# 지하철역 개통과 집값의 상관관계 :
## 지하철이 개통되면 아파트 값이 오를까?

컴퓨터 과학과 20123010440 서교영

# 1. 주제 선정

# 주제 선정

강남 잇는 전철 개통한 일부 지역 매매가, 1년새 20% 올라

'숙원사업' 김포도시철도 개통, 집값 상승 이끌까

이미연

광주 지하철 2호선 첫 삽...광주 부동산시장 '부싯돌' 되나

17년간 표류하던 광주 도시철도 2호선 착공...인근 분양 단지 청약 광풍

최은서 기자 | 승인 2019.09.30 14

01 | 유진의 기자 | joy0536@naver.com

5호선 하남시청역 개통에 '하남 신축아파트' 분양 상승세

수도권 광역철도 개통 호재… 집값도 뻥 뚫었다!

'골드라인' 개통에 바닥 친 김포 집값… "급등하긴 어려워"

조선비즈 | 유한빛 기자

철도망 개통 호재...착공 이후 집값 '고공행진'

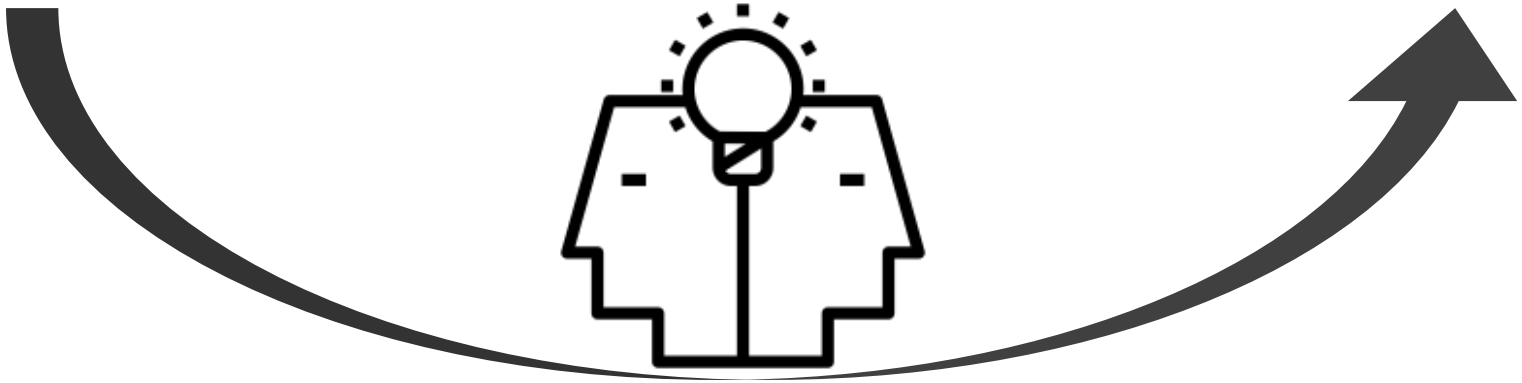
지하철 5·6호선 개통 앞두고 하남·신내동 아파트값 '들썩'

등록 2019-12-04 오전 10:48:11
수정 2019-12-04 오전 10:48:11

남빛하늘 기자 | 승인 2019.12.14 06:15

# 주제 선정

지하철이 개통하면
아파트 값이 오를까?



