주차 수요 예측

단계1 : 데이터 전처리

0.미션

단지별 등록 차량 수를 예측하고자 합니다. 이에 맞게 데이터프레임을 생성하시오.

- 1) 단지별 데이터와 상세 데이터를 분리합니다.
- 2) 단지별 데이터
  - 범주의 수 줄이기 : 너무 종류가 많은 범주는 의미상 적절하게 병합합니다.
- 3) 상세 데이터를 단지별로 집계해야 합니다.
  - 전용면적 : 구간으로 나누어 집계하기
  - 임대보증금 혹은 임대료 : 전용면적별 세대수를 감안하여 집계하기.
- 4) 단지별 데이터와 집계데이터를 하나로 합칩니다.
- 5) (옵션)추가 변수 만들기
  - 등록 차량수를 예측하기 위해 필요한 변수를 추가해 봅시다.

1.환경설정

- 세부 요구사항
  - 경로 설정 : 다음의 두가지 방법 중 하나를 선택하여 폴더를 준비하고 데이터를 로딩하시오.
    - 1) 로컬 수행(Ananconda)
      - 제공된 압축파일을 다운받아 압축을 풀고
      - anaconda의 root directory(보통 C:/Users/< ID > 에 project 폴더를 만들고, 복사해 넣습니다.
    - 2) 구글콜랩
      - 구글 드라이브 바로 밑에 project 폴더를 만들고,
      - 데이터 파일을 복사해 넣습니다.
  - 라이브러리 설치 및 로딩
    - requirements.txt 파일로 부터 라이브러리 설치
  - 기본적으로 필요한 라이브러리를 import 하도록 코드가 작성되어 있습니다.
    - 필요하다고 판단되는 라이브러리를 추가하세요.

(1) 한글폰트 설치

In [1]:
```
# # 한글폰트
# !sudo apt-get install -y fonts-nanum
```

```
# !sudo fc-cache -fv
# !rm ~/.cache/matplotlib -rf
```

- (구글콜랩) 한글폰트 설치후 런타임 재시작!

(2) 경로 설정

1) 로컬 수행(Anaconda)

- project 폴더에 필요한 파일들을 넣고, 본 파일을 열었다면, 별도 경로 지정이 필요하지 않습니다.

In [2]:
```
path = 'C:/Users/User/program/mini_pjt/mini_3/실습파일_에이블러용/데이터/'
```

2) 구글 콜랩 수행

- 구글 드라이브 연결

In [3]:
```
# from google.colab import drive
# drive.mount('/content/drive')
```

In [4]:
```
# path = '/content/drive/MyDrive/project/'
```

(3) 라이브러리 설치 및 불러오기

1) 설치

- requirements.txt 파일을 아래 위치에 두고 다음 코드를 실행하시오.
  - 로컬 : 다음 코드셀 실행
  - 구글콜랩 : requirements.txt 파일을 왼쪽 [파일]탭에 복사해 넣고 다음 코드셀 실행

In [5]:
```
#!pip install -r requirements.txt
```

2) 라이브러리 로딩

In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import joblib

# 필요한 라이브러리 로딩
from sklearn.model_selection import train_test_split
#from sklearn.ensemble import RandomForestClassifier
#from sklearn.metrics import *
```

In [7]:
```
import matplotlib.font_manager as fm
font_list = [font.name for font in fm.fontManager.ttflist]
font_list.sort()
font_list
```

Out[7]:
```
['Agency FB',
 'Agency FB',
 'Algerian',
 'Ami R',
 'Arial',
 'Arial',
 'Arial',
 'Arial',
 'Arial',
 'Arial',
 'Arial',
 'Arial',
 'Arial',
 'Arial',
 'Arial Rounded MT Bold',
 'Bahnschrift',
 'Baskerville Old Face',
 'Batang',
 'Bauhaus 93',
 'Bell MT',
 'Bell MT',
 'Bell MT',
 'Berlin Sans FB',
 'Berlin Sans FB',
 'Berlin Sans FB Demi',
 'Bernard MT Condensed',
 'Blackadder ITC',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Bodoni MT',
 'Book Antiqua',
 'Book Antiqua',
 'Book Antiqua',
 'Book Antiqua',
 'Bookman Old Style',
 'Bookman Old Style',
 'Bookman Old Style',
 'Bookman Old Style',
 'Bookshelf Symbol 7',
 'Bradley Hand ITC',
 'Britannic Bold',
 'Broadway',
 'Brush Script MT',
 'Calibri',
 'Calibri',
 'Calibri',
 'Calibri',
 'Calibri',
 'Calibri',
 'Californian FB',
 'Californian FB',
 'Californian FB',
 'Calisto MT',
```

```
                    'Calisto MT',
                    'Calisto MT',
                    'Calisto MT',
                    'Cambria',
                    'Cambria',
                    'Cambria',
                    'Cambria',
                    'Candara',
                    'Candara',
                    'Candara',
                    'Candara',
                    'Candara',
                    'Candara',
                    'Castellar',
                    'Centaur',
                    'Century',
                    'Century Gothic',
                    'Century Gothic',
                    'Century Gothic',
                    'Century Gothic',
                    'Century Schoolbook',
                    'Century Schoolbook',
                    'Century Schoolbook',
                    'Century Schoolbook',
                    'Chiller',
                    'Colonna MT',
                    'Comic Sans MS',
                    'Comic Sans MS',
                    'Comic Sans MS',
                    'Comic Sans MS',
                    'Consolas',
                    'Consolas',
                    'Consolas',
                    'Consolas',
                    'Constantia',
                    'Constantia',
                    'Constantia',
                    'Constantia',
                    'Cooper Black',
                    'Copperplate Gothic Bold',
                    'Copperplate Gothic Light',
                    'Corbel',
                    'Corbel',
                    'Corbel',
                    'Corbel',
                    'Corbel',
                    'Corbel',
                    'Courier New',
                    'Courier New',
                    'Courier New',
                    'Courier New',
                    'Curlz MT',
                    'D2Coding',
                    'DejaVu Sans',
                    'DejaVu Sans',
                    'DejaVu Sans',
                    'DejaVu Sans',
                    'DejaVu Sans Display',
                    'DejaVu Sans Mono',
                    'DejaVu Sans Mono',
```

```
'DejaVu Sans Mono',
'DejaVu Sans Mono',
'DejaVu Serif',
'DejaVu Serif',
'DejaVu Serif',
'DejaVu Serif',
'DejaVu Serif Display',
'Dubai',
'Dubai',
'Dubai',
'Dubai',
'Ebrima',
'Ebrima',
'Edwardian Script ITC',
'Elephant',
'Elephant',
'Engravers MT',
'Eras Bold ITC',
'Eras Demi ITC',
'Eras Light ITC',
'Eras Medium ITC',
'Felix Titling',
'Footlight MT Light',
'Forte',
'Franklin Gothic Book',
'Franklin Gothic Book',
'Franklin Gothic Demi',
'Franklin Gothic Demi',
'Franklin Gothic Demi Cond',
'Franklin Gothic Heavy',
'Franklin Gothic Heavy',
'Franklin Gothic Medium',
'Franklin Gothic Medium',
'Franklin Gothic Medium Cond',
'Freestyle Script',
'French Script MT',
'Gabriola',
'Gadugi',
'Gadugi',
'Garamond',
'Garamond',
'Garamond',
'Georgia',
'Georgia',
'Georgia',
'Georgia',
'Gigi',
'Gill Sans MT',
'Gill Sans MT',
'Gill Sans MT',
'Gill Sans MT',
'Gill Sans MT Condensed',
'Gill Sans MT Ext Condensed Bold',
'Gill Sans Ultra Bold',
'Gill Sans Ultra Bold Condensed',
'Gloucester MT Extra Condensed',
'Goudy Old Style',
'Goudy Old Style',
'Goudy Old Style',
'Goudy Stout',
```

