

강원대학교  
AI 소프트웨어학과

---

# 데이터 전처리

## - 데이터저장 -

---

## 데이터 저장

- 효율적이고 안정적인 액세스 및 검색을 보장하기 위해 다양한 형식의 데이터를 디지털 정보를 저장하는 프로세스
- 사용 및 이해가 쉬움
- 다양한 프로그래밍 언어 및 플랫폼과의 호환성이 좋음
- 빠른 처리 속도
- 빅데이터 프로그램에 적합
- 실시간 데이터 처리를 빠르게 할 수 있음

- CSV

- 텍스트 기반이며 사람이 읽을 수 있음
- 각 줄이 쉼표로 구분된 값으로 데이터 레코드를 나타내는 간단한 형식

- Pickle

- Python 특정 바이너리 직렬화와 역직렬화 할 수 있음(무손실 압축)
- 파이썬에서 사용하는 딕셔너리, 리스트, 데이터프레임 등의 자료형을 변환 없이 그대로 읽거나 쓰는 모듈(Python에 특화)

- HDF5(Hierarchical Data Format version 5)

- 대량의 수치 데이터를 저장하도록 설계된 바이너리 파일 형식(무손실 압축)
- 메타데이터를 포함한 복잡한 데이터 유형 및 계층을 지원함(C, C++, Java, ...)

- JSON (JavaScript Object Notation)

- 텍스트 기반의 가벼운 형식으로 사람이 읽고 쓸 수 있음
- 데이터는 Python의 사전과 같은 키-값 쌍

## Pickle

- 효율적이고 안정적인 액세스 및 검색을 보장하기 위해 다양한 형식의 데이터를 디지털 정보를 저장하는 프로세스

python=3.9.12

```
import pickle  
data1 = {'A': [1, 2, 3], 'B': [4, 5, 6]}  
data1 = pd.DataFrame(data1)
```

```
with open("data.pickle", "wb") as file:  
    pickle.dump(data1, file)
```

→ wb : 바이너리 쓰기(저장)

```
with open("data.pickle", "rb") as file:  
    data = pickle.load(file)
```

→ rb: 바이너리 읽기(불러오기)

data

## Pickle

- 여러 개의 데이터프레임을 하나의 pickle파일에 저장해보자

```
import pickle
data1 = {'A': [1, 2, 3], 'B': [4, 5, 6]}
data2 = {'X': ['apple', 'banana', 'cherry'], 'Y': ['dog', 'elephant', 'fox']}
```

```
df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)
```

```
dataframes = [df1, df2]
```

```
with open("dataframes.pickle", "wb") as file:
    pickle.dump(dataframes, file)
```

```
with open("dataframes.pickle", "rb") as file:
    data = pickle.load(file)
```

```
data[0]
data[1]
```

## HDF5(Hierarchical Data Format version 5)

- 대용량 데이터를 효율적으로 저장하고 검색할 수 있도록 설계됨(다양한 프로그래밍 언어에 사용 가능)
- 다차원 배열의 슬라이싱, 필터링 및 부분적으로 읽기와 같은 작업을 빠르게 처리할 수 있음

```
import tables
```

```
data1 = {'A': [1, 2, 3], 'B': [4, 5, 6], 'C': ["a", "b", "c"]}
data2 = {'X': ['apple', 'banana', 'cherry'], 'Y': ['dog', 'elephant', 'fox']}
```

```
df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)
```

```
loaded_dataframes = []
```

```
filename = "all_dataframes.h5"
df1.to_hdf(filename, key='df1', mode='w') → 처음에 h5파일을 쓰기(저장) 모드
df2.to_hdf(filename, key='df2', mode='a') → 이후 h5파일에 추가(Add) 모드
```

```
#저장한 h5파일 읽기(불러오기)
with pd.HDFStore(filename, 'r') as store:
    for key in store.keys():
        loaded_dataframes.append(store[key])
```

## JSON (JavaScript Object Notation)

- API 및 구성에 대한 사실상의 표준
- JSON은 다양한 시스템과 언어에서 쉽게 읽고 쓸 수 있는 형식이 필요할 때 가장 적합함(나머지는 데이터 처리에 특화)

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

```
data1 = {'A': [1, 2, 3], 'B': [4, 5, 6]}
data2 = {'X': ['apple', 'banana', 'cherry'], 'Y': ['dog', 'elephant', 'fox']}
```

```
df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)
```

```
df1.to_json("df1.json")
df2.to_json("df2.json")
```

→ Json 파일 쓰기(저장)

```
loaded_df1 = pd.read_json("df1.json")
loaded_df2 = pd.read_json("df2.json")
```