# 주제 선정

## 우이신설선

개통일: 2017년 9월

## 신분당선(정자-광교)


개통일: 2016년 1월

## 9호선(개화-신논현)


개통일: 2009년 7월

# 2. 데이터 수집

# 데이터 수집

## ◆ 모집 대상

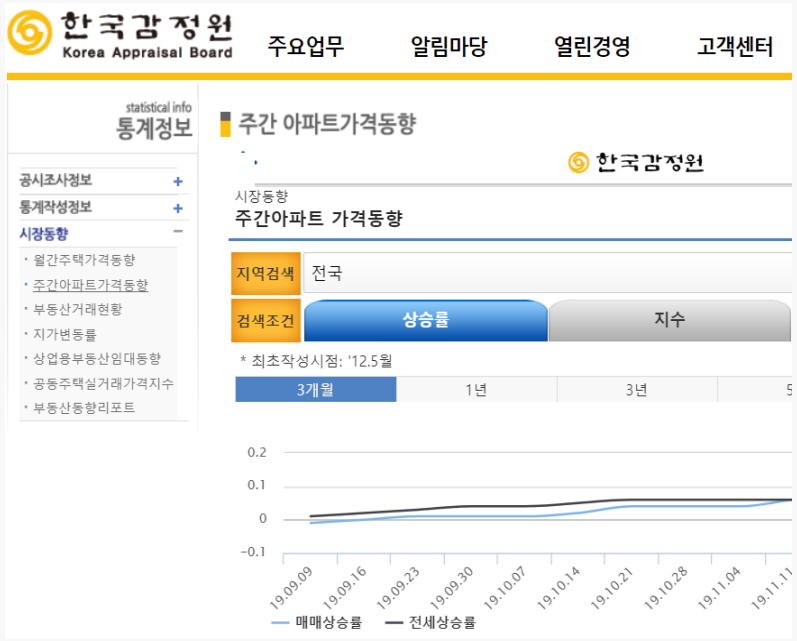
우이신설선 라인 역세권 아파트 4837세대

신분당선 라인 역세권 아파트 7107세대

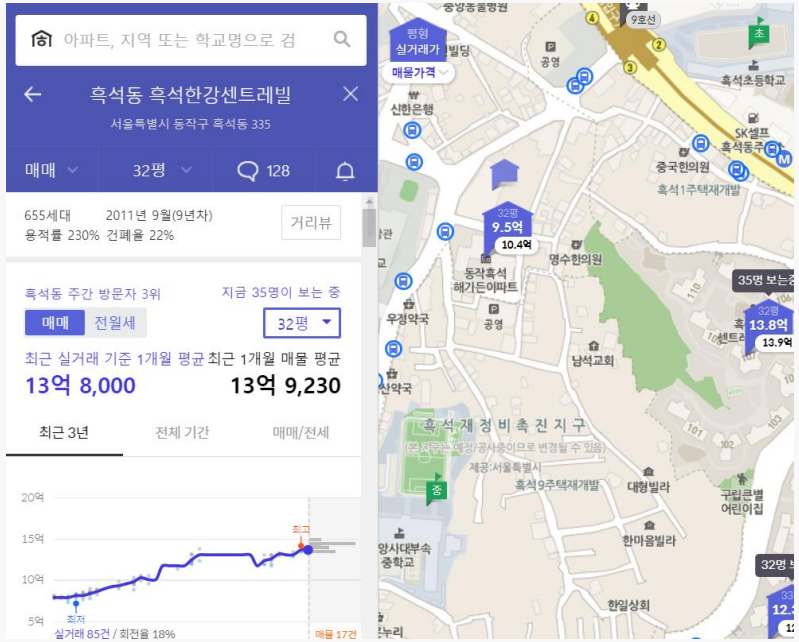
9호선 라인 역세권 아파트 3286세대

# 데이터 수집

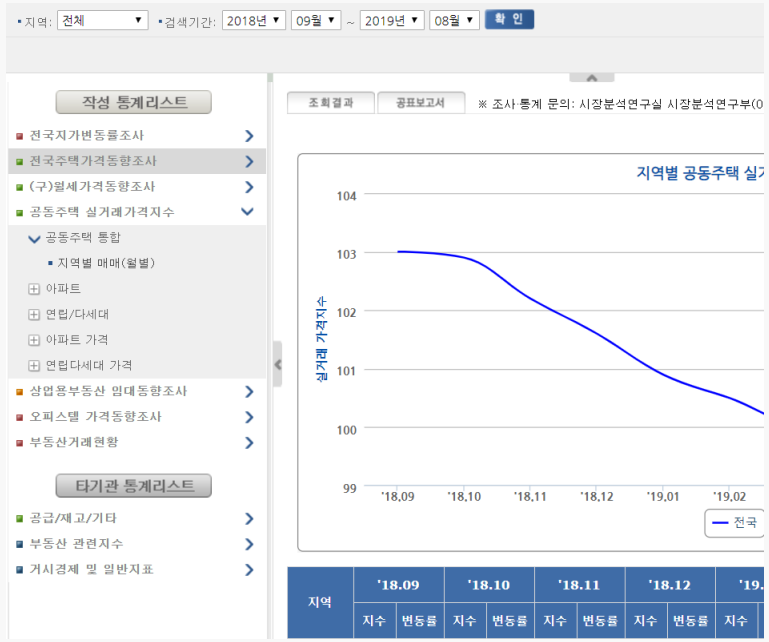
## 한국 감정원 사이트




## 호갱 노노




## R-ONE



# 데이터 수집

## ◆ R-one 사이트

셀레니움을 사용

```
#매매가격지수 클릭
wd.find_element_by_xpath("""//*[@id="HOUSE_21210"]/a""").click()
wd.implicitly_wait(5)
time.sleep(1)
wd.find_element_by_xpath("""//*[@id="HOUSE_21211"]/a""").click()
wd.implicitly_wait(12)
time.sleep(1)
```

```
#year
if(idx != 0):
    #year
    date[1] = date[1]+1
    if(date[1] == 13):
        date[0] = date[0]+1
        date[1] = 1
print(date)
```

```
# 2004 or 2019를 만들기 위해
if(len(str(date[0])) == 1):
    year_month = '200'+str(date[0])+" "+str(date[1])
else:
    year_month = '20'+str(date[0])+" "+str(date[1])
```

```
result.append([city]+ [year_month]+ [value])
```

