2주차

AI를 활용한 계량연구

R X ChatGPT

Lion

Graph Visualization

실 습

5.1 실습 Intro

- <프롬프트 구조>

- 목표 : 나는 A를 하려고해. # 내가 원하는 목적

- 구체적인 그림 : B, C D 가 구현되었으면 좋겠어.

- 경우의 수 분류 : A인 경우에는 ㄱ으로 분류하고 ,
  B 인경우에는 ㄴ으로 분류해줘.

- 예시 제공 : df의 "서울특별시 종로구 청운동" 은 df_cleaned에서 서울특별시 / 종로구 / 청운동 3개 값으로 나누어서 저장해야 해.

- 결과 : ChatGPT가 해줬으면 하는 것
  ex) ~하는 파이썬 코드를 짜줘.

[프롬프트 예시]
나는 파이썬을 이용해서 데이터를 처리하고, 그래프를 그리려고 해.

df 데이터프레임 뒤쪽에 새로운 열 3개를 추가한 뒤, B열 "시군구" 데이터들을 공백을 기준으로 나누어서 추가하고, df_cleaned라는 데이터프레임으로 저장할거야.

예를 들어 df의 "서울특별시 종로구 청운동" 은 df_cleaned에서 서울특별시 / 종로구 / 청운동 3개 값으로 나누어서 저장해야해. "서울특별시 성동구 상왕십리동"의 경우 서울특별시 /성동구 / 상왕십리동 3개 값으로 나누어 저장할거야.

열제목은 "소재지 시", "소재지 구", "소재지 동"라고 붙이는 코드짜줘.

5.1 실습 Intro

- **<주의사항>**

- 한글을 폰트 관련 작업을 미리 하고 시작하는 것을 추천

- ChatGPT와 이야기 할 때는 행/열의 “한글이름” 말고, 알파벳과 숫자로! Ex) 3행, B열 계약일자 등

- 열제목은 정확하게 지시해주어야 합니다. ex) 전용면적 (X), 전용면적(㎡) (O)

- 현재 어떤 데이터프레임을 쓰고 있는건지 기억하기

- 코드는 특히 중복해서 짜주는 경향이 많다

데이터프레임(DataFrame)은 판다스(Pandas) 라이브러리에서 제공하는 데이터 구조로, 표 형식의 2차원 데이터 구조를 나타냅니다. 데이터프레임은 엑셀 스프레드시트와 유사하며, 행과 열로 구성되어 있습니다. 각 열은 데이터 시리즈(Series)를 나타내며, 행과 열에 각각 인덱스가 있습니다.

데이터프레임을 쉽게 이해하기 위해 다음과 같은 예시를 생각해 보세요:

인덱스	이름	나이	성별
0	홍길동	25	남자
1	이순신	45	남자
2	유관순	19	여자

위 표는 간단한 데이터프레임의 예입니다. 각 행은 사람에 대한 정보를 나타내고, 각 열은 그 사람의 속성을 나타냅니다.

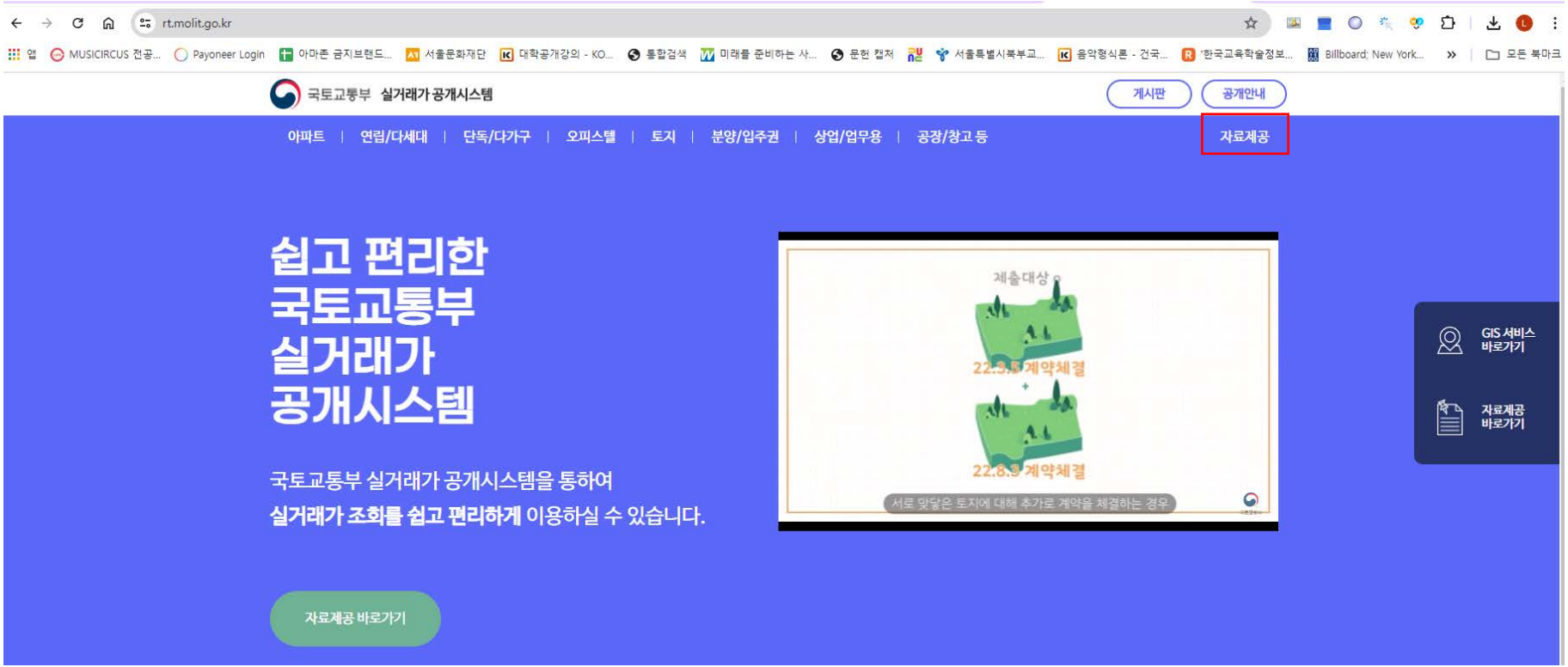
주요 특징

1. **2차원 구조**: 데이터프레임은 행(row)과 열(column)로 이루어진 2차원 데이터 구조입니다.
2. **레이블**: 각 행과 열에는 레이블이 있어 데이터에 쉽게 접근할 수 있습니다.
3. **다양한 데이터 타입**: 데이터프레임의 각 열은 서로 다른 데이터 타입을 가질 수 있습니다 (숫자, 문자열, 날짜 등).

