

데이터 마이닝 / 정보 디자인

날씨 데이터를 이용한 폭염 예측



2018204085 박지영 (대표)

2017204077 유 준

2019204032 송인섭

2019204037 오수빈

2019204061 이규민

CONTENTS

1. 주제

2. 데이터 전처리

3. 데이터 분석

4. 대시보드

CONTENTS

1. 주제

- 분석 목표

2. 데이터 전처리

3. 데이터 분석

4. 대시보드

한반도 이상기후 변화 및 분석

폭우

가뭄

서울의 날씨 데이터를 이용한
폭염 예측

폭설

이상저온

우박

폭염의 기준 : 최고 기온이 33도 이상

폭염 주의보

최고 기온이 33도 이상인 상태가
2일 이상

폭염 경보

최고 기온이 35도 이상인 상태가
속될 예정

최고 기온에 영향을 끼치는 변수를 찾아
가까운 미래의 폭염 예측

CONTENTS






1. 주제

2. 데이터 전처리

- 데이터셋
- 결측치 대체

3. 데이터 분석

4. 대시보드

	OBS_ASOS_DD_1907_1909.csv
	OBS_ASOS_DD_1910_1919.csv
	OBS_ASOS_DD_1920_1929.csv
	OBS_ASOS_DD_1930_1939.csv
	OBS_ASOS_DD_1940_1949.csv
	OBS_ASOS_DD_1950_1959.csv
	OBS_ASOS_DD_1960_1969.csv
	OBS_ASOS_DD_1970_1979.csv
	OBS_ASOS_DD_1980_1984.csv
	OBS_ASOS_DD_1985_1989.csv
	OBS_ASOS_DD_1990_1994.csv
	OBS_ASOS_DD_1995_1999.csv
	OBS_ASOS_DD_2000_2004.csv
	OBS_ASOS_DD_2005_2009.csv
	OBS_ASOS_DD_2010_2014.csv
	OBS_ASOS_DD_2015_2021.csv
	OBS_ASOS_DD_2021_6.csv

- 1907년 ~ 2021년까지의 **종관기상관측데이터**
-> 파일을 불러와 하나의 데이터프레임으로 **합치**는 작업을 실행

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
지점	지점명	일시	평균기온(°C)	최저기온(°C)	최저기온 시	최고기온(°C)	최고기온 시	강수 계속시	10분 최대 강수량(mm)	10분 최대 강수량 시	1시간 최대 강수량(mm)	1시간 최대 강수량 시	일강수량(mm)	최대 순간 풍속(mph)	최대 순간 풍속 시	최대 순간 풍속(mph)	최대 순간 풍속 시	최대 순간 풍속(mph)	최대 순간 풍속 시	평균 풍속(mph)	풍정합(100)	최대 풍향(10°)

- 변수의 이름을 재지정

일시	평균기온(°C)	최저기온(°C)	최저기온 시	최고기온(°C)	최고기온 시	강수 계속시	10분 최대 강	10분 최대 강	1시간 최대 강	1시간 최대 강	일강수량(mm)	최대 순간 강	최대 순간 강	최대 순간 강
1907-10-01	13.5	7.9		20.7									0	
1907-10-02	16.2	7.9		22							0.2		0	
1907-10-03	16.2	13.1		21.3							2.4		0	
1907-10-04	16.5	11.2		22									0	
1907-10-05	17.6	10.9		25.4									0	

















- 과거의 기술의 문제로 **관측되지 않은 변수들에 결측치가 존재**
-> 변수의 결측치가 **10000개 이상**인 변수는 삭제
- 1950년~1953년도**엔 많은 변수들에 결측치가 존재

```
> sapply(dat, function(x) sum(is.na(x)))
```

date	tempAvg	tempLow	tempHigh	windMaxInstantDir	windMax	windMaxDir
0	346	347	348	378	467	407
windAvg	airDXSum	RHMin	RHAvg	VPAvg	LocalAPAvg	seaAPAvg
410	3121	5971	346	346	7047	9540
sunlightTimeSum	warCloudAvg	groundTempAvg	grassTempMin	temp5Avg	temp10Avg	temp20Avg
3252	346	6161	9885	9426	7606	7615
temp30Avg	temp_5	temp1	temp1_5	evapnSmallSum		
7632	7693	7729	7731	2070		

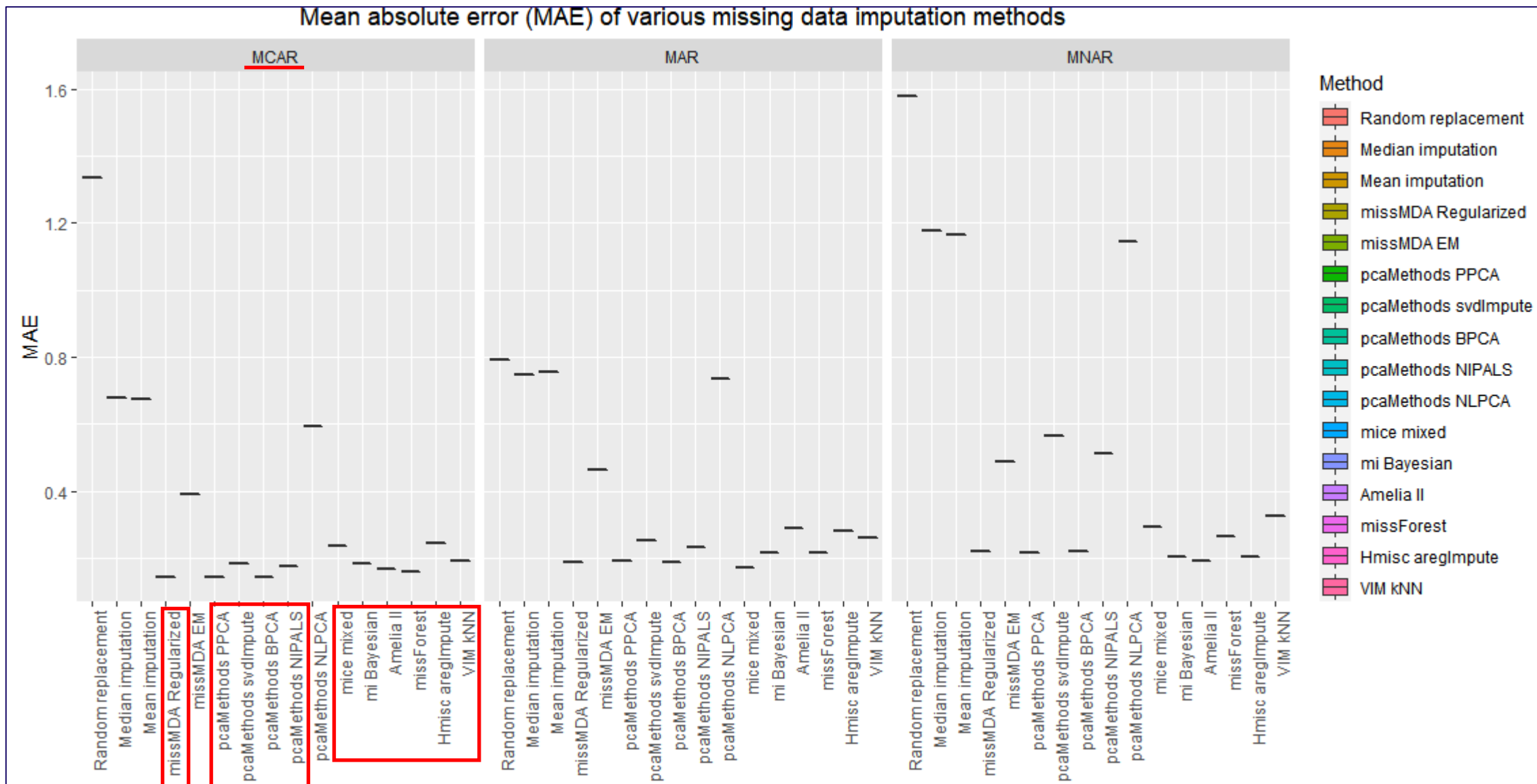
- 해당 데이터셋에 가장 적합한
결측치 대체 방법을 탐색

<결측치 처리 방법>

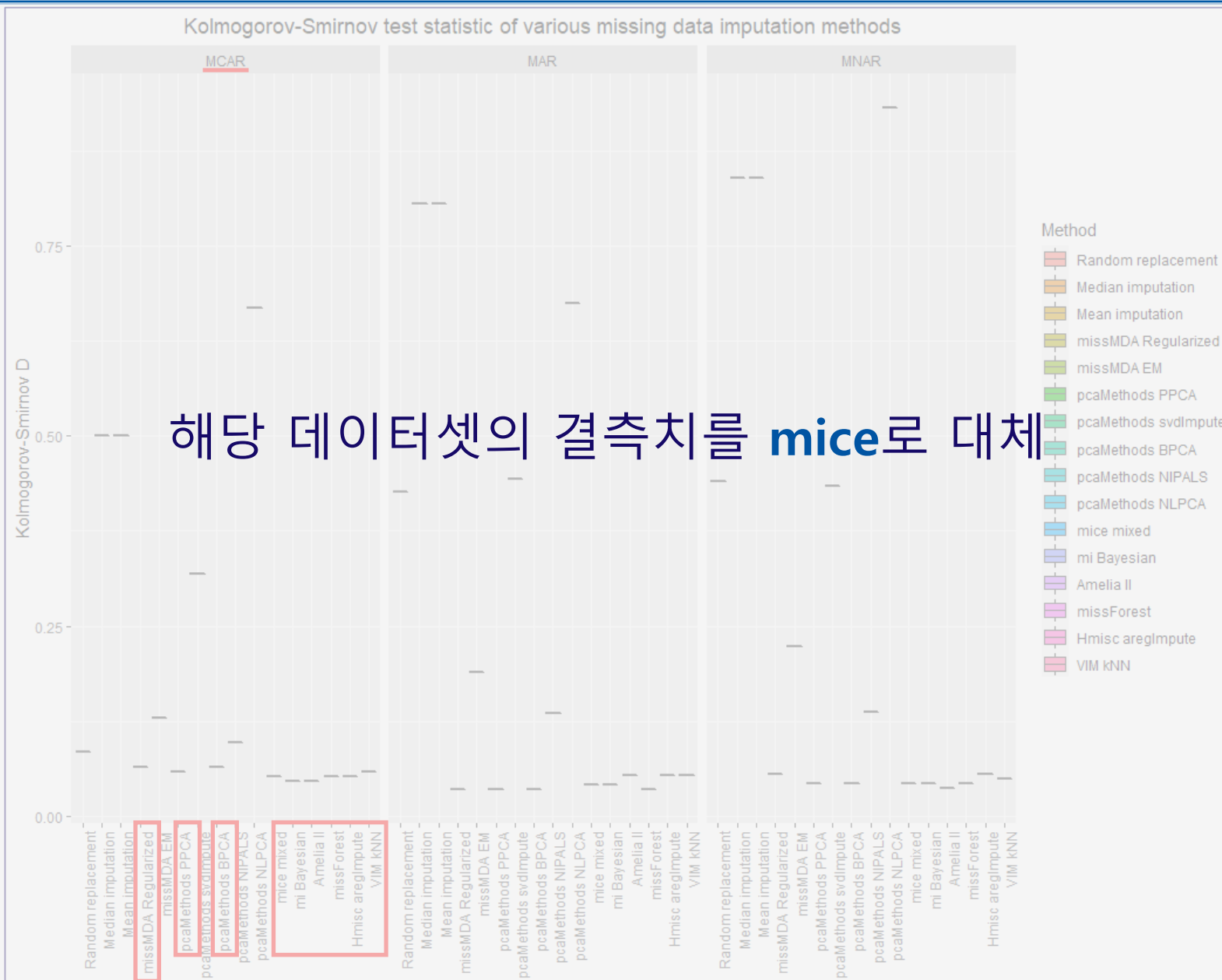
Method	
	Random replacement
	Median imputation
	Mean imputation
	missMDA Regularized
	missMDA EM
	pcaMethods PPCA
	pcaMethods svdImpute
	pcaMethods BPCA
	pcaMethods NIPALS
	pcaMethods NLPCA
	mice mixed
	mi Bayesian
	Amelia II
	missForest
	Hmisc aregImpute
	VIM kNN

