

건물 전력사용량 예측 프로젝트

빅데이터 분석 미니 프로젝트01

DACON 전력사용량 예측 대회 참여



프로젝트 기간

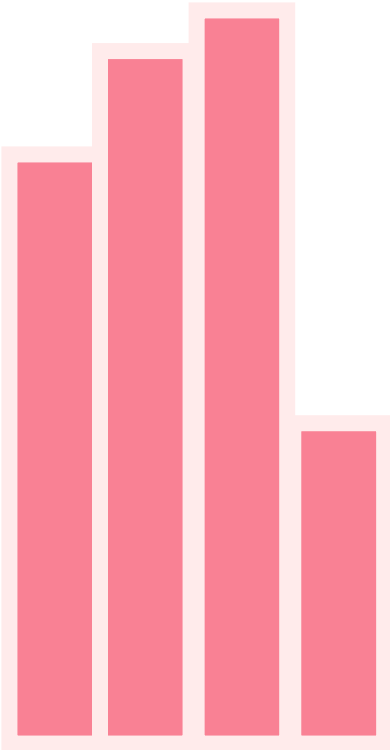
2023년 8월 9일 ~ 2023년 8월 21일

발표일

2023년 8월 21일

팀

일렉트릭쇼크



CONTENTS

Chapter. 1

프로젝트의 이해

Chapter. 2

선행 연구 검토

Chapter. 3

데이터 준비

Chapter. 4

모델링

Chapter. 5

평가

프로젝트 주제 선정

전력수요 예측, 기상 거대자료(빅데이터)로 오차 줄여

등록부서 : 기상융합서비스팀 | 2014/08/21

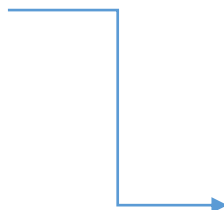
조회수 6750

전력수요 예측, 기상 거대자료(빅데이터)로 오차 줄여

- 기상 거대자료(빅데이터) 활용 시, 연간 1,200억 원의 경제적 효과
- 전력수요 예측 오차를 무려 25%나 개선할 수 있어 효과 특특히

출처 : 기상청 홈페이지

누적된 기상 데이터를 활용하여 전력 예측의 오차범위를 줄임



데이터 분석의 대표적인 활용 사례

프로젝트 개요 (2023 전력사용량 예측 AI 경진대회)

프로젝트 목표 및 KPI

- 전력사용량 예측 모델 생성
- SMAPE기준 0.1 미만 목표

규칙

- 대회 제공 데이터 이외에 외부 데이터 사용 금지
- 사용에 법적 제한이 없으며 논문으로 공개된 베이스의 사전 학습 모델 (Pre-trained Model) 사용 가능
- 사용 가능 언어: 파이썬, R

평가

- 심사기준: SMAPE
- 2022-08-25 ~ 2022-08-31의 실제 전력사용량 데이터로 평가
- (Kdigital) 미니 프로젝트 발표 일시 2023년 8월 21일 오후
- (DACON) 1차 제출 마감 일시 2023년 8월 23일 10:00

프로젝트 수행 계획

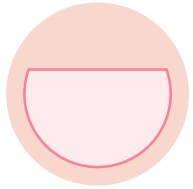


목적 및 배경

대회 참가 & 미니 프로젝트 수행

데이터 탐색 및 통계 모형 수립, 머신 러닝 분석을 경험

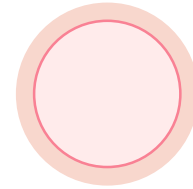
대회에 참가하여 커리어를 지원



기대 효과

데이터 분석 역량의 내재화 (기초통계 분석, 결측치 핸들링, 이상치 핸들링, 통계 모형과 머신 러닝 모형 만들기, 정리하여 발표하기)

SMAPE와 같은 평가 지표를 활용하여 데이터 분석의 전략적인 마인드 세팅



수행 방법

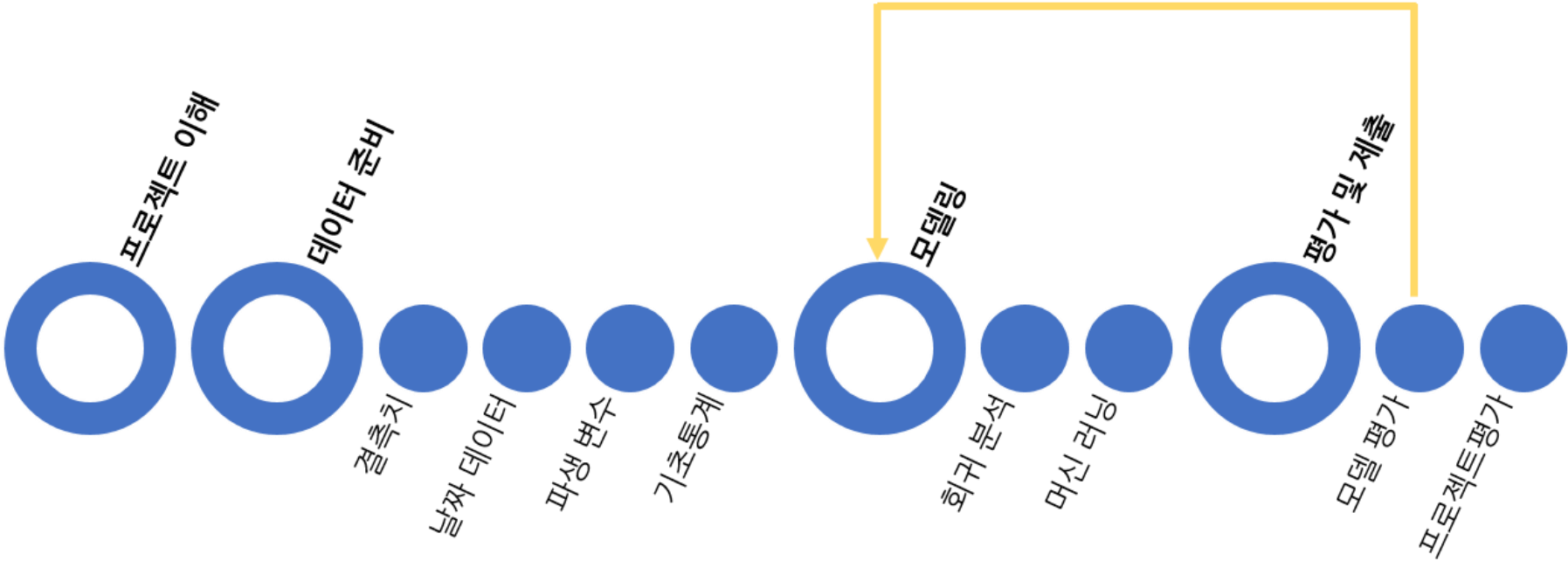
CRISP-DM을 참조하여 수행 방법을 정의

각 단계 간 피드백으로 단계별 완성도를 높임

프로젝트 이해, 데이터 준비, 모델링, 평가 순서로 프로젝트 수행

프로젝트 수행 계획

프로젝트 진행 과정



프로젝트 수행 계획

[illegible]

지난 대회 수상작 분석

코드 분석 및 데이터 탐색 참고

전력사용량



전체

학습

코드 공유

토크

대회

유저

17건의 검색 결과가 있습니다



재작년 1등 코드 따라하기 XGB 코드공유
2023 전력사용량 예측 AI 경진대회



[태블로]ESG와 첨단AI산단 솔루션



taegu private 8위 5.24453
전력사용량 예측 AI 경진대회



Doge | public 1st, private 3rd
전력사용량 예측 AI 경진대회



private 4위 qwopqwop ligbm + catboost
전력사용량 예측 AI 경진대회



j_sean팀 | Private 1위(5.0293) | XGBoost 단일 모형
전력사용량 예측 AI 경진대회



[Analytics] 234팀, 건물별로 전력사용량 분석하기
전력사용량 예측 AI 경진대회



[Analytics] 하르딘, 제공데이터 최대한 활용하기
전력사용량 예측 AI 경진대회

관련 연구 분석
(건물 에너지 사용량 예측)

사용 모델 및 평가 스코어

Muhammad Faiq 외 (2023)
C. Robinson 외 (2017)
Z. Wang 외 (2018)
S. Karatasou 외 (2006)

사용 모델 및
평가 스코어

Table 7. Summary of performance evaluation on the models.

Model	MAE (kWh)	RMSE (kWh)
LSTM	165.203	572.545
SVR	2851.339	3270.836
GPR	999.880	1310.105

변수 (일별)

기압
온도
상대 습도
풍속
강우 기간
강우량
날 유형
COVID로 건물 폐쇄일

Table 4. Weather variables and values range at the MET Melaka Weather Station.

Variable	Abbreviation	Type	Measurement	Range of values
Pressure	Press	Continuous	hPa	1006 – 1014
Environmental temperature	Temp	Continuous	Deg. C	22 – 31
Relative humidity	Hum	Continuous	%	58 – 94
Wind velocity	Wind	Continuous	m/s	0.5 – 5.3
Rainfall duration	RainDur	Continuous	Minutes	0 – 60
Rainfall amount	RainAm	Continuous	mm	–14 – 4
Type of day	Day	Categorical	Weekday, weekend and holiday	0 for weekday and 1 for weekend and holiday
Type of lockdown	Lock	Categorical	No MCO, MCO and RMCO	0 for no MCO, 1 for MCO and 2 for RMCO

관련 연구 분석

(건물 에너지 사용량 예측)

중요 변수 리스트

Machine learning approaches for estimating commercial building energy consumption

<ul style="list-style-type: none">- Principal building activity- Square footage- Number of floors- Heating degree days- Cooling degree days	<ul style="list-style-type: none">- 건물 종류- 면적- 층수- 난방 일수- 냉방 일수
---	---

Z. Wang, Y. Wang, R. Zeng, R.S. Srinivasan, S. Ahrentzen Random Forest based hourly building energy prediction

<ul style="list-style-type: none">- Outdoor temperature- Dew point- Relative humidity- Brometric pressure- Precipitation- Wind speed- Solar radiation- Number of occupants- Time of day- Workday type- Day type	<ul style="list-style-type: none">- 온도- 이슬점- 습도- 기압- 강수량- 풍속- 태양 복사- 거주자 수- 시간- 워킹 데이- 요일
---	---

Modeling and predicting building's energy use with artificial neural networks: Methods and results

<ul style="list-style-type: none">- Ambient temperature- Solar flux- Humidity- Hour of the day- Day of the week- Day of the year	<ul style="list-style-type: none">- 온도- 일사- 습도- 시간- 요일- 년도
---	---

Muhammad Faiq 외 (2023)
C. Robinson 외 (2017)
Z. Wang 외 (2018)
S. Karatasou 외 (2006)

건물 단위로 모델을 설정했을 때의 장점

1. 개별적인 특성 반영

각 건물은 개별적인 특성을 가질 수 있으므로 이러한 특성을 고려하여 건물 별로 모델을 설정하면, 각 건물이 가진 특성을 더 정확하게 모델링 할 수 있다.