| | A | B |
|---|---|---|
| 1 | date | value |
| 2 | 2006 2 | 0.84 |
| 3 | 2006 3 | 1.03 |
| 4 | 2006 4 | 1.49 |
| 5 | 2006 5 | 2.29 |
| 6 | 2006 6 | 1.28 |
| 7 | 2006 7 | 0.59 |
| 8 | 2006 8 | 0.57 |
| 9 | 2006 9 | 1.19 |
| 10 | 2006 10 | 3.3 |
| 11 | 2006 11 | 8.05 |
| 12 | 2006 12 | 4.16 |
| 13 | 2007 1 | 2.33 |
| 14 | 2007 2 | 0.9 |
| 15 | 2007 3 | 0.5 |
| 16 | 2007 4 | 0.26 |
| 17 | 2007 5 | -0.05 |
| 18 | 2007 6 | 0.16 |
| 19 | 2007 7 | 0.39 |

# 데이터 수집

## ◆ 호갱노노 사이트

셀레니움을 사용

```
#회원 로그인
wd.find_element_by_xpath("""//*[@id="container"]/div[3]/a""").send_keys(Keys.ENTER)
wd.find_element_by_name("username").send_keys('01063955862')
wd.find_element_by_name("password").send_keys('446602')
#클릭
wd.find_element_by_xpath("""/html/body/div[2]/div/div[2]/div[2]/div/div/div[2]/div[2]/a[1]""").send_keys(Keys.ENTER)
wd.implicitly_wait(10)
time.sleep(1)

#실거래가 스크롤 끝까지 내리기
for idx in count():
    try:
        wd.find_element_by_xpath("""/html/body/div[2]/div/div[1]/div[1]/div[3]/div/div[4]/div/div/div/div[1
    except:
        break
for i in count():
    try:
        tds = trs[i].find_elements_by_css_selector('td')
        result.append([tds[0].text] + [tds[1].text] + [tds[2].text])
    except:
        break
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | 0 | 계약일 | 가격 | 타입 |
| 2 | 1 | 2019.10.12 | 130000 | 109타입 |
| 3 | 2 | 2019.10.12 | 128000 | 109타입 |
| 4 | 3 | 2019.08.24 | 120000 | 109타입 |
| 5 | 4 | 2019.08.01 | 121000 | 109타입 |
| 6 | 5 | 2019.07.15 | 126000 | 109타입 |
| 7 | 6 | 2019.07.13 | 124000 | 109타입 |
| 8 | 7 | 2019.07.03 | 120500 | 109타입 |
| 9 | 8 | 2019.07.02 | 114500 | 109타입 |
| 10 | 9 | 2019.06.28 | 117000 | 109타입 |
| 11 | 10 | 2019.05.25 | 121000 | 109타입 |
| 12 | 11 | 2019.05.11 | 114600 | 109타입 |
| 13 | 12 | 2019.04.10 | 111000 | 109타입 |

# 3. 데이터 전처리

# 데이터 전처리 1단계

## ◆ 각 아파트별 csv안 한글 제거 후 숫자로 변환

### 크롤링한 데이터 편집

- 호갱노노 사이트에선 '억' 한글이 숫자를 대체하고 있기 때문에 나중에 평균값과 변동률 계산을 위해선 무조건 숫자로 변환해야 함.

```
for i in range(0,df.shape[0]):

    value = df['가격'][i].replace('억','').split()

    try:
      value[1] = value[1].replace(',','')

      if(len(value[1])==4):
        value = value[0]+value[1]
      elif(len(value[1])==3):
        value = value[0]+'0'+value[1]
      elif(len(value[1])==2):
        value = value[0]+'00'+value[1]
      elif(len(value[1])==1):
        value = value[0]+'000'+value[1]
    except:
      value = value[0]+'0000'

