

```
In [1]: # 필요 라이브러리 로드
        from tqdm import tqdm
```

```
In [2]: # 필요 라이브러리 로드
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import seaborn as sns
#출력 그림 크기 조절
sns.set(rc={'figure.figsize':(12,12)})
plt.style.use('ggplot')
from matplotlib import font_manager, rc
font_path = "C:/Windows/Fonts/NGULIM.TTF"
font = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font)

import warnings
warnings.filterwarnings("ignore")
```

```
In [3]: df = pd.read_csv("행정동_월별_배출량.csv", encoding = "cp949")
```

```
In [4]: df
```

```
Out[4]:
```

	emd_nm	y_m	em_g
0	외도동	2018-01	95864700
1	외도동	2018-02	88582050
2	외도동	2018-03	97356250
3	외도동	2018-04	88844750
4	외도동	2018-05	94064300
...
1671	한경면	2021-02	17276700
1672	한경면	2021-03	21007400
1673	한경면	2021-04	22908150
1674	한경면	2021-05	26927600
1675	한경면	2021-06	27060150

1676 rows × 3 columns

성능 비교용 데이터셋 생성

```
In [5]: train_df = df.loc[df["y_m"] < "2021-04"]
```

```
In [6]: test_df = df.loc[df["y_m"] >= "2021-04"]
```

```
In [7]: temp = pd.DataFrame(data= test_df["emd_nm"].unique())
temp.rename(columns = {0 : "emd_nm"}, inplace=True)
```

```

for i in test_df["y_m"].unique():
    temp = pd.merge(temp, df.loc[df["y_m"] == i, ["emd_nm", "em_g"]], how="right", on = "emd_nm")
    temp.rename(columns = {"em_g" : i}, inplace=True)
temp.to_csv("행정동_월별_배출량_2021_04.csv", encoding = "cp949", index = False)

```

In [8]: temp

Out[8]:

	emd_nm	2021-04	2021-05	2021-06
0	외도동	89457350	97421950	100096900
1	아라동	94682900	107045700	112190800
2	노형동	212851000	227061100	239069250
3	도두동	17377150	19663550	19733350
4	오라동	51424200	57321800	58187300
5	삼양동	84829840	93122870	96172500
6	이도2동	192527200	215882550	225423200
7	일도2동	131462200	147183350	147438200
8	건입동	35386850	38047500	36493300
9	화북동	99636900	109316800	110750050
10	봉개동	13278500	15601950	16048700
11	삼도1동	51074200	57099250	59222300
12	이도1동	39182450	42806500	28492400
13	용담2동	58446350	61815700	62862000
14	연동	167509250	181776100	189432610
15	이호동	19633000	20363000	21586550
16	애월읍	91456441	103211100	103756400
17	동흥동	93011250	94027400	86524400
18	대정읍	101782450	113929800	109850800
19	영천동	41495250	32951350	35074700
20	서흥동	39314300	38265400	29767600
21	효돈동	25200200	21724850	27554100
22	중문동	61827300	68885700	66140250
23	대륜동	44380900	46104350	46745900
24	남원읍	61957700	59816600	62927050
25	대천동	53705700	58341550	56178400
26	안덕면	40209250	44037200	46467450
27	천지동	31792150	27860850	31676600
28	예래동	17646850	19494750	18753350

	emd_nm	2021-04	2021-05	2021-06
29	표선면	38444700	36223050	42453750
30	알수없음	1936900	1910750	2157850
31	성산읍	63529350	57482500	85688760
32	정방동	21122400	22329500	21580950
33	송산동	22429600	26770600	31243800
34	중앙동	43809700	39221700	41422700
35	삼도2동	23157600	24945700	24066500
36	일도1동	18833000	18297050	7451200
37	용담1동	28097300	33500950	33693350
38	조천읍	58934750	58813300	63927750
39	구좌읍	33131350	34451200	37807150
40	한림읍	73755600	79295550	82746990
41	한경면	22908150	26927600	27060150

날짜 지정 범위 생성과 시리즈 객체 변환

```
In [9]: # 데이터프레임을 Series로 변환하는 함수
def time_series(df, i):
    # 행정동별로 데이터를 뽑습니다.
    location = df[df['emd_nm']==i]
    ## 날짜 지정 범위는 시작 월부터 2021년 5월 전까지 기준
    index = pd.date_range(min(location['y_m']), '2021-04', freq='BM')
    ## 시리즈 객체로 변환
    ts = pd.Series(location['em_g'].values, index=index)
    return ts
```

행정동별 배출 특성

계절성이 있는 행정동

adf - test는 통계학에서 실행하는 가설 검정의 절차를 따른다.

- 귀무가설: 시계열 자료가 정상 시계열이 아니다.
- 대립 가설: 시계열 자료가 정상성을 만족한다.

pmdarima 패키지에 있는 ADFTest클래스 import

p-value 0.05보다 낮으면 대립가설이 채택 -> 정상시계열이다. -> 차분이 필요하지 않다.

0.05보다 높으면 귀무가설이 채택 -> 정상시계열이 아니다 -> 차분이 필요하다.

```
In [10]: new_df = train_df
```

```
In [11]: from pmdarima.arima import ADFTest
## ARIMA 모델의 차분 여부를 결정하기 위한 단위근 검정
```

[illegible]

A boxplot showing the distribution of the number of children per family. The x-axis ranges from 0.0 to 0.8. The box is red, with a median line at approximately 0.33. The whiskers extend from approximately 0.05 to 0.9.

5

42개의 행정동중 5개를 제외한 37개의 행정동이 정상시계열이 아니다.

