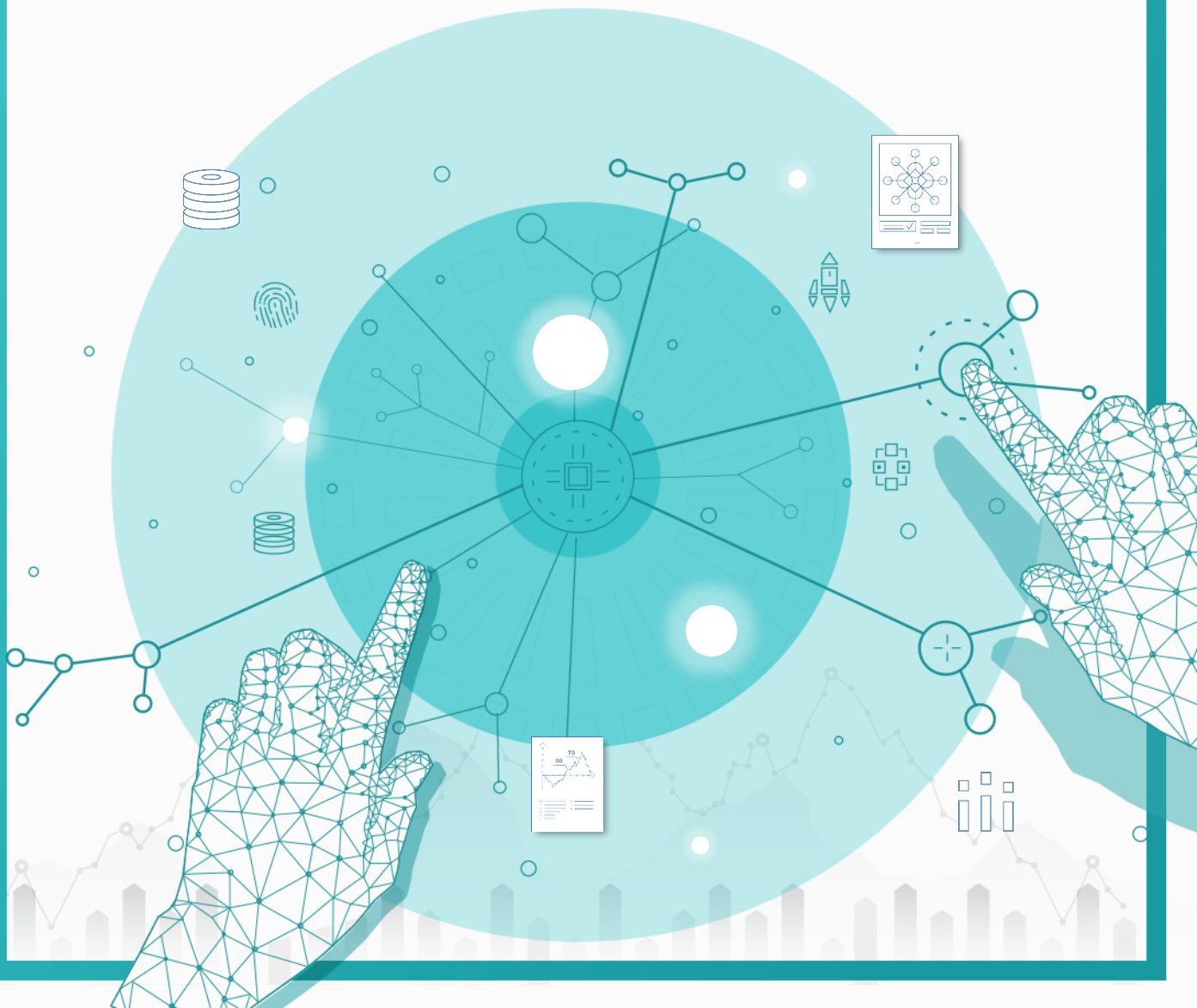
한국기술교육대학교
온라인평생교육원

파이썬 라이브러리로 하는 데이터 분석과 시각화

파이썬을 활용한 데이터 분석 실습



파이썬을 활용한 데이터 분석 실습

학습 목표

1. 파이썬의 자료형, 기본 문법을 활용하여 데이터를 분석할 수 있다.
2. 파이썬의 정규표현식을 활용하여 데이터를 분석할 수 있다.
3. 실제 데이터를 직접 분석하고 활용할 수 있다.

