Pandas

Pandas 라이브러리

- Pandas 패키지 로드

```
1 import pandas as pd
```

- Series – 1차원, 1개의 column을 series 라고 함

```
1 a = [1, 2, 3, 4]
2 s = pd.Series(a)
3 print(s)
4 print(type(s))
```

```
0    1
1    2
2    3
3    4
dtype: int64
<class 'pandas.core.series.Series'>
```

Pandas 라이브러리

- DataFrame – 1. 리스트로 만들기

```
company1 = [['삼성', 2000, '스마트폰'],
            ['현대', 1000, '자동차'],
            ['네이버', 500, '포털']]
df1 = pd.DataFrame(company1)
df1
```

	0	1	2
0	삼성	2000	스마트폰
1	현대	1000	자동차
2	네이버	500	포털

Pandas 라이브러리

- 컬럼명 만들기

```
1 df1.columns = ['기업명', '매출액', '업종']
2 df1
```

	기업명	매출액	업종
0	삼성	2000	스마트폰
1	현대	1000	자동차
2	네이버	500	포털

Pandas 라이브러리

- DataFrame – 2. 딕셔너리로 만들기

```
company2 = {'기업명':['삼성', '현대', '네이버'],
            '매출액': [2000, 1000, 500],
            '업종': ['스마트폰', '자동차', '포털']
           }
df2 = pd.DataFrame(company2)
df2
```

	기업명	매출액	업종
0	삼성	2000	스마트폰
1	현대	1000	자동차
2	네이버	500	포털

Pandas 라이브러리

- 인덱스명 만들기

```
1 df2.index = ['회사1','회사2','회사3']
2 df2
```

	기업명	매출액	업종
회사1	삼성	2000	스마트폰
회사2	현대	1000	자동차
회사3	네이버	500	포털

Pandas 라이브러리

- Series == Column 확인?

```
1 print(df1['기업명'])
2 print(type(df1['기업명']))
```

```
0     삼성
1     현대
2    네이버
Name: 기업명, dtype: object
<class 'pandas.core.series.Series'>
```

Pandas – CSV 파일 읽어오기

- pd.read_csv('경로')

```
1 df = pd.read_csv('data/korean-idol.csv')
2 df
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	지민	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	지드래곤	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
2	강다니엘	NaN	커넥트	남자	1996-12-10	180.0	A	8273745
3	뷔	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501

Pandas – 기본정보 알아보기

- .head() – 테이블 첫 행 부터 5개 행을 보여줌

```
1 df.head()
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	지민	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	지드래곤	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
2	강다니엘	NaN	커넥트	남자	1996-12-10	180.0	A	8273745
3	뷔	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501
4	화사	마마무	RBW	여자	1995-07-23	162.1	A	7650928

- .tail() – 테이블 끝 행 부터 5개 행을 보여줌

```
1 df.tail()
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
10	태연	소녀시대	SM	여자	1989-03-09	NaN	A	3918661
11	차은우	아스트로	판타지오	남자	1997-03-30	183.0	B	3506027
12	백호	뉴이스트	플레디스	남자	1995-07-21	175.0	AB	3301654
13	JR	뉴이스트	플레디스	남자	1995-06-08	176.0	O	3274137
14	슈가	방탄소년단	빅히트	남자	1993-03-09	174.0	O	2925442

```
1 df.head(10)
```

```
1 df.tail(10)
```

Pandas – 기본정보 알아보기

- .columns – 열 이름 출력

```
1 df.columns
```

```
Index(['이름', '그룹', '소속사', '성별', '생년월일', '키', '혈액형', '브랜드평판지수'], dtype='object')
```

- 열 이름 전체 변경

```
1 df.columns = ['name', '그룹', '소속사', '성별', '생년월일', '키', '혈액형', '브랜드평판지수']
```

- 특정 열 이름 변경

```
1 df.rename(columns={'name':'이름'}, inplace=True)
```

- .index – 행 출력

```
1 df.index
```

```
RangeIndex(start=0, stop=15, step=1)
```

Pandas – 기본정보 알아보기

- .info() – 기본 정보 표시. 주로 Null 값과 데이터 타입을 볼 때 활용

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   이름       15 non-null     object
 1   그룹       14 non-null     object
 2   소속사      15 non-null     object
 3   성별       15 non-null     object
 4   생년월일     15 non-null     object
 5   키        13 non-null     float64
 6   혈액형      15 non-null     object
 7   브랜드평판지수  15 non-null     int64
dtypes: float64(1), int64(1), object(6)
memory usage: 1.1+ KB
```