파이썬에서 R 시계열 패키지 forecast를 통한 모델링

```
In [15]: # import rpy2
```

```
# from rpy2.objects.packages import importr # rpy2 내의 패키지를 불러올 importr 클래스

# utils = importr('utils') # utils 패키지를 임포트
# utils.install_packages('forecast') # r의 forecast 패키지 설치.
# utils.install_packages('forecastHybrid') # r의 forecastHybrid 패키지 설치
```

시계열 모델 선택과 검증

1. 자기회귀누적이동평균 모델(ARIMA)

```
In [16]: import rpy2.objects as robjects # r 함수를 파이썬에서 사용 가능하게 변환하는 모듈
from rpy2.objects import pandas2ri # 파이썬 자료형과 R 자료형의 호환을 도와주는 모듈

# pandas2ri를 활성화
pandas2ri.activate()

auto_arima = """
def function(ts):
    library(forecast) # forecast 패키지 로드
    d_params = ndiffs(ts) # 시계열 자료의 차분 횟수 계산
    model = auto.arima(ts, max.p=2, d=d_params) # auto.arima 모델 생성
    forecasted_data = forecast(model, h=3) # 이후 3개월(h=3)을 예측
    out_df = data.frame(forecasted_data$mean) # 예측값을 R의 데이터프레임으로 변환
    colnames(out_df) = c('em_g') # em_g 열로 이름을 지정
    out_df
"""

# r() 함수로 r 자료형을 파이썬에서 사용 가능
auto_arima = robjects.r(auto_arima) # str 형식으로 정의된 auto_arima
ts = robjects.r('ts') # r 자료형 time series 자료형으로 만들어주는 함수
c = robjects.r('c') # r 자료형 벡터를 만들어주는 함수

loc_name = []
final_pred1 = []
final_pred2 = []
final_pred3 = []

for i in tqdm(new_df.emd_nm.unique()):
    location = new_df[new_df['emd_nm']==i]
    start_year = int(min(location['y_m'][:4])) ## 시작 년도
    start_month = int(min(location['y_m'][:5])) ## 시작 월
    # R의 ts 함수로 time series 데이터로 변환
    train = ts(location['em_g'], start=c(start_year, start_month), frequency=12)
    # 자동회귀누적이동평균 model
    forecast = auto_arima(train)
    # 3개월 예상, final_pred에 추가
    loc_name.append(i)
    final_pred1.append(pandas2ri.r2py(forecast).values[0])
    final_pred2.append(pandas2ri.r2py(forecast).values[1])
    final_pred3.append(pandas2ri.r2py(forecast).values[2])
    print(i,np.array(pandas2ri.r2py(forecast).values))
```

2%|■■■

1/42 [00:01<01:05, 1.60s/it]

외도동 [[84893533.80947447]

[92014593.45666127]

[90137546.02054177]]

7%|■■■■■

3/42 [00:02<00:37, 1.03it/s]

아라동 [[9.95446602e+07]
[1.10694729e+08]
[1.05399967e+08]]
노형동 [[2.18951305e+08]
[2.32767632e+08]
[2.25583058e+08]]

12%|■■■■■■■■■

5/42 [00:03<00:28, 1.29it/s]
도두동 [[16083648.79755872]
[17565761.59760864]
[17946925.27239759]]
오라동 [[51289802.60598852]
[58050681.62702025]
[60028648.5646284]]

17%|■■■■■■■■■■■

7/42 [00:04<00:21, 1.61it/s]
삼양동 [[80831886.81395541]
[93640383.5747817]
[91380129.01525098]]
이도2동 [[1.67660925e+08]
[1.93390420e+08]
[1.86933972e+08]]

19%|■■■■■■■■■■■■■

8/42 [00:04<00:17, 1.94it/s]
일도2동 [[1.15034912e+08]
[1.28583312e+08]
[1.18235362e+08]]

24%|■■■■■■■■■■■■■■■

10/42 [00:05<00:15, 2.13it/s]
건입동 [[29181228.23125161]
[33455828.2312516]
[32795728.2312516]]
화북동 [[9.51071344e+07]
[1.11418066e+08]
[1.12684119e+08]]

29%|■■■■■■■■■■■■■■■■■

12/42 [00:06<00:13, 2.21it/s]
봉개동 [[12260368.63578491]
[13776684.19673446]
[14344438.56898575]]
삼도1동 [[43038209.30906738]
[49238309.30906738]
[45153264.30906737]]

33%|■■■■■■■■■■■■■■■■■■■

14/42 [00:06<00:09, 2.89it/s]
이도1동 [[32889599.99999999]
[38228249.99999999]
[33045149.99999999]]
용담2동 [[50256549.51305522]
[55490999.51305523]
[51534699.51305524]]

38%|■■■■■■■■■■■■■■■■■■■■■

16/42 [00:07<00:10, 2.47it/s]
연동 [[1.65625077e+08]
[1.82498386e+08]
[1.75568267e+08]]
이호동 [[17132759.34921785]
[20668201.25162677]
[18455044.93015027]]

애월읍

삼양동 8.083189e+07

이도2동	1.676609e+08
일도2동	1.150349e+08
건입동	2.918123e+07
화북동	9.510713e+07
봉개동	1.226037e+07
삼도1동	4.303821e+07
이도1동	3.288960e+07
용담2동	5.025655e+07
연동	1.656251e+08
이호동	1.713276e+07
애월읍	9.243368e+07
동홍동	1.023098e+08
대정읍	1.076730e+08
영천동	4.499550e+07
서흥동	4.310920e+07
효돈동	2.556637e+07
중문동	6.002835e+07
대륜동	4.488739e+07
남원읍	6.260215e+07
대천동	5.664369e+07
안덕면	3.903739e+07
천지동	3.244086e+07
예래동	1.521305e+07
표선면	3.455936e+07
알수없음	2.223482e+06
성산읍	6.001435e+07
정방동	2.308807e+07
송산동	2.335030e+07
중앙동	4.557880e+07
삼도2동	2.396473e+07
일도1동	1.608532e+07
용담1동	2.008296e+07
조천읍	5.952920e+07

0

구좌읍	3.091935e+07
한림읍	6.978055e+07
한경면	2.100740e+07

```
In [18]: #5월 예측값
data_df2 = pd.DataFrame(data = final_pred2, index = loc_name)
data_df2
```

Out[18]:

0

외도동	9.201459e+07
아라동	1.106947e+08
노형동	2.327676e+08
도두동	1.756576e+07
오라동	5.805068e+07
삼양동	9.364038e+07
이도2동	1.933904e+08
일도2동	1.285833e+08
건입동	3.345583e+07
화북동	1.114181e+08
봉개동	1.377668e+07
삼도1동	4.923831e+07
이도1동	3.822825e+07
용담2동	5.549100e+07
연동	1.824984e+08
이호동	2.066820e+07
애월읍	9.460520e+07
동홍동	1.035211e+08
대정읍	1.094690e+08
영천동	4.499550e+07
서홍동	4.294394e+07
효돈동	2.443699e+07
중문동	6.002835e+07
대륜동	4.857432e+07
남원읍	6.260215e+07
대천동	6.799165e+07