    df['가격'][i] = value
```

| A | B | C | D |
|---|---|---|---|
| 0 | 계약일 | 가격 | 타입 |
| 1 | 2019.11.2 | 10억 | 110L |
| 2 | 2019.11.2 | 11억 5,000 | 111H |
| 3 | 2019.11.2 | 11억 | 110E |
| 4 | 2019.11.2 | 11억 5,000 | 111G |
| 5 | 2019.11.2 | 11억 4,000 | 110C |
| 6 | 2019.11.0 | 10억 4,000 | 110B |
| 7 | 2019.11.0 | 9억 5,000 | 109A |
| 8 | 2019.11.0 | 10억 5,000 | 111G |
| 9 | 2019.11.0 | 11억 | 111F |
| 10 | 2019.10.3 | 10억 5,500 | 111G |
| 11 | 2019.10.3 | 11억 | 110C |
| 12 | 2019.10.3 | 11억 | 110B |
| 13 | 2019.10.2 | 9억 9,500 | 111H |
| 14 | 2019.10.2 | 9억 3,000 | 110L |
| 15 | 2019.10.2 | 11억 | 111G |

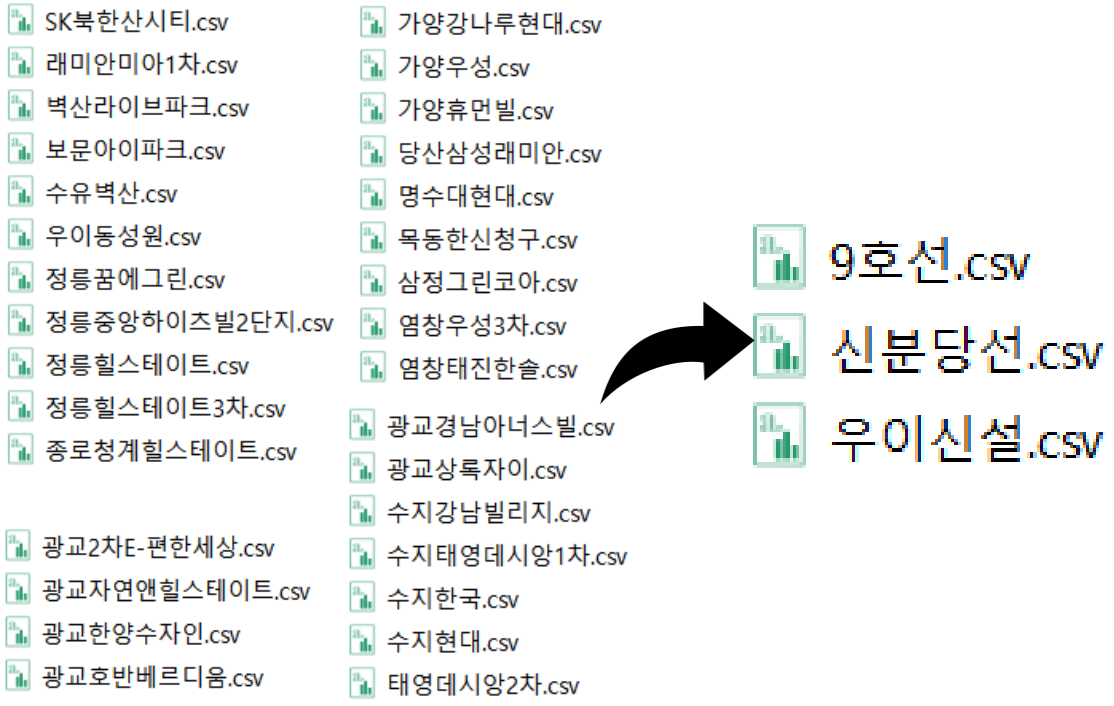
| A | B | C | D |
|---|---|---|---|
| 0 | 계약일 | 가격 | 타입 |
| 1 | 2019.11.2 | 100000 | 110L |
| 2 | 2019.11.2 | 115000 | 111H |
| 3 | 2019.11.2 | 110000 | 110E |
| 4 | 2019.11.2 | 115000 | 111G |
| 5 | 2019.11.2 | 114000 | 110C |
| 6 | 2019.11.0 | 104000 | 110B |
| 7 | 2019.11.0 | 95000 | 109A |
| 8 | 2019.11.0 | 105000 | 111G |
| 9 | 2019.11.0 | 110000 | 111F |
| 10 | 2019.10.3 | 105500 | 111G |
| 11 | 2019.10.3 | 110000 | 110C |
| 12 | 2019.10.3 | 110000 | 110B |
| 13 | 2019.10.2 | 99500 | 111H |
| 14 | 2019.10.2 | 93000 | 110L |
| 15 | 2019.10.2 | 110000 | 111G |

# 데이터 전처리 2단계

## ◆ 각 아파트별 csv를 노선별 csv로 합치기

### 크롤링한 데이터 모으기

- 같은 노선에 있는 아파트들을 한 개의 csv파일로 합치기
- 들어간 데이터 값의 오류가 없는지 확인하기 위해서 pandas.read_csv()를 이용해 파일 내용을 가져오고 읽어온 파일들 합쳐서 pandas.DataFrame() 과 to_csv()를 사용하여 저장함



# 데이터 전처리 3단계_1

## ◆ 평균값 구하기1

### 크롤링한 데이터 평균

- 계약일이 아파트별로 흩어져 있기 때문에
  3중 for문을 돌면서 해당 월에 맞는 칼럼을 찾아
  result에 저장함

