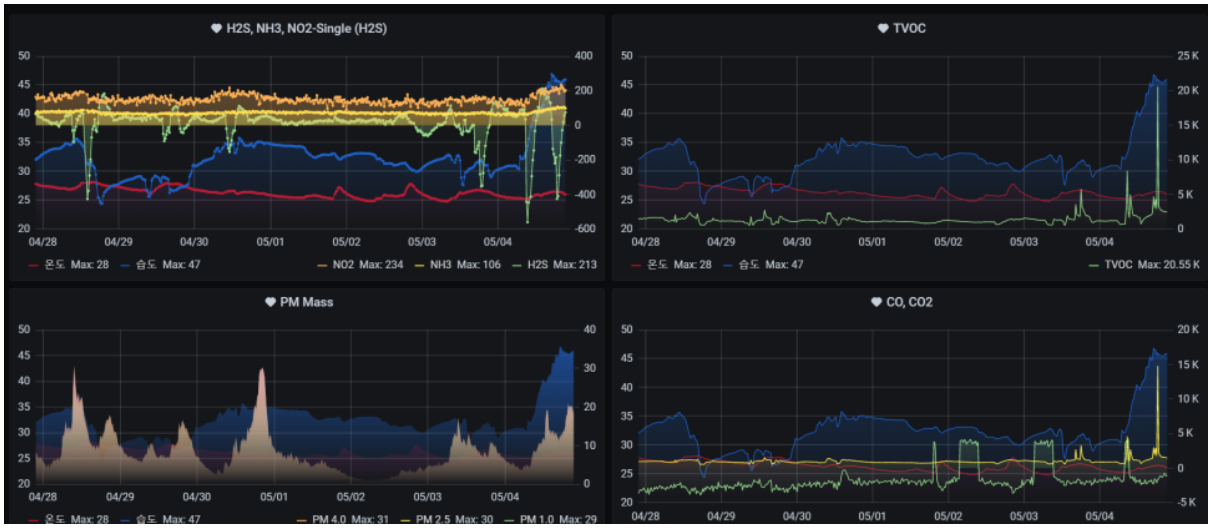
이상치 검출

Grafana 데이터 사용



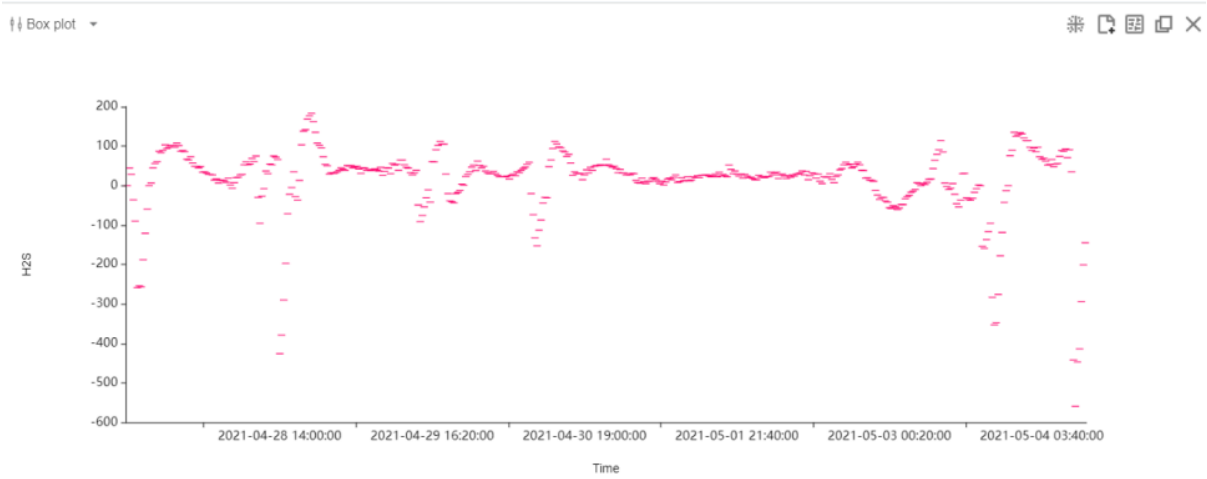
Time	온도	습도	H2S	NH3	NO2
#######	28	32	30	73	173
#######	28	32	44	78	170
#######	28	32	28	80	175
#######	28	32	-36	73	184
#######	28	32	-90	71	158
#######	28	31	-259	82	167
#######	28	31	-254	73	190
#######	28	31	-257	80	173
#######	28	30	-188	79	184
#######	28	31	-121	79	158
#######	28	31	-60	79	161
#######	28	31	0	74	173
#######	28	31	6	75	131
#######	28	31	45	74	173
#######	28	31	56	78	164
#######	28	31	61	80	149
#######	28	32	87	80	175
#######	28	32	83	83	152
#######	28	31	89	72	155
#######	28	31	103	80	155