학습 내용

1. 파이썬의 자료형, 기본 문법을 활용한 데이터 분석
2. 파이썬의 정규표현식을 활용한 데이터 분석
3. 영화 장르 데이터 분석 실습

1. 파이썬의 자료형, 기본 문법을 활용한 데이터 분석

1) 간단한 텍스트 데이터 분석하기

(1) 일부 데이터 추출

- 실제 데이터에서 원하는 데이터만 출력하기
  - 데이터의 구조를 파악한 뒤 패턴을 찾음
  - 불필요한 데이터를 하나씩 제거함

(예) 1,2,3,4,5에서 콤마를 제거한 12345 데이터만 추출하기

```
a = "1,2,3,4,5"
print(a)
```

```
1,2,3,4,5
```

- 반복문, 조건문 활용

```
for i in a:
    if i != ',':
        print(i,end='')
```

```
12345
```

1. 파이썬의 자료형, 기본 문법을 활용한 데이터 분석

1) 간단한 텍스트 데이터 분석하기

(1) 일부 데이터 추출

- 문자열 함수를 활용한 특정 문자열 제거

```
print(a.replace(',',''))
```

```
12345
```

- 문자열 자료형의 범위 선택 연산 활용

```
print(a[::2])
```

```
12345
```

1. 파이썬의 자료형, 기본 문법을 활용한 데이터 분석

1) 패턴이 있는 여러 줄의 텍스트 데이터 분석하기

(1) 패턴이 있는 일부 데이터 추출

(예) 복잡한 데이터에서 이름(홍길동, 이길동, 박길동)만 추출하기

- 데이터 구조 파악
  - 이름 뒤엔 콜론(:)이 있음
  - 각 데이터는 엔터(₩n)로 구분되어 있음
  - 데이터 간에 공백이 있음
  - 기타 등등

```
b = """
    홍길동: 1234, Gil-Dong Hong,
    이길동: 3915, Gil-Dong Lee,
    박길동: 0000, Gil-Dong Park
"""
```

- 데이터 좌우 공백 제거

```
b = b.strip()
print(b)
```

```
홍길동: 1234, Gil-Dong Hong,
  이길동: 3915, Gil-Dong Lee,
  박길동: 0000, Gil-Dong Park
```

1) 패턴이 있는 여러 줄의 텍스트 데이터 분석하기

(1) 패턴이 있는 일부 데이터 추출

(예) 복잡한 데이터에서 이름(홍길동, 이길동, 박길동)만 추출하기

- 패턴을 찾아 데이터 작게 분리

```
b = b.split('\n')
print(b)
```

```
['홍길동: 1234, Gil-Dong Hong,', '    이길동: 3915, Gil-Dong Lee,',
```

- 반복문을 활용해 분리한 데이터에서 원하는 데이터 추출

```
for i in b:
    i = i.split(':')
    i = i[0].strip()
    print(i)
```

```
홍길동
이길동
박길동
```

2) 홈페이지 소스와 같이 복잡한 텍스트 데이터 분석하기

(1) 복잡한 텍스트에서 일부 데이터 추출

(예) 홈페이지 소스에서 연관 SNS 사이트 주소만 추출

```
<div class="select_div">
    <span data-i18n="footer.relationSites">연관사이트</span>
    <ul>


    </ul>
</div>
<ul class="list_sns">
    <li><a href="https://www.facebook.com/ekoreatech" target="_blank"><img src="/res/lms/img/common/btn_footer_sns1.png" alt="페이스북"></a></li>
    <li><a href="https://blog.naver.com/e-koreatech" target="_blank"><img src="/res/lms/img/common/btn_footer_sns2.png" alt="블로그"></a></li>
    <li><a href="https://www.youtube.com/user/koreatecholei" target="_blank"><img src="/res/lms/img/common/btn_footer_sns3.png" alt="유트브"></a></li>
</ul>
```