```
for j in range(2006, 2020):
        for i in range(0, df.shape[0]):
            date = df['계약일'][i].split('.')
            if(date[0] == str(j)):
                for k in range(1, 13):
                    if(int(date[1]) == k):
                        result[k] = result[k]+int(df['가격'][i])
                        count[k] = count[k] + int('1')
```

```
#print(df['가격'][i])
result = {}
count = {}
for i in range(1,13):
    result[i] = 0
for i in range(1,13):
    count[i] = 0

for j in range(2006, 2020):
    for i in range(0, df.shape[0]):
        date = df['계약일'][i].split('.')
        if(date[0] == str(j)):
            for k in range(1, 13):
                if(int(date[1]) == k):
                    result[k] = result[k]+int(df['가격'][i])
                    count[k] = count[k] + int('1')
    print(count)
    for i in range(1, 13):
        if(count[i] != 0):
            result[i] = result[i]//count[i]
        new_date = str(j)+' '+str(i)
        res.append([new_date] + [result[i]])
    print(f"%s년===" %j, result)
    for i in range(1,13):
        result[i] = 0
    for i in range(1,13):
        count[i] = 0
```

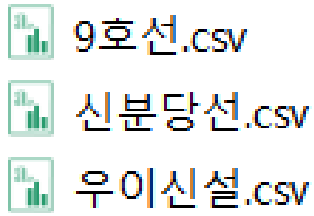
# 데이터 전처리 3단계_2

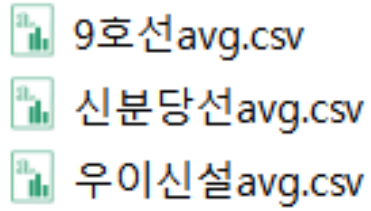
## ◆ 평균값 구하기2

### 크롤링한 데이터 평균

- 각 월마다 평균 가격을 구해야 하는데 해당 월에 계약건수를 알아야 하므로 count[]를 사용해 총 개수를 저장함 count[]를 이용해서 평균을 구함

```
for i in range(1, 13):
            if(count[i] != 0):
                result[i] = result[i]//count[i]
            new_date = str(j)+' '+str(i)
            res.append([new_date] + [result[i]])
        print(f"%s년==" %j, result)
        for i in range(1,13):
            result[i] = 0
        for i in range(1,13):
            count[i] = 0
```




| 계약일 | 가격 | 타입 |
|---|---|---|
| 2019.10.31 | 48000 | 109타입 |
| 2019.10.28 | 52000 | 109타입 |
| 2019.10.14 | 48500 | 109타입 |
| 2019.10.14 | 50000 | 109타입 |
| 2019.10.11 | 52900 | 109타입 |
| 2019.09.26 | 51800 | 111C |
| 2019.09.24 | 53500 | 111C |
| 2019.09.16 | 53500 | 111C |
| 2019.09.04 | 52000 | 111B |
| 2019.09.02 | 47000 | 109타입 |
| 2019.08.28 | 52500 | 109타입 |
| 2019.08.16 | 51600 | 109타입 |
| 2019.08.10 | 51500 | 109타입 |
| 2019.08.08 | 52000 | 109타입 |

| date | average |
|---|---|
| 2006 1 | 20175 |
| 2006 2 | 23950 |
| 2006 3 | 24197 |
| 2006 4 | 24512 |
| 2006 5 | 21985 |
| 2006 6 | 23153 |
| 2006 7 | 26109 |
| 2006 8 | 23643 |
| 2006 9 | 24636 |
| 2006 10 | 27848 |
| 2006 11 | 31402 |
| 2006 12 | 31913 |
| 2007 1 | 32181 |
| 2007 2 | 34137 |
| 2007 3 | 32765 |

# 데이터 전처리 4단계

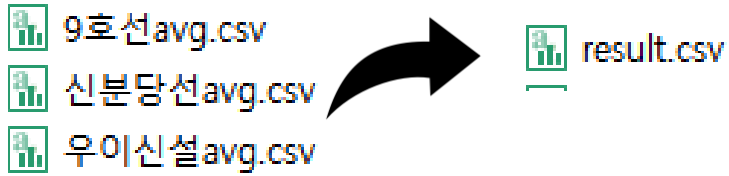
## ◆ 변동률 구하기

평균값들을 월별 변동률로 계산

- 해당 노선의 평균값들이 저장되어 있는 csv를 불러온 뒤 for문을 통해 앞뒤로 붙어있는 값들의 변동률을 구함

* 다만 값이 0일 경우 에러가 나지 않도록 if문을 설계함

