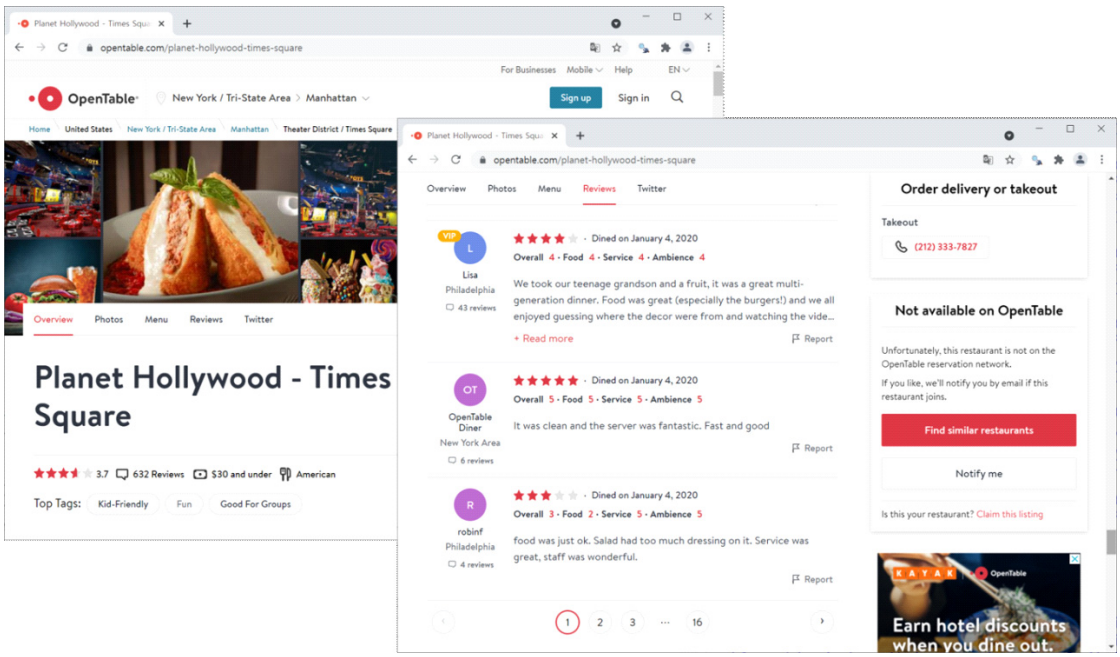
레스토랑 리뷰 @오픈테이블

수집하려는 데이터가 하나의 웹페이지에 모두 포함되어 있는 경우라면 지금까지 해왔던 방법대로 웹스크레이핑을 하면 충분하다. 그런데 웹사이트의 데이터를 추출하다보면 그렇지 않은 경우도 종종 접하게 된다. 예를 들어, 오픈테이블(OpenTable) 웹사이트(https://www.opentable.com)에서 검색한 레스토랑에 대한 고객의 리뷰글과 평점 데이터는 여러 페이지에 걸쳐 게시되어 있다. 따라서 이런 형식으로 게시된 데이터를 수집하기 위해서는 일련의 연속된 웹페이지에 반복적으로 접근할 수 있는 체계적 방법이 필요하다.

오픈테이블 웹사이트로부터 여러 페이지에 걸쳐 게시된 레스토랑의 리뷰 데이터를 추출해보자. [그림 4-26]은 미국 뉴욕에 위치한 레스토랑 Planet Hollywood를 방문한 고객의 리뷰이다. 웹페이지로부터 레스토랑 이름, 리뷰글 작성자, 작성 일자, 리뷰글, 평점 데이터를 수집한다.

