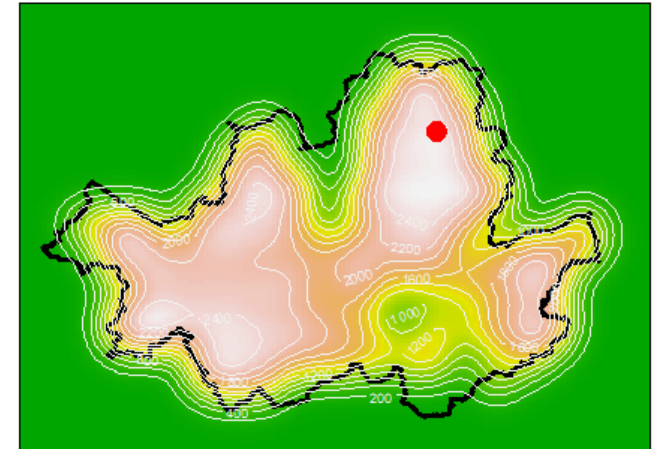
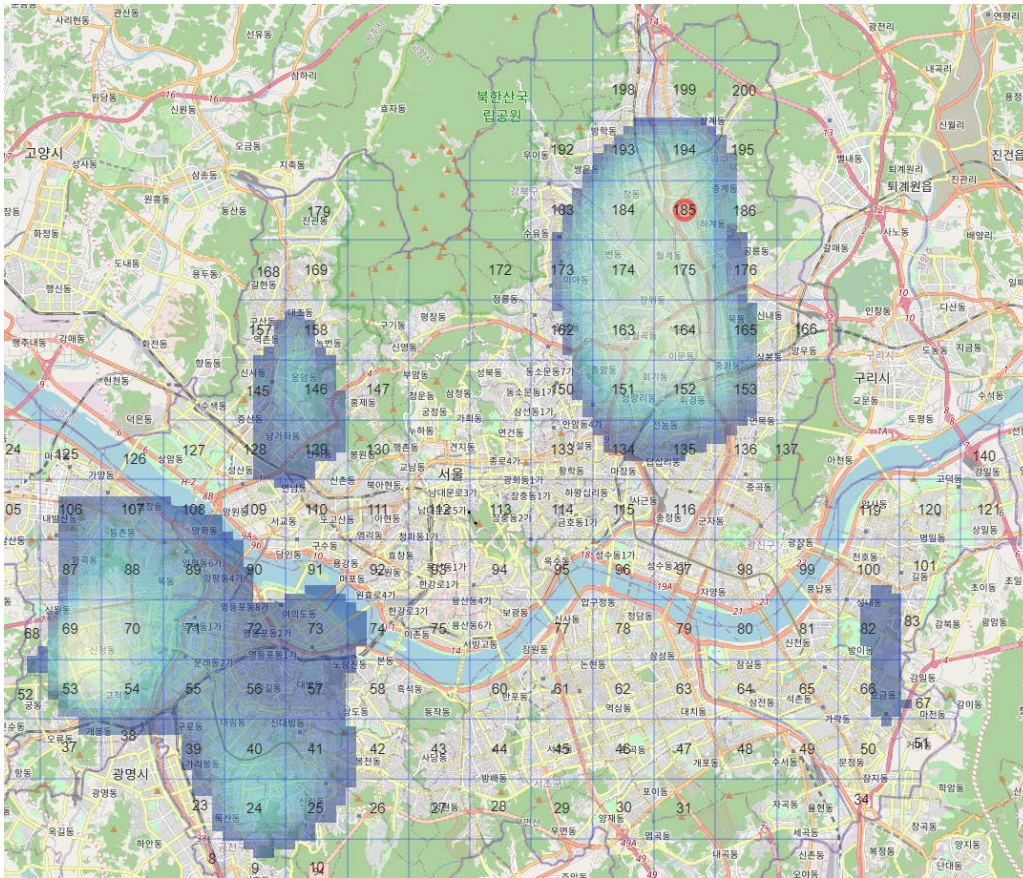


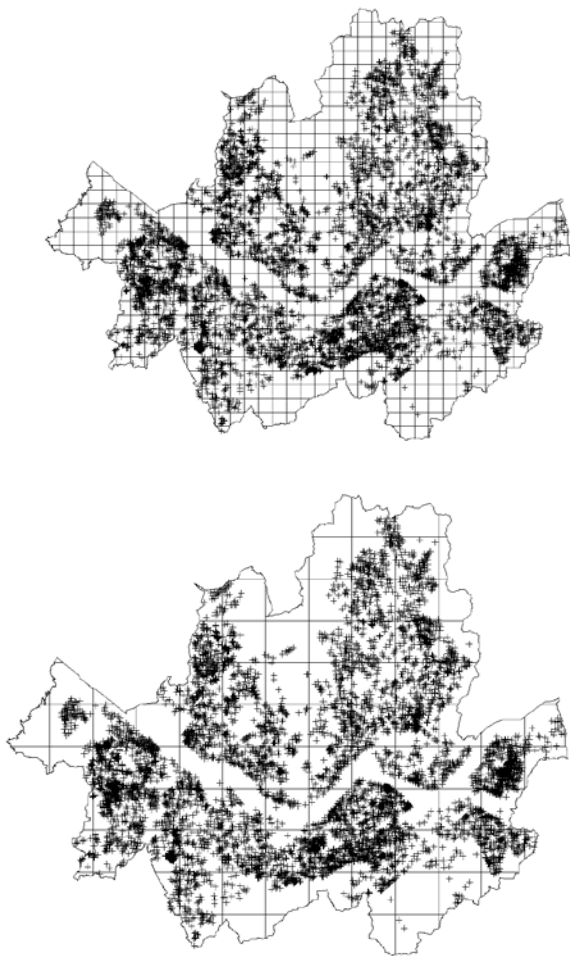
시공간 특성을 고려한 주택하위시장 분석

플랫폼 구축과 활용



본 발표 자료는 2019년 SH 도시연구원이 공모한
「신진연구자 지원사업」 결과보고서를 기반으로 작성되었음

2021.1.
강원대 부동산학과 시간강사 김철민



- 목차 -

I. 서론

II. 현황분석

III. 자료수집·분석방법론

IV. 실증분석

V. 결론

I. 서론

□ 연구배경

- 불안정한 부동산 시장과 정부의 핀셋규제
 - 최근 강력한 수요 억제 정책에도 불구하고 부동산 시장의 **변동성 및 불확실성 증가**
 - 정부: 서울의 부동산 가격 불안정성이 지속된다면 **핀셋규제**를 '동 단위'까지 확대 검토

- 핀셋규제에 대한 언론의 부정적 반응

'핀셋 규제'로 부동산 잡겠다는 정부

한국경제 2019.10.16

초고소득층 '핀셋 과세'로
집권 초 재미 본 文정부
이번엔 분양가 상한제에 '핀셋'

쉴림 심한 주택 시장에서
정밀타격 방식 안 통해
장기 안정대책 마련해야



중앙일보 2020.2.20

부동산거래 규제지역 지정 현황

● 투기지역(16곳) ● 투기과열지구(31곳)
○ 조정대상지역(44곳)

정부 19번째 부동산 대책 발표
경기 남부 **풍선 효과**에 핀셋 규제
중저가 아파트 대출 규제도 강화



I. 서론

□ 연구 필요성

- 부동산 하위시장에 대한 정밀분석 필요

- 부동산 규제정책은 특정 지역에 국한되어 살펴볼 문제는 아님
- 주택시장 → 다양·복잡화 → 공간적 특성을 보다 **정확 분석 & 적절 정책** 도입 중요

- 공간적 특성을 고려한 정책 수립

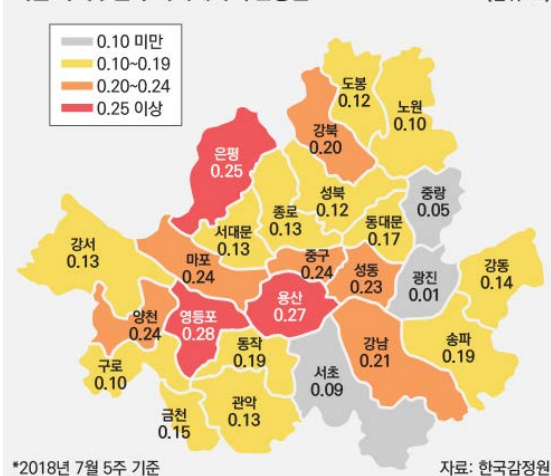
- 주택시장의 공간적 특성을 고려한 정책 수립 과정에서 가장 기초가 되는 것은 부동산 하위시장(sub-market)을 식별하는 것임 → **많은 분석의 최소 분석단위가 [시군구]**

민간 및 공공기관
분석 단위 → 구

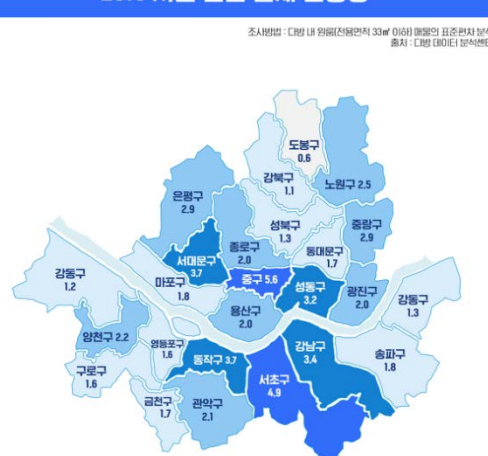


정밀 하위시장
분석에 한계

서울 자치구별 주택매매가격 변동률 [단위: %]