- .shape

```
1 df.shape
```

```
(15, 8)
```

Pandas – 통계정보 알아보기

- .describe() – column 데이터 중, 수치 데이터의 통계치를 보여줌

```
1 df.describe()
```

	키	브랜드평판지수
count	13.000000	1.500000e+01
mean	175.792308	5.655856e+06
std	5.820576	2.539068e+06
min	162.100000	2.925442e+06
25%	174.000000	3.712344e+06
50%	177.000000	4.668615e+06
75%	179.200000	7.862214e+06
max	183.000000	1.052326e+07

Pandas – 정렬

- .sort_index() – 오름차순 정렬

```
1 df.sort_index()
```

- .sort_index(ascending=False) – 내림차순 정렬

```
1 df.sort_index(ascending=False)
```

- .sort_values(by='컬럼명') – 오름차순 정렬

```
1 df.sort_values(by='키')
```

- .sort_values(by='컬럼명', ascending=False) – 내림차순 정렬

```
1 df.sort_values(by='키', ascending=False)
```

Pandas – 정렬

- 여러 개의 컬럼명으로 정렬 하고 싶다면?

```
1  df.sort_values(by=['키', '브랜드평판지수'])
```

- 내림차순 정렬은?

```
1  df.sort_values(by=['키', '브랜드평판지수'], ascending=False)
```

Pandas – 선택

- 특정 열 선택

```
1 df['이름']
```

```
0       지민
1     지드래곤
2     강다니엘
3        뷔
4       화사
5       정국
6       민현
7       소연
8        진
9      하성운
10      태연
11     차은우
12      백호
13      JR
14      슈가
Name: 이름, dtype: object
```

Pandas – 복수 선택

- 행 선택

```
1 df[:5]
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	지민	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	지드래곤	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
2	강다니엘	NaN	커넥트	남자	1996-12-10	180.0	A	8273745
3	뷔	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501
4	화사	마마무	RBW	여자	1995-07-23	162.1	A	7650928

Pandas – 복수 선택

- .loc[] - .iloc[]와 더불어 많이 쓰임

```
1 df.loc[:, '이름']
```

```
0      지민
1    지드래곤
2    강다니엘
3       뷔
4      화사
5      정국
```

```
1 df.loc[:, ['이름', '생년월일']]
```

	이름	생년월일
0	지민	1995-10-13
1	지드래곤	1988-08-18
2	강다니엘	1996-12-10
3	뷔	1995-12-30
4	화사	1995-07-23
5	정국	1997-09-01

```
1 df.loc[1:5, '이름':'생년월일']
```

	이름	그룹	소속사	성별	생년월일
1	지드래곤	빅뱅	YG	남자	1988-08-18
2	강다니엘	NaN	커넥트	남자	1996-12-10
3	뷔	방탄소년단	빅히트	남자	1995-12-30
4	화사	마마무	RBW	여자	1995-07-23
5	정국	방탄소년단	빅히트	남자	1997-09-01

! .loc[]에서 행은 마지막 행까지 포함

Pandas – 복수 선택

- .iloc[]

```
1 df.iloc[:, 0]
```

```
0      지민
1    지드래곤
2     강다니엘
3       뷔
4       화사
5       정국
```

```
1 df.iloc[:, [0, 4]]
```

	이름	생년월일
0	지민	1995-10-13
1	지드래곤	1988-08-18
2	강다니엘	1996-12-10
3	뷔	1995-12-30
4	화사	1995-07-23
5	정국	1997-09-01

```
1 df.iloc[1:5, 0:5]
```

	이름	그룹	소속사	성별	생년월일
1	지드래곤	빅뱅	YG	남자	1988-08-18
2	강다니엘	NaN	커넥트	남자	1996-12-10
3	뷔	방탄소년단	빅히트	남자	1995-12-30
4	화사	마마무	RBW	여자	1995-07-23

! .iloc[]에서 마지막 포함 하지 않음

Pandas – 복수 선택

- Boolean Indexing – 조건을 활용한 색인

```
1 df['키'] > 180 # 조건
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6      True
7     False
8     False
9     False
10    False
11     True
12    False
13    False
14    False
Name: 키, dtype: bool
```

```
1 df[df['키'] > 180]
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
6	민현	뉴이스트	플레디스	남자	1995-08-09	182.3	0	4989792
11	차은우	아스트로	판타지오	남자	1997-03-30	183.0	B	3506027

? 모든 열이 출력 되어서 불편함. 특정 열만 출력 할 수 있을까?