먼저 리뷰 첫 페이지의 데이터를 추출한 후 이를 전체 페이지로 확장해보자. 첫 번째 리뷰 페이지로부터 다음과 같이 웹페이지의 HTML 문서를 읽어들인 후 파싱한다. 그리고 데이터 전처리를 위한 `tidyverse` 패키지를 메모리에 적재한다.[17]

그림 4–26 **레스토랑 리뷰.** 오픈테이블 웹사이트로부터 Planet Hollywood 레스토랑 고객의 리뷰 데이터를 수집한다.




```
> library(httr)
> library(XML)
> url <- "https://www.opentable.com/planet-hollywood-times-square"
> html <- GET(url)
> html.parsed <- htmlParse(html)
> library(tidyverse)
```

HTML 문서를 성공적으로 파싱했으면 이제 파싱한 문서로부터 데이터를 추출해보자. 우선 레스토랑의 이름을 수집한다. 레스토랑 이름은 다음과 같이 `<h1>` 노드로부터 추출할 수 있다. 추출한 텍스트로부터 간혹 이름 내에 포함되어 있는 비ASCII 문자(non-ASCII character)와 불필요한 공백을 제거한다.

```
<h1 class="eM9Li2wbkQvvjxZB11sV vPFUfeHOataYlYsp0m6G">Planet Hollywood - Times Square</h1>
```

17 `tidyverse` 패키지에는 데이터 및 텍스트 전처리(`forcats`, `tibble`, `stringr`, `readr`, `dplyr`, `purrr`, `tidyr` 등)와 그래프 생성(`ggplot2`)에 필요한 유용한 패키지들이 포함되어 있다. 따라서 `tidyverse` 패키지를 적재하는 것만으로 이 패키지들을 개별적으로 메모리에 적재하는 과정 없이도 한 번에 사용할 수 있다. `tidyverse` 패키지를 처음 접하거나 익숙하지 않은 독자는 '부록 Tidyverse'를 먼저 학습하기 바란다.

```
> name <- xpathSApply(html.parsed, "//h1", xmlValue)
> name <- str_replace_all(iconv(name, to="ascii", sub=""), "\\s+", " ")
> name
[1] "Planet Hollywood - Times Square"
```

리뷰글 작성자, 리뷰글 작성 일자, 리뷰글, 평점 데이터는 다음과 같이 모두 <ol> 노드 아래에 모여 있다. 노드 아래의 <li> 노드에 리뷰별로 데이터가 저장되어 있다. 현재 한 페이지에 40개의 리뷰가 게시되어 있으므로 <ol> 노드 아래에는 리뷰별로 모두 40개의 <li> 노드가 존재한다.

```
▼<ol aria-label="Reviews List" id="restProfileReviewsContent" class="dxLeFbidu7bvjN_7QEC2" data-test="reviews-list">
  ▶<li class="mbK0eco4h3AaJgKLBhb9">…</li> flex
  ▶<li class="mbK0eco4h3AaJgKLBhb9">…</li> flex
  ⋮
  ▶<li class="mbK0eco4h3AaJgKLBhb9">…</li> flex
  ▼<li class="mbK0eco4h3AaJgKLBhb9"> flex == $0
    ▼<section class="Dy9dg_fThB5zI1c7xOg4"> flex
      <div class="fay3YS5td0q6mbuTy24Y vd5ofjvNcRSMKcw9O0D8">r</div> flex
      <p class="V0gqMM0V5nr33Ha56k83 mnvRsuAUHkGrXs3xjRoy">robinf</p>
      <p class="UGzxfqPCuflxNBXSUvL_ mnvRsuAUHkGrXs3xjRoy">Philadelphia</p>
    ▶<div class="SaFk3NbFipF4UzPmNV1S NvQhR1VNbSvDtuZB7N6I">…</div> flex
    </section>
    ▼<section class="ADHOnxYxENUUfweaCdW6">
      ▼<section class="c3G0xjzKlnaEr6jJr29v mnvRsuAUHkGrXs3xjRoy"> flex
        ▶<div class="Ky7CQq2rCwG8t_monZNj">…</div> flex
        ▶<div class="SaFk3NbFipF4UzPmNV1S pZDUTHhABtOStIz6eCrJ">…</div>
        <p class="Xfrgl6cRPxn4vwFrFgk1">Dined on January 5, 2020</p>
      </section>
      <span class="ntw1CLrK8o6PhYhcRux1">overall</span>
      ▼<span class="Q2DbumELlxH4s85dk8Mj">
        "3"
        ::after
      </span>
      <span class="ntw1CLrK8o6PhYhcRux1">food</span>
      ▼<span class="Q2DbumELlxH4s85dk8Mj">
        "2"
        ::after
      </span>
      <span class="ntw1CLrK8o6PhYhcRux1">service</span>
      ▼<span class="Q2DbumELlxH4s85dk8Mj">
        "5"
        ::after
      </span>
      <span class="ntw1CLrK8o6PhYhcRux1">ambience</span>
      <span class="Q2DbumELlxH4s85dk8Mj">5</span>
      ▼<div class="f3cNr0xDcwvMaBkpjQj5">
        ▼<p class="t9JcvSL3Bsj1lxMSi3pz h_kb2PFOoyZe1skyGiz9 Ti64w3n01MDTYZb59n6Q" style="line-height:24px;
        max-height:72px;overflow:hidden;-webkit-line-clamp:3">
          "food was just ok. Salad had too much dressing on it. Service was great, staff was wonderful."
        </p>
      </div>
    ▶<div class="HL583KqLswHVweEv7sYa">…</div> flex
    </section>
  </li>
</ol>
```

