오픈API 연동 (2) - 영화 진흥 위원회 박스 오피스 데이터 활용

영화진흥위원회 OpenAPI를 활용하면 현재 극장에서 상영중인 영화들에 대한 박스오피스 데이터를 조회할 수 있습니다.

이번 포스팅에서는 영화진흥원 OpenAPI를 통해 JSON 형식의 데이터를 수집하고 이를 그래프로 시각화 하는 내용을 소개합니다.

#01. 기본 준비

1) Key 발급받기

1. 영화진흥위원회 OpenAPI 사이트에 회원가입 후 로그인을 수행한다.
2. **키 발급/관리** 메뉴를 통해 연동에 필요한 인증키 발급받는다.

2) 연동 스팩 확인하기

http://www.kobis.or.kr/kobisopenapi/homepg/apiservice/searchServiceInfo.do 페이지를 통해 요청에 필요한 정보와 응답 형식을 확인한다.

3) 브라우저를 통한 URL 확인

영화진흥원 OpenAPI 명세서를 통해 요청변수 항목들을 확인하여 전체 URL 구성 후 브라우저를 통해 접근한다.

> http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.json?key=자신의Key값&targetDt=20200110

targetDt값의 경우 하루 전 날짜까지 적용 가능하다. 오늘에 해당하는 값을 전달한 경우 아직 집계되지 않은 데이터이므로 데이터가 조회회지 않는다.

웹 브라우저를 통해 JSON을 직접 확인하는 경우 JSON 전문이 한 줄로 표시되어 확인하기 어려운 부분이 있다.

이 경우 크롬 브라우저를 통해 **크롬 웹 스토어**에서 **JSONView** 확장 프로그램 설치하고 재접속 하면 JSON 데이터를 보기 좋게 정렬하고 컬러링을 적용해 준다.

3) 패키지 로드

```
REPO_URL <- "https://cran.seoul.go.kr/"
if (!require(httr))       install.packages("httr", repos=REPO_URL)
if (!require(rjson))      install.packages("rjson", repos=REPO_URL)
if (!require(dplyr))      install.packages("dplyr", repos=REPO_URL)
if (!require(ggplot2))    install.packages("ggplot2", repos=REPO_URL)
if (!require(extrafont))  install.packages("extrafont", repos=REPO_URL)

library(httr)       # 온라인상의 데이터를 가져오기 위한 통신 기능 패키지
library(rjson)      # JSON 처리 패키지
library(dplyr)      # 데이터 정제 패키지
library(ggplot2)    # 그래프 패키지
library(extrafont)  # 폰트 관리 패키지
```

#02. 데이터 수집하기

1) API 접근에 필요한 URL 구성하기

요청변수로 사용할 하루 전 날짜값 만들기

오늘 날짜 가져오기

`Sys.Date()` 는 현재 사용중인 시스템의 현재 시각을 반환한다.

```
today <- Sys.Date()
today
```

▶ 출력결과

```
2019-11-13
```

날짜 계산

```
yesterday = today-1
yesterday
```

▶ 출력결과

```
2019-11-12
```

날짜 형식 지정

- `%d` : 숫자형식의 날짜
- `%m` : 숫자형식의 월
- `%b` : 약자 형식의 월 이름
- `%B` : 월에 대한 영문 이름
- `%y` : 2자리 숫자 형식
- `%Y` : 4자리 숫자 형식

```
targetDt <- format(yesterday, "%Y%m%d")
targetDt
```

▶ 출력결과

```
'20191112'
```

API키와 날짜를 조합하여 접속할 URL 구성

```
# 영화 진흥원에서 발급받은 API키
kobis_api_key = "발급받은APIKey"

# 영화진흥원 API URL
kobis_api_url = "http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.jso

# `%s`로 지정된 부분에 변수값을 치환하여 전체 주소 결정
api_url <- sprintf(kobis_api_url, kobis_api_key, targetDt)
api_url
```

▶ 출력결과

```
'http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.json?key=발급받은APIK
```

2) OpenAPI와 연동하여 데이터 수집하기

API 접속 후 데이터 수집하기

```
resp <- GET(api_url)
resp
```

▶ 출력결과

```
Response [http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.json?key=6
  Date: 2020-01-16 02:47
  Status: 200
  Content-Type: application/json;charset=utf-8
  Size: 3.66 kB
```

수집 결과를 데이터프레임으로 변환

```
# 리스트로 변환하기
resp_dat <- content(resp, as="parse", encoding="utf-8")
# 리스트에서 배열 부분만 추출하여 데이터프레임으로 변환
resp_df <- bind_rows(resp_dat$boxOfficeResult$dailyBoxOfficeList)
resp_df
```

