In [1]:
```
## 라이브러리 설치

import numpy as np
import pandas as pd

import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import colour
from colour import Color
import missingno as msno
import json
import geopandas as gpd
import folium

import json
import geopandas as gpd
import folium

from scipy.optimize import leastsq
from scipy.optimize import curve_fit

import random
from scipy.integrate import odeint
from scipy.misc import derivative
import statsmodels.formula.api as smf
from datetime import datetime

from sklearn.linear_model import LinearRegression

import warnings
%matplotlib inline
%config InlineBackend.figure_format = 'retina'

mpl.rc('font', family='NanumGothic') # 폰트 설정
mpl.rc('axes', unicode_minus=False) # 유니코드에서 음수 부호 설정

# 차트 스타일 설정
sns.set(font="NanumGothic", rc={"axes.unicode_minus":False}, style='white')
plt.rc("figure", figsize=(10,8))

warnings.filterwarnings("ignore")
```

In [16]:
```
## 데이터 불러오기

target=pd.read_csv('C://Users//User//Desktop//bass_top5.csv', encoding='cp949')
```

In [17]:
```
target
```

Out[17]:

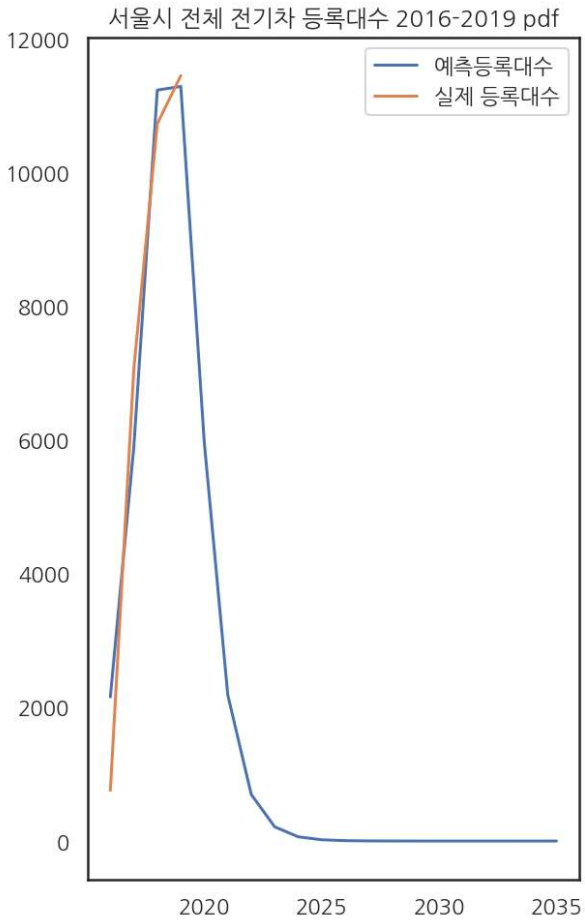
	행정동코드	시도명	시군구명	행정동	2015년	2016년	2017년	2018년	2019년	2020년	2021년
0	11110615	서울	종로구	종로1.2.3.4가동	12	15	37	36	45	64	150
1	11410585	서울	서대문구	신촌동	1	1	4	6	12	20	36
2	11560540	서울	영등포구	여의동	30	93	230	546	1169	1585	1672
3	11650610	서울	서초구	방배2동	0	1	4	19	24	44	74
4	11680640	서울	강남구	역삼1동	15	13	40	141	334	496	727