- 이 중 H2S 열 선택

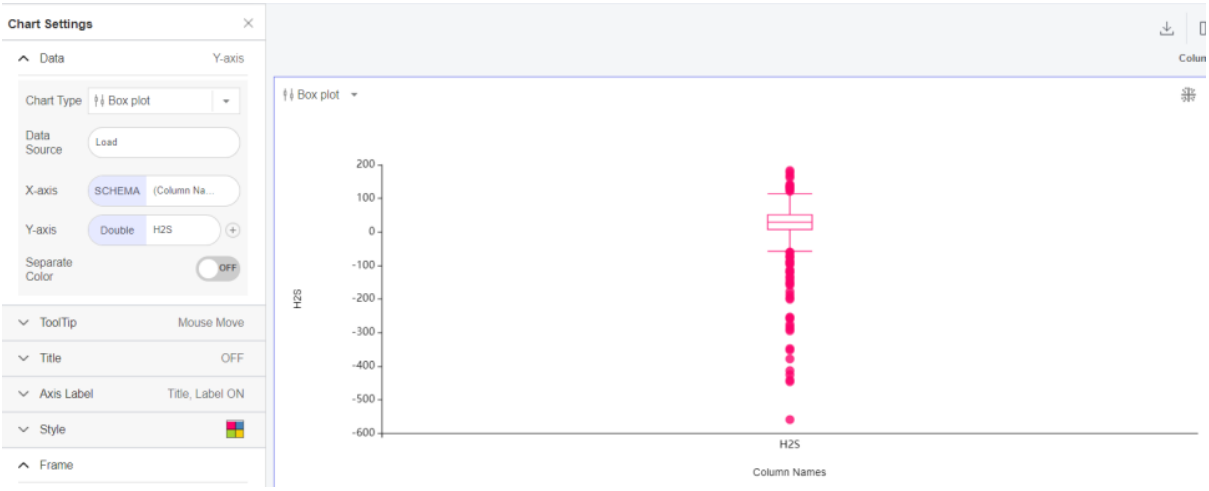
- 이상치 검출

1. 삼성SDS Brightics를 활용한 이상치 탐지

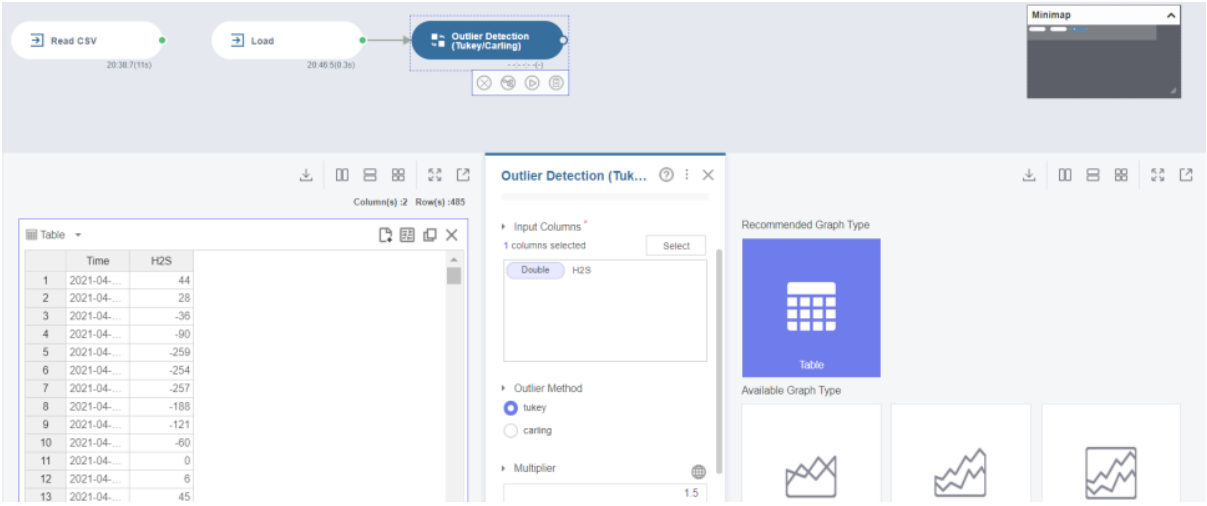
