15장 파이썬을 이용한 데이터과학

학습 목표

- 데이터 과학의 개념을 살펴본다.
- 데이터 과학의 응용 분야를 살펴본다.
- 판다스의 각 기능을 간단히 살펴본다.
- 실제 CSV 파일을 읽어서 분석해본다.

이번 장에서 만들 프로그램

- 타이타닉 승객 파일에서 여러 가지 정보를 추출해본다. 예를 들어서 승객 중에서 최고령자가 누구였을까?

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
1	0	3	Braund, Mr. Ov	male	22	1	0
2	1	1	Cumings, Mrs.	female	38	1	0
3	1	3	Heikkinen, Mis	female	26	0	0
4	1	1	Futrelle, Mrs. J	female	35	1	0
5	0	3	Allen, Mr. Willi	male	35	0	0
6	0	3	Moran, Mr. Jan	male		0	0

데이터 과학이란

- 카드 결제 데이터나 택배 송장 데이터를 이용하여 장사가 잘 되는 지역을 찾을 수 있을까?
- 산악기상 데이터나 등산로 등의 정보를 이용하여 산림재해를 예측할 수 있을까?
- CCTV가 적게 설치된 곳에서 실제로 범죄가 많이 일어날까?
- 20-30대가 많이 사는 지역의 커피 샵이 더 많은 매출을 올리고 있을까?
- 지하철 승하차가 가장 많이 발생하는 역은 어떤 역일까?
- 1년 중에서 일교차가 가장 심했던 달은 어떤 달이었을까?

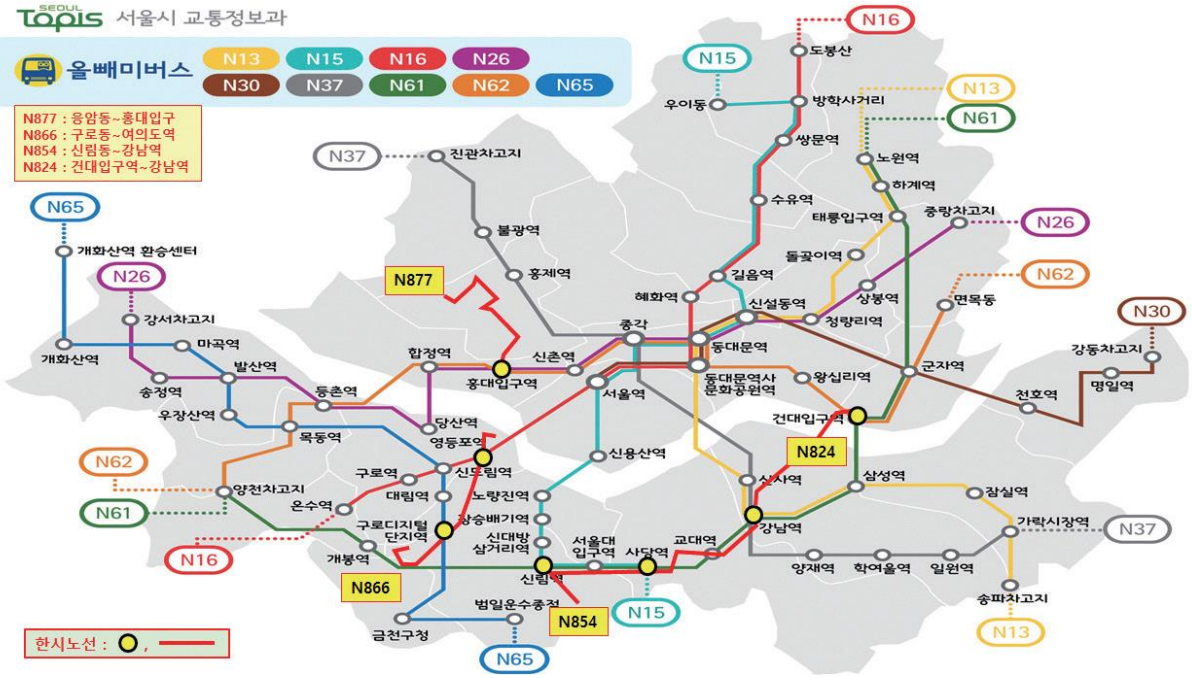
- 데이터 과학(data science) : 데이터에서 정보나 지식을 추출하는 학문