	0
안덕면	4.327990e+07
천지동	3.211597e+07
예래동	1.631550e+07
표선면	3.722784e+07
알수없음	2.223482e+06
성산읍	6.001435e+07
정방동	2.305638e+07
송산동	2.592895e+07
중앙동	4.567250e+07
삼도2동	2.892709e+07
일도1동	1.596496e+07
용담1동	2.440795e+07
조천읍	5.952920e+07
구좌읍	3.091935e+07
한림읍	6.978055e+07
한경면	2.100740e+07

```
In [19]: #6월 예측값
data_df3 = pd.DataFrame(data = final_pred3, index = loc_name)
data_df3
```

	0
외도동	9.013755e+07
아라동	1.054000e+08
노형동	2.255831e+08
도두동	1.794693e+07
오라동	6.002865e+07
삼양동	9.138013e+07
이도2동	1.869340e+08
일도2동	1.182354e+08
건입동	3.279573e+07
화북동	1.126841e+08
봉개동	1.434444e+07
삼도1동	4.515326e+07
이도1동	3.304515e+07

용담2동	5.153470e+07
연동	1.755683e+08
이호동	1.845504e+07
애월읍	9.677672e+07
동흥동	1.047142e+08
대정읍	1.112649e+08
영천동	4.499550e+07
서흥동	4.276181e+07
효돈동	2.343977e+07
중문동	6.002835e+07
대륜동	4.782523e+07
남원읍	6.260215e+07
대천동	7.153251e+07
안덕면	4.546953e+07
천지동	3.191713e+07
예래동	2.163726e+07
표선면	4.190595e+07
알수없음	2.223482e+06
성산읍	6.001435e+07
정방동	2.303514e+07
송산동	2.719348e+07
중앙동	4.574025e+07
삼도2동	3.124070e+07
일도1동	1.584460e+07
용담1동	2.316744e+07
조천읍	5.952920e+07
구좌읍	3.091935e+07
한림읍	6.978055e+07
한경면	2.100740e+07

```
In [20]: # ARIMA를 활용한 4,5,6월 예측 데이터프레임 생성
final = pd.merge(data_df1,data_df2,how = "left",on = data_df1.index)
final.rename( columns = {"key_0" : "행정동명","0_x" : "2021-04","0_y" : "2021-05"},inplace=True)
final = pd.merge(final,data_df3,how = "left",on = final.행정동명)
final = final.drop("key_0",axis = 1)
```

```
final.rename(columns = {0:"2021-06"}, inplace=True)
final
```

Out[20]:

	행정동명	2021-04	2021-05	2021-06
0	외도동	8.489353e+07	9.201459e+07	9.013755e+07
1	아라동	9.954466e+07	1.106947e+08	1.054000e+08
2	노형동	2.189513e+08	2.327676e+08	2.255831e+08
3	도두동	1.608365e+07	1.756576e+07	1.794693e+07
4	오라동	5.128980e+07	5.805068e+07	6.002865e+07
5	삼양동	8.083189e+07	9.364038e+07	9.138013e+07
6	이도2동	1.676609e+08	1.933904e+08	1.869340e+08
7	일도2동	1.150349e+08	1.285833e+08	1.182354e+08
8	건입동	2.918123e+07	3.345583e+07	3.279573e+07
9	화북동	9.510713e+07	1.114181e+08	1.126841e+08
10	봉개동	1.226037e+07	1.377668e+07	1.434444e+07
11	삼도1동	4.303821e+07	4.923831e+07	4.515326e+07
12	이도1동	3.288960e+07	3.822825e+07	3.304515e+07
13	용담2동	5.025655e+07	5.549100e+07	5.153470e+07
14	연동	1.656251e+08	1.824984e+08	1.755683e+08
15	이호동	1.713276e+07	2.066820e+07	1.845504e+07
16	애월읍	9.243368e+07	9.460520e+07	9.677672e+07
17	동홍동	1.023098e+08	1.035211e+08	1.047142e+08
18	대정읍	1.076730e+08	1.094690e+08	1.112649e+08
19	영천동	4.499550e+07	4.499550e+07	4.499550e+07
20	서홍동	4.310920e+07	4.294394e+07	4.276181e+07
21	효돈동	2.556637e+07	2.443699e+07	2.343977e+07
22	중문동	6.002835e+07	6.002835e+07	6.002835e+07
23	대륜동	4.488739e+07	4.857432e+07	4.782523e+07
24	남원읍	6.260215e+07	6.260215e+07	6.260215e+07
25	대천동	5.664369e+07	6.799165e+07	7.153251e+07
26	안덕면	3.903739e+07	4.327990e+07	4.546953e+07
27	천지동	3.244086e+07	3.211597e+07	3.191713e+07
28	예래동	1.521305e+07	1.631550e+07	2.163726e+07
29	표선면	3.455936e+07	3.722784e+07	4.190595e+07
30	알수없음	2.223482e+06	2.223482e+06	2.223482e+06
31	성산읍	6.001435e+07	6.001435e+07	6.001435e+07

	행정동명	2021-04	2021-05	2021-06
32	정방동	2.308807e+07	2.305638e+07	2.303514e+07
33	송산동	2.335030e+07	2.592895e+07	2.719348e+07
34	중앙동	4.557880e+07	4.567250e+07	4.574025e+07
35	삼도2동	2.396473e+07	2.892709e+07	3.124070e+07
36	일도1동	1.608532e+07	1.596496e+07	1.584460e+07
37	용담1동	2.008296e+07	2.440795e+07	2.316744e+07
38	조천읍	5.952920e+07	5.952920e+07	5.952920e+07
39	구좌읍	3.091935e+07	3.091935e+07	3.091935e+07
40	한림읍	6.978055e+07	6.978055e+07	6.978055e+07
41	한경면	2.100740e+07	2.100740e+07	2.100740e+07

```
In [21]: final.to_csv("final_1_2021456.csv",encoding = "cp949",index=False)
```

