부동산과 뉴스의 관계 비교 및 시각화

The Relationship between Real Estate and News: A Comparative Analysis and Visualization

3조 백인범, 서준원, 방다은

목차

주제 선정 이유

1. **부동산**은 통계청(2017)에 따르면 자산의 75%를 차지하는 재산 목록 1호로 **많은 사람이 관심**을 갖는 주제

2. 대다수 사람이 관심을 갖기 때문에 언론에서도 **부동산 뉴스**의 **가치를 높게** 판단 (김수영 · 박승관; 2010, 하현종, 2015)

➡ 부동산을 **미시적인 측면**에서 살펴 보기 위해 **부동산 매매 실거래가, 매매가격지수, 매매 건수, 소비심리지수, 언론 기사** 등을 비교

데이터 수집 및 전처리

1. 데이터 수집

(1) **네이버 부동산 뉴스** 크롤링

1) 선정 이유: 한국언론진흥재단(2017년) 조사에 따르면 인터넷을 통한 뉴스 이용률은 75.6%이며 뉴스 수용자가 일주일 간 가장 많이 접속한 포털은 네이버가 68.4%

2) 검색어: 서울 아파트 매매

3) 선정 대상: 2018년 ~ 2022년 총 26,536건의 부동산 뉴스

(2) **한국부동산원**: 부동산시장 소비심리지수(2018년 ~ 2022년), 월별 아파트 거래(2018년 ~ 2022년)

(3) **국민은행**: 월간 아파트 매매가격지수(2018년 ~ 2022년)

(4) **국토교통부**: 아파트 매매 실거래가(2018년 ~ 2022년)

데이터 수집 및 전처리

1. 데이터 수집

(1) 네이버 부동산 뉴스 크롤링

```
검색어 입력: 서울 아파트 매매
시작 날짜 입력(예시: 2019.01.04): 2020.04.01
종료 날짜 입력(예시: 2019.01.05): 2020.06.30
최대 크롤링할 페이지 수를 입력하세요: 300
100%|██████████| 4550/4550 [00:00<00:00, 1086112.53it/s]
100%|██████████| 1550/1550 [1:07:08<00:00,  2.60s/it]
```

```python
# 크롤링할 url 생성하는 함수 만들기(검색어, 시작 날짜, 종료 날짜, 최대 페이지)
def makeUrl(search, s_date, e_date, maxpage):
    s_from = s_date.replace(".","")
    e_to = e_date.replace(".","")
    url = [f"https://search.naver.com/search.naver?where=news&query={search}&sort=0&ds={s_date}&de={e_date}&nso=so%3Ar%2Cp%3Afrom{s_from}to{e_to}%2Ca%3A&start={i}" for i in range(1, int(maxpage) * 10, 10)]
    return url

# html에서 원하는 속성 추출하는 함수 만들기(기사, 추출하려는 속성값)
def news_attrs_crawler(articles, attrs):
    attrs_content = [i.attrs[attrs] for i in articles]
    return attrs_content

# ConnectionError방지
headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/98.0.4758.102"}

#html 생성해서 기사 크롤링하는 함수 만들기(url): 링크를 반환
def articles_crawler(url):
    #html 불러오기
    original_html = requests.get(i,headers=headers)
    html = BeautifulSoup(original_html.text, "html.parser")

    url_naver = html.select("div.group_news > ul.list_news > li div.news_area > div.news_info > div.info_group > a.info")
    url = news_attrs_crawler(url_naver,'href')
    return url
```