[실제 홈페이지 소스]

```
c = """
    <div class="select_div">
      <span data-i18n="footer.relationSites">연관사이트</span>
      <ul>


      </ul>
    </div>
    <ul class="list_sns">
      <li><a href="https://www.facebook.com/ekoreatech" target="_blank"><img src="/res/lms/img/common/btn_footer_sns1.png" alt="페이스북"></a></li>
      <li><a href="https://blog.naver.com/e-koreatech" target="_blank"><img src="/res/lms/img/common/btn_footer_sns2.png" alt="블로그"></a></li>
      <li><a href="https://www.youtube.com/user/koreatecholei" target="_blank"><img src="/res/lms/img/common/btn_footer_sns3.png" alt="유트브"></a></li>
    </ul>
"""
```

[파이썬으로 복사, 붙여넣기한 코드]

2) 홈페이지 소스와 같이 복잡한 텍스트 데이터 분석하기

(1) 복잡한 텍스트에서 일부 데이터 추출

- 데이터 구조 파악하기

  - 추출할 데이터는 list_sns라는 텍스트 아래에 있음

```
<ul class="list_sns">
    <li><a href="https://www.facebook.com/ekoreatech" target="_blank
    <li><a href="https://blog.naver.com/e-koreatech" target="_blank"
    <li><a href="https://www.youtube.com/user/koreatecholei" target=
</ul>
```

  - 연관 사이트는 li 태그로 구분되어 있음

```
    <li><a href="https://www.facebook.com/ekoreatech" target="_blank"
    <li><a href="https://blog.naver.com/e-koreatech" target="_blank">
    <li><a href="https://www.youtube.com/user/koreatecholei" target="
</ul>
```

  - 각 연관 사이트 이름은 alt로 구분되어 있음

```
/res/lms/img/common/btn_footer_sns1.png" alt="페이스북"></a></li>
res/lms/img/common/btn_footer_sns2.png" alt="블로그"></a></li>
g src="/res/lms/img/common/btn_footer_sns3.png" alt="유트브"></a></li>
```

  - 링크는 href 태그의 큰따옴표(“”) 안에 있음

```
    <li><a href="https://www.facebook.com/ekoreatech" target="_blank"
    <li><a href="https://blog.naver.com/e-koreatech" target="_blank">
    <li><a href="https://www.youtube.com/user/koreatecholei" target="
</ul>
```

  - 기타 등등

2) 홈페이지 소스와 같이 복잡한 텍스트 데이터 분석하기

(1) 복잡한 텍스트에서 일부 데이터 추출

- 데이터 작게 분리하기(1)

```
print(c.find('list_sns'))
```

```
169
```

```
c = c[169:]
print(c)
```

```
list_sns">
        <li><a href="https://www.facebook.com/ekoreatech" ta
        <li><a href="https://blog.naver.com/e-koreatech" targe
        <li><a href="https://www.youtube.com/user/koreatech
    </ul>
```

- 데이터 작게 분리하기(2)

```
c = c.split('</li>')
print(c)
```

```
['list_sns">₩n        <li><a href="https://www.facebook.com
ef="https://blog.naver.com/e-koreatech" target="_blank"><in
r/koreatecholei" target="_blank"><img src="/res/lms/img/com
```

2) 홈페이지 소스와 같이 복잡한 텍스트 데이터 분석하기

(1) 복잡한 텍스트에서 일부 데이터 추출

- 일부 데이터에서 원하는 데이터 추출하기
  - 첫 번째 데이터에서 URL 추출

```
print(c[0])
```

```
list_sns">
        <li><a href="https://www.facebook.com/ekoreatech" target
```

```
url = c[0].split('" target')[0]
print(url)
```

```
list_sns">
        <li><a href="https://www.facebook.com/ekoreatech
```

```
url = url.split('href="')[1]
print(url)
```

```
https://www.facebook.com/ekoreatech
```

2) 홈페이지 소스와 같이 복잡한 텍스트 데이터 분석하기

(1) 복잡한 텍스트에서 일부 데이터 추출

- 일부 데이터에서 원하는 데이터 추출하기
  - 첫 번째 데이터에서 사이트 이름 추출

```
print(c[0])
```

```
list_sns">
        <li><a href="https://www.facebook.com/ekoreatech" targ
```

```
name = c[0].split('alt="')[1]
print(name)
```

```
페이스북"></a>
```

```
name = name[:name.find('"')]
print(name)
```

```
페이스북
```

2) 홈페이지 소스와 같이 복잡한 텍스트 데이터 분석하기

(1) 복잡한 텍스트에서 일부 데이터 추출

- 반복문을 활용한 데이터 정리

```
for i in range(len(c)-1):
    url = c[i].split('" target')[0]
    url = url.split('href="')[1]
    print(url)
    name = c[i].split('alt="')[1]
    name = name[:name.find('"')]
    print(name)
```

```
https://www.facebook.com/ekoreatech
페이스북
https://blog.naver.com/e-koreatech
블로그
https://www.youtube.com/user/koreatecholei
유트브
```

2. 파이썬의 정규표현식을 활용한 데이터 분석

1) 정규표현식의 개념

(1) 정규표현식이란?