지수평활법

```
In [22]: import rpy2.robjects as robjects # r 함수를 파이썬에서 사용 가능하게 변환하는 모듈
from rpy2.robjects import pandas2ri # 파이썬 자료형과 R 자료형의 호환을 도와주는 모듈

# pandas2ri를 활성화
pandas2ri.activate()

ets = """
    function(ts){
        library(forecast) # forecast 패키지 로드
        model = ets(ts) # AIC가 낮은 지수평활 모델을 찾음
        forecasted_data = forecast(model, h=3) # 이후 3개월(h=3)을 예측
        out_df = data.frame(forecasted_data$mean) # 예측값을 R의 데이터프레임으로 변환
        colnames(out_df) = c('em_g') # amount라는 열로 이름을 지정
        out_df
    }
"""

# r() 함수로 r 자료형을 파이썬에서 사용 가능
ets = robjects.r(ets)# str 형식으로 정의된 ets
ts = robjects.r('ts')# r 자료형 time series 자료형으로 만들어주는 함수
c = robjects.r('c') # r 자료형 벡터를 만들어주는 함수

loc_name = []
final_pred1 = []
final_pred2 = []
final_pred3 = []

for i in tqdm(new_df.emd_nm.unique()):
    location = new_df[new_df['emd_nm']==i]
    start_year = int(min(location['y_m'][:4])) ## 시작 년도
    start_month = int(min(location['y_m'][:5])) ##시작 월
    # R의 ts 함수로 time series 데이터로 변환
    train = ts(location['em_g'], start=c(start_year, start_month), frequency=12)
    # 지수평활법 model
    forecast = ets(train)
    # 3개월 예상값, final_pred에 추가
```

```
loc_name.append(i)
final_pred1.append(pandas2ri.ri2py(forecast).values[0])
final_pred2.append(pandas2ri.ri2py(forecast).values[1])
final_pred3.append(pandas2ri.ri2py(forecast).values[2])
print(i,np.array(pandas2ri.ri2py(forecast).values))
```

2%|■■■|
1/42 [00:00<00:20, 1.99it/s]
외도동 [[87174852.27080067]
[93087256.8444881]
[93247819.82043226]]

7%|■■■■■|
3/42 [00:01<00:18, 2.08it/s]
아라동 [[1.03950141e+08]
[1.14746906e+08]
[1.14090460e+08]]
노형동 [[2.27157087e+08]
[2.38977224e+08]
[2.42072950e+08]]

10%|■■■■■■■|
4/42 [00:01<00:17, 2.13it/s]
도두동 [[15539211.21971884]
[17264706.77666835]
[17398695.28315141]]

12%|■■■■■■■■■|
5/42 [00:02<00:16, 2.22it/s]
오라동 [[52869879.87912866]
[56936021.5159065]
[60334973.67950378]]

14%|■■■■■■■■■■■|
6/42 [00:02<00:16, 2.22it/s]
삼양동 [[77338428.80633053]
[87276349.03970344]
[85641217.91401659]]

17%|■■■■■■■■■■■■■|
7/42 [00:03<00:15, 2.25it/s]
이도2동 [[1.89884596e+08]
[2.02684975e+08]
[2.08237901e+08]]

21%|■■■■■■■■■■■■■■■|
9/42 [00:03<00:14, 2.30it/s]
일도2동 [[1.22420323e+08]
[1.30737193e+08]
[1.34690080e+08]]
건입동 [[29618801.80011652]
[32179962.6053163]
[32490762.28165858]]

24%|■■■■■■■■■■■■■■■■■|
10/42 [00:04<00:13, 2.36it/s]
화북동 [[9.32765337e+07]
[1.00979845e+08]
[1.03432184e+08]]

26%|■■■■■■■■■■■■■■■■■■■|
11/42 [00:04<00:12, 2.40it/s]
봉개동 [[12652784.55886332]
[13340721.43032964]
[13367241.58100004]]
삼도1동


```

62%|███████████████████████████████|
26/42 [00:11<00:06, 2.43it/s]
대전동 [[59315857.61378794]
[59315857.61378794]
[59315857.61378794]]

64%|███████████████████████████████|
27/42 [00:11<00:06, 2.49it/s]
안덕면 [[39450203.54158856]
[39450203.54158856]
[39450203.54158856]]

67%|███████████████████████████████|
28/42 [00:11<00:05, 2.51it/s]
천지동 [[31777964.82829031]
[31777964.82829031]
[31777964.82829031]]

71%|███████████████████████████████|
30/42 [00:12<00:04, 2.50it/s]
예래동 [[17300426.50142905]
[17300426.50142905]
[17300426.50142905]]
표선면 [[34066584.87775268]
[34066584.87775268]
[34066584.87775268]]

76%|███████████████████████████████|
32/42 [00:13<00:04, 2.49it/s]
알수없음 [[2153783.11159438]
[2153783.11159438]
[2153783.11159438]]
성산읍 [[59607515.09162527]
[59607515.09162527]
[59607515.09162527]]

79%|███████████████████████████████|
33/42 [00:13<00:03, 2.44it/s]
정방동 [[23135052.66942503]
[23135052.66942503]
[23135052.66942503]]

81%|███████████████████████████████|
34/42 [00:14<00:03, 2.44it/s]
송산동 [[20041713.96577217]
[20041713.96577217]
[20041713.96577217]]

83%|███████████████████████████████|
35/42 [00:14<00:02, 2.45it/s]
중앙동 [[45293053.71195993]
[45293053.71195993]
[45293053.71195993]]
삼도2동

86%|███████████████████████████████|
36/42 [00:15<00:02, 2.46it/s]
[[25172376.36873215]
[27649614.23687689]
[31252330.38836759]]

90%|███████████████████████████████|
38/42 [00:15<00:01, 2.52it/s]
일도1동 [[16315630.39743951]
[16315630.39743951]
[16315630.39743951]]
용담1동 [[22124230.41506654]

```

[illegible]

