AI와 데이터 기초

pandas 자료구조

오늘 수업은

- 데이터 종류와 구조
- 예제로 보는 데이터 구성
  - seaborn 데이터셋
- 데이터 준비하기
  - pandas
- pandas 자료구조
  - 데이터 생성
  - Series
  - DataFrame
- 유기동물보호현황 확인하기
  - 예제로 보는 데이터 구성
  - 데이터 수집

빅데이터 종류

❖정형데이터(structured data)

- 미리 정해 놓은 형식과 구조에 따라 저장된 데이터

  예) 관계형 데이터베이스의 테이블, 스프레드시트, CSV 등

❖반정형데이터(semi-structured data)

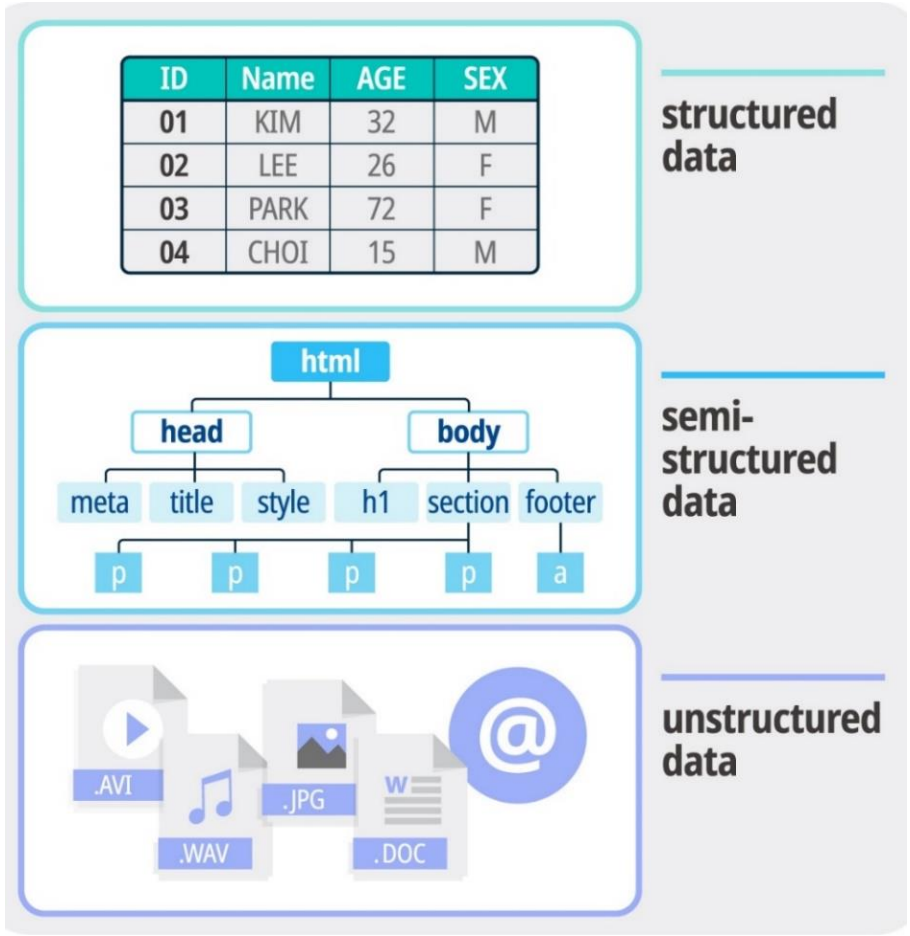
- 일정한 규칙의 고정된 필드에 저장되어 있지 않지만 데이터의 구조 정보를 데이터와 함께 제공하는 데이터

  예)XML, HTML, JSON, 웹문서, 웹로그 등

❖비정형데이터(unstructured data)

- 정의된 구조가 없이 데이터 자체만으로 내용에 대한 질의 처리를 할 수 없는 데이터

  예) 소셜 데이터, 텍스트 문서, 동영상/이미지/음성 데이터, 문서(PDF) 등



[이미지 출처]: 정보통신용어사전
(http://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=175128-2)

(정형) 데이터 구조

- 데이터(표)는 행(row)과 열(column)로 구성
  - 행(row): 하나의 단위로 다루어지는 데이터의 집합(표의 가로축)
  - 열(column): 특정 자료형을 갖는 일련의 데이터 값(표의 세로축)

