

© 집 사러 가영

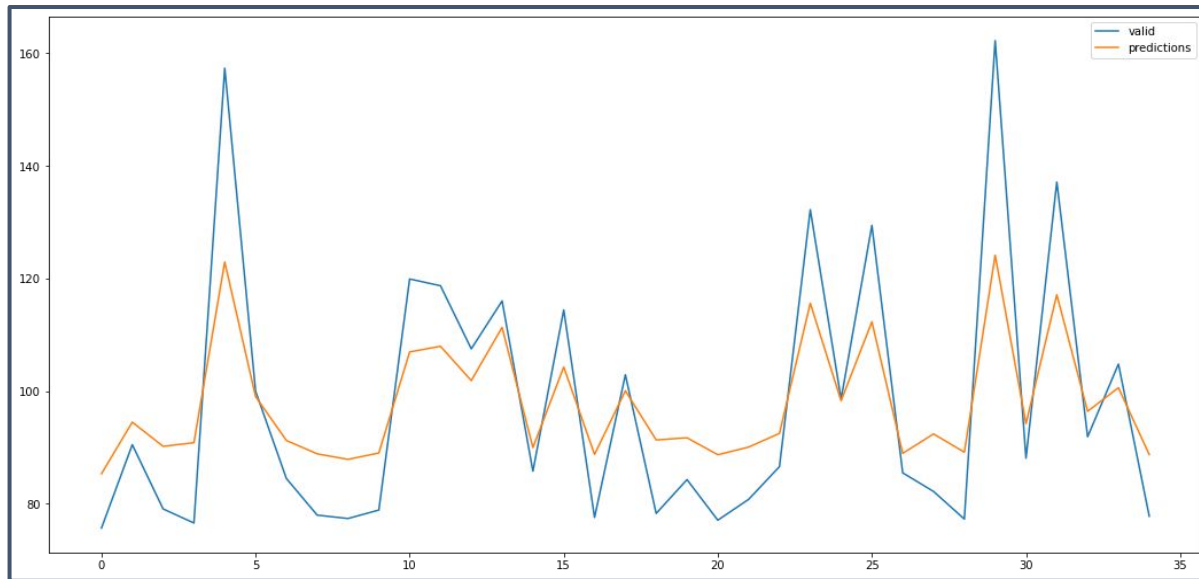
# 한국 부동산 데이터 시각화 경진대회

김가영, 도준희, 차은혜, 태형배

# 주제

## 세부 주제

데이콘에서 제공된 데이터와 외부데이터를 활용하여 **서울** 부동산 실거래가 예상



# DATA

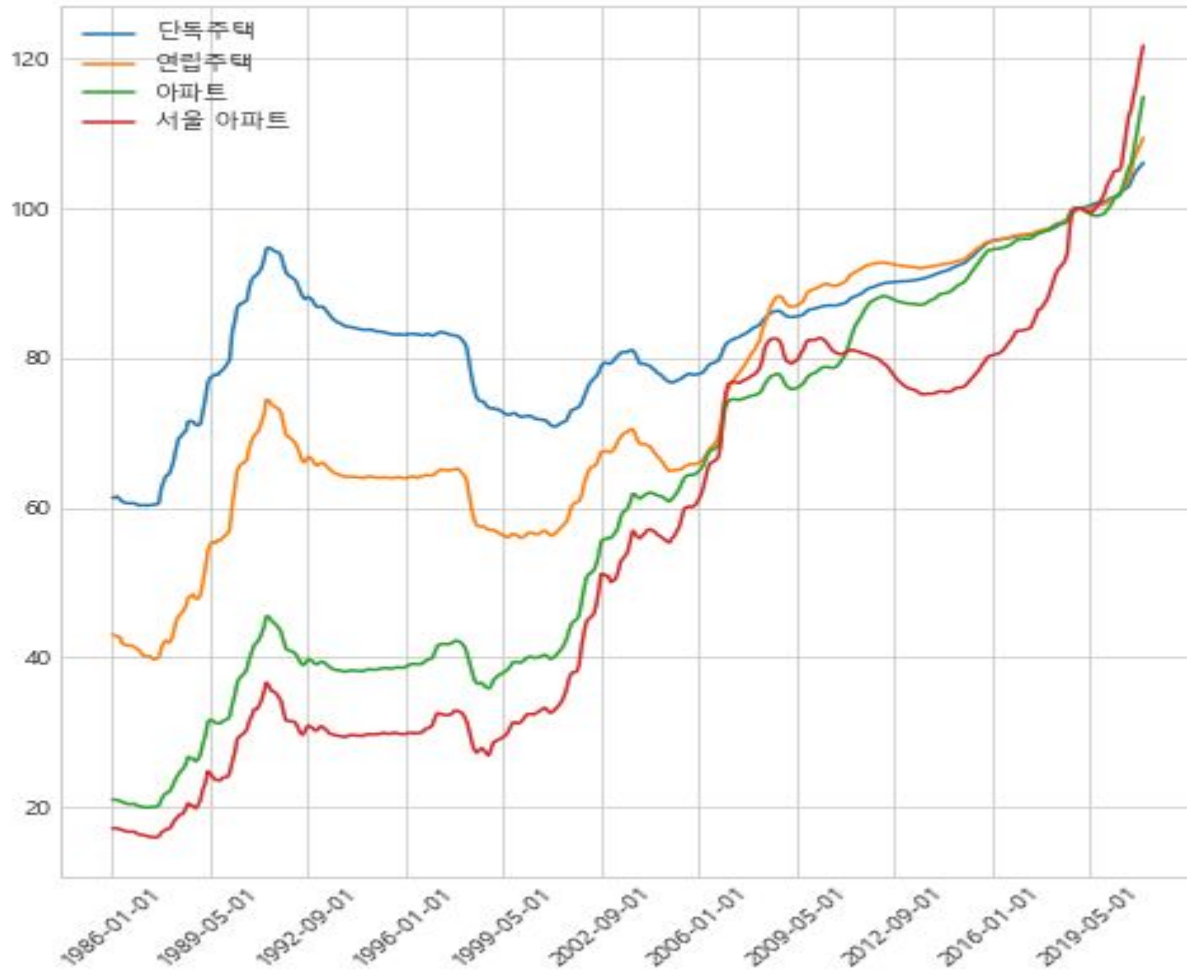
## 외부데이터

주거구분별(1)	주거구분별(2)	주거구분별(3)	2012. 01	2012. 02	2012. 03	2012. 04	2012. 05
연월일	2008년	1월	16,510,434			60,848	
등록면적	200	시점	항목	전국		서울특별시	
평균	200	2000. 01	계 (명)		61644	12866	
승가	200	2000. 01	남자 (명)		32317	6738	
승가	200	2000. 01	여자 (명)		29327	6128	
승가	200	2000. 02	계 (명)		56723	11797	
승가	200	2000. 02	남자 (명)		29693	6227	
승가	200	2000. 02	여자 (명)		27030	5570	
승가	200	2000. 03	계 (명)		59878	12453	
승가	200	2000. 03	남자 (명)		31432	6452	
승가	200	2000. 03	여자 (명)		28446	6001	
승가	200	2000. 04	계 (명)		53058	10782	

서울 DATA

- 코스피 지수
- 아파트거래량
- 출생아 수
- 자동차 등록대수

# 주택매매가격지수(KB)

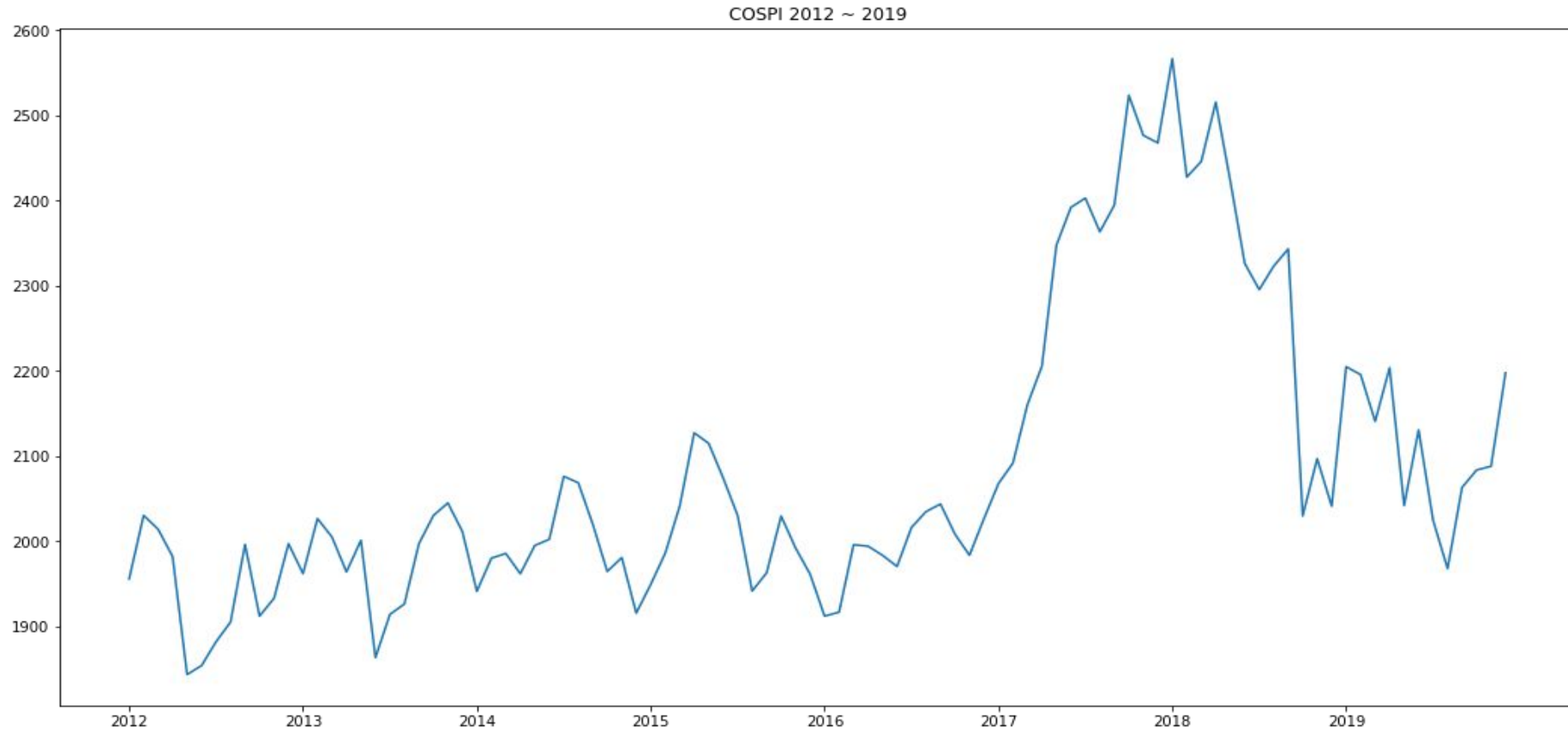


## 주택 유형별 매매 가격 비교

- 서울의 아파트값이 서울의 총 지수와 비슷한 형태를 띄고 있다.
- 아파트값이 단독주택, 연립주택에 높은 것을 확인할 수 있다.
- 2012년-2014년 사이, 서울 아파트의 주택거래량이 줄어든 데다가 정부의 가계부채 후속대책, 시장금리 인상 등이 복합적으로 작용했기 때문에 일시적으로 그래프가 감소하는 것으로 사료됨.

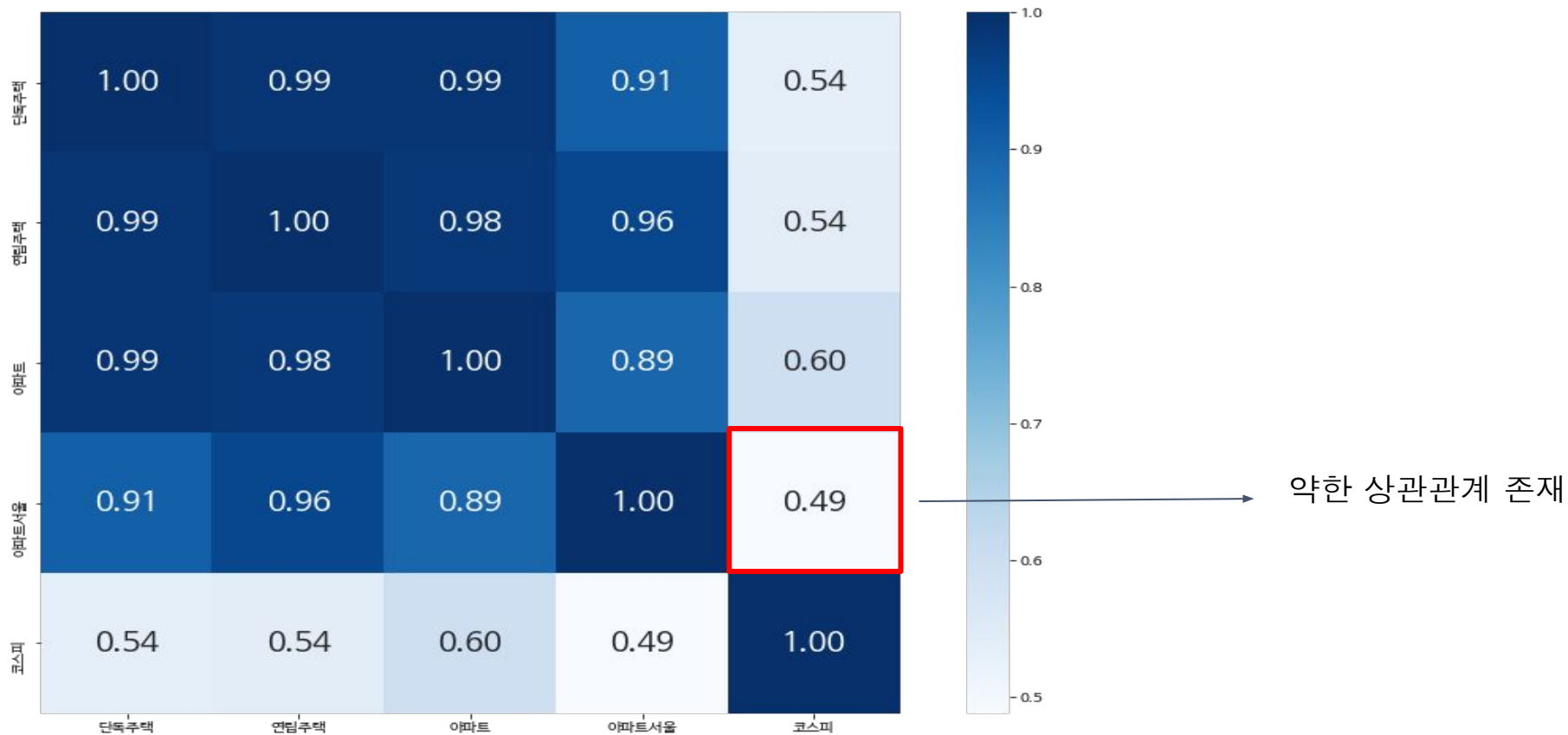
# 코스피 지수

# 코스피 지수



- 코스피 지수는 2012~2019년까지의 데이터를 사용
- 2017에 급등 했다가 2018에 급락하는 추세를 보이고 있다. 2018년에 경제 대공황의 영향을 받았기 때문에 이러한 추세를 보였다고 추측됨

