

```
col_list = list(df.columns)
col_list[0] = 'local'
col_list[3] = 'rateMarriageableAge'
df.columns = col_list
```

데이터프레임 df의 컬럼명이 한글로 되어 있기 때문에 영어로 바꾸는 작업을 실시한다. 변수 col_list에 df의 컬럼 리스트를 대입한다. 0번째와 3번째의 컬럼명을 변경하였고 이를 df.columns에 대입한다.

```
df2 = pd.read_csv('/content/fulldata_01_01_04_P_산후조리업.csv', encoding='cp949')
df2['시'] = df2['소재지전체주소'].str.split(' ', n=1).str[0]
df2['시'] = df2['시'].replace(['경상남도', '경상북도'], '경상도')
```

변수 df2에 엑셀 파일을 불러와 읽어 저장한다. 이때 encoding='cp949'란 한글 전용 인코딩 방식이다. 데이터프레임의 '소재지전체주소' 열을 기반으로 '시' 정보를 추출하려고 한다. 그러기 위해서는 먼저 str.split()을 사용하여 공백을 기준으로 나누고, 그 결과에서 첫 번째 요소를 선택하여 '시' 정보를 얻는다. 이 정보는 새로운 '시' 열에 저장된다. 정보 간편화를 위해 경상북도, 경상남도는 replace()로 각각 경상도로 통합하였다.

```
total_count = df2.groupby('시')['영업상태명'].count().reset_index(name='전체 수')
closed_count = df2[df2['영업상태명'] == '폐업'].groupby('시')['영업상태명'].count().reset_index(name='폐업 수')
merged_df2 = pd.merge(total_count, closed_count, on='시')
merged_df2['폐업률(%)'] = (merged_df2['폐업 수'] / merged_df2['전체 수']) * 100
```

'시'별로 그룹화 한 후, 각 그룹에서 '영업상태명' 열의 개수를 세어서 '전체 수'라는 새로운 열을 만들어 변수 total_count에 대입한다. '영업상태명'이 '폐업'인 행들만 선택한 뒤, '시'별로 그룹화 한 후, 각 그룹에서 '영업상태명' 열의 개수를 세어서 만든 새로운 열을 변수 closed_count에 저장한다. '시' 열을 기준으로 '전체 수(total_count)'와 '폐업 수(closed_count)' 데이터프레임을 병합한다. '폐업 수'와 '전체 수'를 사용하여 '폐업률(%)'을 계산한다.

```
a = merged_df2[['rateClosed']]
dfT = pd.concat([df, a], axis = 1)
dfT = dfT.drop([dfT.columns[4], dfT.columns[5]], axis=1)
```

데이터프레임 merged_df2에서 'rateClosed' 열만 추출하여 변수 a에 저장한다. 기존의 데이터프레임 df와 새로 만든 데이터프레임 a를 열을 기준으로 병합하여 변수 dfT에 대입한다. 'axis=1'은 열을 기준으로 하겠다는 의미이다.

```
import matplotlib.pyplot as plt
plt.figure(figsize=(15,7))
plt.bar(dfT.local, dfT.rateMarriageableAge, color = 'green', label = '결혼적령기 인구율') plt.bar(dfT.local,
dfT.rateClosed, bottom = dfT.rateMarriageableAge, label = '산후조리원 폐업률', color = 'red')
```

```
plt.xticks(rotation = 45)
plt.title("지역별 산후조리원 폐업률과 결혼적령기 인구율", fontsize = 20)
plt.xlabel('행정구역')
plt.ylabel('비율(%)')
plt.ylim([0,130])
plt.legend(fontsize = 14)
plt.show()
```

시각화를 하기 위해 matplotlib.pyplot를 plt로 별칭하여 불러왔다. ‘bottom=dfT.rateMarriageableAge’은 dfT.rageMarriageableAge값을 바닥삼아 그 위로 막대그래프를 생성하겠다는 의미이다.

```
df['인허가일자'] = pd.to_datetime(df['인허가일자'])
df['연도'] = df['인허가일자'].dt.year
result_1 = df[df['영업상태구분코드'] == 1].groupby('연도').size()
```

pd.to_datetime은 판다스 라이브러리에서 제공하는 함수로 날짜와 시간 정보를 가지는 데이터를 datetime 형식으로 변환하는 데 사용한다. 데이터프레임(df) ‘인허가일자’ 열의 값을 datetime 형식으로 변환한 후, 새로운 열 ‘인허가일자’에 할당한다. .dt.year는 시리즈의 datetime 객체에서 연도 정보만 추출하는 매서드다. ‘인허가일자’ 열에서 연도(year) 정보를 추출해 ‘연도’ 열에 할당한다. ‘영업상태구분코드’가 1인 경우에만 해당하는 데이터를 선택한다. 연도를 기준으로 데이터를 그룹화하고 각 그룹에서 1의 개수를 계산한다. ‘grouby(‘연도’).size()’는 각 연도별 그룹의 크기, 해당연도에 해당하는 행의 개수를 계산한다.

```
result_1.plot(kind='bar', stacked=True, color='#e35f62', alpha =0.3)
plt.title("연도별 영업상태구분 - 영업")
plt.xlabel("연도")
plt.ylabel("빈도")
plt.show()
```

plot() 매서드를 이용해 시각화를 생성한다. 그래프의 종류는 ‘bar(막대그래프)’로 설정하고 ‘stacked=True’를 통해 연도별로 각각의 ‘영업상태구분코드’ 빈도를 쌓아서 표시한다. ‘alpha=0.3’로 막대의 투명도를 조절한다.

```
del_col = df.loc[:, '상세영업상태코드:']
df.drop(del_col, axis=1, inplace=True)

df.dropna(subset=['도로명전체주소'])
```

‘상세영업상태코드’ 열부터 끝까지 삭제한다. 그리고 ‘도로명전체주소’에서 결측치는 편의상 제거해준다.

```
local_df = pd.DataFrame([[18,10,0,28],[148,136,3,287],[41,42,0,83], [8,6,0,14],[24,24,0,48],[8,23,0,31],
[23,25,2,51],[117,102,4,223],[6,2,0,8], [7,2,0,9],[23,20,1,44],[25,17,1,43], [7,12,0,19],[23,25,0,49]],
index=['강원도','경기도','경상도', '광주광역시','대구','대전', '부산','서울','세종', '울산','인천','전라도', '제주','충청도'],
columns=['working','closed','closing','total'])
```

지역별 ‘영업중’, ‘휴업중’, ‘폐업(직권폐업)’, ‘전체’를 비교하기 위해 새로운 데이터프레임을 생성한다.