데사 B0002

데이터마이닝이해와실습

김 태 완

kimtwan21@dongduk.ac.kr

데이터 읽기 w/DataFrame

- 데이터 경진대회 뿐만 아니라 데이터 분석을 하는 경우 주로 외부 데이터를 csv 형태로 읽어 들어오는 것이 일반적
- CSV는 Comma Seperated Value의 약자로 ',' 쉼표(comma)로 분리된 텍스트 파일
- 이 때, Pandas의 Dataframe 형태로 읽어 들어오는 것을 추천

```
import pandas as pd

df = pd.read_csv('파일경로/파일이름.csv')
print(df)
```

read_csv 함수 옵션

- filepath or buffer
  - 파일경로/파일이름.csv 을 입력하여 파일을 불러옴
- sep or delimiter
  - 초기값은 comma(,) 이나, 만약 쉼표(,)로 분리되어 있지 않은 경우 기준이 되는 값을 입력 (/ or Tab 등)
- header
  - 초기값은 0이며, 컬럼명으로 사용할 행의 번호를 입력
- names
  - 사용할 변수명을 입력. 파일에 변수명이 없다면 header를 None으로 설정해야 함
- index_col
  - 데이터의 인덱스로 사용할 열의 번호를 입력
- skiprows
  - 첫 행을 기준으로 데이터를 얼마나 건너뛰고 읽어올지를 정함
- nrows
  - 파일을 읽어올 행의 수를 입력

데이터 (데이터프레임) 살펴보기

- df.head( ) / df.tail( )
  - 데이터 첫 5개의 행과 마지막 5개의 행을 출력하는 함수
  - default 값이 5이라 괄호안에 아무것도 안 넣은 상태에는 다섯 줄을 출력하지만
  - 숫자를 넣어주면 첫 N 줄과 마지막 N 줄을 출력할 수 있음

```
import pandas as pd
df = pd.read_csv(…/Data01.csv’)

df.head(7)
```

	날짜	상품명	발주가능상태	입고수량	카테고리	출고수량
0	2023-05-16	S7_0	발주가능	65	빵	57
1	2023-05-16	S7_3	발주가능	679	빵	679
2	2023-05-16	S7_7	발주가능	589	우유	584
3	2023-05-16	S7_11	발주가능	408	빵	401
4	2023-05-16	S7_14	발주가능	373	빵	371
5	2023-05-16	S7_16	발주가능	597	주류	589
6	2023-05-16	S7_17	발주가능	157	우유	157

데이터 (데이터프레임) 살펴보기

- df.info( )
  - 데이터프레임의 summary를 출력하는 함수
  - 각 열의 데이터 타입 (int64, float64등), 결측 값을 제외한 데이터값 개수와 메모리 사용량 등 정보를 확인

- df.shape( )
  - 데이터 프레임의 행과 열을 튜플 형태로 반환해주는 함수

```
df.info()
df.shape
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   날짜      15000 non-null  object
 1   상품명     15000 non-null  object
 2   발주가능상태  15000 non-null  object
 3   입고수량    15000 non-null  int64
 4   카테고리    15000 non-null  object
 5   출고수량    15000 non-null  int64
dtypes: int64(2), object(4)
memory usage: 703.2+ KB
(15000, 6)
```

데이터 (데이터프레임) 살펴보기

- df.dtypes
  - 각 열의 데이터 타입을 출력하는 함수
  - object : 범주형 데이터

```
날짜          object
상품명         object
발주가능상태     object
입고수량        int64
카테고리        object
출고수량        int64
dtype: object
```

- df.columns
  - 열 이름을 확인할 수 있는 함수

```
Index(['날짜', '상품명', '발주가능상태', '입고수량', '카테고리', '출고수량'], dtype='object')
```

- len(df)
  - 데이터프레임의 행 개수를 출력하는 함수

데이터 (데이터프레임) 살펴보기

- 숫자형 데이터 분석 : **df.describe()**
  - 숫자형 자료구조인 column의 기본 통계값 출력 메소드
  - 기본적으로 결측값 (NaN)을 제외하고 통계 값 계산
  - count : column 별 총 데이터 수
  - mean/std : column 별 데이터 평균 / 표준편차
  - min/max : column 별 데이터 최소 / 최대
  - 25%/50%/75%/ : column 별 사분위수

```
df.describe()
df['입고수량'].describe()
```