HTML 코드를 살펴보면 각 리뷰별 데이터를 저장하고 있는 <li> 노드에는 하위에 두 개의 <section> 노드가 있다. 첫 번째 <section> 노드에는 리뷰글 작성자가 포함되어 있고 두 번째 <section> 노드에는 리뷰글 작성 일자, 리뷰글, 평점 데이터가 저장되어 있다. 이러한 정보를 바탕으로 차례대로 데이터를 수집해보자.

우선 리뷰글 작성자 데이터는 다음과 같이 추출한다. 노드 아래의 첫 번째 <section> 노드셋으로부터 추출할 수 있다. 한 페이지에 40개의 리뷰가 있으므로 40명의 작성자 데이터가 수집된다.

```
> author <- xpathSApply(html.parsed,
+                       "//ol[@id='restProfileReviewsContent']/li/section[1]/p[1]",
+                       xmlValue)
> author
 [1] "OSCARP"            "FrankD"            "OpenTable Diner"   "OpenTable Diner"
 [5] "natalieg"          "OpenTable Diner"   "OpenTable Diner"   "Jane"
…(중략)
[33] "OpenTable Diner"   "vipTK719"          "GregH"             "OpenTable Diner"
[37] "vipBert"           "vipLisa"           "OpenTable Diner"   "robinf"
```

이어서 작성 일자를 수집해보자. 작성 일자는 다음과 같이 <li> 노드 아래의 두 번째 <section> 노드셋으로부터 추출할 수 있다.

```
> date <- xpathSApply(html.parsed,
+                     "//ol[@id='restProfileReviewsContent']/li/section[2]/section[1]/p",
+                     xmlValue)
> date
 [1] "Dined on March 12, 2020"    "Dined on March 1, 2020"
 [3] "Dined on February 27, 2020" "Dined on February 26, 2020"
…(중략)
[37] "Dined on January 5, 2020"   "Dined on January 4, 2020"
[39] "Dined on January 4, 2020"   "Dined on January 4, 2020"
```

추출한 텍스트는 lubridate 패키지의 mdy() 함수를 이용하여 날짜 형식으로 변환한다. 날짜 형식의 변환을 위해 Sys.setlocale() 함수를 이용하여 시간의 로케일 설정을 북미표준으로 일시 변경한다.

```
> library(lubridate)
> Sys.setlocale("LC_TIME", "English")
[1] "English_United States.1252"
```

```
> mdy(date)
 [1] "2020-03-12" "2020-03-01" "2020-02-27" "2020-02-26" "2020-02-25" "2020-02-23"
 [7] "2020-02-20" "2020-02-20" "2020-02-19" "2020-02-19" "2020-02-19" "2020-02-18"
…(중략)
[31] "2020-01-25" "2020-01-24" "2020-01-23" "2020-01-21" "2020-01-19" "2020-01-05"
[37] "2020-01-05" "2020-01-04" "2020-01-04" "2020-01-04"
> Sys.setlocale()
[1] "LC_COLLATE=Korean_Korea.949;LC_CTYPE=Korean_Korea.949;LC_MONETARY=Korean_Korea.949;
LC_NUMERIC=C;LC_TIME=Korean_Korea.949"
```