2. 상황별 모델링

건물마다 용도, 구조, 위치 등의 차이로 인해 전력 소비나 사용 패턴이 다를 수 있다. 그룹 별로 모델을 만들면 이러한 상황을 고려하여 최적화된 모델을 개발할 수 있다.

3. 변수 중요도 강조

건물 별로 모델을 설정하면, 각 건물이나 그룹별로 중요한 변수를 식별할 수 있다.

특정 건물이나 그룹에만 해당하는 중요한 변수들을 모델에 포함시키면 해당 건물의 예측 능력을 향상시킬 수 있다.

4. 모델의 해석 용이성

그룹 별로 모델을 만들 경우, 각 건물이나 그룹의 특성을 모델 결과와 연결하여 해석할 수 있다.

모델의 예측 결과를 해당 건물의 특성과 연결하여 해석하면서 의사 결정에 활용할 수 있다.

5. 데이터 부족 상황 대응

특정 건물의 데이터가 다른 건물에 비해 부족한 경우, 해당 건물을 대상으로 모델을 설정하여 데이터 부족 상황을 극복할 수 있다.

6. 특정 건물 관리 및 최적화

특정 건물의 에너지 사용량을 관리하거나 최적화하는 것이 목표일 경우, 해당 건물 별로 모델을 설정하여 개별 관리 및 개선 전략을 수립할 수 있다.

훈련 대상 데이터셋(1/2) – 건물번호로 연결

- train 데이터 : 100개 건물들의 **2022년 06월 01일부터 2022년 08월 24일**까지의 데이터
- 일시별 기온, 강수량, 풍속, 습도, 일조, 일사 정보 포함
- 전력사용량(kWh) 포함

□	num_date_time ▾	건물번호 ▾	일시 ▾	기온(C) ▾	강수량(mm) ▾	풍속(m/s) ▾	습도(%) ▾	일조(hr) ▾	일사(MJ/m... ▾	전력소비량(kWh) ▾
1	1_20220601 00	1	20220601 00	18.6		0.9	42.0			1085.28
2	1_20220601 01	1	20220601 01	18.0		1.1	45.0			1047.36
3	1_20220601 02	1	20220601 02	17.7		1.5	45.0			974.88
4	1_20220601 03	1	20220601 03	16.7		1.4	48.0			953.76
5	1_20220601 04	1	20220601 04	18.4		2.8	43.0			986.4
6	1_20220601 05	1	20220601 05	17.2		2.1	46.0			1087.2
7	1_20220601 06	1	20220601 06	16.3		1.0	50.0	0.0	0.05	1314.72
8	1_20220601 07	1	20220601 07	17.4		1.3	50.0	1.0	0.55	1684.8

훈련 대상 데이터셋(1/2) - 건물번호로 연결

- 100개 건물 정보
- 건물 번호, 건물 유형, 연면적, 냉방 면적, 태양광 용량, ESS 저장 용량, PCS 용량

<input type="checkbox"/>	건물번호 ▾	건물유형 ▾	연면적(m2) ▾	냉방면적(m2) ▾	태양광용량(kW) ▾	ESS저장용량(kWh) ▾	PCS용량(kW) ▾
1	1	건물기타	110634.00	39570.00	-	-	-
2	2	건물기타	122233.47	99000.00	-	-	-
3	3	건물기타	171243.00	113950.00	40	-	-
4	4	건물기타	74312.98	34419.62	60	-	-
5	5	건물기타	205884.00	150000.00	-	2557	1000
6	6	건물기타	205754.00	74565.00	-	-	-
7	7	건물기타	101711.52	41341.10	-	800	300
8	8	건물기타	75344.54	24117.00	-	-	-

예측 대상 데이터셋

- test 데이터 : 100개 건물들의 **2022년 08월 25일부터 2022년 08월 31일**까지의 데이터
- 일시별 기온, 강수량, 풍속, 습도의 예보 정보

<input type="checkbox"/>	num_date_time ▾	건물번호 ▾	일시 ▾	기온(C) ▾	강수량(mm) ▾	풍속(m/s) ▾	습도(%) ▾
3	1_20220825 02	1	20220825 02	22.7	0.0	1.5	75
4	1_20220825 03	1	20220825 03	22.1	0.0	1.3	78
5	1_20220825 04	1	20220825 04	21.8	0.0	1.0	77
6	1_20220825 05	1	20220825 05	21.6	0.0	1.6	81
7	1_20220825 06	1	20220825 06	21.5	0.0	2.3	84
8	1_20220825 07	1	20220825 07	21.7	0.0	1.4	83
9	1_20220825 08	1	20220825 08	22.3	0.0	2.2	82
10	1_20220825 09	1	20220825 09	22.7	0.0	2.4	78

제출용 데이터셋 – 예측값을 입력하여 제출하는 양식

- 제출을 위한 양식
- 100개 건물들의 **2022년 08월 25일부터 2022년 08월 31일**까지의 전력사용량(kWh)을 예측
- num_date_time은 건물번호와 시간으로 구성된 ID
- 해당 ID에 맞춰 전력사용량 예측값을 answer 컬럼에 기입해야 함

<input type="checkbox"/>	num_date_time	answer
1	1_20220825 00	0
2	1_20220825 01	0
3	1_20220825 02	0
4	1_20220825 03	0
5	1_20220825 04	0
6	1_20220825 05	0
7	1_20220825 06	0
...



결측치 핸들링 - 강수량, 일조, 일사, 풍속, 습도(train)

- 강수량이 결측치인 관측(74629 ~ 74636)의 습도가 강수량이 결측치가 아닌 관측(74637 ~ 74643)의 습도보다 낮은 경향을 보임
- 모든 관측치에서 같은 경향을 확인
- 강수량이 결측치인 경우 비가 내리지 않아 측정되지 못한 것 이라고 해석
- 강수량 결측치 NA를 0으로 대체**

	num_date_time	건물번호	일시	기온.C.	강수량.mm.	풍속.m.s.	습도...	일조.hr.	일사.MJ.m2.	전력소비량.kWh.
74629	37_20220720 12	37	20220720 12	29.6	NA	2.6	67	0.6	2.54	5698.8
74630	37_20220720 13	37	20220720 13	30.0	NA	2.9	62	0.4	2.26	5691.6
74631	37_20220720 14	37	20220720 14	30.1	NA	3.4	60	0.6	2.56	5638.2
74632	37_20220720 15	37	20220720 15	30.2	NA	3.2	62	0.8	2.61	5586.0
74633	37_20220720 16	37	20220720 16	29.6	NA	2.8	60	0.1	1.48	5559.6
74634	37_20220720 17	37	20220720 17	29.0	NA	2.4	64	0.0	0.79	5484.6
74635	37_20220720 18	37	20220720 18	28.5	NA	2.2	67	0.0	0.52	4917.6
74636	37_20220720 19	37	20220720 19	27.9	NA	1.9	69	0.0	0.16	4648.2
74637	37_20220720 20	37	20220720 20	27.3	0.0	1.8	74	0.0	0.01	2604.0
74638	37_20220720 21	37	20220720 21	26.9	0.0	1.6	75	NA	NA	2158.2
74639	37_20220720 22	37	20220720 22	26.2	0.3	1.4	81	NA	NA	1370.4
74640	37_20220720 23	37	20220720 23	25.3	0.0	1.8	79	NA	NA	1161.0
74641	37_20220721 00	37	20220721 00	24.3	0.8	1.7	85	NA	NA	1147.2
74642	37_20220721 01	37	20220721 01	24.0	0.9	2.8	91	NA	NA	1090.8
74643	37_20220721 02	37	20220721 02	23.6	0.5	2.5	92	NA	NA	1069.2

결측치 핸들링 - 강수량, 일조, 일사, 풍속, 습도(train)

- 일조, 일사가 NA인 관측치들은 대부분 밤 시간대로 추정
- 같은 시간대에 NA가 아닌 경우에도 0으로 관측된 측정값 존재
- 일조: 태양의 직사광이 지표면에 비친 시간
- 일사: 태양복사에너지가 지표에 닿는 양
- 두 속성 모두 태양빛이 없는 밤 시간대에는 측정 불가
- 일조, 일사 결측치 NA를 0으로 대체

	num_date_time	일 조.hr.	일 사.MJ.m2.
119312	59_20220712 07	0.6	0.43
119313	59_20220712 08	0.0	0.53
119314	59_20220712 09	0.1	1.09
119315	59_20220712 10	0.0	1.15
119316	59_20220712 11	0.1	1.65
119317	59_20220712 12	0.0	1.50
119318	59_20220712 13	0.0	1.33
119319	59_20220712 14	0.0	1.48
119320	59_20220712 15	0.0	1.10
119321	59_20220712 16	0.0	0.67
119322	59_20220712 17	0.0	0.47
119323	59_20220712 18	0.0	0.29
119324	59_20220712 19	0.0	0.07
119325	59_20220712 20	0.0	0.02
119326	59_20220712 21	0.0	0.00
119327	59_20220712 22	0.0	0.00
119328	59_20220712 23	0.0	0.00
119329	59_20220713 00	0.0	0.00
119330	59_20220713 01	0.0	0.00
119331	59_20220713 02	0.0	0.00
119332	59_20220713 03	0.0	0.00
119333	59_20220713 04	0.0	0.00
119334	59_20220713 05	0.0	0.00
119335	59_20220713 06	0.0	0.00
119336	59_20220713 07	0.0	0.03
119337	59_20220713 08	0.0	0.08
119338	59_20220713 09	0.0	0.13
119339	59_20220713 10	0.0	0.13

	num_date_time	일 조.hr.	일 사.MJ.m2.
178472	88_20220712 07	0.1	0.31
178473	88_20220712 08	0.0	0.47
178474	88_20220712 09	0.1	1.04
178475	88_20220712 10	0.0	1.20
178476	88_20220712 11	0.1	1.41
178477	88_20220712 12	0.2	1.72
178478	88_20220712 13	0.0	1.41
178479	88_20220712 14	0.0	1.17
178480	88_20220712 15	0.0	0.99
178481	88_20220712 16	0.0	0.93
178482	88_20220712 17	0.0	0.74
178483	88_20220712 18	0.0	0.50
178484	88_20220712 19	0.0	0.14
178485	88_20220712 20	0.0	0.04
178486	88_20220712 21	NA	NA
178487	88_20220712 22	NA	NA
178488	88_20220712 23	NA	NA
178489	88_20220713 00	NA	NA
178490	88_20220713 01	NA	NA
178491	88_20220713 02	NA	NA
178492	88_20220713 03	NA	NA
178493	88_20220713 04	NA	NA
178494	88_20220713 05	NA	NA
178495	88_20220713 06	0.0	0.00
178496	88_20220713 07	0.0	0.05
178497	88_20220713 08	0.0	0.06
178498	88_20220713 09	0.0	0.12
178499	88_20220713 10	0.0	0.08