데이터 과학이란

- 데이터 과학은 여러 학문 분야에 걸친 접근 방식(통계학, 컴퓨터 과학, 기계 학습 등의 많은 분야에서 추출한 기법과 이론)을 필요로 한다.
- 파이썬은 수많은 라이브러리를 가지고 있으며 데이터 과학의 요구를 쉽게 처리할 수 있는 기능을 내장하고 있는 동시에, 범용 프로그래밍 언어이기 때문에 최근에 데이터 과학 언어로 각광을 받고 있다

데이터 과학으로 무엇을 하는가

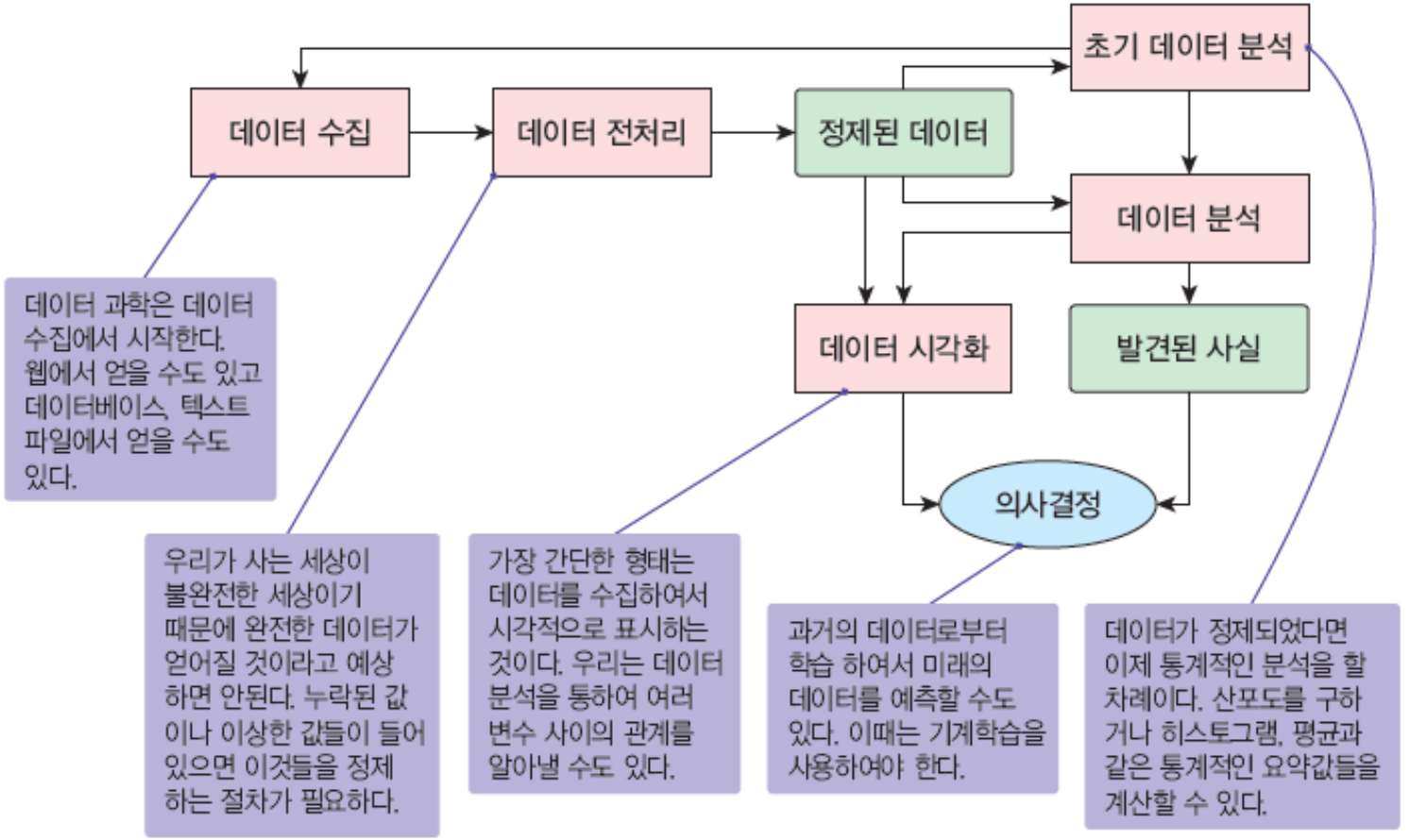
- 서울시. 심야 버스 노선을 설계 - 심야 택시 승하차 데이터와 한 달간 자정부터 새벽 5시까지의 KT의 통화량 데이터 30억 건을 분석하여 지도상에서 사람들이 심야에 어디에서 어디로 가장 많이 이동하는지를 파악하였다.



