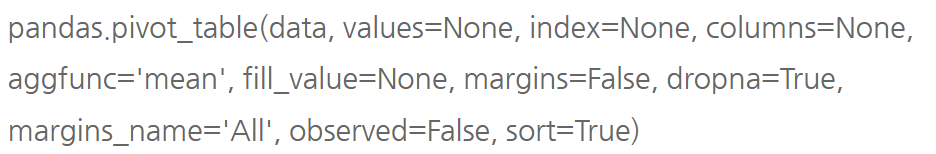
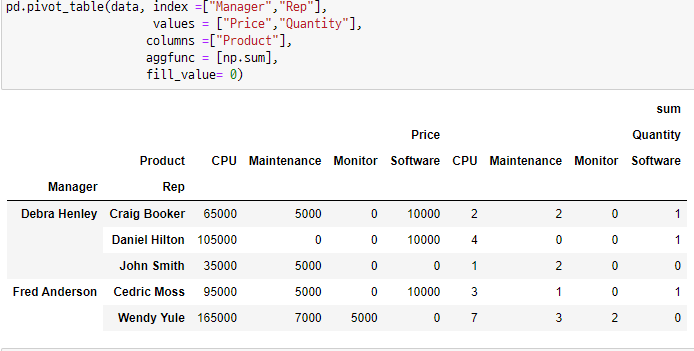
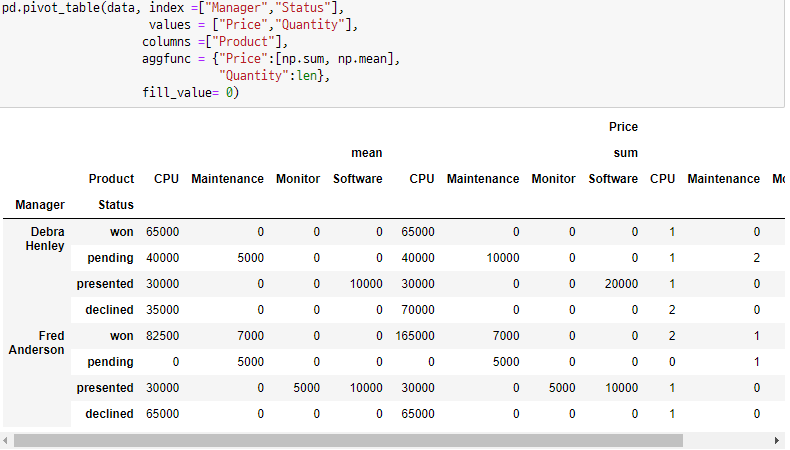
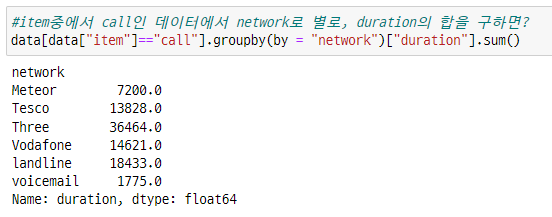
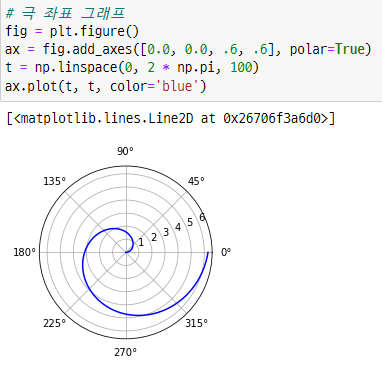
22-02-14

Pivot + GroupBy + Seaborn

* Pivot\_table : 데이터 프레임의 행과 열을 내가 원하는데로 바꿀 수 있음 
* pd.pivot\_table(data, index = [”pd컬럼1”,”pd컬럼2”]) :여러 개면 리스트로 받기
* pd.pivot\_table(data, index = [”pd컬럼1”,”pd컬럼2”], values = [”pd컬럼3”]) : 안에 채울, 데이터 값 지정
* pd.pivot\_table(data, index = [”pd컬럼1”,”pd컬럼2”], values = [”pd컬럼3”], aggfunc = [np.mean, len] ) : value에 대한 평균과 개수
* pd.pivot\_table(data, index = [”pd컬럼1”,”pd컬럼2”], values = [”pd컬럼3”], columns = [”pd컬럼4”], aggfunc = [np.sum], fill\_value = 0 ) : x축에 컬럼1,2 y축에 컬럼4 를 컬럼3을 내용으로 합을 채워넣음, 빈 곳은 NaN이 아니라 0으로 채움 
* columns별로 다른 연산을 보고자 한다면 dict 이용하기 
* Pd.crosstab(data[”컬럼1”], data[”컬럼2”]) 컬럼1을 가로로 컬럼2를 세로로 안에 무조건 빈도로 채움 (몇 건 있었는지)
* data[”컬럼”].value\_counts( ) : 컬럼의 종류와 각각 몇 개가 있는지
* data.describe( ) : 전반적인 통계값 확인( 최대, 최소, 4분위수)
* data.info( ) : 데이터 null 수, 타입 확인
* 데이터를 컬럼안에 있는 그룹끼리 묶어서 보기 : data.groupby(by = “컬럼”)
* 어떻게 묶였는지 확인 하기 : data.groupby(by = “컬럼”).groups.keys( )
* 데이터 중 원하는 조건으로만 보고 싶다면 불리언 인덱싱 + 뒤에 이어 붙이기 
* 묶는 조건이 여러 개 일 때 리스트로 묶기: data.groupby(by = [“컬럼1”, “컬럼2”] )
* 묶은 기준에 대해서 볼 것이 여러 개 일 때 : data.groupby(by = [“컬럼1”, “컬럼2”] ).agg({“보고 싶은 컬럼”: “count”, “보고 싶은 컬럼” : “first”})
