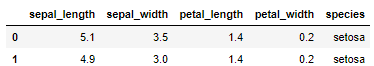
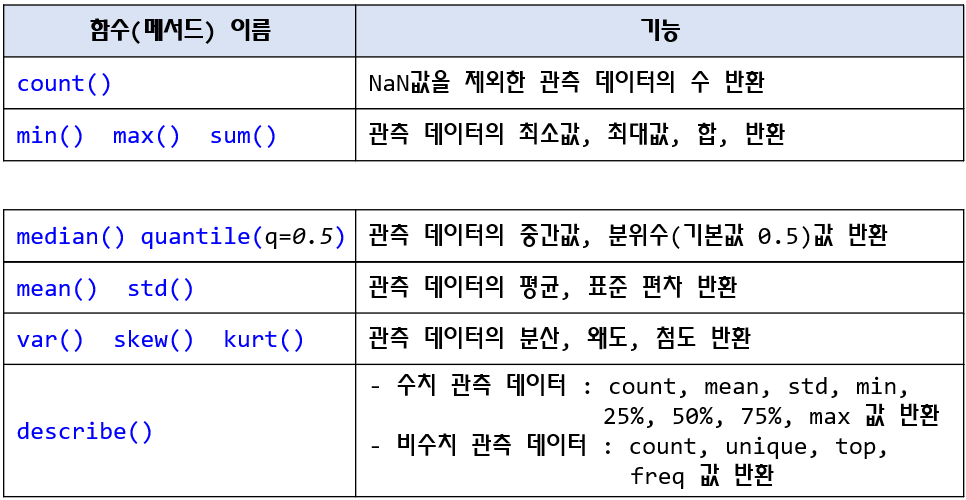
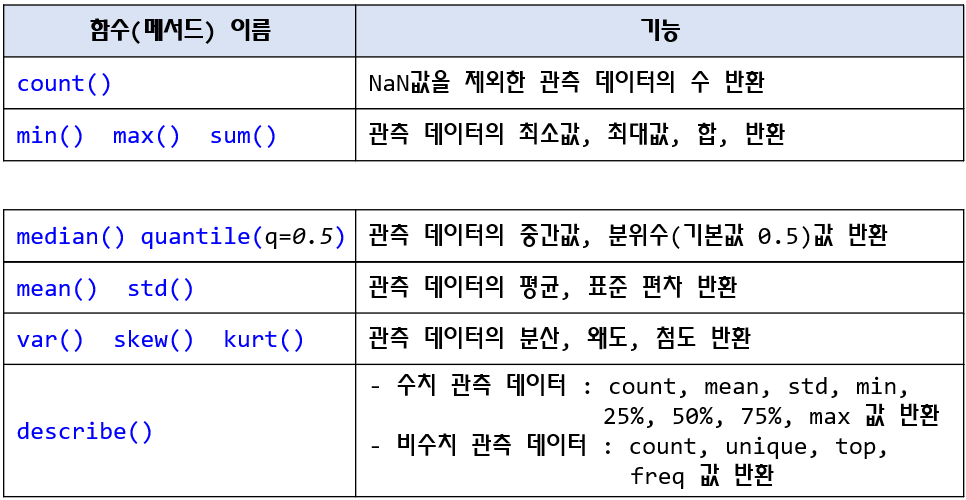
1. seborn 라이브러리에서 내장 데이터 불러오기(as sns)  
   sns.load\_dataset(“데이터 세트 이름”) ex: sns.load\_dataset(“iris”)  
    
2. 정규표현식을 활용한 열 검색 : 변수명.filter( regex = ‘정규표현식’ )

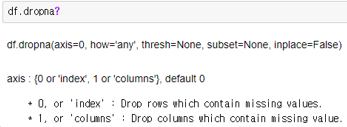
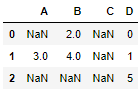
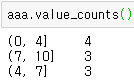
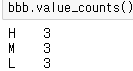
|  |  |
| --- | --- |
| df.filter(regex='.p') p앞에 문자 있는 것 | \d 숫자, [0-9]도 숫자 의미 |
| df.filter(regex='length$') length로 끝나는 것 | \D 문자, [^0-9]도 문자 의미 |
| df.filter(regex='^sepal') sepal로 시작하는 것 | \s 공백이나 탭 |
| df.filter(regex='l\d+') l(엘)뒤에 숫자 있는 것 | \S 공백 문자 이외의 문자 |
| df.filter(regex='width?') width이 있는 것 | \w 알파벳 |
| df.filter(regex='l\d{3}') l뒤에 숫자 3개 있는 것 | \W 알파벳 이 외의 문자(한글..) |
| df.filter(regex='sepal' '|' 'width') sepal 이나 width 있는 것 | \ 이스케이프문자 ( . \* ? + 등의 앞에) |