# 코스피 지수











경제 지표인 코스피와 서울의 아파트 가격은 뚜렷한 상관관계를 보이지 않았다. 변수로 사용하기 어려울 것으로 보임

# 아파트 거래량



# 아파트 거래량

DATA

	아파트거래내역_2012
	아파트거래내역_2013
	아파트거래내역_2014
	아파트거래내역_2015
	아파트거래내역_2016
	아파트거래내역_2017
	아파트거래내역_2018
	아파트거래내역_2019

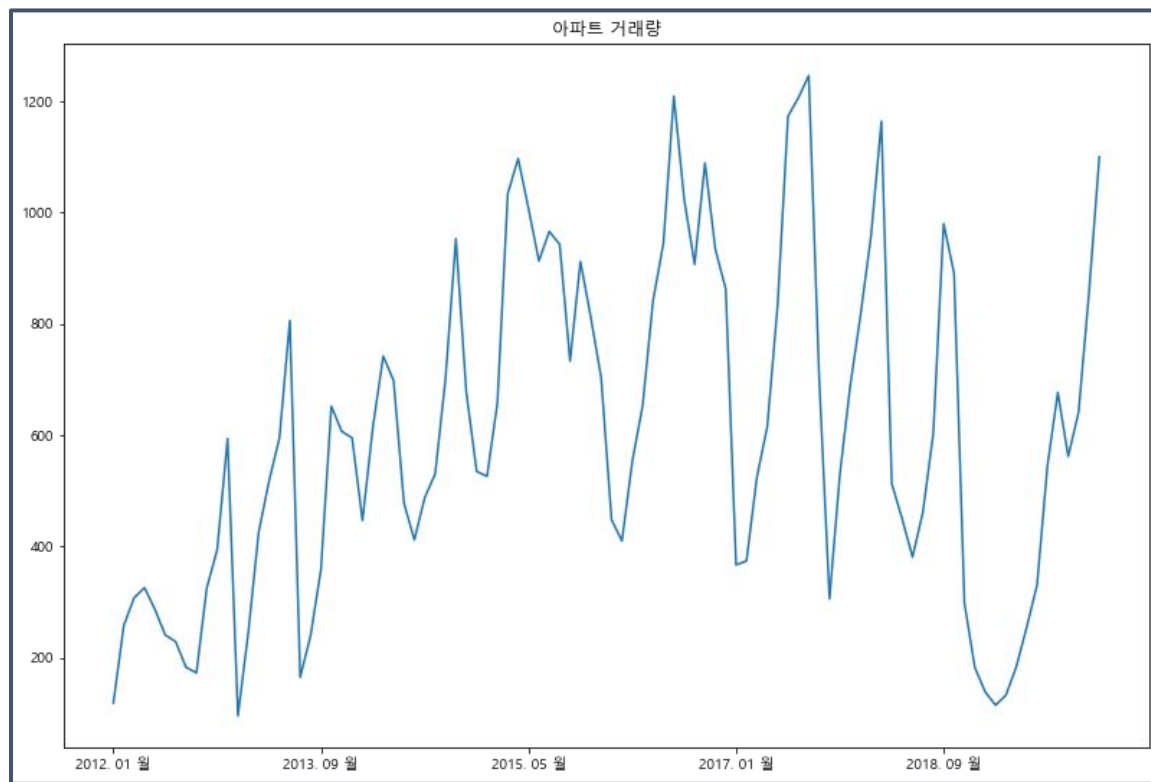


	호수	면적
2012. 01 월	1536	119
2012. 02 월	3512	258
2012. 03 월	4087	308
2012. 04 월	4214	326
2012. 05 월	3785	287
...	...	...
2019. 08 월	8586	677
2019. 09 월	7096	562
2019. 10 월	8274	641
2019. 11 월	10955	855
2019. 12 월	14117	1100
96 rows × 2 columns		

하나의 DataFrame으로 변환

# 아파트 거래량

## 아파트 거래량



- 아파트 거래량의 변동이 존재
- 거래가 감소하는 요인 : 수요의 감소
- 가격 상승 -> 실수요 감소, 투자수요 증가

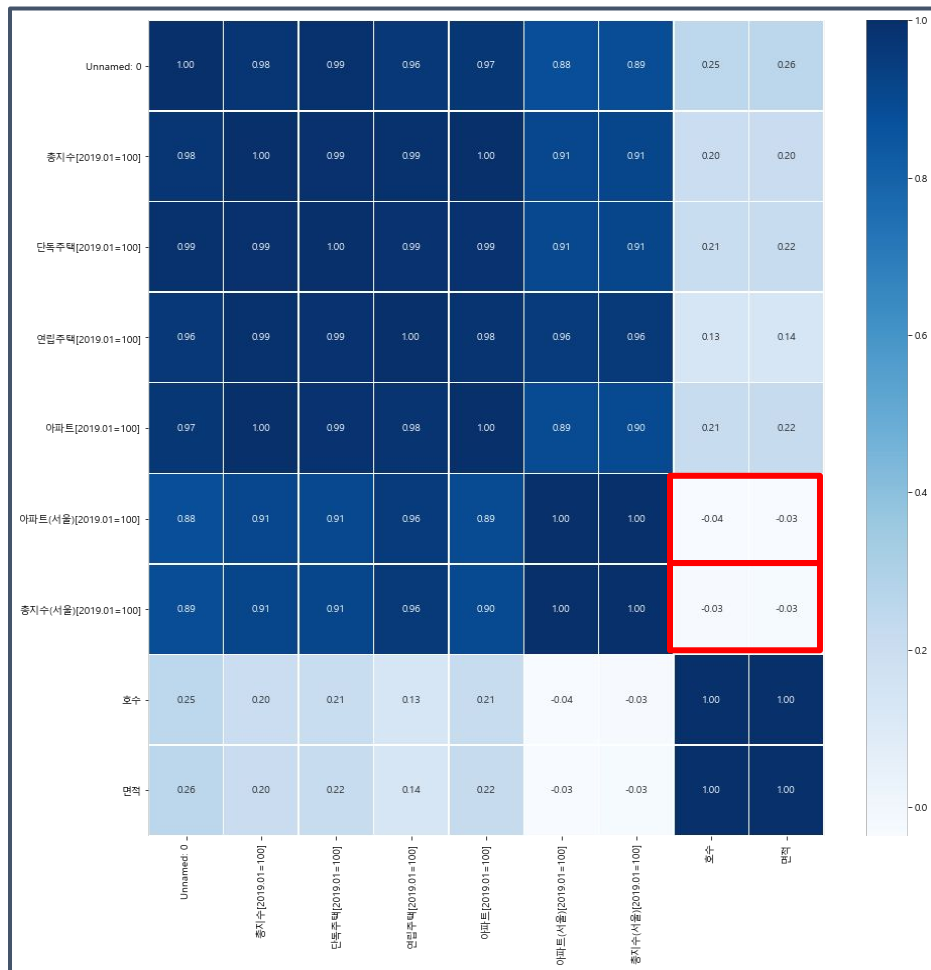


### [가격 상승에도 불구하고 수요가 감소하는 이유]

실수요뿐만 아니라 투자수요 역시 감소하는 추세  
아파트 가격과 수요는 밀접한 관계 有  
따라서 아파트 거래량은 가격과 연관이 있다고 추측

# 아파트 거래량

## 아파트 거래량(호수, 면적)과 주택매매지수의 상관관계

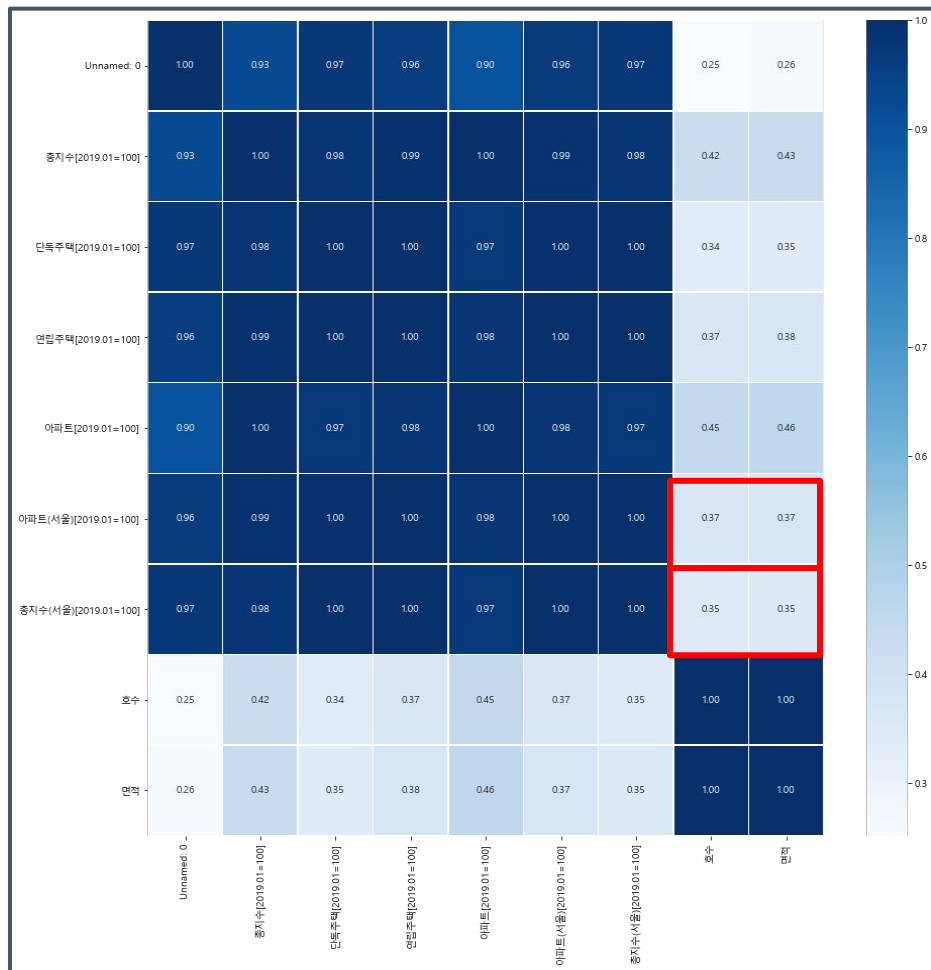


상관관계 존재하지 않음

서울 아파트 가격과 연관이 있을 것이라고 예상했으나  
상관관계가 존재하지 않았음

# 아파트 거래량

## 아파트 거래량(호수, 면적)과 주택전세지수의 상관관계



→ 약한 상관관계 존재

서울 아파트 가격과 연관이 있을 것이라고 예상했으나  
상관관계가 존재하지 않았음

출생아 수

# 출생아 수

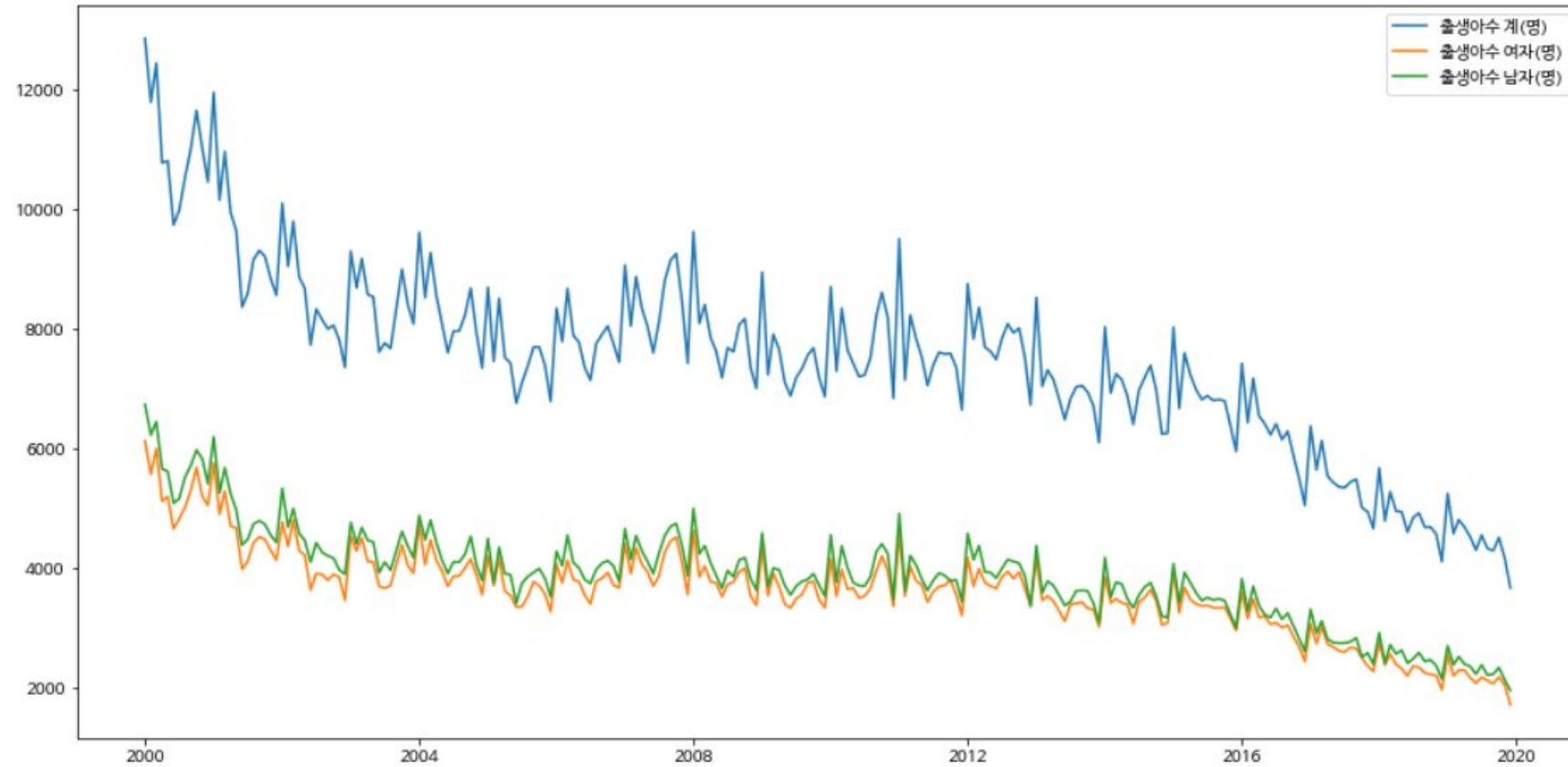
## DATA

	시점	서울특별시 계(명)	서울특별시 남자(명)	서울특별시 여자(명)
0	2000. 01	12866	6738	6128
1	2000. 02	11797	6227	5570
2	2000. 03	12453	6452	6001
3	2000. 04	10782	5662	5120
4	2000. 05	10817	5619	5198
...	...	...	...	...
235	2019. 08	4330	2211	2119
236	2019. 09	4291	2225	2066
237	2019. 10	4513	2337	2176
238	2019. 11	4170	2128	2042
239	2019. 12	3670	1953	1717