```
                    'Gulim',
                    'HYGothic-Extra',
                    'HYGothic-Medium',
                    'HYGraphic-Medium',
                    'HYGungSo-Bold',
                    'HYHeadLine-Medium',
                    'HYMyeongJo-Extra',
                    'HYPMokGak-Bold',
                    'HYPost-Light',
                    'HYPost-Medium',
                    'HYShortSamul-Medium',
                    'HYSinMyeongJo-Medium',
                    'Haettenschweiler',
                    'Harlow Solid Italic',
                    'Harrington',
                    'Headline R',
                    'High Tower Text',
                    'High Tower Text',
                    'HoloLens MDL2 Assets',
                    'Impact',
                    'Imprint MT Shadow',
                    'Informal Roman',
                    'Ink Free',
                    'Javanese Text',
                    'Jokerman',
                    'Juice ITC',
                    'Kristen ITC',
                    'Kunstler Script',
                    'Leelawadee',
                    'Leelawadee',
                    'Leelawadee UI',
                    'Leelawadee UI',
                    'Leelawadee UI',
                    'Lucida Bright',
                    'Lucida Bright',
                    'Lucida Bright',
                    'Lucida Bright',
                    'Lucida Calligraphy',
                    'Lucida Console',
                    'Lucida Fax',
                    'Lucida Fax',
                    'Lucida Fax',
                    'Lucida Fax',
                    'Lucida Handwriting',
                    'Lucida Sans',
                    'Lucida Sans',
                    'Lucida Sans',
                    'Lucida Sans',
                    'Lucida Sans Typewriter',
                    'Lucida Sans Typewriter',
                    'Lucida Sans Typewriter',
                    'Lucida Sans Typewriter',
                    'Lucida Sans Unicode',
                    'MS Gothic',
                    'MS Outlook',
                    'MS Reference Sans Serif',
                    'MS Reference Specialty',
                    'MT Extra',
                    'MV Boli',
                    'Magic R',
```

```
                    'Magneto',
                    'Maiandra GD',
                    'Malgun Gothic',
                    'Malgun Gothic',
                    'Malgun Gothic',
                    'Matura MT Script Capitals',
                    'Microsoft Himalaya',
                    'Microsoft JhengHei',
                    'Microsoft JhengHei',
                    'Microsoft JhengHei',
                    'Microsoft New Tai Lue',
                    'Microsoft New Tai Lue',
                    'Microsoft PhagsPa',
                    'Microsoft PhagsPa',
                    'Microsoft Sans Serif',
                    'Microsoft Tai Le',
                    'Microsoft Tai Le',
                    'Microsoft Uighur',
                    'Microsoft Uighur',
                    'Microsoft YaHei',
                    'Microsoft YaHei',
                    'Microsoft YaHei',
                    'Microsoft Yi Baiti',
                    'MingLiU-ExtB',
                    'Mistral',
                    'Modern No. 20',
                    'MoeumT R',
                    'Mongolian Baiti',
                    'Monotype Corsiva',
                    'Myanmar Text',
                    'Myanmar Text',
                    'New Gulim',
                    'Niagara Engraved',
                    'Niagara Solid',
                    'Nirmala UI',
                    'Nirmala UI',
                    'Nirmala UI',
                    'OCR A Extended',
                    'Old English Text MT',
                    'Onyx',
                    'Palace Script MT',
                    'Palatino Linotype',
                    'Palatino Linotype',
                    'Palatino Linotype',
                    'Palatino Linotype',
                    'Papyrus',
                    'Parchment',
                    'Perpetua',
                    'Perpetua',
                    'Perpetua',
                    'Perpetua',
                    'Perpetua Titling MT',
                    'Perpetua Titling MT',
                    'Playbill',
                    'Poor Richard',
                    'Pristina',
                    'Pyunji R',
                    'Rage Italic',
                    'Ravie',
                    'Rockwell',
```

```
'Rockwell',
'Rockwell',
'Rockwell',
'Rockwell Condensed',
'Rockwell Condensed',
'Rockwell Extra Bold',
'STIXGeneral',
'STIXGeneral',
'STIXGeneral',
'STIXGeneral',
'STIXNonUnicode',
'STIXNonUnicode',
'STIXNonUnicode',
'STIXNonUnicode',
'STIXSizeFiveSym',
'STIXSizeFourSym',
'STIXSizeFourSym',
'STIXSizeOneSym',
'STIXSizeOneSym',
'STIXSizeThreeSym',
'STIXSizeThreeSym',
'STIXSizeTwoSym',
'STIXSizeTwoSym',
'Sans Serif Collection',
'Script MT Bold',
'Segoe Fluent Icons',
'Segoe MDL2 Assets',
'Segoe Print',
'Segoe Print',
'Segoe Script',
'Segoe Script',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI',
'Segoe UI Emoji',
'Segoe UI Historic',
'Segoe UI Symbol',
'Segoe UI Variable',
'Showcard Gothic',
'SimSun',
'SimSun-ExtB',
'Sitka',
'Sitka',
'Snap ITC',
'Stencil',
'Sylfaen',
'Symbol',
'Tahoma',
'Tahoma',
'Tempus Sans ITC',
'Times New Roman',
```