낮

밤

낮

결측치 핸들링 - 강수량, 일조, 일사, 풍속, 습도(train)

건물번호		일시	풍속.m.s.	습도...
9	2022-06-14 11:00:00		NA	NA
15	2022-08-06 17:00:00		NA	NA
16	2022-08-03 15:00:00		NA	77
26	2022-06-27 16:00:00		NA	93
26	2022-07-09 09:00:00		NA	84
42	2022-07-03 10:00:00		NA	66
50	2022-07-03 10:00:00		NA	66
52	2022-08-06 15:00:00		NA	72
52	2022-08-06 16:00:00		NA	73
87	2022-07-14 05:00:00		NA	NA
87	2022-07-14 06:00:00		NA	NA
87	2022-07-14 07:00:00		NA	NA
87	2022-07-14 08:00:00		NA	NA
87	2022-07-14 09:00:00		NA	NA
90	2022-08-06 17:00:00		NA	NA
97	2022-08-06 15:00:00		NA	72
97	2022-08-06 16:00:00		NA	73
100	2022-06-08 15:00:00		NA	NA
100	2022-07-23 03:00:00		NA	99

```
train %>%
```

```
`filter(is.na(train$풍속.m.s.) | is.na(train$습도...)) %>%`  
select(건물번호, 일시, 풍속.m.s., 습도...)
```

- 19개의 풍속 결측치
- 9개의 습도 결측치
- 풍속이 결측인 관측은 습도가 결측인 관측을 포함
- 제거하거나 단순 대치하기 어려운 결측치
- 풍속이나 습도는 시간에 따라 연속적인 경향을 보임
- 풍속과 습도 데이터에 연속성을 가정하여 **회귀 대치법** 사용
- 시계열 속성이 존재해야 작동

```
library(zoo)
```

```
train$풍속.m.s. <- na.approx(train$습도...)
```

```
train$습도... <- na.approx(train$습도...)
```

시계열 데이터 – 시계열 속성으로 변환

- LSTM, ARIMA 모델 사용시 시계열 속성 필요

```
> str(train$일시)
chr [1:204000] "20220601 00" "20220601 01" "20220601 02" ...
```

```
train$일시 <- as.POSIXct(train$일시, format="%Y%m%d %H")
```

```
> str(train$일시)
POSIXct[1:204000], format: "2022-06-01 00:00:00" "2022-06-01 01:00:00" ...
```

시계열 데이터 – 요일, 주중/휴일 변수 생성(WeekD, WorkD)

- WeekD : 범주형 변수
- WorkD : 범주형 또는 수치형 (1, 0)
- lubridate 패키지

```
install.packages("lubridate")
```

```
library(lubridate)
```

```
train$WeekD <- weekdays(train$일시)
```

```
train$WorkD <- ifelse(train$WeekD %in% c("토요일", "일요일"), 0,  
                      ifelse(format(train$일시, "%Y-%m-%d") %in% c("2022-06-01", "2022-06-06", "2022-08-15"), 0, 1))
```


훈련용 데이터 셋 – train + building_info

```
train <- building_info %>%
  select(건물번호, 건물유형, 냉방면적.m2.) %>%
  left_join(train, by="건물번호")
```

```
train <- train %>%
  select(건물번호, 일시, 전력소비량.kWh., 기온.C., 강수량.mm.,
  풍속.m.s., 습도..., 일사.MJ.m2., 냉방면적.m2., WorkD, WeekD, 건물유형)
```

연속형

범주형



	건물 번호	일시	전력소 비 량.kWh.	기 온.C.	강수 량.mm.	풍 속.m.s.	습 도...	일 사.MJ.m2.	냉방면 적.m2.	WorkD	WeekD	건물 유형
31934	16	2022-07-26 13:00:00	5002.08	29.5	0.0	2.5	68.0	2.21	95175	1	화요일	공공
31935	16	2022-07-26 14:00:00	4814.88	30.1	0.0	3.1	67.0	2.70	95175	1	화요일	공공
31936	16	2022-07-26 15:00:00	4878.24	29.6	0.0	3.0	70.0	3.10	95175	1	화요일	공공
31937	16	2022-07-26 16:00:00	4916.16	29.9	0.0	2.5	68.0	2.56	95175	1	화요일	공공
31938	16	2022-07-26 17:00:00	4890.72	29.5	0.0	2.7	69.0	1.94	95175	1	화요일	공공
31939	16	2022-07-26 18:00:00	4561.44	29.5	0.0	2.5	69.0	1.15	95175	1	화요일	공공
31940	16	2022-07-26 19:00:00	4664.16	27.1	0.0	2.2	74.0	0.41	95175	1	화요일	공공

훈련 세트 변수 정리

변수	변수명	단위	데이터 타입	범위
건물 번호	BID		범주형	1 ~ 100
일시	DateTime	2022-06-01 00:00:00 형태	시계열	2022년 6월 1일 0시 ~ 2022년 8월 17일 23시
기온	Temp	C. 섭씨 온도	연속형	10.10 ~ 37.10
강수량	Rain	mm	연속형	0 ~ 92.2
풍속	WS	m/s	연속형	0 ~ 13.3
습도	HM	%	연속형	13 ~ 100
일조	SS	hr	연속형	0~1
일사	SR	MJ/m^2	연속형	0 ~ 3.92
전력소비량	PC	kWh	연속형	0 ~ 25488
시간	Time	시	범주형	0 ~ 23 각 7800개
요일	WeekD	월요일 ~ 일요일	범주형	월요일 ~ 일요일 각 26400개
워킹 데이	WorkD	평일: 1. 주말, 공휴일: 0	범주화 수치	0: 60000개, 1: 127200개
체감온도	WC		연속형	11.67 ~ 40.30
불쾌지수	DI		연속형	50.48 ~ 87.79
불쾌지수(범주형)	DILv	Comfortable, Uncomfortable, Very Uncomfortable, Discomfort	범주형(순서가 있는)	Comfortable: 22313개, Uncomfortable: 72911개 Very Uncomfortable: 38130개, Discomfort: 53846개

상관성 확인 - 상관성 높은 변수만 추출

- 숫자형 데이터만 추출하여 상관성 확인

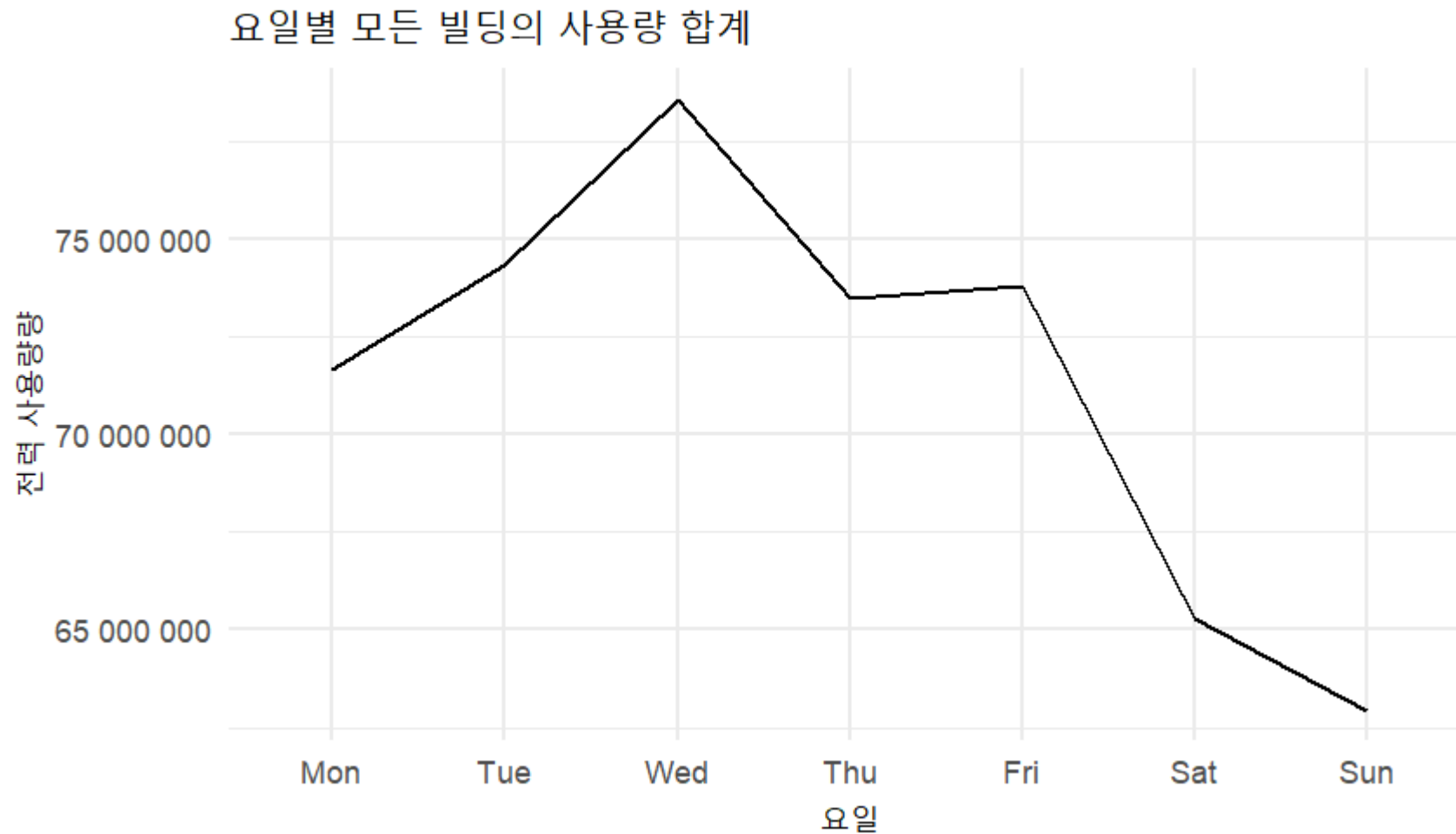
building_r	temperatu	rainfall	windspee	humidity	sunshine	solar_radi	power_co	month	day	hour	total_area	cooling_ai	solar_pow	ess_capac	pcs_capac	solar	ess	pcs	holiday
1	-0.0045	-0.00413	0.021461	0.022994	0.017324	-0.00903	-0.18047	0	-2.14E-22	0	-0.11532	-0.1162	-0.04557	-0.12468	-0.17229	-0.1761	-0.20584	-0.20584	4.05E-23
-0.0045	1	-0.06864	0.220236	-0.38989	0.419242	0.463308	0.17119	0.4272	0.069498	0.291429	-0.00334	-0.00306	0.006434	0.011059	0.008055	0.013545	0.004943	0.004943	-0.04277
-0.00413	-0.06864	1	0.048084	0.199563	-0.10048	-0.09937	0.015083	0.043436	0.027107	0.013729	-0.00263	-0.00275	0.005575	0.004215	0.005436	0.001401	0.007118	0.007118	0.067913
0.021461	0.220236	0.048084	1	-0.21911	0.167914	0.225474	0.109956	-0.0863	0.100171	0.177125	-0.05501	-0.05379	-0.05036	-0.00369	-0.01261	-0.07494	-0.0237	-0.0237	0.045767
0.022994	-0.38989	0.199563	-0.21911	1	-0.57244	-0.56124	-0.12955	0.253898	0.118432	-0.28061	-0.02012	-0.02052	-0.00084	-0.04695	-0.0492	-0.0256	-0.0527	-0.0527	0.093861
0.017324	0.419242	-0.10048	0.167914	-0.57244	1	0.764701	0.094748	-0.04952	-0.07541	0.159753	0.003834	0.0041	-0.01533	-0.00874	-0.00941	0.002448	-0.00654	-0.00654	-0.01366
-0.00903	0.463308	-0.09937	0.225474	-0.56124	0.764701	1	0.17775	-0.03801	-0.05684	0.152989	-0.06576	-0.06483	-0.01983	0.011289	0.010919	0.009663	0.012078	0.012078	-0.01098
-0.18047	0.17119	0.015083	0.109956	-0.12955	0.094748	0.17775	1	0.055842	0.009233	0.099594	0.013463	0.016444	0.052039	-0.03186	-0.03388	0.16432	-0.05995	-0.05995	0.068082
0	0.4272	0.043436	-0.0863	0.253898	-0.04952	-0.03801	0.055842	1	-0.13332	0	-7.70E-19	8.30E-19	1.60E-18	2.53E-18	-2.24E-18	-1.51E-18	-3.99E-18	-3.99E-18	0.034825
-2.14E-22	0.069498	0.027107	0.100171	0.118432	-0.07541	-0.05684	0.009233	-0.13332	1	0	9.18E-21	5.56E-20	2.97E-19	-1.12E-19	1.89E-21	5.05E-19	5.50E-19	5.50E-19	0.037557
0	0.291429	0.013729	0.177125	-0.28061	0.159753	0.152989	0.099594	0	0	1	-1.01E-24	9.50E-24	0	-1.17E-23	0	0	-1.13E-23	-1.13E-23	0
-0.11532	-0.00334	-0.00263	-0.05501	-0.02012	0.003834	-0.06576	0.013463	-7.70E-19	9.18E-21	-1.01E-24	1	0.998884	-0.03429	-0.02174	-0.02277	-0.06188	-0.02716	-0.02716	-1.29E-20
-0.1162	-0.00306	-0.00275	-0.05379	-0.02052	0.0041	-0.06483	0.016444	8.30E-19	5.56E-20	9.50E-24	0.998884	1	-0.03405	-0.0192	-0.01971	-0.05926	-0.02374	-0.02374	3.02E-20
-0.04557	0.006434	0.005575	-0.05036	-0.00084	-0.01533	-0.01983	0.052039	1.60E-18	2.97E-19	0	-0.03429	-0.03405	1	-0.0126	-0.01801	0.530358	-0.0222	-0.0222	5.19E-20
-0.12468	0.011059	0.004215	-0.00369	-0.04695	-0.00874	0.011289	-0.03186	2.53E-18	-1.12E-19	-1.17E-23	-0.02174	-0.0192	-0.0126	1	0.969688	0.03966	0.874992	0.874992	4.40E-20
-0.17229	0.008055	0.005436	-0.01261	-0.0492	-0.00941	0.010919	-0.03388	-2.24E-18	1.89E-21	0	-0.02277	-0.01971	-0.01801	0.969688	1	0.026242	0.925021	0.925021	-2.69E-20
-0.1761	0.013545	0.001401	-0.07494	-0.0256	0.002448	0.009663	0.16432	-1.51E-18	5.05E-19	0	-0.06188	-0.05926	0.530358	0.03966	0.026242	1	0.019118	0.019118	2.43E-19
-0.20584	0.004943	0.007118	-0.0237	-0.0527	-0.00654	0.012078	-0.05995	-3.99E-18	5.50E-19	-1.13E-23	-0.02716	-0.02374	-0.0222	0.874992	0.925021	0.019118	1	1	8.33E-20
-0.20584	0.004943	0.007118	-0.0237	-0.0527	-0.00654	0.012078	-0.05995	-3.99E-18	5.50E-19	-1.13E-23	-0.02716	-0.02374	-0.0222	0.874992	0.925021	0.019118	1	1	8.33E-20
4.05E-23	-0.04277	0.067913	0.045767	0.093861	-0.01366	-0.01098	0.068082	0.034825	0.037557	0	-1.29E-20	3.02E-20	5.19E-20	4.40E-20	-2.69E-20	2.43E-19	8.33E-20	8.33E-20	1