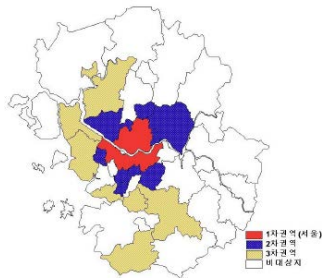
2019 서울 원룸 월세 변동성



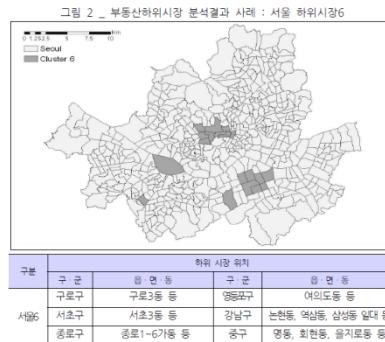
I. 서론

□ 주택하위시장 관련 주요 선행연구

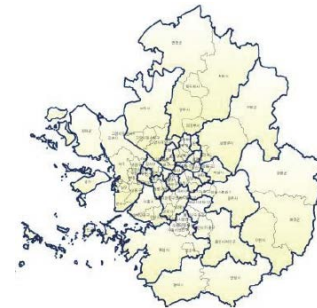
- 주택하위시장 관련 → 다양한 관련 연구들이 수행되어 왔음
- 김주영·우경(2004): 가격 및 크기 등을 요인/군집분석으로 하위시장 구분
- 박천규 외(2011): 주택특성 변수 기반 k-mean 클러스터링 활용 하위시장 구분
- 김태환·박태원(2015): 가구 이주범위를 intra-max 기법으로 분석하여 하위시장 구분



김주영·우경(2004)



박천규 외(2011)



김태환·박태원(2015)

- 연구들이 **거시적 관점**(시·군·구단위) 접근 ⇒ **정밀성 부족**
- **특정시점**을 기준으로 지역 간 특성 구분 ⇒ **역동성 고려 부족**

I. 서론

□ 주택하위시장 연구에서 극복되어야 할 핵심 문제들

- 정밀분석 위하여 기존연구에서 극복되어야 할 문제점
 - **분석단위의 결정**: 동질 또는 이질 집단의 기본단위(unit) 설정문제
 - **시계열 변화 고려**: 집합적 분석의 한계를 극복하여 역동적 변화 고려 문제

□ 연구목적

- 최적 분석단위 설정 + 시간적 특성 고려한 공간적 유사성 지도 구축
 - ⇒ (1) 서울시 주택 하위시장(전세 APT) 분석 **시각화 지도 구축**
- 반응형 웹기반 애플리케이션 구축
 - ⇒ (2) 하위시장(공간적 유사성) **지도 웹기반 어플 구축** → 정책결정 지원도구

I. 서론

□ 연구의 흐름

- (1) 이론적 고찰: 분석 방법론

- 최적분석단위(공간자기상관관계) / 시계열 유사도(DTW 분석)

- (2) 현황분석: 2011-2019 서울시 전세 아파트 특성 분석

- 지역별·규모별 특성 분석 / 3.3m²당 보증금 변화 분석

- (3) 자료수집과 분석과정

- 공공빅데이터 수집 및 전처리 / 최적 분석 공간단위 결정

- (4) 실증분석: 하위시장 분석 지도 작성 및 웹기반 플랫폼 구축

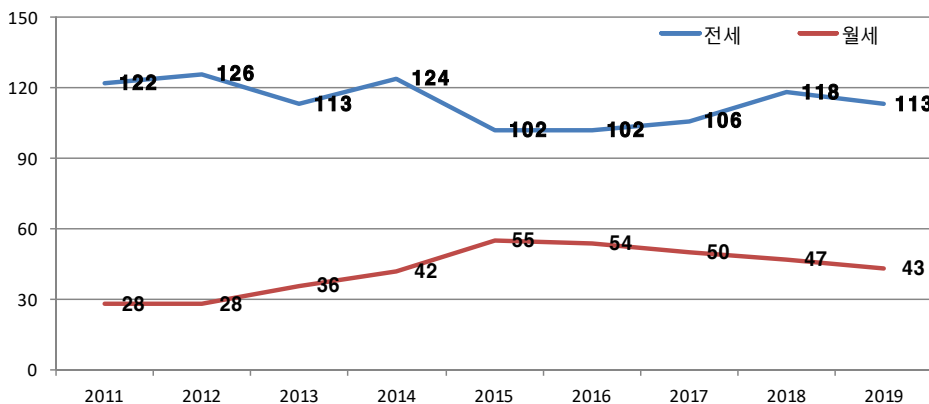
- 시계열 변화 패턴 기반 → 아파트 규모(소 · 중 · 대)별 전세주택 하위시장 분석
- 주택하위시장 웹기반 어플리케이션 구축 및 활용 방안 제시

II. 현황분석

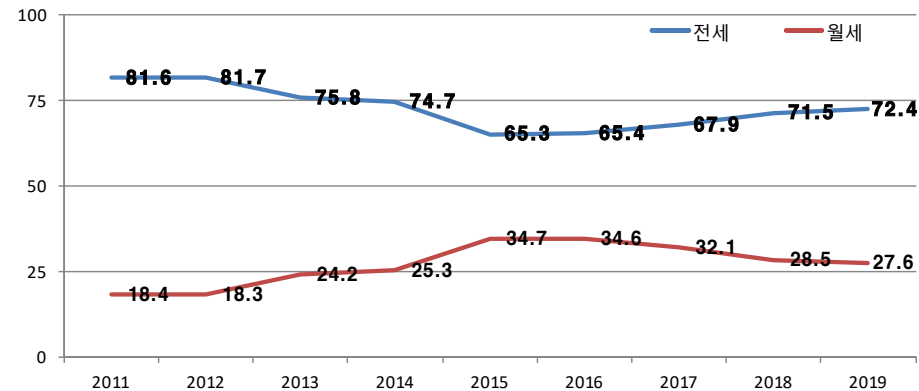
□ 서울시 임대차 아파트(전월세) 거래 현황

- 2011-2019 수집된 전월세 자료 → **140만 건**(전세 102만/월세 38만)
- 전세 → 연평균 10~12만 건 / 월세 → 연평균 2.8 ~ 5.5만 건
- 2015년 기점으로 **전세 거래건수 ↑** / **월세 거래건수 ↓**
- 2015년 까지 전세 거래비율 ↓ 하였지만 **최근에는 다시 회복세 보이고 있음**

전월세 거래건수 변화 (단위: 천 건)



전월세 거래비율 변화(단위: %)



II. 현황분석

□ 서울시 지역별 전세 아파트 평균보증금 및 거래 현황

- 지역에 따라 전세 APT 평균 보증금 및 거래량 차이를 보임

- 대형 → 강남구(평균 보증금 7.8억 원, 거래량 비율 14.8%)
- 중형 → 송파구(평균 보증금 4.4억 원, 거래량 비율 10.7%)
- 소형 → 강남구(평균 보증금 2.7억 원) But 노원구(거래량 비율 14.8%)

표 2-3 대형 아파트 지역별 전세거래 현황(2011~2019)

(단위: 만원, 건수, %)

대형	평균 보증금	건수	비율	대형	평균 보증금	건수	비율
강남구	78,295	29,406	14.8	서대문구	39,590	3,603	1.8
강동구	35,968	10,350	5.2	서초구	65,881	25,725	13.0
강북구	30,277	1,825	0.9	성동구	49,020	6,582	3.3
강서구	39,778	7,115	3.6	성북구	36,565	6,607	3.3
관악구	34,931	4,345	2.2	송파구	58,272	20,773	10.5
광진구	59,566	3,497	1.8	양천구	54,440	13,470	6.8
구로구	34,080	4,322	2.2	영등포구	47,071	9,343	4.7
금천구	25,797	1,762	0.9	용산구	58,233	8,431	4.2
노원구	35,991	6,607	3.3	은평구	53,423	5,787	2.9
도봉구	32,307	2,880	1.5	종로구	55,058	1,788	0.9
동대문구	32,930	5,107	2.6	중구	52,265	2,605	1.3
동작구	45,966	7,086	3.6	중랑구	32,698	2818	1.4
마포구	49,970	6,724	3.4	합계	53,054	198,558	

표 2-4 중형 아파트 지역별 전세거래 현황(2011~2019)

(단위: 만원, 건수, %)

대형	평균 보증금	건수	비율	대형	평균 보증금	건수	비율
강남구	49,672	36,157	8.1	서대문구	32,928	12,421	1.8
강동구	32,267	28,750	6.4	서초구	51,352	24,639	13.0
강북구	25,479	7,735	1.7	성동구	40,178	19,333	3.3
강서구	30,599	26,586	5.9	성북구	30,693	17,843	3.3
관악구	29,341	11,406	2.5	송파구	44,623	47,901	10.5
광진구	40,189	11,965	2.7	양천구	33,310	26,996	6.8
구로구	27,074	20,015	4.5	영등포구	31,899	18,855	4.7
금천구	24,293	5,366	1.2	용산구	39,343	8,055	4.2
노원구	26,684	24,106	5.4	은평구	25,888	16,431	2.9
도봉구	22,514	15,133	3.4	종로구	39,880	3,004	0.9
동대문구	31,095	12,316	2.7	중구	39,207	4,785	1.3
동작구	38,900	19,973	4.5	중랑구	25,714	10,159	1.4
마포구	39,690	18,617	4.2	합계	35,788	448,547	

표 2-5 소형 아파트 지역별 전세거래 현황(2011~2019)

(단위: 만원, 건수, %)