- 데이터셋에 **결측치**가 존재하는 행을 모두 **삭제**
- 결측치가 없는 데이터셋에서 **임의로 결측치를 생성**
- 16가지 방법으로 **결측치를 대체**
- 결측치를 대체한 값과 실제값을 비교해 각 결측치 대체 방법의 **결과 비교**

02.데이터 전처리 - 결측치 처리



02.데이터 전처리 - 결측치 처리



CONTENTS

1. 주제

2. 데이터 전처리

3. 데이터 분석

- 선형회귀모델
- 시계열모델

4. 대시보드

선형회귀모델

로지스틱회귀모델

시계열모델

사결정모델

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	418	45
1	6	22

선형회귀모델 → 단기 예측

시계열모델 → 장기 예측

1) 목표

```
'data.frame': 43414 obs. of 27 variables:
 $ date      : chr  "1907-10-01" "1907-10-02" "1907-10-03" "1907-10-04" ...
 $ tempAvg   : num  0.038 0.185 0.185 0.201 0.261 ...
 $ tempLow   : num  0 0 0.283 0.179 0.163 ...
 $ tempHigh  : num  0.127 0.196 0.159 0.196 0.376 ...
 $ windMaxInstantDir: num  -0.259 -0.259 -0.259 -0.259 -0.259 ...
 $ windMax   : num  -0.519 -1.074 0.407 -1.259 -1 ...
 $ windMaxDir : num  0 0 0 -2 -2 ...
 $ windAvg   : num  0.143 -0.857 0 -0.714 -0.429 ...
 $ airDXSum  : num  0.187 -0.829 0.042 -0.683 -0.393 ...
 $ RHMin     : num  -0.455 0.5 0.864 0.636 0.136 ...
 $ RHAvg     : num  -0.0305 0.3249 0.6954 0.7107 0.1878 ...
 $ VPAvg     : num  0.024 0.312 0.416 0.44 0.336 ...
 $ LocalAPAv : num  -0.15 0.599 0.626 0.136 0.578 ...
 $ seaAPAv   : num  -0.512 0.25 -0.242 -0.274 0.266 ...
 $ sunlightTimeSum : num  0.424 -0.273 0.455 -0.455 0.455 ...
 $ warCloudAvg : num  -0.69 0.345 0.069 0.241 -0.862 ...
 $ groundTempAvg : num  0.163 0.313 0.524 0.197 0.514 ...
 $ grassTempMin : num  0.0825 0.0206 0.1134 0.3299 0.0979 ...
 $ temp5Avg   : num  -0.0513 0.1218 0.109 0.0641 -0.0577 ...
 $ temp10Avg  : num  -0.47 -0.47 -0.47 -0.47 -0.47 ...
 $ temp20Avg  : num  -0.481 -0.481 -0.481 -0.481 -0.481 ...
 $ temp30Avg  : num  -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 ...
 $ temp_5     : num  0.103 0.23 0.309 0.121 0.152 ...
 $ temp1      : num  0.23704 0.38519 0.47407 -0.04444 0.00741 ...
 $ temp1_5    : num  0.35 0.505 0.534 -0.456 -0.32 ...
 $ evapnSmallSum : num  0.4 0 0.9 -0.167 0.5 ...
 $ to_tempHigh : num  0.196 0.159 0.196 0.376 0.159 ...
```

- 종속변수 : 1,2,3,4,5,6,7일 후의 최고기온
- 독립변수 : 평균기온, 최저기온, 최고기온, 최대순간풍속 풍향, 최대풍속, 최대풍속 풍향, 평균풍속, 풍정합, 최소상대습도, 평균상대습도, 평균증기압, 평균현지기압, 최고해면기압, 합계일조시각, 평균전운량, 평균지면온도, 최저초상온도, 평균 5,10,20,30cm지중온도, 0.5,1.0,1.5m 지중온도, 합계소형증발량

2) 정규화/표준화

Scaling으로 해결

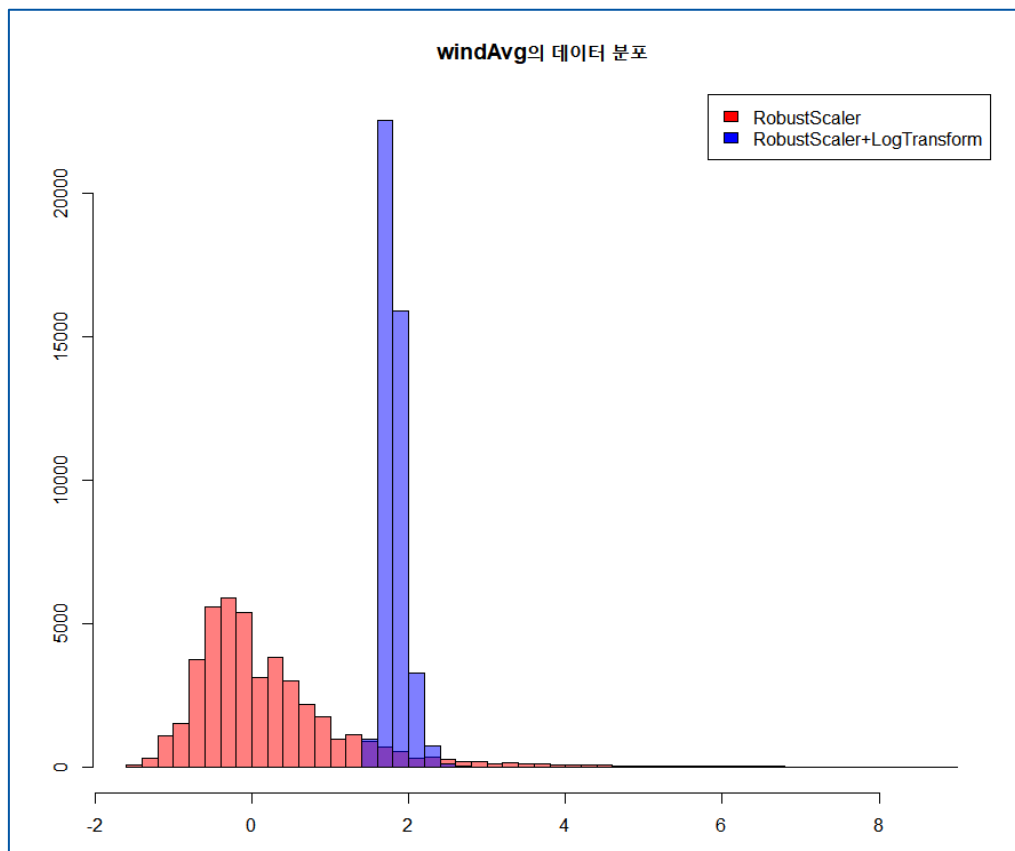
RHMin	RHAvg	VPAvg	Loc					windMaxD	windAvg	airDXSum
34	67.7	9.6	1008.6	1009.7	8	1.3	3.7	270	2.5	2160
55	74.7	13.2	1019.6	1019.3	2	7.3	2.2	270	1.1	950
63	82	14.5	1020	1013.1	0	5.7	6.2	270	2.3	1987
58	82.2	14.8	1012.8	1012.7	4	6.7	1.7	270	1.2	1123
47										1469
63										2419

Robust Scaler 사용

$$Y = \frac{(X - X_{median})}{(X_{IRQ,75\%} - X_{IRQ,25\%})}$$

- 평균과 분산 대신
중간값과 사분위값을 사용하는 Scaler
- 아웃라이어의 영향을 최소화
- 아주 동떨어진 데이터 제거

2) 정규화/표준화



- 데이터의 분포가 **한쪽으로 치우침**
(ex. windAvg)