ess_capacity, ess, total_area
변수 제거

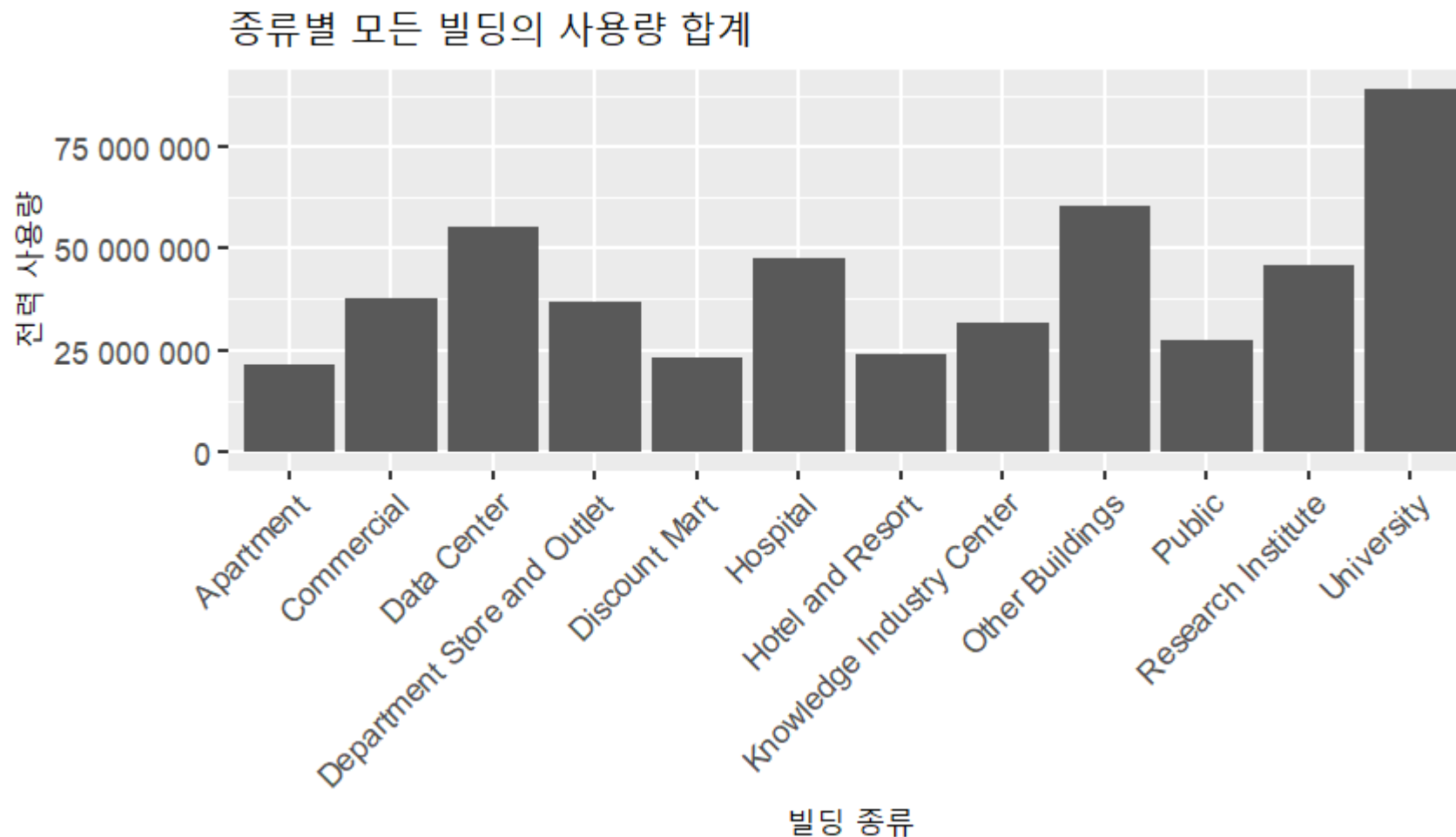
데이터 탐색 – 전체 건물의 전력 사용량은 어떤 특징이 있는가?



데이터 탐색 - 전체 건물의 전력 사용량은 어떤 특징이 있는가?

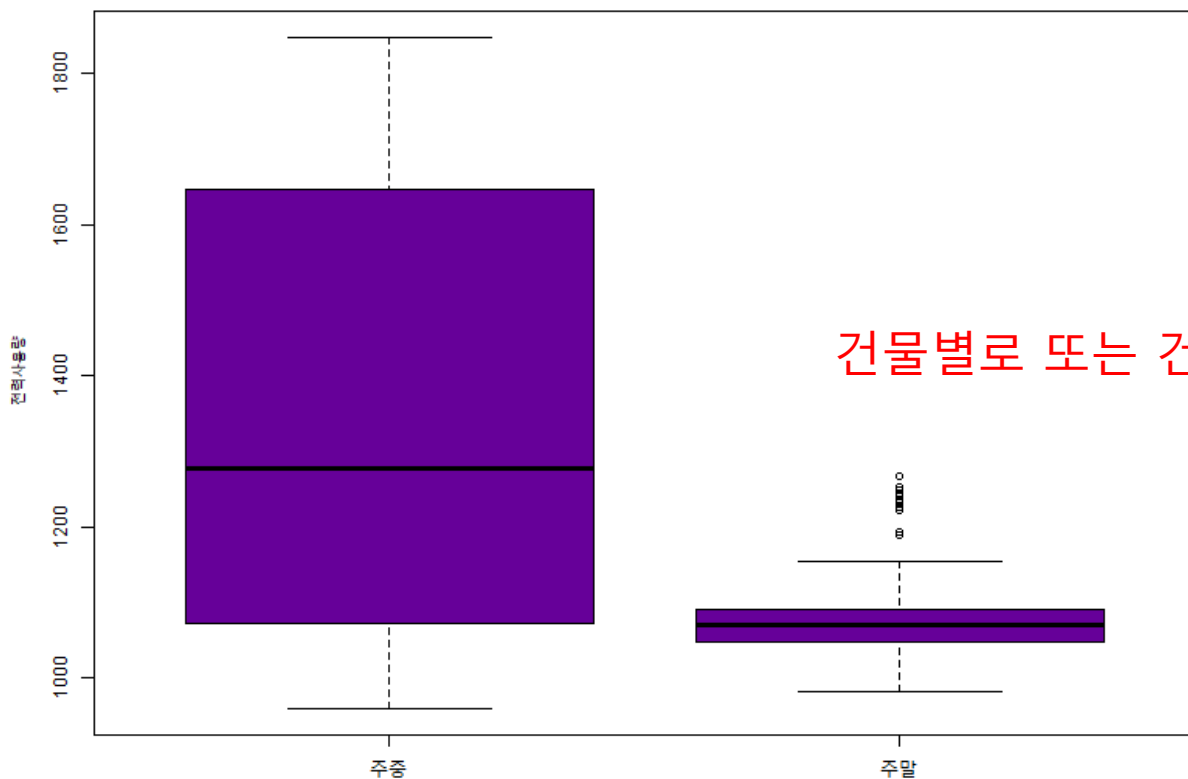


데이터 탐색 - 건물 종류별 전력 사용량



데이터 탐색 - 각 건물은 주중/휴일을 기준으로 전력 사용량에 차이가 존재할까?