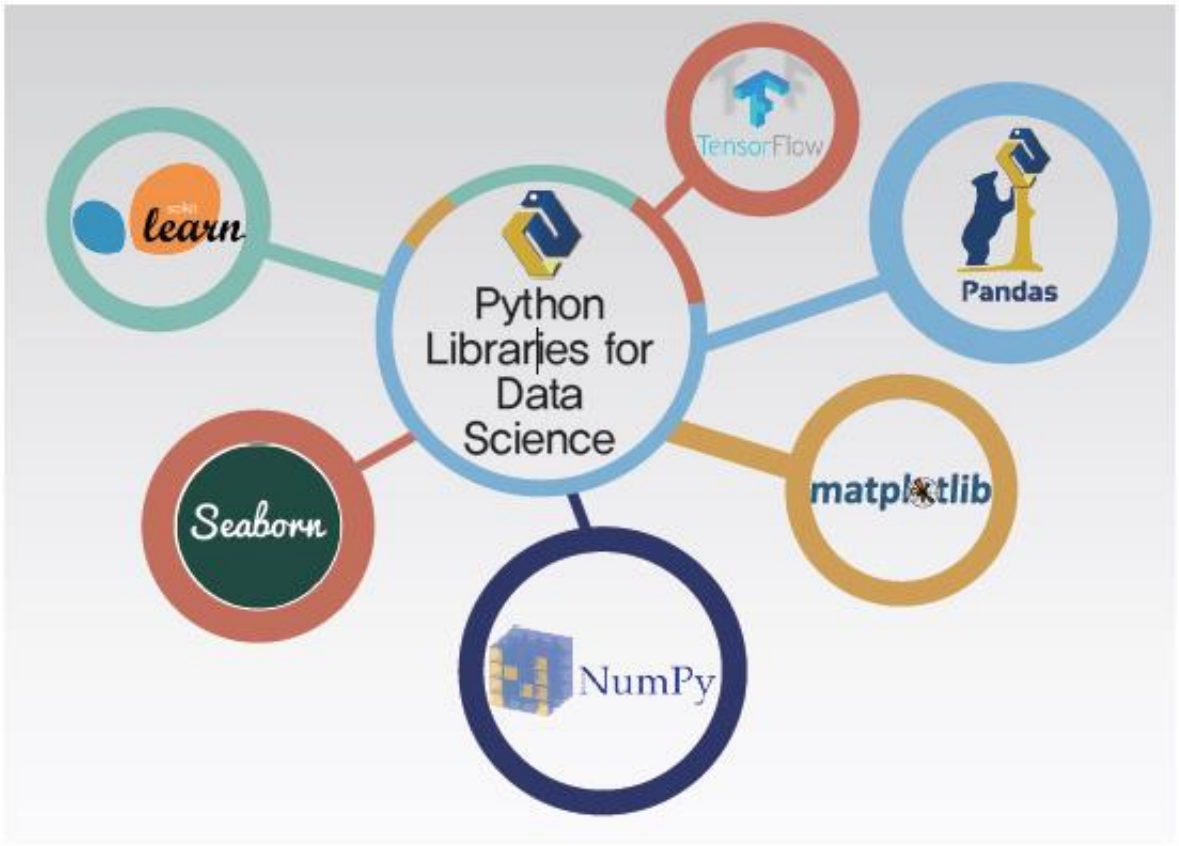
데이터 과학의 응용

- 추천 시스템 – 구매자 요구를 예측하는 모델을 작성하고 구매자가 구매할 가능성이 높은 제품을 보여주는 시스템
- 재무 위험 관리 – 불량 부채를 피함으로써 신용과 관련된 손실 최소화
- 주식 거래 – 과거의 주가 정보와 새로운 정보를 이용하여 미래의 주가를 예측