Pandas – 복수 선택

- 1. 맨 뒤에 출력 할 컬럼명 추가

```
1 df[df['키'] > 180]['이름']
```

```
6      민현
11    차은우
Name: 이름, dtype: object
```

```
1 df[df['키'] > 180][['이름', '키']]
```

	이름	키
6	민현	182.3
11	차은우	183.0

! 괄호에 주의

- 2. .loc[] 활용

```
1 df.loc[df['키'] > 180, '이름']
```

```
6      민현
11    차은우
Name: 이름, dtype: object
```

```
1 df.loc[df['키'] > 180, ['이름', '키']]
```

	이름	키
6	민현	182.3
11	차은우	183.0

Pandas – 복수 선택

- .isin(조건) – 조건에 부합하는 색인만 선택

```
1 my_condition = ['플레디스', 'SM']
```

```
1 df['소속사'].isin(my_condition)
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6      True
7     False
8     False
9     False
10     True
11    False
12     True
13     True
14    False
Name: 소속사, dtype: bool
```

```
1 df.loc[ df['소속사'].isin(my_condition), ['소속사', '브랜드평판지수'] ]
```

	소속사	브랜드평판지수
6	플레디스	4989792
10	SM	3918661
12	플레디스	3301654
13	플레디스	3274137

Pandas – NaN

- Not a Number, 결측값(Null, 비어있는 값)

```
1 df
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	지민	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	지드래곤	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
2	강다니엘	NaN	커넥트	남자	1996-12-10	180.0	A	8273745
3	뷔	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501
4	화사	마마무	RBW	여자	1995-07-23	162.1	A	7650928
5	정국	방탄소년단	빅히트	남자	1997-09-01	178.0	A	5208335
6	민현	뉴이스트	플레디스	남자	1995-08-09	182.3	O	4989792
7	소연	아이들	큐브	여자	1998-08-26	NaN	B	4668615
8	진	방탄소년단	빅히트	남자	1992-12-04	179.2	O	4570308
9	하성운	핫샷	스타크루이엔티	남자	1994-03-22	167.1	A	4036489
10	태연	소녀시대	SM	여자	1989-03-09	NaN	A	3918661
11	차은우	아스트로	판타지오	남자	1997-03-30	183.0	B	3506027
12	백호	뉴이스트	플레디스	남자	1995-07-21	175.0	AB	3301654
13	JR	뉴이스트	플레디스	남자	1995-06-08	176.0	O	3274137
14	슈가	방탄소년단	빅히트	남자	1993-03-09	174.0	O	2925442

Pandas – NaN

- Not a Number, 결측값(Null, 비어있는 값)

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   이름       15 non-null     object
 1   그룹       14 non-null     object
 2   소속사      15 non-null     object
 3   성별       15 non-null     object
 4   생년월일     15 non-null     object
 5   키        13 non-null     float64
 6   혈액형      15 non-null     object
 7   브랜드평판지수  15 non-null     int64
dtypes: float64(1), int64(1), object(6)
memory usage: 1.1+ KB
```

Pandas – NaN

- .isna() – 결측 값 찾기 (not available)

```
1 df.isna()
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	True	False	False	False	False	False	False
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	

- .isnull()

```
1 df.isnull()
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	True	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False

Pandas – NaN

- Nan값만 찾기

```
1 df['키'].isnull()
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7      True
8     False
9     False
10     True
11    False
12    False
13    False
14    False
Name: 키, dtype: bool
```

```
1 df[df['키'].isnull()]
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
7	소연	아이들	큐브	여자	1998-08-26	NaN	B	4668615
10	태연	소녀시대	SM	여자	1989-03-09	NaN	A	3918661

- Nan값이 아닌 행 만 찾기

```
1 df[df['키'].notnull()]
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	지민	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	지드래곤	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
5	정국	방탄소년단	빅히트	남자	1997-09-01	178.0	A	5208335
6	민현	뉴이스트	플레디스	남자	1995-08-09	182.3	O	4989792
8	진	방탄소년단	빅히트	남자	1992-12-04	179.2	O	4570308
9	한성윤	한샘	스타크루이엔티	남자	1994-03-22	167.1	A	4036489