	입고수량	출고수량
count	15000.000000	15000.000000
mean	351.089400	346.551400
std	202.067387	202.081703
min	0.000000	-9.000000
25%	177.000000	172.000000
50%	351.000000	347.000000
75%	526.000000	522.000000
max	699.000000	699.000000

데이터 (데이터프레임) 살펴보기

- 범주형 데이터 분석

```
df['카테고리'].unique()
df['카테고리'].value_counts()
df['카테고리'].describe()  #범주형 데이터만 요약해서 정리 가능
```

```
빵          4265
물          3115
우유          2750
간식          1732
주류          1709
배달음식        1038
스낵           238
기타           153
Name: 카테고리, dtype: int64
```

```
count     15000
unique        8
top           빵
freq       4265
Name: 카테고리, dtype: object
```

데이터 (데이터프레임) 살펴보기

- 범주형 데이터 분석

```
df[['상품명', '카테고리']].describe()
```

	상품명	카테고리
count	15000	15000
unique	528	8
top	S7_114	빵
freq	47	4265

- 모든 범주형 데이터 분석

```
numeric_list = df.describe().columns.tolist()
total_list = df.columns.tolist()
char_list= []

for i in total_list:
    if i not in numeric_list:
        char_list.append(i)

char_list

df[char_list].describe()
```

	날짜	상품명	발주가능상태	카테고리
count	15000	15000	15000	15000
unique	84	528	3	8
top	2023-05-26	S7_114	발주가능	빵
freq	226	47	14700	4265

데이터 (데이터프레임) 살펴보기

- 데이터 정렬

```
df.sort_values(by='입고수량') #오름차순
df.sort_values(by='입고수량', ascending=False) #내림차순
```

- 입고수량이 만약 같은 경우 출고수량으로 정렬하고 싶을 때

```
df.sort_values(by=['입고수량', '출고수량']) #오름차순
df.sort_values(by=['입고수량', '출고수량'], ascending=[True,
False]) # 입고수량은 오름차순, 출고수량은 내림차순
```