데이터 수집 및 전처리

1. 데이터 수집

(1) 네이버 부동산 뉴스 크롤링

데이터 수집 및 전처리

1. 데이터 수집

(1) 네이버 부동산 뉴스 크롤링

	date	name	title	content
960	2021-10-03	연합뉴스	2030 서울아파트 매매 절반 이상	(서울=연합뉴스) 홍해인 기자 = 정부의 부동산 규제 정책 등으로 올해 아파...
1305	2021-10-04	매일경제	[단독] "세금 폭탄에 매물 쏟아진다더니"...1년새 서울 경기 아파트 2만채 사라졌다	작년 7·10이후 다주택자 매매 현황 분석서울 아파트 매도건수 37% 목	종과로 ...
991	2021-10-05	뉴스1	서울 아파트 평균 매매가 12억 육박	(서울=뉴스1) 오대일 기자 = 서울의 평균 아파트값이 올해 들어서만 1억 5천만 ...
1297	2021-10-05	매일경제	'이젠 화도 안난다"...서울 아파트값 평균 12억 육박, 올해만 1.5억 올랐다	강북권 9억5000만원, 강남권 14억원 돌파경기 전일比 2292만원↑ 인천 4억...
990	2021-10-05	뉴스1	서울 아파트 평균 매매가격 12억원 돌파 눈앞	(서울=뉴스1) 오대일 기자 = 서울의 평균 아파트값이 올해 들어서만 1억 5천만 ...
937	2021-10-05	머니투데이	언제 내릴까..서울 아파트 평균 매매가격 12억원 돌파 눈앞	서울 용산구 남산N서울타워에서 바라본 도심 아파트단지. /사진제공=뉴스1서울 아파트...
1298	2021-10-05	중앙일보	강북 아파트도 절반이 9억 넘겼다..서울 집값 상승률 최고치 경신	KB국민은행 9월 주택가격 동향강북 아파트 중위가격 9억500만원 목서울중의술에...
1304	2021-10-05	SBS	서울 평균 아파트값, 12억 원 육박..올해 1억5천만 원 이상 ↑	서울의 평균 아파트값이 올해 들어서만 1억5천만 원 넘게 오르며 약 12억 원에 육...
1294	2021-10-06	국민일보	강북 중위 아파트 9억 넘겨... 올해만 서울 평균 1억5000만원 급등	서울 아파트값 평균 12억원 '눈앞' 강북 패닉바잉으로 실수요자 돌려 강남권 아...
1306	2021-10-06	세계일보	서울 평균 아파트값 12억 육박	올해 들어 1억5000만원 넘게 올라 뉴스1 서울 평균 아파트 매매가격이...

```
# 뉴스 내용 크롤링
for i in tqdm(final_urls):
    #각 기사 html get하기
    news = requests.get(i,headers=headers)
    news_html = BeautifulSoup(news.text,"html.parser")

    time.sleep(2)

    # 언론사명 가져 오기
    names = news_html.select('#contents > div.copyright > div > p')
    if names:
      name = names[0].string[12:-38]
    elif names != names:
      time.sleep(2)
      names = news_html.select('#content > div.end_ct > div > div.copyright > div > p')
      name = names[0].string[12:-38]
    else:
      name = ''

    # 뉴스 제목 가져 오기
    title = news_html.select_one("#ct > div.media_end_head.go_trans > div.media_end_head_title > h2")
    if title == None:
        title = news_html.select_one("#content > div.end_ct > div > h2")

    # 뉴스 본문 가져 오기
    content = news_html.select("div#dic_area")
    if content == []:
        content = news_html.select("#articeBody")

    # 기사 텍스트만 가져 오기:  list합치기
    content = ''.join(str(content))
```

데이터 수집 및 전처리

1. 데이터 수집

(1) 네이버 부동산 뉴스 크롤링

```
# html 태그 제거 및 텍스트 다듬기
pattern1 = '<[^>]*>' # 태그 제거
pattern2 = r'[₩n₩t<>]|&lt;|&gt;' # 특수 기호 제거

title = re.sub(pattern=pattern1, repl='', string=str(title))
title = re.sub(pattern=pattern2, repl='', string=str(title))

content = re.sub(pattern=pattern1, repl='', string=content)
pattern3 = """[₩n₩n₩n₩n₩n// flash 오류를 우회하기 위한 함수 추가₩nfunction _flash_removeCallback() {}"""
content = content.replace(pattern3, '')
content = re.sub(pattern2, '', content)[1:-1] # 기사 본문 맨 앞, 맨 뒤 [ ] 제거

news_names.append(name)
news_titles.append(title)
news_contents.append(content)
```

데이터 수집 및 전처리

1. 데이터 수집

(1) 네이버 부동산 뉴스 크롤링

```
###데이터 프레임으로 만들기###
import pandas as pd

#데이터 프레임 만들기
news_df = pd.DataFrame({'date':news_dates, 'name': news_names, 'title':news_titles, 'content':news_contents, 'link': final_urls})

#중복 행 지우기
news_df = news_df.drop_duplicates(keep='first', ignore_index=True)

#데이터 프레임을 엑셀 파일로 저장
outputFileName = f'{s_date} ~ {e_date}.xlsx'
news_df.to_excel(outputFileName, sheet_name='sheet1')
```

데이터 수집 및 전처리

2. 데이터 전처리

(1) 네이버 부동산 뉴스 결측치 처리

```
[ ]  news_df[news_df['name'] == '']
```

	date	name	title	content	link
612	2021-12-14		20억 아파트도 못 피했네...김새른 충간 소음 고백	김새른, 성동구 위치한 아파트 거주매매가 최저 17억 원...최대 23억"같은 건물 사...	https://n.news.naver.com/mnews/article/015/000...
997	2021-11-17		김의철 KBS 사장 후보, 위장전입해 아파트 분양..."깊이 반성"	4억짜리 아파트 '다운계약서'...취등록세 1천400만원 적게 내 김의철 KBS 사장 ...	https://n.news.naver.com/mnews/article/001/001...
1122	None		None		https://sports.news.naver.com/news.nhn?oid=529...

```
[ ]  news_df[news_df['name'] == '']['link'].values
```

```
array(['https://n.news.naver.com/mnews/article/015/0004640235?sid=106',
       'https://n.news.naver.com/mnews/article/001/0012800282?sid=106',
       'https://sports.news.naver.com/news.nhn?oid=529&aid=0000061491',
       'https://n.news.naver.com/mnews/article/422/0000509250?sid=101'],
      dtype=object)
```