주요 기능

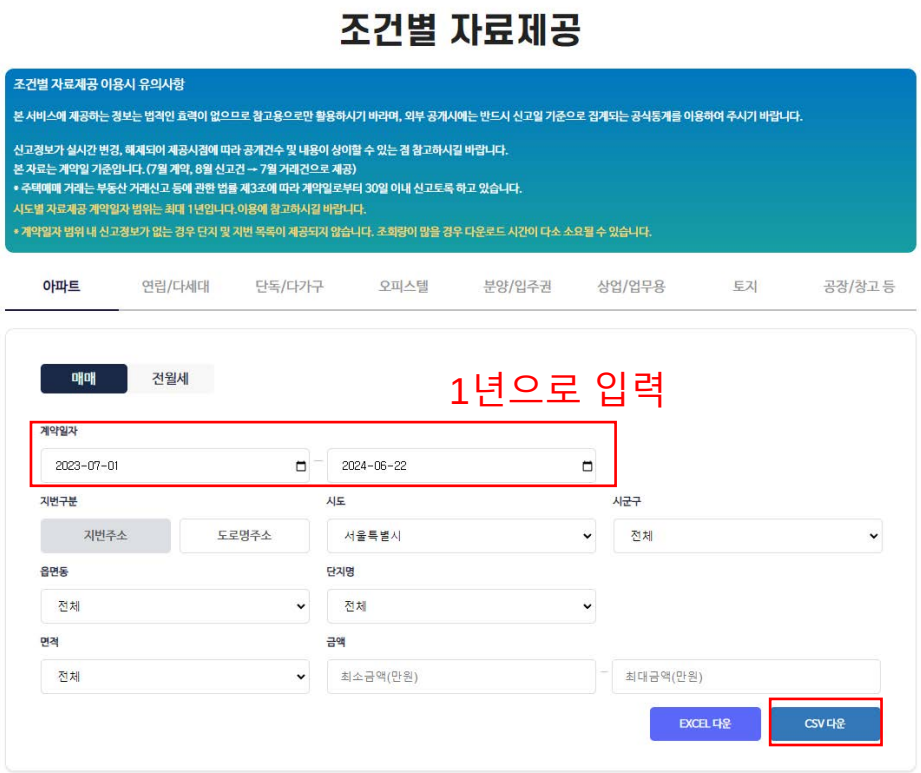
1. **데이터 조작**: 데이터 추가, 삭제, 정렬, 필터링 등 다양한 데이터를 조작하는 기능을 제공합니다.
2. **통계 연산**: 데이터프레임은 평균, 합계, 표준편차 등 다양한 통계 연산 기능을 지원합니다.
3. **데이터 변환**: 다른 데이터 형식(CSV, 엑셀, SQL 등)으로부터 데이터를 불러오거나 저장할 수 있습니다.

5.2 자료 다운로드 및 불러오기



검색창 입력 : “국토부 실거래가 다운로드”

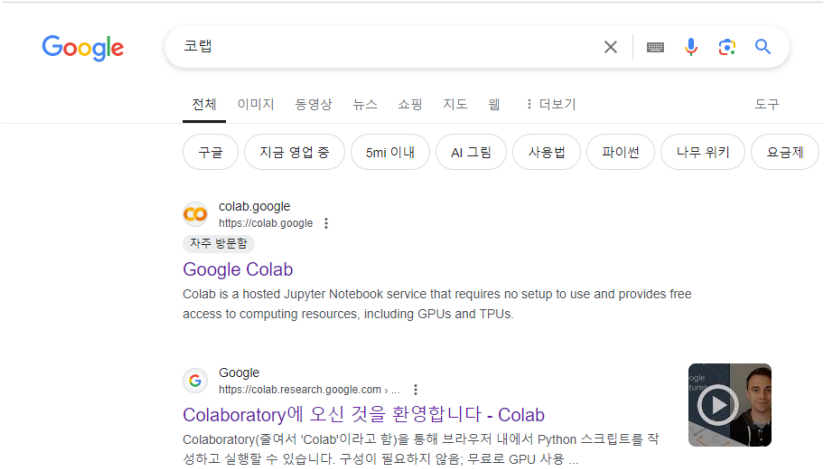
5.2 자료 다운로드 및 불러오기

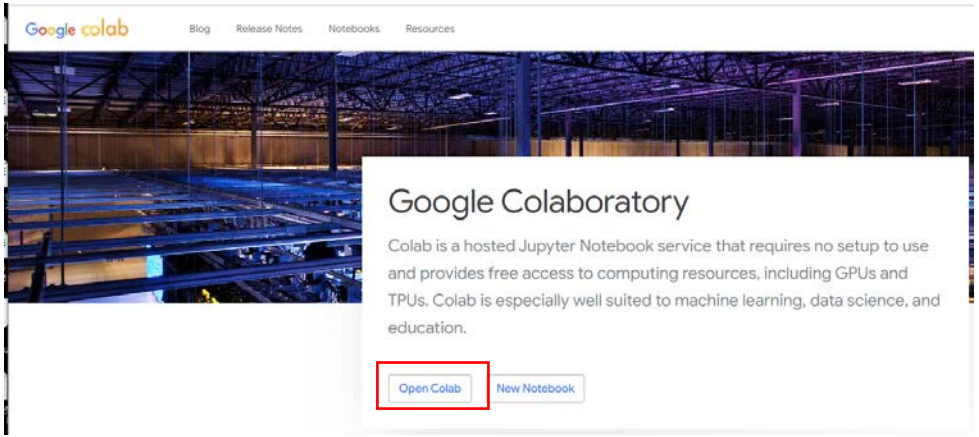


다운받고 나면
파일명은 영문+숫자로 변경
(UTF 오류 피하기 위함)

