

OpenAPI 활용: 수집에서 시각화까지

백석대학교 강윤희



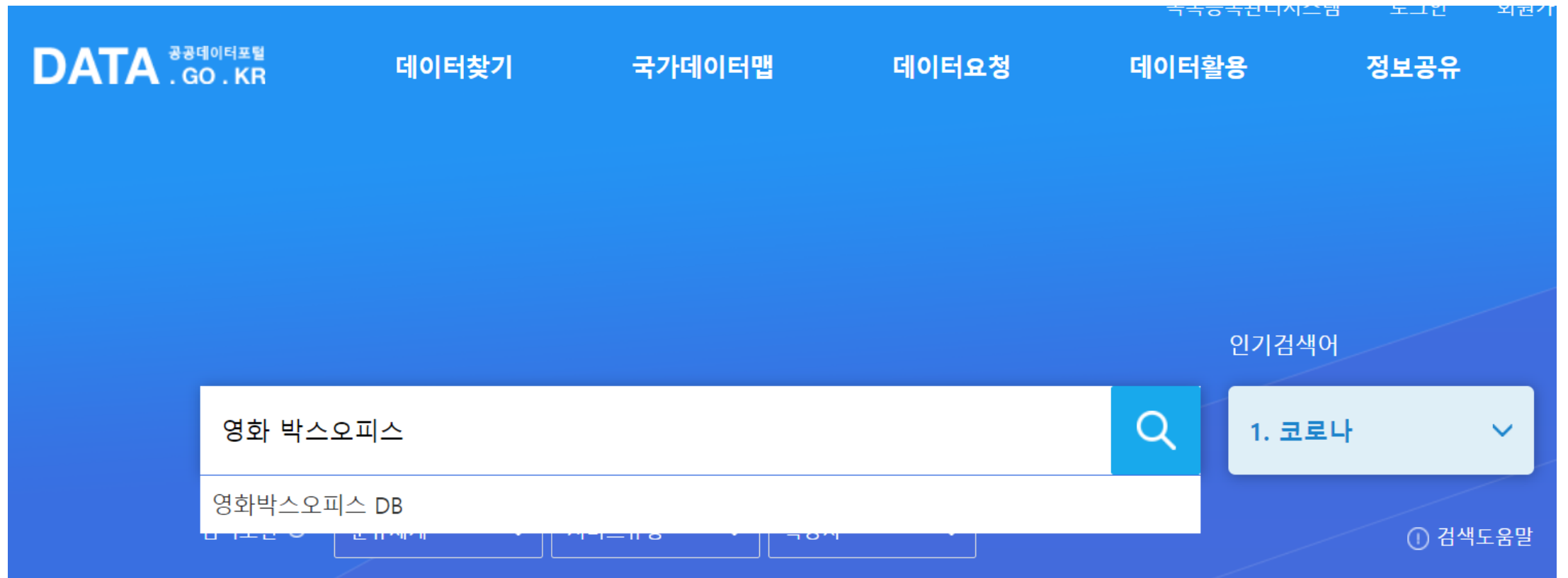
개요

- 공공개방데이터를 활용한 주제 선정에서 데이터 수집 준비
- 오픈API 요청하기 : requests
- 수집데이터 해석 JSON vs. XML : json, bs4
- 수집데이터 전처리 : pandas
- 수집데이터 시각화 하기 : matplotlib



(계획) 데이터 수집

- 주제 선정하기 (data.go.kr)



데이터 수집

• 주제 선정하기

오픈 API (26건)

문화관광 공공기관

XML 영화박스오피스 DB

전국영화관 발권데이터를 실시간 집계처리하여 일간/주간/월간 등의 박스오피스 제공정보

수정일 2014-03-20 조회수 485 활용신청 2239

문화관광 공공기관

XML 영화정보 DB

각종 영화관련 정보서비스를 위해 영화코드/영화명 등 기본적으로 필요한 영화메타정보로 홈페이지를 통해 서비스 제공

수정일 2015-07-30 조회수 299 활용신청 6901

문화관광 공공기관

XML 영화사 DB

영화작품 참여 영화사(제작사, 배급사 등) 정보

수정일 2016-03-24 조회수 87 활용신청 249



데이터 수집

XML 영화정보 DB

각종 영화관련 정보서비스를 위해 영화코드/영화명 등 기본적으로 필요한 영화메타정보로 홈페이지를 통해 서비스 제공



0

관심

OpenAPI 정보

분류체계	문화체육관광 - 문화예술	제공기관	영화진흥위원회
관리부서명	정책연구부	관리부서 전화번호	02-958-7516
보유근거	영화 및 비디오물의 진흥에 관한 법률	수집방법	
API 유형	LINK	데이터포맷	XML
활용신청	6901	데이터한계	
키워드	영화, 박스오피스, 통계		
등록	2014-07-03	수정	2014-08-14
URL	http://www.kobis.or.kr/kobisopenapi/homepg/main/main.do		
비용부과유무	무료		
이용허락범위	이용허락범위 제한 없음		