240 rows × 4 columns

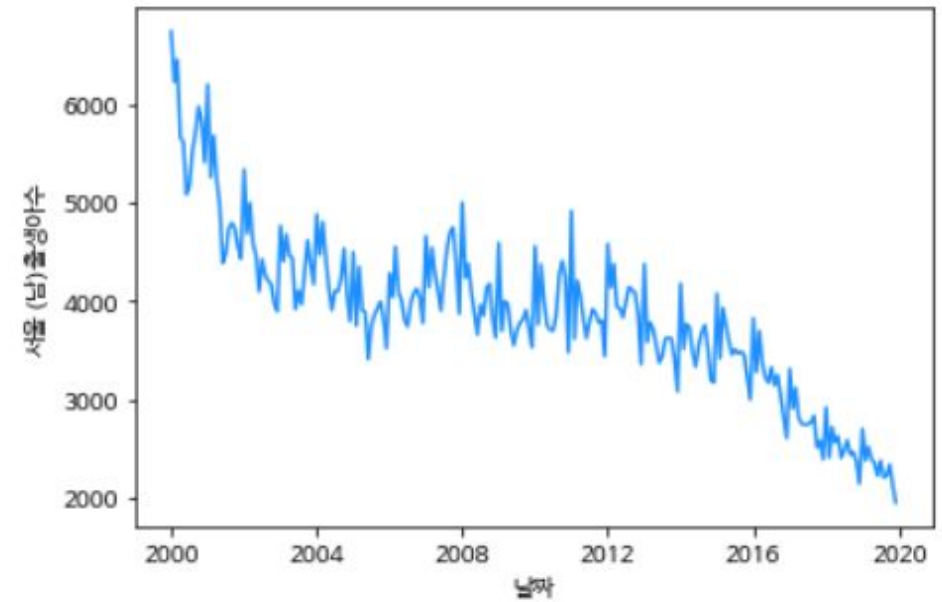
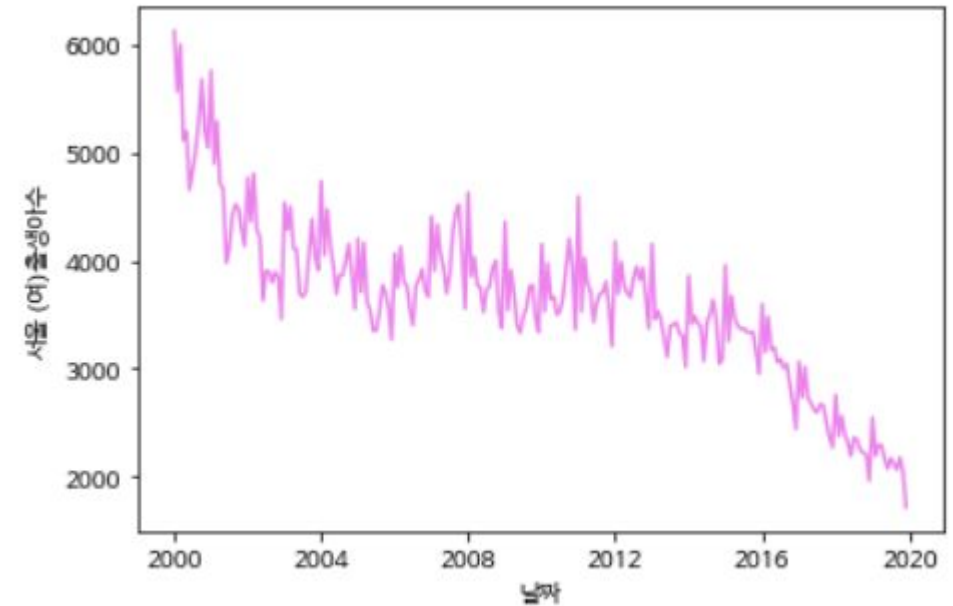
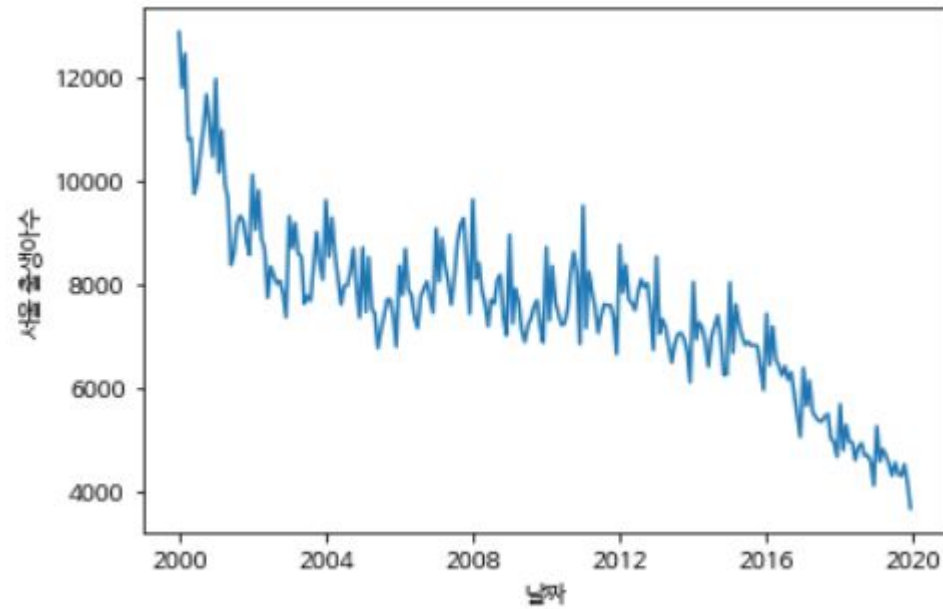
- 서울시 2000년 1월 부터 2019년 12월까지의 출생자수 데이터 사용

# 출생아 수



- 전체 출생아수는 계속 하락하는 추세이다.
- 남, 여 출생아수 또한 2020년 까지 계속 하락하는 중이다.

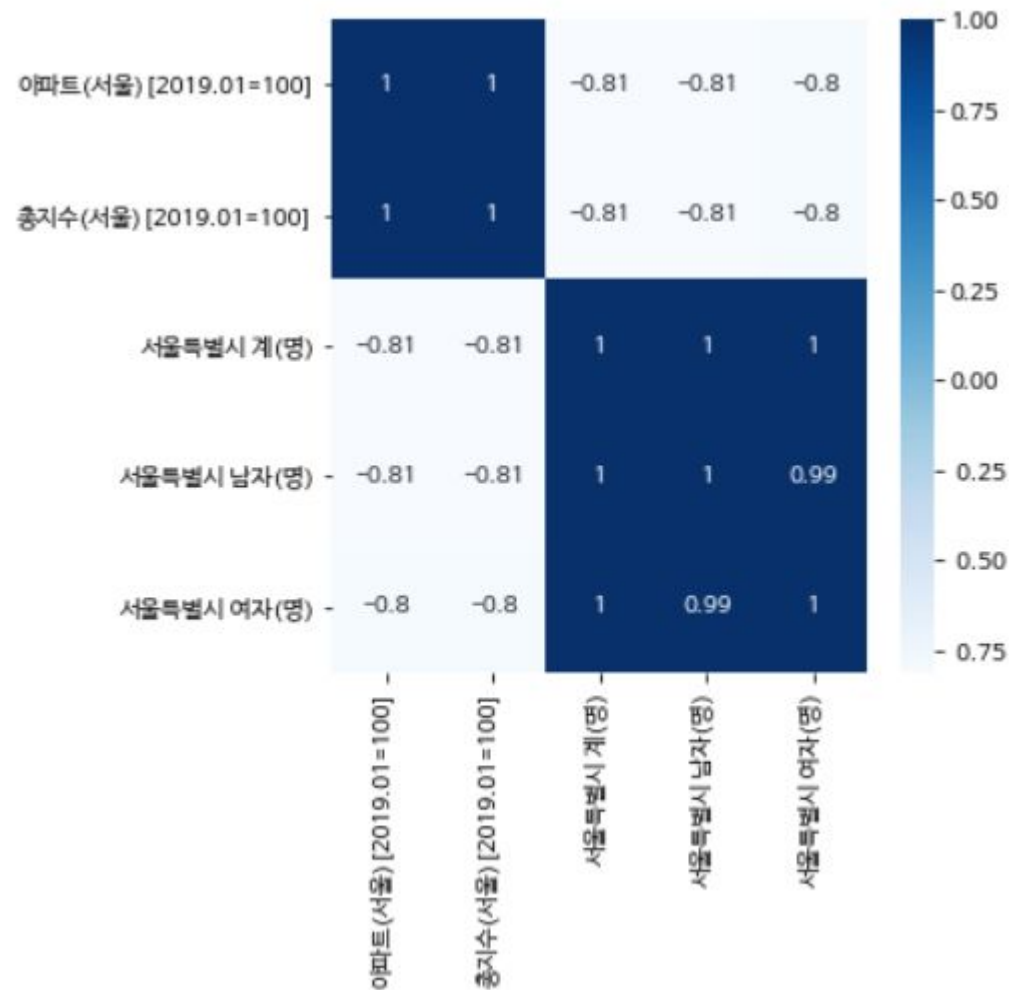
# 출생아 수



- 서울 전체 출생아수, 남/여 출생아수 각각의 그래프이다.
- 그래프의 형태가 모두 유사한것을 볼 수 있다.

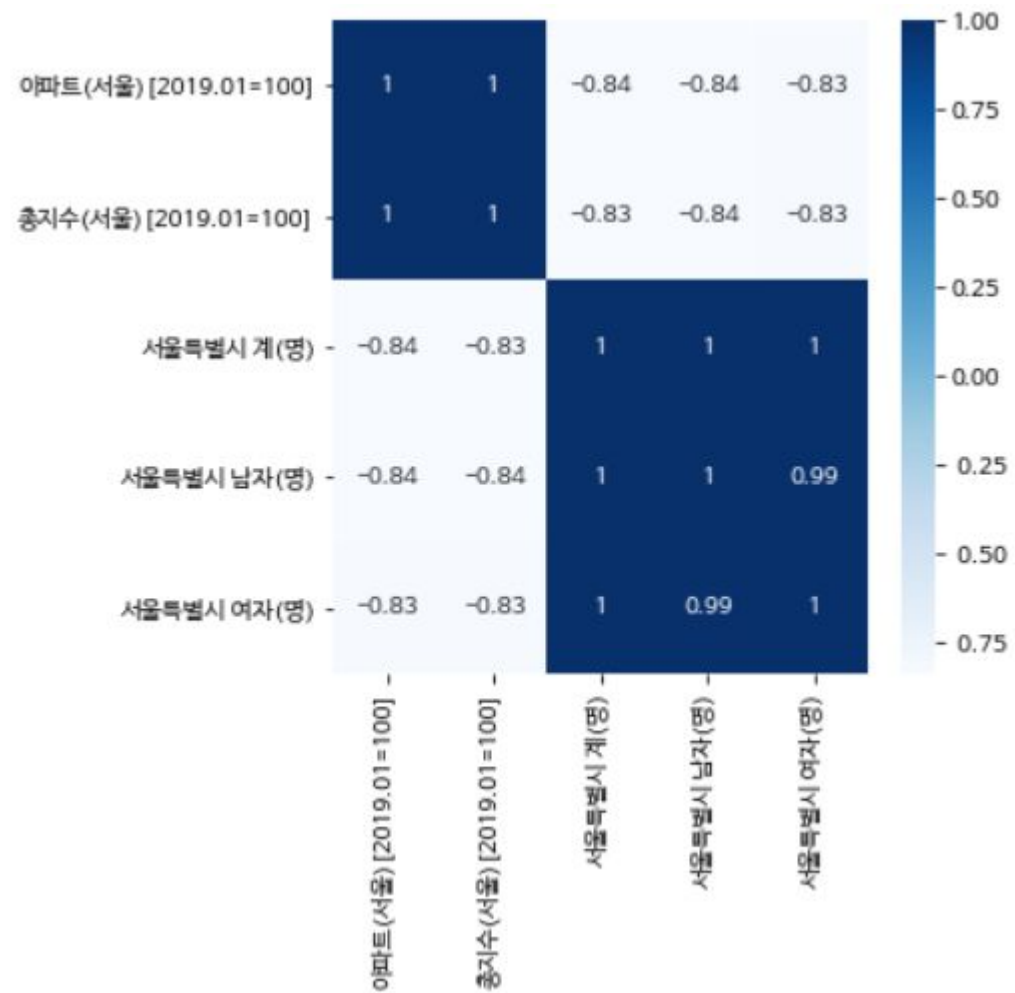


# 출생아 수와 주택매매가격지수 상관관계



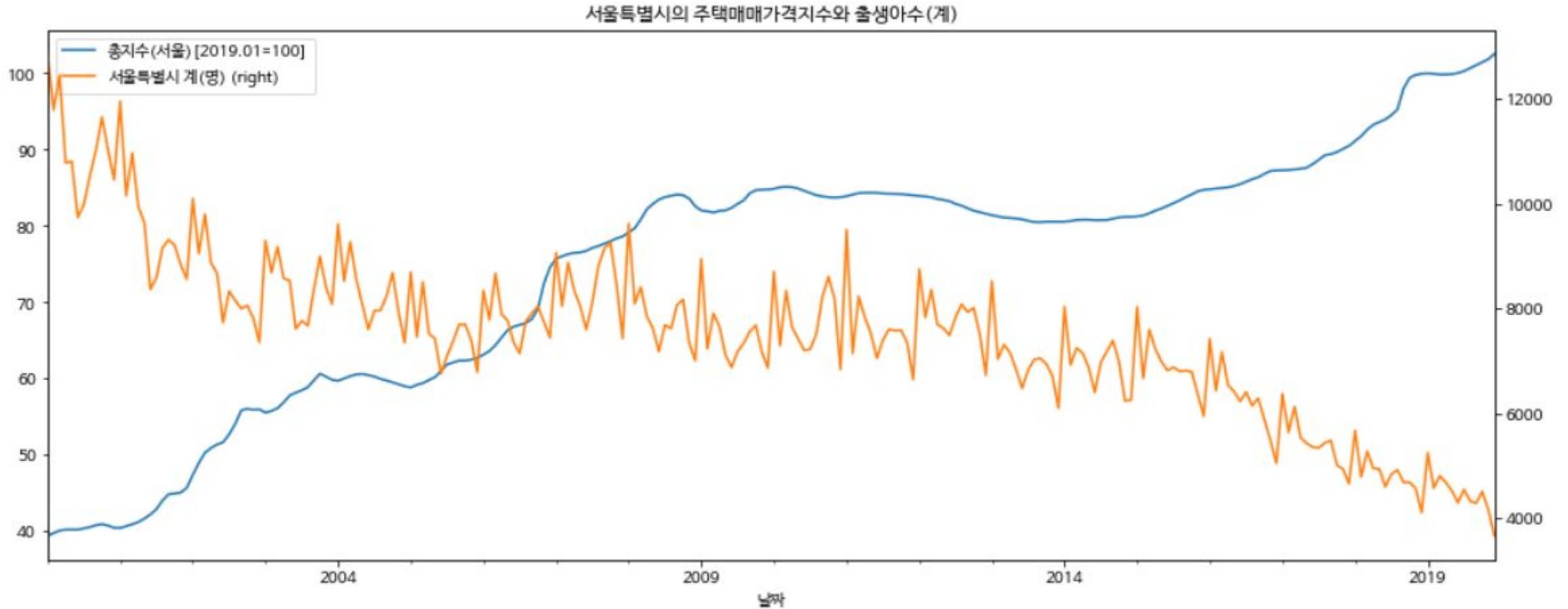
- 서울 출생아 수와 주택매매가격지수의 상관관계이다.
- 출생아수와 주택매매가격지수는 음의 상관관계를 나타나는 것을 볼 수 있다.