```
 'Times New Roman',
 'Times New Roman',
 'Times New Roman',
 'Trebuchet MS',
 'Trebuchet MS',
 'Trebuchet MS',
 'Trebuchet MS',
 'Tw Cen MT',
 'Tw Cen MT',
 'Tw Cen MT',
 'Tw Cen MT',
 'Tw Cen MT Condensed',
 'Tw Cen MT Condensed',
 'Tw Cen MT Condensed Extra Bold',
 'Verdana',
 'Verdana',
 'Verdana',
 'Verdana',
 'Viner Hand ITC',
 'Vivaldi',
 'Vladimir Script',
 'Webdings',
 'Wide Latin',
 'Wingdings',
 'Wingdings 2',
 'Wingdings 3',
 'Yet R',
 'Yu Gothic',
 'Yu Gothic',
 'Yu Gothic',
 'Yu Gothic',
 'cmb10',
 'cmex10',
 'cmmi10',
 'cmr10',
 'cmss10',
 'cmsy10',
 'cmtt10']
```

In [8]:
```
# 한글폰트설정
plt.rc('font', family='Malgun Gothic')
plt.rcParams['axes.unicode_minus'] = False
```

(4) 데이터 불러오기

- 주어진 데이터셋
  - train.xlsx : 학습 및 검증용
  - test.xlsx : 테스트용

1) 데이터로딩

In [9]:
```
train = pd.read_excel(path + 'train.xlsx')
test = pd.read_excel(path + 'test.xlsx')
```

In [10]:
```
train.head()
```

Out[10]:

	단지코드	단지명	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	단지내주차면수	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0001	엘에이치서초4단지	78	35	서울	20131204.0	계단식	개별가스난방	전체동설치	120	51.89	19.2603	50758000	620370	109
1	C0001	엘에이치서초4단지	78	43	서울	20131204.0	계단식	개별가스난방	전체동설치	120	59.93	22.2446	63166000	665490	109
2	C0002	LH삼성아파트	35	26	서울	20130801.0	복도식	개별가스난방	전체동설치	47	27.75	16.5375	63062000	458640	35
3	C0002	LH삼성아파트	35	9	서울	20130801.0	복도식	개별가스난방	전체동설치	47	29.08	17.3302	63062000	481560	35
4	C0003	강남LH8단지	88	7	서울	20131023.0	계단식	개별가스난방	전체동설치	106	59.47	21.9462	72190000	586540	88

In [11]:
```
test.head()
```

Out[11]:

	단지코드	단지명	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	단지내주차면수	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0005	서울석촌도시형주택(공임10년)	20	6	서울	20121115.0	복도식	개별가스난방	전체동설치	9	17.53	11.7251	50449000	263710	21
1	C0005	서울석촌도시형주택(공임10년)	20	10	서울	20121115.0	복도식	개별가스난방	전체동설치	9	24.71	16.5275	52743000	321040	21
2	C0005	서울석촌도시형주택(공임10년)	20	4	서울	20121115.0	복도식	개별가스난방	전체동설치	9	26.72	17.8720	53890000	332510	21
3	C0017	대구혁신센텀힐즈	822	228	대구경북	20180221.0	계단식	지역난방	NaN	824	51.87	20.9266	29298000	411200	797
4	C0017	대구	822	56	대구	20180221.0	계단	지역	NaN	824	59.85	24.1461	38550000	462600	797

단지코드	단지명	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	단지내주차면수	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
	혁신센텀힐ㅈ			경북		식	난방							

2) 기본 정보 조회

In [12]:
```
train.describe()
```

Out[12]:

	총세대수	전용면적별세대수	준공일자	단지내주차면수	전용면적	공급면적(공용)	임대보증금
count	1157.000000	1157.000000	1.103000e+03	1157.000000	1157.000000	1157.000000	1.157000e+03
mean	659.075194	163.691443	1.973918e+07	682.261884	51.565584	20.562360	2.850789e+07
std	456.110643	166.766358	2.392214e+06	473.331805	18.243315	5.164405	2.890687e+07
min	1.000000	1.000000	2.002000e+03	10.000000	17.590000	5.850000	0.000000e+00
25%	315.000000	44.000000	2.004052e+07	308.000000	39.480000	16.997400	1.379700e+07
50%	595.000000	112.000000	2.009102e+07	629.000000	46.900000	20.384700	1.997300e+07
75%	918.000000	229.000000	2.013121e+07	911.000000	59.810000	23.722500	3.375300e+07
max	2289.000000	1258.000000	2.022071e+07	4553.000000	139.350000	42.760000	2.549220e+08

In [13]:
```
train.shape
```

Out[13]:
```
(1157, 15)
```

In [14]:
```
test.describe()
```

Out[14]:

	총세대수	전용면적별 세대수	준공일자	단지내주차 면수	전용면적	공급면적 (공용)	임대보증금	
count	104.000000	104.000000	9.600000e+01	104.000000	104.000000	104.000000	1.040000e+02	
mean	732.567308	172.480769	2.005319e+07	712.259615	50.786890	20.596660	2.788852e+07	16
std	525.731940	220.291554	2.030039e+05	357.793182	19.134072	4.746655	3.909525e+07	13
min	20.000000	4.000000	1.900010e+07	9.000000	17.530000	11.603700	0.000000e+00	
25%	262.000000	59.000000	2.005097e+07	449.500000	36.640000	16.742925	9.484000e+06	8
50%	768.000000	129.500000	2.010570e+07	818.000000	46.895000	20.793000	1.790650e+07	16
75%	1017.000000	204.750000	2.012587e+07	882.000000	59.770000	24.721700	3.511750e+07	22
max	2389.000000	1885.000000	2.022071e+07	1604.000000	84.990000	28.619400	2.150570e+08	78

In [15]:
```
test.shape
```

Out[15]:
```
(104, 15)
```

2.데이터 전처리①

- **세부 요구사항**
  - NaN 조치
    - NaN은 조회합니다.
    - 준공일자만 여기서 조치합시다.
      - 준공연도만 필요합니다. 문자열로 변환한 후 앞 4자리를 잘라 준공연도로 변환합니다.
      - 잘못 들어간 데이터들과 nan 문자를 준공연도 최소값으로 치환합니다.
    - 나머지 변수들은 탐색 이후에 조치를 취합니다.
  - 불필요한 데이터 제거

(1) NaN 조치

In [16]:
```
train.isna().sum()
```

Out[16]:
```
단지코드          0
단지명           0
총세대수          0
전용면적별세대수      0
지역            0
준공일자         54
건물형태         22
난방방식         75
승강기설치여부      98
단지내주차면수       0
전용면적          0
공급면적(공용)      0
임대보증금         0
임대료           0
실차량수          0
dtype: int64
```

In [17]:
```
train['준공일자'] = train['준공일자'].astype(str).str[:4]
train.head()
```

Out[17]:

	단지코드	단지명	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	단지내주차면수	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0001	엘에이치서초4단지	78	35	서울	2013	계단식	개별가스난방	전체동설치	120	51.89	19.2603	50758000	620370	109
1	C0001	엘에이치서초4단지	78	43	서울	2013	계단식	개별가스난방	전체동설치	120	59.93	22.2446	63166000	665490	109
2	C0002	LH삼성아파트	35	26	서울	2013	복도식	개별가스난방	전체동설치	47	27.75	16.5375	63062000	458640	35
3	C0002	LH삼성아파트	35	9	서울	2013	복도식	개별가스난방	전체동설치	47	29.08	17.3302	63062000	481560	35