```
for i in range(0, (df.shape[0]-1)):
        if(df['average'][i] != 0 and df['average'][i+1] != 0):
            print('==%s' %(df['date'][i+1]))
            value = round((df['average'][i+1] -
df['average'][i])/df['average'][i],2)
            print(value)
            date = df['date'][i+1]
            res.append([date] + [value])
        else:
            value = 0
            date = df['date'][i+1]
            res.append([date] + [value])
```




| | date | average |
|---|---|---|
| 0 | 2006 1 | 28327 |
| 1 | 2006 2 | 30459 |
| 2 | 2006 3 | 32674 |
| 3 | 2006 4 | 37768 |
| 4 | 2006 5 | 39340 |
| 5 | 2006 6 | 28728 |
| 6 | 2006 7 | 30704 |
| 7 | 2006 8 | 36796 |
| 8 | 2006 9 | 36066 |
| 9 | 2006 10 | 41037 |
| 10 | 2006 11 | 0 |
| 11 | 2006 12 | 46330 |
| 12 | 2007 1 | 45900 |
| 13 | 2007 2 | 44875 |
| 14 | 2007 3 | 0 |
| 15 | 2007 4 | 46750 |
| 16 | 2007 5 | 40125 |

| | date | value |
|---|---|---|
| 0 | 2006 2 | 0.08 |
| 1 | 2006 3 | 0.07 |
| 2 | 2006 4 | 0.16 |
| 3 | 2006 5 | 0.04 |
| 4 | 2006 6 | -0.27 |
| 5 | 2006 7 | 0.07 |
| 6 | 2006 8 | 0.2 |
| 7 | 2006 9 | -0.02 |
| 8 | 2006 10 | 0.14 |
| 9 | 2006 11 | 0 |
| 10 | 2006 12 | 0 |
| 11 | 2007 1 | -0.01 |
| 12 | 2007 2 | -0.02 |
| 13 | 2007 3 | 0 |
| 14 | 2007 4 | 0 |
| 15 | 2007 5 | -0.14 |
| 16 | 2007 6 | 0.03 |

# 4. 시각화 및 분석


USERS: LAST 7 DAYS USING MEDIAN
Bounce Rate (LUX)
40.6%
Sessions (LUX)
479K
Session Length (LUX)
17min
PVs Per Session (LUX)
2pvs

# 시각화

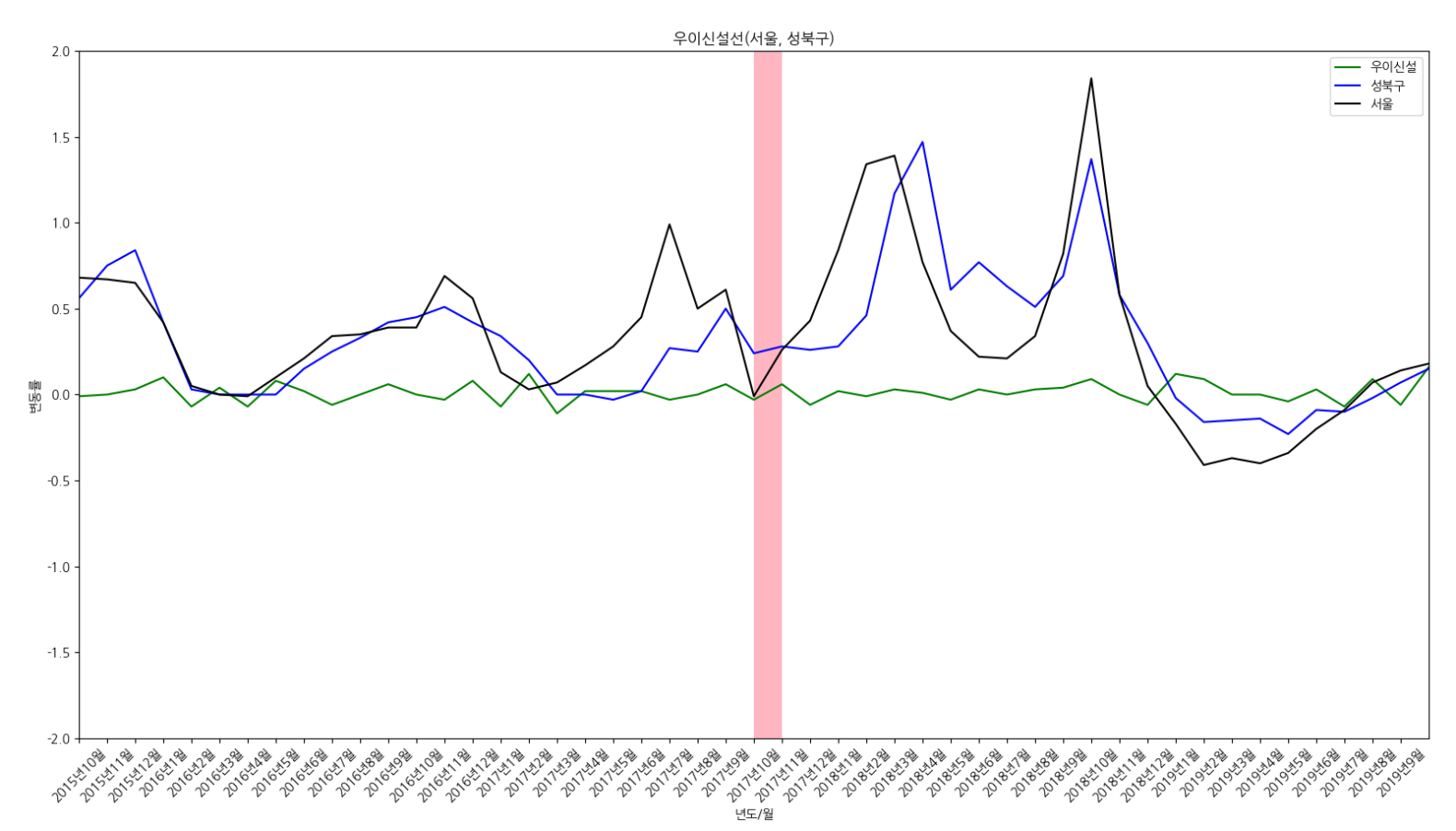
가설을 확인하고 더 나아가 지하철 개통 발표만으로도 집값에 영향이 있는지를 확인합니다.

1. 지하철 개통 전 후를 비교

2. 지하철 개통 확정 발표 시기를 전 후로 비교