Out[23]: 0

외도동	8.717485e+07
아라동	1.039501e+08
노형동	2.271571e+08
도두동	1.553921e+07
오라동	5.286988e+07
삼양동	7.733843e+07
이도2동	1.898846e+08
일도2동	1.224203e+08
건입동	2.961880e+07
화북동	9.327653e+07
봉개동	1.265278e+07
삼도1동	4.478444e+07
이도1동	3.418088e+07
용담2동	5.030738e+07
연동	1.688994e+08
이호동	1.781209e+07
애월읍	9.694066e+07
동홍동	1.019166e+08
대정읍	1.073729e+08
영천동	4.455205e+07

0

서흥동	4.304849e+07
효돈동	2.684533e+07
중문동	6.002781e+07
대륜동	4.550934e+07
남원읍	6.260151e+07
대천동	5.931586e+07
안덕면	3.945020e+07
천지동	3.177796e+07
예래동	1.730043e+07
표선면	3.406658e+07
알수없음	2.153783e+06
성산읍	5.960752e+07
정방동	2.313505e+07
송산동	2.004171e+07
중앙동	4.529305e+07
삼도2동	2.517238e+07
일도1동	1.631563e+07
용담1동	2.212423e+07
조천읍	6.238903e+07
구좌읍	3.230001e+07
한림읍	6.782643e+07
한경면	2.064433e+07

```
In [24]: #5월 예측값
data_df2 = pd.DataFrame(data = final_pred2, index = loc_name)
data_df2
```

Out[24]:

0

외도동	9.308726e+07
아라동	1.147469e+08
노형동	2.389772e+08
도두동	1.726471e+07
오라동	5.693602e+07
삼양동	8.727635e+07
이도2동	2.026850e+08

일도2동	1.307372e+08
건입동	3.217996e+07
화북동	1.009798e+08
봉개동	1.334072e+07
삼도1동	4.639070e+07
이도1동	3.807475e+07
용담2동	5.375048e+07
연동	1.805918e+08
이호동	1.958824e+07
애월읍	1.022843e+08
동홍동	1.019166e+08
대정읍	1.094162e+08
영천동	4.455205e+07
서홍동	4.304849e+07
효돈동	2.684533e+07
중문동	6.002781e+07
대륜동	4.550934e+07
남원읍	6.260151e+07
대천동	5.931586e+07
안덕면	3.945020e+07
천지동	3.177796e+07
예래동	1.730043e+07
표선면	3.406658e+07
알수없음	2.153783e+06
성산읍	5.960752e+07
정방동	2.313505e+07
송산동	2.004171e+07
중앙동	4.529305e+07
삼도2동	2.764961e+07
일도1동	1.631563e+07
용담1동	2.195925e+07
조천읍	6.524933e+07
구좌읍	3.377559e+07

0

한림읍 6.867431e+07

한경면 2.111854e+07

```
In [25]: #6월 예측값  
data_df3 = pd.DataFrame(data = final_pred3, index = loc_name)  
data_df3
```

Out[25]:

0

외도동 9.324782e+07

아라동 1.140905e+08

노형동 2.420729e+08

도두동 1.739870e+07

오라동 6.033497e+07

삼양동 8.564122e+07

이도2동 2.082379e+08

일도2동 1.346901e+08

건입동 3.249076e+07

화북동 1.034322e+08

봉개동 1.336724e+07

삼도1동 4.557313e+07

이도1동 3.798892e+07

용담2동 5.352653e+07

연동 1.786082e+08

이호동 1.991405e+07

애월읍 1.065592e+08

동흥동 1.019166e+08

대정읍 1.114594e+08

영천동 4.455205e+07

서흥동 4.304849e+07

효돈동 2.684533e+07

중문동 6.002781e+07

대륜동 4.550934e+07

남원읍 6.260151e+07

대천동 5.931586e+07

안덕면 3.945020e+07

0

천지동	3.177796e+07
예래동	1.730043e+07
표선면	3.406658e+07
알수없음	2.153783e+06
성산읍	5.960752e+07
정방동	2.313505e+07
송산동	2.004171e+07
중앙동	4.529305e+07
삼도2동	3.125233e+07
일도1동	1.631563e+07
용담1동	2.120676e+07
조천읍	6.810962e+07
구좌읍	3.525116e+07
한림읍	6.952219e+07
한경면	2.159276e+07

```
In [26]: # 지수평활법을 활용한 4,5,6월 예측 데이터프레임 생성
final = pd.merge(data_df1,data_df2,how = "left",on = data_df1.index)
final.rename( columns = {"key_0" : "행정동명","0_x" : "2021-04","0_y" : "2021-05"},inplace=True)
final = pd.merge(final,data_df3,how = "left",on = final.행정동명)
final = final.drop("key_0",axis = 1)
final.rename(columns = {0:"2021-06"},inplace=True)
final
```

Out[26]:	행정동명	2021-04	2021-05	2021-06
0	외도동	8.717485e+07	9.308726e+07	9.324782e+07
1	아라동	1.039501e+08	1.147469e+08	1.140905e+08
2	노형동	2.271571e+08	2.389772e+08	2.420729e+08
3	도두동	1.553921e+07	1.726471e+07	1.739870e+07
4	오라동	5.286988e+07	5.693602e+07	6.033497e+07
5	삼양동	7.733843e+07	8.727635e+07	8.564122e+07
6	이도2동	1.898846e+08	2.026850e+08	2.082379e+08
7	일도2동	1.224203e+08	1.307372e+08	1.346901e+08
8	건입동	2.961880e+07	3.217996e+07	3.249076e+07
9	화북동	9.327653e+07	1.009798e+08	1.034322e+08
10	봉개동	1.265278e+07	1.334072e+07	1.336724e+07
11	삼도1동	4.478444e+07	4.639070e+07	4.557313e+07