Log Transform

- 각 변수에 **로그**를 취함
-> **0이나 음수**는 일정한 상수를 더한 후 로그를 취함
- 이질적 분산**들을 바로잡음

3) Scaling

Train Data

1907년 ~ 2016년

Test Data

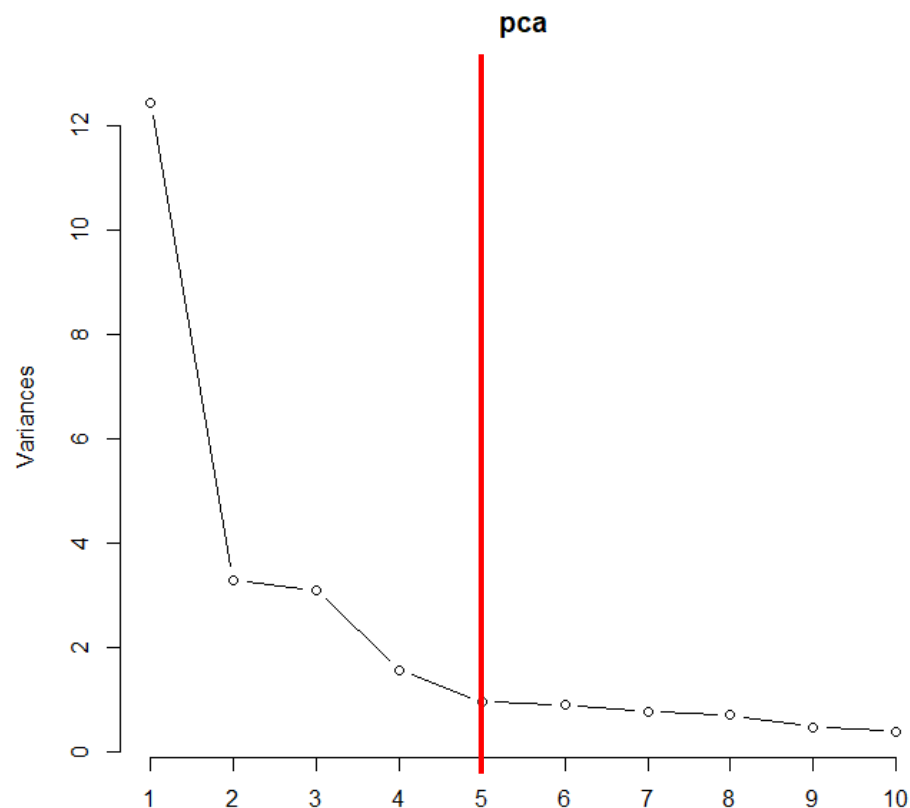
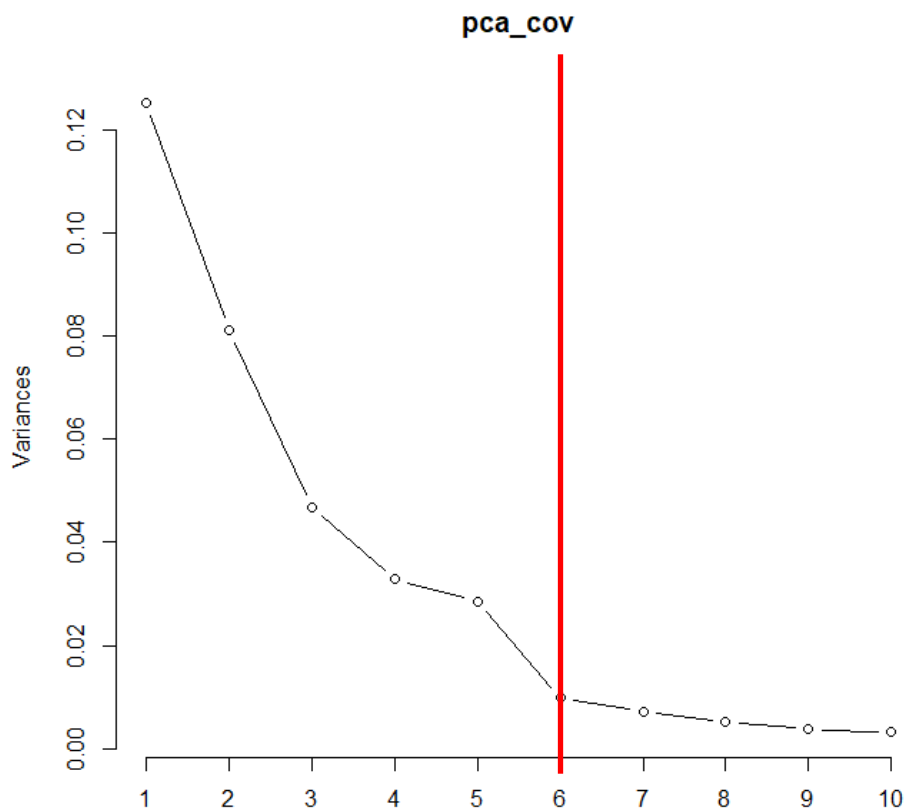
2017년 ~ 2021년 5월

4) 차원축소

공분산 행렬

VS

상관계수 행렬



Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	0.3538	0.2848	0.2165	0.18123	0.16889	0.09951
Proportion of Variance	0.3516	0.2278	0.1316	0.09224	0.08011	0.02781
Cumulative Proportion	0.3516	0.5794	0.7110	0.80328	0.88339	0.91120

4) 차원축소

공분산 행렬

VS

상관계수 행렬

- 설문조사와 같은 scale 점수화가 된 경우에는 **공분산 행렬**을 사용
- 변수의 scale이 많이 다른 경우, 특정 변수가 전체적인 경향을 좌우하기 때문에 **상관계수 행렬**을 사용

5) 모델링

전진선택법

VS

후진선택법

```
> summary(model_fwd3)

Call:
lm(formula = to_tempHigh ~ tempHigh + sunlightTimesum + windMax +
    temp5Avg + RHmin + windMaxInstantDir + LocalAPAvg, data = train)

Residuals:
    Min       1Q   Median       3Q      Max
-0.276581 -0.016351  0.002635  0.018416  0.237708

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.0439430   0.0081426    5.397 6.83e-08 ***
tempHigh       0.8085493   0.0040781  198.264 < 2e-16 ***
sunlightTimesum 0.0311895   0.0019166   16.273 < 2e-16 ***
windMax       -0.0316257   0.0012224   -25.873 < 2e-16 ***
temp5Avg       0.1396798   0.0042110   33.170 < 2e-16 ***
RHmin         0.0371713   0.0017282   21.509 < 2e-16 ***
windMaxInstantDir -0.0088153   0.0021597    -4.082 4.48e-05 ***
LocalAPAvg    -0.0018027   0.0006738    -2.675 0.00747 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0298 on 41794 degrees of freedom
Multiple R-squared:  0.9121,    Adjusted R-squared:  0.9121
F-statistic: 6.195e+04 on 7 and 41794 DF, p-value: < 2.2e-16
```

```
> summary(model_bwd3)

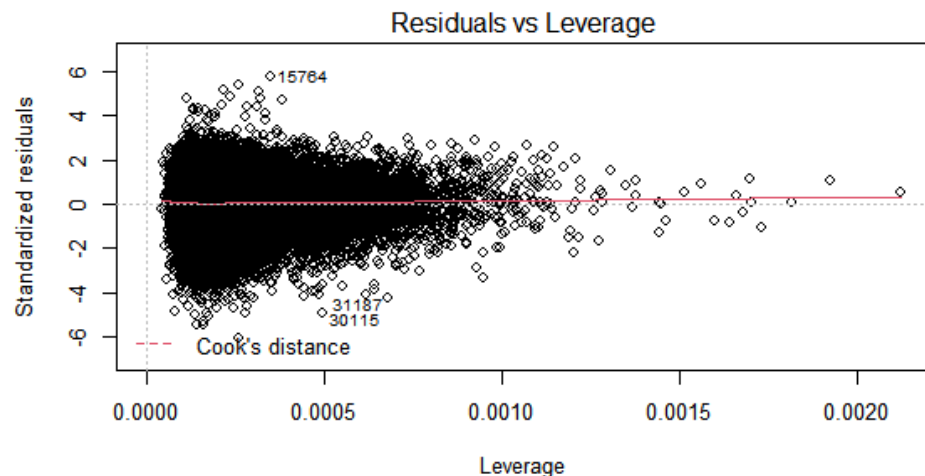
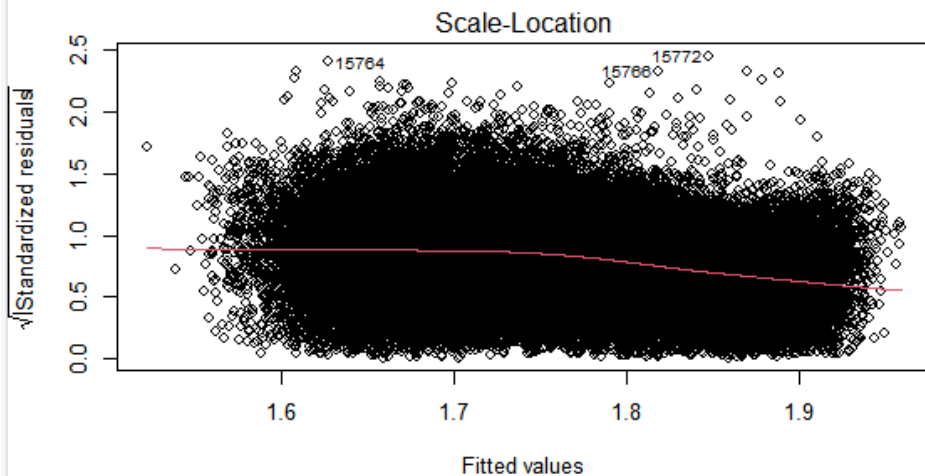
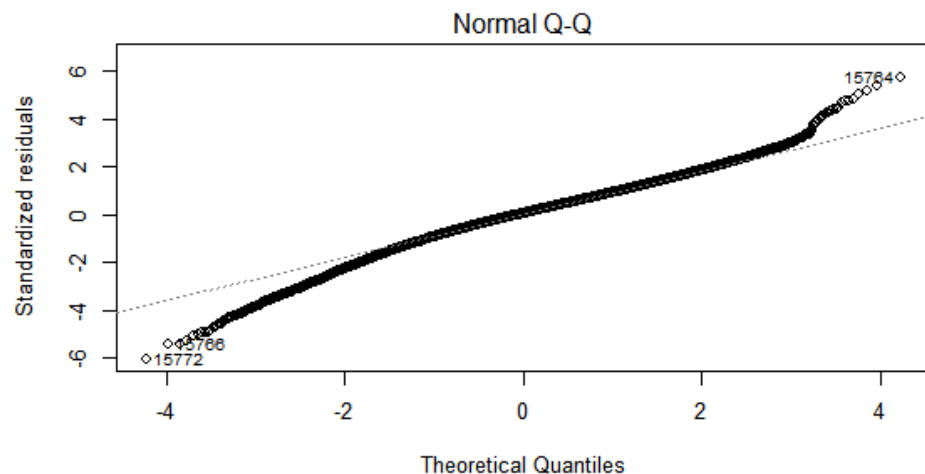
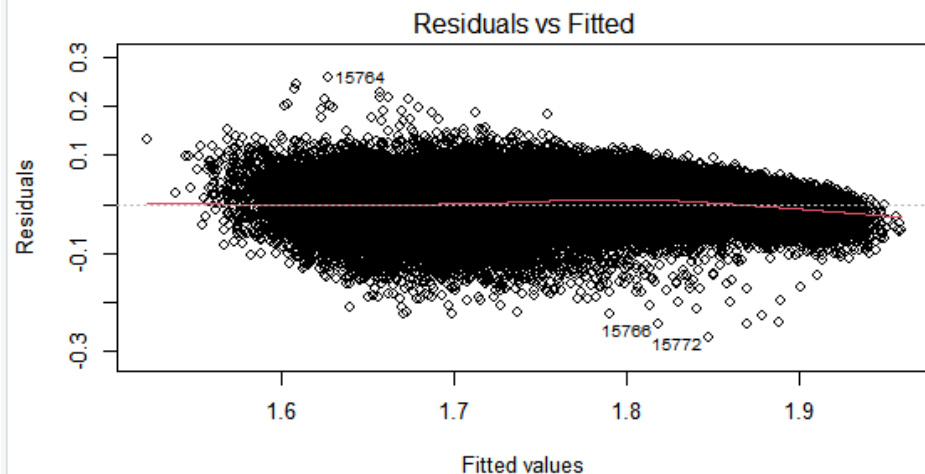
Call:
lm(formula = to_tempHigh ~ grassTempMin + windMax + windAvg +
    RHmin + sunlightTimesum + windMaxInstantDir + temp5Avg, data = train)

Residuals:
    Min       1Q   Median       3Q      Max
-0.274369 -0.021779  0.001429  0.022772  0.283494

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.125637   0.008782   14.306 < 2e-16 ***
grassTempMin   0.572036   0.004995  114.530 < 2e-16 ***
windMax        0.010945   0.002364    4.631 3.65e-06 ***
windAvg       -0.075759   0.002165   -34.989 < 2e-16 ***
RHmin         -0.017529   0.002127    -8.241 < 2e-16 ***
sunlightTimesum 0.136021   0.002297   59.216 < 2e-16 ***
windMaxInstantDir -0.079961   0.002471   -32.361 < 2e-16 ***
temp5Avg       0.379916   0.004978   76.326 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03595 on 41794 degrees of freedom
Multiple R-squared:  0.8721,    Adjusted R-squared:  0.872
F-statistic: 4.07e+04 on 7 and 41794 DF, p-value: < 2.2e-16
```