텍스트로 표현된 리뷰글 작성 일자가 날짜 형식으로 성공적으로 변환되었다. 그런데 이러한 변환이 항상 제대로 작동하는 것은 아니다. 왜냐하면 오픈테이블 웹사이트의 리뷰글 작성 일자는 [그림 4-27]에서 볼 수 있는 것처럼 보다 다양한 형태로 게시되기 때문이다. 리뷰글이 작성된지 얼마 안 되었을 때는 그 기간에 따라 11 hours ago, a day ago(1 day ago), 2 days ago 등과 같은 형태로 표현된다. 이런 경우에는 `lubridate` 패키지의 `mdy()` 함수를 이용하여 텍스트로부터 날짜 데이터를 추출할 수 없다. 따라서 다양한 형태로 표현된 리뷰글 작성 일자를 처리할 수 있는 다음과 같은 사용자 정의 함수를 작성한다.

```
> dateExtract <- function(date) {
+   ifelse(str_detect(date, "hours"), as.character(Sys.Date()),
+          (ifelse(str_detect(date, "(a|1) day"), as.character(Sys.Date() - 1),
+                  (ifelse(str_detect(date, "days"),
+                          as.character(Sys.Date() - parse_number(date)),
+                          as.character(mdy(date)))))))
+ }
```

이 사용자 정의 함수를 이용하면 다음과 같은 형태의 텍스트 데이터를 날짜 형식으로 변환할 수 있다.[18]

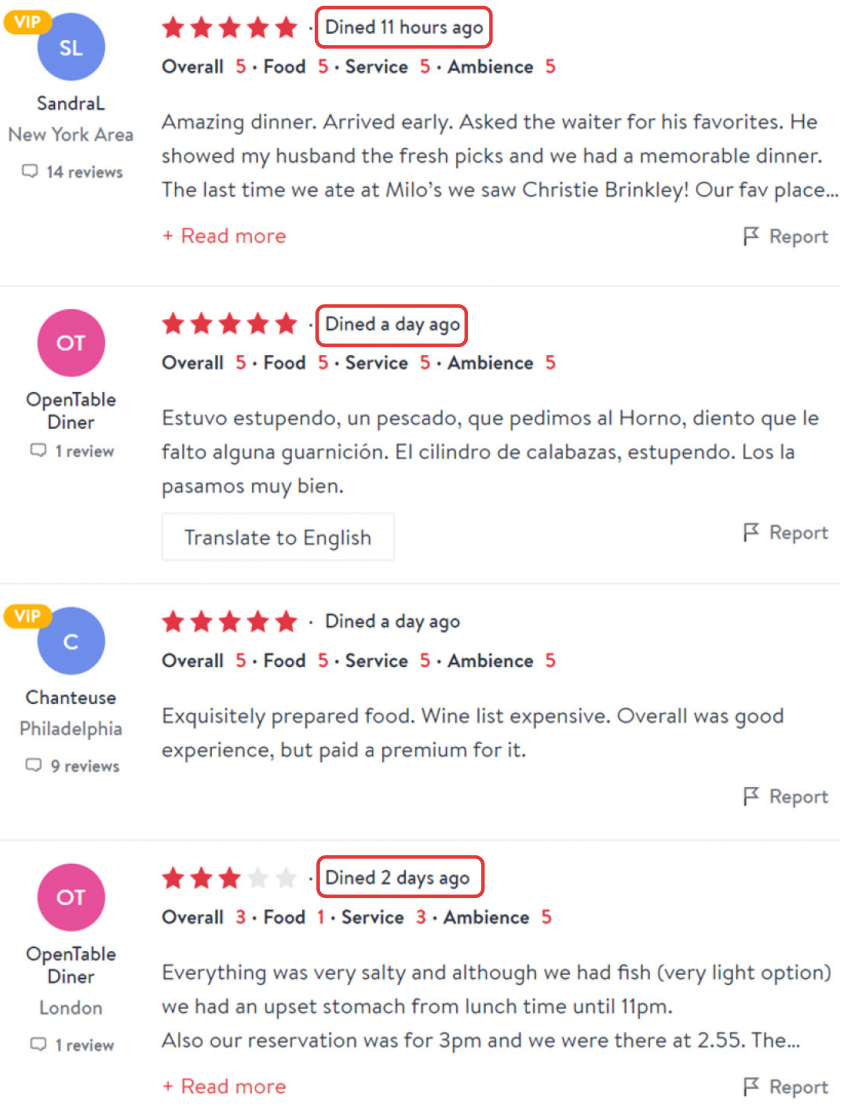
18 여기에서의 변환 결과는 현재 날짜를 2022년 2월 6일로 가정한 것이다.

```
> date.sample <- c("Dined on January 29, 2022", "Dined on 11 hours ago",
+                  "Dined on a day ago", "Dined on 1 day ago", "Dined on 2 days ago")
> Sys.setlocale("LC_TIME", "English")
[1] "English_United States.1252"
> map_chr(date.sample, dateExtract) %>%
+   as.Date()
[1] "2022-01-29" "2022-02-06" "2022-02-05" "2022-02-05" "2022-02-04"
> Sys.setlocale()
[1] "LC_COLLATE=Korean_Korea.949;LC_CTYPE=Korean_Korea.949;LC_MONETARY=Korean_Korea.949;
LC_NUMERIC=C;LC_TIME=Korean_Korea.949"
```