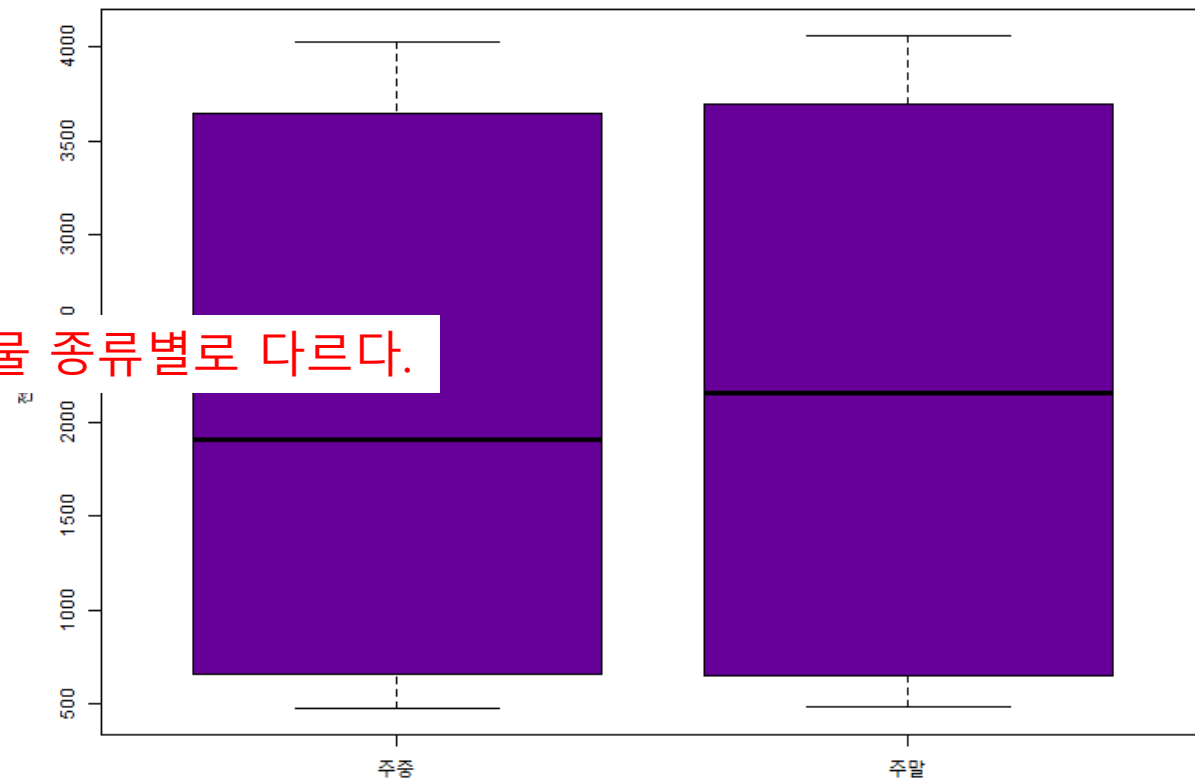
공공 BID 23 워킹데이별 전력사용량 분포



ttest : two.sided
p-value = 2.90578e-214

건물별로 또는 건물 종류별로 다르다.

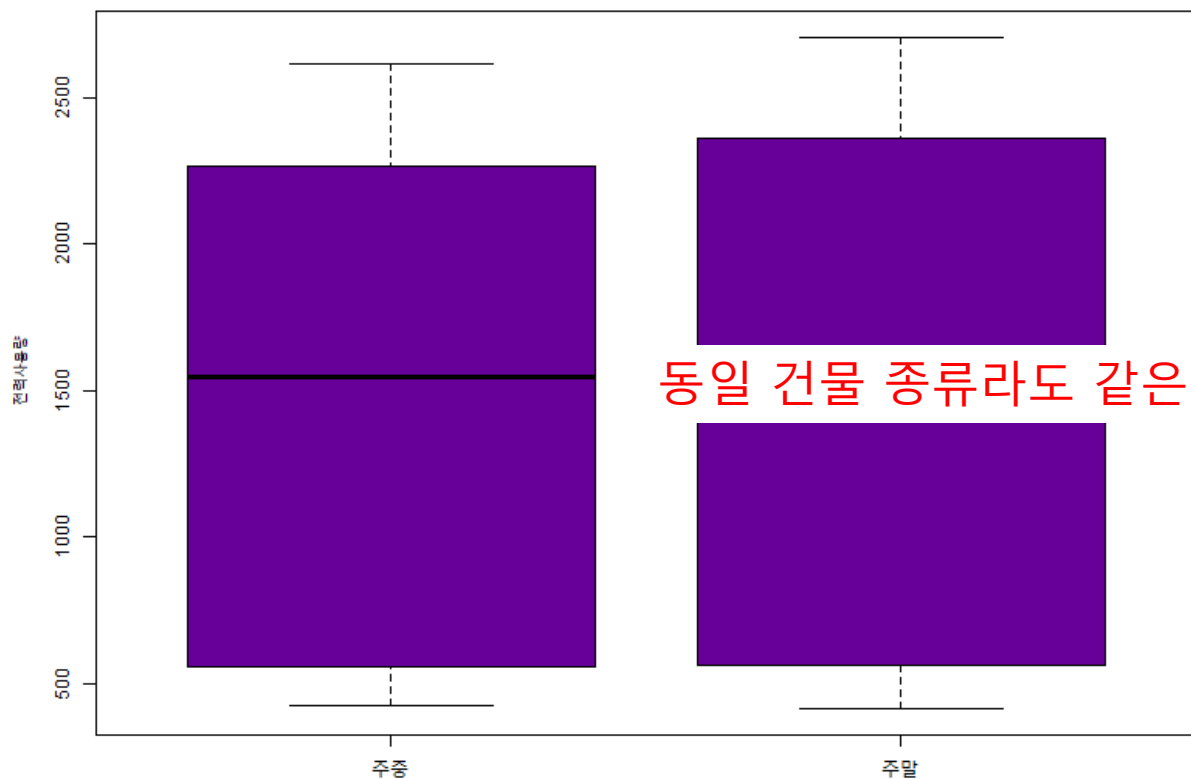
백화점및아울렛 BID 39 워킹데이별 전력사용량 분포



ttest : two.sided
p-value = 0.2466852

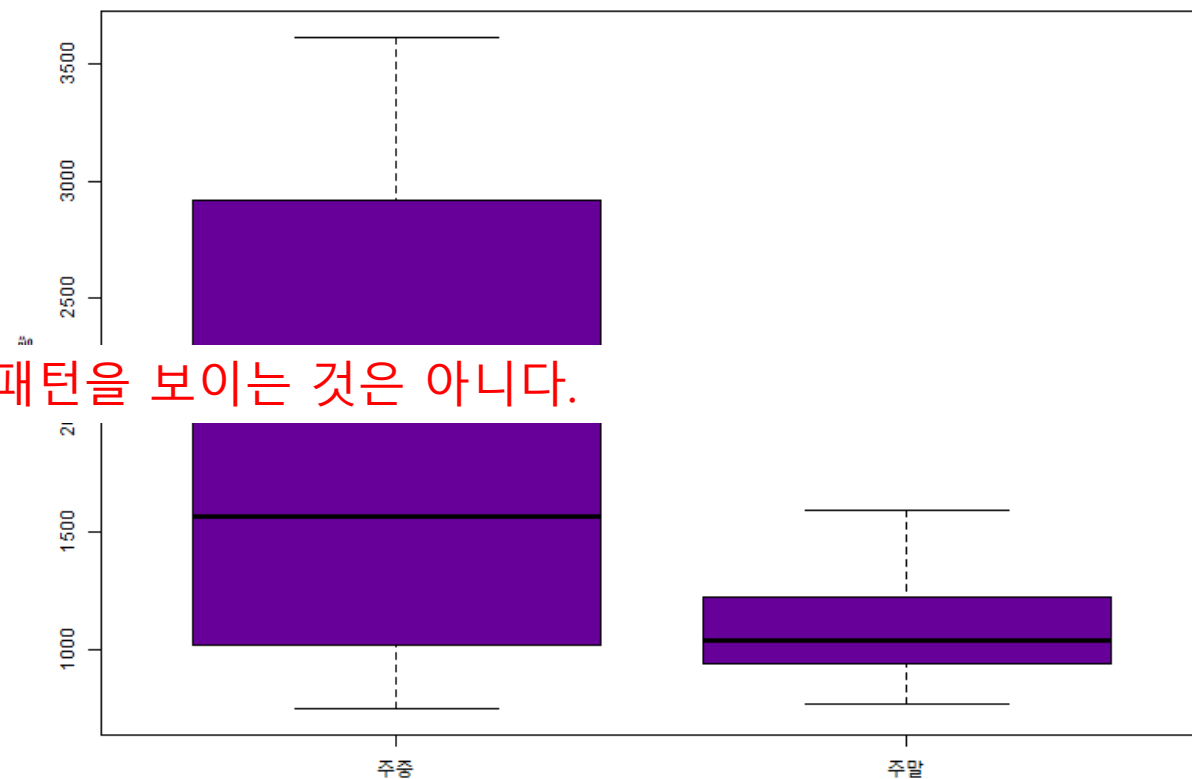
데이터 탐색 - 건물 종류별로 같은 패턴을 보이는가?