데이터 처리 절차


초기 데이터 분석
데이터 수집
데이터 전처리
정제된 데이터
데이터 분석
데이터 시각화
발견된 사실
의사결정
데이터 과학은 데이터 수집에서 시작한다. 웹에서 얻을 수도 있고 데이터베이스, 텍스트 파일에서 얻을 수도 있다.
우리가 사는 세상이 불완전한 세상이기 때문에 완전한 데이터가 얻어질 것이라고 예상하면 안된다. 누락된 값이나 이상한 값들이 들어 있으면 이것들을 정제하는 절차가 필요하다.
가장 간단한 형태는 데이터를 수집하여서 시각적으로 표시하는 것이다. 우리는 데이터 분석을 통하여 여러 변수 사이의 관계를 알아낼 수도 있다.
과거의 데이터로부터 학습 하여서 미래의 데이터를 예측할 수도 있다. 이때는 기계학습을 사용하여야 한다.
데이터가 정제되었다면 이제 통계적인 분석을 할 차례이다. 산포도를 구하거나 히스토그램, 평균과 같은 통계적인 요약값들을 계산할 수 있다.

데이터 과학을 위한 파이썬 라이브러리



판다스

- 강력한 데이터 구조를 사용하여 고성능 데이터 조작 및 데이터 분석에 사용되는 오픈 소스 파이썬 라이브러리
- R과 같은 통계 분석용 언어로 전환하지 않고도 파이썬에서 전체 데이터 분석 과정을 수행할 수 있다.
- R 언어가 제공하는 시리즈(series)와 데이터 프레임(Data Frame)을 파이썬에 추가한다. -> 테이블 처리

판다스로 할 수 있는 작업

- 파이썬 리스트, 딕셔너리, 넘파일 배열을 데이터 프레임으로 변환할 수 있다.
- 판다스로 CSV 파일이나 TSV 파일, 엑셀 파일 등을 열 수 있다.
- URL을 통해 웹 사이트의 CSV 또는 JSON과 같은 원격 파일 또는 데이터베이스를 열 수 있다.
- mean()로 모든 열의 평균을 계산할 수 있다.
- corr()로 데이터 프레임의 열 사이의 상관 관계를 계산할 수 있다.
- 조건을 사용하여 데이터를 필터링할 수 있다.
- sort_values()로 데이터를 정렬할 수 있다.
- groupby()를 이용하여 기준에 따라 몇 개의 그룹으로 데이터를 분할할 수 있다.
- 데이터의 누락 값을 확인할 수 있다.
- 특정한 값을 다른 값으로 대체할 수 있다.

판다스의 데이터 구조

- 시리즈(Series): 1차원 배열. 크기 변경 불가

11	73	53	27	52	65	74	98	13	72

- 데이터 프레임(DataFrame): 2차원 배열(테이블). 크기 변경 가능

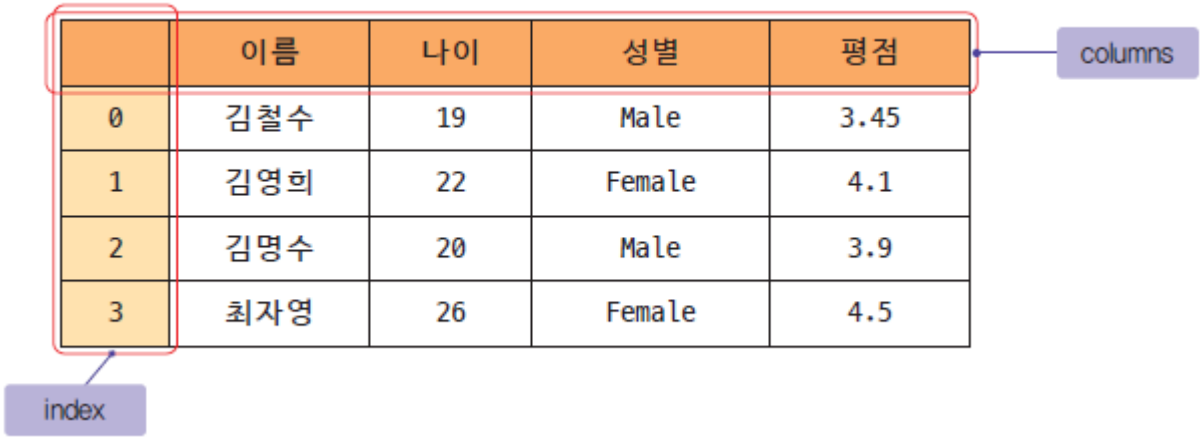
	이름	나이	성별	평점
0	김철수	19	Male	3.45
1	김영희	22	Female	4.1
2	김명수	20	Male	3.9
3	최자영	26	Female	4.5