4	C0003	강남LH8단지	88	7	서울	2013	계단식	개별가스난방	전체동설치	106	59.47	21.9462	72190000	586540	88

In [18]:
```
train.isna().sum()
```

Out[18]:
```
단지코드         0
단지명          0
총세대수         0
전용면적별세대수     0
지역           0
준공일자         0
건물형태        22
난방방식        75
승강기설치여부     98
단지내주차면수      0
전용면적         0
공급면적(공용)     0
임대보증금        0
임대료          0
실차량수         0
dtype: int64
```

In [19]:
```
train['준공일자'].unique()
```

Out[19]:
```
array(['2013', '2014', '2011', '2007', '2012', '2018', '2010', '2008',
       '2009', '2017', '2016', '2019', '2015', '1970', '2006', '2004',
       '2005', '2001', 'nan', '1992', '1995', '2000', '1999', '1996',
       '1997', '1994', '1998', '2003', '2002', '1993', '2022', '1900',
       '2020', '1111', '2021'], dtype=object)
```

In [20]:
```
# 'nan' 값을 NaN으로 변환
train['준공일자'] = train['준공일자'].replace('nan', np.nan)
# 최소값 계산
min_value = train['준공일자'].dropna().astype(int).min()
```

In [21]:
```
train['준공일자'] = train['준공일자'].fillna(min_value)
```

In [22]:
```
train['준공일자'].unique()
```

Out[22]:
```
array(['2013', '2014', '2011', '2007', '2012', '2018', '2010', '2008',
       '2009', '2017', '2016', '2019', '2015', '1970', '2006', '2004',
       '2005', '2001', 1111, '1992', '1995', '2000', '1999', '1996',
       '1997', '1994', '1998', '2003', '2002', '1993', '2022', '1900',
       '2020', '1111', '2021'], dtype=object)
```

In [23]:
```
train.isna().sum()
```

Out[23]:
```
단지코드         0
단지명          0
총세대수         0
전용면적별세대수     0
지역           0
준공일자         0
건물형태        22
난방방식        75
승강기설치여부     98
단지내주차면수      0
전용면적         0
공급면적(공용)     0
임대보증금        0
임대료          0
실차량수         0
dtype: int64
```

In [24]:
```
train['건물형태'].unique()
```

Out[24]:
```
array(['계단식', '복도식', '혼합식', nan], dtype=object)
```

In [25]:
```
train['건물형태'].value_counts()
```

Out[25]:
```
건물형태
복도식    623
계단식    321
혼합식    191
Name: count, dtype: int64
```

In [26]:
```
# 'nan' 값을 NaN으로 변환
train['건물형태'] = train['건물형태'].replace('nan', np.nan)
train['건물형태'] = train['건물형태'].fillna('복도식')
```

In [27]:
```
train['건물형태'].unique()
```

Out[27]:
```
array(['계단식', '복도식', '혼합식'], dtype=object)
```

In [28]:
```
train['건물형태'].value_counts()
```

Out[28]:
```
건물형태
복도식    645
계단식    321
혼합식    191
Name: count, dtype: int64
```

In [29]:
```
train['난방방식'].unique()
```

Out[29]:
```
array(['개별가스난방', '지역난방', nan, '지역가스난방', '중앙가스난방', '개별유류난방', '중앙
난방',
       '지역유류난방', '중앙유류난방'], dtype=object)
```

In [30]:
```
# 'nan' 값을 NaN으로 변환
train['난방방식'] = train['난방방식'].replace('nan', np.nan)
```

In [31]:
```
train['난방방식'] = train['난방방식'].fillna('개별난방')
```

In [32]:
```
train['난방방식'].value_counts()
```

Out[32]:
```
난방방식
개별가스난방    568
지역난방      333
지역가스난방    120
개별난방       75
중앙가스난방     44
중앙난방       11
중앙유류난방      3
지역유류난방      2
개별유류난방      1
Name: count, dtype: int64
```

In [33]:
```
train['난방방식'].unique()
```

Out[33]:
```
array(['개별가스난방', '지역난방', '개별난방', '지역가스난방', '중앙가스난방', '개별유류난방',
'중앙난방',
       '지역유류난방', '중앙유류난방'], dtype=object)
```

In [34]:
```
train['승강기설치여부'].unique()
```

Out[34]:
```
array(['전체동 설치', nan, '일부동 설치', '미설치'], dtype=object)
```

In [35]:
```
# 'nan' 값을 NaN으로 변환
train['승강기설치여부'] = train['승강기설치여부'].replace('nan', np.nan)
train['승강기설치여부'] = train['승강기설치여부'].fillna('전체동 설치')
```

In [36]:
```
train['승강기설치여부'].value_counts()
```

Out[36]:
```
승강기설치여부
전체동 설치    1128
미설치         18
일부동 설치      11
Name: count, dtype: int64
```

In [37]:
```
train['승강기설치여부'].unique()
```

Out[37]:
```
array(['전체동 설치', '일부동 설치', '미설치'], dtype=object)
```

In [38]:
```
train.isna().sum()
```

Out[38]:
```
단지코드        0
단지명         0
총세대수        0
전용면적별세대수    0
지역          0
준공일자        0
건물형태        0
난방방식        0
승강기설치여부     0
단지내주차면수     0
전용면적        0
공급면적(공용)    0
임대보증금       0
임대료         0
실차량수        0
dtype: int64
```

In [39]:
```
test.isna().sum()
```

Out[39]:
```
단지코드        0
단지명         0
총세대수        0
전용면적별세대수    0
지역          0
준공일자        8
건물형태        4
난방방식        1
승강기설치여부     9
단지내주차면수     0
전용면적        0
공급면적(공용)    0
임대보증금       0
임대료         0
실차량수        0
dtype: int64
```

In [40]:
```
test['준공일자'] = test['준공일자'].astype(str).str[:4]
test.head()
```

Out[40]:

	단지코드	단지명	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	단지내주차면수	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0005	서울석촌도시형주택(공임10년)	20	6	서울	2012	복도식	개별가스난방	전체동설치	9	17.53	11.7251	50449000	263710	21
1	C0005	서울석촌도시형주택(공임10년)	20	10	서울	2012	복도식	개별가스난방	전체동설치	9	24.71	16.5275	52743000	321040	21
2	C0005	서울석촌도시형주택(공임10년)	20	4	서울	2012	복도식	개별가스난방	전체동설치	9	26.72	17.8720	53890000	332510	21
3	C0017	대구혁신센텀힐즈	822	228	대구경북	2018	계단식	지역난방	NaN	824	51.87	20.9266	29298000	411200	797
4	C0017	대구	822	56	대구	2018	계단	지역	NaN	824	59.85	24.1461	38550000	462600	797

단지코드	단지명	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	단지내주차면수	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
	혁신센텀힐즈			경북		식	난방							

In [41]:
```
# 'nan' 값을 NaN으로 변환
test['준공일자'] = test['준공일자'].replace('nan', np.nan)
# 최소값 계산
min_value = test['준공일자'].dropna().astype(int).min()