- 특정한 규칙을 가진 문자열을 표현하기 위해 사용하는 형식
- 주로 문자열의 검색 및 치환에 활용함
- 다양한 문법을 제공함

  예) 패턴을 찾아 문자열 검색, 치환

(2) 정규표현식 문법 예시

- . : 1개 문자와 일치
- [] : [,] 사이의 문자 중 하나 선택
  - [abc]d → ab, bd, cd 포함
  - [a-zA-Z] → 알파벳 모두 포함
  - [0-9] → 숫자 모두 포함
- * : 0개 이상의 문자 포함
  - a*b → b, aab, ab, aaaaab 등
- {m,n} : m회 이상 n회 이하
  - a{1,3}b → ab, aab, aaab 포함
- 그 외
  - ?, +, (), ^ 등 다양한 규칙이 존재함

2) 파이썬에서의 정규표현식

(1) re 모듈(Regular Expression) 제공

- 콤마를 제거한 데이터만 추출하기

(예) 1,2,3,4,5에서 12345만 추출

```
import re

a = "1,2,3,4,5"

pattern = re.compile('[^0-9]')
result = re.sub(pattern,'',a)
print(result)
```

```
12345
```

- compile: 패턴을 적용한 객체를 돌려주는 함수
- sub: 특정 문자에서 패턴과 일치하는 것들을 교체해주는 함수

2) 파이썬에서의 정규표현식

(1) re 모듈(Regular Expression) 제공

- 일부 데이터만 추출하기

(예) 복잡한 데이터에서 이름만 추출

- 한글이 아닌 것들을 모두 지움

```
import re

b = """
    홍길동: 1234, Gil-Dong Hong,
    이길동: 3915, Gil-Dong Lee,
    박길동: 0000, Gil-Dong Park
"""
pattern = re.compile('[^가-힣]')
result = re.sub(pattern,'',b)
print(result)
```

```
홍길동이길동박길동
```

2. 파이썬의 정규표현식을 활용한 데이터 분석

2) 파이썬에서의 정규표현식

(1) re 모듈(Regular Expression) 제공

- 일부 데이터만 추출하기

(예) 홈페이지 소스에서 연관 SNS 사이트 주소만 추출

- search: 특정 패턴을 찾아주는 함수
- group: 일치한 문자들을 반환하는 함수

```
import re
c = """
        <div class="select_div">
            <span data-i18n="footer.relationSites">연관사이트</span>
            <ul>


            </ul>
        </div>
        <ul class="list_sns">
            <li><a href="https://www.facebook.com/ekoreatech" target="_blank"><img src="/r
            <li><a href="https://blog.naver.com/e-koreatech" target="_blank"><img src="/re
            <li><a href="https://www.youtube.com/user/koreatecholei" target="_blank"><img
        </ul>
"""

pattern = re.compile('https://.*" target')
result = pattern.search(c)
print(result.group())

pattern = re.compile('[a-z:/.]*')
result = pattern.search(result.group())
print(result.group())
```

```
https://www.facebook.com/ekoreatech" target
https://www.facebook.com/ekoreatech
```


```
pattern = re.compile('https://.*" target')
result = pattern.search(c)
print(result.group())

pattern = re.compile('[a-z:/.]*')
result = pattern.search(result.group())
print(result.group())
```

```
https://www.facebook.com/ekoreatech" target
https://www.facebook.com/ekoreatech
```

3. 영화 장르 데이터 분석 실습

미네소타 대학의 GroupLens라는 연구 프로젝트에서 제공하는
영화 장르 데이터셋을 활용하여 데이터 분석 실습 진행

(약 10,000여 개 영화의 제목, 장르에 대한 데이터를 CSV 파일로 제공)