5) 모델링



6) 모델 시각화 - 최고기온에 영향을 미치는 변수 시각화

```
lm(formula = to_tempHigh ~ tempHigh + sunlightTimeSum + windMax +
temp5Avg + RHMin + windMaxInstantDir + LocalAPAvg, data = train)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.276581	-0.016351	0.002635	0.018416	0.237708

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.0439430	0.0081426	5.397	6.83e-08 ***
tempHigh	0.8085493	0.0040781	198.264	< 2e-16 ***
sunlightTimeSum	0.0311895	0.0019166	16.273	< 2e-16 ***
windMax	-0.0316257	0.0012224	-25.873	< 2e-16 ***
temp5Avg	0.1396798	0.0042110	33.170	< 2e-16 ***
RHMin	0.0371713	0.0017282	21.509	< 2e-16 ***
windMaxInstantDir	-0.0088153	0.0021597	-4.082	4.48e-05 ***
LocalAPAvg	-0.0018027	0.0006738	-2.675	0.00747 **

1일 후의 최고기온 예측 모델

```
lm(formula = to_tempHigh2 ~ sunlightTimeSum + temp5Avg + windMaxInstantDir +
VPAvg + tempHigh + RHAvg + seaAPAvg + LocalAPAvg, data = train2)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.273655	-0.021727	0.003508	0.024948	0.241821

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.141409	0.011921	11.862	< 2e-16 ***
sunlightTimeSum	0.032532	0.002340	13.904	< 2e-16 ***
temp5Avg	0.208589	0.006032	34.578	< 2e-16 ***
windMaxInstantDir	-0.023074	0.002811	-8.210	2.28e-16 ***
VPAvg	0.141345	0.005808	24.338	< 2e-16 ***
tempHigh	0.593580	0.005772	102.835	< 2e-16 ***
RHAvg	-0.019279	0.002502	-7.706	1.33e-14 ***
seaAPAvg	-0.024523	0.003089	-7.939	2.08e-15 ***
LocalAPAvg	0.006833	0.000984	6.944	3.86e-12 ***