# 출생아 수와 주택전세가격지수 상관관계



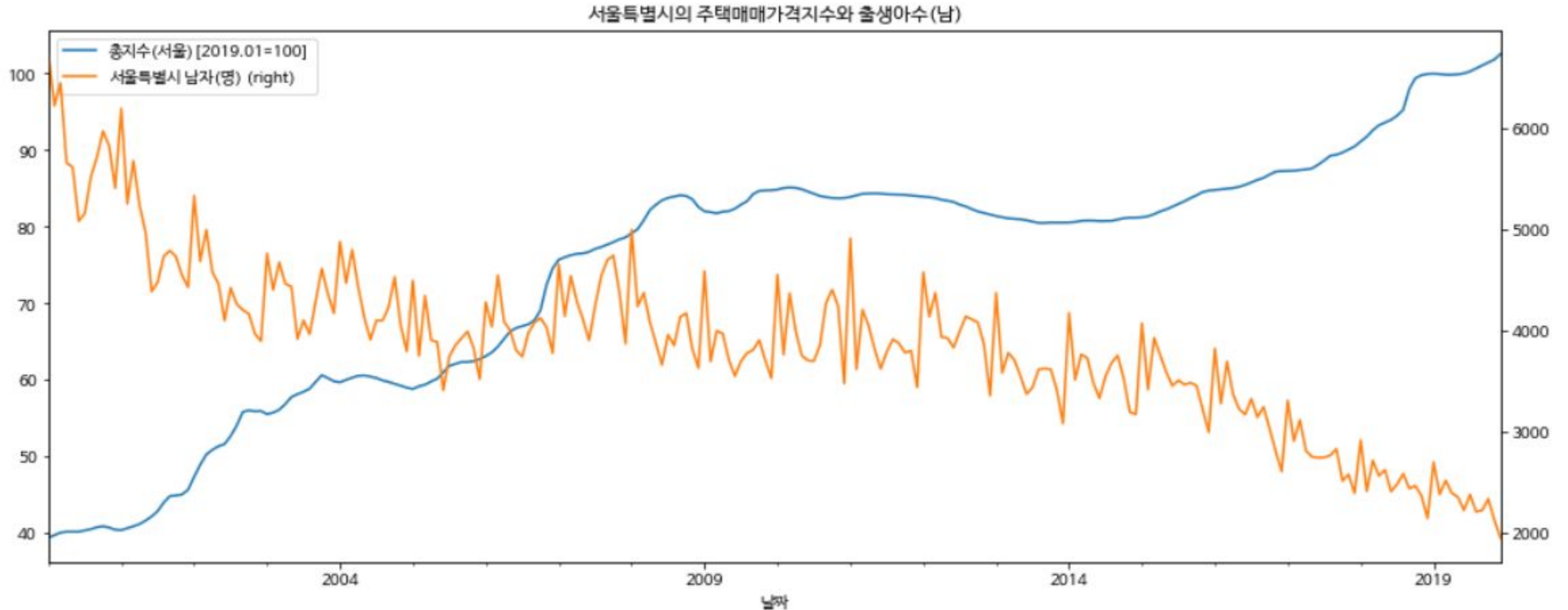
- 서울 출생아 수와 주택전세가격지수의 상관관계이다.
- 출생아수와 주택전세가격지수는 음의 상관관계를 나타나는 것을 볼 수 있다.

# 출생아 수(계)와 주택매매가격지수



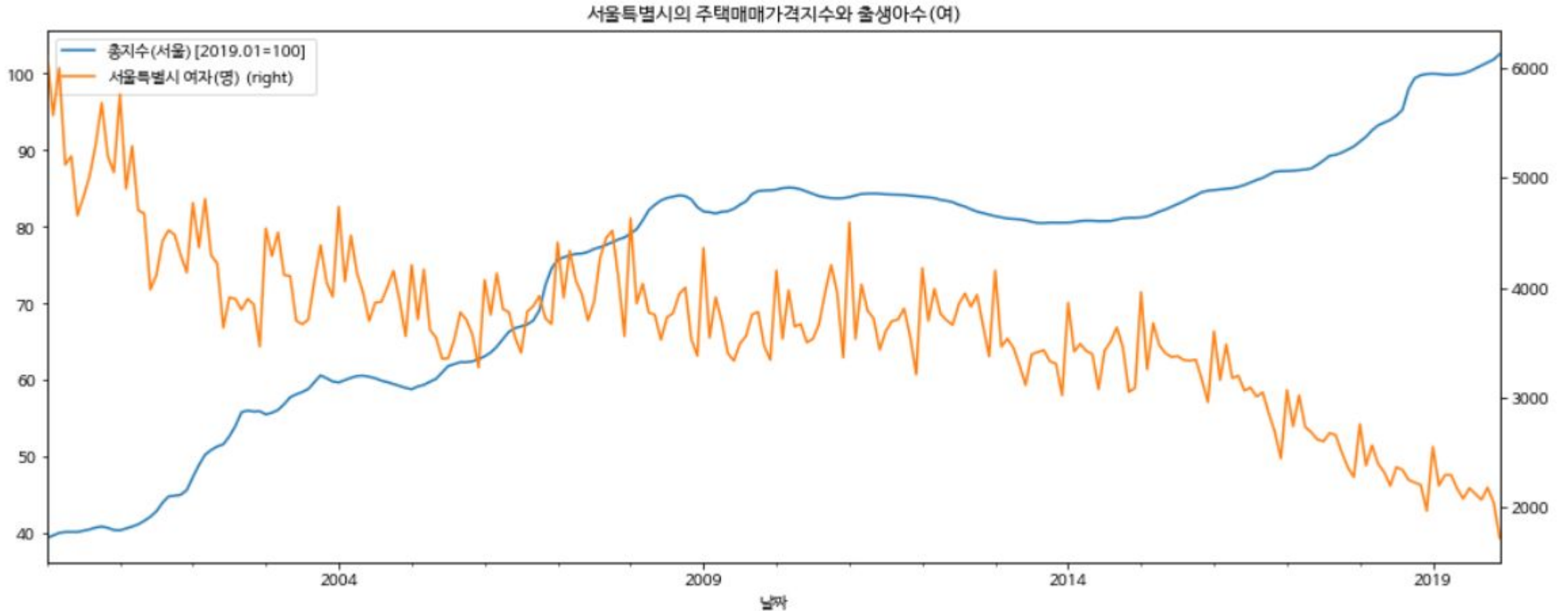
주택매매가격지수는 2019년으로 갈수록 상승하지만  
반대로 서울출생아수(계)는 하락하는 것을 볼 수 있다.

# 출생아 수(남)와 주택매매가격지수



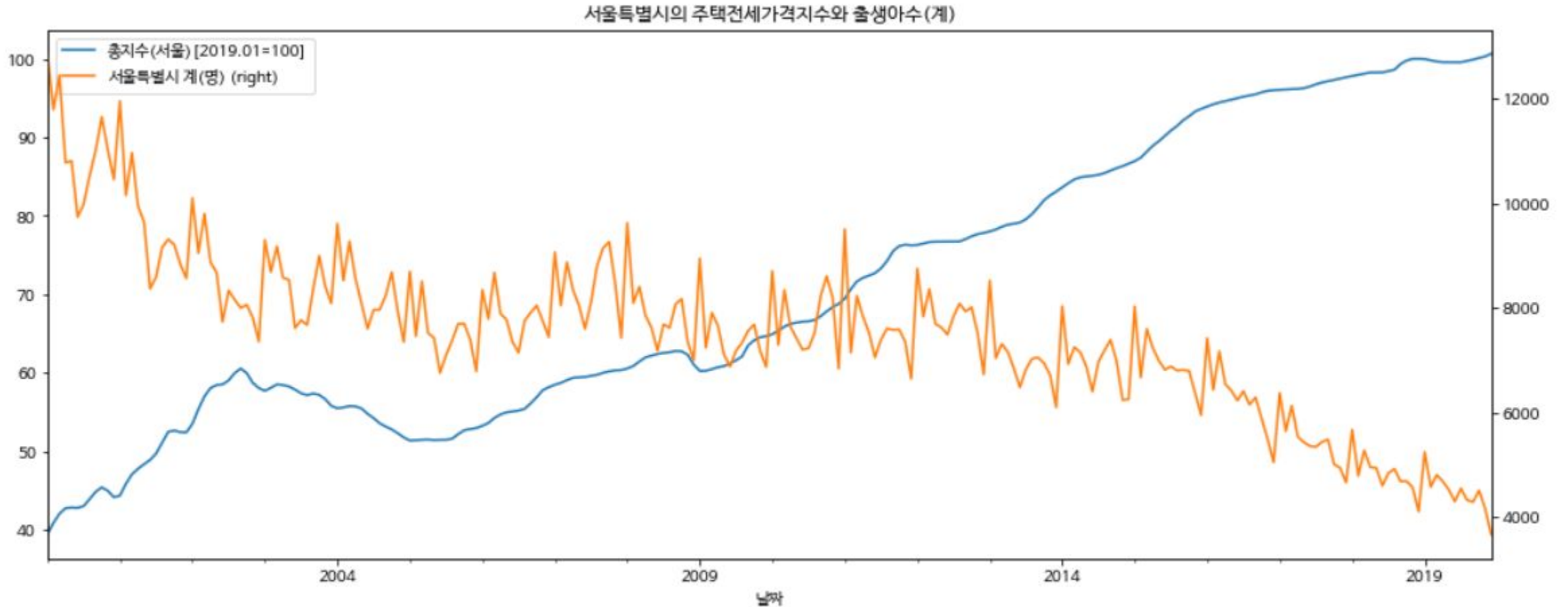
주택매매가격지수는 2019년으로 갈수록 상승하지만  
반대로 서울출생아수(남)는 하락하는 것을 볼 수 있다.

# 출생아 수(여)와 주택매매가격지수



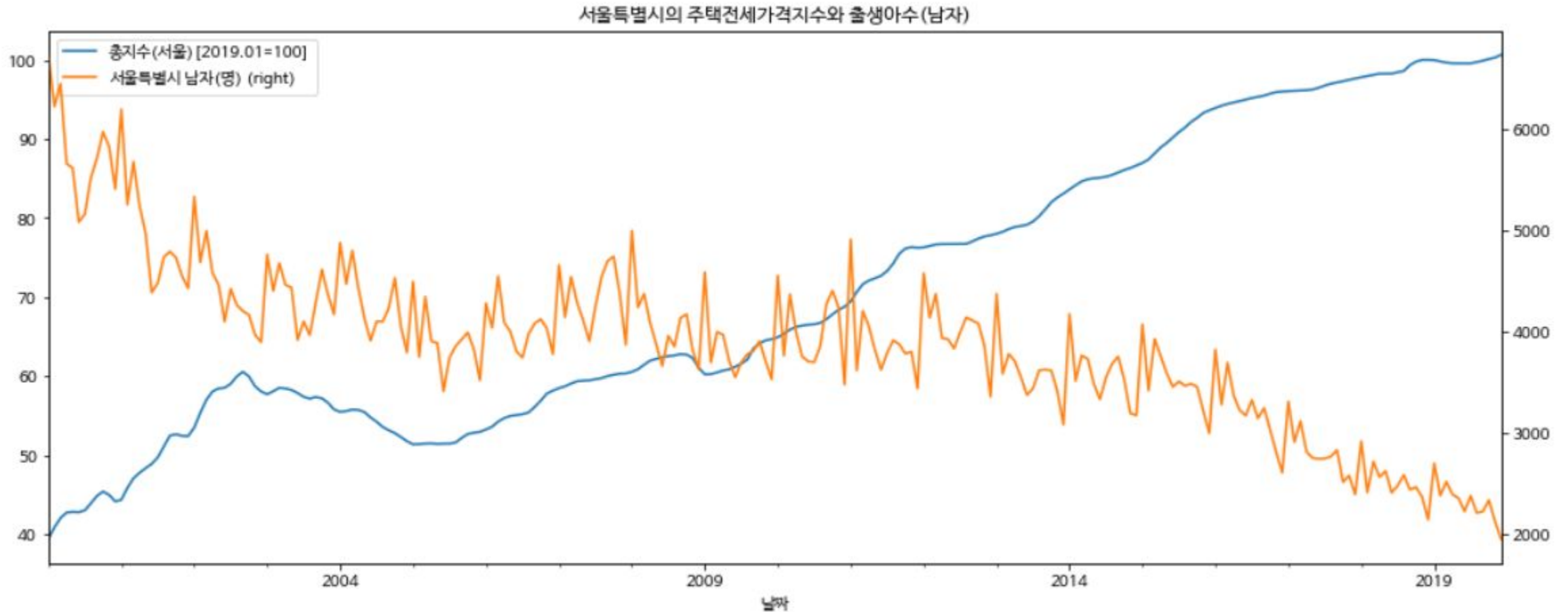
주택매매가격지수는 2019년으로 갈수록 상승하지만  
반대로 서울출생아수(여)는 하락하는 것을 볼 수 있다.

# 출생아 수(계)와 주택전세가격지수



주택전세가격지수는 2019년으로 갈수록 상승하지만  
반대로 서울출생아수(계)는 하락하는 것을 볼 수 있다.