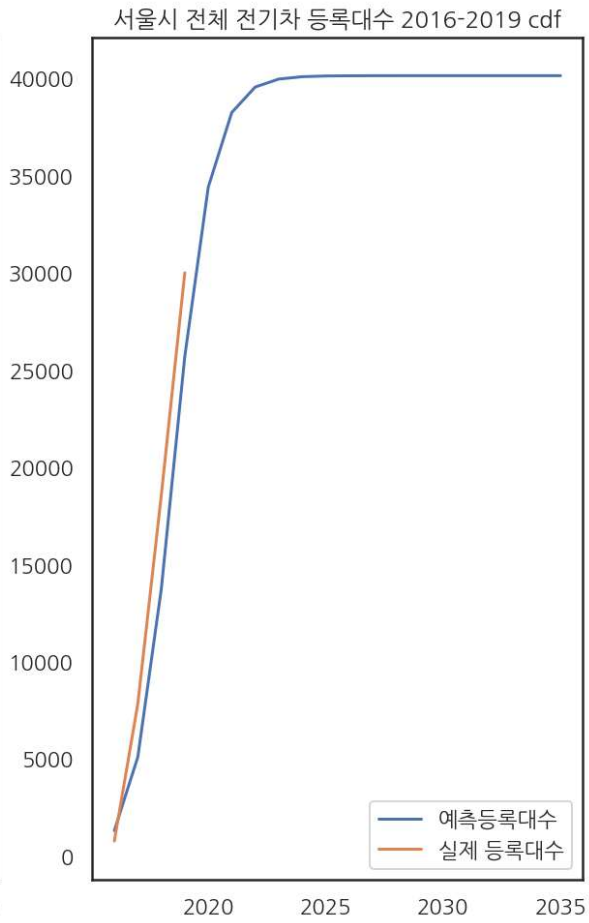
서울시 전체 전기차 판매량 예측

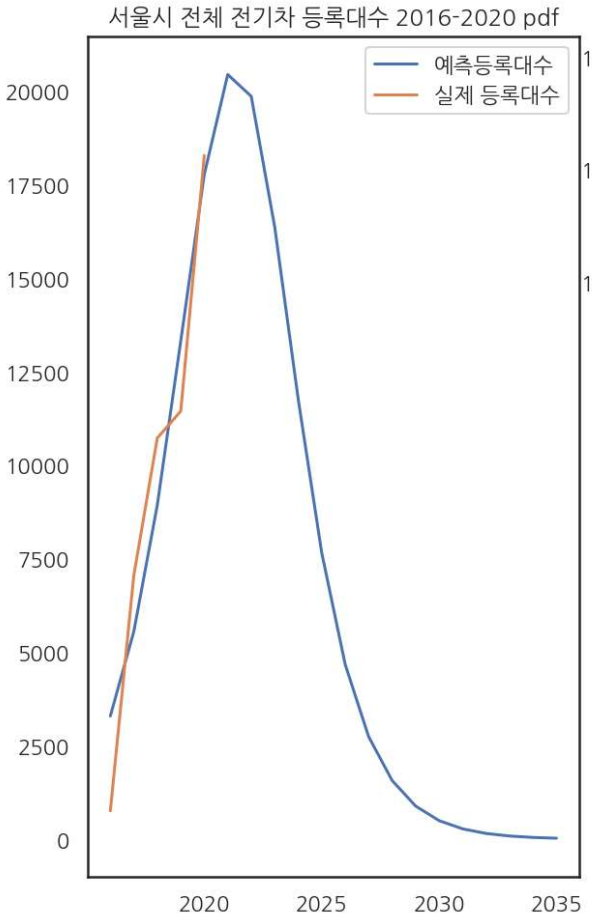
In [2]:
```
date_2019 = [2016, 2017, 2018, 2019]
date_2020 = [2016, 2017, 2018, 2019, 2020]
date_2021 = [2016, 2017, 2018, 2019, 2020, 2021]
date_forecast = [2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2
                     2031,2032, 2033, 2034, 2035]

# residual (error) function
def residual(vars, t, sales):
    M = vars[0]
    P = vars[1]
    Q = vars[2]
    Bass = M * (((P+Q)**2/P)*np.exp(-(P+Q)*t))/(1+(Q/P)*np.exp(-(P+Q)*t))**2
    return (Bass - (sales))

#time intervals
t_2019= np.linspace(1, 4, num=4)
t_2020= np.linspace(1, 5, num=5)
t_2021= np.linspace(1, 6, num=6)

#time interpolation
tp=np.linspace(1, 20, num=20) #2035년까지
```

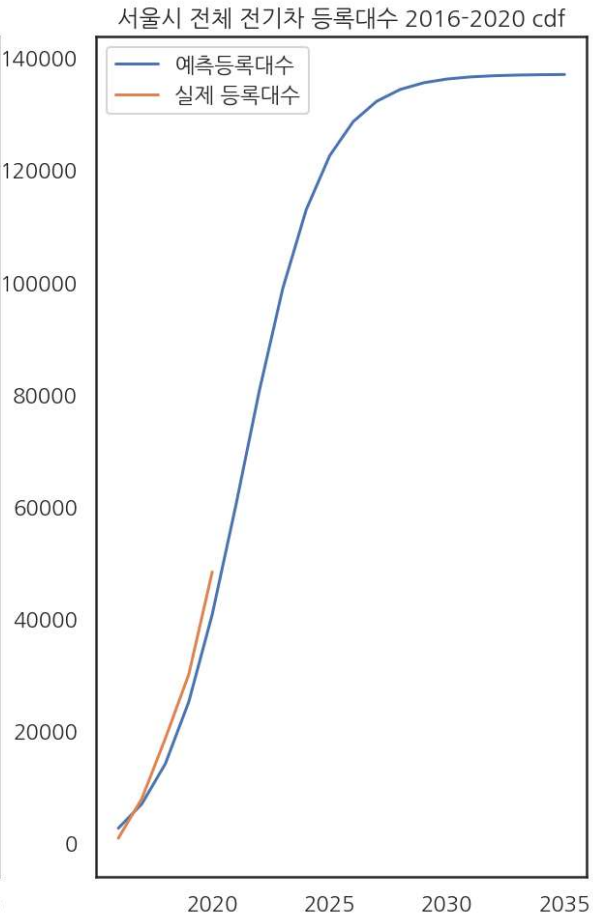
In [8]:
```
# initial variables(M, P & Q)
vars = [500, 0.08,0.58]

# sales vector
sales_2019=np.array([757, 7055, 10727, 11446])
sales_2020=np.array([757, 7055, 10727, 11446, 18280])
sales_2021=np.array([757, 7055, 10727, 11446, 18280, 37576])

# cumulatice sales
csales_2019=np.cumsum(sales_2019)
csales_2020=np.cumsum(sales_2020)
csales_2021=np.cumsum(sales_2021)

# non linear least square fitting
varfinal_2019,success_2019 = leastsq(residual, vars, args=(t_2019, sales_2019))
varfinal_2020,success_2020 = leastsq(residual, vars, args=(t_2020, sales_2020))
varfinal_2021,success_2021 = leastsq(residual, vars, args=(t_2021, sales_2021))

# estimated coefficients
m1 = varfinal_2019[0]
```