	행정동명	2021-04	2021-05	2021-06
12	이도1동	3.418088e+07	3.807475e+07	3.798892e+07
13	용담2동	5.030738e+07	5.375048e+07	5.352653e+07
14	연동	1.688994e+08	1.805918e+08	1.786082e+08
15	이호동	1.781209e+07	1.958824e+07	1.991405e+07
16	애월읍	9.694066e+07	1.022843e+08	1.065592e+08
17	동홍동	1.019166e+08	1.019166e+08	1.019166e+08
18	대정읍	1.073729e+08	1.094162e+08	1.114594e+08
19	영천동	4.455205e+07	4.455205e+07	4.455205e+07
20	서흥동	4.304849e+07	4.304849e+07	4.304849e+07
21	효돈동	2.684533e+07	2.684533e+07	2.684533e+07
22	중문동	6.002781e+07	6.002781e+07	6.002781e+07
23	대륜동	4.550934e+07	4.550934e+07	4.550934e+07
24	남원읍	6.260151e+07	6.260151e+07	6.260151e+07
25	대천동	5.931586e+07	5.931586e+07	5.931586e+07
26	안덕면	3.945020e+07	3.945020e+07	3.945020e+07
27	천지동	3.177796e+07	3.177796e+07	3.177796e+07
28	예래동	1.730043e+07	1.730043e+07	1.730043e+07
29	표선면	3.406658e+07	3.406658e+07	3.406658e+07
30	알수없음	2.153783e+06	2.153783e+06	2.153783e+06
31	성산읍	5.960752e+07	5.960752e+07	5.960752e+07
32	정방동	2.313505e+07	2.313505e+07	2.313505e+07
33	송산동	2.004171e+07	2.004171e+07	2.004171e+07
34	중앙동	4.529305e+07	4.529305e+07	4.529305e+07
35	삼도2동	2.517238e+07	2.764961e+07	3.125233e+07
36	일도1동	1.631563e+07	1.631563e+07	1.631563e+07
37	용담1동	2.212423e+07	2.195925e+07	2.120676e+07
38	조천읍	6.238903e+07	6.524933e+07	6.810962e+07
39	구좌읍	3.230001e+07	3.377559e+07	3.525116e+07
40	한림읍	6.782643e+07	6.867431e+07	6.952219e+07
41	한경면	2.064433e+07	2.111854e+07	2.159276e+07

```
In [27]: # 예측 데이터셋 정리
         final.to_csv("final_2_2021456.csv", encoding = "cp949", index=False)
```

3. STL 분해를 적용한 지수평활법

```
In [28]: from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

# location = time_series(new_df, "구좌읍")
# # STL 분해
# stl = seasonal_decompose(new_df.em_g, freq=12)
# stl.plot()
# plt.show()
```

```
In [29]: import rpy2.robjects as robjects # r 함수를 파이썬에서 사용 가능하게 변환하는 모듈
from rpy2.robjects import pandas2ri # 파이썬 자료형과 R 자료형의 호환을 도와주는 모듈
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

# pandas2ri를 활성화
pandas2ri.activate()
stlm = """
    function(ts){
        library(forecast) # forecast 패키지 로드
        model = stlm(ts, s.window="periodic", method='ets') # STL 분해 후 지수평활법을 통
        forecasted_data = forecast(model, h=3) # 이후 3개월(h=3)을 예측
        out_df = data.frame(forecasted_data$mean) # 예측값을 R의 데이터프레임으로 변환
        colnames(out_df) = c('amount') # amount라는 열로 이름을 지정
        out_df
    }
"""

ets = """
    function(ts){
        library(forecast) # forecast 패키지 로드
        model = ets(ts) # AIC가 낮은 지수평활 모델을 찾음
        forecasted_data = forecast(model, h=3) # 이후 3개월(h=3)을 예측
        out_df = data.frame(forecasted_data$mean) # 예측값을 R의 데이터프레임으로 변환
        colnames(out_df) = c('amount') # amount라는 열로 이름을 지정
        out_df
    }
"""

# r() 함수로 r을 파이썬에서 사용 가능
stlm = robjects.r(stlm)# str 형식으로 정의된 stlm
ets = robjects.r(ets)# str 형식으로 정의된 ets
ts = robjects.r('ts')# r 자료형 time series 자료형으로 만들어주는 함수
c = robjects.r('c') # r 자료형 벡터를 만들어주는 함수

loc_name = []
final_pred1 = []
final_pred2 = []
final_pred3 = []

for i in tqdm(new_df.emd_nm.unique()):
    location = new_df[new_df['emd_nm']==i]
    data_len = len(location)

    start_year = int(min(location['y_m'][:4])) ## 시작 년도
    start_month = int(min(location['y_m'][:5])) ##시작 월
    # R의 ts 함수로 time series 데이터로 변환
    train = ts(location['em_g'], start=c(start_year, start_month), frequency=12)
    # STL 분해를 적용한 지수평활 model
    if data_len > 24:
        forecast = stlm(train)
    # 지수평활 model
    else:
```


29% |

외도동 [[85056673.89059277]]

[92120542.36171003]]

```
[1.13044366e+08]
```

노형동 [[2.23110254e+08]]

```
[2.39203597e+08]]
```

[17256336.4307851]

오라동 [[51372876.44087081]]

[58154402.94679532]]

[85937822.87594652]

이도2동 [[1.80151753e+08]]

[2.03544946e+08]]

[1.29149199e+08]

건입동 [[28776901.54438335]

[33274801.30954733]]

[9.80889269e+07]

봉개동 [[12666493.50100587]]

[13326692.49371009]]

[45924821.14121666]

도 1동 [[32490740.40

[36790609.69637304]

용담2동 [[50072289.54

[53631335.1143288]

[54376581.3578983]]

57% |

연동 [[1.70497675e+08]]

```
[1.82312060e+08]]
```

[19470275.88661582]

```
[19546028.35875276]]
```

에 율음 [[87675095.035

[90804089.94401146]

```
[88839065.3328251  ]]
```

```
[10352599.52700572]
[16326465 85926413]]
```

```
100%|███████████|
42/42 [00:01<00:00, 32.45it/s]
조천음 [[62389029.31384887]
[65249326.22647905]
[68109623.13910922]]
구좌읍 [[32300011.32511281]
[33775587.11432596]
[35251162.9035391 ]]
한림읍 [[67826426.43117933]
[68674308.07903588]
[69522189.72689244]]
한경면 [[20644328.03563642]
[21118544.55799725]
[21592761.08035808]]
```