# 시각화

## ◆ 우이신설선 - 서울, 성북구, 전국



# 시각화

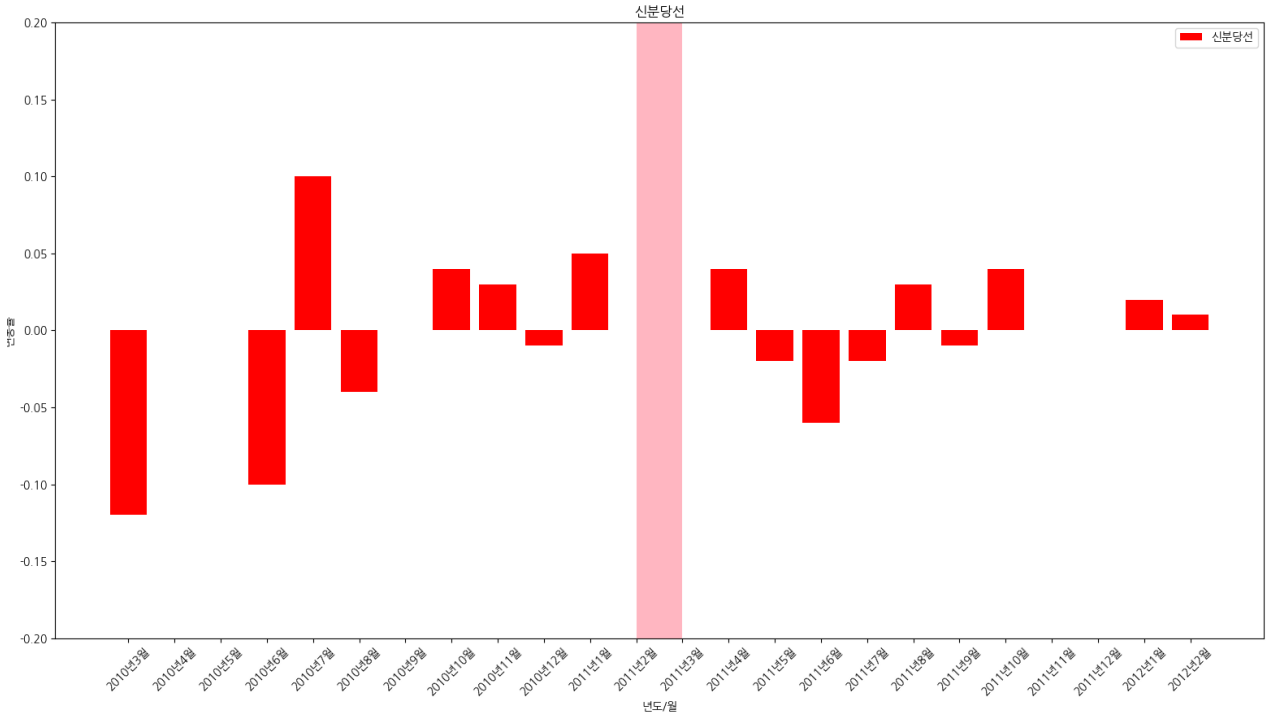
## ◆ 신분당선 - 수도권, 경기도, 전국

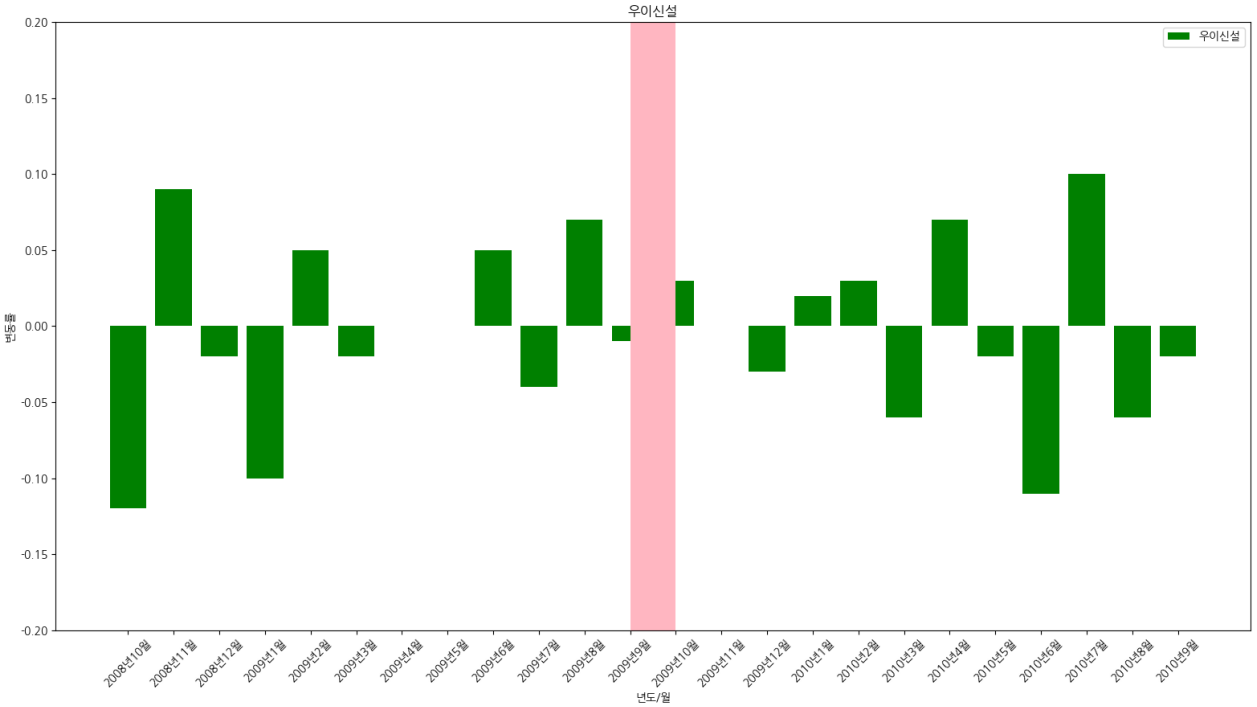
# 시각화

## ◆ 9호선 - 서울, 영등포구, 전국

# 시각화+

## ◆ 개통 확정 발표->공사 시작 후 변화 양상

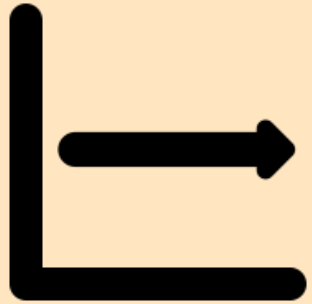


# 분석

## ◆ 결과 분석 - 지하철 개통이 집값 상승의 필수적인 요소는 아니다.

**역세권 아파트의 집값 안전성**



전체 지역의 변동률과는 달리 비교적 안정적인 변동률을 보여줌.

**집 값에 영향을 주는 여러 요인**


학군, 신축 아파트, 상권

# 5. 어려웠던 문제점

# 논의

1. 감정원사이트처럼 태그 접근이 차단된 사이트들이 존재한다.

-대체 사이트를 찾아야 함.

2. 날짜와 같은 크롤링한 데이터가 요소마다 형식이 다르다.

-따라서 정규화 된 형식으로 맞추는 전처리 과정이 소요.

3. 날짜를 csv에 저장할 때 중간에 .이나 /를 넣으면 값이 변형된다.

-년도와 월 사이에 기호없이 띄어 쓰기만을 삽입.

4. 그래프로 시각화 하는 과정에서 음수 기호나 한글이 깨지는 상황.

-matplotlib.rcParams['axes.unicode_minus ]을 설정해야 함.

5. 데이터 모델의 선정기준을 정하기가 애매하고 어려움.

-정해진 형식이 아닌 직접 구상해야 하는 것이기 때문에 정답은 없다.

# Thank you