|  |  |
| --- | --- |
| 교육 제목 | **데이터 살펴보기** |
| 교육 일시 | 2021.9.30 |
| 교육 장소 | C6 |
| **교육 내용** | |
| 오전 | **데이터 프레임 구조**  -데이터 내용 미리보기  앞부분 : DataFrame 객체.head(n)  뒷부분DataFrame 객체.tail(n)  -데이터 요약 정보 확인  -데이터 프레임의 크기(행,열) : DataFrame 객체.shape  - 데이터 프레임의 기본 정보 : DataFrame 객체.info()  -데이터 프레임의 기술통계정보 요약: DataFrame객체.describe()    -데이터 개수 확인  -각 열의 데이터 개수 : DataFrame객체. count()  -열의 고윳값 개수: df [열이름"]. .value\_counts()  각열들의 고유한 값, 종류가 몇개인지 찾을 때        **통계 함수 이용**  **-평균값 : 모든 열의 평균값: DataFrame 객체.mean()**  특정 열의 평균값: DataFrame 객체[“열이름”].mean()  **-중간값: 모든 열의 중간값: DataFrame 객체. median ()**  특정 열의 중간값: DataFrame 객체[“열이름”]. median ()  **-최대값: 모든 열의 최대값: DataFrame 객체. max ()**  특정 열의 최대값: DataFrame 객체[“열이름”]. max ()  **-최소값: 모든 열의 최소값: DataFrame 객체. min ()**  특정 열의 최소값: DataFrame 객체[“열이름”]. min ()  **-표준편차: 모든 열의 표준편차: DataFrame 객체. std ()**  특정 열의 표준편차: DataFrame 객체[“열이름”]. std ()  **-상관계수: 모든 열의 상관계수: DataFrame 객체. corr ()**  특정 열의 상관계수: DataFrame 객체[“열이름”]. corr ()  **판다스 내장 그래프 도구 활용**  -선그래프: DataFrame객체.plot()  -막대그래프: DataFrame객체.plot(kind=’bar’)  -히스토그램: DataFrame객체.plot(kind=’hist’)  -산점도: DataFrame객체.plot(x=’\_’, y=’\_’ ,kind=’scatter’)  -박스플롯: DataFrame객체[[‘mpg(열), ‘cylinders(열)’]].plot(kind=’box’)          **시각화 도구**  **Matplotlib – 기본 그래프 도구**  -선그래프  -기본사용법:모듈을 “as plt”로 가져와 데이터 프레임으로 변환  Df=df.fillna(method=’ffill’) 누락값을 앞 데이터로 채움  ‘bfill’ 뒤의값으로 채움  -차트 제목, 축 이름 추가  제목: title() , x축:xlabel() y축:ylabel()  -matplotlib 한글폰트 오류해결  -그래프 꾸미기  Figure(): 그림틀의 가로사이즈를 더 크게 한다  Xticks(각도): x축 눈금라벨을 반시계 방향으로 회전  -화면분할하여 그래프 여러 개 그리기 -axe객체 사용  Fig 객체에 add\_subplot() 메소드 사용  -선 그래프 꾸미기 옵션  Marker=’o,+,\*,.’ / markerfaceclor=’마커 배경색’  /markersize=\_ /color=선의색 Lineweidth=선의두께  /label=’\_->\_’ 라벨지정      -면적 그래프  Plot() 메소드에 kind=’area’ 옵션을 추가  기본값은 stacked=True =겹치지않게 , False = 겹치게    -막대 그래프  Plot() 메소드에 kind=’bar’ 옵션을 추가  가로형 막대그래프 kind=’barh’    -히스토그램  Plot() 메소드에 kind=’hist’ 옵션을 추가  -산점도: 서로 다른 두 변수 사이의 관계를 나타낸다  Plot() 메소드에 kind=’ scatter’ 옵션을 추가  X=’weight’ 로x축에 위치할 변수 선택  Y=’mpg’로 y축 변수 선택  -파이차트  Plot() 메소드에 kind=’pie옵션을 추가  -박스 플롯  Boxplot() 메소드 사용 |
|  | **Seaborn 라이브러리 – 고급그래프**  Matplotlib 의 기능과 스타일을 확장한 시각화 도구 고급 버전  -데이터셋 가져오기 :seaborn라이브러리 에서 제공하는 ‘titanic’데이터셋 을 사용한다  Load\_dataset함수 사용하여 데이터프레임 을 가져 온다  -회귀선이 있는 선점도: regplot()함수는 서로 다른 2개의 변수 사이의 산점도를 그리고 선형회귀분석에 의한 회귀선을 함께 나타낸다  Fit\_reg=False 옵션 설정 = 회귀선 안보이게 설절    -히스토그램/ 커널밀도 그래프: 단변수 데이터 분포 확인할 때 distplot()  Hist=False:히스토그램 표시x  Kde=False: 밀도그래프 표시x  -히트맵: heatmap()메소드 제공sex=행 ,class=열  aggfunc-‘size’:데이터 값의 크기 기준  -범주형 데이터의 산점도: stripplot()함수,swarmplot()함수 사용  -막대그래프: barploy()함수  -빈도그래프: countplot()함수  -박스플롯/바이올린 그래프: boxplot() , violinplot()함수  -조인트 그래프: jointplot()함수, 회귀선추가(kind=’reg’), 육각 산점도(kind=’hex’),  커널 밀집그래프(kind=’kde’)  -조건을 적용하여 화면을 그리드로 분할: FacetGrid()  -이변수 데이터 분포 : pairplot()함수  **Folium라이브러리 – 지도활용**  -Folium설치  Cmd = pip install folium // jupyter창= !pip install folium  -지도 만들기: map()  -지도 스타일 적용하기:tiles 옵션 (stamenTerrain , stamen Toner)  -지도에 마커 표시하기: marker() 위도,경도 정보전달  -지도 영역에 단계구분도 표시하기: choropleth()함수  어떤 경계에 둘러싸인 영역에 색을 칠하거나  음영 등으로 정보를 나타내는 시각화 방법 |