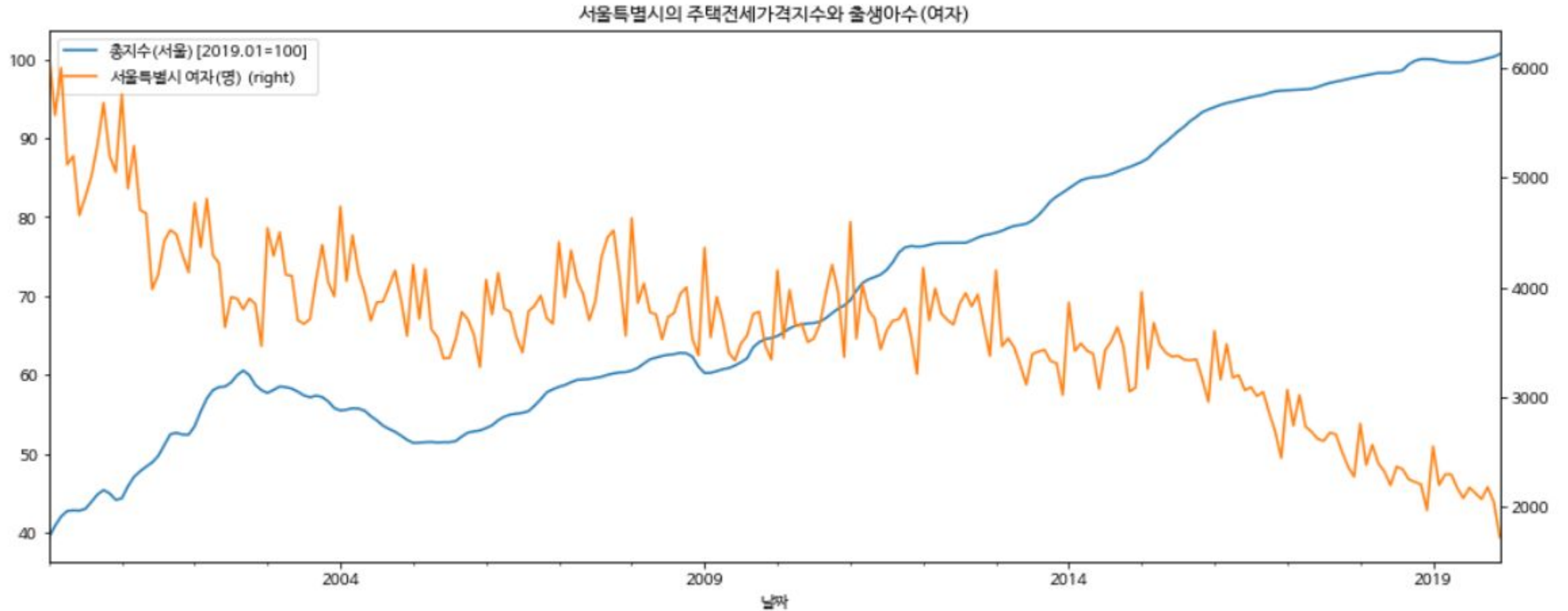
# 출생아 수(남)와 주택전세가격지수



주택전세가격지수는 2019년으로 갈수록 상승하지만  
반대로 서울출생아수(남)는 하락하는 것을 볼 수 있다.



# 출생아 수(여)와 주택전세가격지수

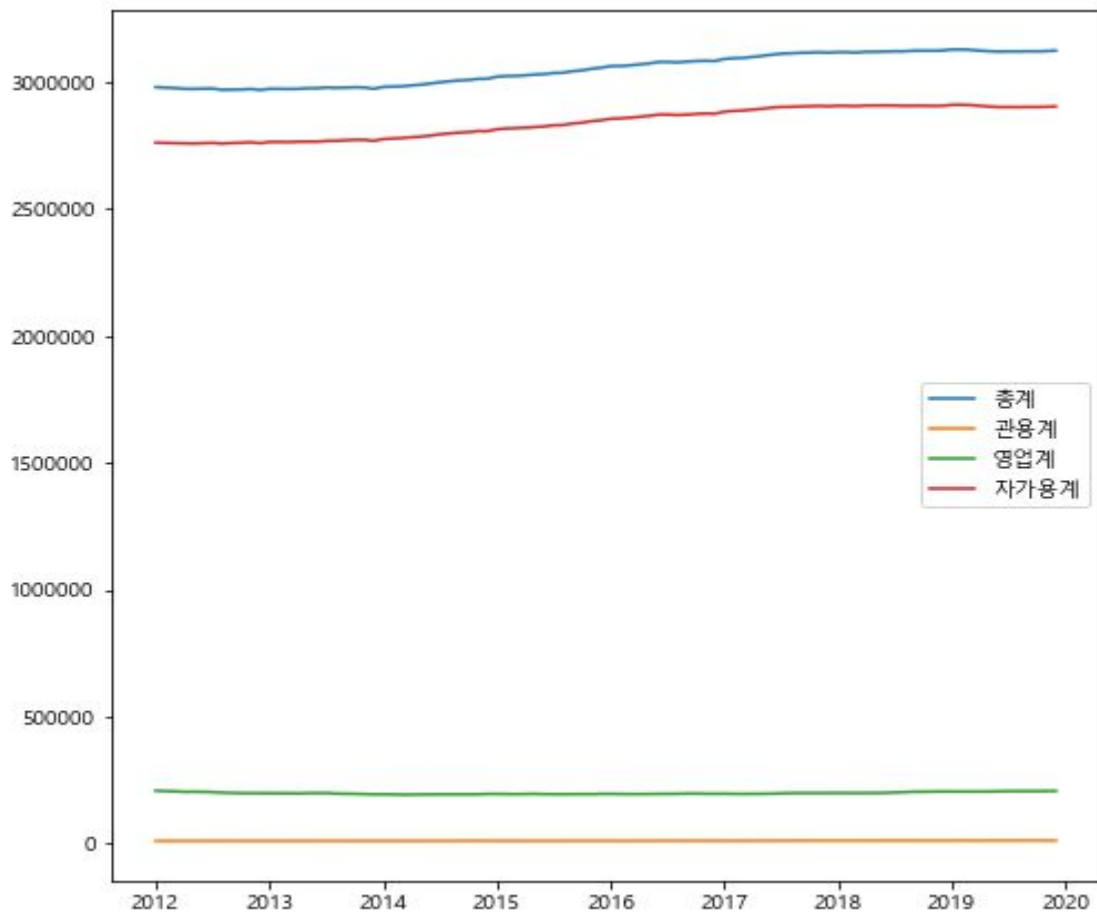


주택전세가격지수는 2019년으로 갈수록 상승하지만  
반대로 서울출생아수(계)는 하락하는 것을 볼 수 있다.



# 자동차 등록대수

# 자동차 등록대수



- 자가용의 수와 관용, 영업용 계의 차이가 매우 크다.
- 자가용의 수와 관용, 영업용 차량의 등록대수 차이가 커서 따로 살펴보도록 한다.

# 자동차 등록대수



- 총계와 자가용의 수는 2014년도를 기점으로 급증하다가 2019년을 기준으로 그 수를 유지하고 있다.
- 서울의 주택 매매지수와 어느정도 비슷한 양상을 띄고 있다.

# 자동차 등록대수

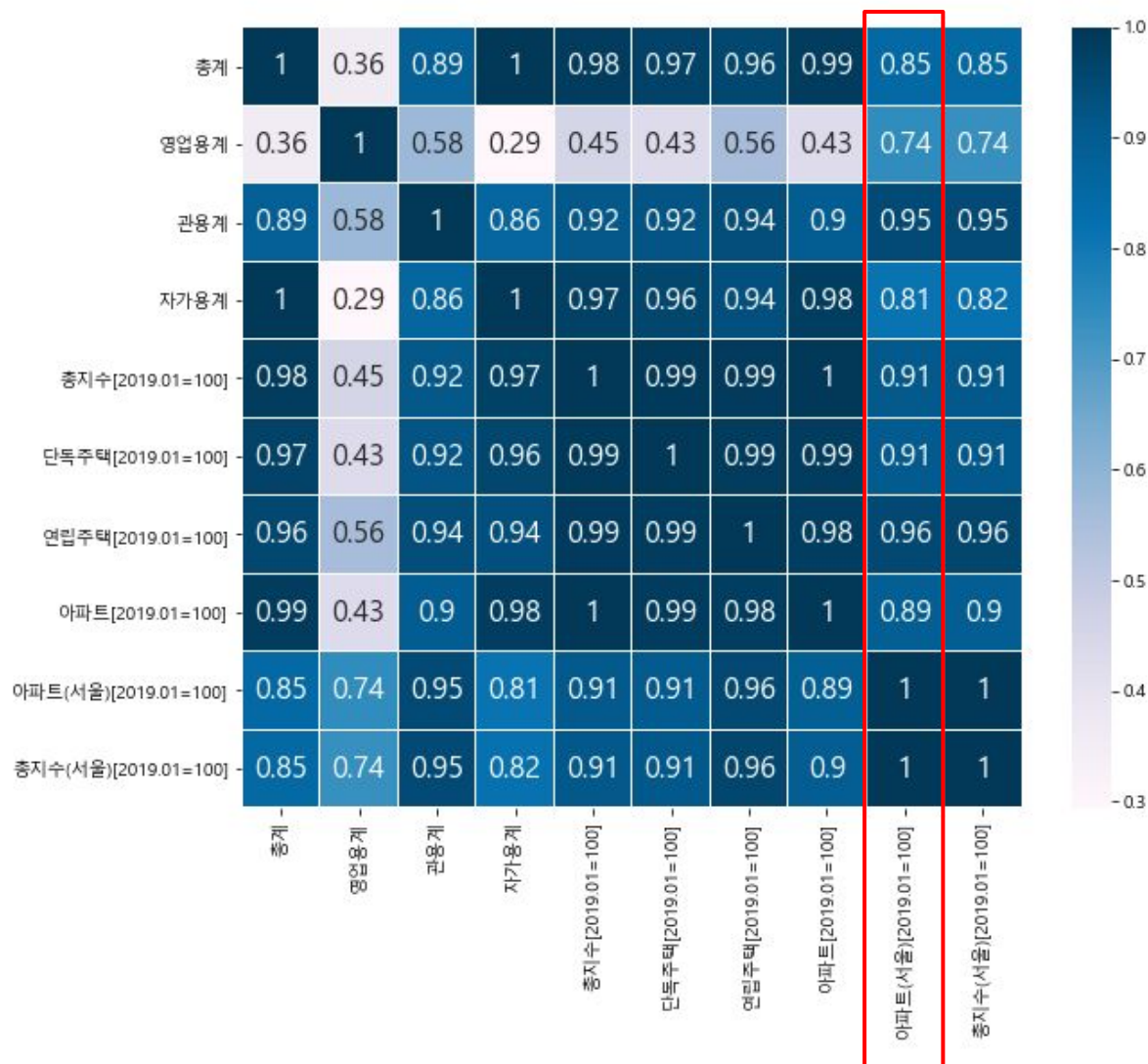


- 영업용 자동차의 수는 2014년도까지 급감하다가 2017년 이후 다시 급증하고 있다.
- 관용계의 수는 2017년을 기점으로 급증하는 모습을 보여주고 있다.

\* 관용차: 부기관과 기업체에서 업무용으로 쓰는 차

# 자동차 등록대수

자동차수&아파트 매매지수



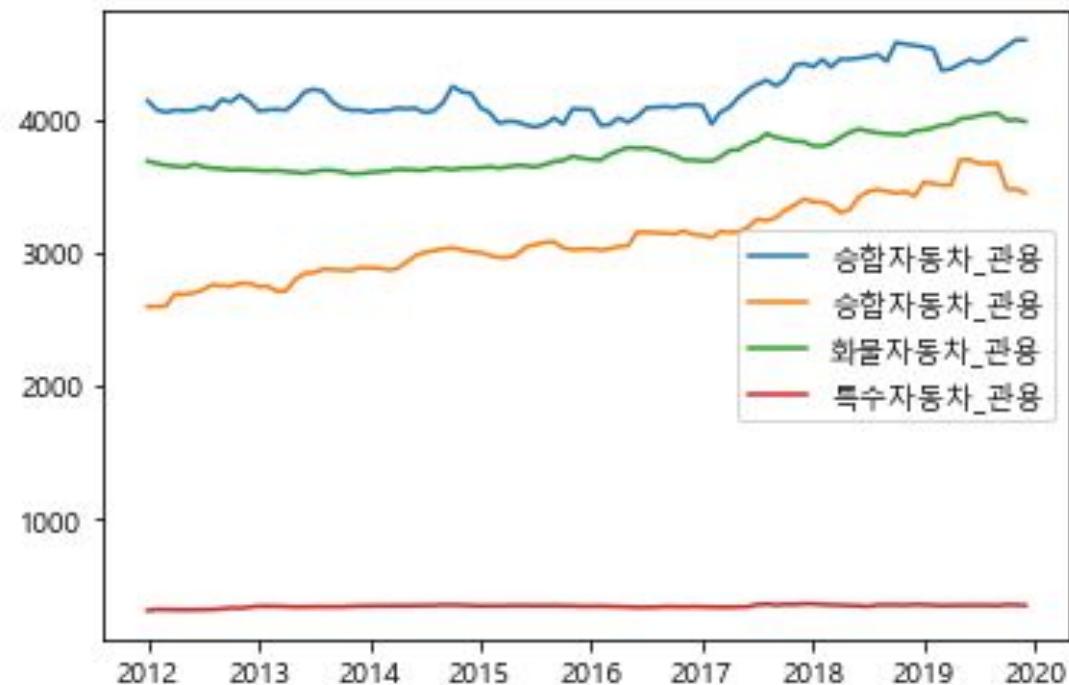
<서울 아파트의 매매지수와 자동차 등록대수의 상관관계 비교>

- 변수가 너무 많아 총계, 영업용계, 관용계, 자가용계만을 이용해 비교하였다.
- 모두 0.7 이상으로 높은 상관관계를 보여주었다.
- 특히, 관용계에서 가장 높은 상관관계를 보여주었다.

