

시와 데이터 기초

데이터의 정보확인

오늘 수업은

❖ 데이터 정보 확인하기

❖ 시도 시군구별 교통사고 확인하기

- 데이터 정보 확인하기
- 데이터 전처리

❖ 데이터 재설정하기

❖ 시도 시군구별 교통사고 확인하기

- 데이터 재설정

공공데이터 수집하기

❖ <https://www.data.go.kr/>



[홈페이지 바로가기](#)

데이터셋



도로교통공단_ 시도 시군구별 교통사고 통계

- 경찰에서 조사, 처리한 교통사고에 대한 통계 정보로 인적 피해가 있는 사고만 집계 됨
- 시도 및 시군구별 교통사고 사고건수 사망자수, 중상자수, 경상자수, 부상신고자수 통계
- 교통사고분석시스템(<http://taas.koroad.or.kr>)의 데이터를 바탕으로 함



0



0

관심

실습을 위해 3개의 파일로 분리해서 저장



서울_부산 교통사고 통계.xlsx



시군구별 교통사고 통계.csv



시군구별 부상신고자수.xlsx

1. 데이터 읽어오기

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

❖ *.CSV 데이터 읽어오기

- `변수명 = pd.read_csv('파일경로명' , encoding= '인코딩방식')`
 - ","로 분리된 .csv 파일을 불러올 때
 - delimiter 옵션은 생략하면 ',' 로 인식
 - 인코딩방식 : 'EUC_KR' (한글이 포함된 일반적인 경우)/ 'cp949' (MS office에서 저장한 파일 형식)
- `변수명 = pd.read_excel('파일경로명')`
 - .xlsx 파일을 불러올 때

2. 데이터 정보 확인하기

❖ 데이터 정보 보기

- 변수명.**shape** : 행과 열의 개수 확인
- 변수명.**info()** : 데이터 타입, 각 아이템 개수, 누락데이터 수 등 확인
- 변수명.**describe(include=매개변수)** : 숫자형 데이터의 통계치 계산
 - include='object': 문자열의 통계분포 확인
 - include='all': 모든 열의 통계분포 확인

3. 데이터에서 일부 내용 보기

❖ 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
 - 열이름에는 작은 따옴표(' ') 또는 큰 따옴표(" ")
- # 여러열 선택 : 변수명[['열이름1' , '열이름2']]
- 변수명[' '][:] : 원하는 열의 특정 행 보기
- 변수명['열이름'].value_counts(normalize=True, sort=False)
 - 해당열의 각 데이터의 개수 확인
 - normalize=True: 데이터가 차지하는 비율을 확인하고자 할때 사용
 - sort=False: 결과에 대한 내림차순을 적용하지 않음

4. 데이터 열 정리하기

❖ 데이터 열 연산 및 새로운 열 생성하기

- 변수명['열이름'] = 변수명['열이름'] + 변수명['열이름']
 - 사칙연산 가능
 - 해당 열이름이 없으면 새로운 열 생성
 - 단, 숫자형 데이터에 한해 연산 가능

❖ 데이터에서 열이름 변경하기

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

❖ 열 데이터 삭제

- 변수명.drop(columns=['열이름'], axis=1, inplace= True)
 - 여러열 삭제 : 변수명.drop(columns=['열이름1' , ' 열이름2'], axis=1)
 - inplace= True : 원본을 변경함

5. 데이터 행 정리하기

❖ 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
 - 변수명.drop(변수명.index[0:17], axis=0): 0~16 index행 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

데이터 재설정

질문하기(교통사고현황)

❖ 교통사고가 가장 많이 발생하는 지자체는 어디인가?

1. 사망자 수가 30명 이상이고 중상자가 800명 이상인 지자체는 어디인가?
2. 사망자 수가 50명 이상이거나 부상자수가 3000명 이상인 지자체는 어디인가?
3. 서울과 대구에서 사망자수가 가장 많은 시군구는 어디인가?
4. 사망자가 가장 많은/가장 적은 10개 지자체는 어디인가?
5. 부상자수가 가장 많은 도시는 어디인가?
6. 사망자 평균이 가장 많은 도시는 어디인가?