열(column)

학번	이름	학과	학년
20211021	박형식	영어영문학과	1
20205412	공유	화학과	2
20210578	아이유	수학과	1
19983125	송중기	경영학과	4

행(row) = 레코드(recored)

(정형) 데이터 구조

❖칼럼(Column) = 열

칼럼=열 (Column)	변수 (Variable)	속성 (Attribute)	특징 (Feature)
통계 분야	컴퓨터 분야	인공지능 분야	패턴인식 분야

학번	이름	학과	학년
20211021	박형식	영어영문학과	1
20205412	공유	화학과	2
20210578	아이유	수학과	1
19983125	송중기	경영학과	4

❖칼럼의 종류

- 수치형(Numeric) : 정수형(int), 실수형(float), Bool형
- 범주형(Categorical) : 순서형(category), 텍스트(object)

1. 레코드(행)는 개수는?
2. 칼럼(변수)의 개수는?
3. 수치형 칼럼 개수는?
4. 범주형 칼럼 개수는?

데이터 정보가 중요한가?

- 행과 열
  - 데이터의 크기를 알 수 있음
  - 처리의 양을 파악 할 수 있음
  - 변수(칼럼)들을 통해서 변수간의 관련성에 대한 의문점을 가질 수 있음

- 칼럼의 종류
  - 칼럼들의 연산 가능여부 확인 가능
  - 간단한 통계정보를 통하여 데이터에 대한 대략적인 분석 가능
  - 칼럼의 종류에 따라 프로그램 작성시 오류 파악 가능

데이터의 크기

- 데이터가 크다? : 행이 많거나 열이 많다
- 데이터 분석을 위해서는 행과 열 중 무엇이 많은게 좋을까? : [답] 열
  - 행이 많은 경우
    - 행의 개수가 많을수록 처리하는 양이 많아짐으로 컴퓨터가 느려짐
    - 물리적인 비용(메모리나 CPU, 분산처리, 클라우드)으로 해결가능
    - 100만명과 100명의 평균 또는 분석 방법이 같다면 데이터 분석의 노력 결과는 달라지지 않음
  - 열이 많은 경우
    - 변수간의 관계에 대해 분석할 수 있는 사항들이 많아짐
    - 분석 방법 및 기술이 다양해짐

- 행과 열의 수보다 다양한 데이터가 더 중요!!!
  - 데이터의 가치는 어떤 현상이 조건에 따라 달라진다는 사실을 발견할 때 생김
  - 조건에 따른 현상은 다양한 데이터에 포함된 변수간의 관련성으로 분석

예제로 보는 데이터 구성 (seaborn 데이터셋)

seaborn 데이터셋 불러오기(titanic)

seaborn

- 데이터 시각화 라이브러리
- 내장 데이터셋 22개 제공(sns.get_dataset_names() 명령으로 확인가능)

In [1]:
```
import seaborn as sns
```

In [2]:
```
df = sns.load_dataset('titanic')
```

In [3]:
```
df.head()
```

Out[3]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	de
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Na
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Na
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Na