# 자동차 등록대수\_관용

관용계세분화&아파트매매지수

	승용자동차_관용	승합자동차_관용	화물자동차_관용	특수자동차_관용	총지수[2019.01=100]	단독주택[2019.01=100]	연립주택[2019.01=100]	아파트[2019.01=100]	아파트(서울)[2019.01=100]	총지수(서울)[2019.01=100]
승용자동차_관용	1	0.77	0.81	0.48	0.71	0.7	0.76	0.68	0.87	0.86
승합자동차_관용	0.77	1	0.91	0.68	0.94	0.96	0.94	0.93	0.89	0.89
화물자동차_관용	0.81	0.91	1	0.44	0.92	0.92	0.96	0.91	0.96	0.96
특수자동차_관용	0.48	0.68	0.44	1	0.54	0.58	0.5	0.53	0.41	0.42
총지수[2019.01=100]	0.71	0.94	0.92	0.54	1	0.99	0.99	1	0.91	0.91
단독주택[2019.01=100]	0.7	0.96	0.92	0.58	0.99	1	0.99	0.99	0.91	0.91
연립주택[2019.01=100]	0.76	0.94	0.96	0.5	0.99	0.99	1	0.98	0.96	0.96
아파트[2019.01=100]	0.68	0.93	0.91	0.53	1	0.99	0.98	1	0.89	0.9
아파트(서울)[2019.01=100]	0.87	0.89	0.96	0.41	0.91	0.91	0.96	0.89	1	1
총지수(서울)[2019.01=100]	0.86	0.89	0.96	0.42	0.91	0.91	0.96	0.9	1	1



<관용 자동차의 세분화와 아파트 매매지수의 상관관계수 비교>

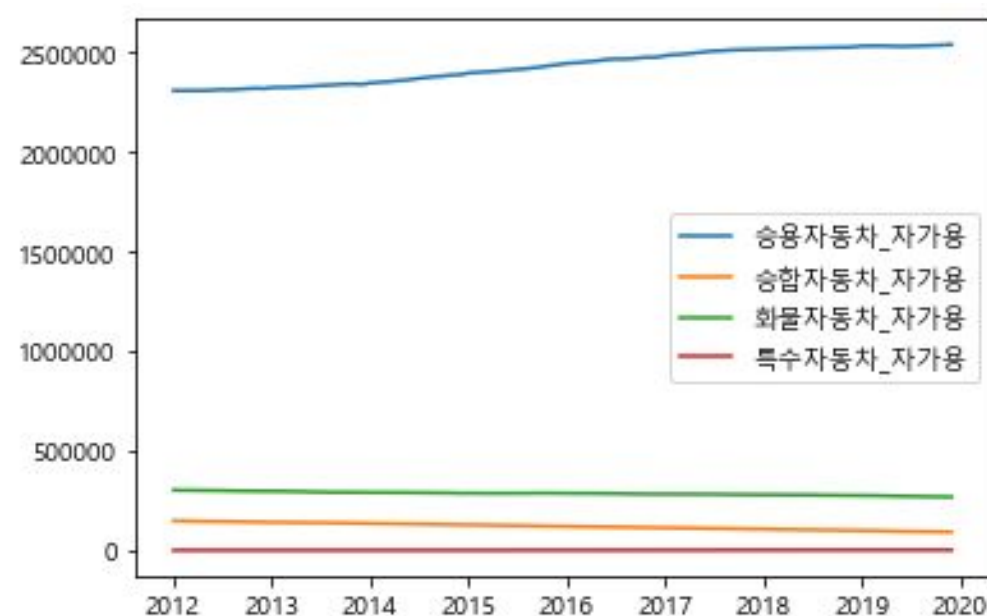
- 승용자동차와 승합자동차 화물자동차의 상관관계는 0.8 이상으로 매우 높은 편임을 알 수 있다.
- 특수자동차는 서울 아파트의 매매지수에 큰 상관관계를 가지지 못하고 특수 자동차외의 차를 관용으로 많이 보유하고있다.



# 자동차 등록대수\_자가용

자가용계세분화&아파트매매지수

승용자동차_자가용	1	-0.99	-0.95	0.97	0.98	0.98	0.95	0.98	0.84	0.84
승합자동차_자가용	-0.99	1	0.98	-0.99	-0.98	-0.99	-0.97	-0.97	-0.89	-0.89
화물자동차_자가용	-0.95	0.98	1	-0.96	-0.93	-0.96	-0.93	-0.92	-0.85	-0.86
특수자동차_자가용	0.97	-0.99	-0.96	1	0.98	0.99	0.99	0.98	0.93	0.93
총지수[2019.01=100]	0.98	-0.98	-0.93	0.98	1	0.99	0.99	1	0.91	0.91
단독주택[2019.01=100]	0.98	-0.99	-0.96	0.99	0.99	1	0.99	0.99	0.91	0.91
연립주택[2019.01=100]	0.95	-0.97	-0.93	0.99	0.99	0.99	1	0.98	0.96	0.96
아파트[2019.01=100]	0.98	-0.97	-0.92	0.98	1	0.99	0.98	1	0.89	0.9
아파트(서울)[2019.01=100]	0.84	-0.89	-0.85	0.93	0.91	0.91	0.96	0.89	1	1
총지수(서울)[2019.01=100]	0.84	-0.89	-0.86	0.93	0.91	0.91	0.96	0.9	1	1

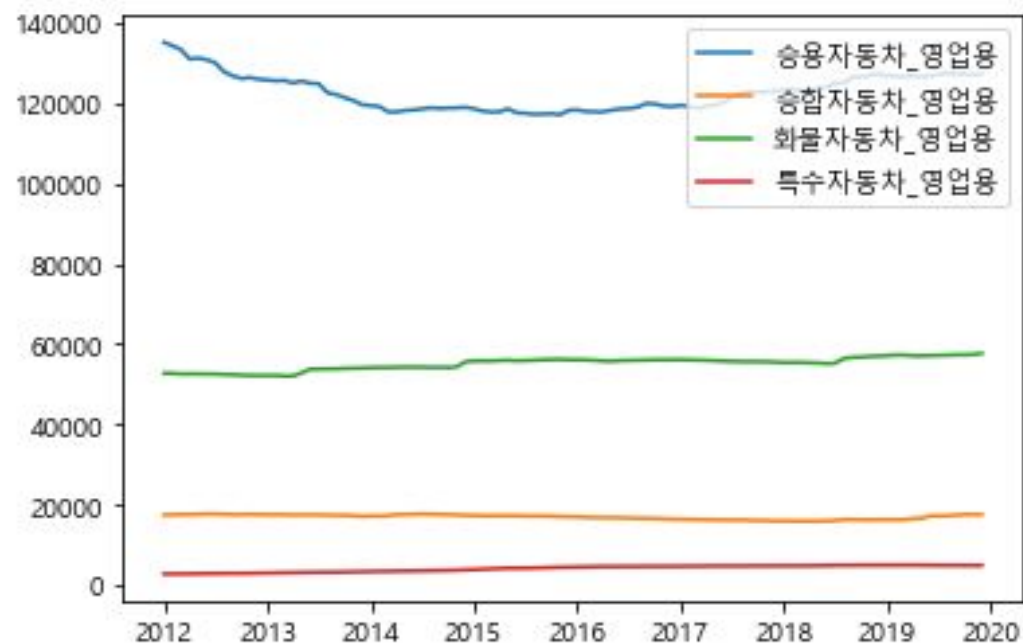
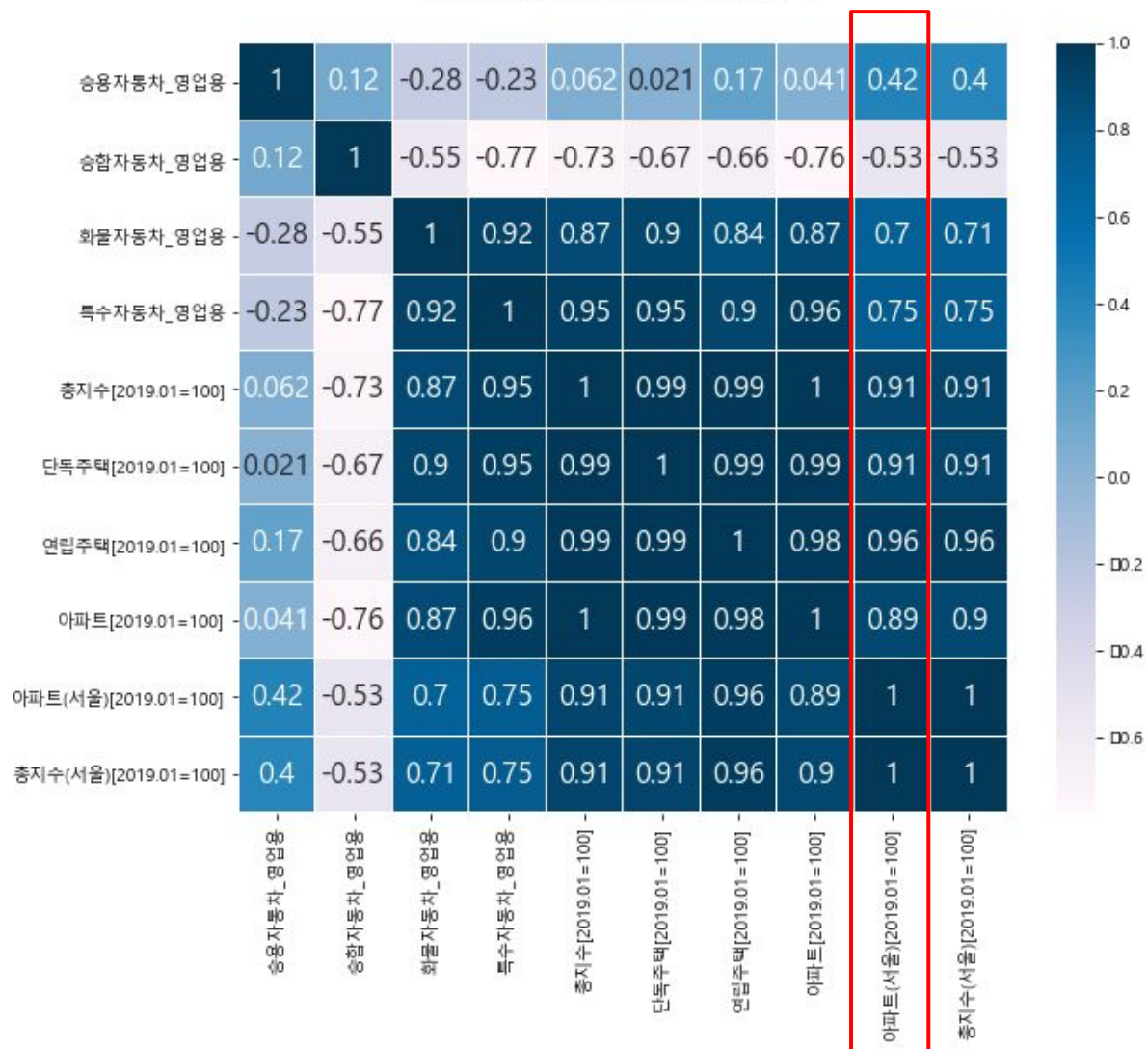


<자가용자동차의 세분화와 아파트 매매지수의 상관관계수 비교>

- 승용자동차와 특수자동차의 경우 아파트 매매지수와 큰 양의관계를 가지지만 승합자동차와 화물 자동차의 경우 음의 관계를 가진다.
- 자가용의 경우는 승용자동차 수가 가장 많음을 알 수 있다.

# 자동차 등록대수\_영업용

영업용계세분화&아파트매매지수



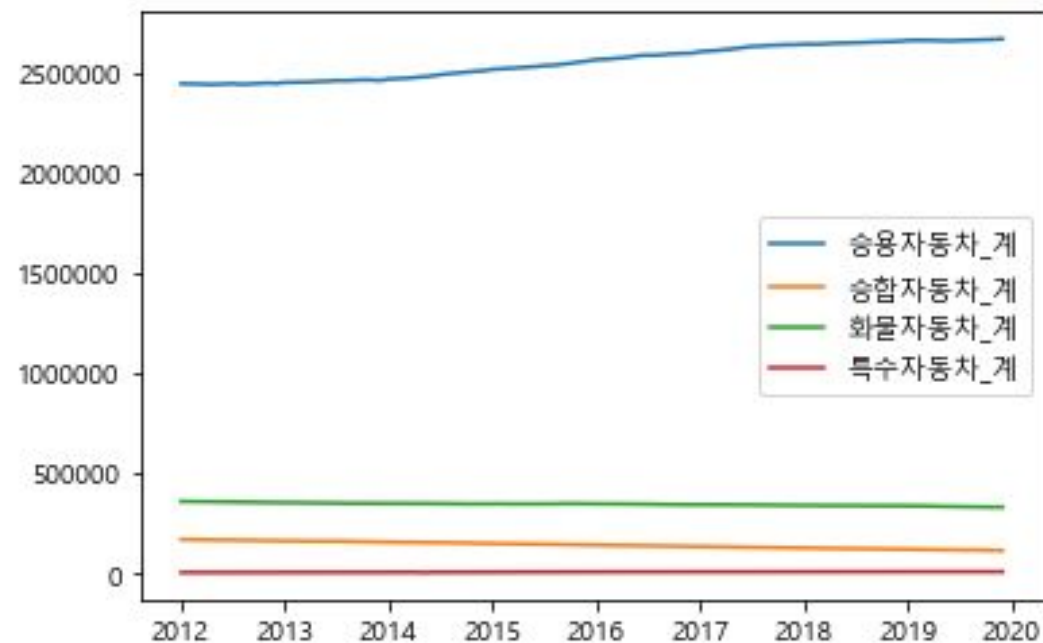
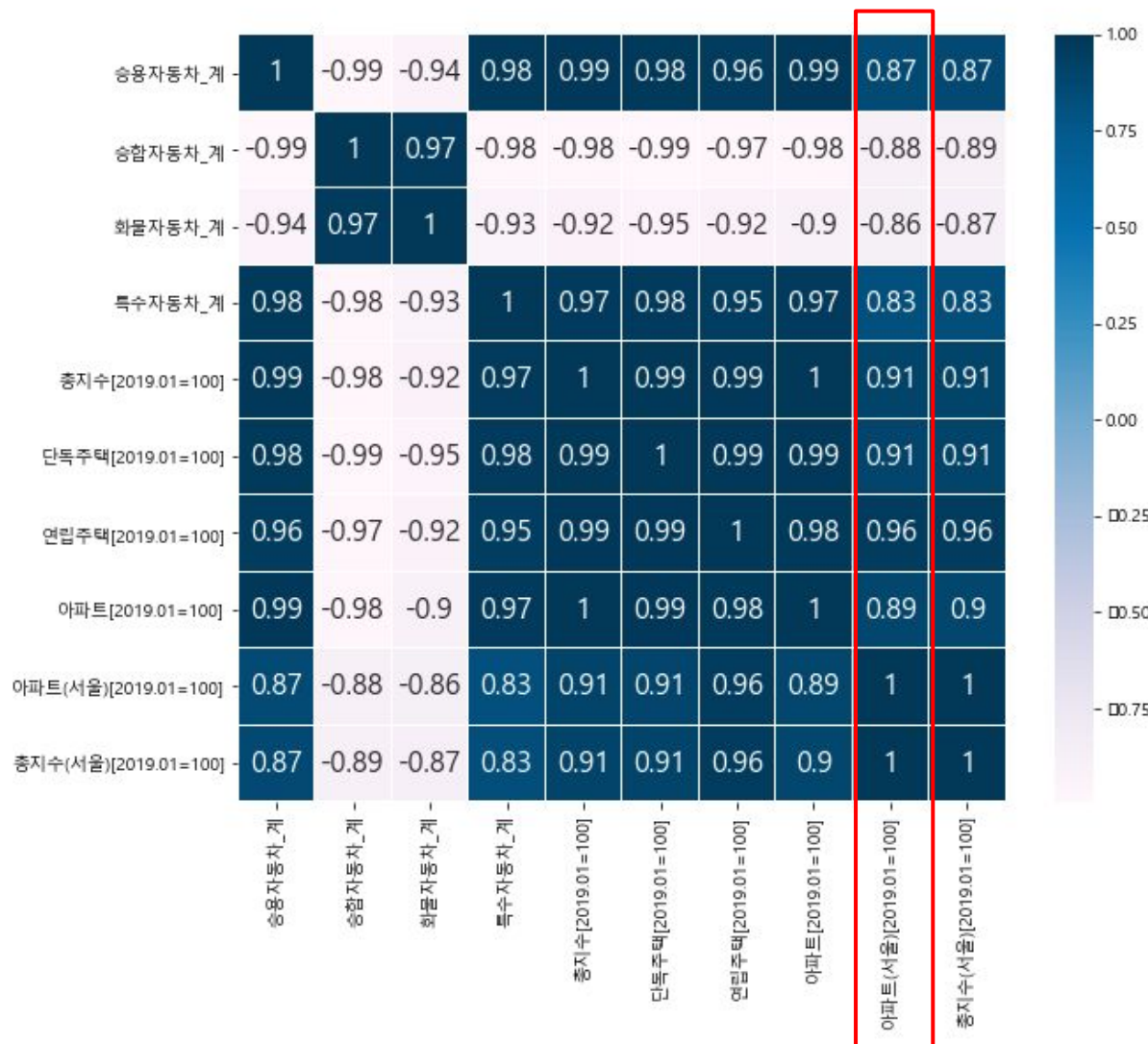
<영업용자동차의 세분화와 아파트 매매지수의 상관관계수 비교>

- 화물자동차와 특수 자동차가 그나마 높은 상관관계를 가진다.
- 다른 자동차의 용도에 비해 상대적으로 다 상관관계가 낮다.