test['준공일자'] = train['준공일자'].fillna(min_value)
```

In [42]:
```
test['준공일자'].unique()
```

Out[42]:
```
array(['2013', '2014', '2011', '2007', '2012', '2018', '2010', '2008',
       '2009', '2017', '2016', '2019'], dtype=object)
```

In [43]:
```
# 'nan' 값을 NaN으로 변환
test['건물형태'] = test['건물형태'].replace('nan', np.nan)
test['건물형태'] = test['건물형태'].fillna('복도식')
```

In [44]:
```
# 'nan' 값을 NaN으로 변환
test['승강기설치여부'] = test['승강기설치여부'].replace('nan', np.nan)
test['승강기설치여부'] = test['승강기설치여부'].fillna('미설치')
```

In [45]:
```
# 'nan' 값을 NaN으로 변환
test['건물형태'] = test['건물형태'].replace('nan', np.nan)
test['건물형태'] = test['건물형태'].fillna('복도식')
```

In [46]:
```
# 'nan' 값을 NaN으로 변환
test['난방방식'] = test['난방방식'].replace('nan', np.nan)
test['난방방식'] = test['난방방식'].fillna('지역유류난방')
```

In [47]:
```
test['승강기설치여부'] = test['승강기설치여부'].replace({'전체동 설치': '설치'})
```

In [48]:
```
test['승강기설치여부'].value_counts()
```

Out[48]:
```
승강기설치여부
설치     95
미설치     9
Name: count, dtype: int64
```

In [49]:
```
test.isna().sum()
```

Out[49]:
```
단지코드         0
단지명          0
총세대수         0
전용면적별세대수     0
지역           0
준공일자         0
건물형태         0
난방방식         0
승강기설치여부      0
단지내주차면수      0
전용면적         0
공급면적(공용)     0
임대보증금        0
임대료          0
실차량수         0
dtype: int64
```

(2) 불필요한 칼럼 제거

- **세부 요구사항**
  - 단지명 : 분석단위로 볼때 단일값.
  - 단지내주차면수 :
    - 본 문제는 등록차량수를 예측하고, 그것을 기반으로 주차면수를 정하는 것이 목적입니다.
    - 그런데, 역으로 주차면수 먼저 정한후 그것을 기반으로 등록차량수를 예측하는 것은 성립될 수 없습니다.

- 불필요한 정보 제거하기

In [50]:
```
train.head()
```

Out[50]:

	단지코드	단지명	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	단지내주차면수	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0001	엘에이치서초4단지	78	35	서울	2013	계단식	개별가스난방	전체동설치	120	51.89	19.2603	50758000	620370	109
1	C0001	엘에이치서초4단지	78	43	서울	2013	계단식	개별가스난방	전체동설치	120	59.93	22.2446	63166000	665490	109
2	C0002	LH삼성아파트	35	26	서울	2013	복도식	개별가스난방	전체동설치	47	27.75	16.5375	63062000	458640	35
3	C0002	LH삼성아파트	35	9	서울	2013	복도식	개별가스난방	전체동설치	47	29.08	17.3302	63062000	481560	35
4	C0003	강남LH8단지	88	7	서울	2013	계단식	개별가스난방	전체동설치	106	59.47	21.9462	72190000	586540	88

In [51]:
```
train.drop(['단지명', '단지내주차면수'], axis=1, inplace=True)
train.head()
```

Out[51]:

	단지코드	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0001	78	35	서울	2013	계단식	개별가스난방	전체동 설치	51.89	19.2603	50758000	620370	109
1	C0001	78	43	서울	2013	계단식	개별가스난방	전체동 설치	59.93	22.2446	63166000	665490	109
2	C0002	35	26	서울	2013	복도식	개별가스난방	전체동 설치	27.75	16.5375	63062000	458640	35
3	C0002	35	9	서울	2013	복도식	개별가스난방	전체동 설치	29.08	17.3302	63062000	481560	35
4	C0003	88	7	서울	2013	계단식	개별가스난방	전체동 설치	59.47	21.9462	72190000	586540	88

In [52]:
```
test.head()
```

Out[52]:

	단지코드	단지명	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	단지내주차면수	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0005	서울석촌도시형주택(공임10년)	20	6	서울	2013	복도식	개별가스난방	설치	9	17.53	11.7251	50449000	263710	21
1	C0005	서울석촌도시형주택(공임10년)	20	10	서울	2013	복도식	개별가스난방	설치	9	24.71	16.5275	52743000	321040	21
2	C0005	서울석촌도시형주택(공임10년)	20	4	서울	2013	복도식	개별가스난방	설치	9	26.72	17.8720	53890000	332510	21
3	C0017	대구혁신센텀힐즈	822	228	대구경북	2013	계단식	지역난방	미설치	824	51.87	20.9266	29298000	411200	797
4	C0017	대구혁신센텀힐즈	822	56	대구경북	2013	계단식	지역난방	미설치	824	59.85	24.1461	38550000	462600	797

In [53]:
```
test.drop(['단지명', '단지내주차면수'], axis=1, inplace=True)
test.head()
```

Out[53]:

	단지코드	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0005	20	6	서울	2013	복도식	개별가스난방	설치	17.53	11.7251	50449000	263710	21
1	C0005	20	10	서울	2013	복도식	개별가스난방	설치	24.71	16.5275	52743000	321040	21
2	C0005	20	4	서울	2013	복도식	개별가스난방	설치	26.72	17.8720	53890000	332510	21
3	C0017	822	228	대구경북	2013	계단식	지역난방	미설치	51.87	20.9266	29298000	411200	797
4	C0017	822	56	대구경북	2013	계단식	지역난방	미설치	59.85	24.1461	38550000	462600	797

3.데이터 전처리②

- **세부 요구사항**
  - 1) 데이터프레임을 두가지 형태로 나눕니다.
  - 2) 상세 데이터 집계하기
  - 3) 단지별 데이터와 상세 데이터 집계 결과 merge 시키기

(1) 데이터프레임 두개로 나누기

- **세부 요구사항**
  - 단지별 데이터
    - 단지코드, 총세대수, 지역, 준공일자, 건물형태, 난방방식, 승강기설치여부, 실차량수
    - 단지별 데이터를 분할 한 후, 중복행을 제거합니다.
    - 중복행이 잘 제거 되었는지 확인합니다.
  - 상세 데이터
    - 단지코드, 전용면적별세대수,전용면적, 공급면적(공용),임대보증금,임대료

In [54]:
```
train.head()
```

Out[54]:

	단지코드	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0001	78	35	서울	2013	계단식	개별가스난방	전체동 설치	51.89	19.2603	50758000	620370	109
1	C0001	78	43	서울	2013	계단식	개별가스난방	전체동 설치	59.93	22.2446	63166000	665490	109
2	C0002	35	26	서울	2013	복도식	개별가스난방	전체동 설치	27.75	16.5375	63062000	458640	35
3	C0002	35	9	서울	2013	복도식	개별가스난방	전체동 설치	29.08	17.3302	63062000	481560	35
4	C0003	88	7	서울	2013	계단식	개별가스난방	전체동 설치	59.47	21.9462	72190000	586540	88

In [55]:
```
drop_cols = ['전용면적별세대수', '전용면적', '공급면적(공용)', '임대보증금', '임대료']
apartment_data = train.drop(drop_cols, axis=1)
```

In [56]:
```
apartment_data.head()
```

Out[56]:

	단지코드	총세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	실차량수
0	C0001	78	서울	2013	계단식	개별가스난방	전체동 설치	109
1	C0001	78	서울	2013	계단식	개별가스난방	전체동 설치	109
2	C0002	35	서울	2013	복도식	개별가스난방	전체동 설치	35
3	C0002	35	서울	2013	복도식	개별가스난방	전체동 설치	35
4	C0003	88	서울	2013	계단식	개별가스난방	전체동 설치	88

In [57]:
```
drop_cols = ['총세대수', '지역', '준공일자', '건물형태', '난방방식', '승강기설치여부', '실차량
detail_data = train.drop(drop_cols, axis=1)
```

In [58]:
```
detail_data.head()
```

Out[58]:

	단지코드	전용면적별세대수	전용면적	공급면적(공용)	임대보증금	임대료
0	C0001	35	51.89	19.2603	50758000	620370
1	C0001	43	59.93	22.2446	63166000	665490
2	C0002	26	27.75	16.5375	63062000	458640
3	C0002	9	29.08	17.3302	63062000	481560
4	C0003	7	59.47	21.9462	72190000	586540

(2) 상세 데이터 집계하기

- **세부 요구사항**
  - 전용면적 구간으로 나누기
    - 전용면적을 의미 있는 단위로 자른 칼럼 만들기 (pd.cut)
    - 구간별 세대 수 집계
    - pivot
  - 임대보증금 혹은 임대료 : 적절하게 집계하기.
    - 평균, 가중평균, 중앙값 중에서 고려하기.
  - 임대건물구분, 공급유형
    - 전체 면적을 계산합니다.(공급면적 * 세대수)
    - 임대건물구분 별, 면적 합계를 구한 후, 면적 비율을 계산합니다.
    - 역시, 공급 유형별, 면적 합계를 구한 후, 면적 비율을 계산합니다.

1) 전용면적

In [59]:
```
#단지내 전용면적별 총면적('전용면적' * '전용면적별 세대수') 계산
detail_data['총면적'] = detail_data['전용면적별세대수'] * detail_data['전용면적']
```

In [60]:
```
detail_data
```

Out[60]:

	단지코드	전용면적별세대수	전용면적	공급면적(공용)	임대보증금	임대료	총면적
0	C0001	35	51.89	19.2603	50758000	620370	1816.15
1	C0001	43	59.93	22.2446	63166000	665490	2576.99
2	C0002	26	27.75	16.5375	63062000	458640	721.50
3	C0002	9	29.08	17.3302	63062000	481560	261.72
4	C0003	7	59.47	21.9462	72190000	586540	416.29
...	...	...	...	...	...	...	...
1152	C0356	956	26.37	12.7500	9931000	134540	25209.72
1153	C0358	66	24.83	15.1557	2129000	42350	1638.78
1154	C0358	54	33.84	20.6553	2902000	57730	1827.36
1155	C0359	149	26.37	13.3800	7134000	118880	3929.13
1156	C0359	298	31.32	13.8500	8122000	131140	9333.36

1157 rows × 7 columns

In [61]:
```
# 전용면적을 의미 있는 단위로 나누기
bins = [-np.inf, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, np.inf]  # 구간 설정
labels = ['30이하', '40이하', '50이하', '60이하', '70이하', '80이하', '90이하', '100이하', '11
detail_data['전용면적구간'] = pd.cut(detail_data['전용면적'], bins=bins, labels=labels)
```

In [62]:
```
df = detail_data.pivot_table(index='단지코드', columns='전용면적구간', values='총면적', aggfun
```

In [63]:
```
df
```

Out[63]:

전용면적구간	30이하	40이하	50이하	60이하	70이하	80이하	90이하	100이하	110이하	120이하	121이상
단지코드											
C0001	0.00	0.00	0.0	4393.14	0.0	0.0	0.00	0.0	0.0	0.0	0.0
C0002	983.22	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.0
C0003	0.00	0.00	0.0	5244.69	0.0	0.0	0.00	0.0	0.0	0.0	0.0
C0004	0.00	0.00	0.0	8994.00	0.0	16189.6	9423.74	0.0	0.0	0.0	0.0
C0006	309.50	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
C1341	3462.06	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.0
C1354	28242.27	9333.36	0.0	896.58	0.0	0.0	0.00	0.0	0.0	0.0	0.0
C2307	4874.28	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.0
C2343	1979.20	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.0
C2349	998.80	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.0

345 rows × 11 columns

In [64]:
```
#• 단지별, 전용면적구간 별, 총면적 집계(groupby)
detail_data.groupby(by=['단지코드', '전용면적구간'])['총면적'].sum()
```

Out[64]:
```
단지코드   전용면적구간
C0001  30이하         0.00
       40이하         0.00
       50이하         0.00
       60이하      4393.14
       70이하         0.00
                  ...
C2349  90이하         0.00
       100이하        0.00
       110이하        0.00
       120이하        0.00
       121이상        0.00
Name: 총면적, Length: 3795, dtype: float64
```

2) 임대보증금, 임대료 집계 하기(평균)

In [65]:
```
detail_data.isna().sum()
```

Out[65]:
```
단지코드          0
전용면적별세대수      0
전용면적          0
공급면적(공용)      0
임대보증금         0
임대료           0
총면적           0
전용면적구간        0
dtype: int64
```

In [66]:
```
# 각 방법으로 집계하기
# 평균
mean_rent_deposit = detail_data['임대보증금'].mean()
mean_rent_fee = detail_data['임대료'].mean()