Pandas – 복사

- 그냥 복사하면 메모리 주소 값만 복사되고, 내용은 공유한다.

```
1 new_df = df
2 print(hex(id(df)))
3 print(hex(id(new_df)))
```

```
0x2150aefbb20
0x2150aefbb20
```

```
1 new_df['이름'] = 0
2 new_df
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	0	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	0	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
2	0	NaN	커넥트	남자	1996-12-10	180.0	A	8273745
3	0	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501

```
1 df
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	0	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	0	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
2	0	NaN	커넥트	남자	1996-12-10	180.0	A	8273745
3	0	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501

Pandas – 복사

- .copy() – 깊은 복사, 별도의 메모리 공간에 복사 해준다.

```
1 copy_df = df.copy()
2 print(hex(id(df)))
3 print(hex(id(copy_df)))
```

```
0x2150aefbb20
0x2150b869d30
```

```
1 copy_df['이름'] = 0
2 copy_df
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	0	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	0	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
2	0	NaN	커넥트	남자	1996-12-10	180.0	A	8273745
3	0	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501

```
1 df
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수
0	지민	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260
1	지드래곤	빅뱅	YG	남자	1988-08-18	177.0	A	9916947
2	강다니엘	NaN	커넥트	남자	1996-12-10	180.0	A	8273745
3	빈	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501

Pandas – 데이터 추가

- 행 추가 – 딕셔너리 형태에 .append() 써서, ignore_index=True 옵션 추가 필수

```
df = df.append({'이름': '쯔위', '그룹': '트와이스', '소속사': 'JYP',
                '성별': '여자', '생년월일': '1999-06-14', '키': 172.0, '혈액형': 'A'
               }, ignore_index=True)
```

- ! 원래의 df 에 대입 해 주어야 반영됨
- ! 값이 없는 열은 NaN 값이 들어감

14	슈가	방탄소년단	빅히트	남자	1993-03-09	174.0	O	2925442.0
15	쯔위	트와이스	JYP	여자	1999-06-14	172.0	A	NaN

Pandas – 데이터 추가

- 열 추가 – 새 컬럼명 지정해서 대입해주면 끝

```
1 df['국적'] = '대한민국'
2 df.head()
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수	국적
0	지민	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260.0	대한민국
1	지드래곤	빅뱅	YG	남자	1988-08-18	177.0	A	9916947.0	대한민국
2	강다니엘	NaN	커넥트	남자	1996-12-10	180.0	A	8273745.0	대한민국
3	뷔	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501.0	대한민국
4	화사	마마무	RBW	여자	1995-07-23	162.1	A	7650928.0	대한민국

Pandas – 데이터 값 변경

- 조건으로 값을 변경할 데이터를 선택 후, 값 변경

```
1 condition = df['이름'] == '쯔위'
2 df.loc[condition]
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수	국적
15	쯔위	트와이스	JYP	여자	1999-06-14	172.0	A	NaN	대한민국

```
1 df.loc[condition, '국적'] = '대만'
2 df.tail()
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수	국적
11	차은우	아스트로	판타지오	남자	1997-03-30	183.0	B	3506027.0	대한민국
12	백호	뉴이스트	플레디스	남자	1995-07-21	175.0	AB	3301654.0	대한민국
13	JR	뉴이스트	플레디스	남자	1995-06-08	176.0	O	3274137.0	대한민국
14	슈가	방탄소년단	빅히트	남자	1993-03-09	174.0	O	2925442.0	대한민국
15	쯔위	트와이스	JYP	여자	1999-06-14	172.0	A	NaN	대만

Pandas – 통계값

- 통계값은 데이터 타입이 int형 또는 float형 인 열을 다룬다.

```
1 df['키'].min()  # 최솟값
```

```
162.1
```

```
1 df['키'].max()  # 최대값
```

```
183.0
```

```
1 df['키'].sum() # 합
```

```
2285.3
```

Pandas – 통계값

- .mean() – 평균값, Nan값 포함 하지 않음

```
1 df['키'].describe()
```

```
count     14.000000
mean     175.521429
std        5.683333
min      162.100000
25%      173.700000
50%      176.500000
75%      178.900000
max      183.000000
Name: 키, dtype: float64
```