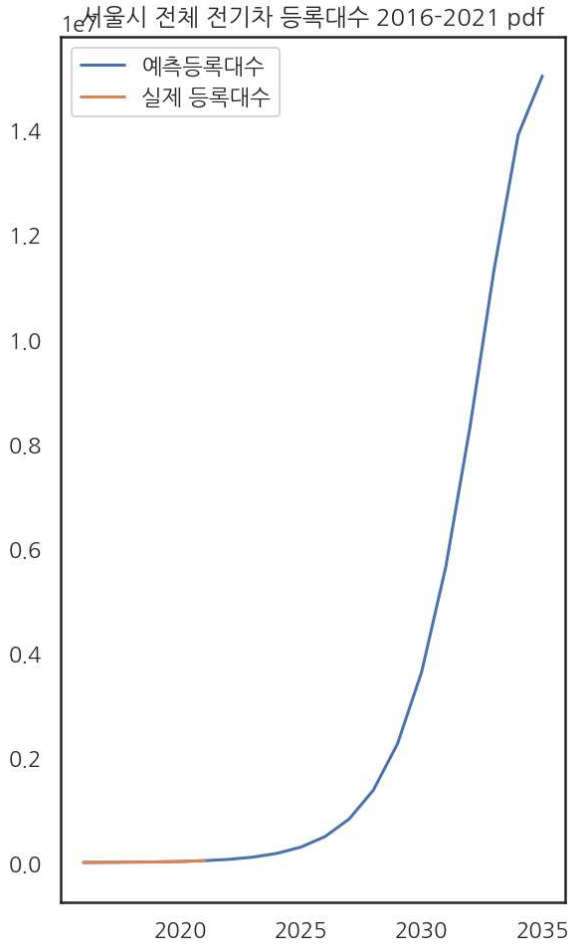
```
p1 = varfinal_2019[1]
q1 = varfinal_2019[2]
m2 = varfinal_2020[0]
p2 = varfinal_2020[1]
q2 = varfinal_2020[2]
m3 = varfinal_2021[0]
p3 = varfinal_2021[1]
q3 = varfinal_2021[2]

#sales plot (pdf)
cofactor_2019= np.exp(-(p1+q1) * tp)
sales_pdf_2019= m1*(((p1+q1)**2/p1)*cofactor_2019)/(1+(q1/p1)*cofactor_2019)**2
cofactor_2020= np.exp(-(p2+q2) * tp)
sales_pdf_2020= m2*(((p2+q2)**2/p2)*cofactor_2020)/(1+(q2/p2)*cofactor_2020)**2
cofactor_2021= np.exp(-(p3+q3) * tp)
sales_pdf_2021= m3*(((p3+q3)**2/p3)*cofactor_2021)/(1+(q3/p3)*cofactor_2021)**2

# Cumulative sales (cdf)
sales_cdf_2019= m1*(1-cofactor_2019)/(1+(q1/p1)*cofactor_2019)
sales_cdf_2020= m2*(1-cofactor_2020)/(1+(q2/p2)*cofactor_2020)
sales_cdf_2021= m3*(1-cofactor_2021)/(1+(q3/p3)*cofactor_2021)

# 시각화
plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2019, date_2019, sales_2019)
plt.title('서울시 전체 전기차 등록대수 2016-2019 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2019, date_2019, csales_2019)
plt.title('서울시 전체 전기차 등록대수 2016-2019 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2020, date_2020, sales_2020)
plt.title('서울시 전체 전기차 등록대수 2016-2020 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2020, date_2020, csales_2020)
plt.title('서울시 전체 전기차 등록대수 2016-2020 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2021, date_2021, sales_2021)
plt.title('서울시 전체 전기차 등록대수 2016-2021 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2021, date_2021, csales_2021)
plt.title('서울시 전체 전기차 등록대수 2016-2021 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

#결과값
pd.DataFrame([sales_cdf_2019, sales_cdf_2020, sales_cdf_2021], columns=date_forecast
```

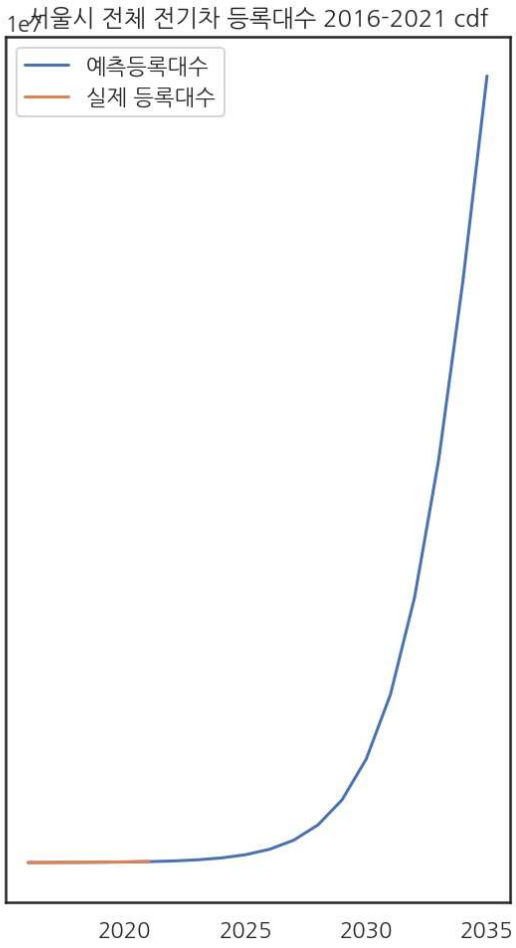

서울시 전체 전기차 등록대수 2016-2019 pdf
예측등록대수
실제 등록대수
12000
10000
8000
6000
4000
2000
0
2020
2025
2030
2035


서울시 전체 전기차 등록대수 2016-2019 cdf
예측등록대수
실제 등록대수
40000
35000
30000
25000
20000
15000
10000
5000
0
2020
2025
2030
2035


서울시 전체 전기차 등록대수 2016-2020 pdf
예측등록대수
실제 등록대수
20000
17500
15000
12500
10000
7500
5000
2500
0
2020
2025
2030
2035


서울시 전체 전기차 등록대수 2016-2020 cdf
예측등록대수
실제 등록대수
140000
120000
100000
80000
60000
40000
20000
0
2020
2025
2030
2035





Out[8]:

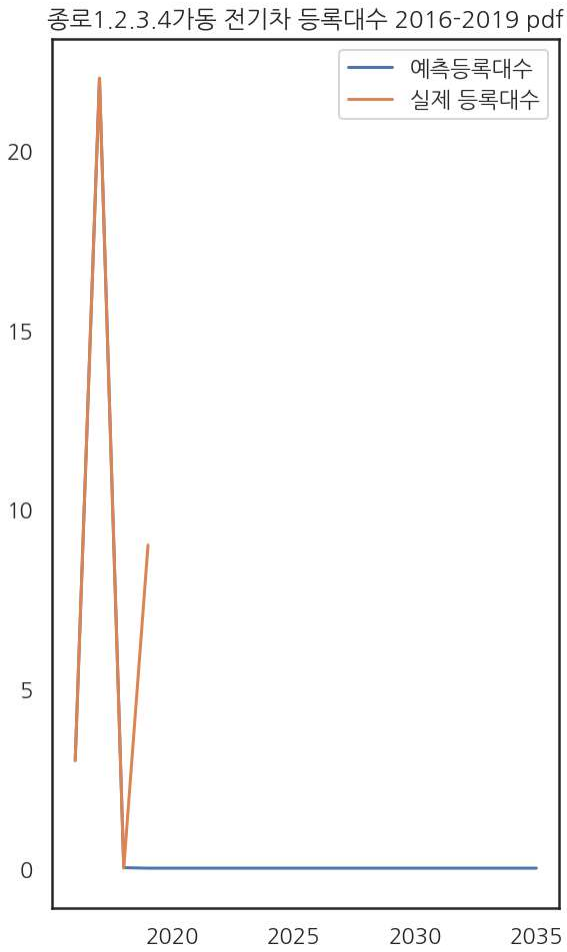
	2016	2017	2018	2019	2020	2021	
2016-2019	1290.896302	5076.380458	13717.910095	25676.247415	34390.485216	38227.225308	39538..
2016-2020	2535.560261	6873.334316	14019.052880	25088.022140	40699.785153	60037.946948	80481..
2016-2021	2018.647516	5438.057970	11230.045465	21040.286277	37654.910204	65788.916484	113416..

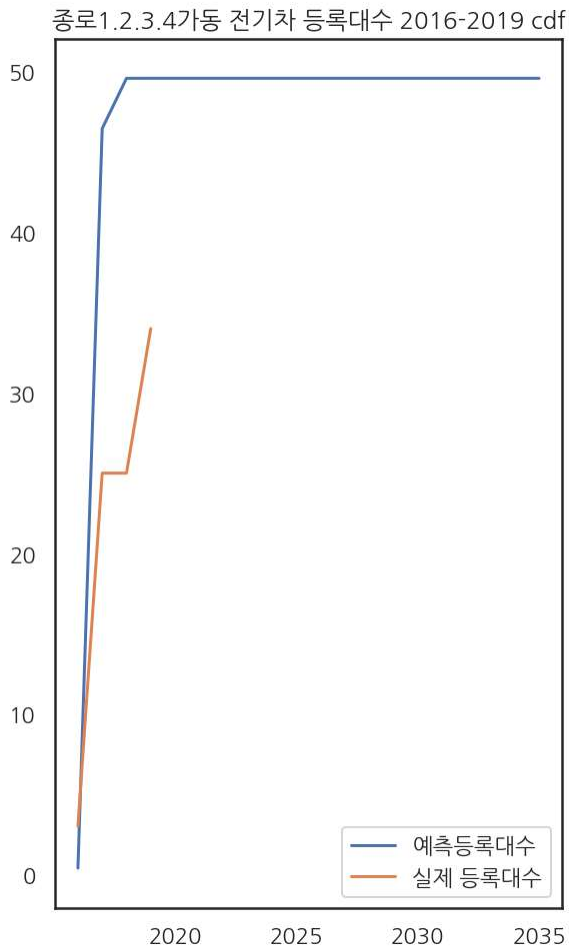
In [7]:
```
# 점점 급증하는 전기차 수로 인해 2019년, 2020년, 2021년으로 갈수록 안정기는 늦게 찾0
```