→ Json 파일 읽기

## JSON (JavaScript Object Notation)

- API 및 구성에 대한 사실상의 표준
- JSON은 다양한 시스템과 언어에서 쉽게 읽고 쓸 수 있는 형식이 필요할 때 가장 적합함(나머지는 데이터 처리에 특화)

```
combined_list = [df1.to_dict(orient='records'), df2.to_dict(orient='records')]
```

```
with open('combined_list.json', 'w') as file:  
    json.dump(combined_list, file)
```



orient='records' : 딕셔너리로 표현되는 리스트 반환

```
combined_list[0]  
combined_list[1]
```

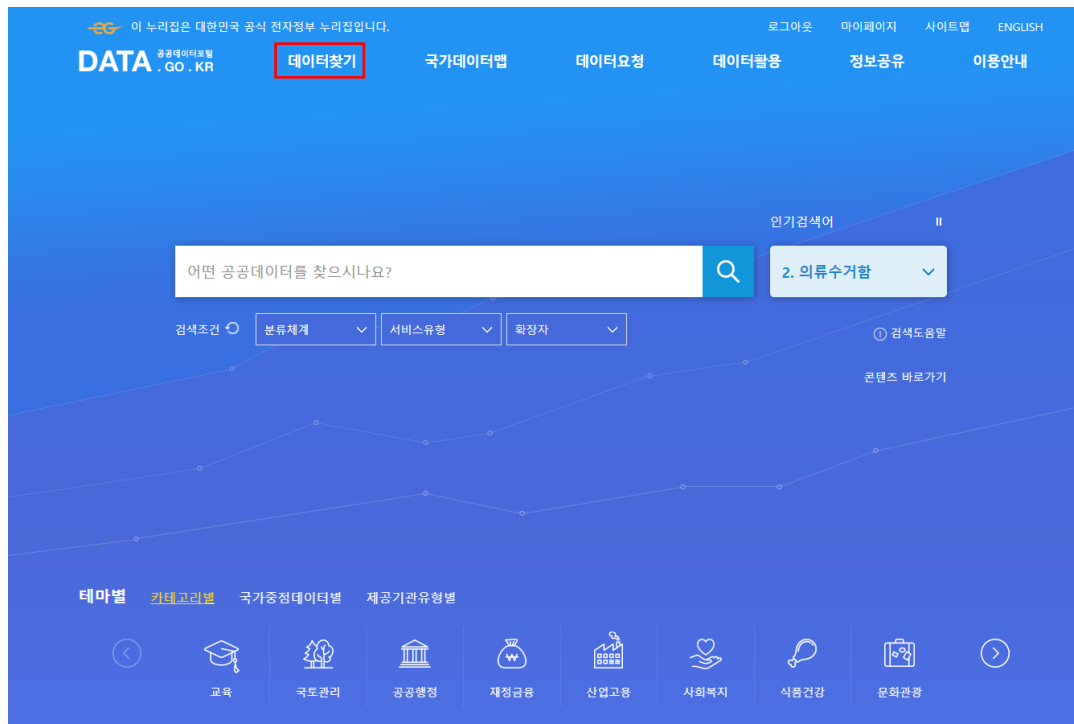
```
pd.DataFrame(combined_list[0])  
pd.DataFrame(combined_list[1])
```

```
with open('combined_list.json', 'r') as file:  
    data = json.load(file)
```

```
pd.DataFrame(data[0])
```



<https://www.data.go.kr/>



<https://www.data.go.kr/>

**DATA** 공공데이터포털  
GO . KR

[데이터찾기](#)
[국가데이터맵](#)
[데이터요청](#)
[데이터활용](#)
[정보공유](#)
[이용안내](#)

[데이터찾기>](#)
[데이터목록](#)
[국가중점데이터](#)
[이슈 및 추천데이터](#)

홈 > 데이터찾기 > 국가중점데이터

국가중점데이터

국민, 기업 등 수요 중심으로 개방의 효과성, 시급성 등이 높은 분야를 선정하고  
민간에서 활용하기 용이한 형태로 정제, 가공하여 개방된 양질의 대용량 데이터를 제공합니다.