(1) 초기 데이터 Box plot 그리기



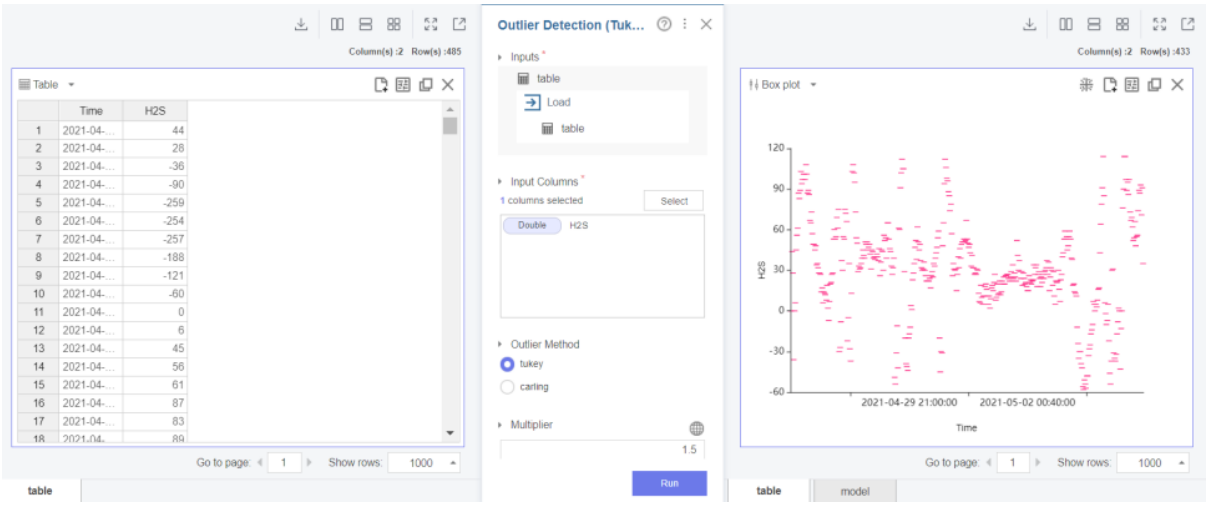
(2) 초기데이터의 X-Axis를 column name으로 두고 다시 plot