2일 후의 최고기온 예측 모델

```
lm(formula = to_tempHigh3 ~ temp5Avg + sunlightTimeSum + VPAvg +
windMaxInstantDir + windMaxDir + tempHigh + RHAvg + seaAPAvg +
LocalAPAvg + windAvg, data = train3)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.27871	-0.02371	0.00323	0.02710	0.26400

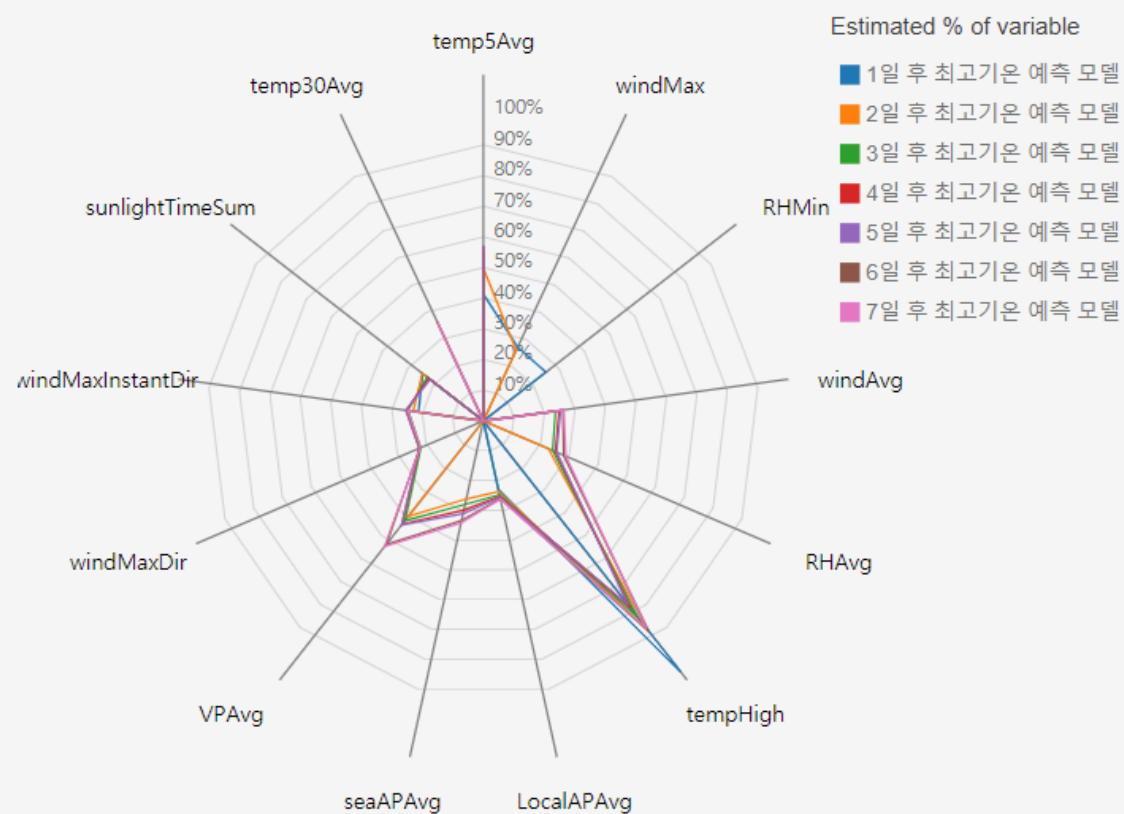
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.155643	0.015057	10.337	< 2e-16 ***
temp5Avg	0.249472	0.006514	38.300	< 2e-16 ***
sunlightTimeSum	0.016215	0.002668	6.077	1.24e-09 ***
VPAvg	0.156639	0.006263	25.008	< 2e-16 ***
windMaxInstantDir	-0.037247	0.003328	-11.192	< 2e-16 ***
windMaxDir	0.013486	0.001171	11.515	< 2e-16 ***
tempHigh	0.536754	0.006353	84.482	< 2e-16 ***
RHAvg	-0.031682	0.002731	-11.599	< 2e-16 ***
seaAPAvg	-0.037927	0.003457	-10.971	< 2e-16 ***
LocalAPAvg	0.014625	0.001099	13.312	< 2e-16 ***
windAvg	0.027557	0.001699	16.220	< 2e-16 ***

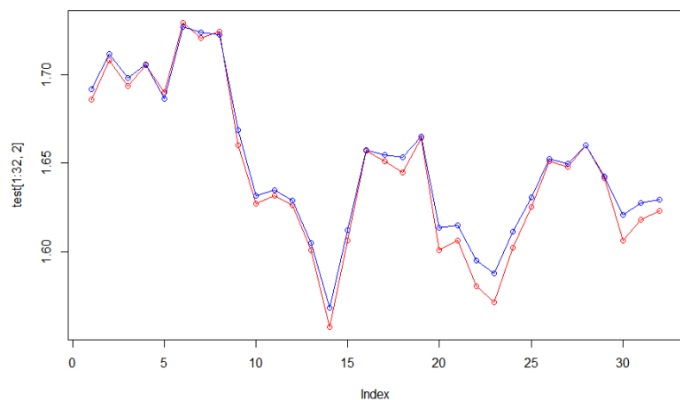
3일 후의 최고기온 예측 모델

6) 모델 시각화 - 최고기온에 영향을 미치는 변수 시각화

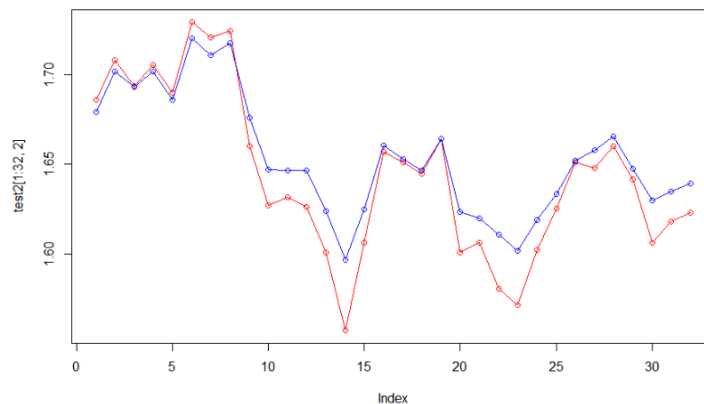
	var_name	coe1_est	coe2_est	coe3_est	coe4_est	coe5_est	coe6_est	coe7_est
1	temp5Avg	0.139679832	0.208588786	0.24947214	0.26732703	0.271446051	NA	NA
2	temp30Avg	NA	NA	NA	NA	NA	0.10073228	0.09743894
3	sunlightTimeSum	0.031189502	0.032532146	0.01621531	0.01217436	0.007513765	NA	NA
4	windMaxInstantDir	-0.008815250	-0.023073948	-0.03724704	-0.04126128	-0.042799757	-0.03791524	-0.03762528
5	windMaxDir	NA	NA	0.01348572	0.01647404	0.017247152	0.01648241	0.01423494
6	VPAvg	NA	0.141345180	0.15663862	0.16849089	0.170339621	0.24141774	0.24963592
7	seaAPAvg	NA	-0.024523033	-0.03792666	-0.05210285	-0.060987949	-0.07839887	-0.08405892
8	LocalAPAvg	-0.001802721	0.006833214	0.01462540	0.01863211	0.020707219	0.02386941	0.02474613
9	tempHigh	0.808549304	0.593579980	0.53675359	0.49909833	0.487152746	0.60424435	0.59303468
10	RHAvg	NA	-0.019278778	-0.03168169	-0.04003729	-0.043635076	-0.06450248	-0.07011942
11	windAvg	NA	NA	0.02755712	0.03740675	0.041817222	0.04859343	0.05044554
12	RHMin	0.037171291	NA	NA	NA	NA	NA	NA
13	windMax	-0.031625683	NA	NA	NA	NA	NA	NA



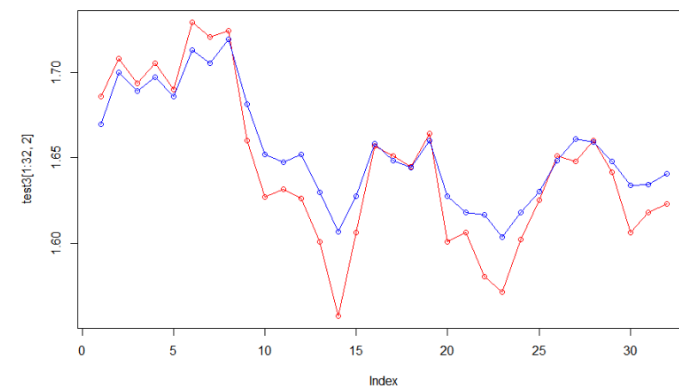
6) 모델 시각화 - 실제값과 예측값 비교



1일 후의 최고기온 예측 모델

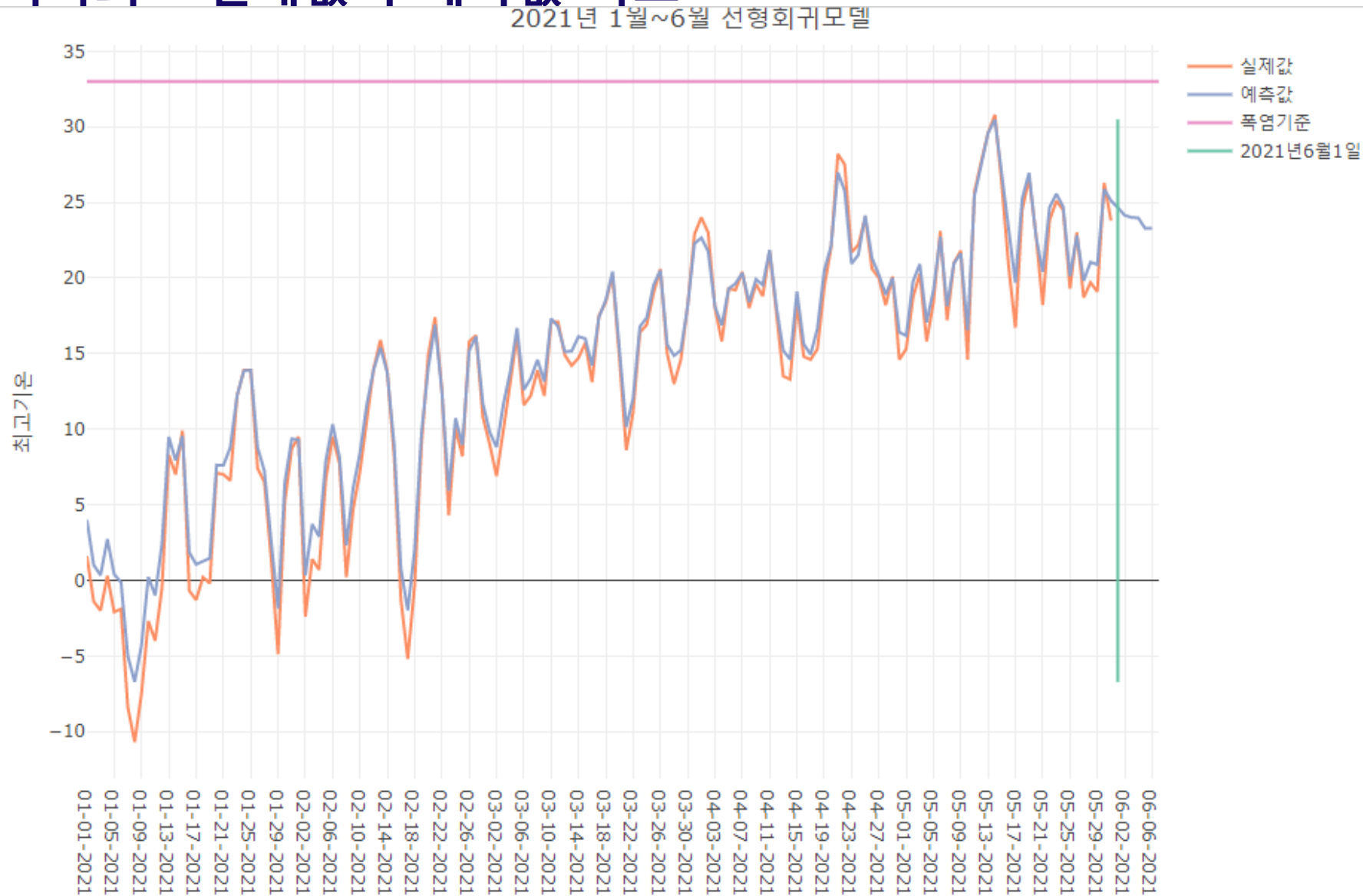


2일 후의 최고기온 예측 모델



3일 후의 최고기온 예측 모델

6) 모델 시각화 - 실제값과 예측값 비교



1) 목표

```
In [ ]: dataset=data

#6.25 기간 제거
data1=dataset[:15434]
data2=dataset[16054:]

dataset=pd.concat([data1,data2],ignore_index=True)
```

- 모델 학습에 악영향을 미쳐 1년 단위로
총 1950~1953년 간의 데이터 일괄 제거

1) 목표

```
In [ ]: features_considered = ['tempHigh', 'tempAvg', 'VPAvg', 'RHAvg', 'groundTempAvg', 'temp_5', 'temp1_5', 'windMax', 'date']
```

- 입력 신호로는 여러가지 변수를 테스트하여 제일 학습 결과가 좋았던 변수들 사용
- 모델이 대략적인 계절 파악에 도움을 주기 위해 1년중 몇일인지를 나타내는 변수로 Date 변수 추가

```
In [ ]: target_names = ['tempHigh', 'tempAvg', 'VPAvg', 'RHAvg', 'groundTempAvg', 'temp_5', 'temp1_5', 'windMax']
```

- 예측한 값을 바탕으로 예측을 하기 위해
출력 신호는 Date 변수를 제외한 나머지 변수들로 예측

2) 정규화/표준화

```
In [ ]: def outliar(data): #이상치 제거 파악
        q1,q3=np.percentile(data,[25,75])
        iqr =q3-q1
        lower=q1-(iqr*1.5)
        upper=q3+(iqr*1.5)
        return np.where((data>upper)|(data<lower))
```

- 이상치 제거

```
In [ ]: x_scaler = MinMaxScaler()
        x_train_scaled = x_scaler.fit_transform(x_train)
        print("Min:", np.min(x_train_scaled))
        print("Max:", np.max(x_train_scaled))
        x_test_scaled = x_scaler.transform(x_test)
        y_scaler = MinMaxScaler()
        y_train_scaled = y_scaler.fit_transform(y_train)
        y_test_scaled = y_scaler.transform(y_test)
        print(x_train_scaled.shape)
        print(y_train_scaled.shape)
```

- 스케일링 : **MinMaxScaler()**

3) Scaling

Train Data

8

:

Test Data

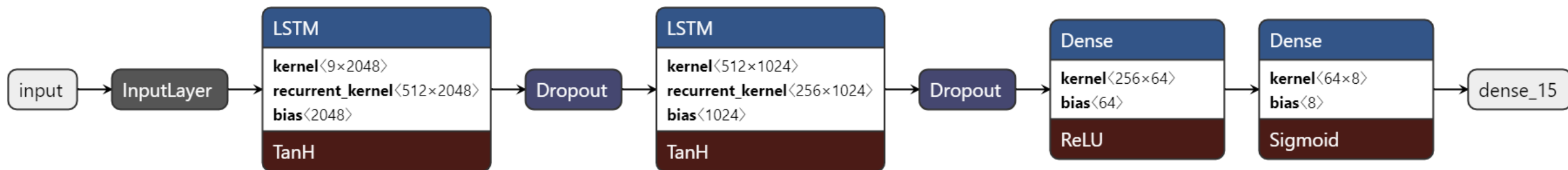
2

5) 모델링