이제 수집한 날짜 텍스트에 앞서 정의한 사용자 정의 함수를 적용하여 다음과 같이 날짜 형식으로 변환된 작성 일자 데이터를 추출한다.

```
> Sys.setlocale("LC_TIME", "English")
[1] "English_United States.1252"
> date <- map_chr(date, dateExtract) %>%
+   as.Date()
```

그림 4-27 **레스토랑 리뷰.** 오픈테이블 웹사이트의 리뷰글 작성 일자는 다양한 형태로 게시된다.



```
> date
 [1] "2020-03-12" "2020-03-01" "2020-02-27" "2020-02-26" "2020-02-25" "2020-02-23"
 [7] "2020-02-20" "2020-02-20" "2020-02-19" "2020-02-19" "2020-02-19" "2020-02-18"
…(중략)
[31] "2020-01-25" "2020-01-24" "2020-01-23" "2020-01-21" "2020-01-19" "2020-01-05"
[37] "2020-01-05" "2020-01-04" "2020-01-04" "2020-01-04"
> Sys.setlocale()
[1] "LC_COLLATE=Korean_Korea.949;LC_CTYPE=Korean_Korea.949;LC_MONETARY=Korean_Korea.949;
LC_NUMERIC=C;LC_TIME=Korean_Korea.949"
```

리뷰글을 추출해보자. 리뷰글 또한 작성 일자와 마찬가지로 <li> 노드 아래의 두 번째 <section> 노드셋에 포함되어 있으므로 다음과 같이 수집할 수 있다.

```
> review <- xpathSApply(html.parsed,
+                       "//ol[@id='restProfileReviewsContent']/li/section[2]/div/p",
+                       xmlValue)
> review
 [1] "Estaba vacío y a pesar de que habíamos reservado con una semana de antelacióny
pudiendonos sentar en mesa con  bancos mas comodos ya que asi lo solicitamos, nos
colocaron en una mesa alta al lado de la barra. No se enteraban de lo que pedíamosy
trajeron comida distinta a pesar de que se lo señalamos en la carta.Y tardaron mucho
en servir la comida y estaba vacío el local."
 [2] "The hostess responsible for greeting was the worst - didn't check us in left us
just waiting and there were no other customers so it was not like she was busy - but
we have an amazing waitress and she was great with my two kids and I felt better."
…(중략)
[39] "It was clean and the server was fantastic. Fast and good"
[40] "food was just ok.  Salad had too much dressing on it.  Service was great, staff
was wonderful."
```