데이터 수집 및 전처리

2. 데이터 전처리

(2) 단어 빈도수 특수기호 제거

	word	freq
0	ㄱㅓ	1
1	ㅅㅅ	1
2	ㅇㅣ	1
3	ㅎㅎ	5
4	ㅎㅎㅎ	2
...	...	...
36429	힘써야	3
36430	힘쓰고	2
36431	힘주어	1
36432	힘찬	3
36433	힙입어	1

36434 rows × 2 columns

```python
import pandas as pd
from konlpy.tag import Okt
from tqdm import tqdm

# okt 객체 생성
okt = Okt()

# content 열에서 Null 값 제거
news_article = df_2020.dropna(subset=['content'])

# content 열의 모든 문장에 대해 형태소 분석을 수행하고, 그 결과를 'morphs' 열에 저장
news_article['morphs'] = news_article['content'].apply(lambda x: okt.morphs(str(x)))

# 단어 집합(vocabulary) 생성
vocabulary = {}
for i in tqdm(news_article['morphs']):
    for j in i:
        if j not in vocabulary:
            vocabulary[j] = 0
        vocabulary[j] += 1

# 빈도수가 높은 순으로 정렬된 단어 리스트 생성
word_list = sorted(vocabulary.items(), key=lambda x: x[1], reverse=True)

# 결과를 엑셀 파일로 저장(단어 포함)
result_df = pd.DataFrame(word_list, columns=['word', 'freq'])
result_df.to_excel('./2020년(단어 포함) 본문 단어 빈도수.xlsx', index=False)


# 결과를 엑셀 파일로 저장(단어 미포함)
word_list = sorted([(k, v) for k, v in vocabulary.items() if k not in ['서울', '아파트', '매매']], key=lambda x: x[1], reverse=True)
result_df = pd.DataFrame(word_list, columns=['word', 'freq'])
result_df.to_excel('./2020년(단어 미포함) 본문 단어 빈도수.xlsx', index=False)
```

데이터 수집 및 전처리

2. 데이터 전처리

(2) 단어 빈도수 특수기호 제거

```
import re

# 문자열로 변환
df_2020['word'] = df_2020['word'].astype(str)

# 정규식을 이용하여 특수기호 제거 및 2글자 이상인 단어만 추출
df_2020['word'] = df_2020['word'].apply(lambda x: re.sub('[^ㄱ-ㅣ가-힣]', '',x))  # 특수기호 제거
df_2020 = df_2020[df_2020['word'].str.len() >= 2]  # 2글자 이상인 단어만 추출
```

데이터 수집 및 전처리

2. 데이터 전처리

(3) 월별 자료를 분기별로 변경

```
change = []
for i in range(len(df_people['date'])):
  if df_people['date'][i][-2:] == '01':
    change.append(df_people['date'][i][:4] + ' 1분기')
  elif df_people['date'][i][-2:] == '04':
    change.append(df_people['date'][i][:4] + ' 2분기')
  elif df_people['date'][i][-2:] == '07':
    change.append(df_people['date'][i][:4] + ' 3분기')
  else:
    change.append(df_people['date'][i][:4] + ' 4분기')
```

```
df_avg = df_new.groupby(np.arange(len(df_new)) // 3).apply(lambda x: x.mean())
df_avg
```

데이터 수집 및 전처리

2. 데이터 전처리

(4) 지수 단위 통일

```
[ ]  from sklearn.preprocessing import MinMaxScaler

     # 가격지수와 부동산심리지수를 스케일링
     scaler = MinMaxScaler()
     df_final[['mean', 'avg']] = scaler.fit_transform(df_final[['mean', 'avg']])
```

```
[ ]  fig, ax1 = plt.subplots(figsize=(10, 5))

     # 왼쪽 y축 설정
     ax1.set_ylabel('거래량')
     ax1.plot(df_final['date'], df_final['서울'], label='거래량', color='blue')

     # 오른쪽 y축 설정
     ax2 = ax1.twinx()
     ax2.set_ylabel('부동산심리지수 / 가격지수')
     ax2.plot(df_final['date'], df_final['mean'], label='부동산심리지수', color='red')
     ax2.plot(df_final['date'], df_final['avg'], label='가격지수', color='green')
     ax2.set_ylim([-0.05, 1.5])

     # 그래프 타이틀 및 라벨 설정
     ax1.set_xlabel('기간')
     ax1.set_xticklabels(df_final['date'], rotation=90)
     plt.title('2018 ~ 2022 서울 분기별 아파트 매매 거래량과 부동산심리지수 / 가격지수', fontsize=15)

     # 범례 추가
     lines1, labels1 = ax1.get_legend_handles_labels()
     lines2, labels2 = ax2.get_legend_handles_labels()
     plt.legend(lines1 + lines2, labels1 + labels2, loc='upper right')

     plt.show()
```