데이터 수집

- 주제 선정하기
 - 영화진흥위원회(www.kobis.or.kr) 제공 오픈 API 활용



데이터 수집

영화진흥위원회 통합전산망 오픈API 서비스 회원가입 확인 메일입니다. > 받은편지함 x

관리자 <openapimaster@kofic.or.kr>

5월 12일 (화) 오후

나에게 ▾

KOFIC 영화관입장권통합전산망

2020-05-12

홈페이지 >

영화관입장권통합전산망

[오픈API 회원 가입 확인]

강윤희님, 안녕하세요.

영화관입장권통합전산망 오픈API를 이용해주셔서 감사합니다.



- 강윤희님의 회원 가입 확인 메일 입니다. 아래 링크를 클릭하시어 인증 후 로그인이 가능합니다.

[회원 인증 처리하러 가기](#)



데이터 수집

• 데이터 수집 서비스 선정

제공 서비스

영화관입장권통합전산망이 제공하는 오픈API 서비스 모음입니다.
사용 가능한 서비스를 확인하고 서비스별 인터페이스 정보를 조회합니다.

 1 박스오피스	<ul style="list-style-type: none">▪ <u>일별 박스오피스</u>▪ 주간/주말 박스오피스
 2 공통코드조회	<ul style="list-style-type: none">▪ 공통코드 조회
 3 영화정보	<ul style="list-style-type: none">▪ 영화목록▪ 영화 상세정보
 4 영화사정보	<ul style="list-style-type: none">▪ 영화사목록▪ 영화사 상세정보
 5 영화인정보	<ul style="list-style-type: none">▪ 영화인목록▪ 영화인 상세정보



데이터 수집

• 오픈API 선정

1. REST 방식

- 기본 요청 URL : <http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.xml> (또는 .json)
- 요청 parameter : 3번항의 요청 인터페이스 정보를 참조하여 GET 방식으로 호출

2. SOAP 방식

- 요청 URL : <http://www.kobis.or.kr/kobisopenapi/webservice/soap/boxoffice>
- WSDL URL : <http://www.kobis.or.kr/kobisopenapi/webservice/soap/boxoffice?wsdl>
- Operation : searchDailyBoxOfficeList

3. 인터페이스

- 요청 인터페이스

요청 변수	값	설명
key	문자열(필수)	발급받은키 값을 입력합니다.
targetDt	문자열(필수)	조회하고자 하는 날짜를 yyyyymmdd 형식으로 입력합니다.
itemPerPage	문자열	결과 ROW 의 개수를 지정합니다.(default : "10", 최대 : "10")
multiMovieYn	문자열	다양성 영화/상업영화를 구분지어 조회할 수 있습니다. "Y" : 다양성 영화 "N" : 상업영화 (default : 전체)
repNationCd	문자열	한국/외국 영화별로 조회할 수 있습니다. "K" : 한국영화 "F" : 외국영화 (default : 전체)
wideAreaCd	문자열	상영지역별로 조회할 수 있으며, 지역코드는 공통코드 조회 서비스에서 "0105000000"로서 조회된 지역코드입니다. (default : 전체)



데이터 수집

SOAP VS REST: 핵심적인 차이들

차이점	SOAP	REST
유형	프로토콜	아키텍처 스타일
기능	기능 위주: 구조화된 정보 전송	데이터 위주: 데이터를 위해서 리소스에 접근
데이터 포맷	XML만 사용	일반 텍스트, HTML, XML, JSON 등 다양한 포맷을 허용
보안	WS-Security와 SSL을 지원	SSL과 HTTPS를 지원
대역폭	상대적으로 더 많은 리소스와 대역폭이 필요	상대적으로 리소스가 적게 필요하고, 무게가 가벼움