# 자동차 등록대수\_계

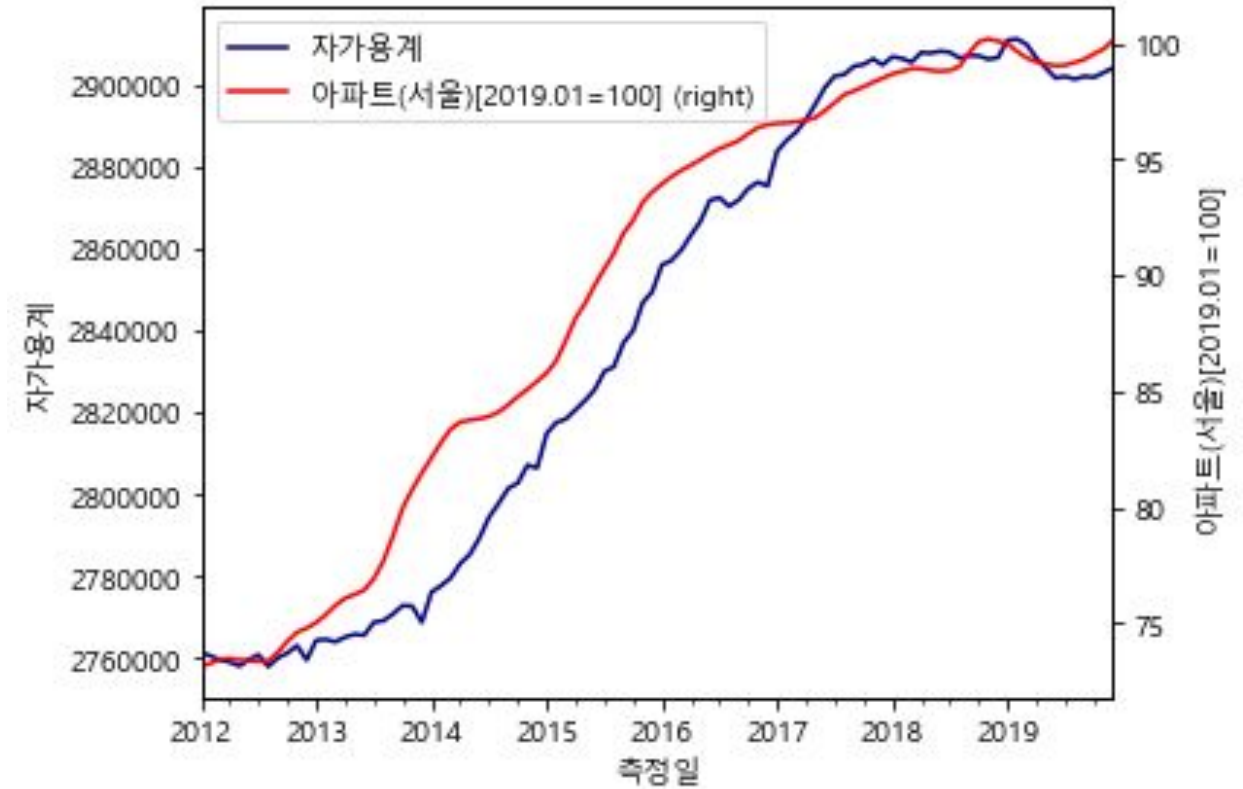
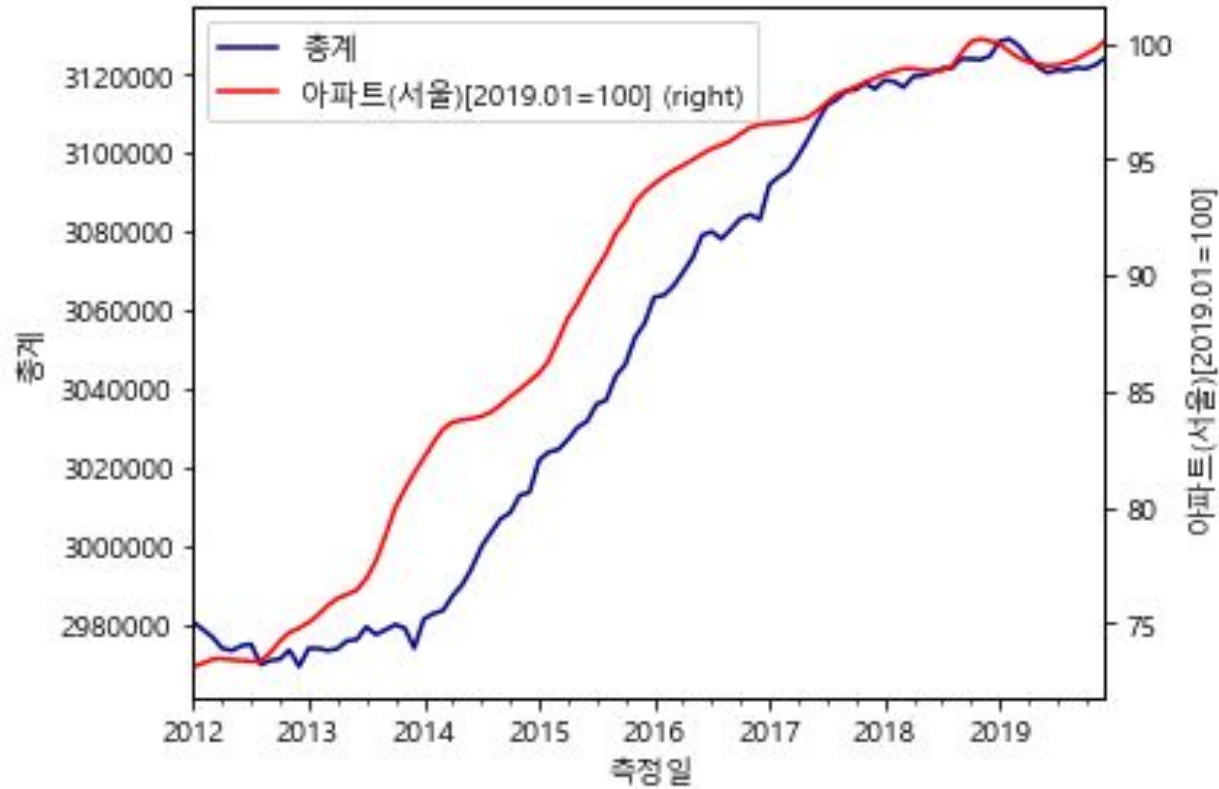
계세분화&아파트매매지수



<영업용자동차의 세분화와 아파트 매매지수의 상관관계수 비교>

- 승용자동차와 특수자동차는 큰 양의 관계를 가지고 승합자동차와 화물자동차는 큰 음의 관계를 가진다.
- 자가용 자동차와 비슷한 형상을 띄고 있다.
- 승용자동차의 수를 제외하고는 모두 감소하는 추세를 보인다.

# 자동차 등록대수\_매매지수비교



- 총계와 자가용계의 경우 서울 아파트 매매지수와 매우 흡사한 형태를 띄고 있다.

# 자동차 등록대수\_매매지수비교



- 관용계는 증가한다는 면에서 흡사하지만 총계와 자가용계에 비해서 아파트 매매지수가 비슷하다고 볼 수 없다.
- 영업용계의 경우는 2014년 이후로는 비슷하지만 그 전에는 매우 다르다.

모델링

# LSTM

## 전처리

출생아수 데이터

시점	항목	전국	서울특별시
2000. 01	계 (명)	61644	12866
2000. 01	남자 (명)	32317	6738
2000. 01	여자 (명)	29327	6128
2000. 02	계 (명)	56723	11797
2000. 02	남자 (명)	29693	6227
2000. 02	여자 (명)	27030	5570
2000. 03	계 (명)	59878	12453
2000. 03	남자 (명)	31432	6452
2000. 03	여자 (명)	28446	6001
2000. 04	계 (명)	53058	10782

아파트 거래 내역

	호수	면적
2012. 01 월	1536	119
2012. 02 월	3512	258
2012. 03 월	4087	308
2012. 04 월	4214	326
2012. 05 월	3785	287
...	...	...
2019. 08 월	8586	677
2019. 09 월	7096	562
2019. 10 월	8274	641
2019. 11 월	10955	855
2019. 12 월	14117	1100

96 rows × 2 columns

자동차 등록 대수

서울특별시 자가용
2,761,512
2,760,526
2,759,618
2,759,047
2,758,106
...
2,901,406
2,902,056
2,901,869
2,903,003
2,904,170

- 출생아 수 데이터: 남녀를 합친 수치 인 계의 서울 특별시 데이터만 사용
- 아파트 거래 내역: 월별 거래 횟수에 해당하는 호수 데이터만 사용
- 자동차 등록 대수: 서울특별시의 자가용 데이터만 사용

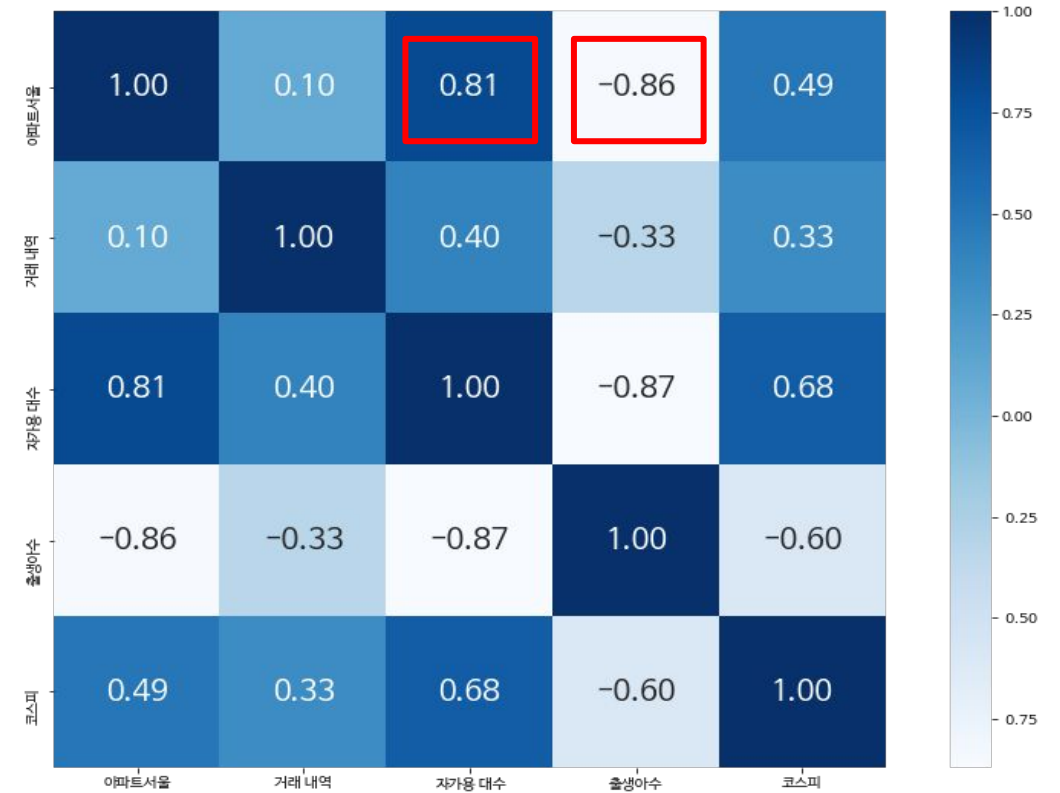
# LSTM

## 전처리

	날짜	총지수	단독주택	연립주택	아파트	아파트서울	총지수서울	거래 내역	자가용 대수	출생아수	코스피
0	2012-01-01	88.425	89.553	92.626	87.785	80.089	83.954	1536	2761512	8758	1955.79
1	2012-02-01	88.573	89.706	92.682	87.953	79.978	83.878	3512	2760526	7837	2030.25
2	2012-03-01	88.707	89.799	92.752	88.122	79.779	83.776	4087	2759618	8364	2014.04
3	2012-04-01	88.802	89.904	92.735	88.247	79.433	83.553	4214	2759047	7700	1981.99
4	2012-05-01	88.845	90.024	92.774	88.272	79.224	83.424	3785	2758106	7629	1843.47
...	...	...	...	...	...	...	...	...	...	...	...
91	2019-08-01	99.586	100.753	100.300	99.014	100.229	100.682	8586	2901406	4330	1967.79
92	2019-09-01	99.640	100.760	100.424	99.041	100.677	101.060	7096	2902056	4291	2063.05
93	2019-10-01	99.733	100.766	100.473	99.152	101.260	101.432	8274	2901869	4513	2083.48
94	2019-11-01	99.874	100.869	100.567	99.297	101.830	101.848	10955	2903003	4170	2087.96
95	2019-12-01	100.222	101.002	100.718	99.726	102.920	102.560	14117	2904170	3670	2197.67

96 rows x 11 columns

## 상관 관계 분석



- 추가 변수인 거래 내역, 자가용 대수, 출생아수, 코스피 데이터를 데이터 기간에 맞춰 2012년부터 2019년까지 데이터 병합
- 상관 관계 분석: 양의 상관관계를 보인 자가용 대수와 음의 상관관계를 보인 출생아 수를 추가변수로 선택