* 출처: https://grouplens.org/datasets/movielens/

1) 데이터 구조 파악하기

- CSV 파일 값이 콤마(,)로 구분되어 있음
- 하나의 영화에 여러 개의 장르가 있을 수 있음
- 영화 아이디, 제목, 장르로 구분되어 있음
- 각 영화의 장르가 2개 이상인 경우 '|'로 구분되어 있음
- 영화 제목마다 뒤에 괄호로 영화의 연도가 적혀 있음

	A	B	C	D	E	F	G
1	movieId	title	genres				
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy				
3	2	Jumanji (1995)	Adventure|Children|Fantasy				
4	3	Grumpier Old Men	Comedy|Romance				
5	4	Waiting to Exhale	Comedy|Drama|Romance				
6	5	Father of the Bride	Comedy				
7	6	Heat (1995)	Action|Crime|Thriller				
8	7	Sabrina (1995)	Comedy|Romance				
9	8	Tom and Huck (19	Adventure|Children				
10	9	Sudden Death (19	Action				
11	10	GoldenEye (1995)	Action|Adventure|Thriller				
12	11	American Presiden	Comedy|Drama|Romance				

3. 영화 장르 데이터 분석 실습

2) 영화 장르의 종류 파악하기

(1) 영화 아이디, 제목, 장르 확인하기

- CSV 모듈을 통해 영화 CSV 파일 로드
  - 영화 아이디, 제목, 장르가 순서대로 저장된 것을 확인할 수 있음

```
import csv

f = open('ml-latest-small/movies.csv','r',encoding='utf8')
data = csv.reader(f)
print(next(data))
print(next(data))
print(next(data))
print(next(data))
print(next(data))
print(next(data))
```

```
['movieId', 'title', 'genres']
['1', 'Toy Story (1995)', 'Adventure|Animation|Children|Comedy|Fantasy']
['2', 'Jumanji (1995)', 'Adventure|Children|Fantasy']
['3', 'Grumpier Old Men (1995)', 'Comedy|Romance']
['4', 'Waiting to Exhale (1995)', 'Comedy|Drama|Romance']
['5', 'Father of the Bride Part II (1995)', 'Comedy']
```

3. 영화 장르 데이터 분석 실습

2) 영화 장르의 종류 파악하기

(2) 전체 영화의 총 장르 개수

- 새로운 리스트에 각 영화의 장르 분리
  - split 함수를 활용하여 '|'을 기준으로 데이터 분리 및 저장
- 전체 영화의 총 장르 개수는 22,084개

```
import csv

f = open('ml-latest-small/movies.csv','r',encoding='utf8')
data = csv.reader(f)
next(data)
result = []

for i in data:
    result.extend(i[2].split('|'))
print(len(result))
print(result)
```

```
22084
['Adventure', 'Animation', 'Children', 'Comedy', 'Fantasy', 'Adventure', 'Children',
medy', 'Romance', 'Adventure', 'Children', 'Action', 'Action', 'Adventure', 'Thriller'
```

3. 영화 장르 데이터 분석 실습

2) 영화 장르의 종류 파악하기

(3) 집합 자료형으로 데이터 변환

- 파이썬의 자료형 특징 활용
  - 콤마로 구분되어 행별로 데이터를 쉽게 불러올 수 있음

➡ no genres listed를 포함하여 총 20개의 장르가 있음을 확인

```
genre = list(set(result))
print(len(genre))
print(genre)
```

```
20
['Adventure', 'Drama', 'Horror', 'Sci-Fi', 'Mystery', 'Comedy',
antasy', 'Western', 'Action']
```

3. 영화 장르 데이터 분석 실습

3) 장르별 영화 개수 파악하기

(1) 장르 개수 카운트

- 파이썬 문법 활용
  - 영화마다 해당 장르가 포함되는 경우: 장르 +1
- 딕셔너리 자료형 활용
  - 해당 장르가 딕셔너리에 없는 경우: 1로 생성
  - 해당 자료가 딕셔너리에 있는 경우: +1

```
count = {}
for i in result:
    if count.get(i):
        count[i] +=1
    else:
        count[i] = 1
print(count)
```

```
{'Adventure': 1263, 'Animation': 611, 'Children': 664, 'Come
```

4) 그 외 분석할 만한 것들

- 장르가 가장 다양한 영화는?
- 연도별 가장 많이 개봉된 영화의 장르는?
- 영화별 장르의 개수 분포는?
- 기타 등등