데이터 수집 및 전처리

2. 데이터 전처리

(5) 분기별 상승/하락 단어 빈도수 추출

```
import pandas as pd

# 데이터프레임 불러오기
df = pd.read_excel('/content/drive.xlsx')

# content 열을 문자열로 변환하여 하나의 문자열로 합침
content_str = ' '.join(df['content'].astype(str).tolist())

# 2018년 1분기, 2분기, 3분기, 4분기로 분할
quarters = ['2018년 1분기', '2018년 2분기', '2018년 3분기', '2018년 4분기']
quarter_contents = [content_str[:len(content_str)//4],
                    content_str[len(content_str)//4:len(content_str)//2],
                    content_str[len(content_str)//2:3*len(content_str)//4],
                    content_str[3*len(content_str)//4:]]

# 분기별 상승/하락 단어 빈도수 계산
up_counts = []
down_counts = []
for content in quarter_contents:
    up_count = content.count('상승')
    down_count = content.count('하락')
    up_counts.append(up_count)
    down_counts.append(down_count)

# 결과 데이터프레임 생성
result_df = pd.DataFrame({'분기': quarters, '상승 단어 빈도수': up_counts, '하락 단어 빈도수': down_counts})
result_df_1 = pd.DataFrame({'분기': [quarters[0]], '상승 단어 빈도수': [up_counts[0]], '하락 단어 빈도수': [down_counts[0]]})
result_df_2 = pd.DataFrame({'분기': [quarters[1]], '상승 단어 빈도수': [up_counts[1]], '하락 단어 빈도수': [down_counts[1]]})
result_df_3 = pd.DataFrame({'분기': [quarters[2]], '상승 단어 빈도수': [up_counts[2]], '하락 단어 빈도수': [down_counts[2]]})
result_df_4 = pd.DataFrame({'분기': [quarters[3]], '상승 단어 빈도수': [up_counts[3]], '하락 단어 빈도수': [down_counts[3]]})
```

데이터 수집 및 전처리

2. 데이터 전처리

(6) 계약 건별 자료를 구별 평균 가격으로 변경

```
[ ]  df_all.head()
```

	시군구	번지	본번	부번	단지명	전용면적(㎡)	계약년월	계약일	거래금액(만원)	층	건축년도	도로명
0	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성아파트1동~8동	79.97	201801	3	130,000	4	1987.0	언주로 3
1	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성아파트1동~8동	79.97	201801	8	117,000	2	1987.0	언주로 3
2	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성아파트1동~8동	79.97	201801	11	130,000	1	1987.0	언주로 3
3	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성아파트1동~8동	79.97	201803	19	139,500	2	1987.0	언주로 3
4	서울특별시 강남구 개포동	658-1	658.0	1.0	개포6차우성아파트1동~8동	54.98	201804	5	107,500	5	1987.0	언주로 3

데이터 수집 및 전처리

2. 데이터 전처리

(6) 계약 건별 자료를 구별 평균 가격으로 변경

```
df_all['거래금액(만원)'] = df_all['거래금액(만원)'].str.replace(',', '').astype(int)
df_all['구'] = df_all['시군구'].str.split(' ', expand=True).iloc[:,1]
df_all = df_all.reindex(columns=['시군구', '구', '번지', '본번', '부번', '단지명', '전용면적(㎡)', '계약년월', '계약일', '거래금액(만원)', '층', '건축년도', '도로명'])
df_all.head()
```

	시군구	구	번지	본번	부번	단지명	전용면적(㎡)	계약년월	계약일	거래금액(만원)	층	건축년도	도로명
0	서울특별시 강남구 개포동	강남구	658-1	658.0	1.0	개포6차우성아파트1동~8동	79.97	201801	3	130000	4	1987.0	언주로 3
1	서울특별시 강남구 개포동	강남구	658-1	658.0	1.0	개포6차우성아파트1동~8동	79.97	201801	8	117000	2	1987.0	언주로 3
2	서울특별시 강남구 개포동	강남구	658-1	658.0	1.0	개포6차우성아파트1동~8동	79.97	201801	11	130000	1	1987.0	언주로 3
3	서울특별시 강남구 개포동	강남구	658-1	658.0	1.0	개포6차우성아파트1동~8동	79.97	201803	19	139500	2	1987.0	언주로 3
4	서울특별시 강남구 개포동	강남구	658-1	658.0	1.0	개포6차우성아파트1동~8동	54.98	201804	5	107500	5	1987.0	언주로 3