# 가중평균
weighted_mean_rent_deposit = (detail_data['임대보증금'] * detail_data['전용면적별세대수']).sum
weighted_mean_rent_fee = (detail_data['임대료'] * detail_data['전용면적별세대수']).sum() / det

# 중앙값
median_rent_deposit = detail_data['임대보증금'].median()
median_rent_fee = detail_data['임대료'].median()

# # 각 값을 천 단위 구분하여 문자열로 변환
# mean_rent_deposit = '{:,.0f}'.format(mean_rent_deposit)
# mean_rent_fee = '{:,.0f}'.format(mean_rent_fee)
# weighted_mean_rent_deposit = '{:,.0f}'.format(weighted_mean_rent_deposit)
# weighted_mean_rent_fee = '{:,.0f}'.format(weighted_mean_rent_fee)
# median_rent_deposit = '{:,.0f}'.format(median_rent_deposit)
# median_rent_fee = '{:,.0f}'.format(median_rent_fee)

print("평균 임대보증금:", mean_rent_deposit)
print("평균 임대료:", mean_rent_fee)
print("가중평균 임대보증금:", weighted_mean_rent_deposit)
print("가중평균 임대료:", weighted_mean_rent_fee)
print("중앙값 임대보증금:", median_rent_deposit)  # 중앙값이 적절하게 표현한 것같음
print("중앙값 임대료:", median_rent_fee)
```

```
평균 임대보증금: 28507893.690579083
평균 임대료: 225940.8815903198
가중평균 임대보증금: 26294545.86015175
가중평균 임대료: 220901.02998558537
중앙값 임대보증금: 19973000.0
중앙값 임대료: 184290.0
```

In [ ]:

3) 단지별 총 면적 구하기

In [67]:
```
# 전체 면적 계산
detail_data['전체면적'] = detail_data['공급면적(공용)'] * detail_data['전용면적별세대수']
```

In [68]:
```
detail_data
```

Out[68]:

	단지코드	전용면적별 세대수	전용면적	공급면적(공용)	임대보증금	임대료	총면적	전용면적구간	전체면적
0	C0001	35	51.89	19.2603	50758000	620370	1816.15	60이하	674.1105
1	C0001	43	59.93	22.2446	63166000	665490	2576.99	60이하	956.5178
2	C0002	26	27.75	16.5375	63062000	458640	721.50	30이하	429.9750
3	C0002	9	29.08	17.3302	63062000	481560	261.72	30이하	155.9718
4	C0003	7	59.47	21.9462	72190000	586540	416.29	60이하	153.6234
...	...	...	...	...	...	...	...	...	...
1152	C0356	956	26.37	12.7500	9931000	134540	25209.72	30이하	12189.0000
1153	C0358	66	24.83	15.1557	2129000	42350	1638.78	30이하	1000.2762
1154	C0358	54	33.84	20.6553	2902000	57730	1827.36	40이하	1115.3862
1155	C0359	149	26.37	13.3800	7134000	118880	3929.13	30이하	1993.6200
1156	C0359	298	31.32	13.8500	8122000	131140	9333.36	40이하	4127.3000

1157 rows × 9 columns

In [69]:

```
# 임대건물구분 별 전체 면적 합계 계산
total_area_by_building_type = detail_data.groupby('임대료')['전체면적'].sum()

# 공급유형별 전체 면적 합계 계산
total_area_by_supply_type = detail_data.groupby('공급면적(공용)')['전체면적'].sum()

# 임대건물구분 별 면적 비율 계산
building_type_area_ratio = total_area_by_building_type / total_area_by_building_type.sum()

# 공급유형별 면적 비율 계산
supply_type_area_ratio = total_area_by_supply_type / total_area_by_supply_type.sum()

print("임대건물구분 별 전체 면적 합계:")
print(total_area_by_building_type)
print("\n공급유형별 전체 면적 합계:")
print(total_area_by_supply_type)
print("\n임대건물구분 별 면적 비율:")
print(building_type_area_ratio)
print("\n공급유형별 면적 비율:")
print(supply_type_area_ratio)
```

```
임대건물구분 별 전체 면적 합계:
임대료
0          129526.5522
36180          52.8619
38670           8.0270
39920         656.0000
41460         330.7200
              ...
879810       3421.0120
945230       4504.6413
962150       1320.3099
1042230      2870.9650
1058030       519.6611
Name: 전체면적, Length: 664, dtype: float64

공급유형별 전체 면적 합계:
공급면적(공용)
5.8500      2281.5000
6.2400       330.7200
6.3840      1774.7520
6.4976       259.9040
6.5320       770.7760
              ...
39.2481     2825.8632
39.4485     3865.9530
39.7714    15550.6174
41.3441     1653.7640
42.7600     1111.7600
Name: 전체면적, Length: 1125, dtype: float64

임대건물구분 별 면적 비율:
임대료
0          0.034262
36180      0.000014
38670      0.000002
39920      0.000174
41460      0.000087
             ...
879810     0.000905
945230     0.001192
962150     0.000349
1042230    0.000759
1058030    0.000137
Name: 전체면적, Length: 664, dtype: float64