데이터는 1년 단위로 제공 : 계약일자 23.07.01~24.06.22(오늘) + 서울시

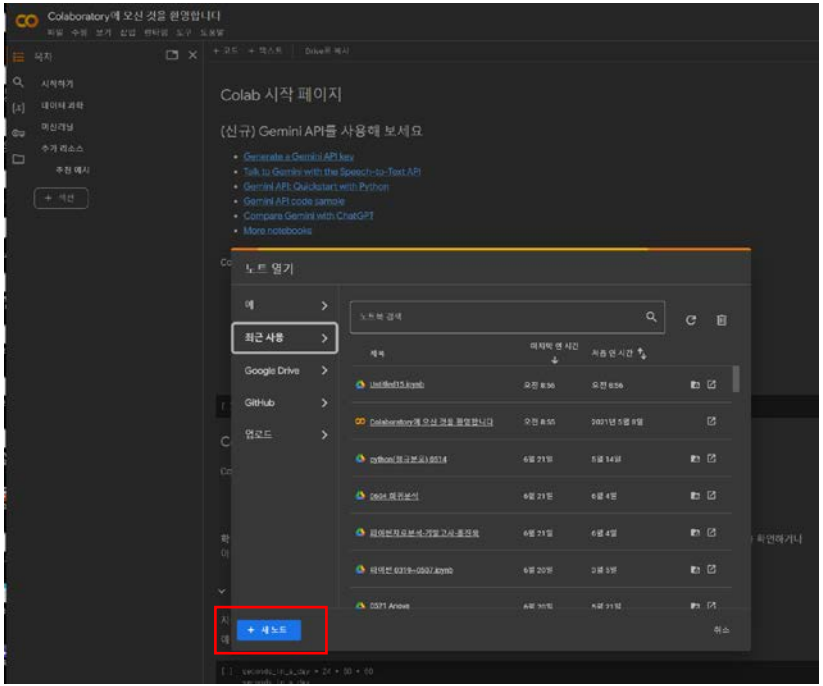
5.2 자료 다운로드 및 불러오기

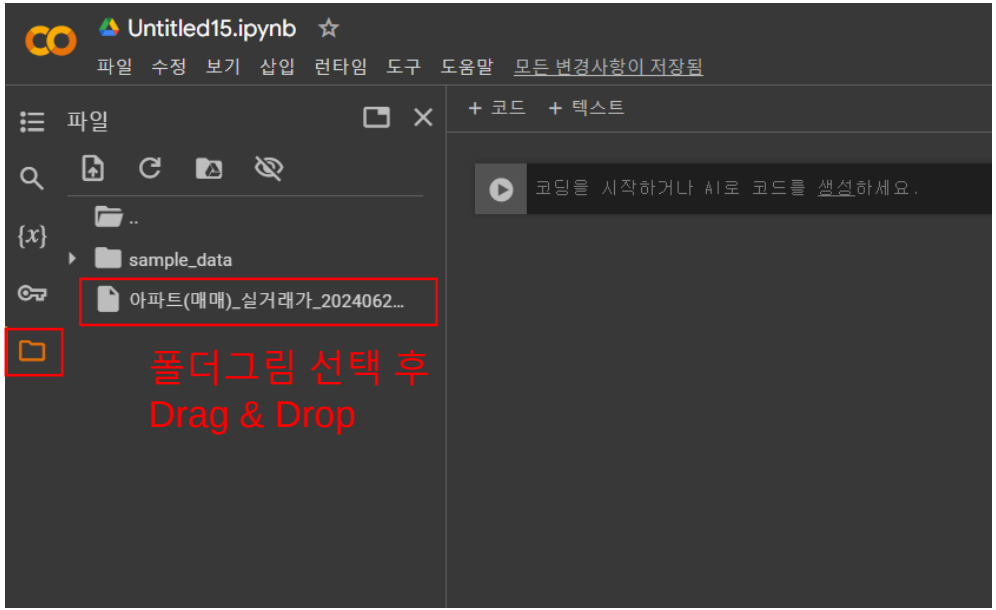




검색창에 " 코랩" 입력 후 Open Colab

5.2 자료 다운로드 및 불러오기





코랩 : 파이썬 언어를 온라인에서 구현할 수 있는 웹사이트. (by Google)

5.2 자료 다운로드 및 불러오기

```
#한글깨짐 방지를 위한 글꼴 설치
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf

#런타임 - 세션 재시작 필요(Ctrl + M)
```

1. 왼쪽 코드 입력 후 “세션 재시작”

2. 세션 재시작 이후
오른쪽 코드 입력

```
[1]  #나눔고딕으로 설정
     import matplotlib.pyplot as plt

     plt.rc('font', family='NanumBarunGothic')
```

그래프 시각화과정이 한글과 친하지 않아, 글꼴깨짐을 방지하는 팁

5.2 자료 다운로드 및 불러오기

코랩에서 "apt_price_seoul_20240622091810.csv" 파일을 불러와서 df로 저장하려고 해

- Python 창들을
“복사-붙여넣기” 해주세요!
(같은 창에 연이어서 복붙 가능)

- 실행은 ctrl + enter

1. `pandas` 라이브러리 임포트

python 코드 복사

```python
import pandas as pd
```

2. 파일 업로드

다음 코드를 실행하면 파일 업로드 창이 뜨고, 여기서 "아파트(매매)_실거래가_20240622091810.csv" 파일을 선택할 수 있습니다.

python 코드 복사

```python
from google.colab import files

uploaded = files.upload()
```

3. 업로드된 파일을 데이터프레임으로 불러오기

업로드가 완료되면 파일명을 이용하여 `pandas` 데이터프레임으로 불러옵니다.

python 코드 복사

```python
import pandas as pd

# 업로드된 파일의 이름을 확인하고 불러옵니다.
file_name = "아파트(매매)_실거래가_20240622091810.csv"
df = pd.read_csv(file_name)

# 데이터프레임을 확인합니다.
print(df.head())
```

Pandas : 데이터 조작과 분석을 위한 파이썬 라이브러리

5.3 자료 전처리(1)- 1차. 미신고건, 계약 해제건 삭제

국토교통부에서는 정부3.0 및 공공데이터 개방, 실거래자료 요청 증가에 따라,
부동산 거래 가격 및 거래 동향을 보다 정확하고 신속히 파악할 수 있도록 **부동산 거래신고제를 통해 수집된 실거래 자료를 공개**하고 있습니다.

매매 공개 대상

매매 실거래 공개는 2006년 1월부터 부동산거래신고 및 주택거래신고를 한 주택(아파트, 연립/다세대, 단독/다가구), 오피스텔, 토지, 상업 업무용, 공장 • 창고 등 부동산 및 2007년 6월29일 이후 체결된 아파트 분양/입주권을 대상으로 하고 있습니다.

전월세 공개 대상

전월세가 실거래가 공개는 2021년 6월부터 임대차계약 신고를 한 주택(아파트, 연립/다세대, 단독/다가구, 오피스텔) 및 2011년 1월부터 주민센터 및 일부 공개 가능한 대법원 등기소의 주택 확정일자 자료를 대상으로 하고 있습니다.

국토교통부 실거래가 공개시스템에서 제공하는 부동산 실거래가 공개는 활용목적, 집계기준 및 관리범위 등이 한국부동산원 부동산통계정보 시스템(R-ONE)에서 제공하는 부동산거래현황통계와 상이하오니 활용 시 참고하시기 바랍니다.

구분	실거래가 공개 (RTMS)	부동산거래현황통계 (R-ONE)
활용목적	실거래가 참고자료 제공	부동산거래량 공표
집계기준	계약일 기준	신고일 이후
관리범위	매매거래	매매 이외 판결, 교환, 증여 등
공개기준	실시간 취합 후 익일 공개	매월말기준 취합 공표

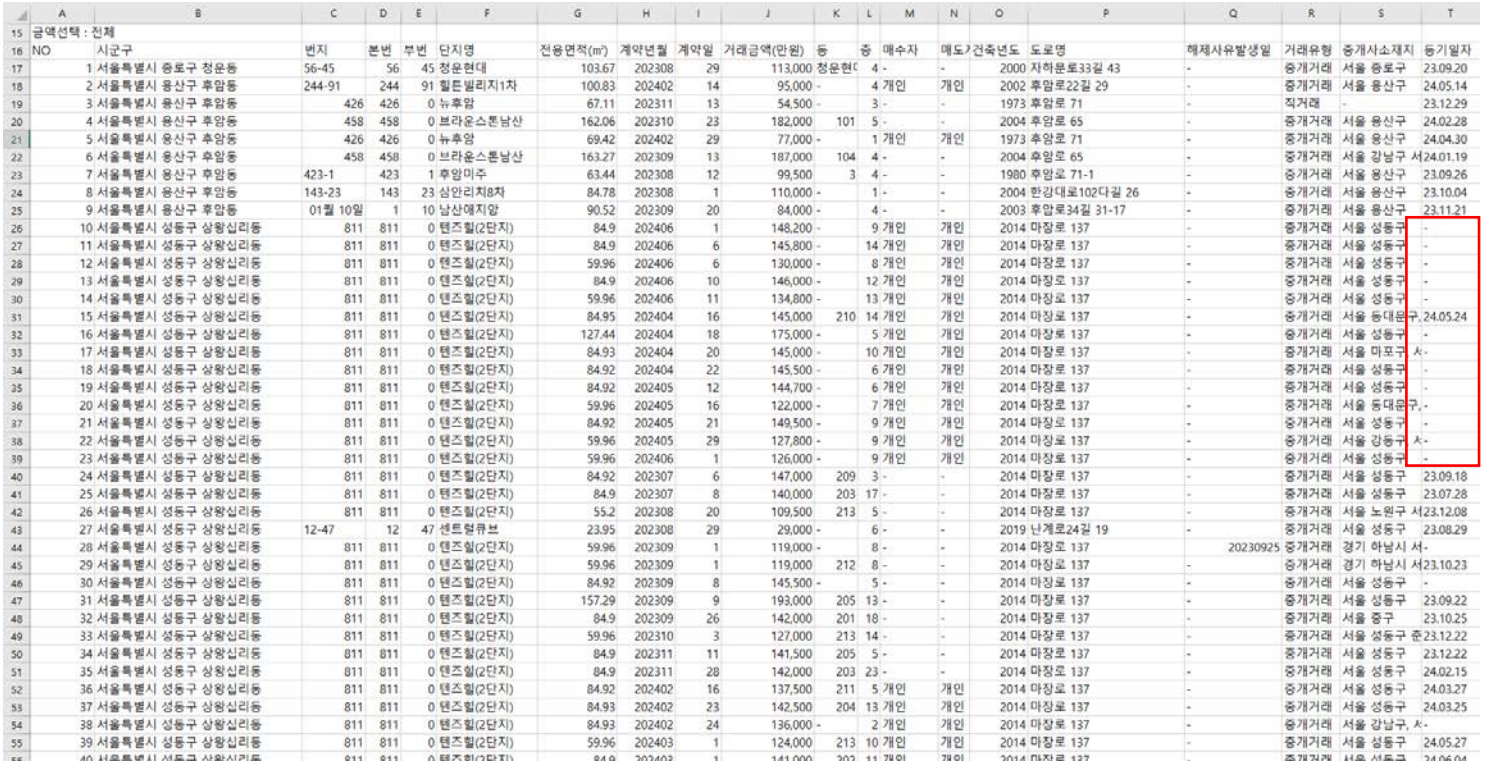
아파트 실거래가는 특정 아파트의 거래가 이루어진 시점, 그리고 그 아파트의 위치, 층수, 전용면적 등 다양한 정보를 포함하고 있습니다.

하지만 이 **실거래가는 부동산 계약일 이후 30일 이내에 신고**하게 돼 있어 소유권 등기 이전을 하지 않고 계약서만 쓴 상태에서 올릴 수 있습니다. 이를 악용해 특정 아파트를 최고가에 허위 거래하고, 인근 단지나 같은 단지에서 최고가에 맞춰 상승 거래가 이뤄지면 기존 거래를 취소하는 방식으로 호가를 띄우는 행위가 벌어지곤 했습니다.

특히 **부동산 소유권 이전은 잔금을 치른 날로부터 60일 이내**에 하도록 돼 있는데, 이번 정책으로 아파트 실거래가 정보에 등기일이 기재 되어 있으면 '진짜 거래'라고 볼 수 있고 그렇지 않다면 허위 신고일 가능성이 커집니다.

실거래가 신고(계약일부터 30일 이내) vs 등기(잔금 치른날로부터 60일 이내)

5.3 자료 전처리(1)- 1차. 미신고건, 계약 해제건 삭제



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
15	금액선택 : 전체																			
16	NO	시군구	번지	본번	부번	단지명	전용면적(㎡)	계약년월	계약일	거래금액(만원)	층		매수자	매도자	건축년도	도로명	해제사유발생일	거래유형	중개사소재지	등기일자
17	1	서울특별시 중로구 청운동	56-45	56	45	청운현대	103.67	202308	29	113,000	청운현대	4	-	-	2000	자하문로33길 43	-	중개거래	서울 중로구	23.09.20
18	2	서울특별시 용산구 후암동	244-91	244	91	힐튼빌리지1차	100.83	202402	14	95,000	-	4	개인	개인	2002	후암로22길 29	-	중개거래	서울 용산구	24.05.14
19	3	서울특별시 용산구 후암동	426	426	0	뉴후암	67.11	202311	13	54,500	-	3	-	-	1973	후암로 71	-	직거래	-	23.12.29
20	4	서울특별시 용산구 후암동	458	458	0	브라운스톤남산	162.06	202310	23	182,000	101	5	-	-	2004	후암로 65	-	중개거래	서울 용산구	24.02.28
21	5	서울특별시 용산구 후암동	426	426	0	뉴후암	69.42	202402	29	77,000	-	1	개인	개인	1973	후암로 71	-	중개거래	서울 용산구	24.04.30
22	6	서울특별시 용산구 후암동	458	458	0	브라운스톤남산	163.27	202309	13	187,000	104	4	-	-	2004	후암로 65	-	중개거래	서울 강남구 서	24.01.19
23	7	서울특별시 용산구 후암동	423-1	423	1	후암미주	63.44	202308	12	99,500	3	4	-	-	1980	후암로 71-1	-	중개거래	서울 용산구	23.09.26
24	8	서울특별시 용산구 후암동	143-23	143	23	삼안리치8차	84.78	202308	1	110,000	-	1	-	-	2004	한강대로102다길 26	-	중개거래	서울 용산구	23.10.04
25	9	서울특별시 용산구 후암동	01월 10일	1	10	남산예지앙	90.52	202309	20	84,000	-	4	-	-	2003	후암로34길 31-17	-	중개거래	서울 용산구	23.11.21
26	10	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.9	202406	1	148,200	-	9	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
27	11	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.9	202406	6	145,800	-	14	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
28	12	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202406	6	130,000	-	8	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
29	13	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.9	202406	10	146,000	-	12	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
30	14	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202406	11	134,800	-	13	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
31	15	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.95	202404	16	145,000	210	14	개인	개인	2014	마장로 137	-	중개거래	서울 동대문구	24.05.24
32	16	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	127.44	202404	18	175,000	-	5	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
33	17	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.93	202404	20	145,000	-	10	개인	개인	2014	마장로 137	-	중개거래	서울 마포구 서	-
34	18	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.92	202404	22	145,500	-	6	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
35	19	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.92	202405	12	144,700	-	6	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
36	20	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202405	16	122,000	-	7	개인	개인	2014	마장로 137	-	중개거래	서울 동대문구	-
37	21	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.92	202405	21	149,500	-	9	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
38	22	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202405	29	127,800	-	9	개인	개인	2014	마장로 137	-	중개거래	서울 강동구 서	-
39	23	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202406	1	126,000	-	9	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	-
40	24	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.92	202307	6	147,000	209	3	-	-	2014	마장로 137	-	중개거래	서울 성동구	23.09.18
41	25	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.9	202307	8	140,000	203	17	-	-	2014	마장로 137	-	중개거래	서울 성동구	23.07.28
42	26	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	55.2	202308	20	109,500	213	5	-	-	2014	마장로 137	-	중개거래	서울 노원구 서	23.12.08
43	27	서울특별시 성동구 상왕십리동	12-47	12	47	센트럴큐브	23.95	202308	29	29,000	-	6	-	-	2019	난계로24길 19	-	중개거래	서울 성동구	23.08.29
44	28	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202309	1	119,000	-	8	-	-	2014	마장로 137	20230925	중개거래	경기 하남시 서	-
45	29	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202309	1	119,000	212	8	-	-	2014	마장로 137	-	중개거래	경기 하남시 서	23.10.23
46	30	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.92	202309	8	145,500	-	5	-	-	2014	마장로 137	-	중개거래	서울 성동구	-
47	31	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	157.29	202309	9	193,000	205	13	-	-	2014	마장로 137	-	중개거래	서울 성동구	23.09.22
48	32	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.9	202309	26	142,000	201	18	-	-	2014	마장로 137	-	중개거래	서울 중구	23.10.25
49	33	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202310	3	127,000	213	14	-	-	2014	마장로 137	-	중개거래	서울 성동구 중	23.12.22
50	34	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.9	202311	11	141,500	205	5	-	-	2014	마장로 137	-	중개거래	서울 성동구	23.12.22
51	35	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.9	202311	28	142,000	203	23	-	-	2014	마장로 137	-	중개거래	서울 성동구	24.02.15
52	36	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.92	202402	16	137,500	211	5	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	24.03.27
53	37	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.93	202402	23	142,500	204	13	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	24.03.25
54	38	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	84.93	202402	24	136,000	-	2	개인	개인	2014	마장로 137	-	중개거래	서울 강남구 서	-
55	39	서울특별시 성동구 상왕십리동	811	811	0	텐즈힐(2단지)	59.96	202403	1	124,000	213	10	개인	개인	2014	마장로 137	-	중개거래	서울 성동구	24.05.27

등기신고(T)가 안 되어있는 것들은 객관성을 위해 제외 or 해제사유(Q)만이라도..

5.3 자료 전처리(1)- 1차. 미신고건, 계약 해제건 삭제

데이터 전처리를 진행하려고 해.
- Q열에 값이 "-"이 아니면 삭제해줘. 예를 들어 20230925 처럼 수치 데이터가 들어간 경우 삭제대상이야.
- T열에 날짜 데이터가 없어서 "-"로 표시되는 경우 삭제해줘.
- 위 두 경우에 하나라도 해당하면 삭제 대상이야.

python 코드 복사

```python
import pandas as pd

# 파일 경로를 지정합니다.
file_path = '/content/apt_price_seoul_20240622091810.csv'

# 처음 15행을 건너뛰고 데이터를 불러옵니다.
df = pd.read_csv(file_path, encoding='cp949', skiprows=15)

# Q열과 T열의 조건을 만족하지 않는 행 삭제
df_cleaned = df[(df['해제사유발'] == '-') & (df['등기일자'] != '-')]

# 데이터프레임을 확인합니다.
print(df_cleaned.head())

# 필요에 따라 결과를 파일로 저장할 수 있습니다.
# df_cleaned.to_csv('/content/cleaned_apt_price_seoul.csv', index=False, encoding='cp949')
```

ChatGPT : 행/열의 “한글이름” +알파벳과 숫자 함께 제시! Ex) 3행, B열

5.3 자료 전처리 (2) – 전처리 후 확인

위에서부터 20줄 까지 보여줄래?

python 코드 복사

```python
import pandas as pd

# 업로드된 파일의 이름을 확인합니다.
file_name = '아파트(매매)_실거래가_20240622091810.csv'

# 처음 15행을 건너뛰고 데이터를 불러옵니다.
df = pd.read_csv(file_name, encoding='cp949', skiprows=15)

# Q열과 T열의 조건을 만족하지 않는 행 삭제
df_cleaned = df[(df['해제사유발'] == '-') & (df['등기일자'] != '-')]

# 위에서부터 20줄을 확인합니다.
df_head_20 = df_cleaned.head(20)

# 데이터프레임을 출력합니다.
df_head_20
```

데이터 전처리 하고나면 꼭 새로운 데이터 프레임을 불러와서 확인하기!

5.3 자료 전처리 (3) – 2차. 시/구/동 나누기

파이썬을 이용해서 뒤쪽에 새로운 열 3개를 추가한 뒤, B열 "시군구" 열을 공백을 기준으로 나누어서 로 저장할거야. 예를 들어 "서울특별시 종로구 청운동" 은 서울특별시 / 종로구 / 청운동 3개 값으로 나누어서 저장해야해. "서울특별시 성동구 상왕십리동"의 경우 서울특별시 /성동구 / 상왕십리동 3개 값으로 나누어 저장할거야. 열제목은 "소재지 시", "소재지 군", "소재지 구"라고 붙이는 코드짜줘.

python 코드 복사

```python
import pandas as pd

# 예시 데이터프레임 생성 (실제 데이터프레임으로 교체할 것)
data = {
    "시군구": ["서울특별시 종로구 청운동", "서울특별시 성동구 상왕십리동"]
}
df_cleaned = pd.DataFrame(data)

# 시군구 열의 데이터를 공백을 기준으로 나누기
split_data = df_cleaned['시군구'].str.split(expand=True)

# 새로운 열 추가
df_cleaned['소재지 시'] = split_data[0]
df_cleaned['소재지 구'] = split_data[1]
df_cleaned['소재지 동'] = split_data[2]

# 결과 확인
print(df_cleaned)
```

필요한 정보가 합쳐져 경우 나눠달라고 요청

5.3 자료 전처리 (4) – 3차. 평행구분 변수 추가

df_cleaned의 "전용면적(㎡)"열을 기준으로 값에 따라 df_cleaned 뒤쪽 열에 "평형구분"이라는 새로운 열을 추가하고 거기에 값을 넣어줘. 이렇게 만들어진 새로운 데이터프레임은 df_claned2라고 저장하는 코드 짜줘.

- 55이하는 "소형평수"
- 55~66까지는 "24평형"
- 66~79까지는 "30평형"
- 79~99까지는 "34평형"
- 99이상은 "대형평수"

```python
import pandas as pd

# 예시 데이터프레임 생성 (실제 데이터프레임으로 교체할 것)
data = {
    "시군구": ["서울특별시 종로구 청운동", "서울특별시 성동구 상왕십리동", "서울특별시 강남구 역
    "전용면적": [85.45, 64.12, 75.30]
}
df_cleaned = pd.DataFrame(data)

# "평형구분" 열 추가
conditions = [
    (df_cleaned['전용면적'] <= 55),
    (df_cleaned['전용면적'] > 55) & (df_cleaned['전용면적'] <= 66),
    (df_cleaned['전용면적'] > 66) & (df_cleaned['전용면적'] <= 79),
    (df_cleaned['전용면적'] > 79) & (df_cleaned['전용면적'] <= 99),
    (df_cleaned['전용면적'] > 99)
]

choices = ['소형평수', '24평형', '30평형', '34평형', '대형평수']

df_cleaned['평형구분'] = pd.cut(df_cleaned['전용면적'],
                            bins=[-float('inf'), 55, 66, 79, 99, float('inf')],
                            labels=['소형평수', '24평형', '30평형', '34평형', '대형평수'])

# 새로운 데이터프레임 저장
df_cleaned2 = df_cleaned.copy()

# 결과 확인
print(df_cleaned2)
```

열이름은 정확해야합니다.
- 전용면적 (X),
- 전용면적(㎡) (O)

전용 59㎡가 25평형 / 74㎡가 30평형 / 84㎡는 34평형 / 114㎡는 43평형

5.3 자료 전처리(5) – 전처리 후 확인(히스토그램)

df_cleaned의 전용면적이 어떻게 분포되어있는지 궁금해. 히스토그램으로 그려줄래?

python 코드 복사

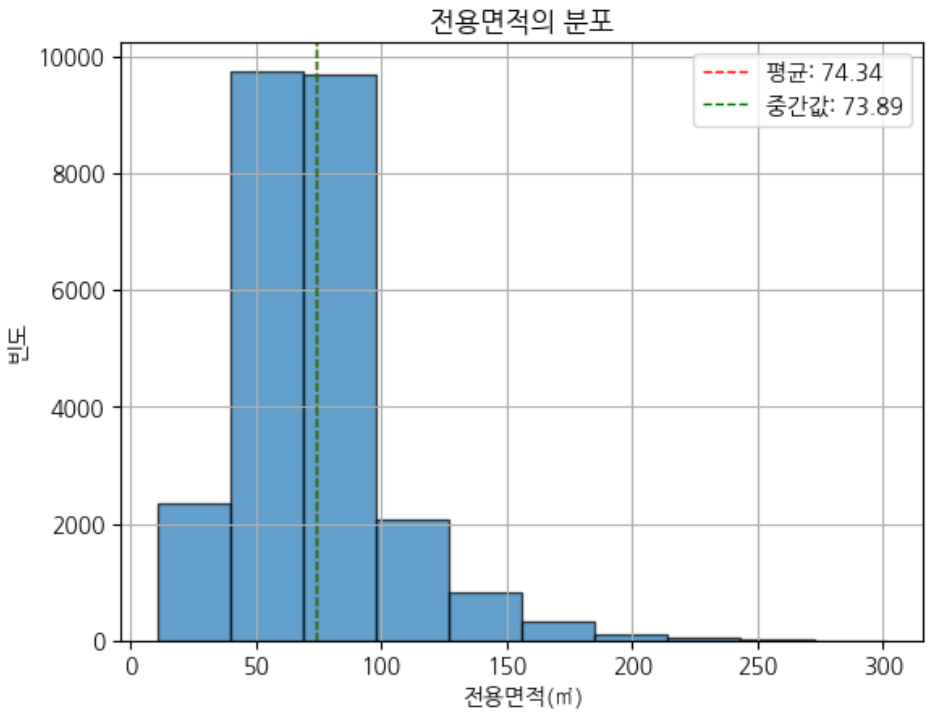
```python
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

# 업로드한 폰트 파일 경로
font_path = '/content/NanumGothic.ttf'  # 실제 업로드된 파일 경로로 변경하세요.

# 폰트 설정
font_prop = fm.FontProperties(fname=font_path)
plt.rc('font', family=font_prop.get_name())

# 예시 데이터프레임 생성 (실제 데이터프레임으로 교체할 것)
data = {
    "시군구": ["서울특별시 종로구 청운동", "서울특별시 성동구 상왕십리동", "서울특별시 강남구
    "전용면적": [85.45, 64.12, 75.30]
}
df_cleaned = pd.DataFrame(data)

# 전용면적의 히스토그램 그리기
plt.figure(figsize=(10, 6))
plt.hist(df_cleaned['전용면적'], bins=10, edgecolor='black')
plt.title('전용면적의 분포')
plt.xlabel('전용면적(㎡)')
plt.ylabel('빈도')
plt.grid(True)
plt.show()
```



ChatGPT(코드생성) ↔ Colab(결과 확인) ↔ ChatGPT(프롬프트 수정)

5.4 그래프화

df_cleaned2 데이터를 가지고 데이터를 그룹으로 분류해서 "거래금액(만원)"을 평균내는 새로운 데이터프레임 df_group을 만들어줘.
그리고 df_group을 가지고 가격추이를 나타내는 그래프를 그릴거야.
X축은 H열 "계약년월"로 두고, Y축은 J열 "거래금액(만원)"의 평균값을 둘거야. 시계열 그래프를 그려줘. 이 때 "평형구분"에 따라 색깔을 다르게 표시해줘.

이 프롬프트는 2단계로 이루어져 있어요
1. 데이터를 “그룹”으로 나누기
2. 그래프 그리기

python 코드 복사

```python
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

# 한글 폰트 설정
font_path = '/mnt/data/NanumGothic.ttf'  # 업로드된 파일 경로로 변경
font_prop = fm.FontProperties(fname=font_path)
plt.rc('font', family=font_prop.get_name())

# 예시 데이터프레임 생성 (실제 데이터프레임으로 교체할 것)
data = {
    "시군구": ["서울특별시 종로구 청운동", "서울특별시 성동구 상왕십리동", "서울특별시 강남구 역
            "서울특별시 서초구 서초동", "서울특별시 송파구 잠실동"],
    "전용면적": [85.45, 64.12, 75.30, 55.00, 99.12],
    "거래금액(만원)": [100000, 85000, 120000, 65000, 200000],
    "계약년월": [202401, 202401, 202402, 202402, 202403]
}

df_cleaned2 = pd.DataFrame(data)

# "평형구분" 열 추가
df_cleaned2['평형구분'] = pd.cut(df_cleaned2['전용면적'],
                              bins=[-float('inf'), 55, 66, 79, 99, float('inf')],
                              labels=['소형평수', '24평형', '30평형', '34평형', '대형평수'])

# 데이터프레임 그룹화
df_group = df_cleaned2.groupby(['계약년월', '평형구분'])['거래금액(만원)'].mean().reset_index

# 결과 확인
print(df_group)
```

내가 원하는 결과물의 그림을 그린 뒤, 그에 맞게 결과를 보면서 코드 수정

5.5 오류 해결

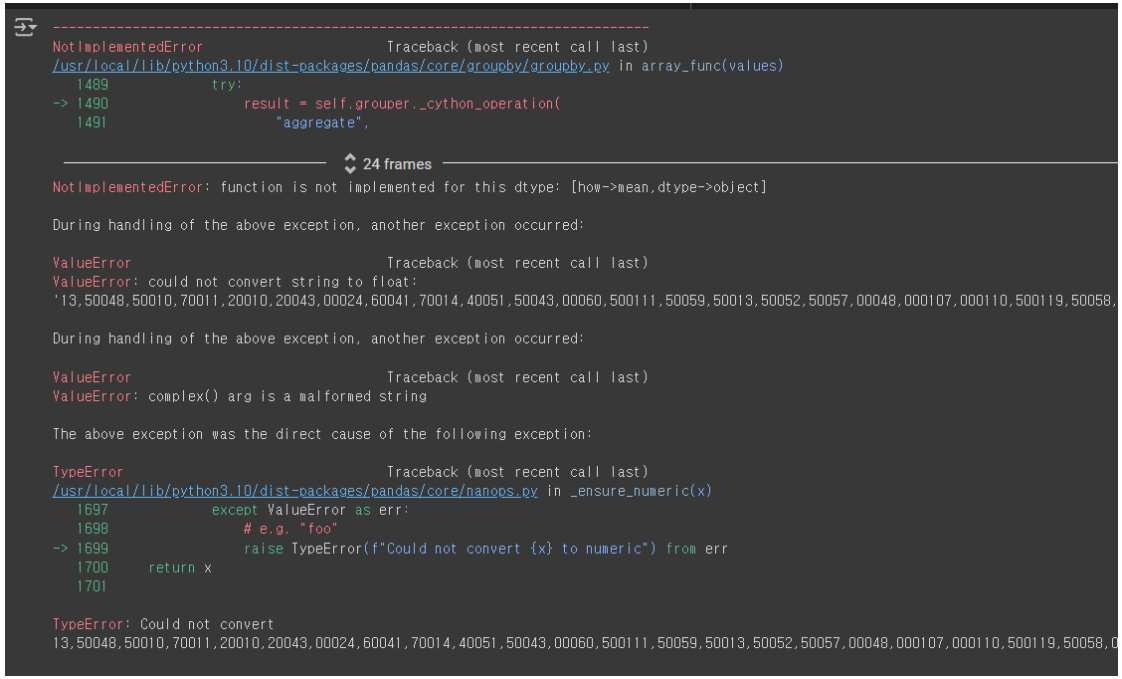
NotImplementedError Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/groupby/groupby.py in array_func(values)
1489 try:
-> 1490 result = self.grouper._cython_operation(
1491 "aggregate",

24 frames
NotImplementedError: function is not implemented for this dtype: [how->mean,dtype->object]

During handling of the above exception, another exception occurred:

ValueError Traceback (most recent call last)
ValueError: could not convert string to float: '13,50048,50010,70011,20010,20043,00024,60041,70014,40051,50043,00060,500111,50059,50013,50052,50057,00048,000107,000110,500119,50058,00013,00014,70015,00085,00023,70014,50049,00010,70081,00035,50033,00041,80033,00036,00029,00028,00027,70020,00047,50045,00049,00051,00049,50...

이런 오류가 났어



오류창을 캡쳐하거나 그대로 복붙해서 ChatGPT한테 물어보세요.

5.5 오류 해결

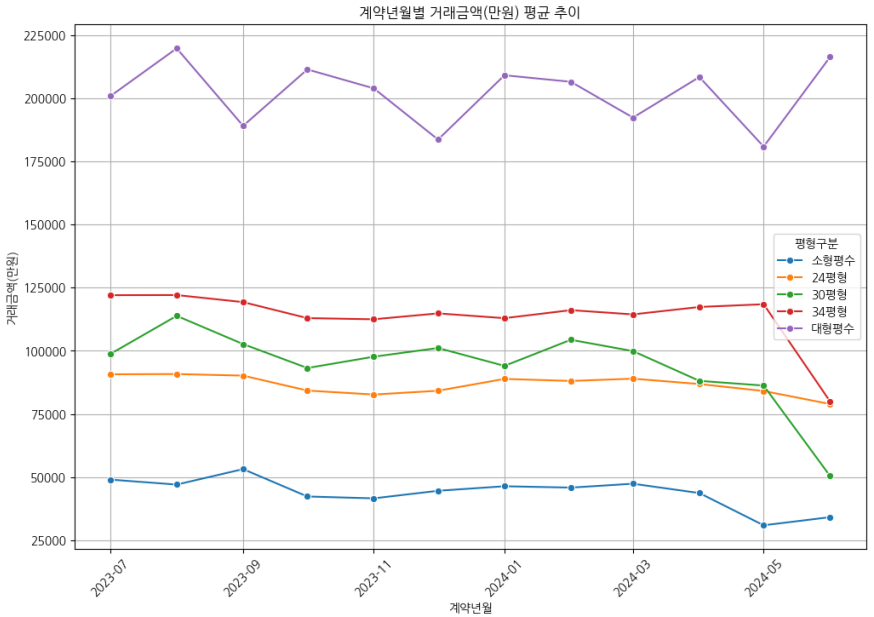
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-20-5d01ce08626f> in <cell line: 2>()
      1 # "거래금액(만원)" 열을 숫자 형식으로 변환
----> 2 df_cleaned2['거래금액(만원)'] = df_cleaned2['거래금액(만
원)'].str.replace(',', '').astype(float)
      3
      4 # "평형구분" 열 추가
      5 df_cleaned2['평형구분'] = pd.cut(df_cleaned2['전용면적(㎡)'],

3 frames
/usr/local/lib/python3.10/dist-
packages/pandas/core/strings/accessor.py in _validate(data)
    233
    234         if inferred_dtype not in allowed_types:
--> 235             raise AttributeError("Can only use .str accessor
with string values!")
    236         return inferred_dtype
    237

AttributeError: Can only use .str accessor with string values!
```

이런 오류가 났어

코드 복붙



경험상 오류가 한번에 해결되는 경우가 많진 않아요. 하지만 해결될때까지... ㅎㅎ

6.1 Outro : 다시 한 번 강조!

- **ChatGPT가 모든 것을 다 해준다?** **No!**
  - 기본적인 언어 문법은 알아야 함. 적어도 검증은 해야하니깐…

- 인간이 해야하는 공부는 어떤 방향? **큰 그림 설계 및 기초!**
  - ex) 어떤 방법론? 어떤 그래프 모습?
  - 머릿속에 결과물을 그리고 이를 말로, 혹은 손으로 표현할 수 있으면 된다.
  - 모범사례 공부를 많이 하자(Few-shot learning)

- 100번 시도해야 1번의 좋은 결과물이 나온다! **무한반복이 답!**
  - AI는 다만 그 속도를 빠르게 해줄뿐, 여러 번의 시도는 무조건 필요하다!
  - 돌려보고, 복붙해서 결과물 보고, 프롬프트 재수정…