(XML) 데이터 수집

- XML 자료형식으로 데이터 수집
 - (선행) requests와 BeautifulSoup4 패키지를 설치

pip3 install requests bs4

```
from bs4 import BeautifulSoup
import requests as req

|
basic_url = "http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.xml?"
mDate=20200401

# key는 http://www.kobis.or.kr/kobisopenapi 에서 발급받아 사용하
mURL = basic_url + "key=" + key + "&" + 'targetDt=' + str(mDate)
```



데이터 수집

- BeautifulSoup
 - HTML이나 XML 자료를 해석하기 위해 사용하는 라이브러리
- Requests
 - HTTP 라이브러리
 - 주어진 URL((Uniform Resource Locator)의 서비스를 요청한 후 자료를 얻기 위해 사용
 - GET, POST, PUT, DELETE 메소드를 지원함



데이터 수집

- requests.get(url)
 - url 요청 결과인 Response 객체를 리턴

```
resp = req.get(mURL)
```

```
soup = BeautifulSoup(resp.content, "xml")  
#soup
```

```
if (resp.status_code == requests.codes.ok):  
    print ("성공적으로 데이터를 수집함")
```

성공적으로 데이터를 수집함



데이터 수집

mURL

'http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.xml?key=daa1451451ff54f42e1f0f50d2256f8c&targetDt=20200401'

```
▼<boxOfficeResult>
  <boxofficeType>일별 박스오피스</boxofficeType>
  <showRange>20200401~20200401</showRange>
  ▼<dailyBoxOfficeList>
    ▼<dailyBoxOffice>
      <rnum>1</rnum>
      <rank>1</rank>
      <rankInten>0</rankInten>
      <rankOldAndNew>NEW</rankOldAndNew>
      <movieCd>20205262</movieCd>
      <movieNm>엽문4: 더 파이널</movieNm>
      <openDt>2020-04-01</openDt>
      <salesAmt>48850700</salesAmt>
      <salesShare>19.3</salesShare>
      <salesInten>48850700</salesInten>
      <salesChange>100</salesChange>
      <salesAcc>48850700</salesAcc>
      <audiCnt>6165</audiCnt>
      <audiInten>6165</audiInten>
      <audiChange>100</audiChange>
      <audiAcc>6165</audiAcc>
      <scrnCnt>175</scrnCnt>
      <showCnt>509</showCnt>
    </dailyBoxOffice>
    ▼<dailyBoxOffice>
      <rnum>2</rnum>
      <rank>2</rank>
      <rankInten>-1</rankInten>
      <rankOldAndNew>OLD</rankOldAndNew>
      <movieCd>20204107</movieCd>
      <movieNm>주디</movieNm>
      <openDt>2020-03-25</openDt>
      <salesAmt>31798560</salesAmt>
      <salesShare>12.6</salesShare>
      <salesInten>-17658360</salesInten>
      <salesChange>-35.7</salesChange>
      <salesAcc>444609280</salesAcc>
      <audiCnt>4105</audiCnt>
      <audiInten>-2300</audiInten>
      <audiChange>-35.9</audiChange>
      <audiAcc>55718</audiAcc>
      <scrnCnt>346</scrnCnt>
      <showCnt>730</showCnt>
    </dailyBoxOffice>
  </dailyBoxOfficeList>
</boxOfficeResult>
```



데이터 수집

mURL

'http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.xml?key=dec1451451ff54f43e1f0f50d3256f86&targetDt=20200401'

```
titles = soup.find_all('movieNm')
```

```
for i in range(0, len(titles)):
    print(titles[i].get_text())
```

엽문4: 더 파이널

주디

1917

인비저블맨

n번째 이별 중

스케어리 스토리: 어둠의 속삭임

날씨의 아이

라라랜드

너의 이름은.