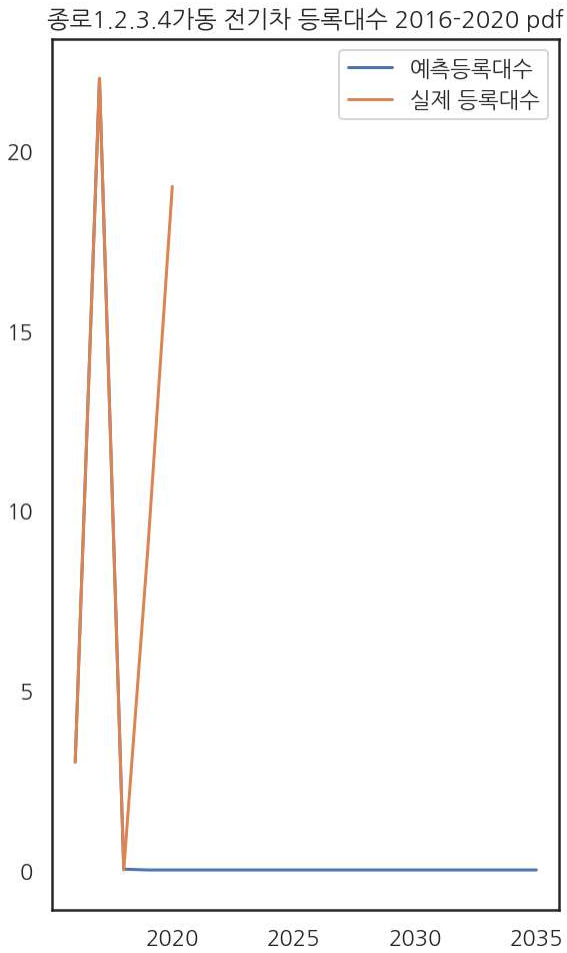
종로1.2.3.4가동

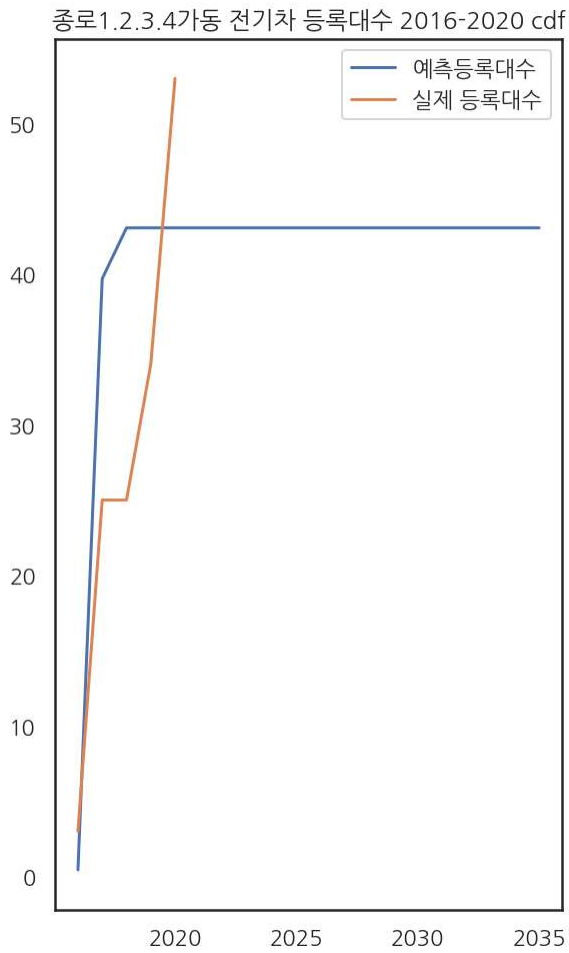
In [13]:
```
# initial variables(M, P & Q)
vars = [3, 0.08,0.58]

# sales vector
sales_2019=np.array([3, 22, 0, 9])
sales_2020=np.array([3, 22, 0, 9, 19])
sales_2021=np.array([3, 22, 0, 9, 19, 86])

# cumulatice sales
csales_2019=np.cumsum(sales_2019)
csales_2020=np.cumsum(sales_2020)
csales_2021=np.cumsum(sales_2021)

# non linear least square fitting
varfinal_2019,success_2019 = leastsq(residual, vars, args=(t_2019, sales_2019))
varfinal_2020,success_2020 = leastsq(residual, vars, args=(t_2020, sales_2020))
```

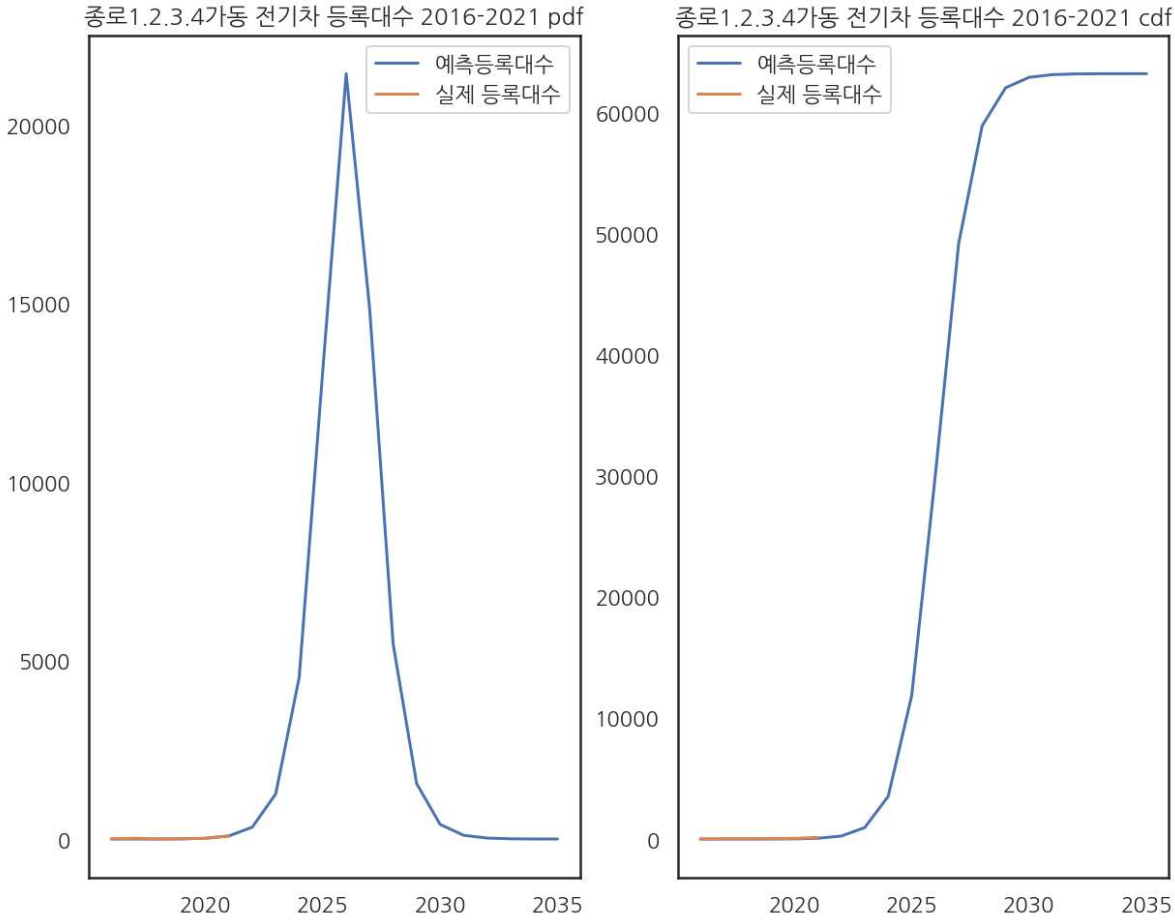
```
varfinal_2021,success_2021 = leastsq(residual, vars, args=(t_2021, sales_2021))

# estimated coefficients
m1 = varfinal_2019[0]
p1 = varfinal_2019[1]
q1 = varfinal_2019[2]
m2 = varfinal_2020[0]
p2 = varfinal_2020[1]
q2 = varfinal_2020[2]
m3 = varfinal_2021[0]
p3 = varfinal_2021[1]
q3 = varfinal_2021[2]

#sales plot (pdf)
cofactor_2019= np.exp(-(p1+q1) * tp)
sales_pdf_2019= m1*(((p1+q1)**2/p1)*cofactor_2019)/(1+(q1/p1)*cofactor_2019)**2
cofactor_2020= np.exp(-(p2+q2) * tp)
sales_pdf_2020= m2*(((p2+q2)**2/p2)*cofactor_2020)/(1+(q2/p2)*cofactor_2020)**2
cofactor_2021= np.exp(-(p3+q3) * tp)
sales_pdf_2021= m3*(((p3+q3)**2/p3)*cofactor_2021)/(1+(q3/p3)*cofactor_2021)**2

# Cumulative sales (cdf)
sales_cdf_2019= m1*(1-cofactor_2019)/(1+(q1/p1)*cofactor_2019)
sales_cdf_2020= m2*(1-cofactor_2020)/(1+(q2/p2)*cofactor_2020)
sales_cdf_2021= m3*(1-cofactor_2021)/(1+(q3/p3)*cofactor_2021)

# 시각화
plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2019, date_2019, sales_2019)
plt.title('종로1.2.3.4가동 전기차 등록대수 2016-2019 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2019, date_2019, csales_2019)
plt.title('종로1.2.3.4가동 전기차 등록대수 2016-2019 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2020, date_2020, sales_2020)
plt.title('종로1.2.3.4가동 전기차 등록대수 2016-2020 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2020, date_2020, csales_2020)
plt.title('종로1.2.3.4가동 전기차 등록대수 2016-2020 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2021, date_2021, sales_2021)
plt.title('종로1.2.3.4가동 전기차 등록대수 2016-2021 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2021, date_2021, csales_2021)
plt.title('종로1.2.3.4가동 전기차 등록대수 2016-2021 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

#결과값
pd.DataFrame([sales_cdf_2019, sales_cdf_2020, sales_cdf_2021], columns=date_forecast
```