```
In [ ]: def batch_generator(batch_size, sequence_length):  
    while True:  
        x_shape = (batch_size, sequence_length, num_x_signals)  
        x_batch = np.zeros(shape=x_shape, dtype=np.float16)  
        y_shape = (batch_size, sequence_length, num_y_signals)  
        y_batch = np.zeros(shape=y_shape, dtype=np.float16)  
  
        for i in range(batch_size):  
            idx = np.random.randint(num_train - sequence_length)  
            x_batch[i] = x_train_scaled[idx:idx+sequence_length]  
            y_batch[i] = y_train_scaled[idx:idx+sequence_length]  
        yield (x_batch, y_batch)
```

데이터 양이 많으므로 전체 데이터를 학습하지 않고
특정 기간을 랜덤하게 묶어서 입력데이터로 사용

5) 모델링



5) 모델링

```
In [228]: path_checkpoint = 'checkpoint.keras'
callback_checkpoint = ModelCheckpoint(filepath=path_checkpoint,monitor='val_loss',verbose=1,
                                     save_weights_only=True,save_best_only=True)
callback_early_stopping = EarlyStopping(monitor='val_loss',patience=5, verbose=1)
callback_tensorboard = TensorBoard(log_dir='./modellogs/',histogram_freq=0,write_graph=False)