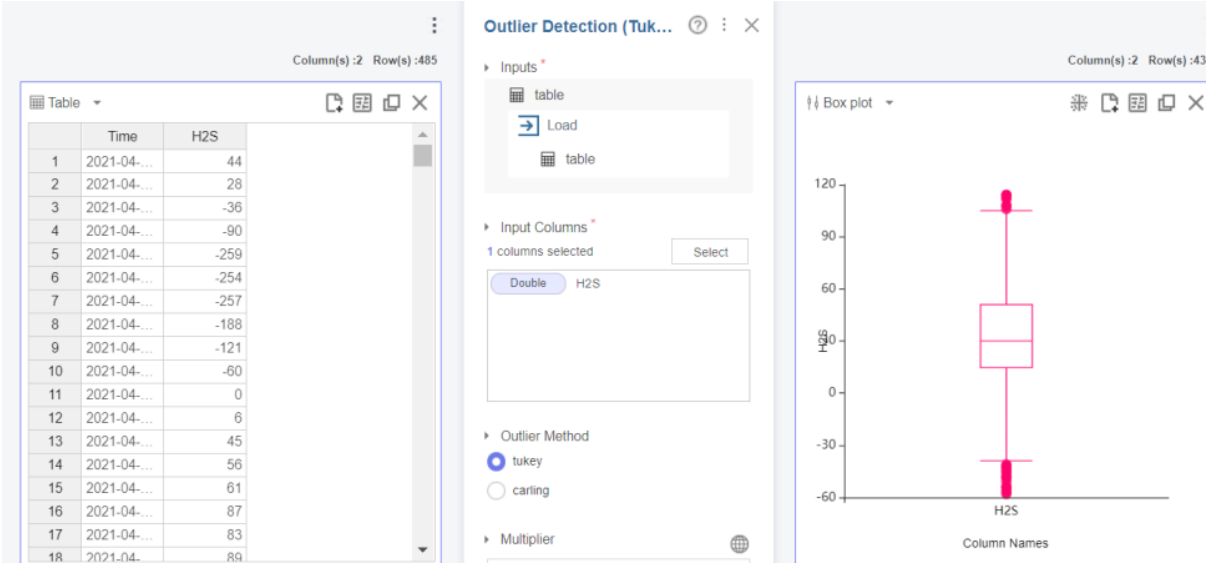
(3) Outlier Detection 중 Turkey/Carling 선택



(4) 모델 구동



(5) X-Axis를 Column name으로 설정



- **이상치 거의 제거됨을 볼 수 있음**
- **시각화 모델이 다양하고 편리**

2. Python 활용한 Anomaly Detection

(1) Turkey Fences

- 데이터 읽기

```
#데이터 읽기(H2S.csv파일)

df=pd.read_csv('H2S.csv')
df.head()
```

	Time	H2S
0	2021-04-27 11:40:00	44
1	2021-04-27 12:00:00	28
2	2021-04-27 12:20:00	-36
3	2021-04-27 12:40:00	-90
4	2021-04-27 13:00:00	-259