오픈테이블은 전반적인 만족도(overall)와 함께 음식(food), 서비스(service), 분위기(ambience)에 대한 개별 평가도 제공한다. 이 네 가지 평점은 모두 <li> 노드 아래 두 번째 <section> 노드 하위의 노드에 포함되어 있으며 다음과 같이 순서 정보를 이용하여 차례대로 추출할 수 있다.

```
> overall <- xpathSApply(html.parsed,
+                        "//ol[@id='restProfileReviewsContent']/li/section[2]/span[2]",
+                        xmlValue) %>%
+   as.numeric()
> overall
 [1] 2 3 1 5 5 1 5 2 2 4 2 5 5 5 5 4 5 4 4 1 4 1 5 2 5 3 4 2 1 2 5 2 1 4 5 5 2 4 5 3
```

```
> food <- xpathSApply(html.parsed,
+                     "//ol[@id='restProfileReviewsContent']/li/section[2]/span[4]",
+                     xmlValue) %>%
+   as.numeric()
> food
 [1] 3 4 2 4 5 1 3 2 3 3 2 5 4 5 5 5 5 4 5 1 3 1 5 2 5 2 4 2 1 1 5 2 1 4 5 5 3 4 5 2
> service <- xpathSApply(html.parsed,
+                        "//ol[@id='restProfileReviewsContent']/li/section[2]/span[6]",
+                        xmlValue) %>%
+   as.numeric()
> service
 [1] 3 4 1 5 5 1 5 5 1 4 3 5 5 5 5 3 5 5 4 2 5 2 5 2 5 3 4 2 4 5 5 4 3 4 5 5 3 4 5 5
> ambience <- xpathSApply(html.parsed,
+                         "//ol[@id='restProfileReviewsContent']/li/section[2]/span[8]",
+                         xmlValue) %>%
+   as.numeric()
> ambience
 [1] 2 4 1 4 5 1 3 5 5 3 2 5 5 5 5 3 5 5 5 1 5 1 5 4 5 2 4 2 1 4 5 2 1 4 5 5 2 4 5 5
```

첫 번째 페이지의 데이터를 모두 성공적으로 추출하였으므로 이제 이를 전체 페이지로 확장해보자. 전체 웹페이지 개수는 다음과 같이 추출할 수 있다. HTML 코드를 살펴보면 <footer> 노드 아래의 <li> 노드 가운데 마지막 노드에 총페이지 개수가 포함되어 있다.

```
▼<footer class="BWbRBv1QpGhTvTkicaoh" data-test="reviews-pagination" data-testid="reviews-pagination">
  ▼<div class="GoM3Km_S9FLVvAsYhBfh" data-test="pagination-controls"> flex
    ▶<div class="wcWSeV93ScrQukE9xxag">…</div>
    ▼<ul class="BklkG9D8lgpsm_Z7JPTQ"> flex
      ▶<li class="NnyJ_FqQ3wSWIkDcb06u" data-test="pagination-link">…</li>
      ▶<li class="NnyJ_FqQ3wSWIkDcb06u" data-test="pagination-link">…</li>
      ▶<li class="NnyJ_FqQ3wSWIkDcb06u" data-test="pagination-link">…</li>
      ▶<li class="NnyJ_FqQ3wSWIkDcb06u" data-test="pagination-link">…</li>
      ▼<li class="NnyJ_FqQ3wSWIkDcb06u" data-test="pagination-link">
          <a href="/" class="JINvM1QBDFqby0Geg27_ yGrkuQrH99UKfIE4YXQJ lq_cIcdgZyPE1fNfg6uz mnvRsuAUHkGrXs3x
          jRoy" aria-label="Go to page number 16">16</a> flex == $0
        </li>
      </ul>
    ▶<div class="iIrPGlN8h2M_kZhJJZH_">…</div>
  </div>
</footer>
```

```
> total.pages <- xpathSApply(html.parsed,
+                            "//footer[@data-test='reviews-pagination']//ul/li[last()]",
+                            xmlValue) %>%
+   as.numeric()
> total.pages
[1] 16
```