* 1. 언더바 가 들어간 열 regex = ‘\_’
  2. df.loc[ df['sepal\_length'] > 5.5, ['sepal\_length','sepal\_width'] ]

1. Summarize Data  
   
   1. df['species'].value\_counts( ) : 각각의 값을 그룹화 해서 개수를 보여줌

pd.DataFrame(df['species'].value\_counts( )) : 위의 결과를 데이터 프레임으로 따로 보기

* 1. len(df) 데이터 프레임의 행이 몇개인지 알려줌 = df.shape[0], len(df) == df.shape[0]는 True
  2. df['species'].nunique( ) : species가 가진 값 종류가 몇 개인지 알려 줌
  3. df.describe( ) df.describe(include = 'all')   
     df.describe(include = [np.object]) df.describe(include = [np.number])

1. 결측값 처리 : isnull( ), notnull( ) 기억 나는지?, dropna( ), fillna( )
   1. 변수명.dropna( )  
        
        
        
      df.dropna(axis = 1, how = 'all')   
      how = all : 전부 다 null값일때 drop적용/ how = any : 하나라도 null값이 있으면 drop적용
   2. df.fillna(0)  
      values = {'A': 0, 'B': 1, 'C': 2, 'D': 3}/ df.fillna(values)  
      df.fillna(df['D'].mean( ))  
      df.fillna(method = 'ffill')  
      df.fillna(method = 'bfill')
2. 새로운 열 만들기: assign( )  
   df의 A열에서 로그를 구해(np.log(df . A)) C라는 열을 만들어라
   1. df[ 'C' ] = np.log( df.A )
   2. df.assign(C = np.log(df.A))
3. 숫자 데이터를 범주형 데이터로 바꿀 때 활용하는 cut, qcut 그리고 깍두기 value\_counts( )
   1. pd.cut( 1차원데이터, 나눌 구간인 bins, right=True, labels = none)  
      1차원 데이터가 연속의미의 숫자일 때 이들을 특정 구간으로 끊어서 나누어 줌  
      나누고 싶은 구간은 bins 자리에서 지정하면 됨, 단일숫자 or 리스트  
      우측 경계숫자 빼려면 right = False, 나뉜 애들 범주값 주려면 labels 옵션(구간 개수 맞춰서)  
        
      pd.cut(df.A, 3) : df의 A열을 3개 구간으로  
      pd.cut(df.A, [0, 4, 7, 10], right = False, labels = ['L', 'M', 'H'])  
      : df.A를 0~4미만(right=True면 0~4이하), 구간3개의 값 각각 L, M, H로 범주값 지정
   2. pd.qcut(1차원데이터, 나눌 구간인 q, labels = none)  
      cut과 매우 유사, right옵션 없음, 사분위수(quantile) 기반으로 비슷하게 나누어 줌, 대량일때  
      백분율 기반 개념이라 bins에 해당하는 q가 1을 넘으면 errer남  
      pd.qcut(df.A, 3)  
      pd.qcut(df.A, [0, 0.3, 0.6, 1], labels = ['L', 'M', 'H'])
   3. 변수명.value\_counts(ascending=False)  
      pd.cut(df.A, [0, 4, 7, 10]) pd.cut(df.A, [0, 4, 7, 10], right = False, labels = ['L', 'M', 'H'])  
       
4. df.max(axis = 1) 와 df.max(axis = 0)  
   df.max(axis = 1) 각 행별 최대값  
   df.max(axis = 0) 각 열별 최대값

1. 임계치 만큼 값을 다듬어 주는 clip( ), 절대값 구하는 abs( )  
    df.A가 가진 값은 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
   1. df['A'].clip(lower=2,upper=8)의 결과: 2, 2, 3, 4, 5, 6, 7, 8, 8, 8
   2. df['B'].abs( ) : df.B에 음수값이 있다면 모두 절대값으로 바꾸어 줌