지식산업센터 BID 81 워킹데이별 전력사용량 분포



ttest: two.sided
p-value = 0.1861687

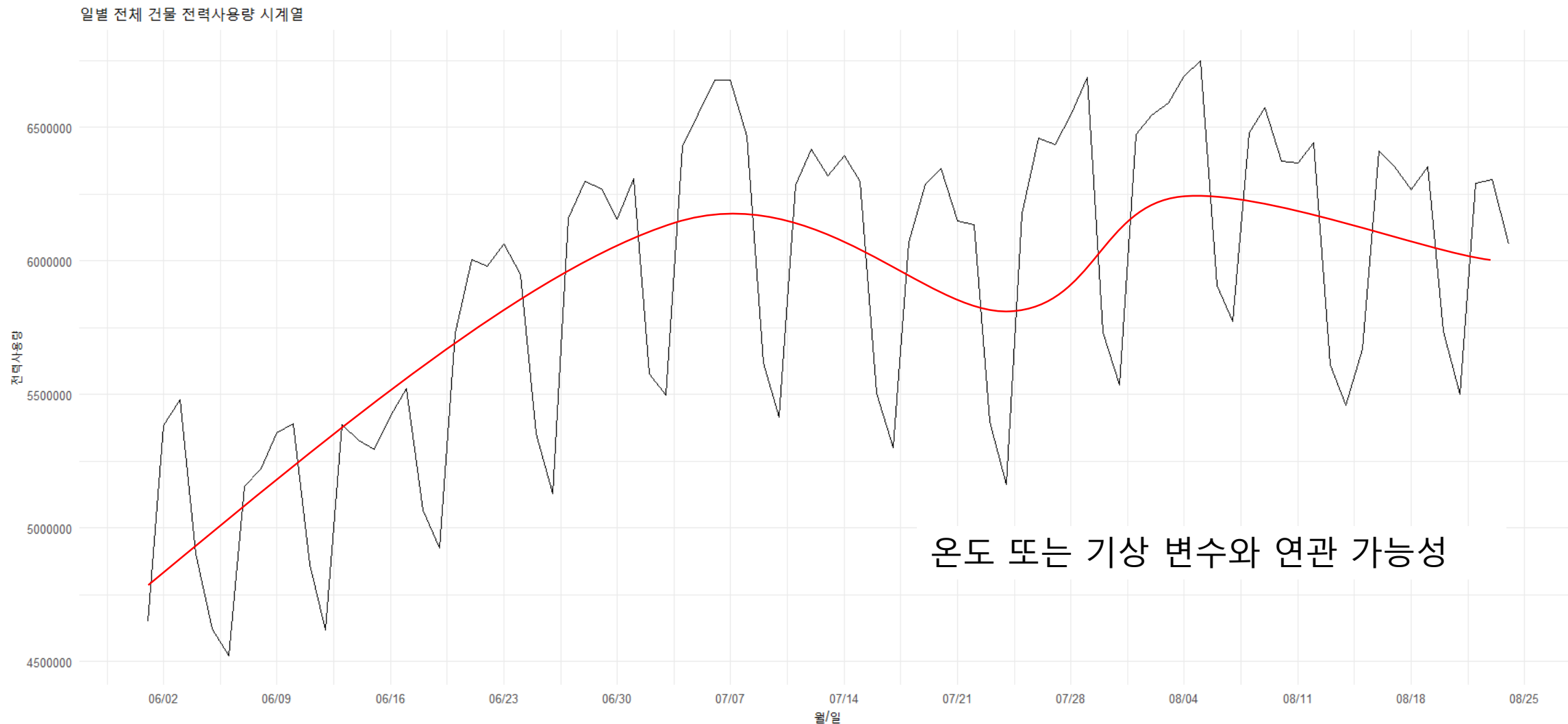
지식산업센터 BID 78 워킹데이별 전력사용량 분포



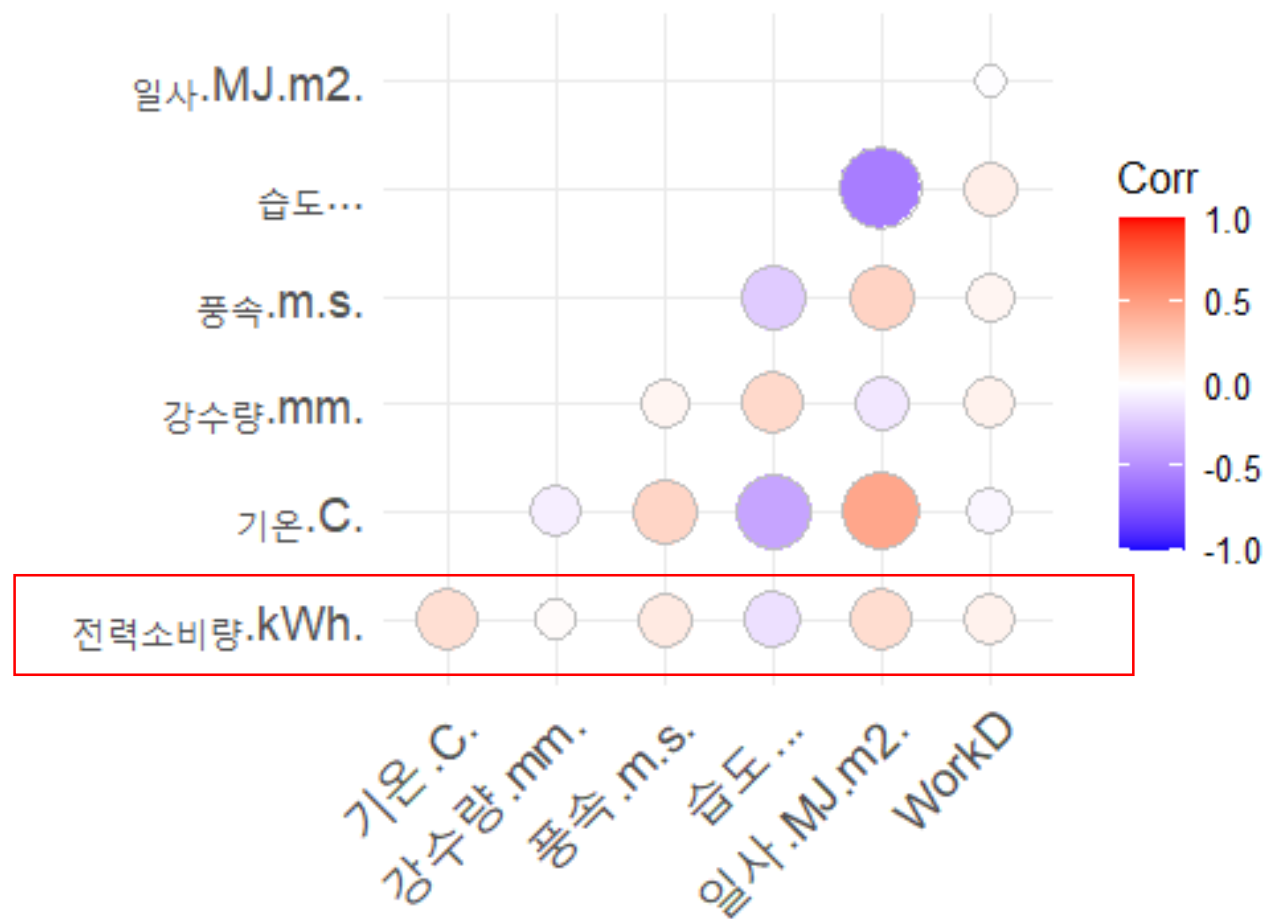
ttest: two.sided
p-value = 7.688675e-179

동일 건물 종류라도 같은 패턴을 보이는 것은 아니다.

데이터 탐색 - 추세는 어떤 변수와 관련이 있을까?