다크 워터스



(JSON) 데이터 수집

- KOBIS 박스오피스 openAPI 사용하기 요청

```
import json
import requests
import pandas as pd
```

```
mDate=20200401
mURL = 'http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.json?key=
```

```
res = requests.get(mURL)
text = res.text
```

mURL

```
'http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.json?key=
&targetDt=20200401'
```



데이터 수집

- requests.get(url)

```
{
  "boxOfficeResult": {
    "boxofficeType": "일별 박스오피스",
    "showRange": "20200401~20200401",
    "dailyBoxOfficeList": [
      {
        "rnum": "1",
        "rank": "1",
        "rankInten": "0",
        "rankOldAndNew": "NEW",
        "movieCd": "20205262",
        "movieNm": "엽문4: 더 파이널",
        "openDt": "2020-04-01",
        "salesAmt": "48850700",
        "salesShare": "19.3",
        "salesInten": "48850700",
        "salesChange": "100",
        "salesAcc": "48850700",
        "audiCnt": "6165",
        "audiInten": "6165",
        "audiChange": "100",
        "audiAcc": "6165",
        "scrnCnt": "175",
        "showCnt": "509"
      },
      {
        "rnum": "2",
        "rank": "2",
        "rankInten": "-1",
        "rankOldAndNew": "OLD",
        "movieCd": "20204107",
        "movieNm": "주디",
        "openDt": "2020-03-25",
        "salesAmt": "31798560",
        "salesShare": "12.6",
        "salesInten": "-17658360",
        "salesChange": "-35.7",
        "salesAcc": "444609280",
        "audiCnt": "4105",
        "audiInten": "2300",
        "audiChange": "35.9",
        "audiAcc": "55718",
        "scrnCnt": "346",
        "showCnt": "730"
      },
      {
        "rnum": "3",
        "rank": "3",
        "rankInten": "-1",
        "rankOldAndNew": "OLD",
        "movieCd": "20192399",
        "movieNm": "1917",
        "openDt": "2020-02-19",
        "salesAmt": "30146760",
        "salesShare": "11.9",
        "salesInten": "-11392000",
        "salesChange": "-27.4",
        "salesAcc": "6548462220",
        "audiCnt": "3745",
        "audiInten": "-1593",
        "audiChange": "-29.8",
        "audiAcc": "718672",
        "scrnCnt": "341",
        "showCnt": "589"
      },
      {
        "rnum": "4",
        "rank": "4",
        "rankInten": "-1",
        "rankOldAndNew": "OLD",
        "movieCd": "20199842",
        "movieNm": "인비저블맨",
        "openDt": "2020-02-26",
        "salesAmt": "19394760",
        "salesShare": "7.7",
        "salesInten": "-11793160",
        "salesChange": "-37.8",
        "salesAcc": "4750664820",
        "audiCnt": "2373",
        "audiInten": "-1558",
        "audiChange": "-39.6",
        "audiAcc": "545382",
        "scrnCnt": "296",
        "showCnt": "489"
      },
      {
        "rnum": "5",
        "rank": "5",
        "rankInten": "0",
        "rankOldAndNew": "NEW",
        "movieCd": "20206446",
        "movieNm": "n번째 이별 중",
        "openDt": "2020-04-01",
        "salesAmt": "17246100",
        "salesShare": "6.8",
        "salesInten": "17246100",
        "salesChange": "100",
        "salesAcc": "17950100",
        "audiCnt": "2148",
        "audiInten": "2148",
        "audiChange": "100",
        "audiAcc": "2212",
        "scrnCnt": "157",
        "showCnt": "335"
      },
      {
        "rnum": "6",
        "rank": "6",
        "rankInten": "-1",
        "rankOldAndNew": "OLD",
        "movieCd": "20205708",
        "movieNm": "스케어리 스토리: 어둠의 속삭임",
        "openDt": "2020-03-25",
        "salesAmt": "15915760",
        "salesShare": "6.3",
        "salesInten": "-692240",
        "salesChange": "-4.2",
        "salesAcc": "264162260",
        "audiCnt": "1893",
        "audiInten": "-104",
        "audiChange": "-5.2",
        "audiAcc": "32067",
        "scrnCnt": "209",
        "showCnt": "416"
      },
      {
        "rnum": "7",
        "rank": "7",
        "rankInten": "2",
        "rankOldAndNew": "OLD",
        "movieCd": "20190280",
        "movieNm": "날씨의 아이",
        "openDt": "2019-10-30",
        "salesAmt": "14284000",
        "salesShare": "5.7",
        "salesInten": "4306000",
        "salesChange": "43.2",
        "salesAcc": "5563697050",
        "audiCnt": "1439",
        "audiInten": "436",
        "audiChange": "43.5",
        "audiAcc": "671550",
        "scrnCnt": "19",
        "showCnt": "34"
      },
      {
        "rnum": "8",
        "rank": "8",
        "rankInten": "-2",
        "rankOldAndNew": "OLD",
        "movieCd": "20167904",
        "movieNm": "라라랜드",
        "openDt": "2016-12-07",
        "salesAmt": "8691000",
        "salesShare": "3.4",
        "salesInten": "-3310600",
        "salesChange": "-27.6",
        "salesAcc": "30247277148",
        "audiCnt": "1091",
        "audiInten": "-453",
        "audiChange": "-29.3",
        "audiAcc": "3627999",
        "scrnCnt": "85",
        "showCnt": "202"
      },
      {
        "rnum": "9",
        "rank": "9",
        "rankInten": "2",
        "rankOldAndNew": "OLD",
        "movieCd": "20161872",
        "movieNm": "너의 이름은.",
        "openDt": "2017-01-04",
        "salesAmt": "8918000",
        "salesShare": "3.5",
        "salesInten": "1100500",
        "salesChange": "14.1",
        "salesAcc": "29983900256",
        "audiCnt": "892",
        "audiInten": "110",
        "audiChange": "14.1",
        "audiAcc": "3723986",
        "scrnCnt": "15",
        "showCnt": "23"
      },
      {
        "rnum": "10",
        "rank": "10",
        "rankInten": "-6",
        "rankOldAndNew": "OLD",
        "movieCd": "20204791",
        "movieNm": "다크 워터스 ",
        "openDt": "2020-03-11",
        "salesAmt": "6237920",
        "salesShare": "2.5",
        "salesInten": "-11799120",
        "salesChange": "-65.4",
        "salesAcc": "1066370760",
        "audiCnt": "806",
        "audiInten": "-1510",
        "audiChange": "-65.2",
        "audiAcc": "123686",
        "scrnCnt": "143",
        "showCnt": "188"
      }
    ]
  }
}
```



데이터 살펴보기

- Panda 데이터 프레임으로 자료 읽어오기

```
res = requests.get(mURL)
text = res.text
```

```
movieData = json.loads(text)
movieLen = len(movieData['boxOfficeResult']['dailyBoxOfficeList'])
movieDF = pd.DataFrame()
movieDF = movieDF.append( {"title": "", "cnt": "", "salesShare": "" }, ignore_index=True, sort=False )
```

```
for i in range(movieLen):
    movieDF.loc[i, "title"] = movieData['boxOfficeResult']['dailyBoxOfficeList'][i]['movieNm']
    movieDF.loc[i, "cnt"] = movieData['boxOfficeResult']['dailyBoxOfficeList'][i]['salesAmt']
    movieDF.loc[i, "salesShare"] = float(movieData['boxOfficeResult']['dailyBoxOfficeList'][i]['salesShare'])
```



데이터 살펴보기

- 데이터프레임 읽기

```
movieDF.head()
```

	cnt	salesShare	title
0	48850700	19.3	엽문4: 더 파이널
1	31798560	12.6	주디
2	30146760	11.9	1917
3	19394760	7.7	인비저블맨
4	17246100	6.8	n번째 이별 중



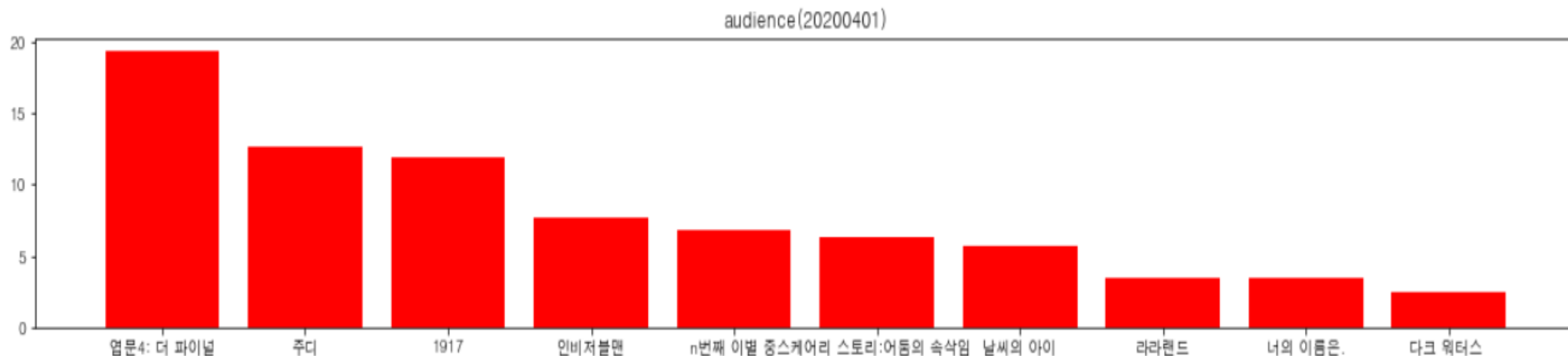
데이터 살펴보기

- 자료 시각화하기

```
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm
fm.get_fontconfig_fonts()

font_location = 'C:/Windows/Fonts/H2GTRM.ttf' # For Windows
font_name = fm.FontProperties(fname=font_location).get_name()

plt.rc('font', family=font_name)
plt.rcParams["figure.figsize"] = (20,3)
plt.title('audience' + '(' + str(mDate) + ')')
plt.bar(movieDF['title'], movieDF['salesShare'], color='r')
plt.show()
```



데이터 살펴보기

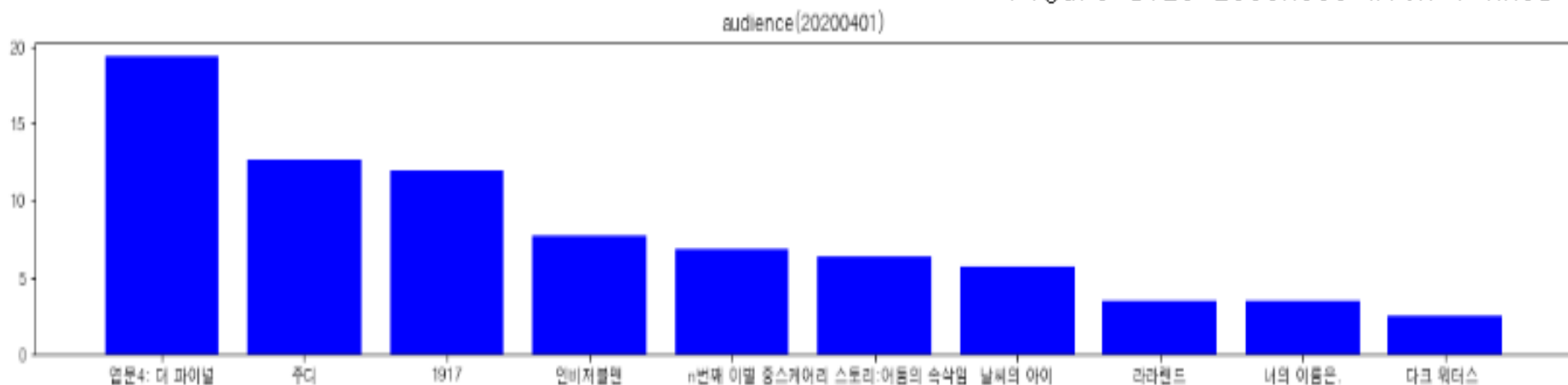
- 자료 시각화하기

```
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm
fm.get_fontconfig_fonts()

font_location = 'C:/Windows/Fonts/H2GTRM.ttf' # For Windows
font_name = fm.FontProperties(fname=font_location).get_name()

plt.rc('font', family=font_name)
plt.rcParams["figure.figsize"] = (20,3)
plt.title('audience' + '(' + str(mDate) + ')')
plt.bar(movieDF['title'], movieDF['salesShare'], color='r')
plt.show()
```

<Figure size 2000x300 with 1 Axes>



데이터 저장하기

- CSV 로 저장하기

```
movieDF.to_csv('movie'+str(mDate)+'.csv', index=False)
```

```
1 cnt,salesShare,title
2 48850700,19.3,엽문4: 더 파이널
3 31798560,12.6,주디
4 30146760,11.9,1917
5 19394760,7.7,인비저블맨
6 17246100,6.8,n번째 이별 중
7 15915760,6.3,스케어리 스토리:어둠의 속삭임
8 14284000,5.7,날씨의 아이
9 8691000,3.4,라라랜드
10 8918000,3.5,너의 이름은.
11 6237920,2.5,다크 워터스
```



참고자료

데이터 수집

- 오픈API 선정
 - SOAP
 - SOAP stands for **S**imple **O**bject **A**ccess **P**rotocol
 - SOAP is an application communication protocol
 - SOAP is a format for sending and receiving messages
 - SOAP is platform independent
 - SOAP is based on XML
 - SOAP is a W3C recommendation