공급유형별 면적 비율:
공급면적(공용)
5.8500     0.000604
6.2400     0.000087
6.3840     0.000469
6.4976     0.000069
6.5320     0.000204
             ...
39.2481    0.000747
39.4485    0.001023
39.7714    0.004113
41.3441    0.000437
42.7600    0.000294
Name: 전체면적, Length: 1125, dtype: float64
```

In [ ]:

In [ ]:

(3) 합치기

- **세부 요구사항**
  - [단지별 데이터]를 기준으로 상세데이터로 만든 데이터셋을 하나씩 merge 합니다.
  - merge를 사용할 때, **how = 'left', on = '단지코드'** 옵션을 이용합니다.
    - [단지별 데이터]가 기준(left)입니다.

- 단지별 데이터 + 전용면적별 세대수

In [70]:
```
trains = pd.merge(train, df, on='단지코드', how='left')
```

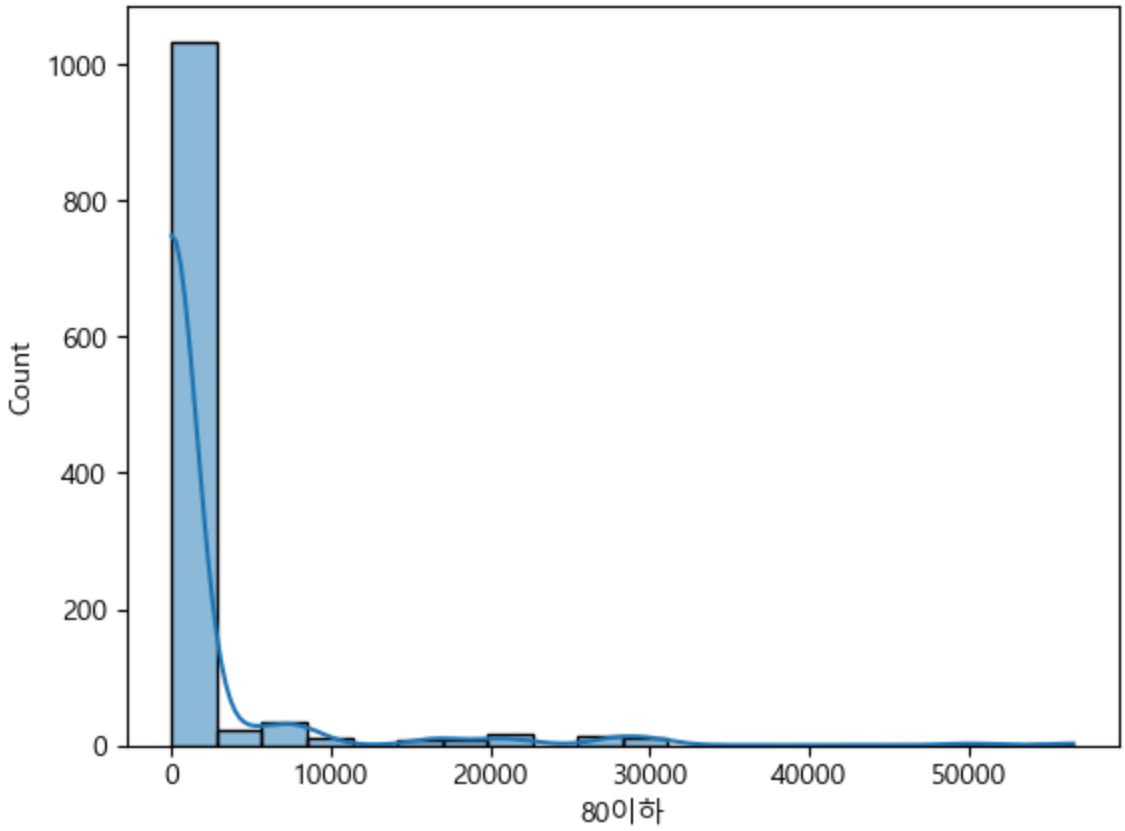
In [71]:
```
trains
```

Out[71]:

	단지코드	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	전용면적	공급면적(공용)	...	40이하	50이하	60이하	70이하	80이하	90이하
0	C0001	78	35	서울	2013	계단식	개별가스난방	전체동설치	51.89	19.2603	...	0.00	0.0	4393.14	0.0	0.0	0.0
1	C0001	78	43	서울	2013	계단식	개별가스난방	전체동설치	59.93	22.2446	...	0.00	0.0	4393.14	0.0	0.0	0.0
2	C0002	35	26	서울	2013	복도식	개별가스난방	전체동설치	27.75	16.5375	...	0.00	0.0	0.00	0.0	0.0	0.0
3	C0002	35	9	서울	2013	복도식	개별가스난방	전체동설치	29.08	17.3302	...	0.00	0.0	0.00	0.0	0.0	0.0
4	C0003	88	7	서울	2013	계단식	개별가스난방	전체동설치	59.47	21.9462	...	0.00	0.0	5244.69	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1152	C0356	956	956	경기	1994	복도식	지역가스난방	전체동설치	26.37	12.7500	...	0.00	0.0	0.00	0.0	0.0	0.0
1153	C0358	120	66	강원	2020	복도식	개별난방	전체동설치	24.83	15.1557	...	1827.36	0.0	0.00	0.0	0.0	0.0

	단지코드	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	전용면적	공급면적(공용)	...	40이하	50이하	60이하	70이하	80이하	90이하
1154	C0358	120	54	강원	2020	복도식	개별난방	전체동설치	33.84	20.6553	...	1827.36	0.0	0.00	0.0	0.0	0.0
1155	C0359	447	149	대구경북	1994	복도식	중앙유류난방	전체동설치	26.37	13.3800	...	9333.36	0.0	0.00	0.0	0.0	0.0
1156	C0359	447	298	대구경북	1994	복도식	중앙유류난방	전체동설치	31.32	13.8500	...	9333.36	0.0	0.00	0.0	0.0	0.0

In [72]:
```
sns.histplot(trains['80이하'], bins=20, kde=True)
plt.show()
```



- 평균 임대 보증금/임대료 합치기

In [73]:
```
rent_deposit = detail_data.groupby(by='단지코드')['임대보증금'].mean()
rent_fee = detail_data.groupby(by='단지코드')['임대료'].mean()
```

In [74]:
```
rent_deposit = pd.DataFrame(rent_deposit)
rent_fee = pd.DataFrame(rent_fee)
```

In [75]:
```
mean_fee_deposit = pd.merge(rent_deposit, rent_fee, on='단지코드', how='left')
```

In [76]:
```
mean_fee_deposit
```

Out[76]:

단지코드	임대보증금	임대료
C0001	5.696200e+07	642930.000000
C0002	6.306200e+07	470100.000000
C0003	7.219000e+07	586540.000000
C0004	1.015167e+08	950305.000000
C0006	5.522750e+07	340148.333333
...	...	...
C1341	1.188600e+07	93000.000000
C1354	8.092875e+06	111848.750000
C2307	1.180250e+07	94055.000000
C2343	1.211700e+07	108000.000000
C2349	9.697000e+06	89270.000000

345 rows × 2 columns

4.데이터셋 저장하기

- **세부 요구사항**
  - joblib.dump를 이용하시오.
  - 저장할 파일의 확장자는 보통 .pkl 입니다.

In [77]:
```
joblib.dump(trains, path + 'trains.pkl')
```

Out[77]:
```
['C:/Users/User/program/mini_pjt/mini_3/실습파일_에이블러용/데이터/trains.pkl']
```

In [78]:
```
joblib.dump(mean_fee_deposit, path + 'mean_fee_deposit.pkl')
```

Out[78]:
```
['C:/Users/User/program/mini_pjt/mini_3/실습파일_에이블러용/데이터/mean_fee_deposit.pkl']
```

In [79]:
```
test
```

Out[79]:

	단지코드	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	전용면적	공급면적(공용)	임대보증금	임대료	실차량수
0	C0005	20	6	서울	2013	복도식	개별가스난방	설치	17.53	11.7251	50449000	263710	21
1	C0005	20	10	서울	2013	복도식	개별가스난방	설치	24.71	16.5275	52743000	321040	21
2	C0005	20	4	서울	2013	복도식	개별가스난방	설치	26.72	17.8720	53890000	332510	21
3	C0017	822	228	대구경북	2013	계단식	지역난방	미설치	51.87	20.9266	29298000	411200	797
4	C0017	822	56	대구경북	2013	계단식	지역난방	미설치	59.85	24.1461	38550000	462600	797
...	...	...	...	...	...	...	...	...	...	...	...	...	...
99	C0353	768	90	대전충남	2014	복도식	중앙난방	설치	40.32	16.5100	8848000	122290	123
100	C0360	588	98	서울	2014	복도식	지역난방	미설치	51.37	21.5569	183228000	0	559
101	C0360	588	186	서울	2013	복도식	지역난방	미설치	51.39	21.5652	183228000	0	559
102	C0360	588	102	서울	2013	복도식	지역난방	미설치	59.76	25.0776	215057000	0	559
103	C0360	588	202	서울	2013	복도식	지역난방	미설치	59.80	25.0944	215057000	0	559

104 rows × 13 columns

In [90]:
```
test['총면적'] = test['전용면적별세대수'] * test['전용면적']
```

In [91]:
```
joblib.dump(test, path + 'test2.pkl')
```

Out[91]:
```
['C:/Users/User/program/mini_pjt/mini_3/실습파일_에이블러용/데이터/test2.pkl']
```

In [86]:
```
train['총면적'] = detail_data['총면적']
```

In [92]:
```
train
```

Out[92]:

	단지코드	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	전용면적	공급면적(공용)	임대보증금	임대료	실차량수	총면적
0	C0001	78	35	서울	2013	계단식	개별가스난방	전체동설치	51.89	19.2603	50758000	620370	109	1816.15
1	C0001	78	43	서울	2013	계단식	개별가스난방	전체동설치	59.93	22.2446	63166000	665490	109	2576.99
2	C0002	35	26	서울	2013	복도식	개별가스난방	전체동설치	27.75	16.5375	63062000	458640	35	721.50
3	C0002	35	9	서울	2013	복도식	개별가스난방	전체동설치	29.08	17.3302	63062000	481560	35	261.72
4	C0003	88	7	서울	2013	계단식	개별가스난방	전체동설치	59.47	21.9462	72190000	586540	88	416.29
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1152	C0356	956	956	경기	1994	복도식	지역가스난방	전체동설치	26.37	12.7500	9931000	134540	243	25209.72
1153	C0358	120	66	강원	2020	복도식	개별난방	전체동설치	24.83	15.1557	2129000	42350	47	1638.78

	단지코드	총세대수	전용면적별세대수	지역	준공일자	건물형태	난방방식	승강기설치여부	전용면적	공급면적(공용)	임대보증금	임대료	실차량수	총면적
1154	C0358	120	54	강원	2020	복도식	개별난방	전체동설치	33.84	20.6553	2902000	57730	47	1827.36
1155	C0359	447	149	대구경북	1994	복도식	중앙유류난방	전체동설치	26.37	13.3800	7134000	118880	78	3929.13
1156	C0359	447	298	대구경북	1994	복도식	중앙유류난방	전체동설치	31.32	13.8500	8122000	131140	78	9333.36

1157 rows × 14 columns

In [94]:
```
joblib.dump(train, path + 'train2.pkl')
```

Out[94]:
```
['C:/Users/User/program/mini_pjt/mini_3/실습파일_에이블러용/데이터/train2.pkl']
```

In [ ]:
```
```