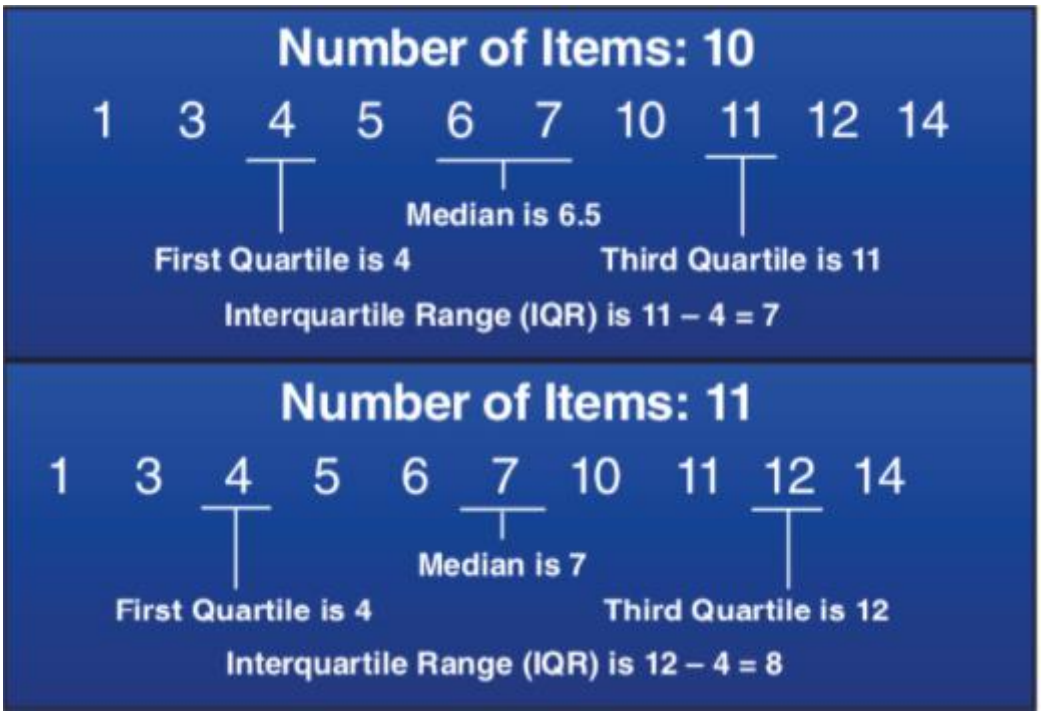
- **outlier_iqr 함수 정의**

```
#1.Turkey Fences
#사분위 범위(IQR,interquartile range)기반
#Q1은 첫번째 25%, Q3는 세번째 25% 보유
#IQR= Q3-Q1

def outliers_iqr(data):
    q1,q3=np.percentile(data,[25,75])
    iqr=q3-q1
    lower_bound=q1-(iqr*1.5)
    upper_bound=q3+(iqr*1.5)
    return np.where((data>upper_bound)|(data<lower_bound))

#np.where()함수는 조건에 만족하는 아이템 반환
#outlier_iqr() 함수는 첫 번째 요소가 이상치(outlier)값을 갖는 행의 인덱스 배열인 튜플 반환
```

-

q1 은 데이터의 처음 25%, q3 는 데이터의 3 번째 25%(75%)를 의미

IQR=Q3-Q1 이라고 생각하면 됨



-

- **이상치 검출**

```
In [64]: for i in outliers_iqr(df.H2S)[0]:
             print(df[i:i+1])
```

```
                   Time  H2S
3   2021-04-27 12:40:00  -90
                   Time  H2S
4   2021-04-27 13:00:00 -259
                   Time  H2S
5   2021-04-27 13:20:00 -254
                   Time  H2S
6   2021-04-27 13:40:00 -257
                   Time  H2S
7   2021-04-27 14:00:00 -188
                   Time  H2S
8   2021-04-27 14:20:00 -121
                   Time  H2S
9   2021-04-27 14:40:00  -60
                    Time  H2S
66  2021-04-28 09:40:00  -96
                    Time  H2S
76  2021-04-28 14:00:00 -426
                    Time  H2S
77  2021-04-28 14:20:00 -379
                    Time  H2S
78  2021-04-28 14:40:00 -290
```