대형	평균 보증금	건수	비율	대형	평균 보증금	건수	비율
강남구	27,974	29,410	7.7	서대문구	24,526	8,690	2.3
강동구	21,265	22,497	5.9	서초구	35,136	12,666	3.3
강북구	18,844	5,704	1.5	성동구	32,766	14,889	3.9
강서구	20,223	29,765	7.8	성북구	24,667	16,289	4.3
관악구	22,477	10,548	2.8	송파구	30,155	26,051	6.9
광진구	29,653	5,712	1.5	양천구	21,380	18,481	4.9
구로구	18,889	18,942	5.0	영등포구	23,471	14,211	3.7
금천구	16,816	5,927	1.6	용산구	30,420	6,434	1.7
노원구	15,398	56,072	14.8	은평구	21,359	9,454	2.5
도봉구	14,455	13,951	3.7	종로구	24,671	2,026	0.5
동대문구	23,620	10,192	2.7	중구	28,182	3,838	1.0
동작구	27,991	10,981	2.9	중랑구	17,106	10,956	2.9
마포구	28,131	15,854	4.2	합계	23,025	379,540	

II. 현황분석

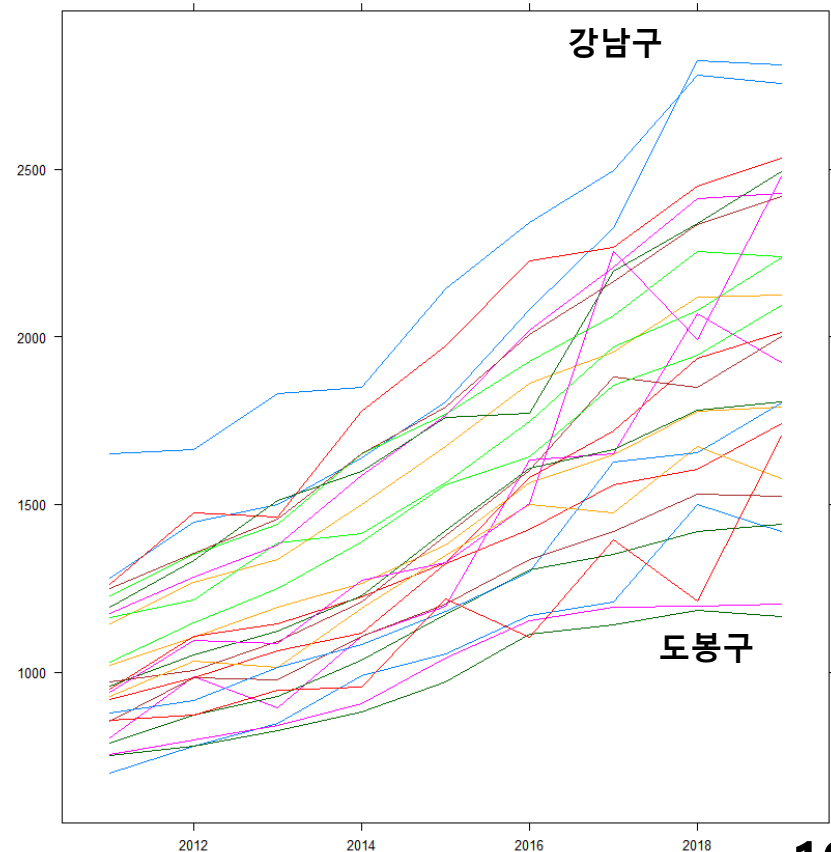
□ 서울시 지역별 전세 아파트 보증금액 변화(평당)

- 강남구 연평균 205만, 도봉구 연평균 64만 → 최대/최소비율 3.2배
- 서울시 전체의 전세보증금 증가는 125.14 만원
- 상승폭 최대/최소비율 지역 차이 3.2배

서울시 전세 아파트의 연단위 보증금액 증가 현황(2011-2019)

(단위: 만원)

지역	Estimate	P-value	지역	Estimate	P-value
강남구	205.6	1.69E-43	동작구	134.4	3.45E-25
종로구	190.0	1.31E-39	은평구	122.7	3.95E-22
성동구	175.0	9.25E-36	성북구	117.1	1.11E-20
송파구	167.7	7.11E-34	강서구	107.4	3.19E-18
강동구	162.9	1.26E-32	관악구	105.1	1.20E-17
중구	162.5	1.56E-32	금천구	99.1	3.86E-16
서초구	159.6	8.76E-32	양천구	95.9	2.25E-15
용산구	156.4	5.92E-31	중랑구	91.0	3.45E-14
서대문구	143.7	1.28E-27	구로구	90.5	4.39E-14
마포구	139.8	1.40E-26	강북구	89.5	7.86E-14
광진구	137.8	4.69E-26	노원구	65.8	1.29E-08
동대문구	137.3	6.21E-26	도봉구	62.4	5.98E-08
영등포구	135.3	2.05E-25			

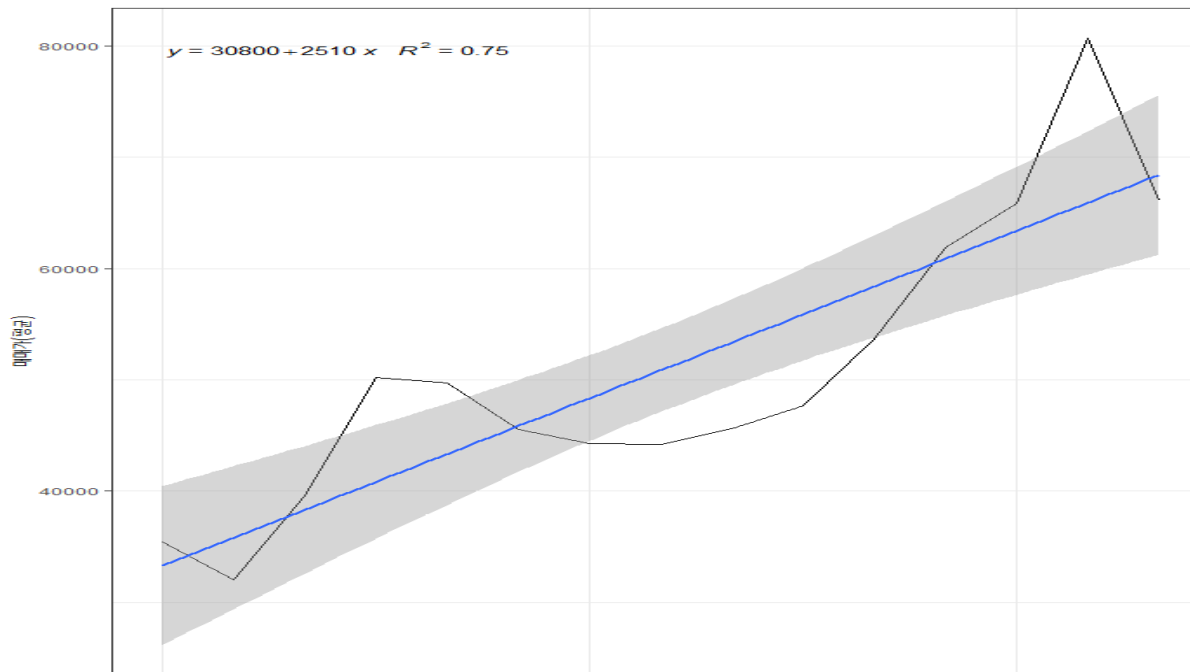


II. 현황분석

□ 참고 1: 서울시 아파트 평균 매매가 변화: 평당

- 연평균 전세 보증금 증가는 125만, 매매가 변화는 2,510만(20배)
- APT 매매거래가(06-19) : $y = 2,510x + 30,800$
- 전세 APT 보증금(11-19): $y = 125.1x - 250,723$

서울시 아파트 평균 매매가 변화(2006-2019)



II. 현황분석

□ 참고 2: 지역별 APT 매매가 / 보증금 비율 비교

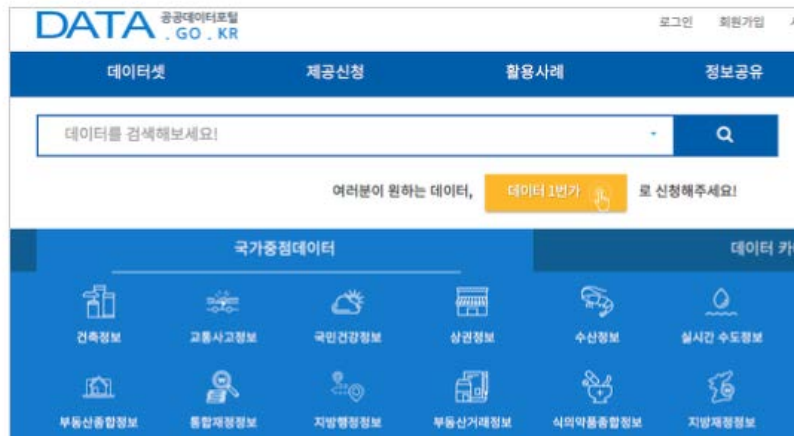
- 연평균 매매가 증가율이 높은 지역일수록 보증금 증가율도 높음
- 매매가/전세보증금 비율 높은 지역: 서초(35.6배) > 용산(33.4배) > 강남(31.8배)
- 매매가/전세보증금 비율 낮은 지역: 금천(14.3배) > 은평(14.6배) > 종로(13.7배)