종로1.2.3.4가동 전기차 등록대수 2016-2019 pdf
예측등록대수
실제 등록대수
0
5
10
15
20
2020
2025
2030
2035


종로1.2.3.4가동 전기차 등록대수 2016-2019 cdf
예측등록대수
실제 등록대수
0
10
20
30
40
50
2020
2025
2030
2035


종로1.2.3.4가동 전기차 등록대수 2016-2020 pdf
예측등록대수
실제 등록대수
0
5
10
15
20
2020
2025
2030
2035


종로1.2.3.4가동 전기차 등록대수 2016-2020 cdf
예측등록대수
실제 등록대수
0
10
20
30
40
50
2020
2025
2030
2035



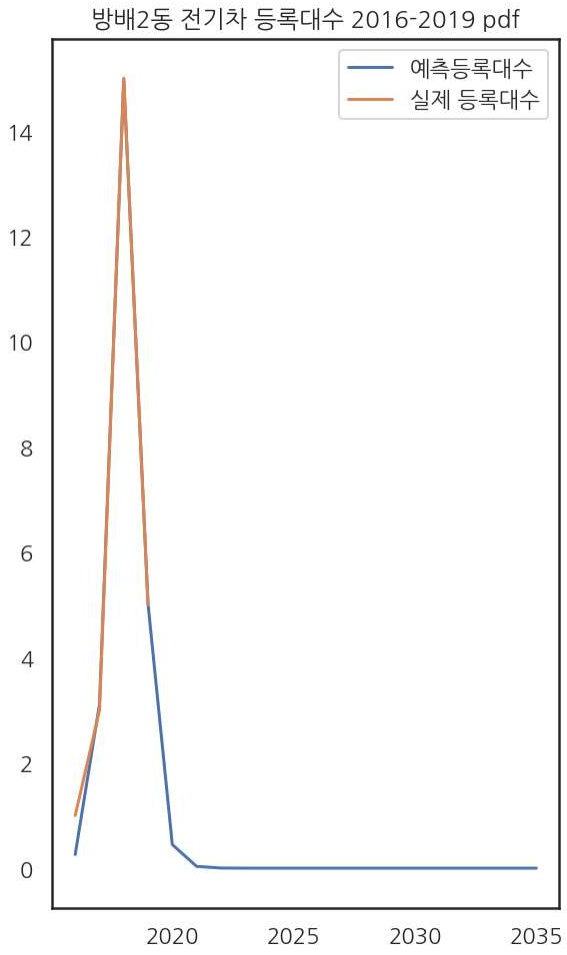
Out[13]:

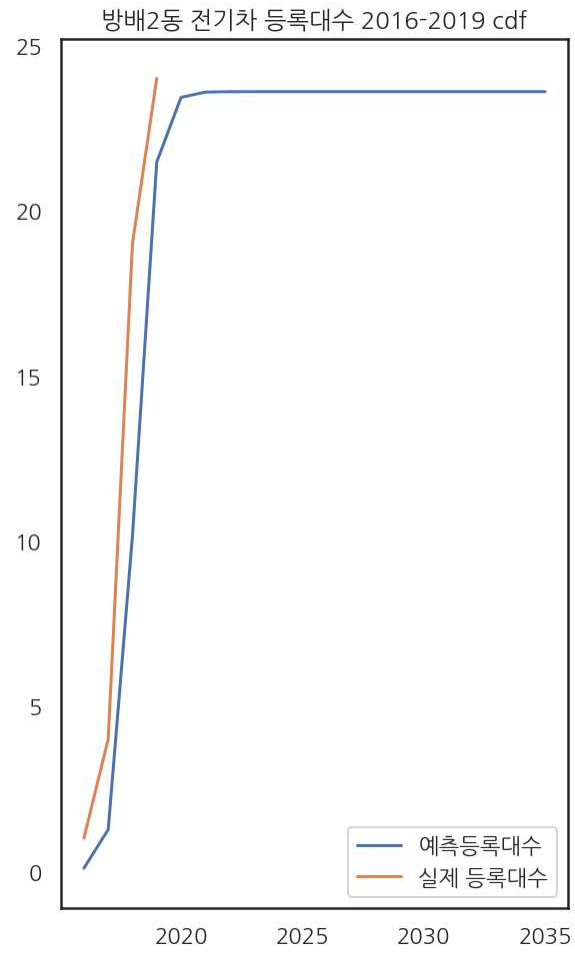
	2016	2017	2018	2019	2020	2021	2022	2023	
2016-2019	0.402999	46.452987	49.581779	49.583624	49.583625	49.583625	49.583625	49.583625	
2016-2020	0.428654	39.698661	43.075852	43.078991	43.078994	43.078994	43.078994	43.078994	
2016-2021	0.051878	0.254207	1.043299	4.120652	16.119788	62.874158	244.561707	943.279448	3

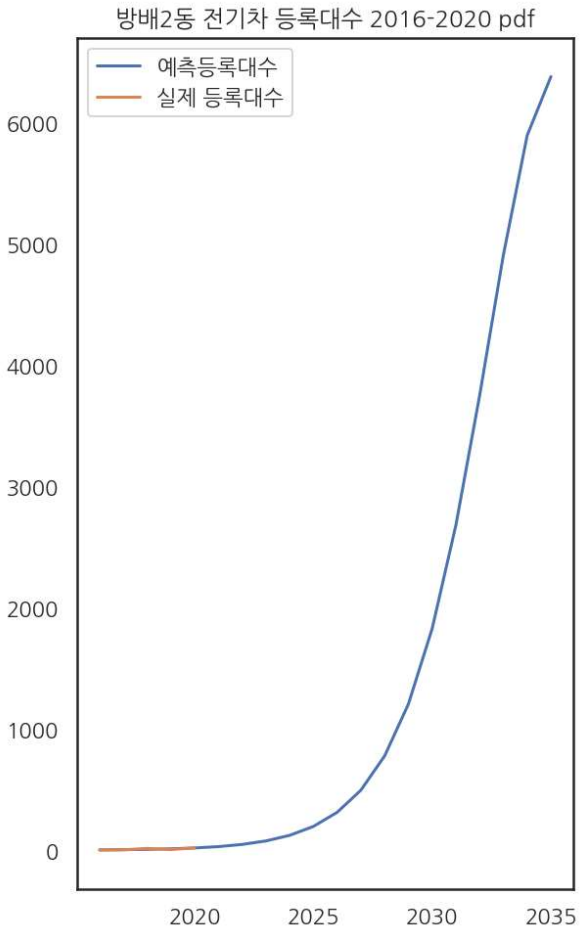
방배2동

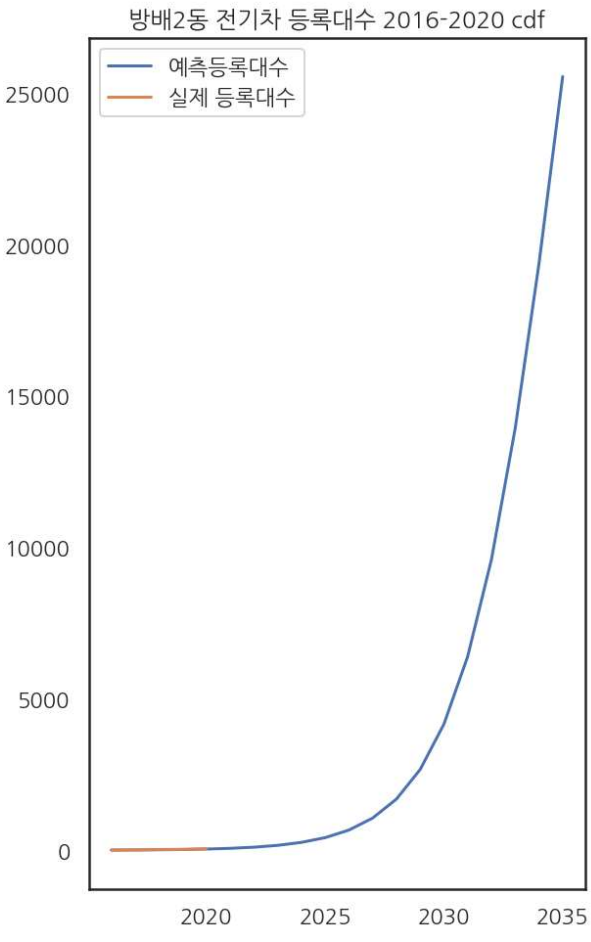
In [14]:
```
# initial variables(M, P & Q)
vars = [1, 0.08,0.58]

# sales vector
sales_2019=np.array([1, 3, 15, 5])
sales_2020=np.array([1, 3, 15, 5, 20])
sales_2021=np.array([1, 3, 15, 5, 20, 30])

# cumulatice sales
csales_2019=np.cumsum(sales_2019)
csales_2020=np.cumsum(sales_2020)
csales_2021=np.cumsum(sales_2021)

# non linear least square fitting
varfinal_2019,success_2019 = leastsq(residual, vars, args=(t_2019, sales_2019))
varfinal_2020,success_2020 = leastsq(residual, vars, args=(t_2020, sales_2020))
varfinal_2021,success_2021 = leastsq(residual, vars, args=(t_2021, sales_2021))

# estimated coefficients
```