Out[30]: 0

외도동	8.505667e+07
아라동	1.020446e+08
노형동	2.231103e+08
도두동	1.589809e+07
오라동	5.137288e+07
삼양동	7.778297e+07
이도2동	1.801518e+08
일도2동	1.199618e+08
건입동	2.877690e+07
화북동	8.959770e+07
봉개동	1.266649e+07
삼도1동	4.359437e+07
이도1동	3.249074e+07
용담2동	5.007229e+07
연동	1.704977e+08
이호동	1.776204e+07
애월읍	8.767510e+07
동홍동	8.638699e+07
대정읍	1.003729e+08
영천동	4.229422e+07
서홍동	3.812484e+07

0

효돈동	2.490296e+07
중문동	5.413980e+07
대륜동	4.069795e+07
남원읍	5.760811e+07
대천동	5.313942e+07
안덕면	3.704956e+07
천지동	2.646058e+07
예래동	1.593789e+07
표선면	3.354464e+07
알수없음	2.408617e+06
성산읍	5.973347e+07
정방동	2.006111e+07
송산동	2.083416e+07
중앙동	4.006267e+07
삼도2동	2.463127e+07
일도1동	1.540505e+07
용담1동	2.221341e+07
조천읍	6.238903e+07
구좌읍	3.230001e+07
한림읍	6.782643e+07
한경면	2.064433e+07

```
In [31]: # 5월 예측데이터  
data_df2 = pd.DataFrame(data = final_pred2, index = loc_name)  
data_df2
```

Out[31]: 0

외도동	9.175564e+07
아라동	1.130444e+08
노형동	2.388897e+08
도두동	1.725634e+07
오라동	5.577932e+07
삼양동	8.593782e+07
이도2동	1.959113e+08
일도2동	1.291492e+08

건입동	3.212800e+07
화북동	9.808893e+07
봉개동	1.325374e+07
삼도1동	4.592482e+07
이도1동	3.617267e+07
용담2동	5.363134e+07
연동	1.835431e+08
이호동	1.947028e+07
애월읍	9.080409e+07
동홍동	1.083211e+08
대정읍	1.085494e+08
영천동	4.556777e+07
서홍동	4.787261e+07
효돈동	2.626083e+07
중문동	6.107091e+07
대륜동	4.697659e+07
남원읍	6.817839e+07
대천동	5.940190e+07
안덕면	3.945665e+07
천지동	3.418666e+07
예래동	1.756698e+07
표선면	3.758068e+07
알수없음	2.875259e+06
성산읍	6.197980e+07
정방동	2.498678e+07
송산동	2.877419e+07
중앙동	4.512801e+07
삼도2동	2.682316e+07
일도1동	1.635260e+07
용담1동	2.293061e+07
조천읍	6.524933e+07
구좌읍	3.377559e+07
한림읍	6.867431e+07

0

한경면 2.111854e+07

```
In [32]: # 6월 예측데이터
data_df3 = pd.DataFrame(data = final_pred3, index = loc_name)
data_df3
```

Out[32]: 0

외도동	9.212054e+07
아라동	1.114467e+08
노형동	2.392036e+08
도두동	1.772072e+07
오라동	5.815440e+07
삼양동	8.656674e+07
이도2동	2.035449e+08
일도2동	1.330070e+08
건입동	3.327480e+07
화북동	1.024868e+08
봉개동	1.332669e+07
삼도1동	4.615670e+07
이도1동	3.679061e+07
용담2동	5.437658e+07
연동	1.823121e+08
이호동	1.954603e+07
애월읍	8.883907e+07
동홍동	1.182342e+08
대정읍	1.064497e+08
영천동	4.582238e+07
서흥동	5.023334e+07
효돈동	2.325548e+07
중문동	7.119261e+07
대륜동	4.623043e+07
남원읍	7.202083e+07
대천동	6.098236e+07
안덕면	4.022167e+07
천지동	3.643348e+07

0

예래동	2.047596e+07
표선면	3.984341e+07
알수없음	3.465458e+06
성산읍	6.251869e+07
정방동	2.639494e+07
송산동	3.027445e+07
중앙동	4.265082e+07
삼도2동	3.031325e+07
일도1동	1.632647e+07
용담1동	2.279036e+07
조천읍	6.810962e+07
구좌읍	3.525116e+07
한림읍	6.952219e+07
한경면	2.159276e+07

```
In [33]: # STL 분해를 적용한 지수평활법 4,5,6 예측 데이터 생성
final = pd.merge(data_df1,data_df2,how = "left",on = data_df1.index)
final.rename( columns = {"key_0" : "행정동명","0_x" : "2021-04","0_y" : "2021-05"},inplace=True)
final = pd.merge(final,data_df3,how = "left",on = final.행정동명)
final = final.drop("key_0",axis = 1)
final.rename(columns = {0:"2021-06"},inplace=True)
final
```

```
Out[33]:
```

	행정동명	2021-04	2021-05	2021-06
0	외도동	8.505667e+07	9.175564e+07	9.212054e+07
1	아라동	1.020446e+08	1.130444e+08	1.114467e+08
2	노형동	2.231103e+08	2.388897e+08	2.392036e+08
3	도두동	1.589809e+07	1.725634e+07	1.772072e+07
4	오라동	5.137288e+07	5.577932e+07	5.815440e+07
5	삼양동	7.778297e+07	8.593782e+07	8.656674e+07
6	이도2동	1.801518e+08	1.959113e+08	2.035449e+08
7	일도2동	1.199618e+08	1.291492e+08	1.330070e+08
8	건입동	2.877690e+07	3.212800e+07	3.327480e+07
9	화북동	8.959770e+07	9.808893e+07	1.024868e+08
10	봉개동	1.266649e+07	1.325374e+07	1.332669e+07
11	삼도1동	4.359437e+07	4.592482e+07	4.615670e+07
12	이도1동	3.249074e+07	3.617267e+07	3.679061e+07