지역	연평균 증가율			지역	연평균 증가율		
	전세 보증금	매매가	매매/전세		전세 보증금	매매가	매매/전세
강남구	205.6	6,541.5	31.8	동작구	134.4	2,978.7	22.2
종로구	190.0	2,610.5	13.7	은평구	122.7	1,788.4	14.6
성동구	175.0	3,712.5	21.2	성북구	117.1	2,020.5	17.3
송파구	167.7	3,632.3	21.7	강서구	107.4	2,357.1	21.9
강동구	162.9	2,149.6	13.2	관악구	105.1	1,597.2	15.2
중구	162.5	2,856.9	17.6	금천구	99.1	1,417	14.3
서초구	159.6	5,689.3	35.6	양천구	95.9	2,254.4	23.5
용산구	156.4	5,222.4	33.4	중랑구	91.0	1,435.1	15.8
서대문구	143.7	2,614.6	18.2	구로구	90.5	1,600.2	17.7
마포구	139.8	3,647.4	26.1	강북구	89.5	1,486.1	16.6
광진구	137.8	3,010.2	21.8	노원구	65.8	1,434.9	21.8
동대문구	137.3	1,999.2	14.6	도봉구	62.4	1,233.2	19.8
영등포구	135.3	2,429.1	18.0				

III. 자료수집과 분석과정

□ 자료수집

- 공공데이터 포털 API를 활용한 전월세 자료수집



- 수집된 자료에 포함된 변수 → 8가지 변수

- ① 거래날짜(연월일), ② 보증금액, ③ 월 임대료, ④ 아파트 주소(지역명 포함),
⑤ 층수, ⑥ 아파트 크기(m²) 그리고 ⑦ 건축연도 같은 데이터들이 포함됨



아파트 크기를 대형(85m² 이상), 중형(60-85m²), 소형(60m² 이하) 구분 → 3.3m² 당 보증금액 산출



google geocoding API 이용
APT 단지별 공간좌표 수집

III. 자료수집과 분석과정

□ 최적분석 단위 결정: MAUP 문제

- 공간분석 → 공간단위 설정에 따라 분석결과가 달라짐
 - 이를 MAUP(Modifiable Areal Unit Problem: 수정 가능한 공간단위 문제)라고 함
 - MAUP 해결책 → 공간자기상관 관계가 가장 높은 공간 단위로 분석 진행

□ 공간자기상관관계: Moran's I

- 공간자기 상관관계의 개념
 - 인접한 것들은 서로 관련 → 가까이 있는 것이 멀리 있는 것보다 관련성이 더 높음
 - 공간자기상관 대표적 측정 기법은 Moran's I → Global(전역) & Local(국지)로 구분

Global Moran's I

$$I = \frac{N \sum_{i=1}^n \sum_{j=1}^n w_{ij} (Y_i - \bar{Y})(Y_j - \bar{Y})}{(\sum_{i=1}^n \sum_{j=1}^n w_{ij}) \sum_{i=1}^n (Y_i - \bar{Y})^2}$$

Local Moran's I

$$I_i = \left[\frac{n^2}{\sum_{i=1}^n \sum_{j=1}^n w_{ij}} \right] \frac{(y_i - \bar{Y}) \sum_{j=1}^n w_{ij} (Y_j - \bar{Y})}{\sum_{i=1}^n (y_i - \bar{Y})^2}$$

$I > 0$: 자료의 공간적 분포가 Positive 자기상관을 가지고 있음

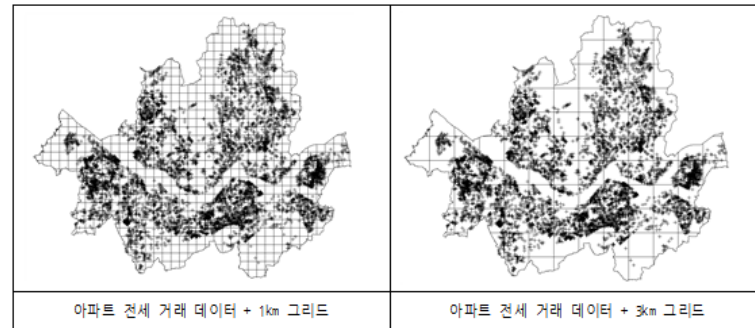
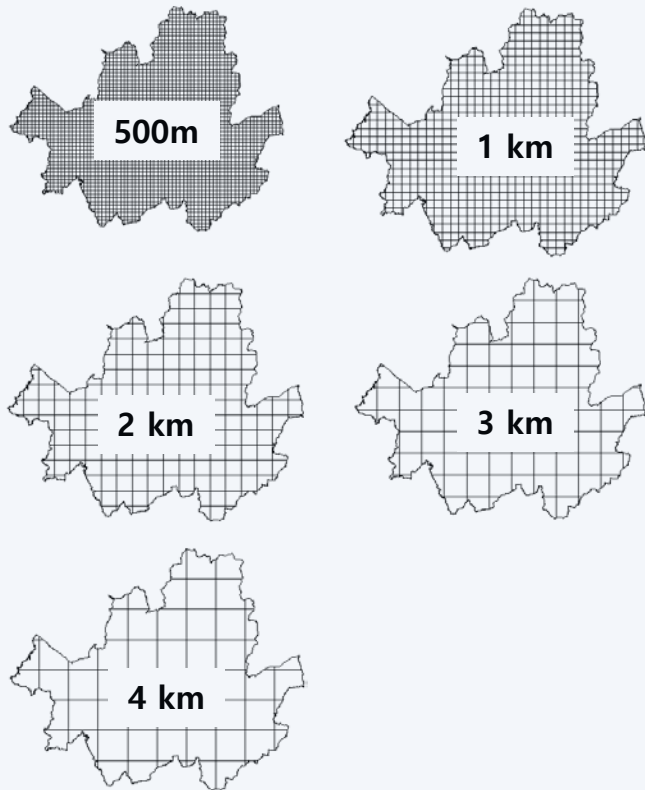
$I = 0$: 자료의 공간적 분포가 랜덤(독립적)임

$I < 0$: 자료의 공간적 분포가 Negative 자기상관을 가지고 있음

III. 자료수집과 분석과정

□ 전세거래 APT 데이터의 MAUP 문제 해결

- 후보군(그리드 500-4km)별로 공간 결합(Spatial Join) → 2km 선택



연도와 공간단위에 따른 공간자기상관지수(Global Moran's I) 변화

구분	500m	1 km	2 km	3 km	4 km	총합계
2011	0.409	0.475	0.520	0.380	0.312	0.419
2012	0.401	0.487	0.537	0.404	0.315	0.429
2013	0.409	0.478	0.512	0.397	0.320	0.423
2014	0.407	0.453	0.498	0.385	0.413	0.431
2015	0.403	0.450	0.483	0.374	0.409	0.424
2016	0.393	0.443	0.489	0.375	0.405	0.421
2017	0.401	0.444	0.499	0.400	0.425	0.434
2018	0.401	0.452	0.516	0.409	0.442	0.444
2019	0.400	0.447	0.520	0.418	0.417	0.440
총합계	0.403	0.459	0.508	0.394	0.384	0.429

※ 2km 단위에서 I 지수가 가장 높게 나옴

```
#-----  
# Chap 2-03 그리드 만들기  
#-----
```

```
# 00. 폴더 세팅
```

```
setwd(dirname(rstudioapi::getSourceEditorContext()$path))  
getwd()
```

```
# 01. 서울시 shape 지도 불러오기
```

```
library(sf)  
map <- st_read("./4_map/admin_map.shp") # shape 불러오기 => sf 포맷  
  
map$CTP_ENG_NM == "Seoul"  
map <- map %>%  
  filter(CTP_ENG_NM == 'Seoul') %>% # 서울 추출하기  
  st_transform("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0")  
  
plot(map$geometry)  
save(map, file="./4_map/seoul_map.Rdata")
```

```

# 02. 바운더리 박스(bbox) 설정하기
bound <- st_bbox(map)

xrange <- bound$xmax - bound$xmin ; xrange # x(lon) 범위
yrange <- bound$ymax - bound$ymin ; yrange # y(lat) 범위

# 03. 그리드 간격 설정하기
grid_size_x <- 1
grid_size_y <- 1

# 04. 필요 그리드 수 계산하기
num_grid <- c(xrange/grid_size_x , yrange/grid_size_y) %>% # 필요한 그리드 수 계산
  ceiling() # 소수점 버리기

num_grid

# 05. 그리드 생성하기
grid_km <- st_make_grid(map, square = T, n = num_grid) %>% # bbox 범위 내 전체 그리드 만들기
  st_intersection(map) %>% # 지도 범위 내 그리드만 클리핑
  st_sf("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0") # 좌표계 WGS84로 변환

# 06. 플로팅 하기
plot(grid_km)

# 07. tmap으로 확인하기
library(tmap)
qtm("Seoul")

qtm("Seoul") +
  tm_shape(grid_km) +
  tm_polygons(alpha = 0.1, border.col = "gray50", border.alpha = .2, colorNA = NULL) +
  tm_text(text="id", size = 1)

```



```
#####  
## 4부. GRID DTW 분석하기 ##  
#####
```

```
#-----  
# chap 4-1. 필터링  
#-----
```

```
memory.limit(500000)
```

```
# 01. 아파트 가격 데이터 불러오기
```

```
library(dplyr) # install.packages("dplyr")  
library(reshape2)  
library(data.table)  
library(dtw)  
library(spdep)
```

```
# 02. 파일 불러오기
```

```
setwd(dirname(rstudioapi::getSourceEditorContext()$path))  
item <- list()  
  
load("./3_merge/07_price_data_3.Rdata")  
load("./4_map/grid_2k.rdata")  
colnames(grid_km@data)[1] = "k2"  
grid_km <- as(grid_km, "sf")
```


03. 동이를 join 하기

```
dong_id <- price_data_3[!duplicated(price_data_3$k2),]  
dong_id <- dong_id[,c("k2", "region", "dong")]  
grid_km <- merge(grid_km, dong_id, by ="k2", all.x=TRUE)
```

04. 반복 데이터 세팅

```
start <- c("201101", "201102", "201103", "201104", "201105", "201106", "201107", "201108", "201109", "201110", "201111", "201112",  
          "201201", "201202", "201203", "201204", "201205", "201206", "201207", "201208", "201209", "201210", "201211", "201212",  
          "201301", "201302", "201303", "201304", "201305", "201306", "201307", "201308", "201309", "201310", "201311", "201312",  
          "201401", "201402", "201403", "201404", "201405", "201406", "201407", "201408", "201409", "201410", "201411", "201412",  
          "201501", "201502", "201503", "201504", "201505", "201506", "201507", "201508", "201509", "201510", "201511", "201512",  
          "201601", "201602", "201603", "201604", "201605", "201606", "201607", "201608", "201609", "201610", "201611", "201612",  
          "201701", "201702", "201703", "201704", "201705", "201706", "201707", "201708", "201709", "201710", "201711", "201712",  
          "201801")  
  
end <- c("201212",  
        "201301", "201302", "201303", "201304", "201305", "201306", "201307", "201308", "201309", "201310", "201311", "201312",  
        "201401", "201402", "201403", "201404", "201405", "201406", "201407", "201408", "201409", "201410", "201411", "201412",  
        "201501", "201502", "201503", "201504", "201505", "201506", "201507", "201508", "201509", "201510", "201511", "201512",  
        "201601", "201602", "201603", "201604", "201605", "201606", "201607", "201608", "201609", "201610", "201611", "201612",  
        "201701", "201702", "201703", "201704", "201705", "201706", "201707", "201708", "201709", "201710", "201711", "201712",  
        "201801", "201802", "201803", "201804", "201805", "201806", "201807", "201808", "201809", "201810", "201811", "201812",  
        "201901", "201902", "201903", "201904", "201905", "201906", "201907", "201908", "201909", "201910", "201911", "201912")  
  
mid <- c("201106", "201107", "201108", "201109", "201110", "201111", "201112",  
        "201301", "201302", "201303", "201304", "201305", "201306", "201307", "201308", "201309", "201310", "201311", "201312",  
        "201401", "201402", "201403", "201404", "201405", "201406", "201407", "201408", "201409", "201410", "201411", "201412")
```

04. DTW 계산하기

```
for(t in 1: length(start)) {  
  var_1 <- as.character(paste0("price_data_5 <- price_data_3 %>% filter(year_M %in% (", start[t], ":", end[t], ") & size == 3) %>% group_by(y  
  eval(parse(text=var_1)) ; rm(var_1)  
  price_data_6 <- reshape2::dcast(price_data_5, k2 ~ year_M, value.var="avg_price", mean) # (2) year_M 별로 합치기  
  rm(price_data_5)  
  price_data_6 <- price_data_6[complete.cases(price_data_6), ] # NaN 포함된 row 지우기  
  rownames(price_data_6) <- price_data_6[,1] # row_names 정리하기  
  price_data_6 <- price_data_6[, -1]  
  price_data_6_1 <- price_data_6[, 1:12] # 쪼개기  
  price_data_6_2 <- price_data_6[, 13:24]  
  price_data_6_1 <- t(apply(price_data_6_1, 1, function(x)(x-min(x))/(max(x)-min(x)))) # 표준화 시키기  
  price_data_6_2 <- t(apply(price_data_6_2, 1, function(x)(x-min(x))/(max(x)-min(x))))  
  price_data_6_1 <- as.data.frame((price_data_6_1))  
  price_data_6_2 <- as.data.frame((price_data_6_2))  
  # rm(price_data_6)  
  item <- list()  
  for (i in 1: nrow(price_data_6_1)) {  
    var_1 <- as.character(paste0("dist <- dist(price_data_6_1[,i,", ], price_data_6_2[,i,", ], method=\\"DTW\\")"))  
    eval(parse(text=var_1))  
    item_temp <- data.table(row.names(dist), dist[1])  
    item[[i]] <- item_temp  
  }  
  dtw_bind <- rbindlist(item)  
  rm(price_data_6_1) ; rm(price_data_6_2) ; rm(item)  
  dtw_bind[,2] <- apply(dtw_bind[,2], 2, function(x)(x-min(x))/(max(x)-min(x)) )  
  var_2 <- as.character(paste0("colnames(dtw_bind) <- c( \"k2\\", \"t_\", mid[t] , \"\\")\" ) )  
  eval(parse(text=var_2))  
  grid_km <- merge(grid_km, dtw_bind, by ="k2", all.x=TRUE)  
  # rm(dtw_bind) ; rm(var_1)  
  pcnt <- round(t/ length(start)*100, 1)  
  message <- paste0(t, "번째 데이터 계산 중입니다 [", pcnt, "%"] )  
  cat(message, "\n\n")  
}
```

k2	region	dong	t_201106	t_201107	t_201108	t_201109	t_201110	t_201111	t_201112	t_201301	t_201302
88	강서구	등촌동	0.21760462	0.3001519	0.11748547	0.2169303	0.34723147	0.13875080	0.35412669	0.379534879	0.69688647
89	강서구	염창동	0.24706901	0.4369854	0.15547607	0.3689332	0.38299115	0.23814037	0.11692541	0.318625879	0.50314922
90	마포구	합정동	0.29747062	0.4023285	0.30812439	0.5192467	0.52819784	0.41009117	0.25041379	0.185785152	0.20462581
91	마포구	구수동	0.64807302	0.5352823	0.35166451	0.4019097	0.35558832	0.36233194	0.36754209	0.284145996	0.14653541
92	마포구	공덕동	0.17151242	0.2865034	0.18327904	0.3410188	0.37854501	0.18312935	0.28368767	0.210684140	0.16541563
93	용산구	갈월동	0.75594212	0.7010231	0.56434647	0.4536401	0.21758383	0.04744240	0.06903782	0.013445746	0.02896380
95	강남구	압구정동	0.55040125	0.4795838	0.26202908	0.4701056	0.18252279	0.00000000	0.00000000	0.075984265	0.10942362

IV. 실증분석 및 웹기반 플랫폼 구축

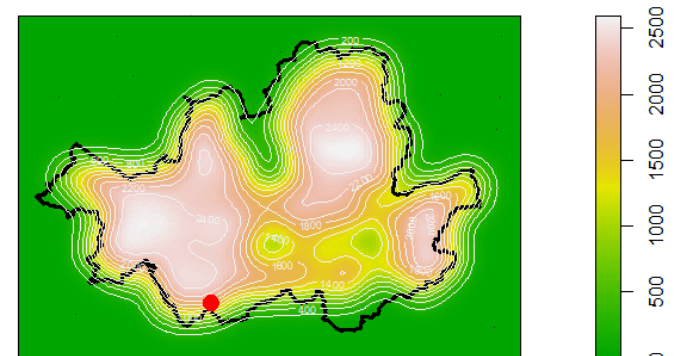
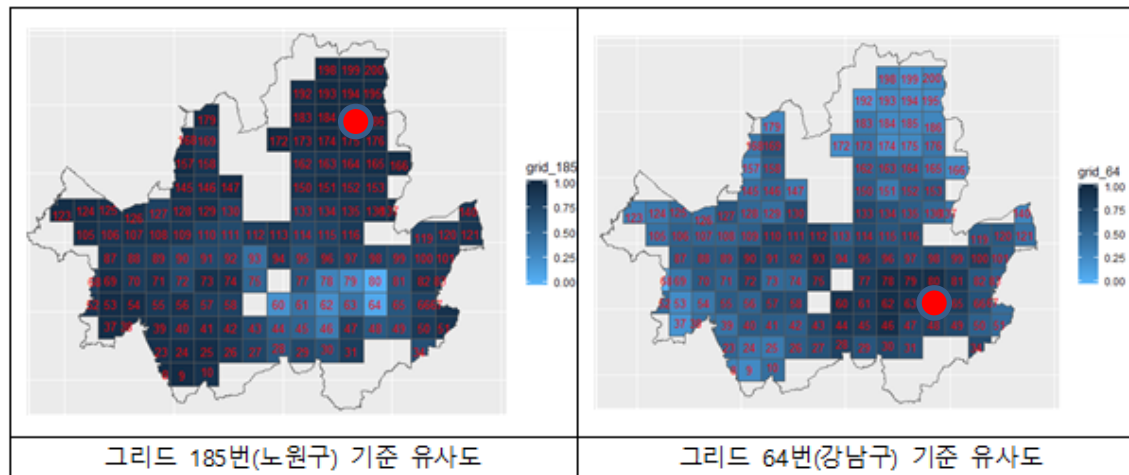
□ DTW를 활용한 보증금 패턴 유사지역 분석: 소형

- 소형전세 APT의 DTW 유사도 비교

- 노원구(grid_185)와 강남구(grid_64)의 DTW 유사도 비교
- 분석 결과 모두 강남과 용산 지역을 기준으로 보증금 패턴 변화가 명확
- 커널밀도분석(동영상)으로 결과를 보정 → 주택하위시장 분류가 명확하게 나타남

grid_10

그림 4-1 서울시 소형 아파트 전세가 패턴변화 유사지역 분석(2011-2019)



IV. 실증분석 및 웹기반 플랫폼 구축

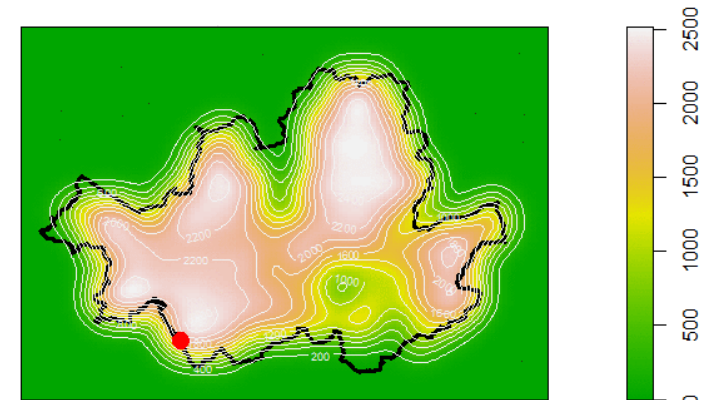
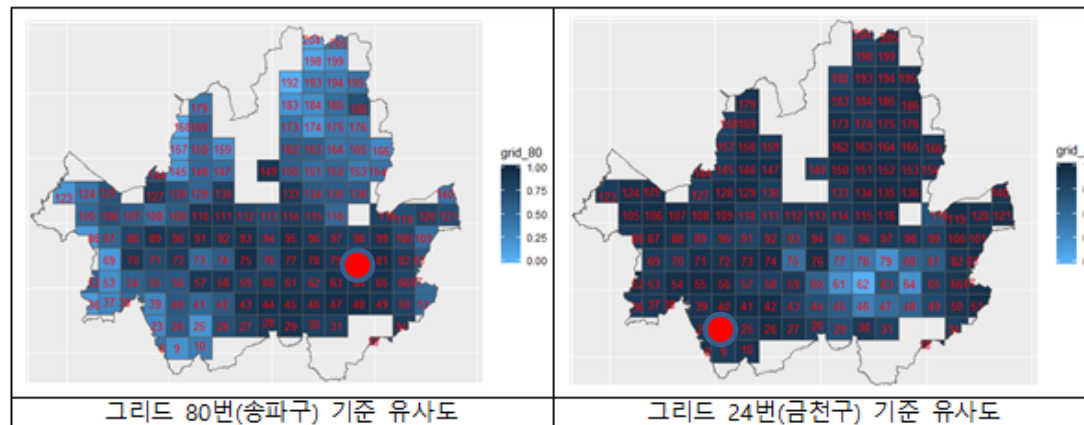
□ DTW를 활용한 보증금 패턴 유사지역 분석: 중형

- 중형전세 APT의 DTW 유사도 비교

- 금천구(grid_24)와 송파구(grid_80)의 DTW 유사도 비교
- 분석 결과 모두 강남 지역을 기준으로 보증금 패턴 변화가 구분됨
- 커널밀도분석(동영상)으로 결과를 보정 → 주택하위시장 분류가 명확하게 나타남

grid_8

그림 4-2 서울시 중형 아파트 전세가 패턴변화 유사지역 분석(2011-2019)



IV. 실증분석 및 웹기반 플랫폼 구축

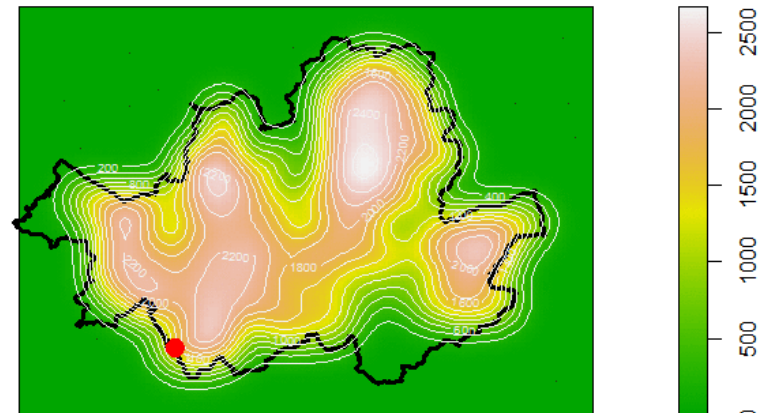
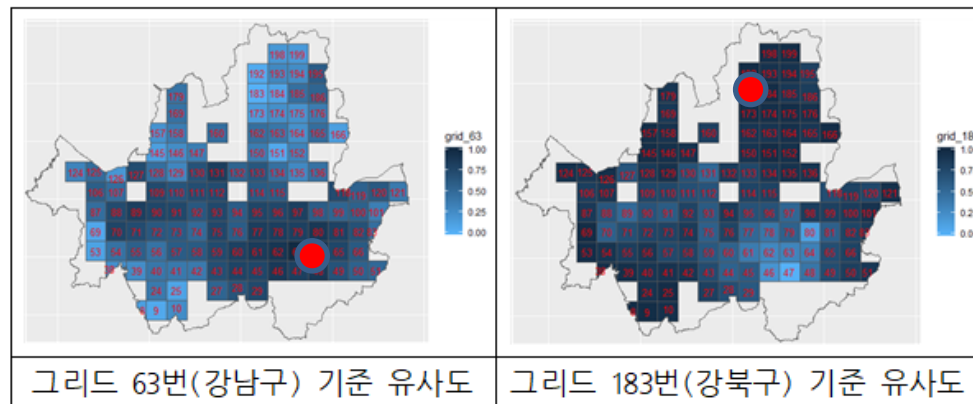
□ DTW를 활용한 보증금 패턴 유사지역 분석: 대형

- 대형전세 APT의 DTW 유사도 비교

- 강북구(grid_183)와 강남구(grid_63)의 DTW 유사도 비교
- 소형과 중형 전세 APT에 비하여 상대적으로 경계가 명확하지 않음
- 커널밀도분석(동영상)으로 결과를 보정 → 주택하위시장 분류가 명확하게 나타남

grid_8

그림 4-3 서울시 대형 아파트 전세가의 DTW 유사지역 분석(2011-2019)



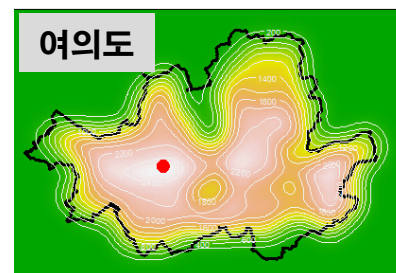
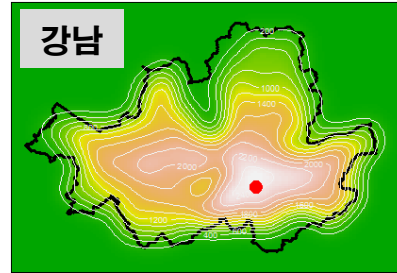
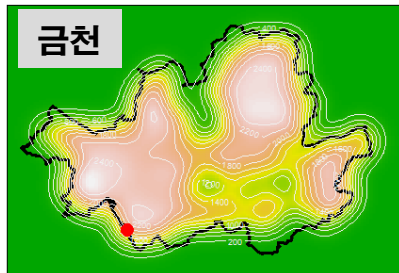
IV. 실증분석 및 웹기반 플랫폼 구축

□ 전세 APT 보증금 변화 → 시계열 패턴 매칭 분석

- DTW 활용하여 그리드 간 보증금 변화 유사도 계산 분석

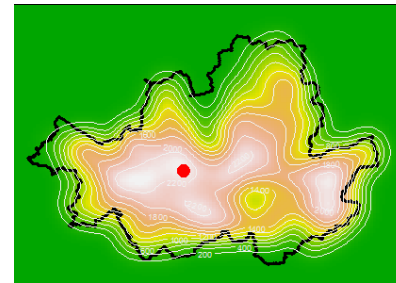
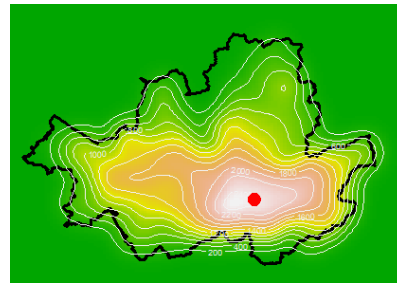
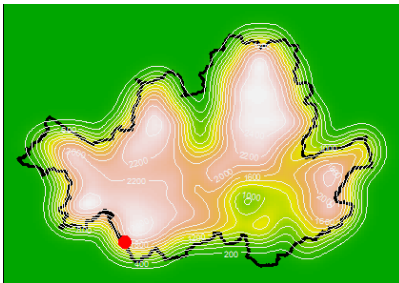
- 전세 APT 보증금 시계열 변화 → 시공간적 동조화 발생지역 도출 (패턴차이)

소형
전세 APT



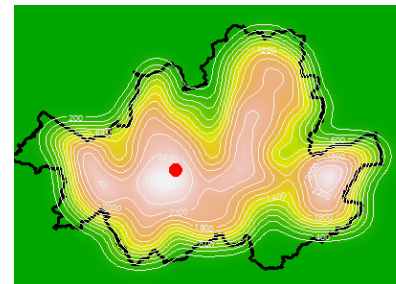
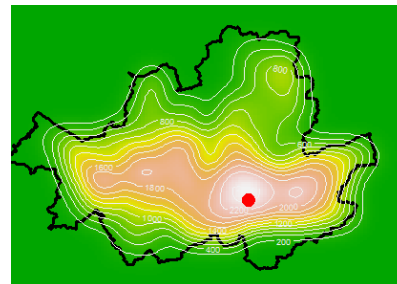
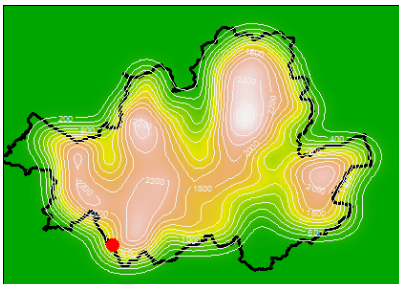
어느지역을 기준으로
하였는가에 따라서
패턴 유사도가 달라짐

중형
전세 APT



강남의 경우, 가격변화 패턴의
유사도가 unique 함

대형
전세 APT



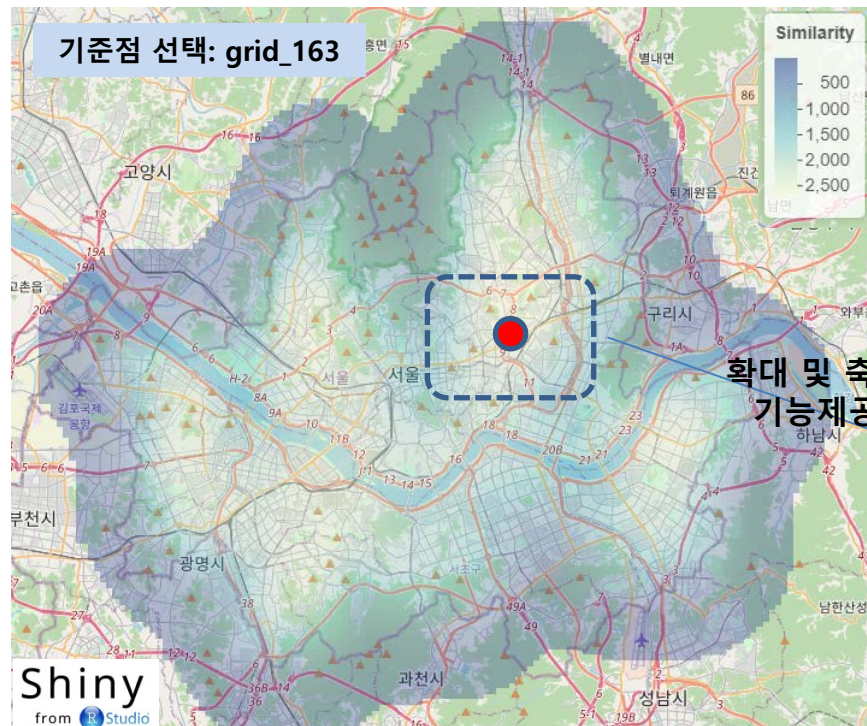
전세가격 변화를 살펴보았을 때
강남을 기준으로 서울의 다른
지역의 가격변화를 이해하고자
하는 노력이 유효한지는 고민
해 보아야 할 문제임

IV. 실증분석 및 웹기반 플랫폼 구축

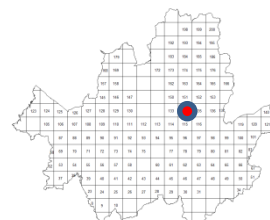
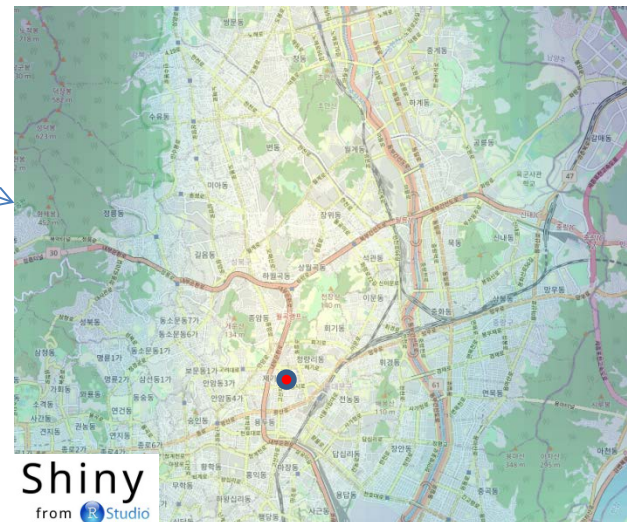
□ Shiny를 활용한 웹기반 플랫폼 구축

- 분석결과를 다른 연구자 또는 대중에게 공유할 수 있는 분석도구인

Shiny를 사용하여 실제 지도 위에 커널밀도추정(KDE) 분석결과 공유



확대 및 축소
기능제공



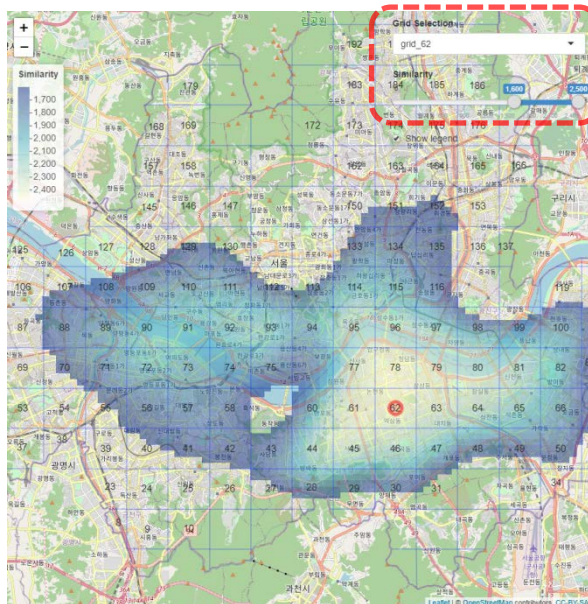
기준점 선택: grid_163

IV. 실증분석 및 웹기반 플랫폼 구축

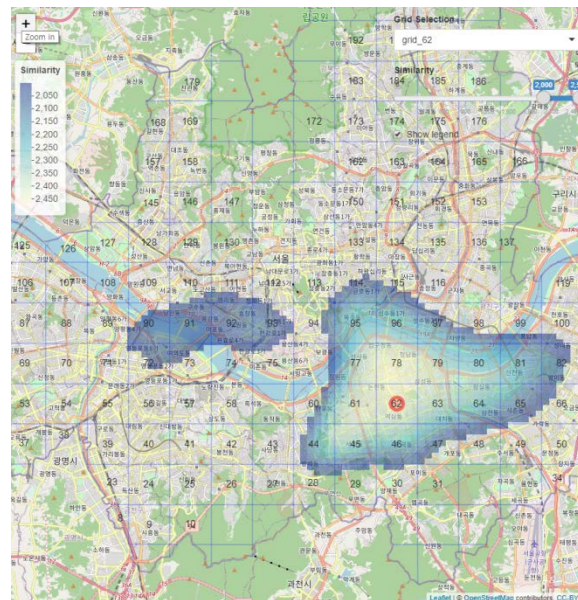
□ Shiny를 활용한 웹기반 플랫폼 구축

- 강남구 역삼동(grid_62)을 기준으로 시공간 패턴변화 유사지역 도출

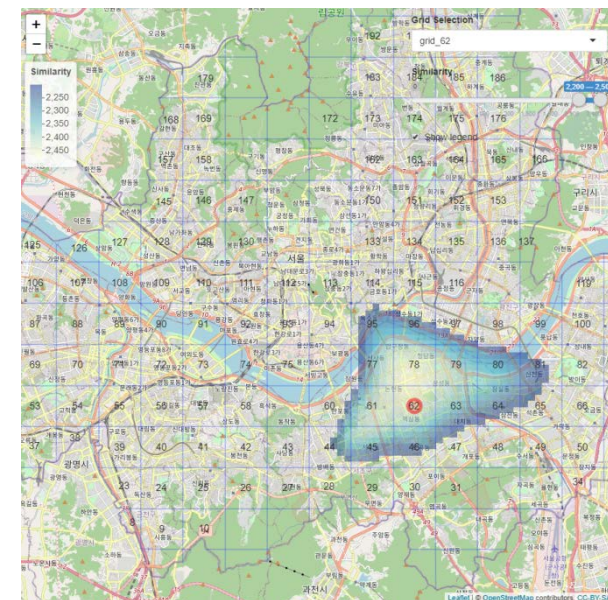
시공간 패턴변화 유사도(커널 범위: 0~2500)가 높아질수록 공간범위가 점차 축소됨



범위: 1,600~2,500



범위: 2,000~2,500



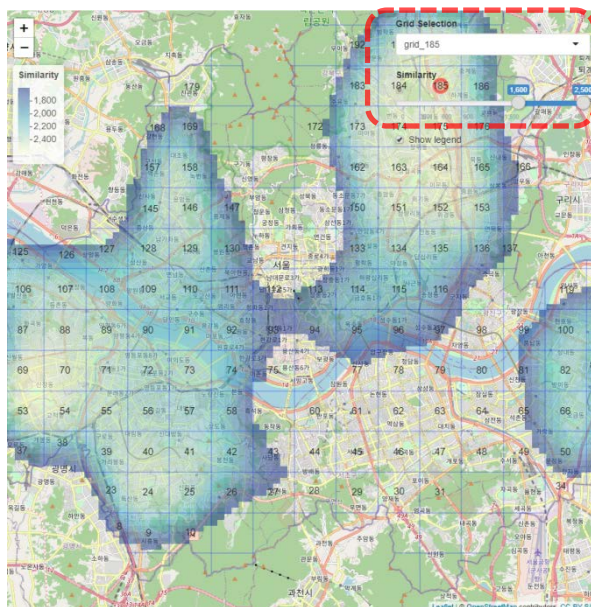
범위: 2,200~2,500

IV. 실증분석 및 웹기반 플랫폼 구축

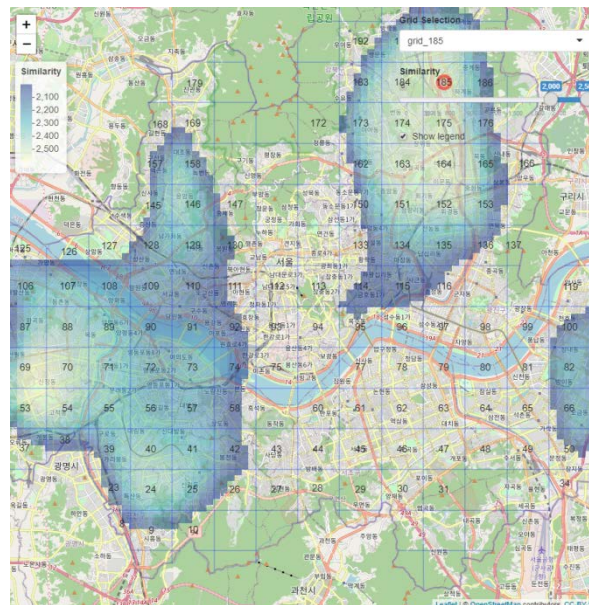
□ Shiny를 활용한 웹기반 플랫폼 구축

- 노원구 하계동(grid_185)을 기준으로 시공간 패턴변화 유사지역 도출

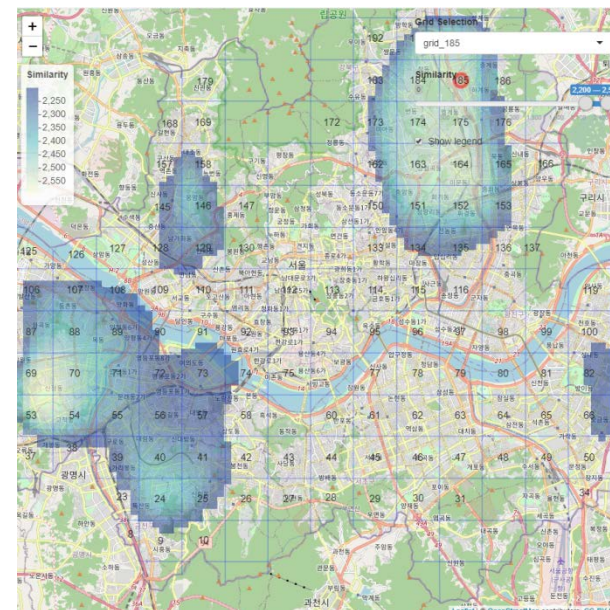
시공간 패턴변화 유사도(커널 범위: 0~2500)가 높아질수록 공간범위가 점차 축소됨



범위: 1,600~2,500



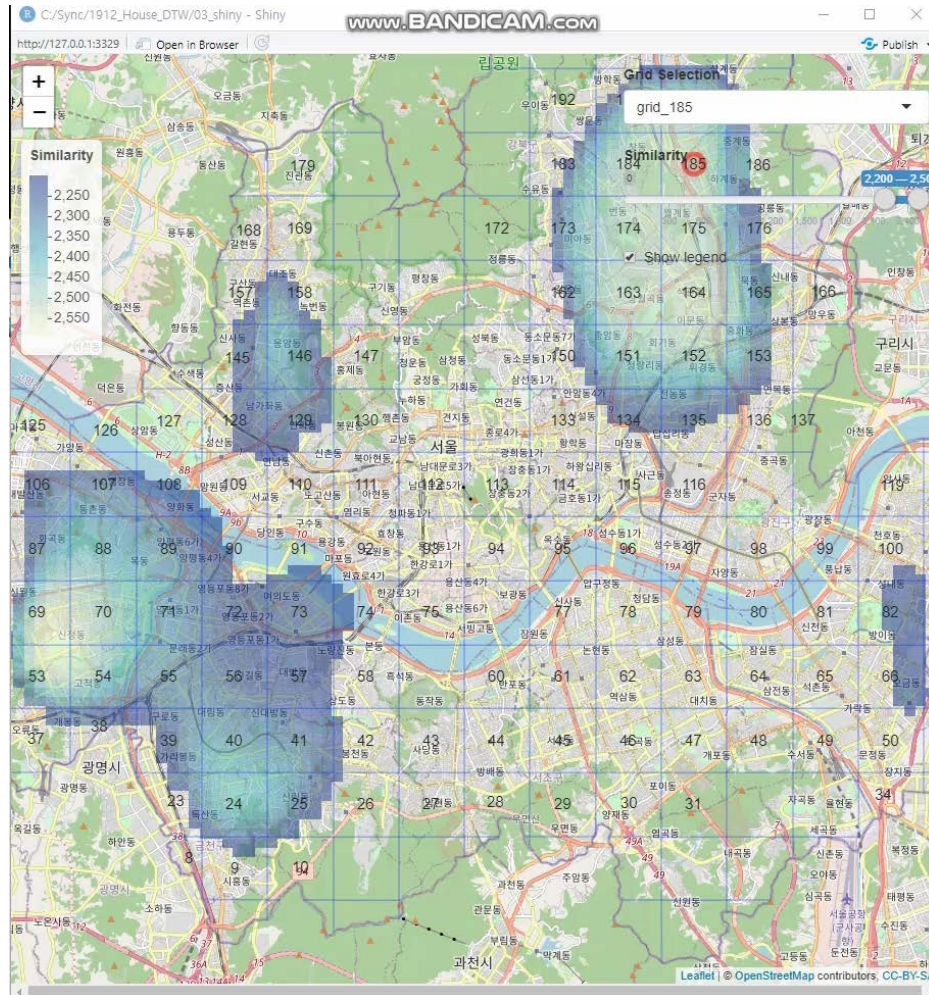
범위: 2,000~2,500



범위: 2,200~2,500

IV. 실증분석 및 웹기반 플랫폼 구축

□ Shiny를 활용한 웹기반 플랫폼 구축

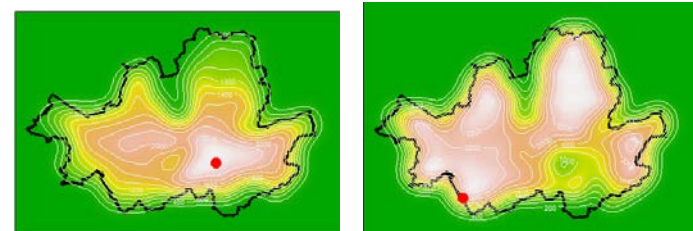
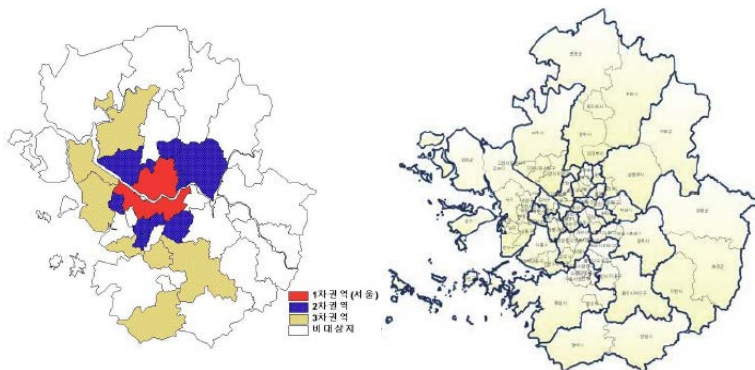


- 분석결과를
인터랙티브하게 활용 가능
(Shiny + Open Street Map)

V. 결론

□ 주요분석 결과

- 시계열 패턴 매칭 방법인 DTW(Dynamic Time Warping)를 활용하여
그리드 간 유사도를 바탕으로 전세가격 변화 유사 패턴 지역 특정
- 이 때, 아파트 전세가격이 주택의 크기에 따라 다르게 나타나기 때문에
크기(소·중·대형)에 따라 DTW 유사성을 별도로 분석
- 선행연구들에서 고려하지 못했던 **(공간적)정밀도**와 **(시간적)역동성** 모두 고려함



시공간 패턴을 고려한
주택하위시장 구분은 역동적임

- 빅데이터 분석 도구의 장점을 살려 → shiny를 활용한 **웹기반 플랫폼 구축**
(다른 연구자 또는 일반인들이 접속하여 분석결과 확인 가능한 플랫폼)

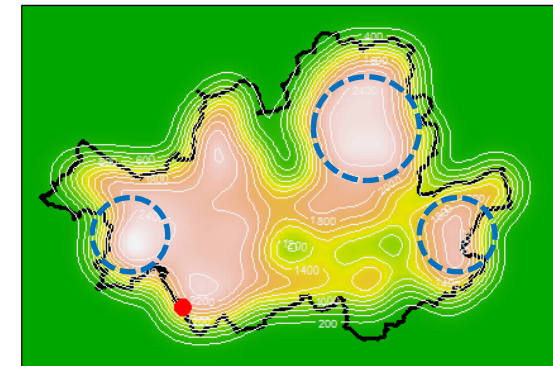
V. 결론

□ 한계와 후속연구

- 빅데이터 연구의 약점과 극복방안

- 현상은 명확하게 보여주지만 인과관계를 설명하지 못한다는 한계를 가지고 있음
- 다중회귀 또는 요인분석을 통하여 각각의 패턴 발생의 공통적 요인 추출 연구 필요
- 그랜저 인과관계 분석을 통하여 패턴 발생의 선후관계 분석

유사패턴 발생지역 분석



- 빅데이터 연구의 장점을 활용한 미래예측

- 머신러닝을 활용하여 유사 주택하위시장별 미래변화 예측

- 연구의 의미

- 학술: 부동산 하위시장을 분류하기 위한 새로운 방법론(시공간특성 고려) 제시
- 활용: APT 보증금 변화패턴 유사지역 도출 → 정밀 주택정책 위한 기초 데이터 제시

감사합니다.