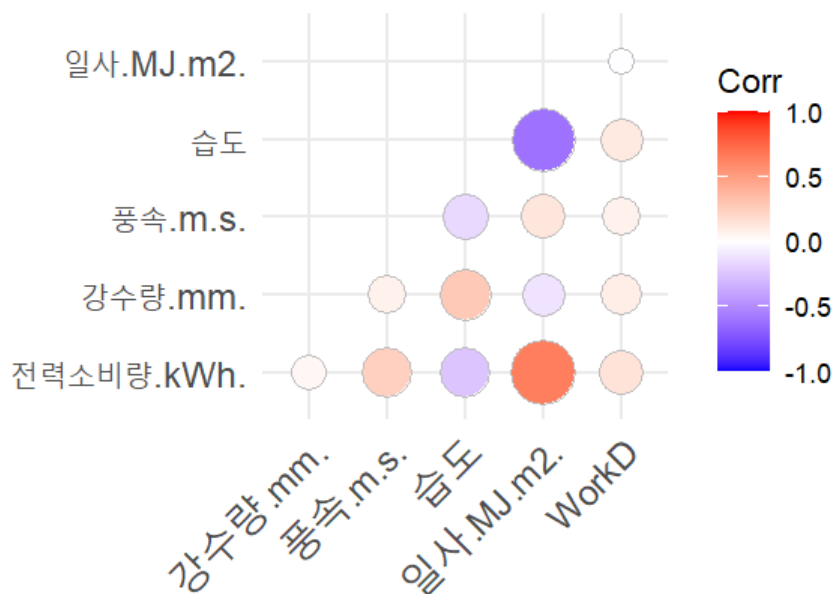


데이터 탐색 - 기상 변수와 전체 건물 전력 사용량의 상관분석

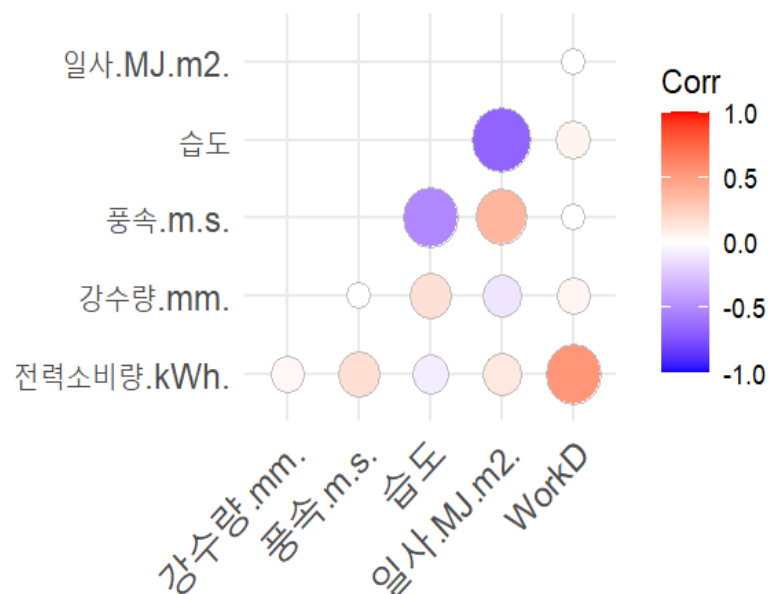


데이터 탐색 - 기상 변수와 건물별 전력 사용량의 상관분석

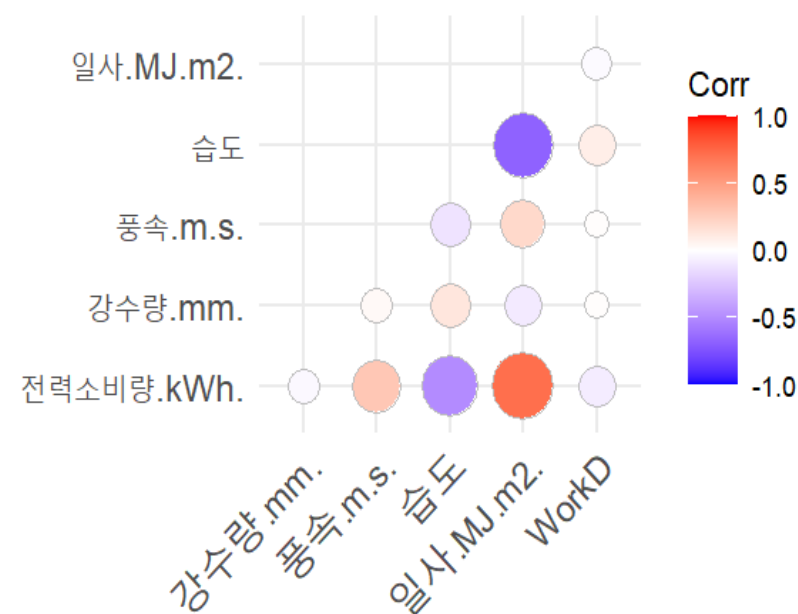
1번 건물 상관분석



30번 건물 상관분석



40번 건물 상관분석



건물별로 다 달라서 공통의 모델을 만드는 것 보다 각 건물별로 잘 맞는 모델을 생성하는 쪽으로 방향을 잡음

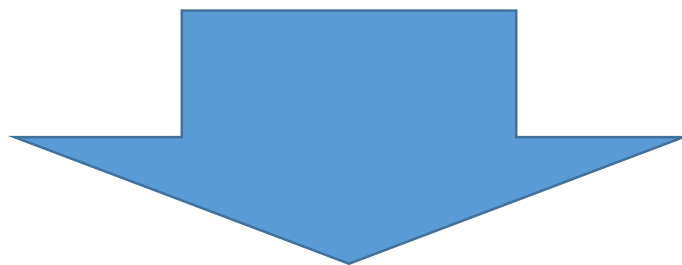
데이터 탐색 - 결론

각 건물별로 다른 특성일 보인다.
-> 건물별 모델 생성

전력사용량의 주기성 <- 평일/휴일, 시간(hour)

전력사용량의 추세 <- 온도

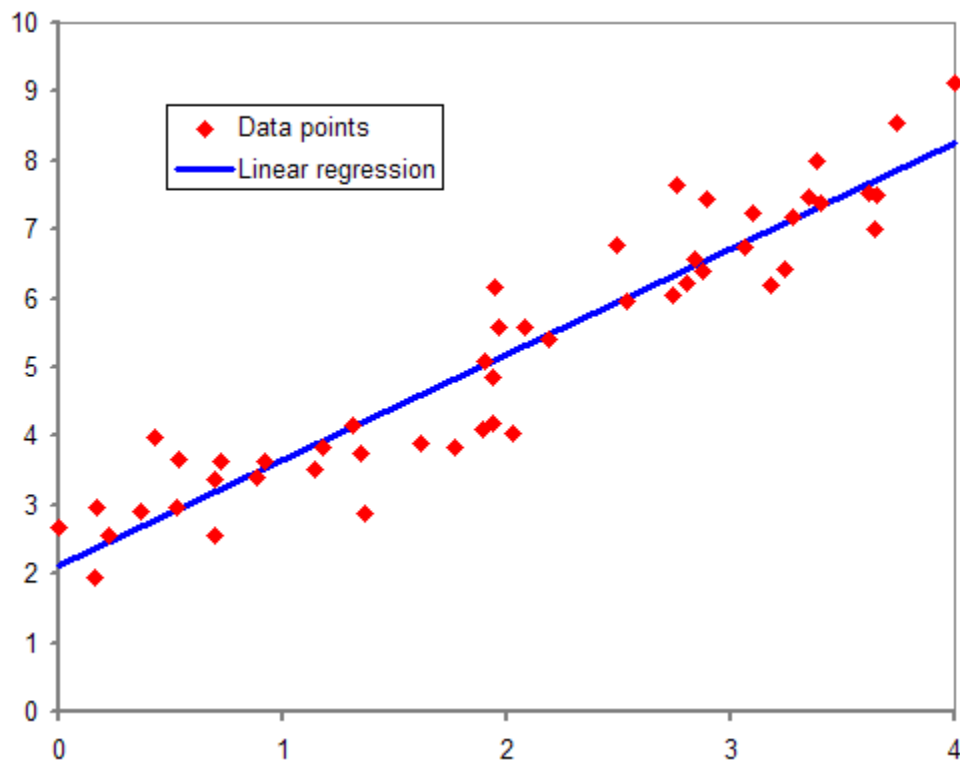
전력사용량의 레벨 <- 냉방면적, 건물타입



회귀 분석
머신 러닝

데이터 탐색 - 사용 모델

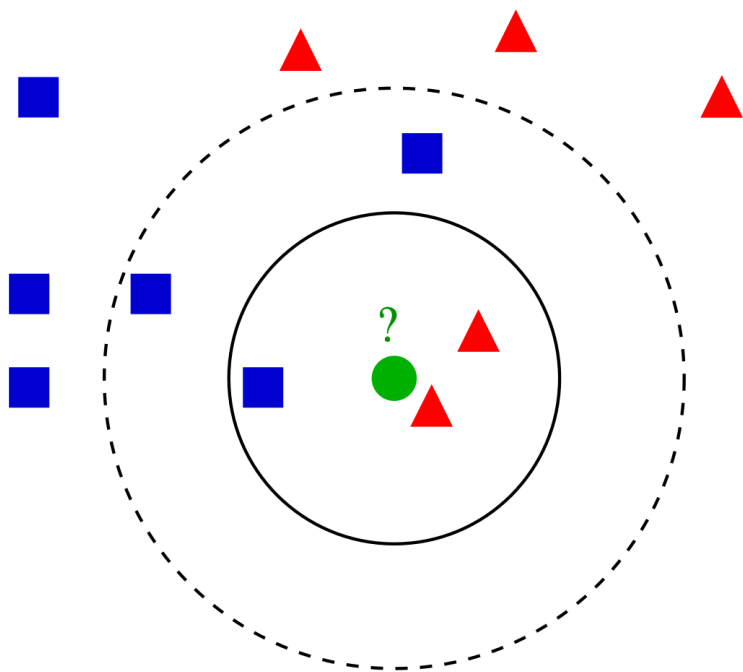
01. 회귀 분석



- 통계적인 분석
- 변화하는 데이터나 어떤 영향, 가설적 실험, 인과 관계의 모델링등의 통계적 예측에 이용됨
- 그러나 많은 경우 가정이 맞는지 아닌지 적절하게 밝혀지지 않은 채로 이용되어 그 결과가 오용되는 경우도 있다.
- 분석이 용이해져서 결과를 쉽게 얻을 수 있지만 분석 방법의 선택이 적절했는지 또한 정보 분석이 정확한지 판단하는 것은 연구자에 달려 있음

데이터 탐색 - 사용 모델

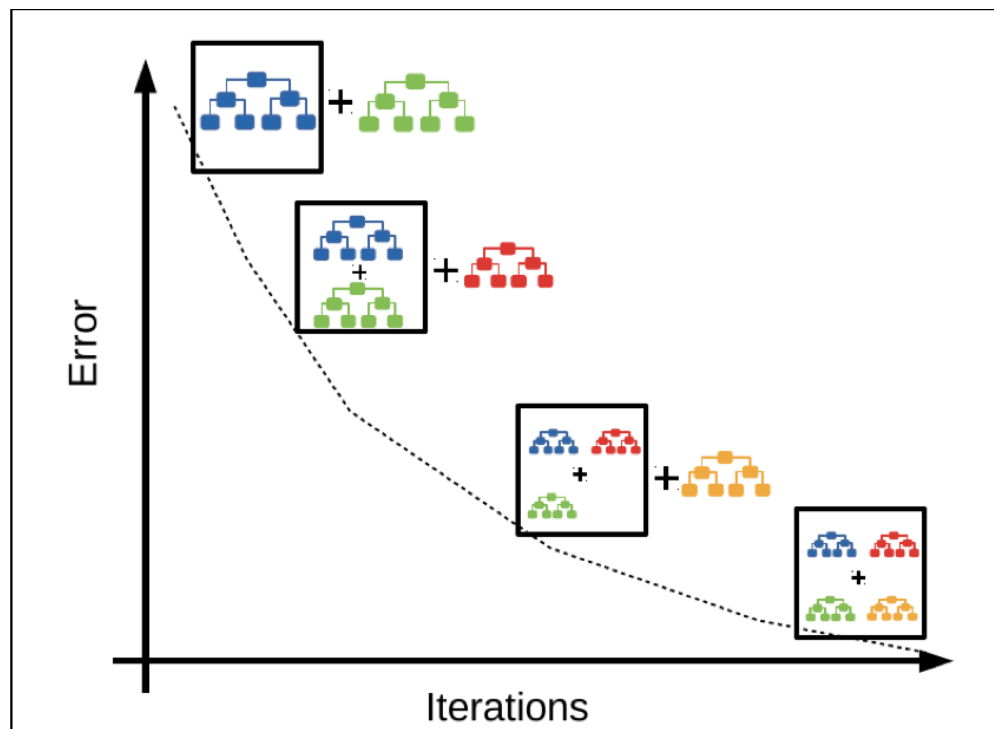
02. kNN



- 중심거리로 분류하는 기계학습 알고리즘
- k 를 지정하면 k 개의 데이터포인트가 지정됨
- 너무 작은 K 값은 노이즈에 민감하게 반응할 수 있으며, 너무 큰 K 값은 경계가 불명확해짐
- 간단하면서도 효과적인 분류 알고리즘으로 예측에도 사용할 수 있음

데이터 탐색 - 결론

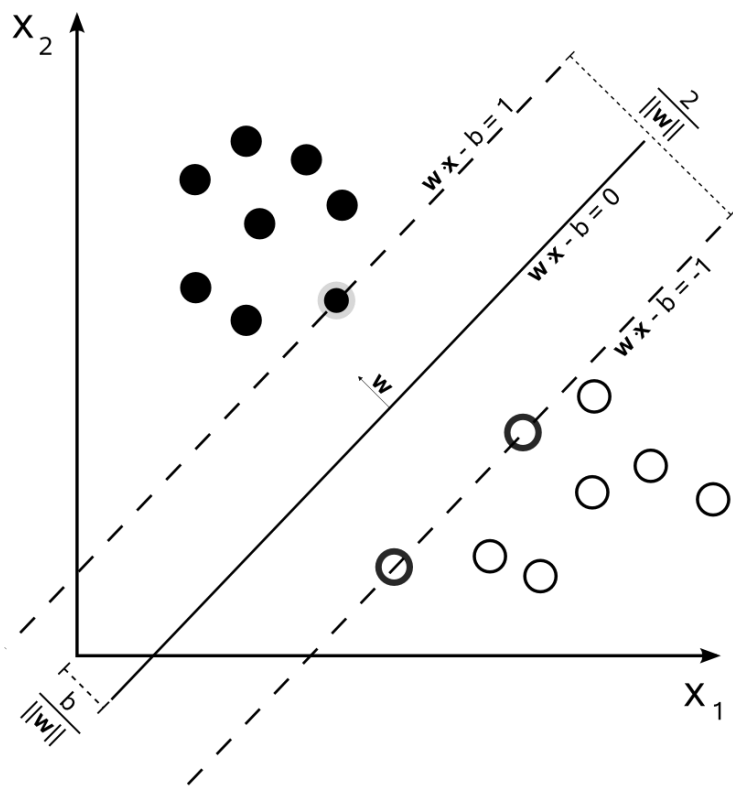
03. GBM



- 앙상블 기법 (부스팅)
- 약한 학습기를 순차적으로 학습시켜 이전 학습기의 오차를 보정
- 경사 하강법을 사용하여 오차를 최소화함
- 복잡도가 높은 데이터셋에서 높은 성능을 보임
- 하이퍼파라미터 튜닝이 중요함