callbacks = [callback_early_stopping,callback_checkpoint,callback_tensorboard]
```

CallBack Option

- 체크 포인트 기록
- 성능 하락시(과적합) 학습 중지
- TensorBoard 로그 작성

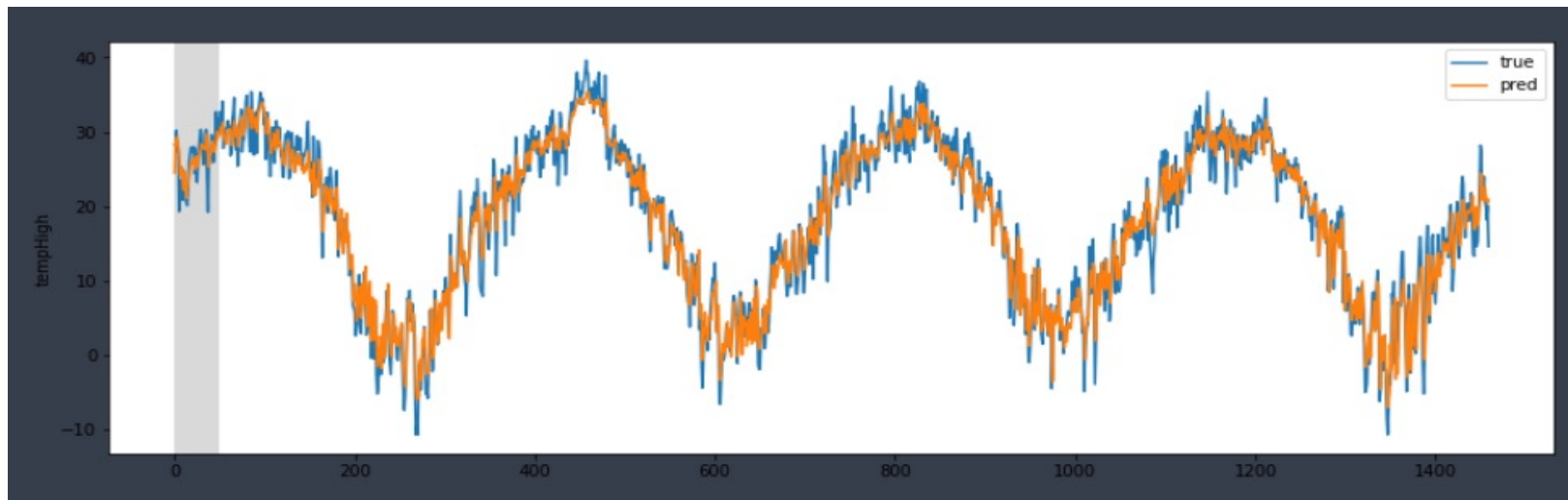
```
In [226]: model.compile(loss=loss_mse_warmup, optimizer='Adam')
model.summary()
```

Optimizer는 Adam, 손실함수는 구간을 별도로 재정의한 MSE를 사용

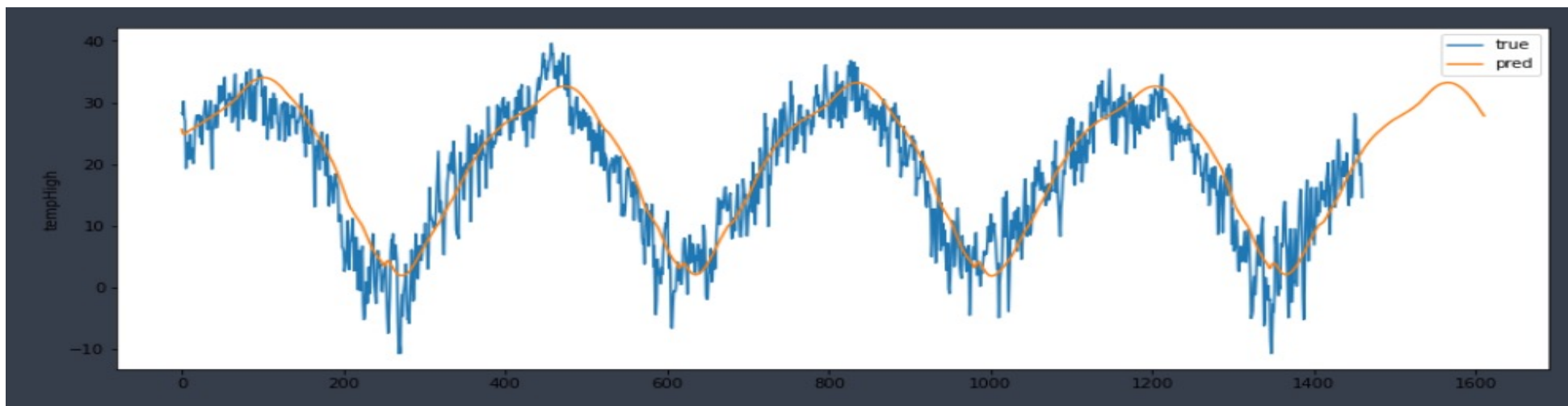
```
In [211]: model.fit(generator,epochs=50,steps_per_epoch=200,validation_data=validation_data,callbacks=callbacks)
```

CallBack에서 과적합시 자동 중지 되는것을 감안하여 Epoch는 여유롭게, Epoch당 step은 200씩

직전까지의 실제 데이터를 바탕으로 계속 다음 날을 예측한 결과



예측한 값을 바탕으로 그 다음날을 연속하여 예측한 결과



CONTENTS

1. 주제

2. 데이터 전처리

3. 데이터 분석

4. 대시보드



기상청



분석적 대시보드 활용

날씨에 영향을 미치는 수 많은 변수들 분석

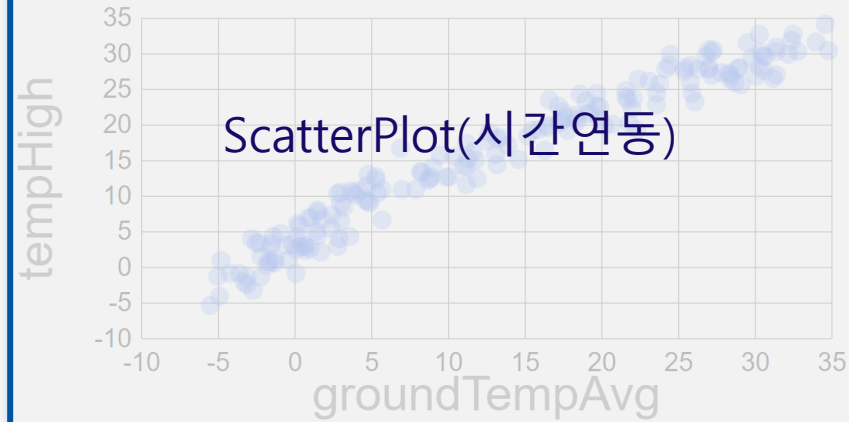
시간연동

Prediction & Analysis 2017 firstHalf

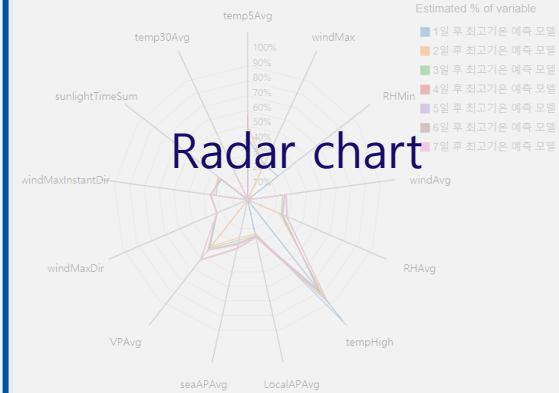
Dashboard

보이는게 전부다 하라

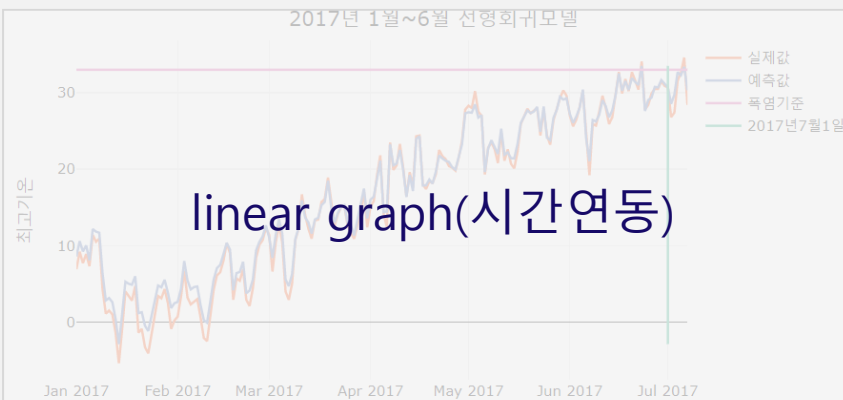
tempHigh vs groundTempAvg



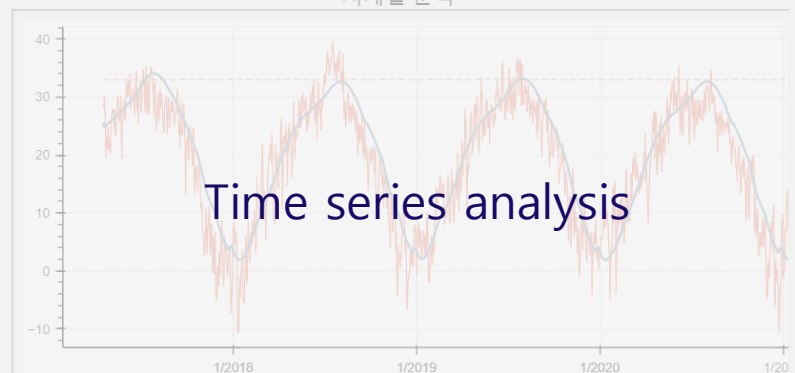
Radar chart



2017년 1월~6월 선형회귀모델

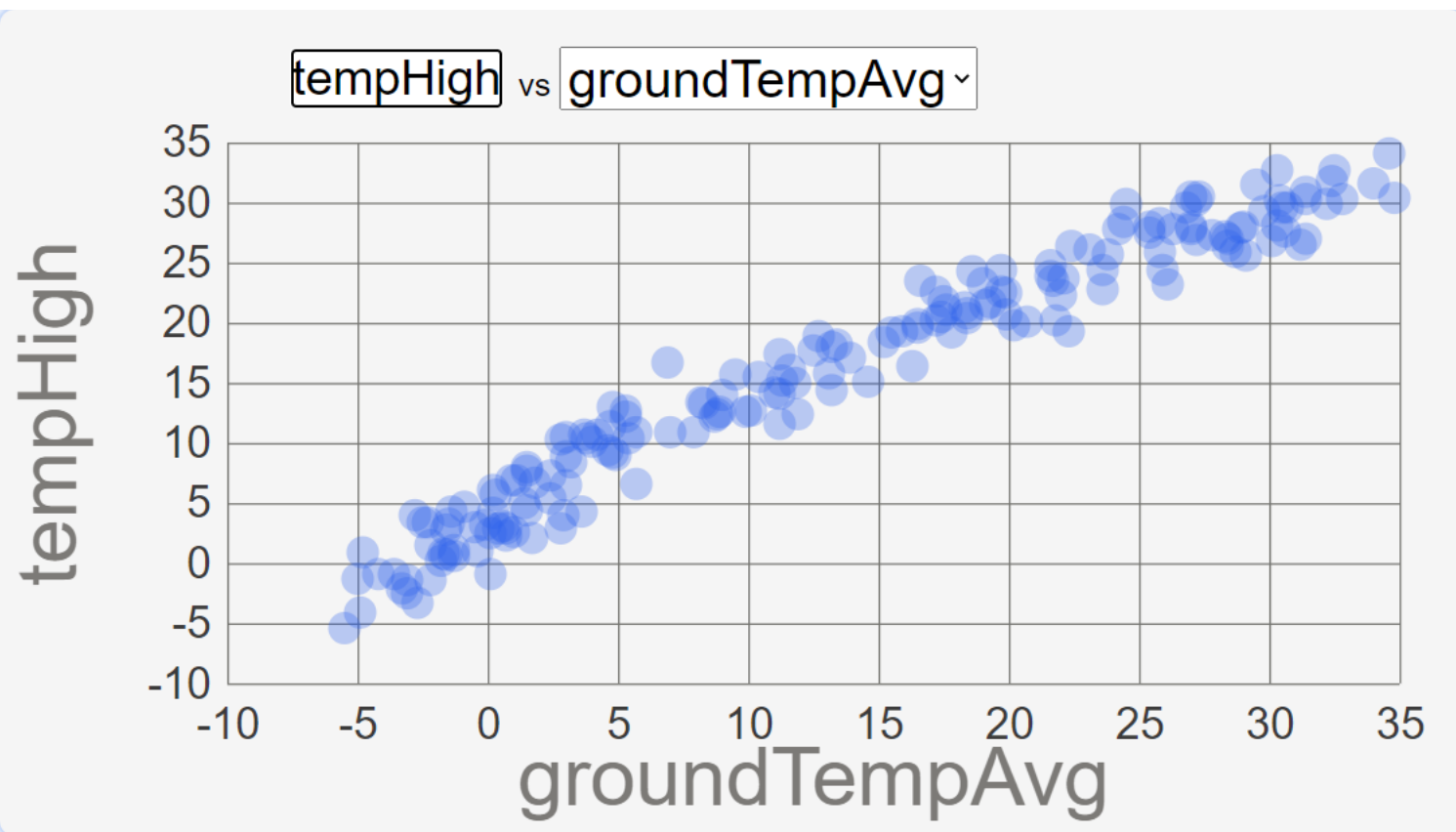


시계열 분석



1. Scatter Plot

Prediction & Analysis 2017 ▾ firstHalf ▾



Scatter Plot을 사용