titanic 데이터 전체보기

```
In [4]: 1 df
```

Out[4]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True

891 rows × 15 columns

titanic 데이터 구성 확인하기

```
In [5]:  1  df.info()
```

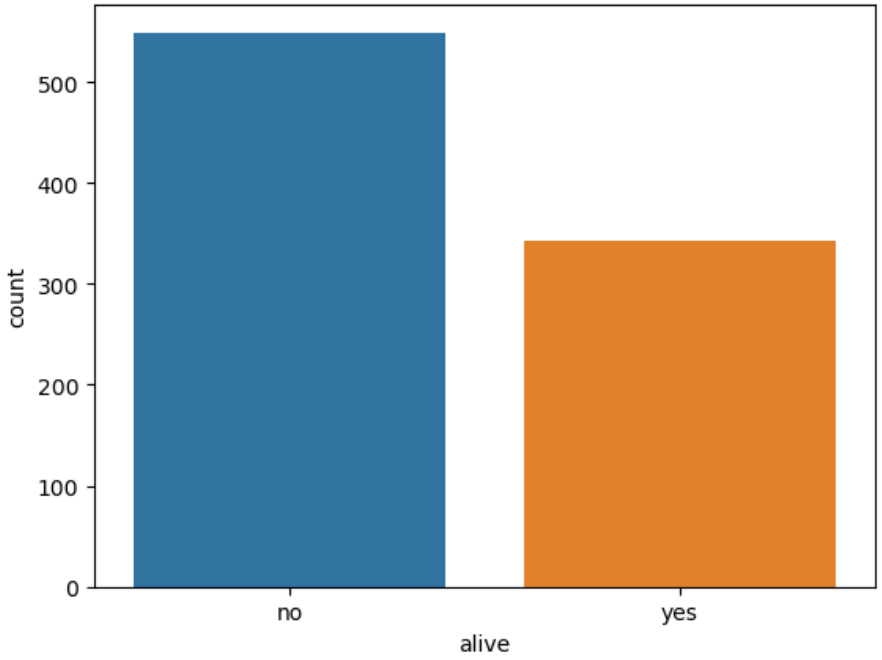
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

❖데이터 정보 확인하기

1. 레코드는 개수는?
2. 칼럼의 개수는?
3. 수치형 칼럼 개수는?
4. 범주형 칼럼 개수는?

titanic 데이터셋 시각화하기

```
In [8]:  1  sns.countplot(data=df, x='alive');
```



데이터 준비하기(pandas)

Pandas란?

❖데이터 분석을 위한 Python 라이브러리

- Panel Data Analysis
- 대용량의 (정형)데이터 처리를 지원함.
- 데이터 관리와 정제 기능을 가진 라이브러리
- 머신러닝, 시각화 등의 데이터 사이언스 관련 라이브러리에서 사용

[pandas.pydata.org]

❖Excel과의 차이점

	엑셀(Excel)	판다스(pandas)
자동화	- 기본적으로 사람의 손으로 작업 - VB로 자동화 가능하기는 함	- 코딩을 통한 자동화
대용량 데이터 처리	- 큰 데이터 처리에 부적합함 - 로딩이 안되는 경우도 있음 - 데이터 처리 속도가 느림	- 데이터 처리 속도 빠름
분석 방법	- 지원되는 기능에 한정	- 사용자가 코딩을 통해 다양한 창의적인 데이터 분석이 가능함

설치방법 및 라이브러리 선언

- 콘솔에서 아래의 명령으로 설치
  - Path 설정에 문제가 있는 경우 아래 명령이 실행 안될 수 있음
  - 그럴 경우 pip 명령이 있는 위치로 경로를 이동한 후 실행

```
pip install pandas
```

- 코랩(colab)에는 이미 설치되어 있음

- 라이브러리 선언

```
import pandas as pd
```

파이썬 기본 자료구조

리스트

- a = [10, 20, 30]
- b = [[1,2], [3,4], [5,6]]
- 데이터 개별 접근 : a[0] + a[2] / a[0:2] / b[2][1]

튜플: 데이터 변경 불가

- a = (10,20,30)
- b = ((1,2), (3,4), (5,6))
- 데이터 개별 접근 : a[0] + a[2] / a[0] = 1 (Error)

딕셔너리

- key : value 형식으로 데이터 저장
- dict = { 'a' : 100 , 'b' : 200, 'c' : 300 }
- 데이터 개별 접근 : dict['a'] + dict ['b']

pandas 자료 구조

Series

- 1차원 배열 형태로써 같은 종류의 데이터가 순서대로 나열된 데이터 구조

2016	2005	2011	2003	2007

DataFrame

- 데이터를 표 형식 데이터(행, 열) 구조로 저장
- 예)CSV, EXCEL

번호	도서명	저자	출판사	발행년도
1	꿀벌의 예언 1	베르나르 베르베르	열린책들	2023
2	메리골드 마음 세탁소	윤정은	북로망스	2023
3	바다가 들리는 편의점	마치다 소노코	모모	2023
4	냉정과 열정사이	츠지 히토나리	소담출판사	2003
5	남한산성:김훈 장편소설	김훈	학고재	2007

Series

- 여러 값을 나열한 1차원 자료구조
- DataFrame을 구성하는 하위요소
- DataFrame을 다루는 함수는 대부분 시리즈를 이용하여 연산함

index 자동 부여

```
1 import pandas as pd
2 data = pd.Series([3, 4, 5])
3 print(data)
4 print(data[0])
5 print(data[1])
6 print(data[2])
```

```
0    3
1    4
2    5
dtype: int64
3
4
5
```

index 직접 부여

```
1 x_data = pd.Series([20, 25, 22], index=['kim','lee','park'])
2 print(x_data)
3 print(x_data[0])
4 print(x_data[1])
5 print(x_data[2])
6 print(x_data['kim'])
7 print(x_data['lee'])
8 print(x_data['park'])
```

```
kim     20
lee     25
park    22
dtype: int64
20
25
22
20
25
22
```

DataFrame

❖DataFrame을 이용한 데이터 생성

- pd.DataFrame({'key1':value1, 'key2':value2, …})
  - key : 열이름(변수명)
  - value : 열이름에 해당하는 데이터들(데이터, 리스트)

	A	B	C
1	학번	이름	학과
2	20211021	박형식	영어영문학과
3	20205412	공유	화학과
4	20210578	아이유	수학과

```
1 import pandas as pd
2
3 no = [20231021, 20225412, 20210578]
4 name = ['박형식','공유', '아이유']
5 major = ['영어영문학과', '화학과','수학과']
6
7 df = pd.DataFrame({'학번':no, '이름':name, '학과':major})
8 df
```

	학번	이름	학과
0	20231021	박형식	영어영문학과
1	20225412	공유	화학과
2	20210578	아이유	수학과

DataFrame

❖DataFrame을 이용한 데이터 생성

- pd.DataFrame([데이터들], columns=['열이름들'])
  - 데이터들 : 각 열의 데이터들(데이터, 리스트)
  - 열이름들 : 각 열의 변수명

	A	B	C
1	학번	이름	학과
2	20211021	박형식	영어영문학과
3	20205412	공유	화학과
4	20210578	아이유	수학과

```
1 import pandas as pd
2
3 AI_class = [[20231021,'박형식','영어영문학과'],
4             [20225412,'공유','화학과'],
5             [20210578,'아이유','수학과']
6             ]
7
8 df = pd.DataFrame(AI_class, columns=['학번','이름','학과'])
9 df
```

```
1 import pandas as pd
2
3 df = pd.DataFrame([[20231021,'박형식','영어영문학과'],
4             [20225412,'공유','화학과'],
5             [20210578,'아이유','수학과']
6             ], columns=['학번','이름','학과'])
7 df
```

	학번	이름	학과
0	20231021	박형식	영어영문학과
1	20225412	공유	화학과
2	20210578	아이유	수학과

pandas 자료구조

[실습내용]

1. 데이터 생성
2. Series
3. DataFrame

1. Series를 활용한 데이터 생성

❖index 자동 부여

- 변수명 = pd.Series([데이터])
  - 데이터 : 리스트 형식으로 작성
  - 데이터 참조: 리스트 참조 방식 사용 ex)a[1]

❖index 직접 부여

- 변수명 = pd.Series([데이터], index=[index이름])
  - 데이터: 리스트 형식으로 작성
  - index이름: 데이터의 개수만큼 생성

2. DataFrame을 활용한 데이터 생성

❖변수명 = pd.DataFrame({'key1':value1, 'key2':value2, …})

- key : 열이름(변수명)
- value : 열이름에 해당하는 데이터들(리스트)

❖변수명 = pd.DataFrame([데이터], columns=['열이름들'])