* 시각화하기 위해 필요한 패키지 : import matplotlib / import matplotlib.pyplot as plt
* 그릴 준비 : fig = plt.figure( ) 그림 포지션 지정 : axes = fig.add-axes( [0, 0, 1, 1] ) 🡪 그래프 여러개 지정 가능 데이터 표시 : axes.plot(x, y, “r”) r은 red색 의미 축에 이름 적기 : axes.set\_xlabel(“넣고 싶은 이름”) 🡪 xlabel 대신 ylabel로 적으면 y축 이름 / title적으면 그래프 제목
* 그림 위치 지정 하지 않고 분할해서 그리기 : ①fig, axes = plt.subplots( ) : 이건 1개 ②fig, axes = plt.subplots(nrows = 1, ncols = 3 ) : 1 by 3
* 분할해서 그릴 때 겹치지 않게 자동으로 간격 조정 : fig.tight\_layout( )
* 분할된 그림 하나씩 인덱스로도 접근 가능 : axes[0.0].plot(x, y,)
* 그래프 옵션 : ①선 두께 : linewidth(lw로 줄여서 가능) = 2 ②선 연결 모양 : linestyle(ls로 줄이기 가능) = “-“ 🡪 참고) ‘-‘, ‘--’, ‘-.’, ‘:’, ‘steps’ ③점 크기 : markersize = 2 ④점 모양 : marker = “+” 🡪 참고) ‘+', 'o', '\*', 's', ',', '.', '1', '2', '3' 등
* 범례 추가 : axes.legend(loc = 0) 숫자는 위치
* 그래프를 그림으로 저장하기 : outpath = “원하는 이름.png” 🡪 참고)파일형식 : PNG, JPG, EPS, SVG, PGF, PDF fig.savefig(outpath)
* 그리드 그리기 : axes.plot(x, x\*\*2) axes.grid(True) or axes.grid(color=”b”, ls = ‘dashed’, lw=2) axes.spines[‘bottom’].set\_color(‘blue’) 🡪 bottom, blue 바꿀 수 있음 axes.splines[‘bottom’].set\_linewidth(5)
* 여러 그래프들 ①scatter : axes.scatter( x, y ) ②histogram :axes.hist( x , cumulative = True(축적) ) axes.set\_xlim( (min(x), max(x) ) ③bar : axes.bar(x, x\*\*2, align="center", width=0.3 ) ④step : axes.step(x, x\*\*2, lw=2) ⑤fill\_between : axes.fill\_between(x, x\*\*2, x\*\*2+10, color="blue", alpha=0.3) ⑥극 좌표 :