	행정동명	2021-04	2021-05	2021-06
13	용담2동	5.007229e+07	5.363134e+07	5.437658e+07
14	연동	1.704977e+08	1.835431e+08	1.823121e+08
15	이호동	1.776204e+07	1.947028e+07	1.954603e+07
16	애월읍	8.767510e+07	9.080409e+07	8.883907e+07
17	동흥동	8.638699e+07	1.083211e+08	1.182342e+08
18	대정읍	1.003729e+08	1.085494e+08	1.064497e+08
19	영천동	4.229422e+07	4.556777e+07	4.582238e+07
20	서흥동	3.812484e+07	4.787261e+07	5.023334e+07
21	효돈동	2.490296e+07	2.626083e+07	2.325548e+07
22	중문동	5.413980e+07	6.107091e+07	7.119261e+07
23	대륜동	4.069795e+07	4.697659e+07	4.623043e+07
24	남원읍	5.760811e+07	6.817839e+07	7.202083e+07
25	대천동	5.313942e+07	5.940190e+07	6.098236e+07
26	안덕면	3.704956e+07	3.945665e+07	4.022167e+07
27	천지동	2.646058e+07	3.418666e+07	3.643348e+07
28	예래동	1.593789e+07	1.756698e+07	2.047596e+07
29	표선면	3.354464e+07	3.758068e+07	3.984341e+07
30	알수없음	2.408617e+06	2.875259e+06	3.465458e+06
31	성산읍	5.973347e+07	6.197980e+07	6.251869e+07
32	정방동	2.006111e+07	2.498678e+07	2.639494e+07
33	송산동	2.083416e+07	2.877419e+07	3.027445e+07
34	중앙동	4.006267e+07	4.512801e+07	4.265082e+07
35	삼도2동	2.463127e+07	2.682316e+07	3.031325e+07
36	일도1동	1.540505e+07	1.635260e+07	1.632647e+07
37	용담1동	2.221341e+07	2.293061e+07	2.279036e+07
38	조천읍	6.238903e+07	6.524933e+07	6.810962e+07
39	구좌읍	3.230001e+07	3.377559e+07	3.525116e+07
40	한림읍	6.782643e+07	6.867431e+07	6.952219e+07
41	한경면	2.064433e+07	2.111854e+07	2.159276e+07

```
In [34]: final.to_csv("final_3_2021456.csv",encoding = "cp949",index=False)
```

성능 비교

```
In [35]: #실제 배출량 값 로드
```



```
df = pd.read_csv("행정동_월별_배출량_2021_04.csv", encoding = "cp949")
#각 모델을 활용한 배출량 값 로드
df1 = pd.read_csv("final_1_2021456.csv", encoding = "cp949")
df2 = pd.read_csv("final_2_2021456.csv", encoding = "cp949")
df3 = pd.read_csv("final_3_2021456.csv", encoding = "cp949")
```

```
In [36]: from sklearn.metrics import make_scorer
import sklearn.metrics as metrics
import numpy as np
```

```
In [37]: # 성능 비교를 위해 지표 함수 생성
def regression_results(y_true, y_pred):
    # Regression metrics
    explained_variance=metrics.explained_variance_score(y_true, y_pred)
    mean_absolute_error=metrics.mean_absolute_error(y_true, y_pred)
    mse=metrics.mean_squared_error(y_true, y_pred)
    mean_squared_log_error=metrics.mean_squared_log_error(y_true, y_pred)
    median_absolute_error=metrics.median_absolute_error(y_true, y_pred)
    r2=metrics.r2_score(y_true, y_pred)

    print('explained_variance: ', round(explained_variance,4))
    print('mean_squared_log_error: ', round(mean_squared_log_error,4))
    print('r2: ', round(r2,4))
    print('MAE: ', round(mean_absolute_error,4))
    print('MSE: ', round(mse,4))
    print('RMSE: ', round(np.sqrt(mse),4))
```

```
In [38]: print("Arima 결과")
print()
print("4월")
regression_results(df["2021-04"], df1["2021-04"])
print()
print("5월")
regression_results(df["2021-05"], df1["2021-05"])
print()
print("6월")
regression_results(df["2021-06"], df1["2021-06"])
```

Arima 결과

4월
explained_variance: 0.9847
mean_squared_log_error: 0.0108
r2: 0.9832
MAE: 3987276.2568
MSE: 36435347931968.59
RMSE: 6036169.9721

5월
explained_variance: 0.9831
mean_squared_log_error: 0.0155
r2: 0.9825
MAE: 4962305.9055
MSE: 46426419659430.18
RMSE: 6813693.5402

6월
explained_variance: 0.96
mean_squared_log_error: 0.0412
r2: 0.9556
MAE: 7946285.146

MSE: 128826441983080.75
RMSE: 11350173.6543

```
In [39]: print("지수평활법 결과")
print()
print("4월")
regression_results(df["2021-04"], df2["2021-04"])
print()
print("5월")
regression_results(df["2021-05"], df2["2021-05"])
print()
print("6월")
regression_results(df["2021-06"], df2["2021-06"])
```

지수평활법 결과

4월
explained_variance: 0.989
mean_squared_log_error: 0.0083
r2: 0.989
MAE: 3805073.4789
MSE: 23940945742115.367
RMSE: 4892948.5734

5월
explained_variance: 0.9846
mean_squared_log_error: 0.0197
r2: 0.9835
MAE: 5257164.0597
MSE: 43779745147747.17
RMSE: 6616626.4174

6월
explained_variance: 0.9758
mean_squared_log_error: 0.0461
r2: 0.9736
MAE: 6747606.7807
MSE: 76699742122353.55
RMSE: 8757838.8957

```
In [40]: print("STL분해를 적용한 지수평활법 결과")
print()
print("4월")
regression_results(df["2021-04"], df3["2021-04"])
print()
print("5월")
regression_results(df["2021-05"], df3["2021-05"])
print()
print("6월")
regression_results(df["2021-06"], df3["2021-06"])
```

STL분해를 적용한 지수평활법 결과

4월
explained_variance: 0.9909
mean_squared_log_error: 0.0116
r2: 0.9869
MAE: 4251519.5825
MSE: 28513011778558.715
RMSE: 5339757.6517

5월
explained_variance: 0.9772
mean_squared_log_error: 0.0242

```
r2: 0.9764
MAE: 6330630.8862
MSE: 62655925301383.46
RMSE: 7915549.5893
```

```
6월
explained_variance: 0.9654
mean_squared_log_error: 0.0504
r2: 0.9645
MAE: 7459888.4979
MSE: 102976521315940.67
RMSE: 10147734.7874
```

In []: