

플레이 데이터 프로젝트

# 주차 수요



차 차 차 : 권 성 재 김 민 정 김 성 현 나 혜 원 이 승 현



PLAY DATA



# 목차

## I 대한민국의 주차 문제

i 현 상황 - 수도권 중심의 주차 문제

ii 데이터 소개

## II 데이터 분석 및 머신러닝

i 데이터 전처리

ii 모델별 비교

iii 적합한 모델 선정

iv 예측 및 결과

## III 결론

i 결론

ii 출처



# I 대한민국의 주차 문제

II 데이터 분석 및 머신러닝

III 결론

## i 현 상황 - 수도권 중심의 주차 문제

### 내차앞에 주차 좀 그만...7만대 달곳 없는 서울 '주차전쟁'

김영진 기자 | 구독

f t d l ★ 📄 가+

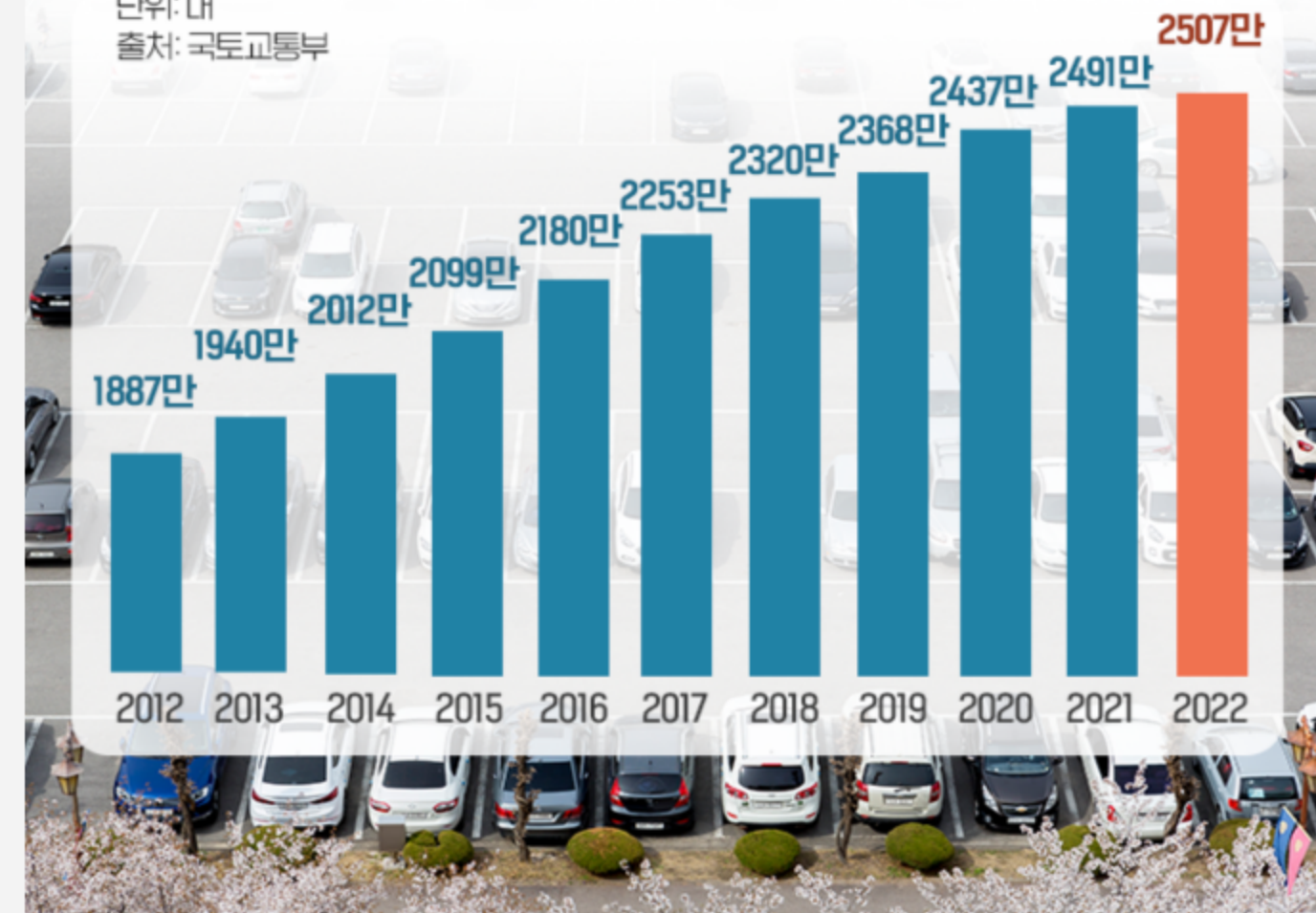
권익위 민원 분석  
2010년 162건→20년 2만4천건 153배 급증  
사유지 주차단속 주차면수 기준 강화 지적도



불법주차 문제는 어제오늘 일이 아니다. 도심에선 주차금지라고 적혀 있는 표지판 앞에 차를 세워둔 모습도 어렵지 않게 볼 수 있다. <한겨레 21> 2007년 3월 20일자에 실렸던 사진이다. <한겨레> 자료사진

### 자동차 등록 현황 추이

단위: 대  
출처: 국토교통부



# I 대한민국의 주차 문제

II 데이터 분석 및 머신러닝

III 결론

## ii 데이터 소개

### 배경

아파트 단지 내 필요한 주차대수

①법정주차대수 ②장래주차수요



②장래주차수요:

인력조사로 인한 오차발생,

현장조사 시점과 실제 건축시점과의 시간차 등의 문제

➡ 과대 또는 과소 산정의 가능성 존재

### 목적

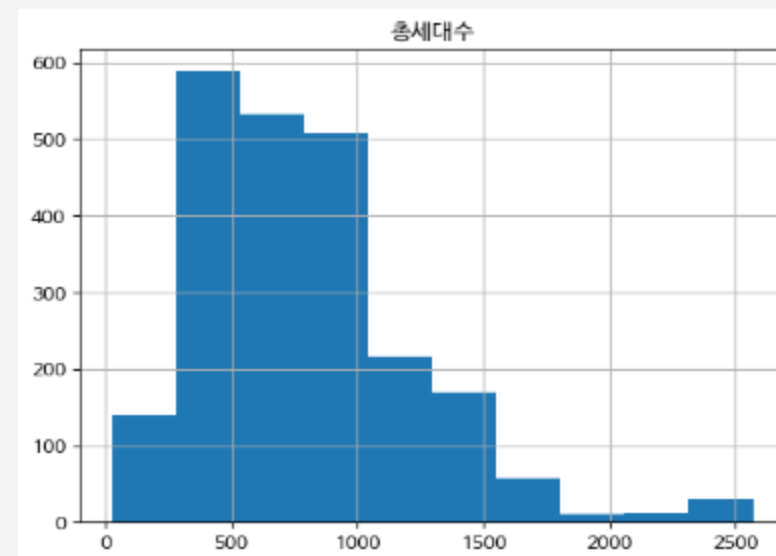
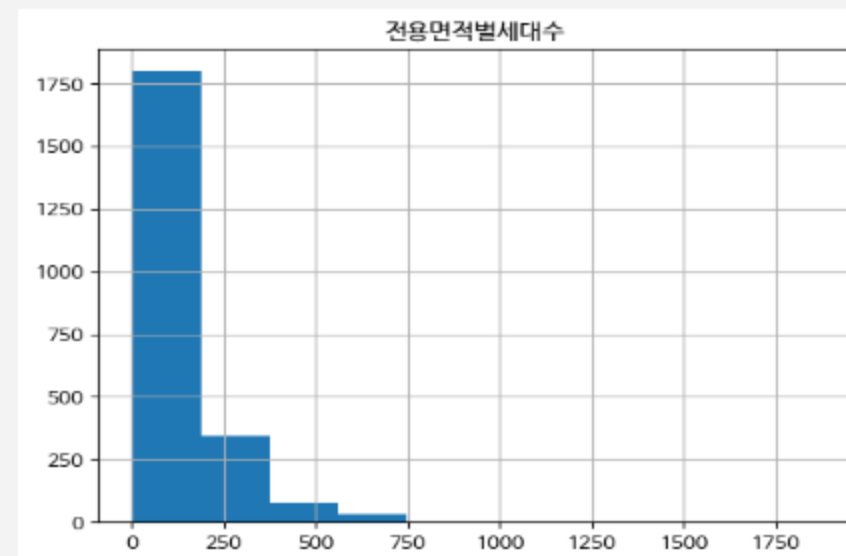
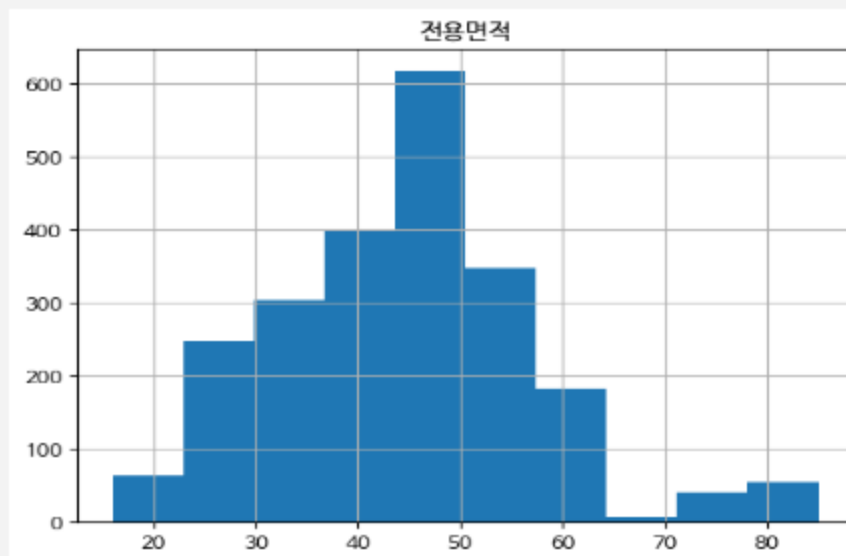
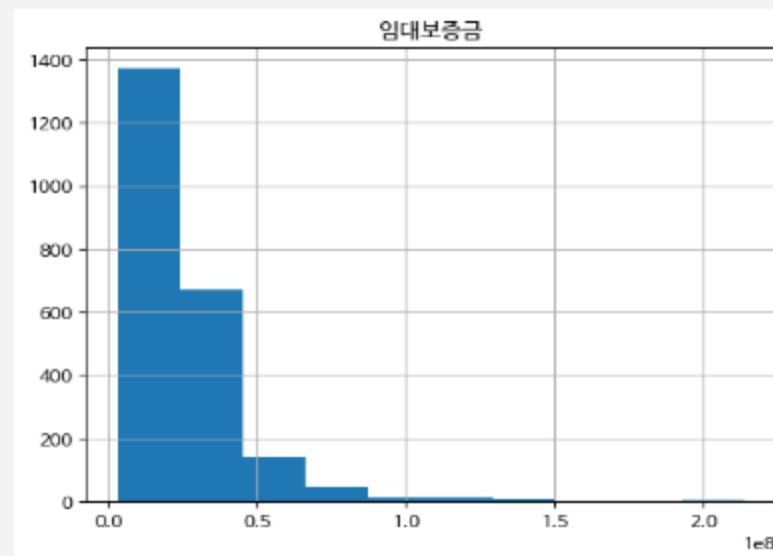
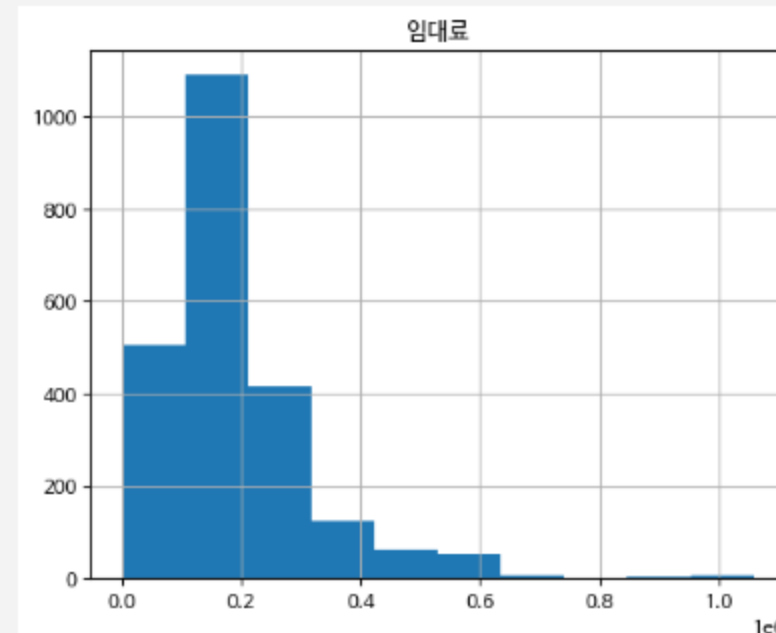
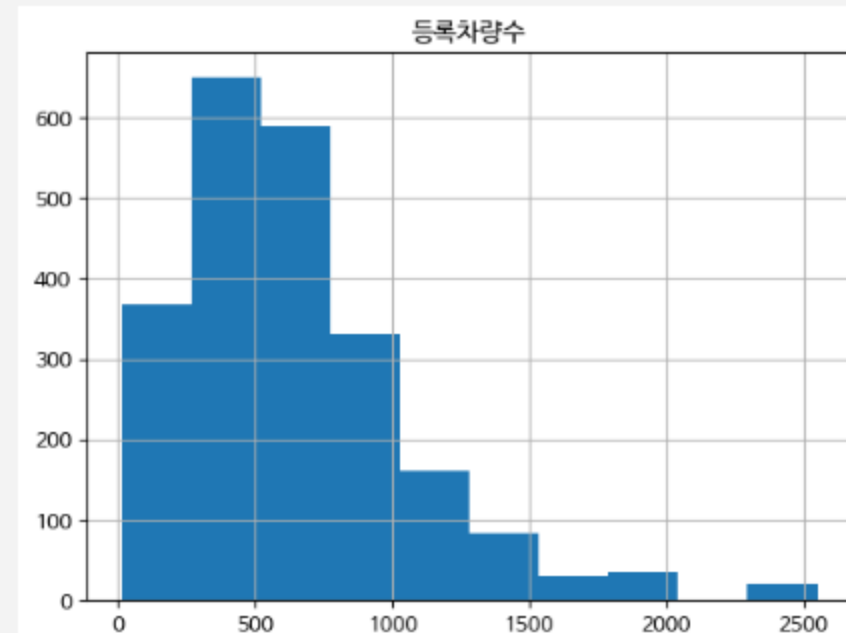
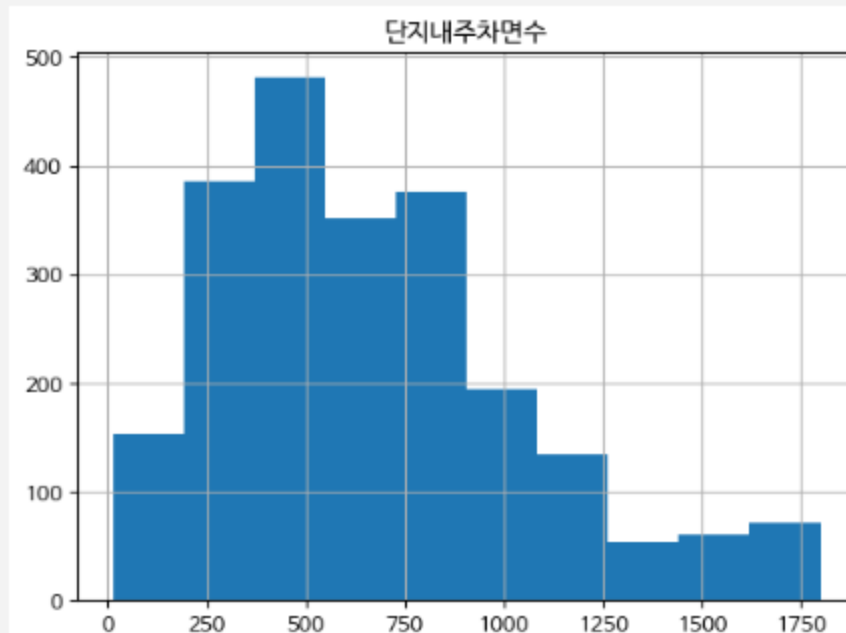
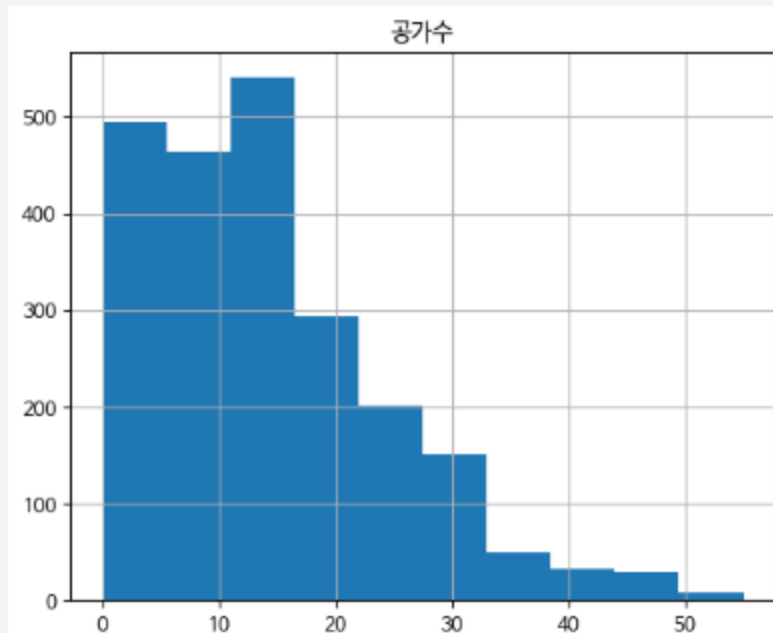
유형별 임대주택 설계 시

주차 수요를 예측

### 데이터 출처

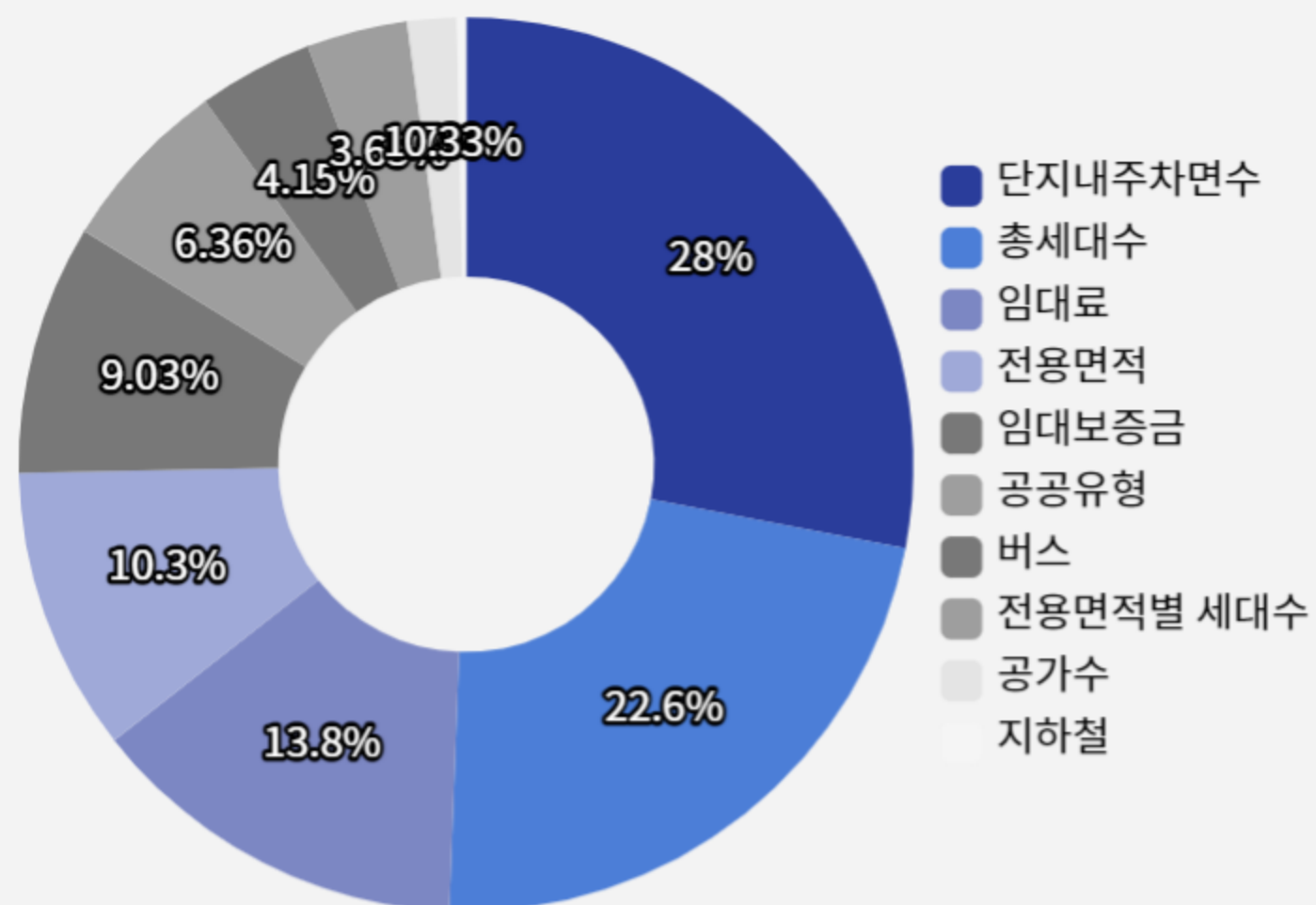
한국토지주택공사

## i 데이터 전처리 - 데이터 분포



## i 데이터 전처리 - 상관관계 분석

상관관계 데이터



## 종속변수

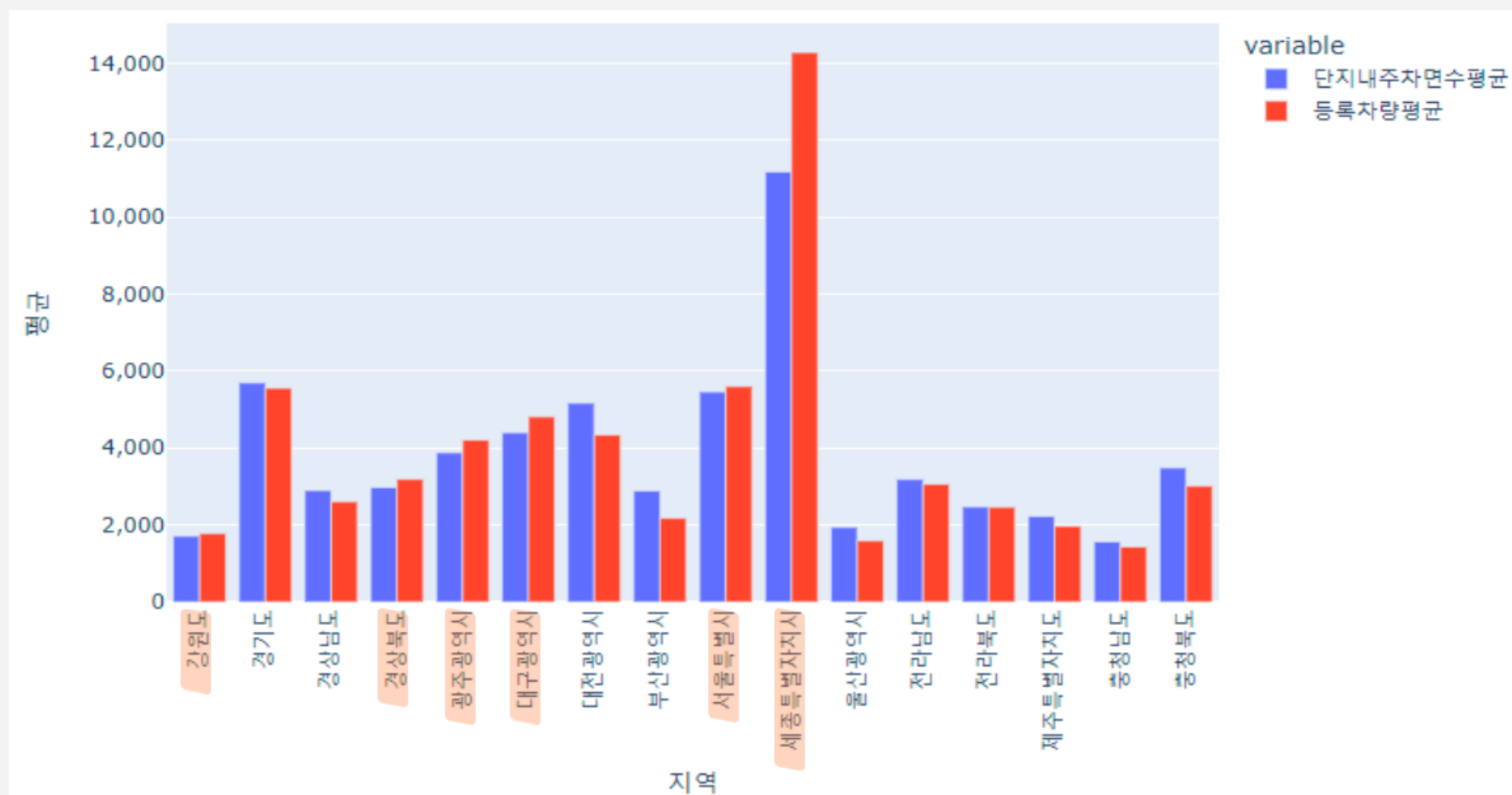
등록차량수

## 독립변수

종속변수와 상관관계 높은 n개의 독립변수  
➡ 독립변수들 간의 다중공선성 등의 문제가 발생  
➡ 독립변수 전체 사용

\* 다중공선성  
: 독립변수들 간에 강한 상관관계가 나타나는 문제

## i 데이터 전처리 - train 데이터 지역별 단지내 주차면수 평균과 등록차량 평균 비교





## i 데이터 전처리 - 원핫인코딩 대상 판단

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2264 entries, 8 to 2951
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   총세대수              2264 non-null   int64
1   임대건물구분          2264 non-null   object
2   지역                  2264 non-null   object
3   공급유형              2264 non-null   object
4   전용면적              2264 non-null   float64
5   전용면적별세대수      2264 non-null   int64
6   공가수                2264 non-null   float64
7   자격유형              2264 non-null   object
8   임대보증금            2264 non-null   float64
9   임대료                2264 non-null   float64
10  지하철               2264 non-null   float64
11  버스                  2264 non-null   float64
12  단지내주차면수        2264 non-null   float64
13  등록차량수            2264 non-null   float64
dtypes: float64(8), int64(2), object(4)
```

## 원핫인코딩 대상

- 임대건물구분
- 지역
- 공급유형
- 자격유형

## 범주형 데이터

지역 : 총 15개  
공급유형 : 총 7개  
자격유형 : 총 13개

➡ 그룹화로 해결



## i 데이터 전처리 - 원핫인코딩 대상 판단

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2264 entries, 8 to 2951
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   총세대수              2264 non-null   int64
1   임대건물구분          2264 non-null   object
2   지역                  2264 non-null   object
3   공급유형              2264 non-null   object
4   전용면적              2264 non-null   float64
5   전용면적별세대수      2264 non-null   int64
6   공가수                2264 non-null   float64
7   자격유형              2264 non-null   object
8   임대보증금            2264 non-null   float64
9   임대료                2264 non-null   float64
10  지하철               2264 non-null   float64
11  버스                  2264 non-null   float64
12  단지내주차면수        2264 non-null   float64
13  등록차량수            2264 non-null   float64
dtypes: float64(8), int64(2), object(4)
```

	지역	공급유형	자격유형
0	경상남도	국민임대	A
1	대전광역시	공공임대(50년)	B
2	경기도	영구임대	C
3	전라북도	공공임대(10년)	E
4	강원도	공공임대(분납)	F
5	광주광역시	장기전세	G
6	충청남도	행복주택	H
7	부산광역시	공공임대(5년)	I
8	제주특별자치도		J
9	울산광역시		K
10	충청북도		L
11	전라남도		M
12	경상북도		N
13	대구광역시		O
14	서울특별시		
15	세종특별자치시		

## 원핫인코딩 대상

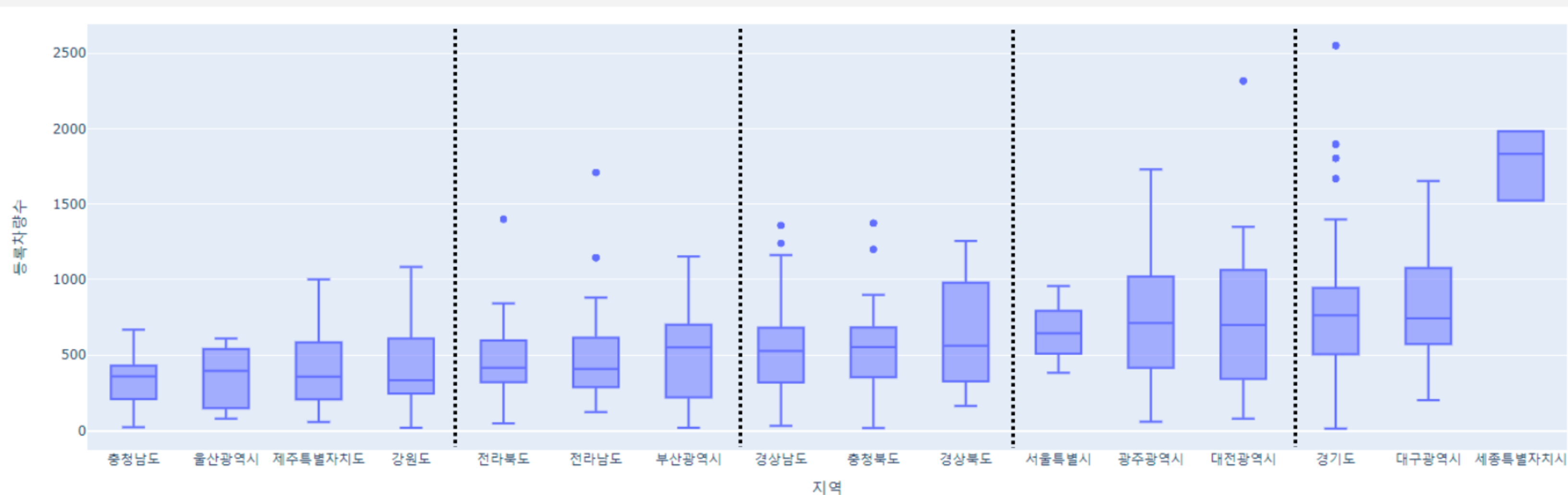
- 임대건물구분
- 지역
- 공급유형
- 자격유형

## 범주형 데이터

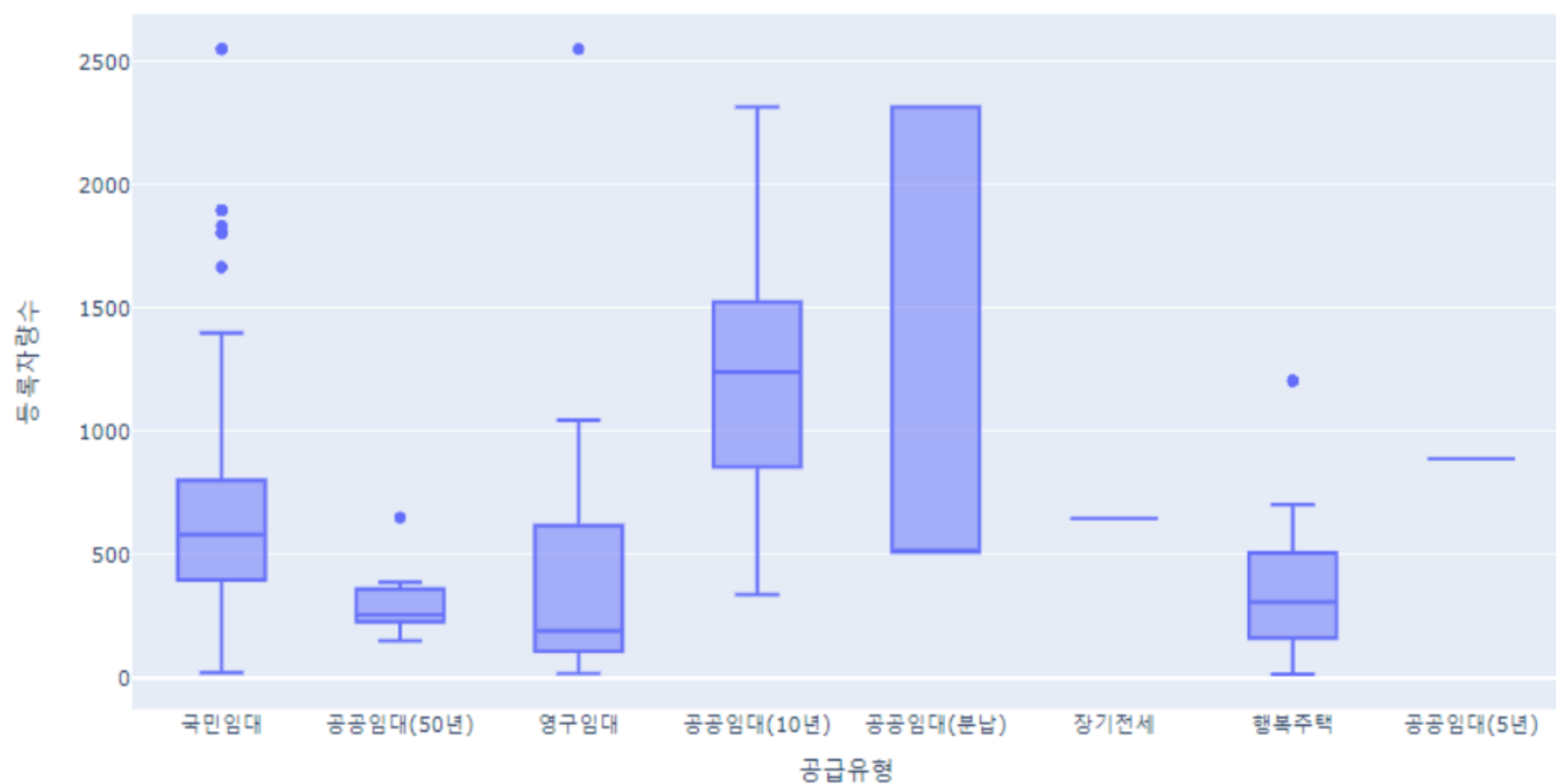
지역 : 총 15개  
공급유형 : 총 7개  
자격유형 : 총 13개

➡ 그룹화로 해결

## i 데이터 전처리 - 그룹화) 지역 분류



## i 데이터 전처리 - 그룹화) 공급유형 분류



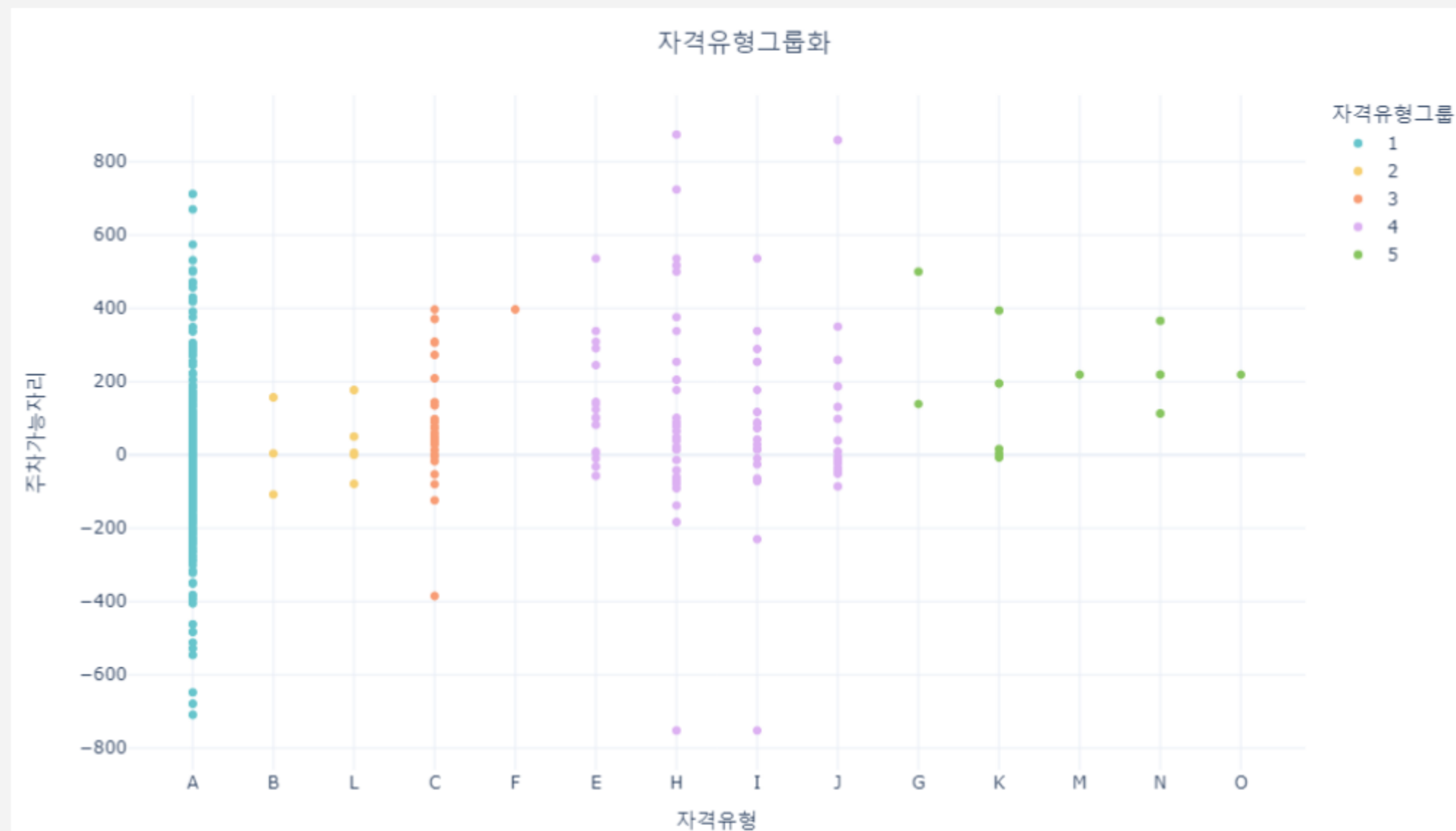
국민임대 ➡ 국민임대

공공임대 (50년) ➡ 장기공공임대

공공임대(분납, 5년, 분납)  
➡ 단기공공임대

영구임대, 장기전세, 행복주택  
➡ 기타

## i 데이터 전처리 - 그룹화) 자격유형 분류



## 주차가능자리

단지내주차면수 - 등록차량수

## 자격유형 분류 결과

1 : A  
2 : B, L  
3 : C, F  
4 : E, H, I, J  
5 : G, K, M, N, O

## i 데이터 전처리 - 원핫인코딩 결과

	총 세 대 수	전 용 적 면 적	전 용 적 면 적 대 수	공 가 수	임 대 보 증 금	임 대 료	지 하 철	버 스	단 지 주 차 면 수	n 지 역 _0	...	공 유 그 룹 _국 민 임 대	공 유 그 룹 _단 기 공 임 대	공 유 그 룹 _장 기 공 임 대	공 유 그 룹 _기 타	자 격 그 룹 _q1	자 격 그 룹 _q2	자 격 그 룹 _q3	자 격 그 룹 _q4	자 격 그 룹 _q5	등 차 량 수
0	545	33.48	276	17.0	9216000.0	82940.0	0.0	3.0	624.0	0	...	1	0	0	0	0	0	0	0	1	205.0
1	545	39.60	60	17.0	12672000.0	107130.0	0.0	3.0	624.0	0	...	1	0	0	0	0	0	0	0	1	205.0
2	545	39.60	20	17.0	12672000.0	107130.0	0.0	3.0	624.0	0	...	1	0	0	0	0	0	0	0	1	205.0
3	545	46.90	38	17.0	18433000.0	149760.0	0.0	3.0	624.0	0	...	1	0	0	0	0	0	0	0	1	205.0
4	545	46.90	19	17.0	18433000.0	149760.0	0.0	3.0	624.0	0	...	1	0	0	0	0	0	0	0	1	205.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2259	239	49.20	19	7.0	11346000.0	116090.0	0.0	1.0	166.0	1	...	1	0	0	0	0	0	0	0	1	146.0
2260	239	51.08	34	7.0	14005000.0	142310.0	0.0	1.0	166.0	1	...	1	0	0	0	0	0	0	0	1	146.0
2261	239	51.73	34	7.0	14005000.0	142310.0	0.0	1.0	166.0	1	...	1	0	0	0	0	0	0	0	1	146.0
2262	239	51.96	114	7.0	14005000.0	142310.0	0.0	1.0	166.0	1	...	1	0	0	0	0	0	0	0	1	146.0
2263	239	54.95	19	7.0	14830000.0	151030.0	0.0	1.0	166.0	1	...	1	0	0	0	0	0	0	0	1	146.0

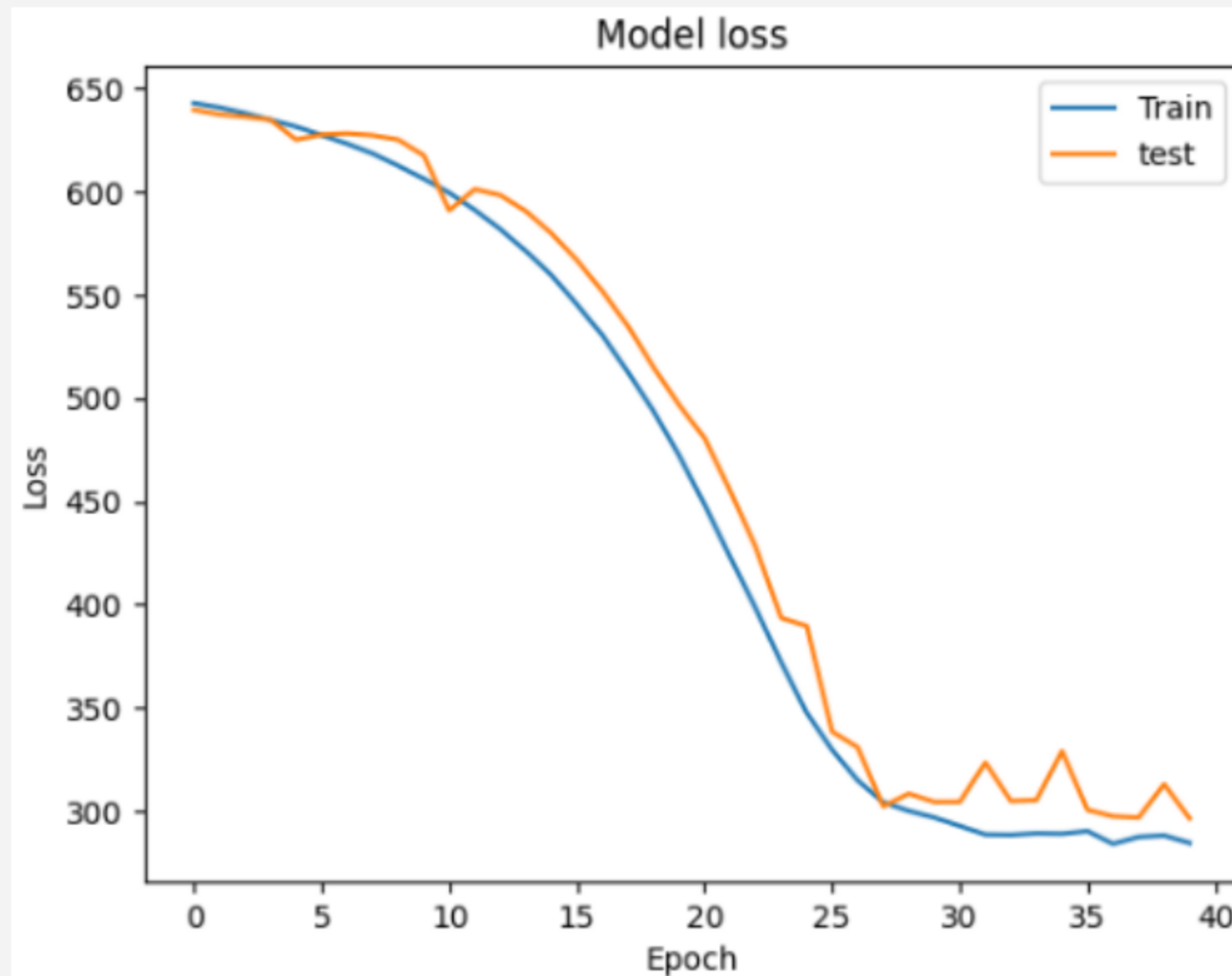
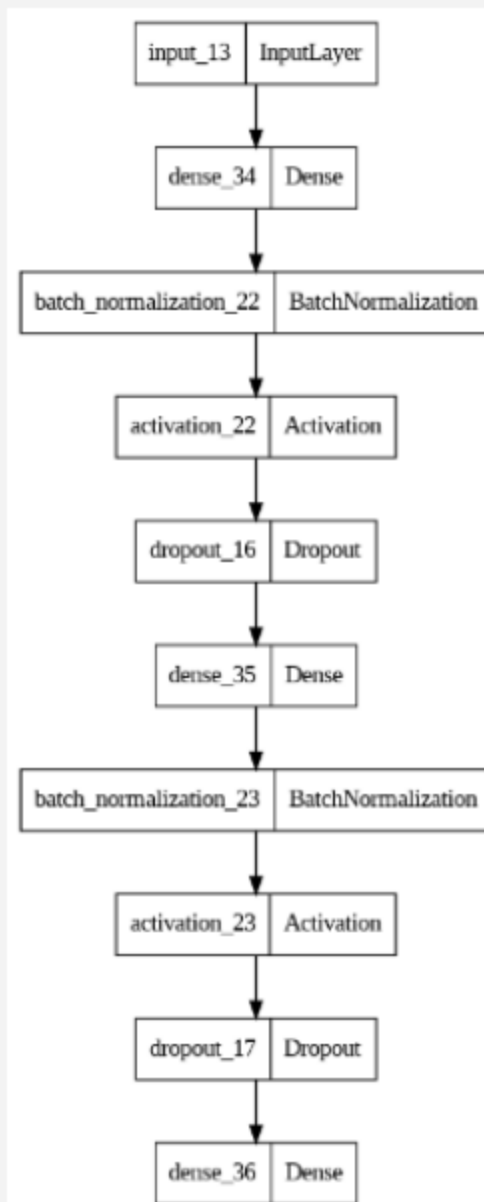
2264 rows × 24 columns

## ii 모델별 비교 - 딥러닝

Model: "model\_1"

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 23)]	0
dense_3 (Dense)	(None, 128)	3072
batch_normalization_2 (Batch Normalization)	(None, 128)	512
activation_2 (Activation)	(None, 128)	0
dropout_2 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 256)	33024
batch_normalization_3 (Batch Normalization)	(None, 256)	1024
activation_3 (Activation)	(None, 256)	0
dropout_3 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 1)	257

Total params: 37,889  
Trainable params: 37,121  
Non-trainable params: 768





## ii 모델별 비교 - pycaret

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
et	Extra Trees Regressor	10.0199	1198.6530	31.8151	0.9926	0.1024	0.0344	0.4710
xgboost	Extreme Gradient Boosting	28.0903	2851.9743	52.8277	0.9824	0.1521	0.0819	0.3340
rf	Random Forest Regressor	29.9223	3259.0749	56.6379	0.9799	0.1615	0.0884	0.9480
dt	Decision Tree Regressor	13.2708	3643.8881	59.0516	0.9775	0.1321	0.0378	0.0710
catboost	CatBoost Regressor	39.4561	3699.6117	60.2848	0.9773	0.1870	0.1157	2.8020
lightgbm	Light Gradient Boosting Machine	43.3005	4522.7895	66.8946	0.9719	0.1948	0.1243	0.2690
gbr	Gradient Boosting Regressor	81.0017	12697.4905	112.4160	0.9218	0.2773	0.2120	0.5540
ada	AdaBoost Regressor	137.3587	29593.4900	171.9110	0.8188	0.4409	0.4479	0.2140
ridge	Ridge Regression	134.8746	35605.8614	188.4694	0.7834	0.4404	0.3199	0.0890
lr	Linear Regression	135.0288	35608.1357	188.4761	0.7834	0.4537	0.3214	0.4240
br	Bayesian Ridge	134.4792	35668.8338	188.6333	0.7831	0.4198	0.3142	0.0800
llar	Lasso Least Angle Regression	134.7040	35848.0928	189.1271	0.7820	0.4369	0.3106	0.1470
lasso	Lasso Regression	134.7040	35848.0928	189.1271	0.7820	0.4369	0.3106	0.0740
en	Elastic Net	140.8105	39149.0852	197.6221	0.7624	0.4173	0.3287	0.0750
huber	Huber Regressor	217.2652	108193.1933	311.5834	0.3686	0.5589	0.5604	0.1450
knn	K Neighbors Regressor	244.4454	121696.0742	347.8657	0.2620	0.6630	0.8633	0.0800

## pycaret

데이터 전처리

➡ 다양한 머신러닝 모델 학습 후 성능 비교

➡ 하이퍼 파라미터 최적화

➡ 모델들 앙상블하여 예측 성능 개선

➡ 최종 모델 학습 후 저장

➡ 예측

➡ 시각화



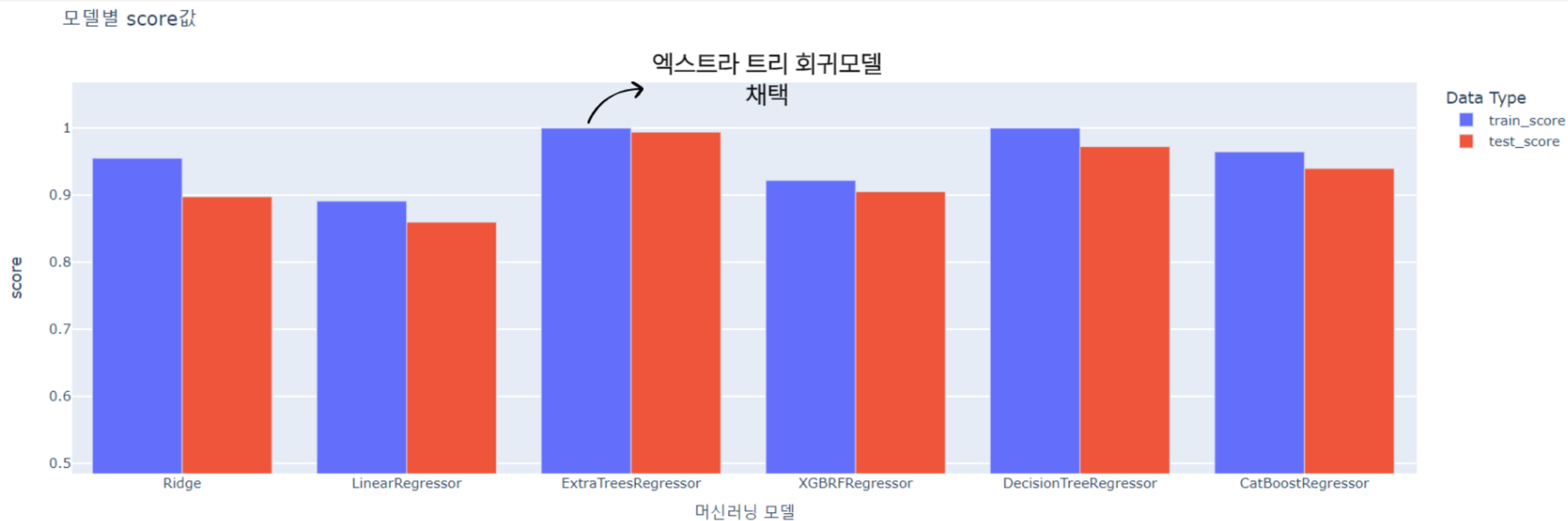
## ii 모델별 비교 - pycaret

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
et	Extra Trees Regressor	10.0199	1198.6530	31.8151	0.9926	0.1024	0.0344	0.4710
xgboost	Extreme Gradient Boosting	28.0903	2851.9743	52.8277	0.9824	0.1521	0.0819	0.3340
rf	Random Forest Regressor	29.9223	3259.0749	56.6379	0.9799	0.1615	0.0884	0.9480
dt	Decision Tree Regressor	13.2708	3643.8881	59.0516	0.9775	0.1321	0.0378	0.0710
catboost	CatBoost Regressor	39.4561	3699.6117	60.2848	0.9773	0.1870	0.1157	2.8020
lightgbm	Light Gradient Boosting Machine	43.3005	4522.7895	66.8946	0.9719	0.1948	0.1243	0.2690
gbr	Gradient Boosting Regressor	81.0017	12697.4905	112.4160	0.9218	0.2773	0.2120	0.5540
ada	AdaBoost Regressor	137.3587	29593.4900	171.9110	0.8188	0.4409	0.4479	0.2140
ridge	Ridge Regression	134.8746	35605.8614	188.4694	0.7834	0.4404	0.3199	0.0890
lr	Linear Regression	135.0288	35608.1357	188.4761	0.7834	0.4537	0.3214	0.4240
br	Bayesian Ridge	134.4792	35668.8338	188.6333	0.7831	0.4198	0.3142	0.0800
llar	Lasso Least Angle Regression	134.7040	35848.0928	189.1271	0.7820	0.4369	0.3106	0.1470
lasso	Lasso Regression	134.7040	35848.0928	189.1271	0.7820	0.4369	0.3106	0.0740
en	Elastic Net	140.8105	39149.0852	197.6221	0.7624	0.4173	0.3287	0.0750
huber	Huber Regressor	217.2652	108193.1933	311.5834	0.3686	0.5589	0.5604	0.1450
knn	K Neighbors Regressor	244.4454	121696.0742	347.8657	0.2620	0.6630	0.8633	0.0800

## 성능 비교 기준

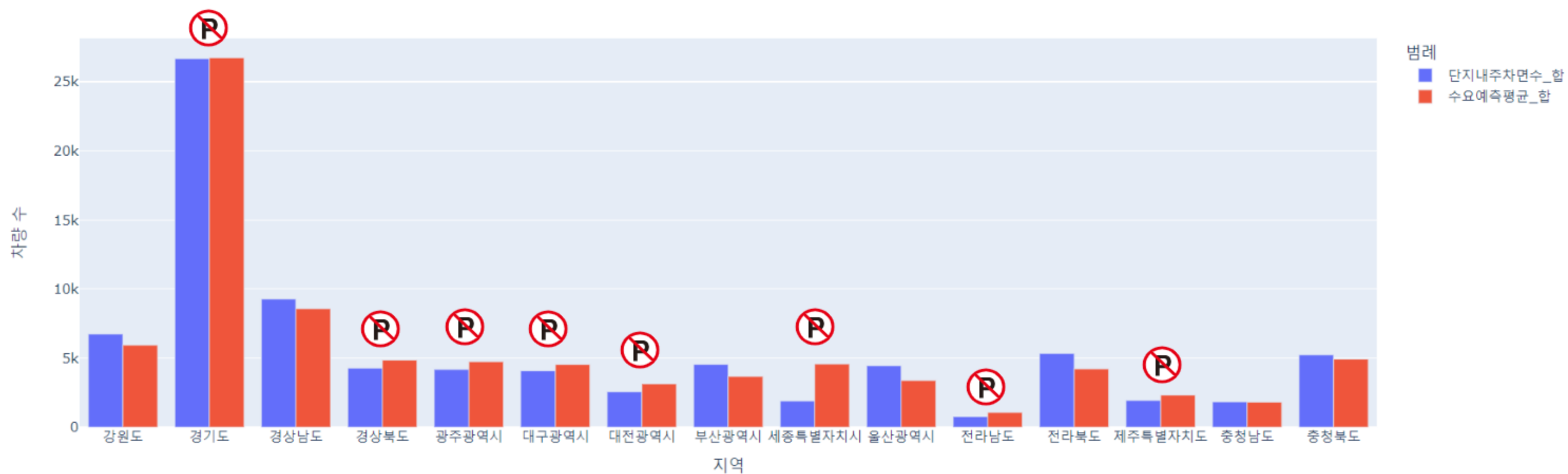
- MAE : 평균 절대 오차
- MSE : 평균 제곱 오차
- RMSE : 평균 제곱근 오차
- R2 : 결정계수
- RMSLE : RMSE에 로그 적용한 값
- MAPE : 평균 절대 비율 오차

## iii 최종 모델 선정 - ExtraTreesRegressor



## iv 예측 및 결과

지역별 단지내주차면수와 주차수요예측



## i 결론

과거에 비해 가구별 차량 보유가 증가하고 있지만, 주차와 관련된 법규나 규정은 바뀌는게 없음

주차 공간 확대, 주차관제 시스템 도입 등의 법 개정 필요

### 본 데이터 분석의 한계점

임대 아파트의 데이터 만으로 일반화하기에는 오류가 있어, 더 많은 데이터로 분석할 필요있음

## ii 출처

[Train] C1786 | 강원도 | 행복주택 | 총 480세대 : 춘천거두2 행복주택

춘천거두2 행복주택 임대조건 : <https://www.myhome.go.kr/hws/portal/sch/selectRsdtrcritNtcDetailView.do?pblancId=1425>

[Train] C1326 | 부산광역시 | 국민임대 | 총 1934세대 : 부산정관 7단지 A-1블록

부산정관 7단지 A-1BL 국민임대 모집공고 : <https://www.myhome.go.kr/hws/portal/sch/selectRsdtrcritNtcDetailView.do?pblancId=915>

[Train] C2186 | 대구광역시 | 국민임대 | 총 924세대 : 대구연경 A-2블록 국민임대주택

대구연경 A-2블록 국민임대주택 입주자모집 안내 : [https://m.blog.naver.com/PostView.nhn?](https://m.blog.naver.com/PostView.nhn?isHttpsRedirect=true&blogId=sugar9025404&logNo=221577519654&categoryNo=59&proxyReferer=)

[isHttpsRedirect=true&blogId=sugar9025404&logNo=221577519654&categoryNo=59&proxyReferer=](https://m.blog.naver.com/PostView.nhn?isHttpsRedirect=true&blogId=sugar9025404&logNo=221577519654&categoryNo=59&proxyReferer=)

[Test] C2152 | 강원도 | 영구임대 | 총 120세대 : 화천신읍 공공실버주택

화천신읍 공공실버주택 예비입주자 모집 : <https://www.myhome.go.kr/hws/portal/sch/selectRsdtrcritNtcDetailView.do?pblancId=9070>

[Test] C1267 | 경상남도 | 행복주택 | 총 675세대 : 창원가포 A-1블록 행복주택

창원가포 A-1블록 행복주택 입주자 모집공고 : <https://www.myhome.go.kr/hws/portal/sch/selectRsdtrcritNtcDetailView.do?pblancId=6373>

[Test] C1006 | 대전광역시 | 영구임대 | 총 1505세대 : 대전둔산1단지

대전둔산1 영구임대주택 그린리모델링세대 입주자 자격완화 모집 :

<https://www.myhome.go.kr/hws/portal/sch/selectRsdtrcritNtcDetailView.do?pblancId=8399>

# 이대로 관참은가 대한민국!

같이 해결해 봐요

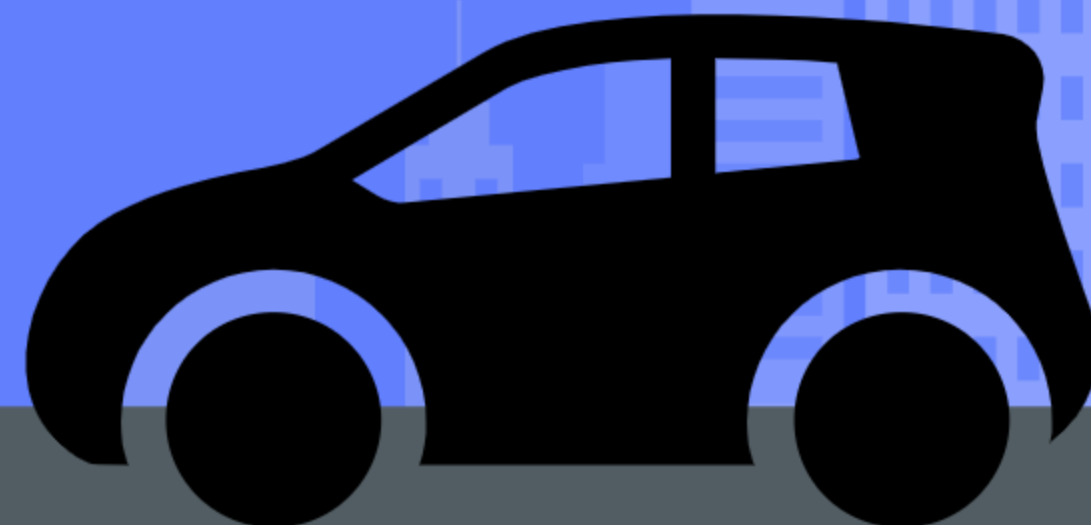
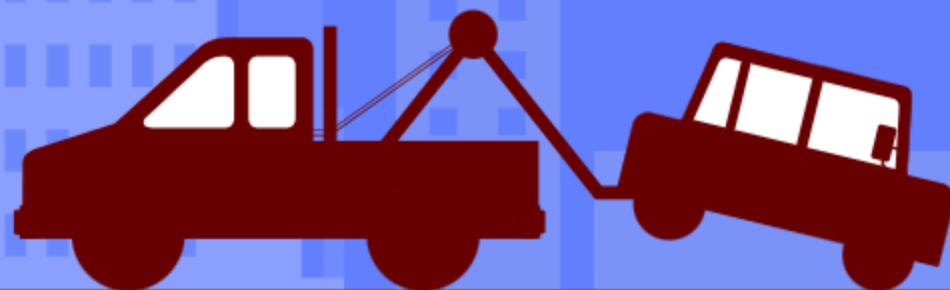
이승현  
ihwh99@gmail.com

나혜원  
n417759@gmail.com

김성현  
ksung1023@gmail.com

김민정  
alswjd4823@naver.com

권성재  
chris123ag@naver.com



PLAY DATA

