

3. 판다스(Pandas) 데이터 분석

- 시리즈와 데이터프레임 클래스
- 시리즈와 데이터프레임의 데이터를 조작
- 데이터프레임의 병합
- Pandas를 이용한 csv 파일 처리
- 데이터를 입출력과 시각화

3.1 시리즈 클래스

- 시리즈 클래스(series)
 - 시리즈 = 값(value) + 인덱스(index)
- 생성
 - 변수 = Pandas.Series([값1(value), 값2(value)...],
index=["라벨1", "라벨2", ...])
 - 변수.index
 - 변수.values
- 인덱싱
 - 인덱스 라벨을 이용한 인덱싱과 슬라이싱
 - 자료의 순서변경 및 특정자료 선택
 - 문자열 라벨을 이용한 슬라이싱
- 시리즈와 딕셔너리 자료형
 - In연산
 - item 메서드
 - for k, v in s.items():
print("%s = %d" % (k, v))
- 인덱스 기반 연산
 - 인덱스가 같은 데이터 연산
 - Series.notnull
 - NaN이 아닌값
- 데이터의 갱신, 추가, 삭제
 - 딕셔너리와 동일
 - 인덱스를 사용

3.2 데이터 프레임 클래스

- 데이터 프레임 클래스(DataFrame)
 - 행 인덱스
 - 열 인덱스
- 생성
 - 우선 하나의 열이 되는 데이터를 리스트나 일차원 배열로 준비한다.
 - 각각의 열에 대한 이름(라벨)을 키로 가지는 딕셔너리를 만든다.
 - 데이터를 DataFrame 클래스 생성자에 넣는다.
 - 열방향 인덱스= columns
 - 행방향 인덱스= index
- 열 데이터의 갱신, 추가, 삭제
 - 열 단위로 데이터를 갱신, 추가, 삭제
- 열 데이터의 인덱싱
 - 열 라벨을 키 값으로 하여 인덱싱
 - 하나의 열만 인덱싱: 시리즈
 - 여러 개의 열을 인덱싱: 데이터프레임
 - ※열 인덱스가 문자열 라벨을 가지고 있는 경우에는 순서를 나타내는 정수 인덱스를 열 인덱싱에 사용할 수 없다.
- 행 데이터 인덱싱
 - 슬라이싱(slicing)
- 개별데이터인덱싱
 - 열라벨로 시리즈 인덱싱=>시리즈를 행 라벨로 인덱싱
 - Ex) ["열 라벨 "] [" 행 라벨 "]

❖ 데이터 프레임 고급 인덱싱

- 2차원, 행 열 인덱싱
 - loc: **라벨값** 인덱스 기준
 - iloc: **행 번호** 기준 (반드시 숫자)
- iloc 인덱서
 - 순서를 나타내는 **정수(integer)** 인덱스 값
 - 인덱싱 값이 하나인 경우, 행 선택
- loc 인덱서
 - 인덱싱 값이 하나
loc[행 인덱싱값]
 - 인덱싱 값, 행과 열
loc[행 인덱싱값, 열 인덱싱값]

인덱싱 값

- 인덱스 데이터
- 인덱스 데이터 슬라이스: 또는 리스트
- 같은 행 인덱스를 가지는 불리언 시리즈 또는 함수(행 인덱싱의 경우)

Ex) df.loc[df.A > 10, ["C", "D"]]

※원래 행 인덱스값이 정수인 경우, 라벨 슬라이싱 방식을 따름(마지막 값 포함)

❖ 데이터 프레임의 데이터 조작

- 데이터 개수 세기
 - Count()
- 카테고리 값 세기
 - value_counts()
 - 데이터프레임: 각 열마다 별도 적용
- 정렬
 - sort_index, sort_values
 - ascending=False : 내림차순 정렬
 - 데이터프레임은 기준열을 지정
 - 변수.sort_values(by='기준열인덱스', 정렬순서)
 - 변수[열인덱스].sort_values()
- 행/열 합계
 - sum(axis인수)
 - axis=0 (생략가능) 열 방향 합계
 - axis=1 행 방향 합계
- apply 변환
 - 행이나 열 단위의 복잡한 처리
 - 행이나 열을 받는 함수(lambda식)를 apply의 인수로 사용
 - Lambda 인자 : 표현식
 - If문의 경우 반환 값을 먼저 기술
 - If와 else만 지원하고 else if는 지원하지 않음
- fillna 메서드
 - NaN 값을 원하는 값으로 변환
- astype 메서드
 - 전체 데이터의 자료형을 변경

❖ 데이터 프레임의 합성

- Merge함수

- 공통 열 또는 인덱스를 기준으로 병합
- 양 쪽에 모두 키가 존재하는 데이터 포함.
(inner join)
- 인수how='outer': 한쪽에만 키가 존재하는 데이터 포함.
(outer join)
- 인수how='left': 첫번째 DF의 키 값 기준
- 인수how='right': 두번째 DF의 키 값 기준
- 중복된 키 값이 존재하는 경우:
모든 경우의 수를 따져서 조합을 만듦
- 기준 열 설정 인수 on
- 기준 열의 이름이 다른 경우 left_on, right_on
- 인덱스를 기준열로 사용 left_index 또는 right_index를
True로 설정

- Concat함수

- 기준 열(key column)을 사용하지
않고 단순히 데이터를 연결
- 위/아래로 데이터 행을 연결
- 옆으로 데이터 열 연결: axis=1인수

3.3 데이터 입출력

- Pandas 지원 데이터 포맷
 - txt, CSV, Excel, HTML, JSON, SQL 등
- CSV파일 입력
 - `pandas.read_csv`
 - `names` : 열 인덱스 정보 추가
 - `Index_col`: 특정열을 행 index로 지정
 - `sep` :구분자, default는 쉼표.
`sep='\\s+'`(가변공백) `sep='\\t'`(tab), `sep=';'`
 - `skiprows`: 특정 행을 skip
 - `Na_values`: 특정값을 NaN값으로 치환
 - `Skip_blank_lines`: 빈 줄을 skip
 - `Nrows`:읽을 행의 개수
- CSV파일 출력
 - `to_csv`
 - `na_rep`: NaN 표시값 변경
 - `Index, header`:인덱스 및 헤더 출력여부
- 인터넷 상의 CSV파일 입력
 - `read_csv(파일url)`
 - `read_html` :웹페이지 크롤링
 - `pandas_datareader`패키지
 - 데이터 제공 사이트 : FRED, World Bank, OECD 등등
 - <https://pandas-datareader.readthedocs.io/en/latest/index.html>
 - `pip install -upgrade pandas-datareader`
 - Ex) FRED데이터 베이스
 - 미국 국가총생산(GDP)
 - 소비자 가격 지수(CPIAUCSL)
 - `pip install -U finance-datareader`: 국내 주식데이터
 - <https://financedata.github.io/posts/finance-data-reader-users-guide.html>

3.4 데이터 시각화

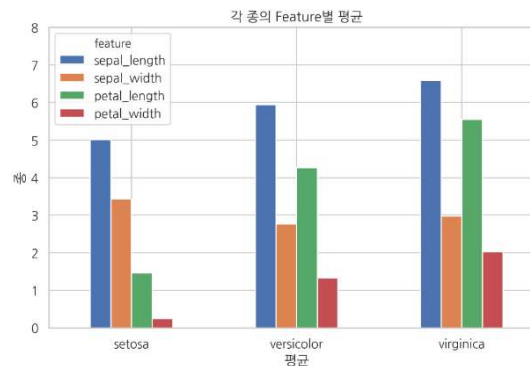
- 시각화 패키지 맷플롯리브 (Matplotlib)
 - 라인 플롯(line plot) , 바 차트(bar chart), 히스토그램(histogram), 박스 플롯(box plot)
 - <http://matplotlib.org/gallery.html>
 - `import matplotlib as mpl`
- Pyplot 서브 패키지
 - 맷랩(matlab)의 시각화 명령을 그대로 사용 가능
 - `import matplotlib.pyplot as plt`
- 라인플롯
 - 선을 그리는 라인 플롯
 - 시간, 순서 등에 따른 변화를 표시
 - title :제목표시
 - show: 시각화 명령을 차트로 렌더링
- 스타일 지정
 - https://matplotlib.org/stable/api/matplotlib_configuration_api.html
 - 문자열 인수를 사용
 - 색깔, 마커, 선 종류의 순서로 지정
 - 색깔
 - 색 이름 혹은 약자(blue b, green g 등등), #RGB코드
 - 마커: 데이터 위치를 나타내는 기호
 - 선 종류
 - - solid line style
 - -- dashed line style
 - -. dash-dot line style
 - : dotted line style
 - 기타 스타일
 - Color c , Linewidth lw , Linestyle ls
 - Marker (마커 종류), markersize Ms(마커 크기)
 - Markeredgecolor Mec (마커 선 색깔), markeredgewidth mew(마커 선 굵기)
 - Markerfacecolor mfc (마커 내부 색깔)

3.5 Pandas의 시각화 기능

- plot 메서드

- matplotlib를 내부에서 임포트하여 사용
- kind인수: 플롯의 종류 설정

Ex) Plot(kind='bar', rot=0) or plot.bar(rot=0)



- Seaborn을 사용한 시각화

<http://seaborn.pydata.org/>