```
1 df['키'].sum() / len(df[df['키'].notnull()])
```

```
175.52142857142857
```

```
1 df['키'].mean()
```

```
175.52142857142857
```

Pandas – 통계값

- .describe() – 열 값이 문자열인 경우

```
1 df['소속사'].describe()
```

```
count      16
unique     10
top       빅히트
freq        5
Name: 소속사, dtype: object
```

- 총 16개의 행
- 총 10개의 유일 값(중복 제거)
- '빅히트' 라는 값이 5개로 제일 많음

Pandas – 통계값

- .describe() – 모든 열

```
1 df.describe(include='all')
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수	국적
count	16	15	16	16	16	14.000000	16	1.500000e+01	16
unique	16	9	10	2	16	NaN	4	NaN	2
top	지민	방탄소년단	빅히트	남자	1995-10-13	NaN	A	NaN	대한민국
freq	1	5	5	12	1	NaN	8	NaN	15
mean	NaN	NaN	NaN	NaN	NaN	175.521429	NaN	5.655856e+06	NaN
std	NaN	NaN	NaN	NaN	NaN	5.683333	NaN	2.539068e+06	NaN
min	NaN	NaN	NaN	NaN	NaN	162.100000	NaN	2.925442e+06	NaN
25%	NaN	NaN	NaN	NaN	NaN	173.700000	NaN	3.712344e+06	NaN
50%	NaN	NaN	NaN	NaN	NaN	176.500000	NaN	4.668615e+06	NaN
75%	NaN	NaN	NaN	NaN	NaN	178.900000	NaN	7.862214e+06	NaN
max	NaN	NaN	NaN	NaN	NaN	183.000000	NaN	1.052326e+07	NaN

Pandas – 통계값

- .count() – 행 개수 세기

```
1 df['키'].count()
```

```
14
```

- .median() – 중앙값, .describe()의 50% 값

```
1 df['키'].median()
```

```
176.5
```

❗ n이 짝수이면 중앙에 위치한 2개의 값을 산술평균 함

- .mode() – 최빈값, 가장 많이 있는 데이터

```
1 df['키'].mode()
```

```
0    178.0
Name: 키, dtype: float64
```

```
1 df['키'].sort_values()
```

```
4     162.1
9     167.1
15    172.0
0     173.6
14    174.0
12    175.0
13    176.0
1     177.0
3     178.0
5     178.0
8     179.2
2     180.0
6     182.3
11    183.0
7       NaN
10      NaN
Name: 키, dtype: float64
```

Pandas – Pivot Table

- 열 데이터를 가지고 내가 원하는 테이블을 조합하는 기능
- .pivot_table(DataFrame, index=‘’, columns=‘’, values=‘’)

```
1 pi_df = pd.pivot_table(df, index='소속사', columns='성별', values='키')
2 pi_df
```

성별	남자	여자
소속사		
JYP	NaN	172.0
RBW	NaN	162.1
YG	177.000000	NaN
빅히트	176.560000	NaN
스타크루이엔티	167.100000	NaN
커넥트	180.000000	NaN
판타지오	183.000000	NaN
플레디스	177.766667	NaN

! 데이터가 모두 Nan 인 경우 행을 표시 하지 않음

Pandas – Pivot Table

- 옵션 : aggfunc=np.sum (기본값: np.mean)

```
1 import numpy as np
2 pd.pivot_table(df, index='소속사', columns='성별', values='키', aggfunc=np.sum)
```

성별	남자	여자
소속사		
JYP	NaN	172.0
RBW	NaN	162.1
SM	NaN	0.0
YG	177.0	NaN
빅히트	882.8	NaN
스타크루이엔티	167.1	NaN
커넥트	180.0	NaN
큐브	NaN	0.0
판타지오	183.0	NaN
플레디스	533.3	NaN

Pandas – GroupBy

- .groupby('열이름') – 그룹으로 묶기

Groupby 와 함께

- count() – 갯수
- sum() – 합계
- mean() – 평균
- var() – 분산
- std() – 표준편차
- min() / max() – 최소값, 최대값

```
1 df.groupby('혈액형')['키'].max()
```

```
혈액형
A     180.0
AB    178.0
B     183.0
O     182.3
Name: 키, dtype: float64
```

```
1 df.groupby('소속사').count()
```

소속사	이름	그룹	성별	생년월일	키	혈액형	브랜드평판지수	국적
JYP	1	1	1	1	1	1	0	1
RBW	1	1	1	1	1	1	1	1
SM	1	1	1	1	0	1	1	1
YG	1	1	1	1	1	1	1	1