행(row)

열(column)

index와 columns 객체

- 데이터 프레임에서는 행이나 열에 붙인 레이블을 중요시한다.
- index 객체 : 행들의 레이블(label)
- columns 객체 : 열들의 레이블



	이름	나이	성별	평점
0	김철수	19	Male	3.45
1	김영희	22	Female	4.1
2	김명수	20	Male	3.9
3	최자영	26	Female	4.5

15.4 판다스 맛보기

- titanic.csv : 타이타닉 탑승자에 대한 데이터셋

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
1	0	3	Braund, Mr. Ov	male	22	1	0
2	1	1	Cumings, Mrs.	female	38	1	0
3	1	3	Heikkinen, Mis	female	26	0	0
4	1	1	Futrelle, Mrs. J	female	35	1	0
5	0	3	Allen, Mr. Willi	male	35	0	0
6	0	3	Moran, Mr. Jan	male		0	0

15.4 판다스 맛보기

- PassengerId: 승객의 ID이다.
- Survived: 생존 여부
- Pclass: 탑승 등급을 나타낸다. 클래스 1, 클래스 2, 클래스 3의 3가지 클래스가 있다.
- Name: 승객의 이름.
- Sex: 승객의 성별.
- Age: 승객의 나이.
- SibSp: 승객에게 형제 자매와 배우자가 있음을 나타낸다.
- Parch: 승객이 혼자인지 또는 가족이 있는지 여부.
- Ticket: 승객의 티켓 번호.
- Fare: 운임.
- Cabin : 승객의 선실.
- Embarked: 탑승한 지역.

15.4 판다스 맛보기

- 예제 코드 보기. p.653 – p.655

15.5 데이터 프레임 생성하기

- 예제 코드 보기. p.655 – p.659

Lab: 데이터 프레임 만들어 보기

countries.csv

```
code,country,area,capital,population
KR,Korea,98480,Seoul,48422644
US,USA,9629091,Washington,310232863
JP,Japan,377835,Tokyo,127288000
CN,China,9596960,Beijing,1330044000
RU,Russia,17100000,Moscow,140702000
```

```
import pandas as pd
countries = pd.read_csv('countries.csv')
countries
```

```
----
code country      area     capital  population
0   KR   Korea     98480      Seoul    48422644
1   US     USA   9629091  Washington   310232863
2   JP   Japan    377835       Tokyo   127288000
3   CN   China   9596960     Beijing  1330044000
4   RU  Russia  17100000      Moscow   140702000
```

15.6 원하는 데이터 선택하기

- 여러가지 쿼리들을 다루는 예제 코드 보기. p.660 – p.664

15.7 행과 열의 추가나 삭제

- 예제 코드 보기. p.664 – p.666

15.8 데이터 통계

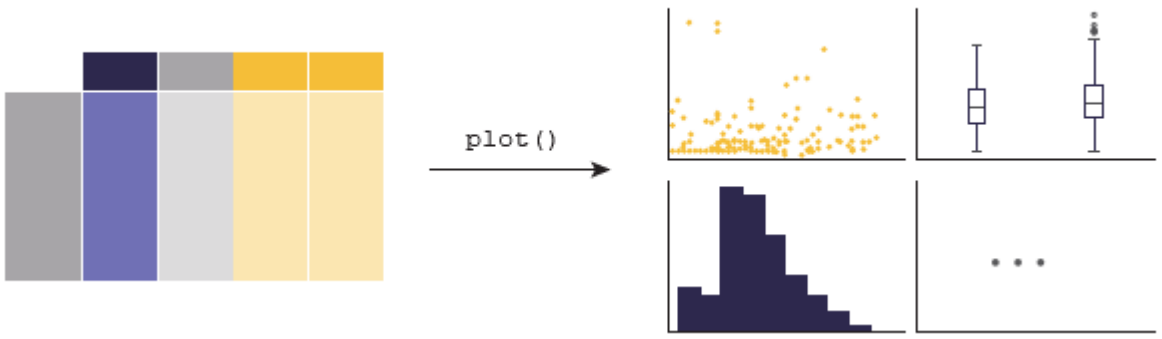
- 예제 코드 보기. p.667 – p.668

15.8 데이터 통계

- 예제 코드 보기. p.667 – p.668

15.9 데이터로 챠트 그리기

- df.plot()와 같이 호출하면 인덱스에 대하여 모든 열을 그리다.
- df.plot(x='col1')와 같이 호출하면 하나의 열만을 그린다.
- df.plot(x='col1', y='col2')와 같이 호출하면 특정 열에 대하여 다른 열을 그리게 된다.