분야별	<input type="checkbox"/> 전체	<input type="checkbox"/> 공공행정	<input type="checkbox"/> 과학기술	<input type="checkbox"/> 교육
	<input type="checkbox"/> 교통물류	<input type="checkbox"/> 국토관리	<input type="checkbox"/> 농축수산	<input type="checkbox"/> 문화관광
	<input type="checkbox"/> 법률	<input type="checkbox"/> 보건의료	<input type="checkbox"/> 사회복지	<input type="checkbox"/> 산업고용
	<input type="checkbox"/> 식품건강	<input type="checkbox"/> 재난안전	<input type="checkbox"/> 재정금융	<input type="checkbox"/> 통일외교안보
	<input type="checkbox"/> 환경기상			
기관명	<input type="checkbox"/> 전체	<input type="checkbox"/> (재)세종테크노파크	<input type="checkbox"/> (주)에스알	▶ 88개


검색 초기화

정보명, 기관명 입력

검색

<https://www.data.go.kr/>

## 국가중점데이터

SRT 승차권 정보 

공공데이터 보기

### ◆ SRT 승차권 데이터란?

「SRT 승차권 데이터」는 ㈜에스알이 보유한 SRT 승차권 발권실적, 승객 이동유형 등 국민의 이동·생활과 밀접한 여객 고속철도 분야의 정보들입니다.

### ◆ 어떤 데이터들이 개방되는 것일까?

데이터 제공기관	개방 데이터		종 개방건수	제공 방식
(주)에스알	승차권 진위확인 데이터	일반승차권 확인, 일반 승차권 진위확인 사용통계 정기승차권 확인, 정기 승차권 진위확인 사용통계	OpenAPI 목록 3건	OpenAPI
	승차권 발권 데이터	발매채널별 승차권 발매현황, SR 운영역 승차권 발매현황, 연도별 SR 승차권 발매현황		
	승객 이동 유형	특설/일반실 승차인원, 운행일별 운행노선별 승차 인원, 운행월별 운행노선별 승차거리, 운행월별 역 간 승차인원, 운행월별 운행노선별 병합승차권 이 용인원, 운행월별 운행노선별 고객유형별 승차인 원, 월별 주중/주말 시간대별 역별 승하차인원		
	공급 좌석	월별 주중/주말 노선별 승차율, 월별 주중/주말 노 선별 이용율, SRT 월별 주중/주말 좌석공급실적		

<(주)에스알 승차권데이터 개방시스템 제공>

### ◆ 어떻게 활용할 수 있을까?

SRT 승차권 발권현황 및 승객 이동유형 데이터로 교통기반시설 계획을 수립하거나, 상권분석, 경제활동에 대해 분석할 수 있는 기초자료를 제공합니다.

<https://www.data.go.kr/>

오픈 API (3건)

더보기 >

교통물류	공공기관	국가중점	미리보기
XML	JSON	(주)에스알_열차운행실적	
국민철도SR에서 제공하는 SRT 열차운행실적 집계 데이터입니다. '운행월별 주중/주말 운행노선별 좌석공급실적', '운행월별 주중/주말 열차별 좌석공급실적', '운행월별 주중/주말 시간...			
제공기관	(주)에스알	수정일	2023-12-12
조회수	202	활용신청	5
키워드	에스알,승객이동유형,통계데이터		
			 활용신청
교통물류	공공기관	국가중점	미리보기
XML	JSON	(주)에스알_SRT 승차권 발권	
국민철도SR 승차권 발매현황 데이터로, "발매채널별 승차권 발매 현황", "SR 운영역 승차권 발매현황", "연도별 SR 승차권 발매현황", "특설/일반실 승차인원" 항목이 있습니다.			
제공기관	(주)에스알	수정일	2022-12-27
조회수	4441	활용신청	94
키워드	에스알,SRT 승차권 발매 현황,통계 데이터		
			 활용신청
교통물류	공공기관	국가중점	미리보기
XML	JSON	(주)에스알_SRT 승객이동유형	
국민철도SR 승객이동유형 데이터로, "운행일별 운행노선별 승차인원", "운행월별 운행노선별 승차거리", "운행월별 정차역별 승하차인원", "운행월별 운행노선별 병합승차권 이용인원", ...			
제공기관	(주)에스알	수정일	2022-12-27
조회수	2251	활용신청	45
키워드	에스알,승객이동유형,통계데이터		
			 활용신청

<https://www.data.go.kr/>

#### 공공데이터 제공제도

- \* 공공데이터중 위치정보를 포함한 서비스를 사용하고자 하는 사업자는 '위치정보의 보호 및 이용 등에 관한 법률'에 따라 방송통신위원회에 '위치정보서비스 허가'를 받거나 '위치기반 서비스사업 신고'를 하여야 합니다.
- \* 이에 해당하는 사업자인 경우에는 첨부파일에 '위치기반서비스사업신고필증'을 첨부해 주시기 바랍니다.
- \* 활용신청 시 '위치기반서비스사업신고필증'이 등록되지 않으면 반려가 될 수 있으니 참고 하시기 바랍니다.

#### 활용목적 선택

\*표시는 필수 입력항목입니다.

