|  |  |
| --- | --- |
| 교육 제목 | **데이터 시각화 연습** |
| 교육 일시 | 2021년 11월 17일 수요일 |
| 교육 장소 | 영우글로벌러닝 2층 |
| **교육 내용** | |
| 오전 | 1. Matplotlib 구조    1. Artist는 matplot에서 데이터 및 시각화를 위한 객체들을 지칭하는 이름matplotlib 은 그래프를 그리기 위해서 Runtime Configuration Parameters, 줄여서 rcParams라는 딕셔너리의 값을 이용해서 그림 rcParams 딕셔너리의 값을 바꿔주면, backend 에서 그릴 때, 다른 방식으로 그리게 됨. 그래프 폰트 사이즈, 크기, 바탕색 등 여러가지 속성들이 모두 matplotlib.rcParams 딕셔너리에 담겨있음 2. 그래프 그리기 예제 4-2    1. # 라이브러리 불러오기       1. import pandas as pd       2. import matplotlib.pyplot as plt    2. # Excel 데이터를 데이터프레임 변환       1. df = pd.read\_excel('data/시도별 전출입 인구수.xlsx', header=0)    3. # 전출지별에서 누락값(NaN)을 앞 데이터로 채움 (엑셀 양식 병합 부분)       1. df = df.fillna(method='ffill')    4. # 서울에서 다른 지역으로 이동한 데이터만 추출하여 정리       1. mask = (df['전출지별'] == '서울특별시') & (df['전입지별'] != '서울특별시')       2. df\_seoul = df[mask]       3. df\_seoul = df\_seoul.drop(['전출지별'], axis=1)       4. df\_seoul.rename({'전입지별':'전입지'}, axis=1, inplace=True)       5. df\_seoul.set\_index('전입지', inplace=True)    5. # 서울에서 경기도로 이동한 인구 데이터 값만 선택       1. sr\_one = df\_seoul.loc['경기도']    6. # x, y축 데이터를 plot 함수에 입력       1. plt.plot(sr\_one.index, sr\_one.values)    7. # 차트 제목 추가       1. plt.title('서울 -> 경기 인구 이동')    8. # 축이름 추가       1. plt.xlabel('기간')       2. plt.ylabel('이동 인구수')       3. plt.show() |
| 오후 | 1. # 예제 4-11 2. # 라이브러리 불러오기    1. import pandas as pd    2. import matplotlib.pyplot as plt 3. # matplotlib 한글 폰트 오류 문제 해결    1. from matplotlib import font\_manager, rc    2. font\_path = "./data/malgun.ttf" #폰트파일의 위치    3. font\_name = font\_manager.FontProperties(fname=font\_path).get\_name()    4. rc('font', family=font\_name) 4. # Excel 데이터를 데이터프레임 변환    1. df = pd.read\_excel('data/시도별 전출입 인구수.xlsx', header=0) 5. # 전출지별에서 누락값(NaN)을 앞 데이터로 채움 (엑셀 양식 병합 부분)    1. df = df.fillna(method='ffill') 6. # 서울에서 다른 지역으로 이동한 데이터만 추출하여 정리    1. mask = (df['전출지별'] == '서울특별시') & (df['전입지별'] != '서울특별시')    2. df\_seoul = df[mask]    3. df\_seoul = df\_seoul.drop(['전출지별'], axis=1)    4. df\_seoul.rename({'전입지별':'전입지'}, axis=1, inplace=True)    5. df\_seoul.set\_index('전입지', inplace=True) 7. # 서울에서 '충청남도','경상북도', '강원도', '전라남도'로 이동한 인구 데이터 값만 선택    1. col\_years = list(map(str, range(1970, 2018)))    2. df\_4 = df\_seoul.loc[['충청남도','경상북도', '강원도', '전라남도'], col\_years] 8. # 스타일 서식 지정    1. plt.style.use('ggplot') 9. # 그래프 객체 생성 (figure에 1개의 서브 플롯을 생성)    1. fig = plt.figure(figsize=(20, 10))    2. ax1 = fig.add\_subplot(2, 2, 1)    3. ax2 = fig.add\_subplot(2, 2, 2)    4. ax3 = fig.add\_subplot(2, 2, 3)    5. ax4 = fig.add\_subplot(2, 2, 4) 10. # axe 객체에 plot 함수로 그래프 출력     1. ax1.plot(col\_years, df\_4.loc['충청남도',:], marker='o', markerfacecolor='green', markersize=10, color='olive', linewidth=2, label='서울 -> 충남')     2. ax2.plot(col\_years, df\_4.loc['경상북도',:], marker='o', markerfacecolor='blue', markersize=10, color='skyblue', linewidth=2, label='서울 -> 경북')     3. ax3.plot(col\_years, df\_4.loc['강원도',:], marker='o', markerfacecolor='red', markersize=10, color='magenta', linewidth=2, label='서울 -> 강원')     4. ax4.plot(col\_years, df\_4.loc['전라남도',:], marker='o', markerfacecolor='orange', markersize=10, color='yellow', linewidth=2, label='서울 -> 전남') 11. # 범례 표시     1. ax1.legend(loc='best')     2. ax2.legend(loc='best')     3. ax3.legend(loc='best')     4. ax4.legend(loc='best') 12. # 차트 제목 추가     1. ax1.set\_title('서울 -> 충남 인구 이동', size=15)     2. ax2.set\_title('서울 -> 경북 인구 이동', size=15)     3. ax3.set\_title('서울 -> 강원 인구 이동', size=15)     4. ax4.set\_title('서울 -> 전남 인구 이동', size=15) 13. # 축 눈금 라벨 지정 및 90도 회전     1. ax1.set\_xticklabels(col\_years, rotation=90)     2. ax2.set\_xticklabels(col\_years, rotation=90)     3. ax3.set\_xticklabels(col\_years, rotation=90)     4. ax4.set\_xticklabels(col\_years, rotation=90) 14. plt.show() |