데이터 수집 및 전처리

2. 데이터 전처리

(6) 계약 건별 자료를 구별 평균 가격으로 변경

```
df_all_gu = df_all.groupby(['구'])[['거래금액(만원)']].mean().sort_values(by = '거래금액(만원)', ascending=False)
df_all_gu
```

구	거래금액(만원)
강남구	180309.027184
서초구	172922.686611
용산구	156453.462071
송파구	122557.564387
성동구	105932.772399
광진구	96971.119095
마포구	94862.886815
동작구	88679.864332
양천구	83427.102419
영등포구	81808.770264
중구	80939.406718
강동구	77500.786222
종로구	76497.222222
서대문구	71700.853720
강서구	63771.402713
동대문구	63347.805668
성북구	62947.259835
은평구	57995.976209
관악구	57352.718830
구로구	51680.573890
강북구	51317.930800
노원구	48837.548162
중랑구	48316.206903
금천구	45735.859601
도봉구	43850.532535

데이터 시각화

1. 기사 분석

2018년

2019년

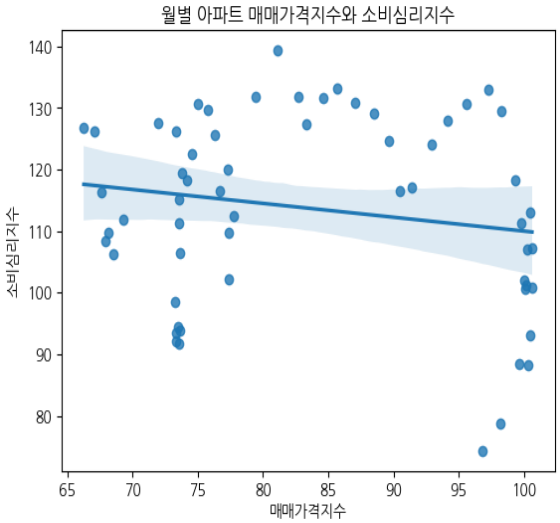
2020년

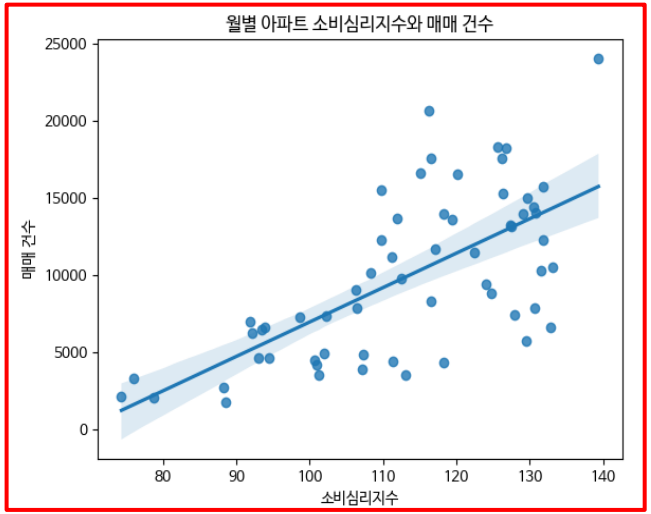
2021년

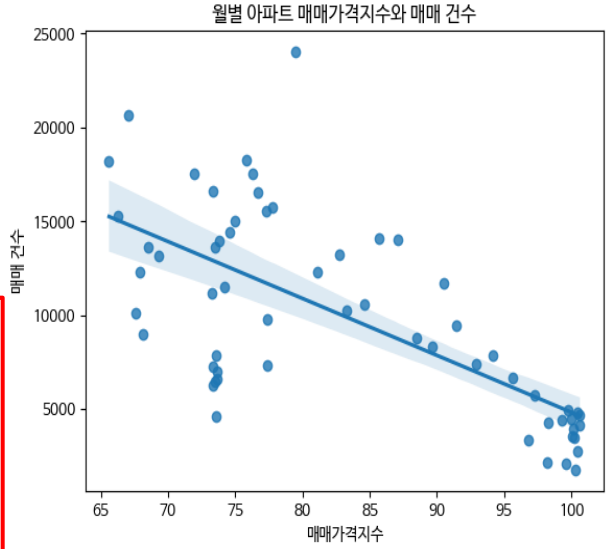
2022년

데이터 시각화

2. 지수 간 연관성

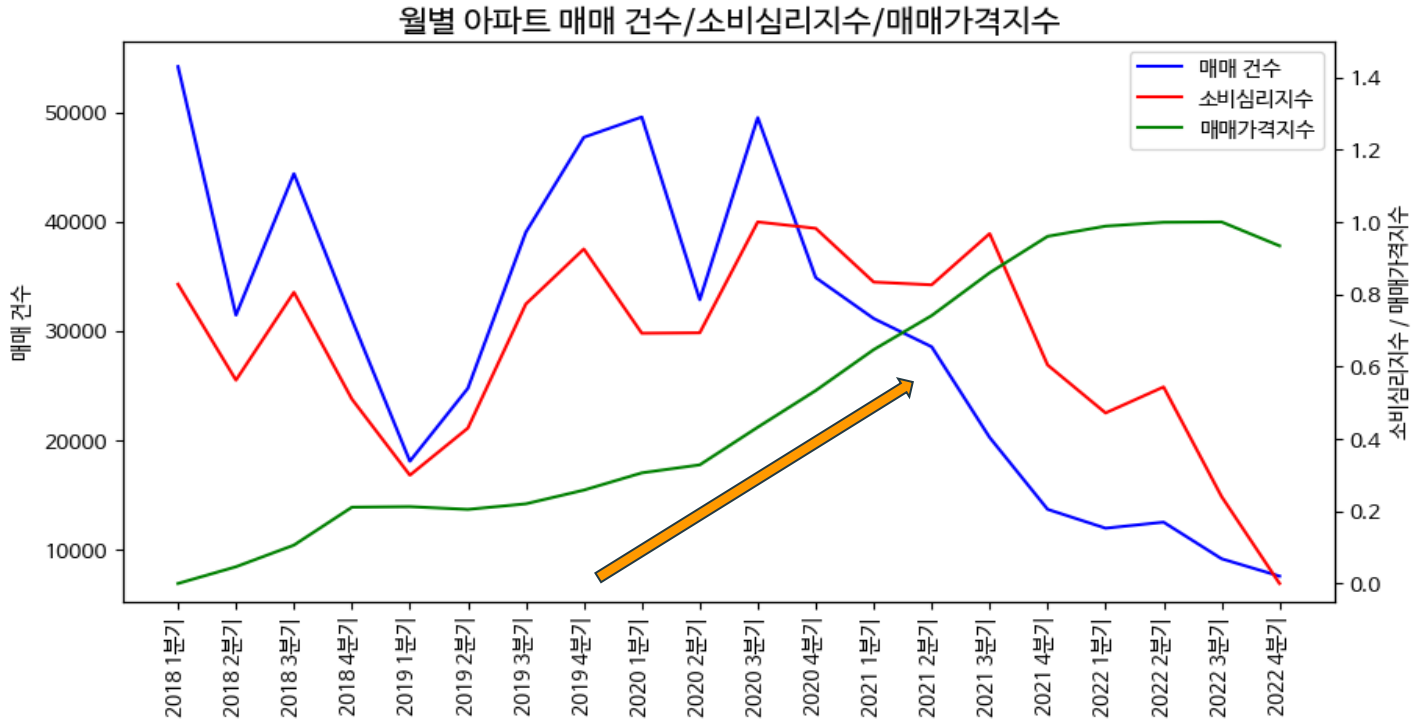