# LSTM

## 전처리

```
#정규화
data = df.loc[:,['자가용 대수', '출생아수', '아파트서울']]
target = df.loc[:,['아파트서울']]

data_scaler = MinMaxScaler()
data_scaled = data_scaler.fit_transform(data)

target_scaler = MinMaxScaler()
target_scaled = target_scaler.fit_transform(target)
```

$$Y = \frac{(X - X_{min})}{(X_{max} - X_{min})}$$

MinMax Scaler 수식



### 정규화 결과

	자가용 대수	출생아수	아파트서울
0	0.022977	1.000000	0.178061
1	0.016541	0.818986	0.174065
2	0.010614	0.922563	0.166901
3	0.006887	0.792060	0.154444
4	0.000744	0.778105	0.146920
...	...	...	...
91	0.936147	0.129717	0.903121
92	0.940390	0.122052	0.919250
93	0.939169	0.165684	0.940238
94	0.946572	0.098270	0.960759
95	0.954189	0.000000	1.000000

96 rows × 3 columns

- MinMaxScaler를 이용하여 데이터 정규화 진행

# LSTM

## 전처리

window size: 5

	자가용 대수	출생아수	아파트서울
0	0.022977	1.000000	0.178061
1	0.016541	0.818986	0.174065
2	0.010614	0.922563	0.166901
3	0.006887	0.792060	0.154444
4	0.000744	0.778105	0.146920
...	...	...	...
91	0.936147	0.129717	0.903121
92	0.940390	0.122052	0.919250
93	0.939169	0.165684	0.940238
94	0.946572	0.098270	0.960759
95	0.954189	0.000000	1.000000

96 rows x 3 columns

```
#window 생성
window_size = 5
target_size = 1

data_list = []
target_list = []

for i in range(len(data) - window_size - target_size):
    data_list.append(np.array(data.iloc[i : i + window_size]))
    target_list.append(np.array(target.iloc[i + window_size : i + window_size + target_size]))

data_ar = np.array(data_list)
target_ar = np.array(target_list)
target_ar = np.reshape(target_ar, (target_ar.shape[0], target_ar.shape[1]))
```

```
# train set, test set 분할
x_train, x_test, y_train, y_test = train_test_split(data_ar, target_ar, test_size=0.2, shuffle=False)
```

```
# train set, test set 분할 후 shape
x_train_shape = np.shape(x_train)
y_train_shape = np.shape(y_train)
x_test_shape = np.shape(x_test)
y_test_shape = np.shape(y_test)

print('x_train:', x_train_shape, 'y_train:', y_train_shape)
print('x_test:', x_test_shape, 'y_test:', y_test_shape)

x_train: (72, 5, 3) y_train: (72, 1)
x_test: (18, 5, 3) y_test: (18, 1)
```

- 과거 5달의 데이터로 다음 달을 예측
- 다음과 같이 window size를 5로 설정 한 후 window를 이동하며 데이터를 생성
- 데이터를 생성한 후 8:2의 비율로 훈련 데이터와 테스트 데이터 생성



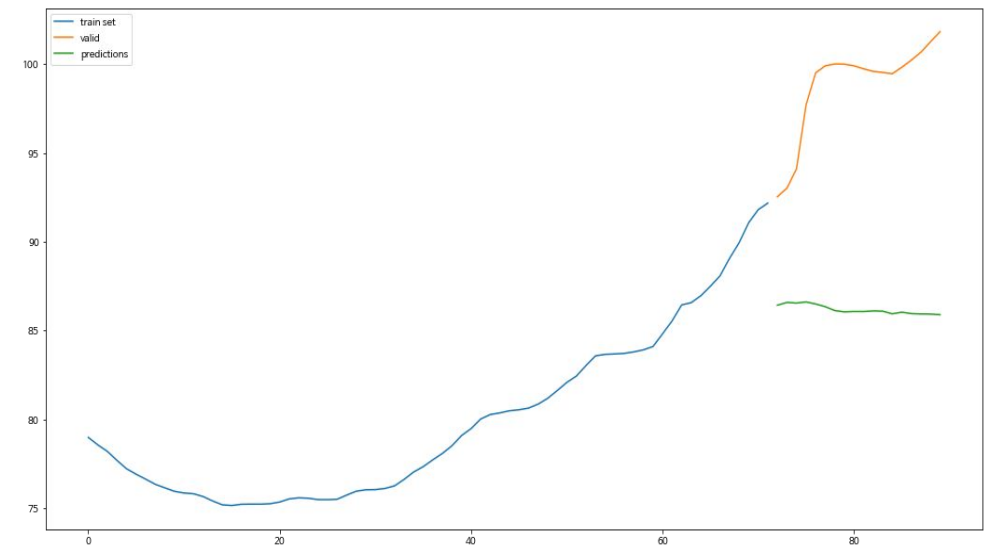
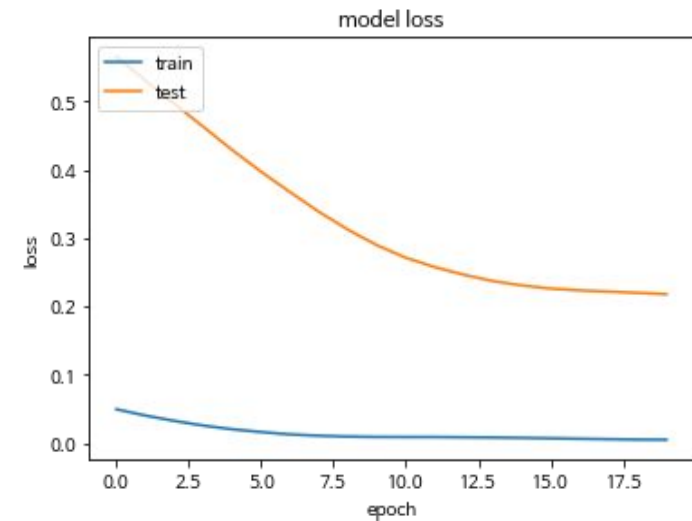
# LSTM

## 모델링

```
#모델 생성
model = Sequential()
model.add(LSTM(5, return_sequences=True, input_shape = (x_train.shape[1], x_train.shape[2])))
model.add(LSTM(10, return_sequences=False))
model.add(Dense(5))
model.add(Dense(1))
model.compile(loss='mse', optimizer='adam')
model.summary()

history = model.fit(x_train, y_train, batch_size=64, epochs=20, validation_data=(x_test, y_test))
```

- LSTM layer 2개와 dense layer 2개를 생성
- loss 함수로 mse를 사용
- **결과:** 과대적합 발생.. 훈련데이터의 loss는 0에 가까운 값에 수렴 했으나 테스트 데이터의 loss는 수렴 하지 않음, 예측값도 안 좋음
- **문제점:** 외부데이터를 사용하기 위해 데이터 기간을 조정하다 보니 데이터가 적어져 학습이 잘 안된 것으로 보임



# LSTM

## 다시 전처리

	날짜	총지수	단독주택	연립주택	아파트	아파트서울	총지수서울
0	1986-01-01	34.656	61.347	43.053	20.973	17.153	30.044
1	1986-02-01	34.656	61.347	42.925	20.973	17.183	30.044
2	1986-03-01	34.708	61.513	42.733	20.935	17.153	30.002
3	1986-04-01	34.449	61.097	42.733	20.819	17.034	29.837
4	1986-05-01	34.293	60.848	41.837	20.664	16.945	29.588
...	...	...	...	...	...	...	...
418	2020-11-01	107.125	104.334	106.398	107.506	114.853	112.139
419	2020-12-01	108.586	104.909	107.237	109.346	116.365	113.532
420	2021-01-01	109.878	105.303	107.895	111.005	118.230	114.974
421	2021-02-01	111.375	105.718	108.706	112.957	120.121	116.290
422	2021-03-01	112.842	106.095	109.425	114.911	121.721	117.411

423 rows × 7 columns

#정규화

```
data = df.loc[:, ['아파트서울']]
scalar = MinMaxScaler()
data_scaled = scalar.fit_transform(data)
```



아파트서울	
0	0.011243
1	0.011526
2	0.011243
3	0.010118
4	0.009276
...	...
418	0.935059
419	0.949356
420	0.966990
421	0.984871
422	1.000000

423 rows × 1 columns

- 그래서... 외부 데이터를 제거한 후 기간을 늘려 데이터를 추가 확보
- 앞에서와 마찬가지로 정규화 진행

# LSTM

## 다시 전처리

아파트서울

window size: 10

0	0.011243
1	0.011526
2	0.011243
3	0.010118
4	0.009276
...	...
418	0.935059
419	0.949356
420	0.966990
421	0.984871
422	1.000000

423 rows x 1 columns

```
#window 생성
data_list = []
target_list = []
past_set = 10

for i in range(past_set, len(data)):
    data_list.append(data.iloc[i-past_set:i,0])
    target_list.append(data.iloc[i,0])

data_ar, target_ar = np.array(data_list), np.array(target_list)
data_ar = np.reshape(data_ar, (data_ar.shape[0], data_ar.shape[1], 1))

print(np.shape(data_ar), np.shape(target_ar))
```

```
# train set, test set 분할
x_train, x_test, y_train, y_test = train_test_split(data_ar, target_ar, test_size=0.2, shuffle=False)
```

```
# train set, test set 분할 후 shape
x_train_shape = np.shape(x_train)
y_train_shape = np.shape(y_train)
x_test_shape = np.shape(x_test)
y_test_shape = np.shape(y_test)

print('x_train:', x_train_shape, 'y_train:', y_train_shape)
print('x_test:', x_test_shape, 'y_test:', y_test_shape)
```

```
x_train: (330, 10, 1) y_train: (330,)
x_test: (83, 10, 1) y_test: (83,)
```

- 사용 데이터 수가 많아졌기 때문에 예측에 사용하는 데이터의 기간을 5달 에서 10달로 늘림 -> 과거 10달 데이터로 다음 달을 예측
- 훈련 데이터, 테스트 데이터 분할은 동일하게 8:2

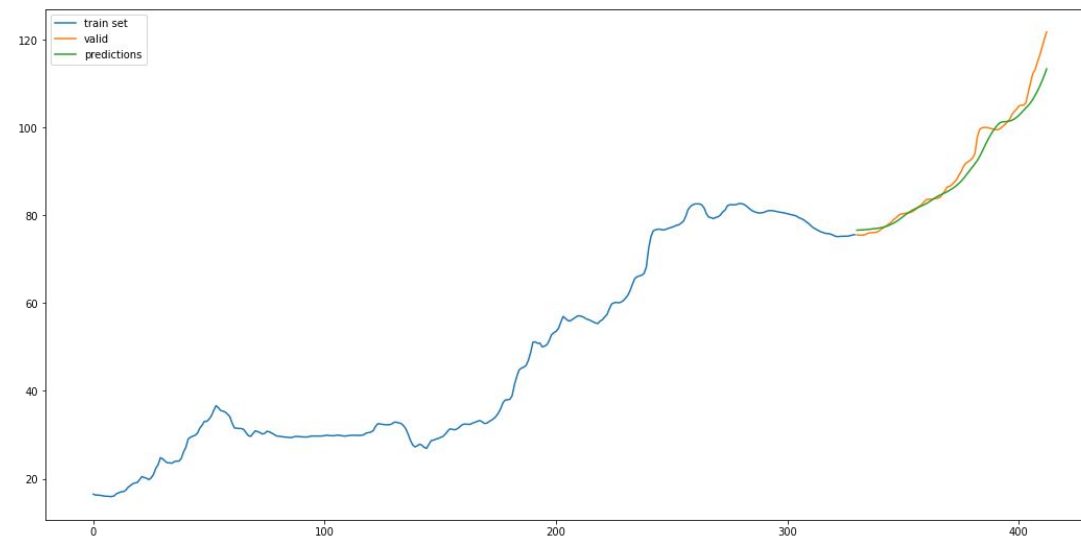
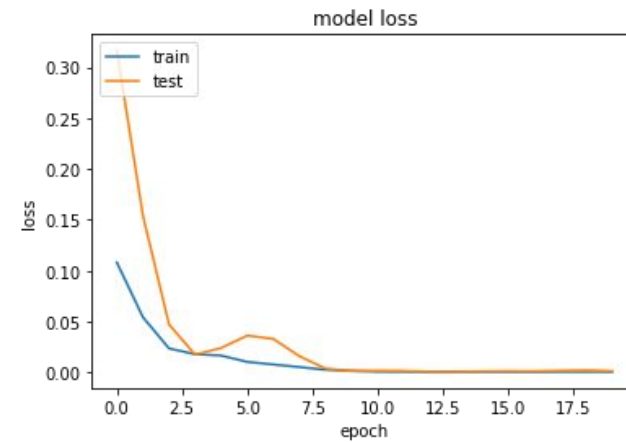
# LSTM

## 다시 모델링

```
#모델 생성
model = Sequential()
model.add(LSTM(10,return_sequences=True, input_shape = (x_train.shape[1],1)))
model.add(LSTM(20,return_sequences=False))
model.add(Dense(10))
model.add(Dense(1))
model.compile(loss='mse', optimizer='adam')
model.summary()

history = model.fit(x_train, y_train, batch_size=64, epochs=20, validation_data=(x_test, y_test))
```

- 모델도 앞에서와 마찬가지로 동일하지만, input 데이터에 맞게 은닉층 개수 조정
- **결과:** train set의 loss와 test set의 loss 모두 0에 가까운 값으로 수렴, 예측 성능 또한 좋음 변수를 하나만 사용하더라도 데이터를 많이 확보한 모델이 성능이 더 좋았다.



# 느낀점

## 가영

원하는 공공데이터를 선정하는 것이 얼마나 어려운지 다시 한번 체감해볼 수 있었지만.. 팀원들과 회의할때 너무 재밌어서 재밌게 분석해볼 수 있었다! 다음 주제는 이번 프로젝트를 데이터를 먼저 살펴보고 골라야지.. 🙄

## 준희

힘들게 외부데이터 찾아서 했는데 성능이 잘 안나와서 허무했고, 이번 프로젝트를 통해 데이터의 중요성을 알았다.

처음에 주제 정하느라 우여곡절도 많았고 힘들었지만 팀원들이 즐겁게 해줘서 고마웠다. 다들 고생하셨습니다. 다음엔 좋은 주제로 만나요..

## 형배

주제 선정하는데에 여러번 엮어져서 어려움이 많았지만 주제 다시 잘 정해서 으쌔으쌔해서 여기까지 잘온거 같습니당~ 다음 주제는 빠르게 잘 선정해서 잘해봅시다 아좌좌 (\*´`o🌈o

## 은혜

처음 주제 선정에 있어서 많은 어려움이 있었지만, 이렇게 팀원들끼리 임의로 주제를 정해서 예측해보는 것도 재밌었다! 항상 정해진 규칙에 맞춰 프로젝트를 했었지만 이렇게 팀원들끼리 데이터를 보면서 이거 해볼까?! 저거 해볼까?! 하면서 프로젝트를 하니까 색다르고 즐겁게 한 것 같다!!!! 두 번째 프로젝트는 주제 선정 잘해봅시다 🤘🤘