Pandas – Multi Index

- 행 인덱스를 복합적으로 구성 할 수 있다.

```
1 df.groupby(['혈액형', '성별']).mean()
```

		키	브랜드평판지수
혈액형	성별		
A	남자	175.140	7591755.20
	여자	167.050	5784794.50
AB	남자	176.500	5687577.50
B	남자	183.000	3506027.00
	여자	NaN	4668615.00
O	남자	177.875	3939919.75

Pandas – fillna

- .fillna(채울 값) – nan 값을 지정한 값으로 채운다

```
1 df2 = df.copy()
2 df2['키']
```

```
0     173.6
1     177.0
2     180.0
3     178.0
4     162.1
5     178.0
6     182.3
7       NaN
8     179.2
9     167.1
10      NaN
11    183.0
12    175.0
13    176.0
14    174.0
15    172.0
Name: 키, dtype: float64
```

```
1 df2['키'].fillna(-1, inplace=True)
2 df2['키']
```

```
0     173.6
1     177.0
2     180.0
3     178.0
4     162.1
5     178.0
6     182.3
7      -1.0
8     179.2
9     167.1
10     -1.0
11    183.0
12    175.0
13    176.0
14    174.0
15    172.0
Name: 키, dtype: float64
```

```
1 df2['키'] = df2['키'].fillna(-1)
2 df2['키']
```

```
0     173.6
1     177.0
2     180.0
3     178.0
4     162.1
5     178.0
6     182.3
7      -1.0
8     179.2
9     167.1
10     -1.0
11    183.0
12    175.0
13    176.0
14    174.0
15    172.0
Name: 키, dtype: float64
```

Pandas – dropna

- .dropna(옵션) – Nan 값이 있는 행 또는 열을 통째로 제거
- 결과는 대입해야 적용됨

- axis = 0 – 행 축 (기본값)
- axis = 1 – 열 축
- how = ‘any’ – nan값이 하나라도 있으면 (기본값)
- how = ‘all’ – 모두 nan값이면

```
1 df2.dropna()
```

```
1 df2.dropna(how='any')
```

```
1 df2.dropna(axis=1)
```

```
1 df2.dropna(axis=1, how='any')
```

Pandas – 중복값 삭제

- .drop_duplicates(옵션)
- 결과는 대입해야 적용됨

- keep = ‘first’ – 첫번째 값을 남김 (기본값)
- keep = ‘last’ – 마지막 값을 남김

```
1 df['혈액형'].drop_duplicates()
```

```
0     A
3    AB
6     O
7     B
Name: 혈액형, dtype: object
```

```
1 df['혈액형'].drop_duplicates(keep='last')
```

```
11     B
12    AB
14     O
15     A
Name: 혈액형, dtype: object
```

```
1 df.drop_duplicates('혈액형')
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수	국적
0	지민	방탄소년단	빅히트	남자	1995-10-13	173.6	A	10523260.0	대한민국
3	뷔	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501.0	대한민국
6	민현	뉴이스트	플레디스	남자	1995-08-09	182.3	O	4989792.0	대한민국
7	소연	아이들	큐브	여자	1998-08-26	NaN	B	4668615.0	대한민국

Pandas – 행/열 삭제

- .drop(행/열 인덱스, 옵션)
- 결과는 대입해야 적용됨

- axis = 0 – 행 삭제
- axis = 1 – 열 삭제, ❗열 삭제시 필수

```
1 df.drop(0)
```

	이름	그룹	소속사	성별	생년월일	키	혈액형	브랜드평판지수	국적
1	지드래곤	빅뱅	YG	남자	1988-08-18	177.0	A	9916947.0	대한민국
2	강다니엘	NaN	커넥트	남자	1996-12-10	180.0	A	8273745.0	대한민국
3	뷔	방탄소년단	빅히트	남자	1995-12-30	178.0	AB	8073501.0	대한민국
4	화사	마마무	RBW	여자	1995-07-23	162.1	A	7650928.0	대한민국

```
1 df.drop(['그룹','소속사'], axis=1)
```

	이름	성별	생년월일	키	혈액형	브랜드평판지수	국적
0	지민	남자	1995-10-13	173.6	A	10523260.0	대한민국
1	지드래곤	남자	1988-08-18	177.0	A	9916947.0	대한민국
2	강다니엘	남자	1996-12-10	180.0	A	8273745.0	대한민국
3	뷔	남자	1995-12-30	178.0	AB	8073501.0	대한민국