데이터 탐색 - 결론

04. 서포트 벡터 머신



- 지도학습-분류
- 범주 간의 경계를 찾고, 데이터 개체로부터의 거리, 마진(margin)을 최대로 하는 경계를 찾는 것이 특징
- 비선형 분류시 데이터를 고차원으로 사상하는 작업 필요(효율적으로 하기 위해 커널 트릭 사용)
- 다양한 라이브러리로 사용 용이
- 분류, 회귀 예측 문제에 동시에 활용 가능
- 이진분류만 가능하며 데이터가 많을 시 모델 학습 시간 소요 큼

통계 분석 – 상관 관계

- 전력 소비량과 기상 변수 간의 상관관계(5번 건물)

```
b5_cor <- b5 %>%
```

```
  group_by(hour) %>%
```

```
  summarise(
```

```
    pc=mean(power_consumption),
```

```
    working = mean(holiday),
```

```
    temp=mean(temperature),
```

```
    rain=sum(rainfall),
```

```
    humi=mean(humidity))
```

```
b5_cor <- ifelse(b5$hour >= 6 & b$hour < 20, 1, 0)
```

- 상관분석

```
cor(b5$pc, b5$holiday) => 0.7911621
```

```
cor(b5$pc, b5$temp) => 0.8399843
```

```
cor(b5$pc, b5$rain) => 0.1608836
```

```
cor(b5$pc, b5$humi) => -0.8321016
```

통계 분석 – 회귀 분석

- 상관관계가 높은 변수만 선택하여 회귀분석

```
step(lm_b5, direction= ' both ' )  
lm_b5 <- lm(pc~hour + temp + holiday, data=b5)  
summary(lm_b5)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-7272.10	1296.97	-5.607	1.73e-05	***
hour	-36.67	12.85	-2.852	0.009846	**
temp	387.41	55.91	6.930	9.94e-07	***
holiday	758.81	172.88	4.389	0.000283	***

머신 러닝 – 라이브러리 생성 및 패키지 설치

- 프로젝트 라이브러리 관리

```
dir.create("C:/R_program/library")
```

```
.libPaths("C:/R_program/library")
```

- 필요 패키지 설치

```
install.packages("dplyr")
```

```
install.packages("kkn")
```

```
install.packages("neuralnet")
```

```
install.packages("Metrics") ★ calculate_errors(actual, forecast) 함수로 평가지표 계산
```

```
install.packages("e1071")
```

```
install.packages("gbm")
```

머신 러닝 – 스코어 저장소 생성

- 데이터 불러오기

```
train <- read.csv("https://raw.githubusercontent.com/Eungyum/Power_Consumption/main/train.csv")
test <- read.csv("https://raw.githubusercontent.com/Eungyum/Power_Consumption/main/test.csv")
train$X <- NULL
test$X <- NULL
```

- 스코어 저장소 생성

```
score_LR <- data.frame(BID = NA, SMAPE = vector("numeric", 100), RMSE = vector("numeric", 100), MAE = vector("numeric", 100), MSE = vector("numeric", 100))
score_SVM <- data.frame(BID = 1:100, SMAPE = vector("numeric", 100), RMSE = vector("numeric", 100), MAE = vector("numeric", 100), MSE = vector("numeric", 100))
score_kNN <- data.frame(BID = 1:100, SMAPE = vector("numeric", 100), RMSE = vector("numeric", 100), MAE = vector("numeric", 100), MSE = vector("numeric", 100))
score_NN <- data.frame(BID = 1:100, SMAPE = vector("numeric", 100), RMSE = vector("numeric", 100), MAE = vector("numeric", 100), MSE = vector("numeric", 100))
score_gbm <- data.frame(BID = 1:100, SMAPE = vector("numeric", 100), RMSE = vector("numeric", 100), MAE = vector("numeric", 100), MSE = vector("numeric", 100))
```


머신 러닝 – 오차 측정 함수 정의

- 오차 측정 함수 정의

```
calculate_errors <- function(actual, predicted) {  
  # SMAPE (Symmetric Mean Absolute Percentage Error)  
  smape <- 100 * mean(2 * abs(predicted - actual) / (abs(predicted) + abs(actual)), na.rm = TRUE)  
  # RMSE (Root Mean Squared Error)  
  rmse <- rmse(actual, predicted)  
  # MAE (Mean Absolute Error)  
  mae <- mae(actual, predicted)  
  # MSE (Mean Squared Error)  
  mse <- mse(actual, predicted)  
  return(data.frame(BID = i, SMAPE = smape, RMSE = rmse, MAE = mae, MSE = mse))  
}
```

머신 러닝 - 오차 측정 함수 정의

평가지표	수식	특징
SMAPE	$\frac{100}{n} \sum \frac{ F_t - A_t }{(A_t + F_t)/2}$	<ul style="list-style-type: none"> 0 근처의 실제값에서 큰 오차가 발생할 경우 보정이 됨
RMSE	$\sqrt{\frac{1}{n} \sum (F_t - A_t)^2}$	<ul style="list-style-type: none"> 큰 오차에 더 큰 가중치를 부여 : 큰 오차를 강조 단위가 원래 데이터와 동일하여 다른 데이터셋간의 비교가 어려움
MAE	$\frac{1}{n} \sum F_t - A_t $	<ul style="list-style-type: none"> 오차의 크기에 따라 동일하게 가중치를 부여 큰 오차와 작은 오차를 동일하게 취급하여 큰 오차의 영향을 덜 받음
MSE	$\frac{1}{n} \sum (F_t - A_t)^2$	<ul style="list-style-type: none"> 큰 오차에 더 큰 가중치를 부여 큰 오차의 영향을 많이 받음

F_t : 예측값, A_t : 실제값

머신 러닝 – LR(Linear Regression)

```
for (i in 1:100) {  
  # 훈련세트 : train_LR  
train_LR <- train %>%  
  filter(BID==i)  
  # 테스트 세트 : test_LR  
test_LR <- test %>%  
  filter(BID==i)  
  # 모델 생성  
model_LR <- lm(PC~Temp+Rain+WS+HM+WorkD+WC+DI, data=train_LR)  
  # 스텝와이즈로 최적의 모델 생성(최종 모델)  
model_LR <- step(model_LR, direction="both")  
  # 모델을 저장 : model_LR_i  
saveRDS(model_LR, file=paste0("model_LR_", i))  
  # 실제값 저장 : actual  
actual <- test_LR$PC  
  # 예측값 생성  
forecast <- predict(model_LR, test_LR)  
  
  ## 스코어s ##  
score_LR[i,]<-calculate_errors(actual, forecast)  
} # for문 끝
```

머신 러닝 – SVM(Support Vector Machine)

```
for (i in 1:100) {  
  train_SVM <- train %>%  
    filter(BID==i) %>%  
    select(PC, Temp, Rain, WS, HM, Time, WorkD)  
  test_SVM <- test %>%  
    filter(BID==i) %>%  
    select(PC, Temp, Rain, WS, HM, Time, WorkD)  
  # 모델 생성  
  model_SVM <- svm(PC~., data=train_SVM, type = "eps-regression", kernel = "linear")  
  # 모델을 저장 : model_SVM_i  
  saveRDS(model_SVM, file=paste0("model_SVM_", i))  
  # 실제값 저장 : actual  
  actual <- test_SVM$PC  
  # 예측값 생성  
  forecast <- predict(model_SVM, test_SVM)  
  
  ## 스코어s ##  
  score_SVM[i,]<-calculate_errors(actual, forecast)  
} # for문 끝
```

머신 러닝 – GBM(Gradient Boosting Machine)

```
for ( i in 1:100 ){
  train_gbm <- train %>%
    filter(BID==i)
  test_gbm <- test %>%
    filter(BID==i)
  train_gbm$BID <- NULL
  train_gbm$DateTime <- NULL
  train_gbm$WeekD <- NULL
  train_gbm$DILv <- NULL
  test_gbm$BID <- NULL
  test_gbm$DateTime <- NULL
  test_gbm$WeekD <- NULL
  test_gbm$DILv <- NULL
  # 모델 생성
  model_gbm <- gbm(PC ~ ., data=train_gbm, distribution="gaussian", n.trees = 100, interaction.depth = 5,
    shrinkage = 0.05, n.minobsinnode=23)
  # 모델을 저장 : model_kNN_i
  saveRDS(model_gbm, file=paste0("model_gbm_", i))
  # 실제값 저장 : actual
  actual <- test_gbm$PC
  # 예측값 생성 : forecast
  forecast <- predict(model_gbm, test_gbm)

  ## 스코어s ##
  score_gbm[i]<-calculate_errors(actual, forecast)
} # for문 끝
```

머신 러닝 – kNN(k-Nearest Neighbor)

```
for(i in 1:100) {  
  train_kNN <- train %>%  
    filter(BID==i) %>%  
    select(PC, Temp, Rain, WS, HM, Time, WorkD)  
  test_kNN <- test %>%  
    filter(BID==i) %>%  
    select(PC, Temp, Rain, WS, HM, Time, WorkD)  
  # 모델 생성  
  model_kNN <- train.kknn(PC ~ ., train_kNN, kmax = 2, kernel = "optimal")  
  # 모델을 저장 : model_kNN_i  
  saveRDS(model_kNN, file=paste0("model_kNN_", i))  
  # 실제값 저장 : actual  
  actual <- test_kNN$PC  
  # 예측값 생성 : forecast  
  forecast <- predict(model_kNN, test_kNN)  
  
  ## 스코어s ##  
  score_kNN[i,]<-calculate_errors(actual, forecast)  
} # for문 끝
```

평가 - 모델 평가

★ SMAPE의 평균값이 낮을수록 성능이 좋은 모델 ★

1 GBM - 7.329276

2 KNN - 11.964883

3 SVM - 19.674414

4 LR - 22.036654