- 예제 코드 보기. p.669 – p.672

15.10 테이블의 레이아웃을 바꾸는 방법

- 피벗 테이블 : 엑셀. 데이터 집계표 기능
- 판다스 라이브러리는 pivot_table()이라는 함수를 제공한다.

학생	과목	성적
홍길동	수학	100
홍길동	과학	95
홍길동	사회	90
최자영	수학	90
최자영	과학	96
최자영	사회	100

학생	수학	과학	사회
홍길동	100	95	90
최자영	90	95	100

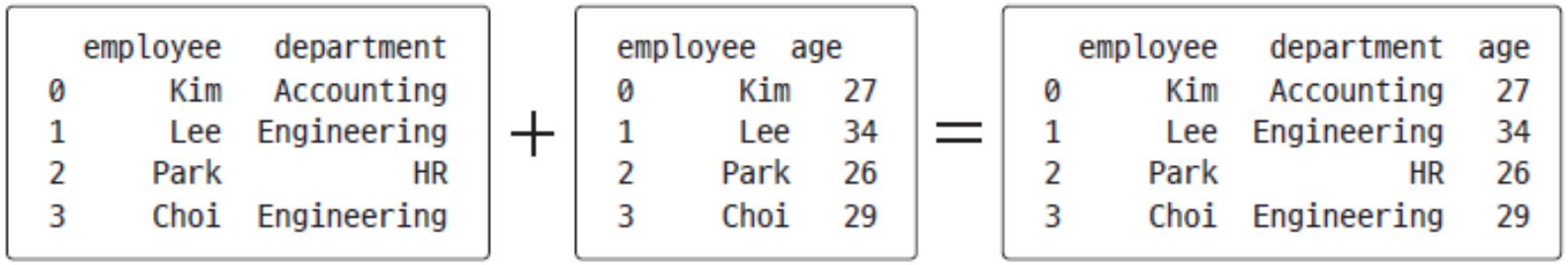
- 예제 코드 보기. p.673 – p.677

15.11 데이터 병합

- 여러 개의 데이터 테이블을 사용하여 데이터를 관리하는 것이 더 쉽고, 중복성을 피할 수 있고, 디스크 공간을 절약할 수 있다. 또 크기가 작으면 테이블을 빨리 쿼리할 수 있다.
- 병합(merging) : 여러 테이블에 나누어져 저장된 데이터들을 가져와서 합치는 작업. SQL의 JOIN 메서드와 유사

- merge()을 사용하면 공통 열이나 인덱스를 사용하여 데이터를 결합한다.
- join()을 사용하면 키 열이나 인덱스를 사용하여 데이터를 결합한다.
- concat()을 사용하면 테이블의 행이나 열을 결합한다.

merge()



	employee	department
0	Kim	Accounting
1	Lee	Engineering
2	Park	HR
3	Choi	Engineering

+

	employee	age
0	Kim	27
1	Lee	34
2	Park	26
3	Choi	29

=

	employee	department	age
0	Kim	Accounting	27
1	Lee	Engineering	34
2	Park	HR	26
3	Choi	Engineering	29

```
df1 = pd.DataFrame({'employee': ['Kim', 'Lee', 'Park', 'Choi'],
                    'department': ['Accounting', 'Engineering', 'HR', 'Engineering']})
df2 = pd.DataFrame({'employee': ['Kim', 'Lee', 'Park', 'Choi'],
                    'age': [27, 34, 26, 29]})
df3 = pd.merge(df1, df2)
```

15.12 데이터 정제

- 판다스에서는 결손값을 NaN으로 나타낸다.
- 데이터를 처리하기 전에 결손값을 처리하지 않으면 어떠한 데이터 분석도 불가능하다. 결손값은 삭제하거나 다른 값으로 교체하여야 한다.

- 예제 코드 보기. p.679 – p.680