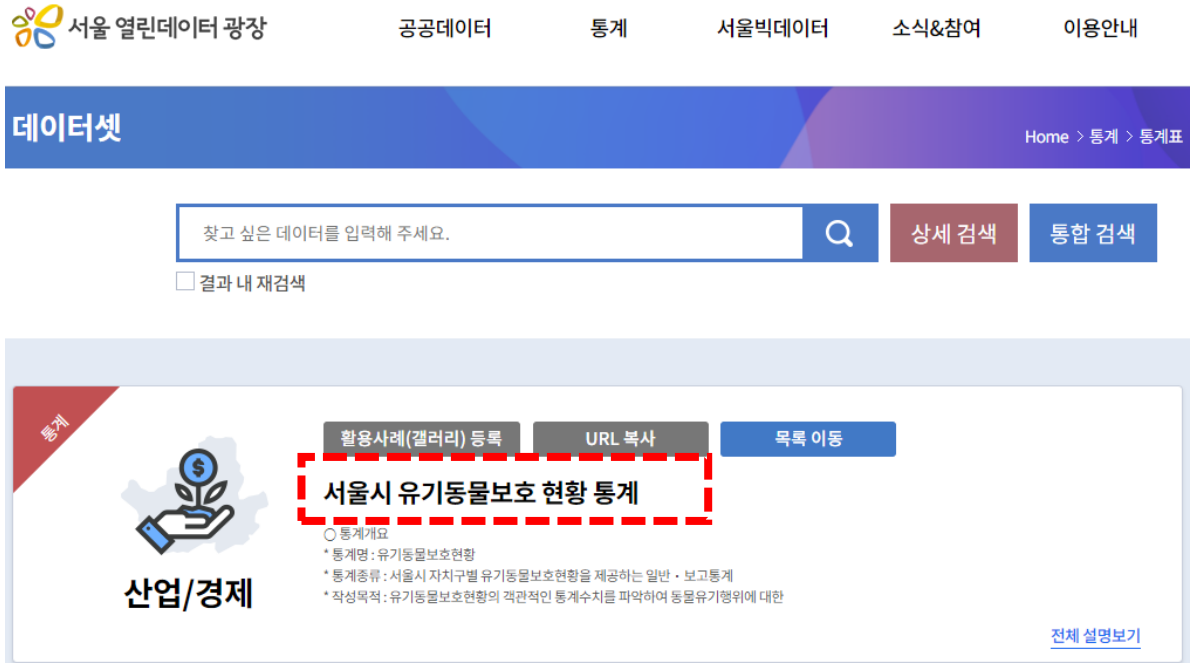
- 데이터들 : 각 열의 데이터들(리스트)
- 열이름들 : 각 열의 변수명

유기동물보호현황 확인하기

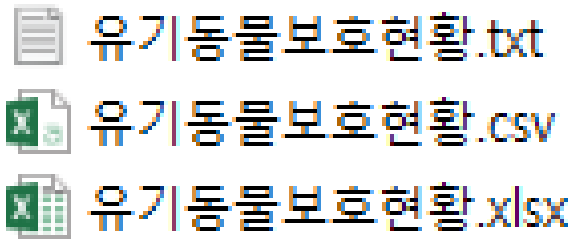
[실습내용]

1. 데이터 구성
2. 데이터 수집

1. 데이터 수집하기



- EXCEL(xlsx) : **셀병합 check 해제**
- CSV
- TXT



위 3개의 파일을 각각 선택하여 [다운로드]

https://data.seoul.go.kr/dataList/369/S/2/datasetView.do

2. 데이터 읽어오기

```
import pandas as pd     # 데이터 관리와 정제 기능을 가진 라이브러리
```

❖xlsx 파일을 불러올때 :

- 변수명 = pd.read_excel('파일경로명', 속성들)

❖csv 또는 txt 파일을 불러올때

- 변수명 = pd.read_csv('파일경로명', 속성들)

2. 데이터 읽어오기

❖pd.read_csv()와 pd.read_excel()의 속성들

- delimiter='구분기호'
  - 생략시 ","로 인식
  - \t: tab을 열 구분 문자로 인식
- header=[행번호] :
  - 위에서 몇 째줄 부터 읽어올지 지정(줄 수는 0부터 시작)
  - 열이름(변수이름)으로 설정할 index 번호 기술
- encoding= '인코딩방식'
  - 한글이 깨져서 보일 경우 인코딩 방식 설정
  - EUC_KR' (한글이 포함된 일반적인 경우)/ 'cp949' (MS office에서 저장한 파일 형식)
- thousands=','
  - ',가 포함된 문자열 데이터에서 ',를 삭제하고 숫자형 데이터를 변경

3. 데이터 정보 보기

❖데이터 정보 보기

- 변수명.info() : 데이터 타입, 각 아이템 개수, 누락데이터 수 등 확인

4. 데이터를 화일에 저장하기

❖파일에 저장하기

- 변수명.to_csv('화일명')
  - DataFrame을 csv 파일로 저장

- 변수명.to_excel('화일명')
  - DataFrame을 excel 파일로 저장