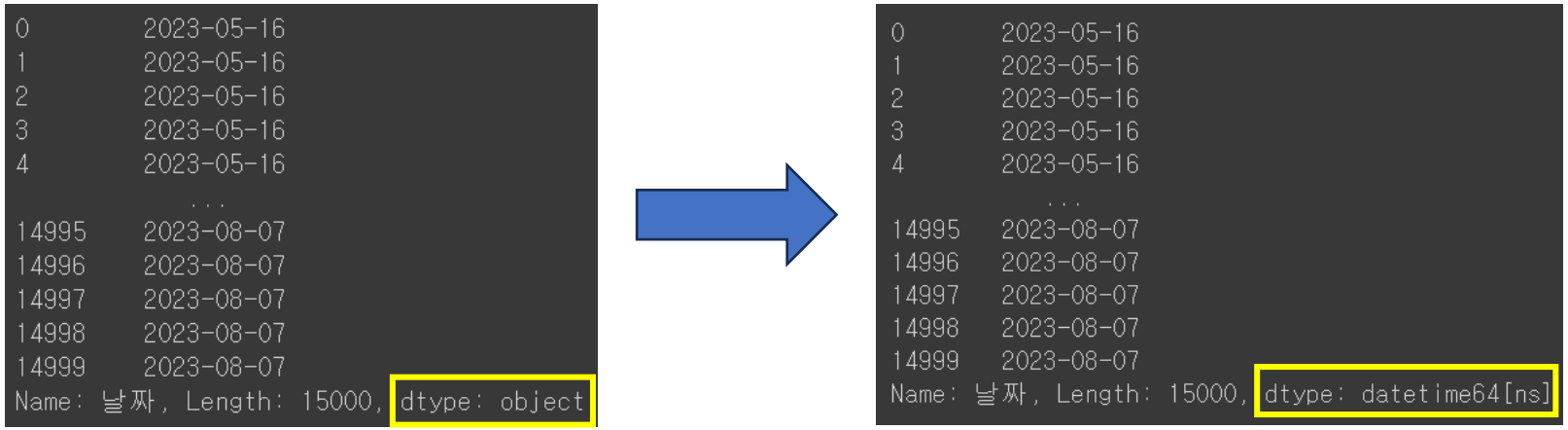
데이터 (데이터프레임) 살펴보기

- 날짜 데이터 처리
  - 기본적으로 문자형 데이터로 인식

```
df['날짜']
```

  - pandas 이용 시 문자형 자료구조가 아닌 '날짜' 자료구조 사용 추천

```
pd.to_datetime(df['날짜'])
```



데이터 (데이터프레임) 살펴보기

- 날짜 데이터 처리
  - 실제 연도/월/주차/일 출력 가능

```
df['날짜2'] = pd.to_datetime(df['날짜'])


df['연도'] = df['날짜2'].dt.year
df['월'] = df['날짜2'].dt.month
df['주차'] = df['날짜2'].dt.week
df['일자'] = df['날짜2'].dt.day
df['요일'] = df['날짜2'].dt.day_name()


df.head(3)
```

	날짜	상품명	발주가능상태	입고수량	카테고리	출고수량	날짜2	연도	월	주차	일자	요일
0	2023-05-16	S7_0	발주가능	65	빵	57	2023-05-16	2023	5	20	16	Tuesday
1	2023-05-16	S7_3	발주가능	679	빵	679	2023-05-16	2023	5	20	16	Tuesday
2	2023-05-16	S7_7	발주가능	589	우유	584	2023-05-16	2023	5	20	16	Tuesday

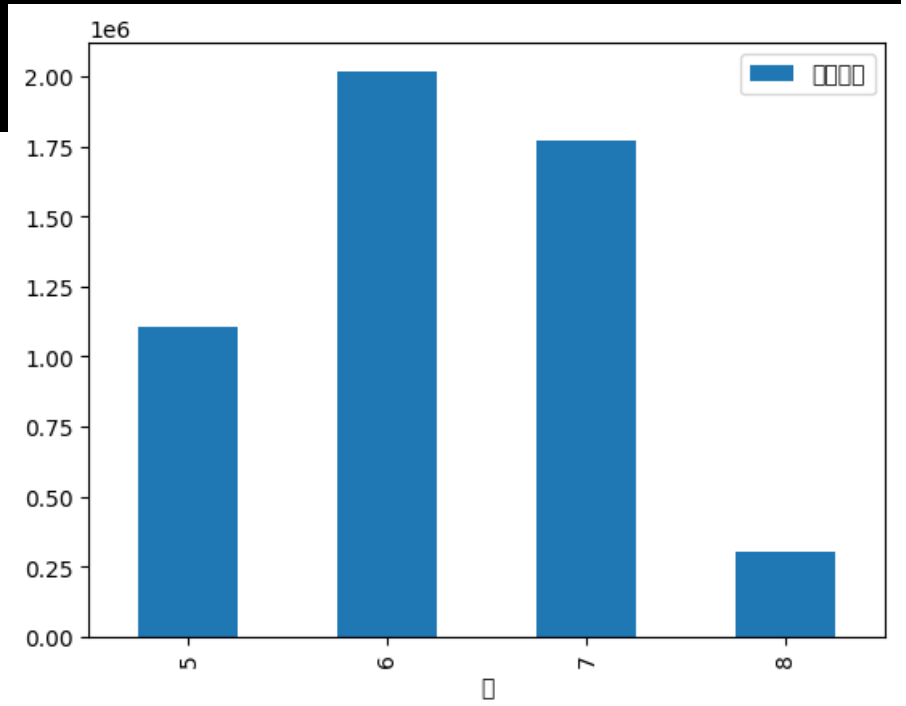
데이터 (데이터프레임) 살펴보기

- 날짜 데이터 처리
  - excel 처럼 pivot table 기능도 사용 가능

```
pd.pivot_table(data=df, index='월', values='출고수량', aggfunc='sum')

pd.pivot_table(data=df, index='월', values='출고수량',
aggfunc='sum').plot(kind='bar')
```

월	출고수량
5	1103918
6	2018510
7	1771355
8	304488



감사합니다

kimtwan21@dongduk.ac.kr

김 태 완