공공데이터 수집하기

❖ <https://www.data.go.kr/>



데이터셋



URL 복사

도로교통공단_ 시도 시군구별 교통사고 통계

- 경찰에서 조사, 처리한 교통사고에 대한 통계 정보로 인적 피해가 있는 사고만 집계 됨
- 시도 및 시군구별 교통사고 사고건수 사망자수, 중상자수, 경상자수, 부상신고자수 통계
- 교통사고분석시스템(<http://taas.koroad.or.kr>)의 데이터를 바탕으로 함



0



0

관심

실습을 위해 3개의 파일로 분리해서 저장



서울_부산 교통사고 통계.xlsx



시군구별 교통사고 통계.csv



시군구별 부상신고자수.xlsx

1. 데이터 읽어오기

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

❖ *.CSV 데이터 읽어오기

- `변수명 = pd.read_csv('파일경로명' , encoding= '인코딩방식')`
 - ", "로 분리된 .csv 파일을 불러올 때
 - delimiter 옵션은 생략하면 ',' 로 인식
 - 인코딩방식 : 'EUC_KR' (한글이 포함된 일반적인 경우)/ 'cp949' (MS office에서 저장한 파일 형식)
- `변수명 = pd.read_excel('파일경로명')`
 - .xlsx 파일을 불러올 때

2. 데이터 살펴보기

❖ 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 열 정리하기

❖ 데이터 열 연산 및 새로운 열 생성하기

- 변수명['열이름'] = 변수명['열이름'] + 변수명['열이름']
 - 사칙연산 가능
 - 해당 열이름이 없으면 새로운 열 생성
 - 단, 숫자형 데이터에 한해 연산 가능

❖ 데이터에서 열이름 변경하기

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

❖ 열 데이터 삭제

- 변수명.drop(columns=['열이름'], axis=1, inplace= True)
 - 여러열 삭제 : 변수명.drop(columns=['열이름1' , ' 열이름2'], axis=1)
 - inplace= True : 원본을 변경함

4. 데이터 행 정리하기

❖ 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
 - 변수명.drop(변수명.index[0:17], axis=0): 0~16 index행 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

5. 데이터 결합하기

❖ 열방향으로 테이블 결합하기

- 변수명 = `pd.merge`(변수명1, 변수명2, `on`='결합기준열이름', `how`='결합방향')
- 2개의 테이블을 열방향으로 결합
- 변수명1, 변수명2: DataFrame 구조를 갖는 테이블
- 결합하고자 하는 테이블들은 반드시 **동일한 열이름이 존재해야 함**
- `On`='결합기준열이름' : 두개의 테이블을 결합할때 기준이 되는 열이름
`on`=['열이름1', '열이름'] : 결합 기준열이 여러 개일 경우
- `how`='outer' : 일치하지 않은 결합기준의 경우 행으로 추가

❖ 행방향으로 테이블 결합하기

- 변수명 = `pd.concat`([변수명1, 변수명2])
- 변수명2를 변수명1에 행방향으로 추가
- 변수명1과 변수명2에는 따로표로 묶지 않는다.

6. 조건에 맞는 데이터 추출하기

❖ 조건에 맞는 데이터 검색하기

- 변수명[열이름] 검색 조건
 - 검색조건: 검색하고자 하는 열이름을 비교 연산자와 논리연산자를 사용하여 기술
- 변수명.query("검색 조건")(['열이름1','열이름2'])
 - []내에 출력하고자 하는 열이름 기술
 - 단, 열이 2개 이상일 경우에는 대괄호([]) 2개 사용

7. 데이터 그룹 및 정렬하기

❖ 데이터 그룹화하여 간단한 통계 확인하기

- 변수명.**groupby**("그룹열이름") [['열이름1','열이름2']]. 통계함수()
 - 그룹열이름 : 그룹화할 열의 이름
 - 열이름: 그룹별로 통계 데이터를 확인하고자 하는 열의 이름

❖ 데이터 정렬하기

- 변수명.**sort_values**(['열이름'], ascending=**True**)
 - ascending = True:오름차순, False:내림차순, 생략:오름차순