Pandas를 이용한 데이터 준비 (추가함수)

Soongsil University

숭실대학교
베어드교양대학
강의선 교수
iami86@ssu.ac.kr

열이름 변경하기

- 데이터에서 열이름 변경하기
  - **변수명.rename(columns = { '열이름' : '새로운 열이름' }, inplace= True)**
    # 데이터가 저장된 변수명의 열이름을 새로운 열이름으로 변경
    # inplace = True 옵션은 원본데이터를 변경함

```
data.rename(columns={'합계.1':'검거합계'}, inplace=True)
data.head()
```

```
data.rename(columns={'합계.1':'검거합계', '살인.1':'살인검거','강도.1':'강도검거'}, inplace=True)
data.head()
```


행과 열 삭제하기

- 행 데이터 삭제
  - **변수명.drop(index=행번호, axis=0) : index가 '행번호' 인 행 삭제**
    # 여러행 삭제 : 변수명.drop(index=[0,1,2], axis=0) : index가 0,1,2인 행(3줄) 삭제
    # inplace= True 옵션을 추가하면 원본을 변경함

```
data.drop(index=0, axis=0)
data.drop(index=[1,2], axis=0, inplace=True)
```

- 열 데이터 삭제
  - **변수명.drop(colums=['열이름'], axis=1) : '열이름' 열 삭제**
    # 여러열 삭제 : 변수명.drop(colums=['열이름1','열이름2'], axis=1) : '열이름1','열이름2'열 삭제
    # inplace= True 옵션을 추가하면 원본을 변경함

```
data.drop(columns=['기간'], axis=1)
data.drop(columns=['절도.1','폭력.1'], axis=1, inplace=True)
```


index 번호 재설정

- 인덱스 리셋
  - **변수명.reset_index(drop=True, inplace=True)**
    #drop=True 옵션은 기존 인덱스는 버리고 새로 인덱스 설정

```
1 data.reset_index()
```

	index	기간	자치구	합계	검거합계	살인
0	3	2019	중구	4327	2804	2
1	4	2019	용산구	3313	2611	3
2	5	2019	성동구	2512	1838	6

```
1 data.reset_index(drop=True, inplace=True)
2 data
```

	기간	자치구	합계	검거합계	살인	살인검거	강
0	2019	중구	4327	2804	2	1	
1	2019	용산구	3313	2611	3	3	
2	2019	성동구	2512	1838	6	5	

데이터 정렬하기

- 데이터 정렬하기
  - **변수명.sort_values(by= '정렬기준 열이름' , ascending=True)**
    # ascending = True : 오름차순, False : 내림차순

```
data.sort_values(by='합계', ascending=True)
```

	기간	자치구	합계	검거합계	살인	살인검거	강도	강도검거
8	2019	도봉구	2110	1497	1	1	5	5
2	2019	성동구	2512	1838	6	5	9	10
6	2019	성북구	2877	2323	3	3	3	3
11	2019	서대문구	2943	2020	2	1	5	5

```
data.sort_values(by='합계', ascending=False)
```

	기간	자치구	합계	검거합계	살인	살인검거	강도
21	2019	강남구	7304	5069	5	3	5
17	2019	영등포구	5820	3787	10	10	3
22	2019	송파구	5698	3799	7	8	10
20	2019	서초구	5542	3750	5	5	7

Soongsil University

데이터 열 연산하기

- 데이터 열 연산하기
  - **변수명['열이름'] = 변수명['열이름'] + 변수명['열이름 ']**
    #사칙연산 가능
    #해당 열이름이 없으면 새로운 열 생성

```
#data가 숫자가 아닌 경우에는 to_numeric()을 이용하여 숫자로 선변환하기
data['살인'] = pd.to_numeric(data['살인'])
data['살인검거'] = pd.to_numeric(data['살인검거'])

data['살인검거율'] = data['살인검거']/data['살인'] *100
data
```

	기간	자치구	합계	검거합계	살인	살인검거	강도	강도검거	강간강제추행	강간강제추행.1	절도	폭력	살인검거율
0	2019	중구	4327	2804	2	1	6	5	195	115	2202	1922	50.000000
1	2019	용산구	3313	2611	3	3	3	4	272	237	999	2036	100.000000
2	2019	성동구	2512	1838	6	5	9	10	133	96	970	1394	83.333333
3	2019	광진구	4011	2816	4	5	6	5	273	213	1875	1853	125.000000

Soongsil University

서울시 구별 CCTV 현황 파악하기

Soongsil University

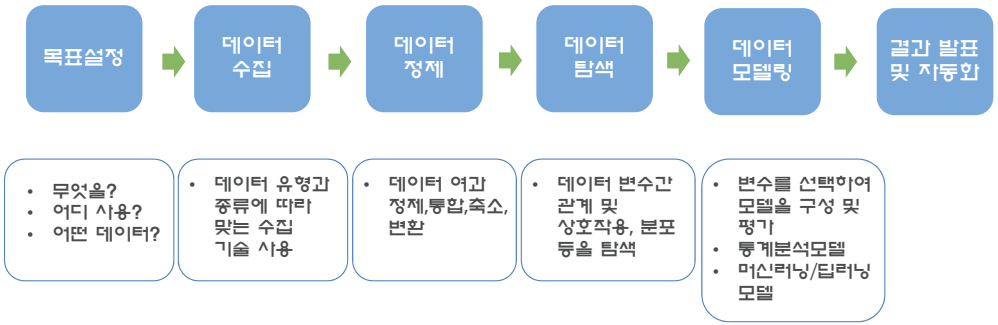
베어드교양대학
강의선
백마관 203호
02-828-7264
iami86@ssu.ac.kr

목차

- 공공데이터를 읽어와 살펴보기
  - 서울시 자치구별 CCTV 현황 데이터
- pandas를 이용한 데이터 관리 및 정제
- matplotlib을 이용한 데이터 분석 및 시각화

Soongsil University

빅데이터 분석의 과정(온라인)



1. 데이터 읽어오기

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

- 주요함수
  - 데이터 읽어오기

```
# 인코딩방식 :'cp949'(MS office에서 저장한 파일 형식) /'utf-8'(그 외 일반적인 경우)

# ","로 분리된 .csv 파일을 불러올 때
✓변수명 = pd.read_csv( '파일경로명' , delimiter= ',' , encoding= '인코딩방식' )
   • delimiter 옵션은 생략하면 ',' 로 인식
   • header = 숫자 옵션은 위에서 몇 째줄 부터 읽어올지 지정(줄 수는 0부터 시작)

# "tab"으로 분리된 .txt 파일을 불러올 때
✓변수명 = pd.read_csv( '파일경로명' , delimiter = '\t' , encoding= '인코딩방식' )

# .xlsx 파일을 불러올 때
✓변수명 = pd.read_excel( '파일경로명' )
   • header = 숫자  옵션은 위에서 몇 째줄 부터 읽어올지 지정(줄 수는 0부터 시작)
```


2. 데이터 살펴보기

- 주요 함수
  - 데이터에서 일부 내용 보기

```
✓변수명 : 전체 데이터 보기
✓변수명.head() : 위에서 5행 보기 / 변수명.head(3) : 위에서 3행 보기
✓변수명.tail() : 아래서 5행 보기 / 변수명.tail(3) : 아래서에서 3행 보기
✓변수명[ : ] : 원하는 행부터 원하는 행까지 보기
✓변수명[ '' ] : 원하는 열 데이터 보기
 # 여러열 선택 : 변수명[[ '열이름1' , '열이름2' ]]
✓변수명[ '' ][ : ] : 원하는 열의 특정 행 보기
```

  - 데이터 정보 보기

```
✓변수명.describe() : 숫자형 데이터의 통계치 계산
✓변수명.info() : 데이터 타입, 각 아이템 개수, 누락데이터 수 등 확인
```


3. 데이터 정리하기

- 추가 주요 함수
  - 데이터에서 열이름 변경하기

```
✓변수명.rename(columns = { '열이름' : '새로운 열이름' }, inplace= True)
# 데이터가 저장된 변수명의 열이름을 새로운 열이름으로 변경
# inplace = True 옵션은 원본데이터를 변경함
```

  - 행 데이터 삭제

```
✓변수명.drop(index= '행번호' , axis=0) : index가 0인 행 삭제
# 여러행 삭제 : 변수명.drop(index=[0,1,2], axis=0) : index가 0,1,2인 행(3줄) 삭제
# inplace= True 옵션을 추가하면 원본을 변경함
```

  - 열 데이터 삭제

```
✓변수명.drop(colums=[ '열이름' ], axis=1) : '열이름' 열 삭제
# 여러열 삭제 : 변수명.drop(colums=[ '열이름1' ,' 열이름2' ], axis=1) : '열이름1' ,' 열이름2' 열 삭제
# inplace= True 옵션을 추가하면 원본을 변경함
```

  - 인덱스 리셋

```
✓변수명.reset_index(drop=True, inplace=True)
#drop=True 옵션은 기존 인덱스는 버리고 새로 인덱스 설정
```

4. 데이터 자세히 보기(CCTV 현황)

- CCTV의 전체 개수가 가장 적은/가장 많은 상위 5개 구는 어디일까?
  - ✓변수명.sort_values(by='정렬기준 열이름', ascending=True)
  # ascending = True : 오름차순, False : 내림차순
- 최근 3년간 CCTV 증가율을 계산하여 '최근증가율'을 알아보자.
  - ✓변수명['열이름'] + 변수명['열이름'] : 사칙연산 가능
  - ✓변수명['최근증가율'] = 열단위 연산식 : 해당 열이름이 없으면 새로운 열 생성
- 최근증가율이 가장 높은 상위 5개 구는 어디일까?
  - ✓변수명.sort_values(by='최근증가율', ascending=False).head()

5. 시각화 하기

```
import matplotlib.pyplot as plt  # 다양한 그래프 기능 제공 라이브러리
```

- 자치구별 CCTV 수가 가장 많은 구는 어디인지 그래프로 나타내보자.
  - ✓plt.bar(변수명['열이름1'], 변수명['열이름2']) #막대차트
  - ✓plt.plot(변수명['열이름1'], 변수명['열이름2']) # 라인차트
  - ✓plt.plot(변수명['열이름1'], 변수명['열이름2'], marker='') # 라인차트에 마크 추가하기
  - ✓plt.plot(변수명['열이름1'], 변수명['열이름2'], label = '') # 각 레이블 추가하기
    # plt.legend() #차트의 레이블을 추가할 경우 반드시 삽입
  - ✓plt.title('텍스트') #차트 제목 삽입
  - ✓plt.xlabel('텍스트') #x축 레이블 삽입
  - ✓plt.ylabel('텍스트') #y축 레이블 삽입
  - ✓plt.grid() #차트 배경에 눈금선 삽입