▶ 출력결과

A tibble: 10 × 18

rnum	rank	rankInten	rankOldAndNew	movieCd	movieNm	openDt	salesAmt	salesShar
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	1	0	OLD	20181434	신의 한 수: 귀수편	2019-11-07	856972020	47.4
2	2	0	OLD	20191029	82년생 김지영	2019-10-23	345542600	19.1
3	3	0	OLD	20192083	터미네이터: 다크 페이트	2019-10-30	256206060	14.2
4	4	0	OLD	20190280	날씨의 아이	2019-10-30	78179420	4.3
5	5	1	OLD	20193801	닥터 슬립	2019-11-07	44852480	2.5
6	6	-1	OLD	20197279	아담스 패밀리	2019-11-07	40601960	2.2
7	7	1	OLD	20199950	조커	2019-10-02	25453550	1.4
8	8	1	OLD	20191589	말레피센트 2	2019-10-17	21928100	1.2
9	9	7	OLD	20196284	윤희에게	2019-11-14	15832000	0.9
10	10	4	OLD	20195002	블랙머니	2019-11-13	16500000	0.9

#03. 데이터 전처리

1) 필요한 데이터만 추출

`영화제목`과 `관람객 수` 데이터만 추출후 결과 확인

`관람객 수` 데이터가 문자열(chr)임이 확인된다. 이는 숫자가 아니므로 통계값 산출이 불가능하기 때문에 변환이 필요하다.

```
boxoffice_df <- resp_df %>% select(movieNm, audiCnt)
boxoffice_df
```

▶ 출력결과

A tibble: 10 × 2

movieNm	audiCnt
<chr>	<chr>
신의 한 수: 귀수편	105215
82년생 김지영	43843
터미네이터: 다크 페이트	31914
날씨의 아이	9213
닥터 슬립	5515
아담스 패밀리	5391
조커	3084
말레피센트 2	2758
윤희에게	1844
블랙머니	1770

2) 관람객 수 데이터 타입을 integer로 변경

```
boxoffice_df$audiCnt <- as.integer(boxoffice_df$audiCnt)
boxoffice_df
```

▶ 출력결과

A tibble: 10 × 2

movieNm	audiCnt
<chr>	<int>
신의 한 수: 귀수편	105215
82년생 김지영	43843
터미네이터: 다크 페이트	31914
날씨의 아이	9213
닥터 슬립	5515
아담스 패밀리	5391
조커	3084
말레피센트 2	2758
윤희에게	1844
블랙머니	1770

3) 컬럼이름 변경

```
영화별관람객수df <- rename(boxoffice_df, '영화제목'=movieNm, '관람객수'=audiCnt)
영화별관람객수df
```

▶ 출력결과

A tibble: 10 × 2

영화제목	관람객수
<chr>	<int>
신의 한 수: 귀수편	105215
82년생 김지영	43843
터미네이터: 다크 페이트	31914
날씨의 아이	9213
닥터 슬립	5515
아담스 패밀리	5391
조커	3084
말레피센트 2	2758
윤희에게	1844
블랙머니	1770

#03. 데이터 시각화

1) 그래프 크기 환경설정

```
options(repr.plot.width=20, repr.plot.height=10, warn=-1)
```

2) 폰트 설정

```
# 폰트 가져오기
font_import(pattern="NanumGothic.ttf")
```

▶ 출력결과

```
Importing fonts may take a few minutes, depending on the number of fonts and the speed of the system.
Continue? [y/n]  y

Scanning ttf files in C:\Windows\Fonts ...
Extracting .afm files from .ttf files...
... 생략 ...
```

```
# 폰트 로드
loadfonts(device="win")   # Windows
#loadfonts()              # Mac
```

▶ 출력결과

```
NanumGothic already registered with windowsFonts().
NanumGothicExtraBold already registered with windowsFonts().
NanumGothic Light already registered with windowsFonts().
... 생략 ...
```

```
# 폰트테이블 확인
fonts <- fonttable()

# 중복된 이름을 제거하고 출력
unique(fonts$FamilyName)
```

▶ 출력결과

```
1. 'NanumGothic'
2. 'NanumGothicExtraBold'
3. 'NanumGothic Light'
4. 'NanumBarunGothic'
... 생략 ...
```

3) 그래프 표현하기

```
ggplot(data=영화별관람객수df) +
    geom_col(aes(x=영화제목, y=관람객수, fill=영화제목)) +
    # 가로 막대 그래프로 설정
    coord_flip() +
    # 배경을 흰색으로 설정
    theme_bw() +
    # 그래프 타이틀 설정
    ggtitle(sprintf("%s 영화별 관람객 수", yesterday)) +
    # x축 제목 설정
    xlab("영화제목") +
    # y축 제목 설정
    ylab("관람객수") +
    # y축 값의 간격 및 세자리마다 콤마 적용
    scale_y_continuous(breaks=seq(0, max(영화별관람객수df$관람객수), 10000), labels=scales::comma) +
    # 각 텍스트의 색상, 크기, 각도, 글꼴 설정
    theme(plot.title=element_text(family="NanumGothic", color="#0066ff", size=25, face="bold", hjust=0.5),
          axis.title.x=element_text(family="NanumGothic", color="#999999", size=18, face="bold"),
          axis.title.y=element_text(family="NanumGothic", color="#999999", size=18, face="bold"),
          axis.text.x=element_text(family="NanumGothic", color="#000000", size=16, angle=0),
          axis.text.y=element_text(family="NanumGothic", color="#000000", size=16, angle=0)) +
    # 범례 지우기
    theme(legend.position = "none")
```

▶ 출력결과