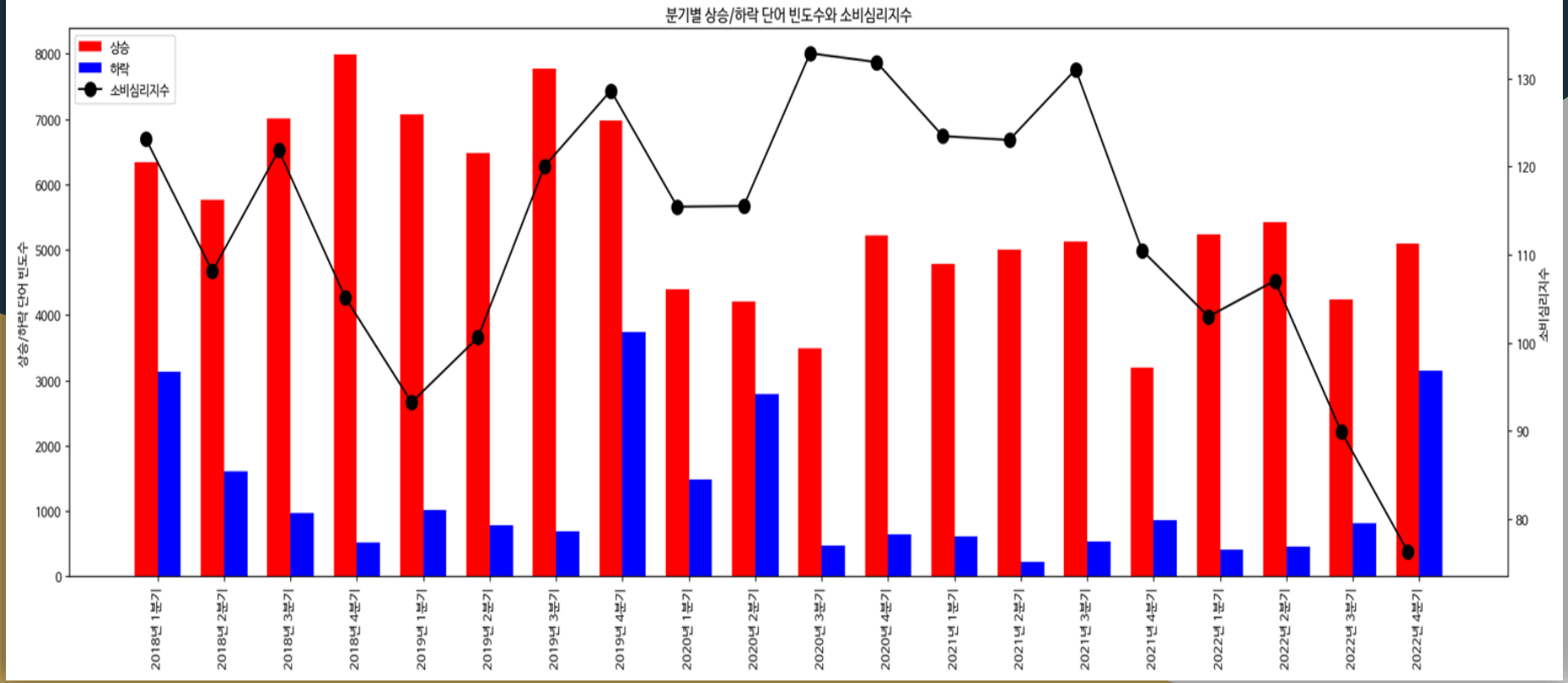
데이터 시각화

2. 지수 간 연관성



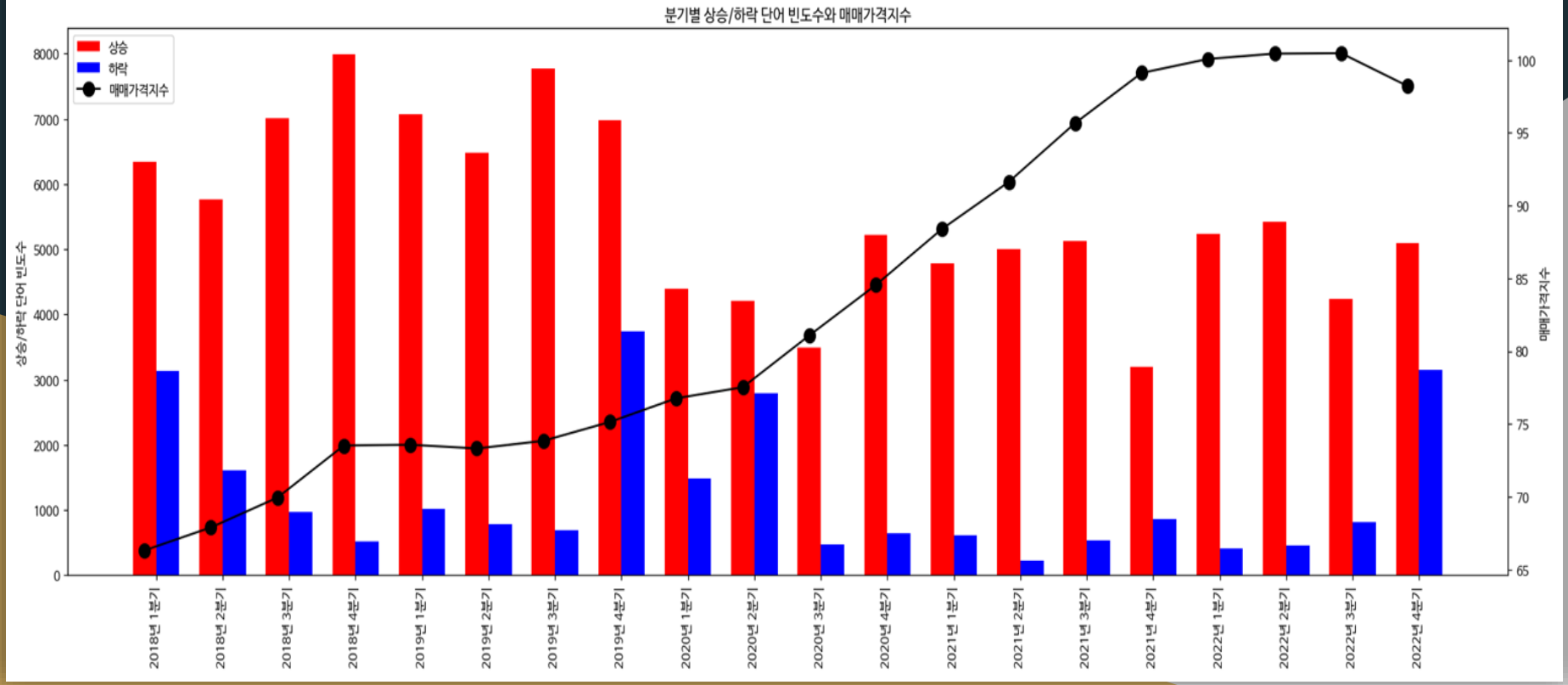
데이터 시각화

3. 지수와 기사의 연관성



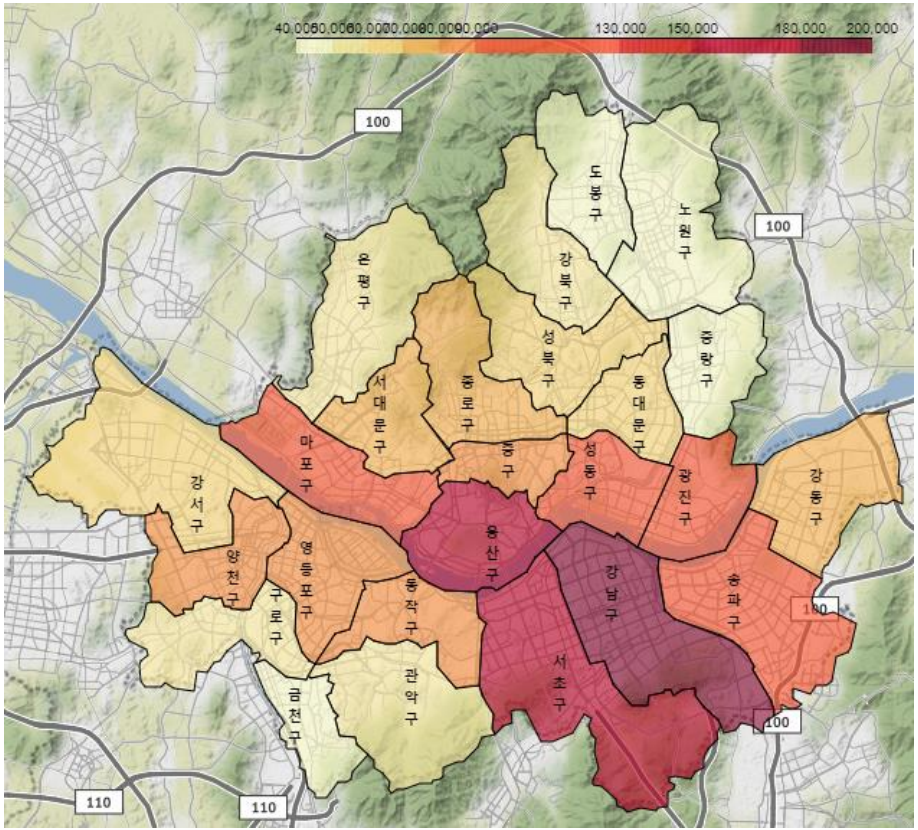
데이터 시각화

3. 지수와 기사의 연관성



데이터 시각화

4. 구별 평균 가격



결론

1. 기사는 **2년**마다 사용하는 **단어 빈도수**에 변화를 보임
2. **월별 아파트 매매 건수**와 **소비심리지수**는 **연관성이 매우 높지만** 매매가격지수와는 연관성 낮음
3. **매매가격지수**는 조사 대상 기간 동안 거의 **내내 상승**
4. 상승, 하락 단어 빈도수와 매매가격지수, 소비심리지수 사이에 연관성 낮음
5. **실거래가 평균**은 **강남구**를 중심으로 멀어질수록 낮아짐

소감

1. 데이터를 **시각화**하니 **연관성**을 파악하기 쉬움
2. **크롤링**이 너무 **오래** 걸림
3. 예상과 달리 **연관성이 없는 경우가** 존재
4. 부동산 가격 **상승, 하락**에 **영향**을 주는 요소 찾기
5. 부동산 **가격 예측**까지 추가

참고문헌

1. 경제뉴스와 부동산 시장의 관계에 관한 빅데이터 시계열 분석(김선우, 2018)

2. 2016년 국민대차대조표(잠정)(통계청, 2017)

3. 방송 경제위기 뉴스의 정치 의미화 과정에 관한 연구,『한국언론학보』, 54(5), 301-32(김수영 · 박승관, 2010)

4. [3 부동산] 뉴스의 지나친 '부동산 사랑',『방송기자』, 24, 14-1(하현종, 2015)

질의응답

감사합니다