지금까지 작성한 코드를 바탕으로 다음과 같이 레스토랑의 리뷰 데이터를 추출하는 opentableReview() 함수를 작성한다. opentableReview() 함수는 인수로 제공한 웹페이지의 URL(baseurl)과 추출할 페이지 개수(n)를 바탕으로 오픈테이블의 고객 리뷰 데이터를 수집하여 데이터 프레임(티블) 형식으로 저장한다.

```
> opentableReview <- function(baseurl, n=NULL) {
+   library(tidyverse)
+   library(lubridate)
+   library(httr)
+   library(XML)
+   html <- GET(baseurl)
+   html.parsed <- htmlParse(html)
+ if (is.null(n)) {                                                    ❶
+     total.pages <- xpathSApply(html.parsed,
+                                "//footer[@data-test='reviews-pagination']//ul/li[last()]",
+                                xmlValue) %>%
+     as.numeric()
+   n <- total.pages
+ }
+   opentable.review <- tibble()
+   name <- xpathSApply(html.parsed, "//h1", xmlValue)
+   name <- str_replace_all(iconv(name, to="ascii", sub=""), "\\s+", " ")
+ dateExtract <- function(date) {                                      ❷
+   ifelse(str_detect(date, "hours"), as.character(Sys.Date()),
+          (ifelse(str_detect(date, "(a|1) day"), as.character(Sys.Date() - 1),
+                  (ifelse(str_detect(date, "days"),
+                          as.character(Sys.Date() - parse_number(date)),
+                          as.character(mdy(date)))))))
+ }
+   Sys.setlocale("LC_TIME", "English")
+   for (i in c(1:n)) {
+   if (i==1) {url <- baseurl}                                         ❸
+   else {
+     url <- str_c(baseurl, "?page=", i)
+   }
+     html <- GET(url)
+     html.parsed <- htmlParse(html)
+   author <- xpathSApply(html.parsed,                                 ❹
+                         "//ol[@id='restProfileReviewsContent']/li/section[1]/p[1]",
+                         xmlValue)
+   date <- xpathSApply(html.parsed,
+                       "//ol[@id='restProfileReviewsContent']/li/section[2]/section[1]/p",
```

```
+                       xmlValue) %>%
+     map_chr(dateExtract) %>%
+     as.Date()
+   review <- xpathSApply(html.parsed,
+                         "//ol[@id='restProfileReviewsContent']/li/section[2]/div/p",
+                         xmlValue)
+   overall <- xpathSApply(html.parsed,
+                          "//ol[@id='restProfileReviewsContent']/li/section[2]/span[2]",
+                          xmlValue) %>%
+     as.numeric()
+   food <- xpathSApply(html.parsed,
+                       "//ol[@id='restProfileReviewsContent']/li/section[2]/span[4]",
+                       xmlValue) %>%
+     as.numeric()
+   service <- xpathSApply(html.parsed,
+                          "//ol[@id='restProfileReviewsContent']/li/section[2]/span[6]",
+                          xmlValue) %>%
+     as.numeric()
+   ambience <- xpathSApply(html.parsed,
+                           "//ol[@id='restProfileReviewsContent']/li/section[2]/span[8]",
+                           xmlValue) %>%
+     as.numeric()
+   if (length(date) > 0 ) {
+     opentable.r <- tibble(name=name, author=author, date=date, review=review,
+                           rating.overall=overall, rating.food=food,
+                           rating.service=service, rating.ambience=ambience)
+     opentable.review <- bind_rows(opentable.review, opentable.r)
+   }
+   else break
+   Sys.sleep(sample(10, 1)*0.2)                                          ❺
+ }
+ Sys.setlocale()
+ opentable.review <- bind_cols(id=1:nrow(opentable.review), opentable.review)   ❻
+ return(opentable.review)
+}
```

❶ 추출할 페이지 개수가 주어지지 않을 경우(즉 n=NULL) 전체 웹페이지 개수 추출
❷ 날짜 추출 함수 정의
❸ 웹페이지 URL 생성
❹ 항목별 데이터 추출
❺ 웹페이지 접속 간격 통제
❻ 리뷰 ID 부여