\*활용목적

☐ 웹 사이트 개발 ☐ 앱개발 (모바일,솔루션등) ☐ 기타 ☐ 참고자료 ☒ 연구(논문 등)

연구 및 분석

7/250

활용목적 입력

파일 선택

Drag & Drop으로 파일을 선택 가능합니다.

첨부파일

#### 라이선스 표시

\*이용허락범위

이용허락범위 제한 없음

☒ 동의합니다.

취소

활용신청

<https://www.data.go.kr/>

### 활용신청 현황

계정	전체	신청일		~	
신청유형	전체	처리상태	전체		
데이터명					
<div>초기화</div> <div>검색</div>					

<b>신청 0건</b> >	<b>활용 5건</b> >	<b>중지 0건</b> >
신청중인 단계	승인되어 활용중인 단계	중지신청하여 운영이 중지된 단계
<div>보류</div> <div>반려</div>	<div>변경신청</div>	
<div>0건</div> <div>0건</div>	<div>0건</div>	

<https://www.data.go.kr/>

#### 기본정보

데이터명	(주)에스알_열차운행실적	상세설명	
서비스유형	REST	심의여부	자동승인
일 호출 제한	10000		
신청유형	개발계정   활용신청	처리상태	승인
활용기간	2024-01-25 ~		

#### 서비스정보

데이터포맷	JSON+XML
Base URL	api.odcloud.kr/api
Swagger URL	https://infuser.odcloud.kr/api/stages/50674/api-docs?1702348097990
API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다. 포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다. * 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.	
일반 인증키 (Encoding)	azi%2FgZE382GbDYFbEBh1CWktKefC5Z4kRolyEzSzpZyob3OJKhQMlcRwwOWjh6J9%2BQzzRst9XM%2BMBOCK2TrBnQ%3D%3D
일반 인증키 (Decoding)	azi/gZE382GbDYFbEBh1CWktKefC5Z4kRolyEzSzpZyob3OJKhQMlcRwwOWjh6J9+QzzRst9XM+MBOCK2TrBnQ==

<https://www.data.go.kr/>

## 오픈API 상세



URL 복사

XML JSON (주)에스알\_열차운행실적

데이터조회하기

활용신청

국민철도SR에서 제공하는 SRT 열차운행실적 집계 데이터입니다. '운행월별 주중/주말 운행노선별 좌석공급실적', '운행월별 주중/주말 열차별 좌석공급실적', '운행월별 주중/주말 시간별 정차역별 승하차인원', '운행월별 주중/주말 운행노선별 승차율', '운행월별 주중/주말 운행노선별 이용률'을 제공합니다.



0



0



관심

## OpenAPI 정보



메타데이터 다운로드



오픈API 에러코드

데이터 개선요청

오픈신고 및 문의

분류체계	교통및유통 - 철도	제공기관	(주)에스알
관리부서명	미래연구원	관리부서 전화번호	02-6484-4221
API 유형	REST	데이터포맷	JSON+XML
활용신청	8	키워드	에스알,승객이동유형,통계데이터
등록	2023-12-12	수정	2023-12-12
비용부과유무	무료	신청가능 트래픽	개발계정 : 10,000 / 운영계정 : 활용사례 등록시 신청하면 트래픽 증가 가능
심의유형	개발단계 : 자동승인 / 운영단계 : 심의승인		
이용허락범위	이용허락범위 제한 없음		
참고문서			



## 샘플코드

Java

Javascript

C#

PHP

Curl

Objective-C

Python

Nodejs

R

# Python3 샘플 코드 #

import requests

url = 'http://kosis.kr/openapi/Data/statisticsData.do'

```
params ={'serviceKey': '서비스키', 'orgId': '101', 'tblId': 'DT_1B01003', 'objL1': '0001', 'objL2': '000', 'objL3': 'T1', 'objL4': '', 'objL5': '', 'objL6': '', 'objL7': '', 'objL8': '', 'itmlId': 'T001',  
'loadGubun': '2', 'prdSe': 'Y', 'startPrdDe': '', 'endPrdDe': '', 'newEstPrdCnt': '1', 'prdInterval': '', 'format': 'json', 'method': 'getList', 'jsonVD': 'Y' }
```

response = requests.get(url, params=params)

print(response.content)

```
pip install xmltodict
```

```
import requests  
import xmltodict
```

```
url = 'API url'  
params = {'serviceKey' : '서비스키' ~~~~ 다양한 옵션}
```

```
response = requests.get(url, params=params)
```

```
print(response.content)
```

```
pip install xmltodict
```

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

```
dict = xmltodict.parse(response.content)  
jsonS=json.dumps(dict["response"]["body"], ensure_ascii=False)  
jsonO=json.loads(jsonS)
```

```
pd.DataFrame(jsonO["items"]["item"])
```