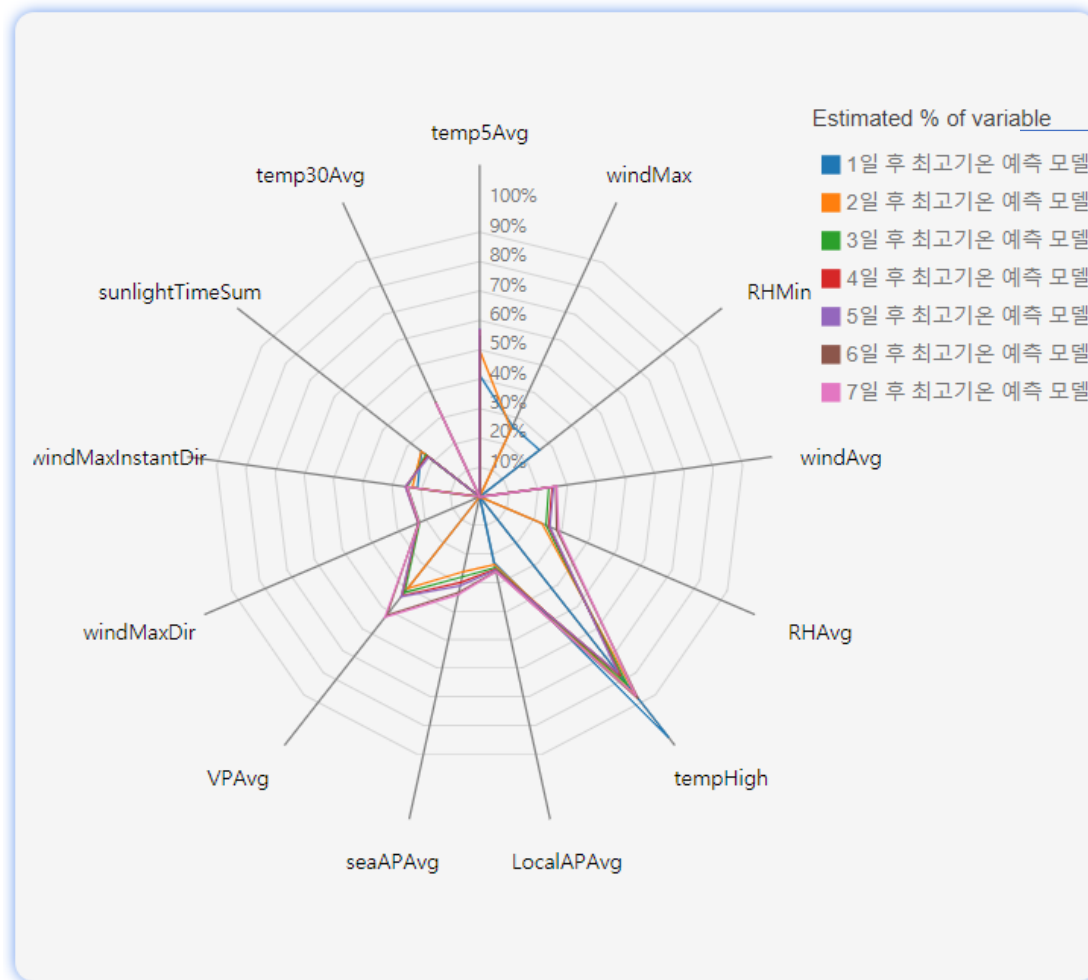


변수 간 상관관계 분석



Dropdown을 통해 데이터 분리
(2017 ~ 2021.6)

2. Radar chart



텍스트에 마우스오버 시 관련 변수의 색을 레이더차트에 표현

선형 회귀 모델 사용



분석모델의 추정치 사용(%)

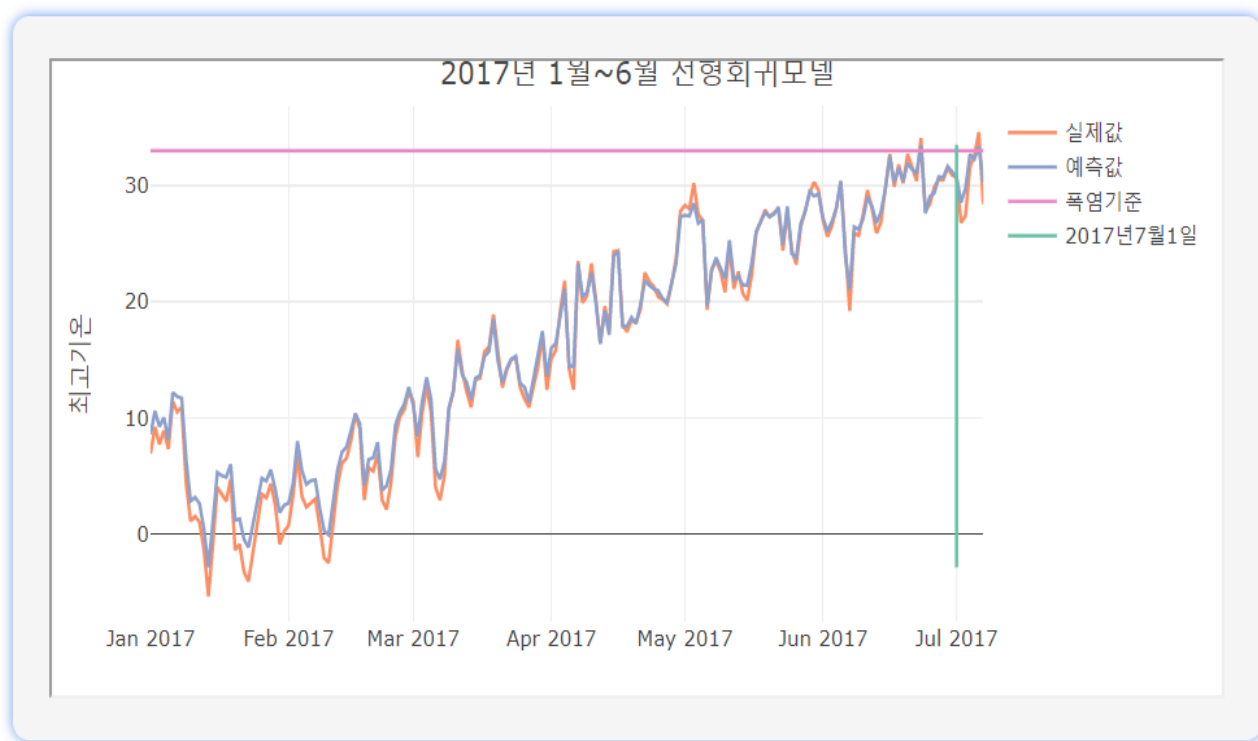


1~7일 후 최고기온 예측에 특정 변수들이 미치는 영향의 정도 시각화

3. Linear regression model graph

Prediction & Analysis 2017~ firstHalf

Scatter Plot과 마찬가지로 시간연동
(Dropdown)



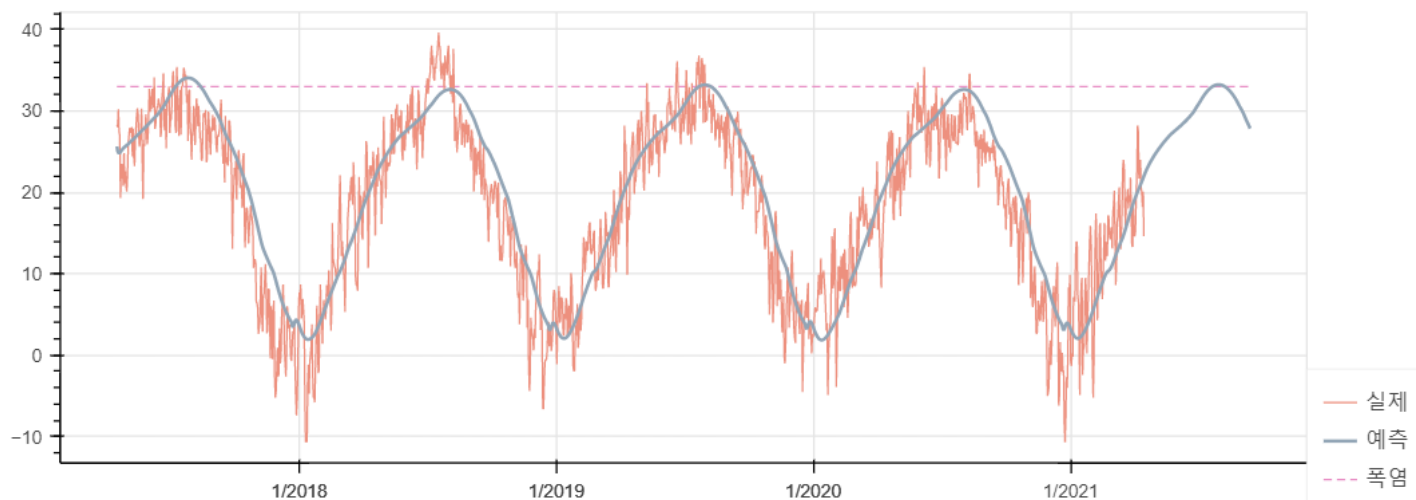
2017~2020.6 까지의 최고기온 분석

마우스 오버 시 각 시간별 온도 표현

실제값과 예측값의 정확도 분석

확대/축소 및 다양한 툴 사용

4. Time series analysis graph



시계열 분석을 통한 계절의 경향성 파악

X축을 1년 단위의 시간축 설정

폭염기준선 표현(33도 이상 2일 지속 시)

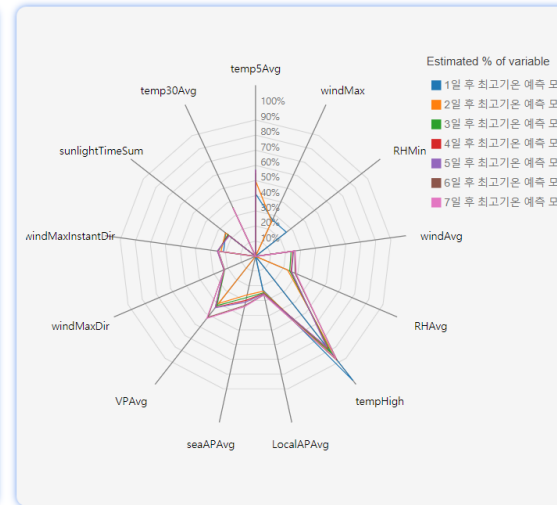
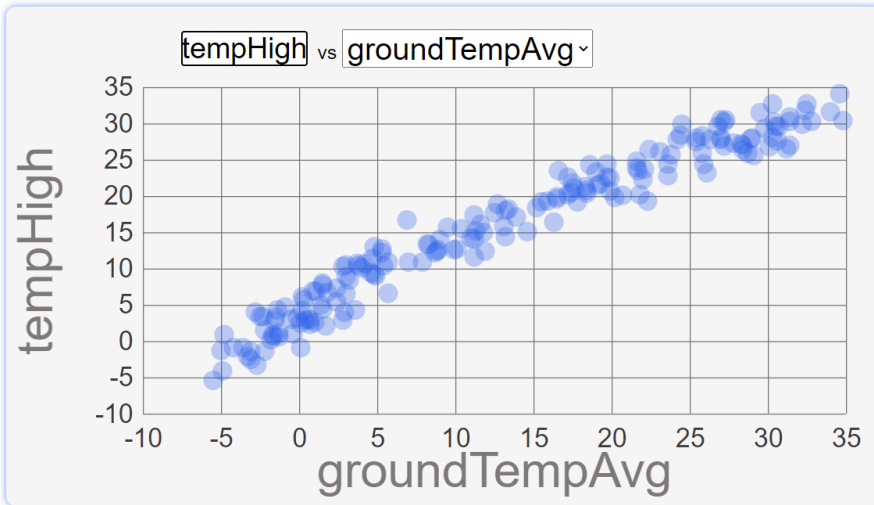
예측에 의하면 2021년 여름은 평균 이하

마우스 오버 시 날짜별 온도 표현



Dashboard

Prediction & Analysis 2017~ firstHalf



Dropdown을 이용하여 시간을 분리

선형회귀모델 그래프와 Scatter Plot을 연동

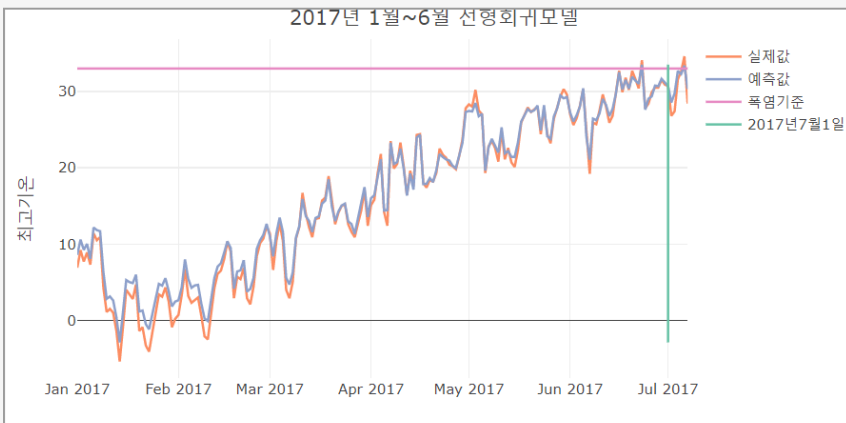
2017 ~ 2020 년 여름까지의 기간을 상반기, 하반기별로 확인 가능

Scatter Plot에 Animation을 넣어 보는 재미 상승

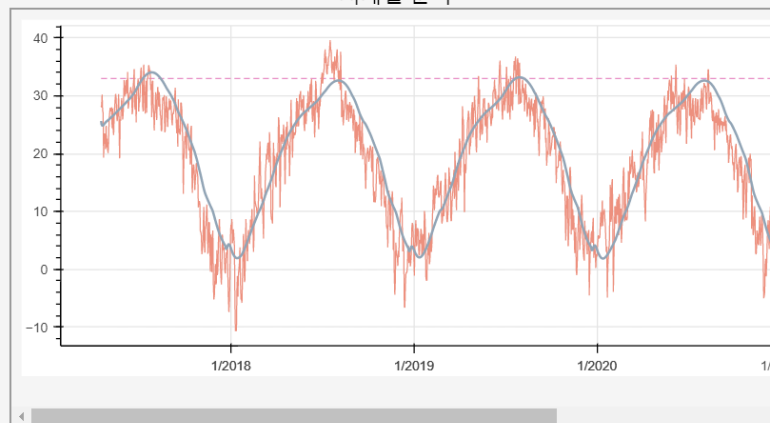
각각 차트와 그래프별 MouseOver시 관련 정보표현

날씨를 분석하는 데 다양한 모델을 사용하려고 시도했기 때문에 그것들을 최대한 표현 하되 **interactive**하게 만들기 위해 노력

2017년 1월~6월 선형회귀모델



시계열 분석



Q&A