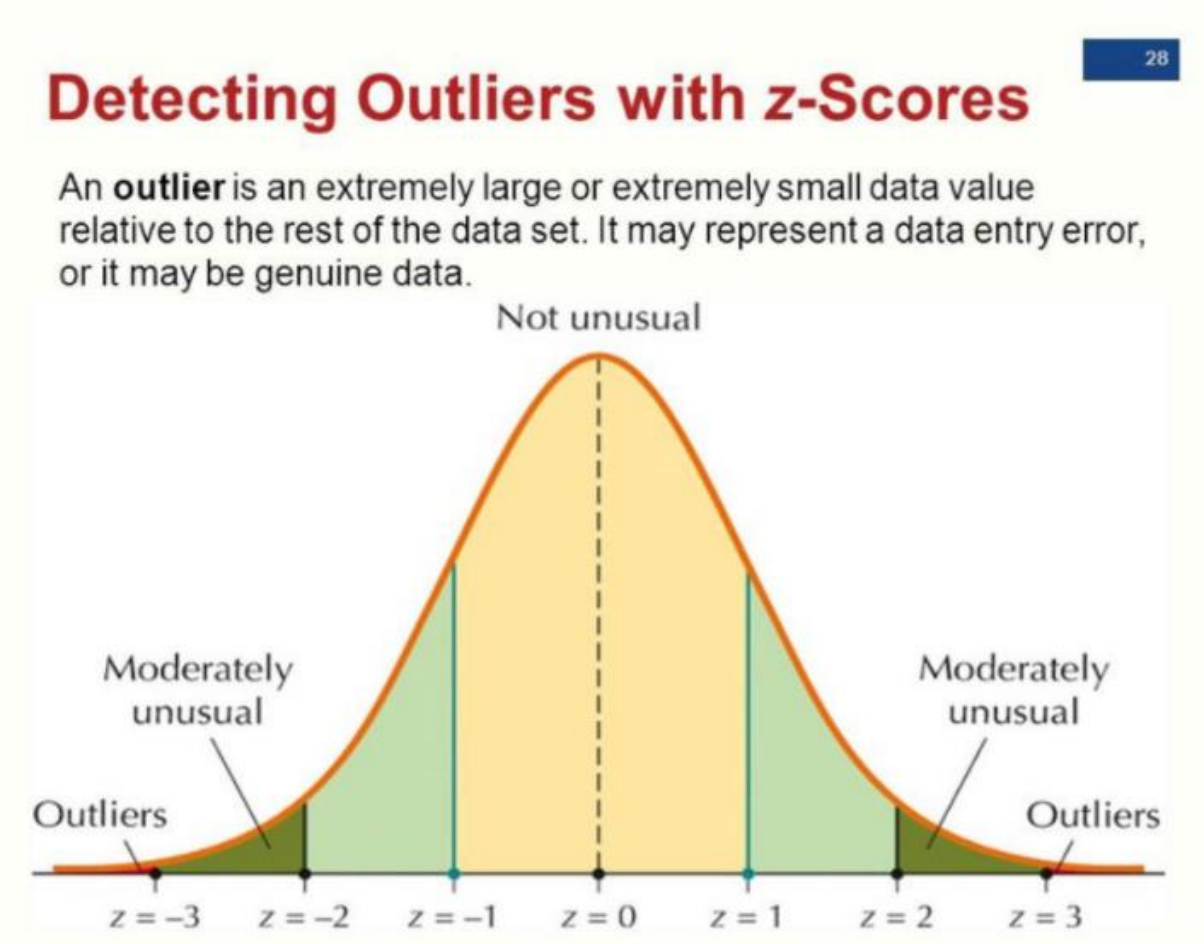
- 이상치 잘 detect함을 볼 수 있음

(2) Z-Score

- outlier_z_score 함수 정의

```
In [75]: #2.Z-score

         def outlier_z_score(data):
             threshod=3
             mean=np.mean(data)
             std=np.std(data)
             z_scores=[(y-mean)/std for y in data]
             return np.where(np.abs(z_scores)>threshold)
```

여기서 threshold 값이 중요한데,
어느 정도의 threshold 값을 매기냐에 따라 검출 가능한 이상치의 범위가 정해진다.
- 데이터 포인트의 68%는 +-1 표준 편차 사이에 있다.
- 데이터 포인트의 95%가 +-2 표준 편차 사이에 있다.
- 데이터 포인트의 99.7%가 +-3 표준 편차 사이에 있다.
라고 생각하면 된다.



- **이상치 검출**

In [73]:
```
for i in outlier_z_score(df.H2S)[0]:
                    print(df[i:i+1])
```

```
                    Time  H2S
3  2021-04-27 12:40:00  -90
                    Time  H2S
4  2021-04-27 13:00:00 -259
                    Time  H2S
5  2021-04-27 13:20:00 -254
                    Time  H2S
6  2021-04-27 13:40:00 -257
                    Time  H2S
7  2021-04-27 14:00:00 -188
                    Time  H2S
8  2021-04-27 14:20:00 -121
                     Time  H2S
18  2021-04-27 17:40:00  103
                     Time  H2S
21  2021-04-27 18:40:00  101
                     Time  H2S
22  2021-04-27 19:00:00  101
                     Time  H2S
```