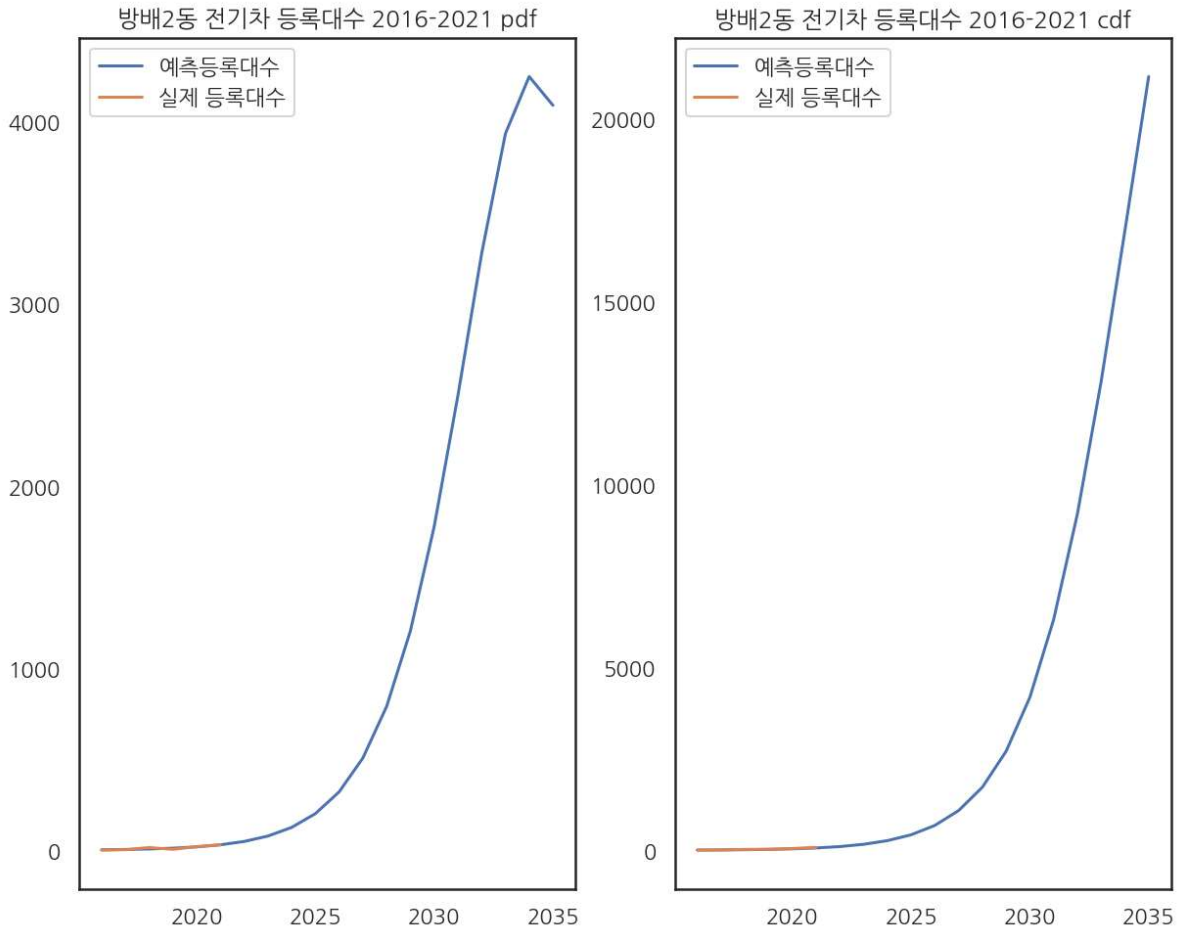
```
m1 = varfinal_2019[0]
p1 = varfinal_2019[1]
q1 = varfinal_2019[2]
m2 = varfinal_2020[0]
p2 = varfinal_2020[1]
q2 = varfinal_2020[2]
m3 = varfinal_2021[0]
p3 = varfinal_2021[1]
q3 = varfinal_2021[2]

#sales plot (pdf)
cofactor_2019= np.exp(-(p1+q1) * tp)
sales_pdf_2019= m1*(((p1+q1)**2/p1)*cofactor_2019)/(1+(q1/p1)*cofactor_2019)**2
cofactor_2020= np.exp(-(p2+q2) * tp)
sales_pdf_2020= m2*(((p2+q2)**2/p2)*cofactor_2020)/(1+(q2/p2)*cofactor_2020)**2
cofactor_2021= np.exp(-(p3+q3) * tp)
sales_pdf_2021= m3*(((p3+q3)**2/p3)*cofactor_2021)/(1+(q3/p3)*cofactor_2021)**2

# Cumulative sales (cdf)
sales_cdf_2019= m1*(1-cofactor_2019)/(1+(q1/p1)*cofactor_2019)
sales_cdf_2020= m2*(1-cofactor_2020)/(1+(q2/p2)*cofactor_2020)
sales_cdf_2021= m3*(1-cofactor_2021)/(1+(q3/p3)*cofactor_2021)

# 시각화
plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2019, date_2019, sales_2019)
plt.title('방배2동 전기차 등록대수 2016-2019 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2019, date_2019, csales_2019)
plt.title('방배2동 전기차 등록대수 2016-2019 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2020, date_2020, sales_2020)
plt.title('방배2동 전기차 등록대수 2016-2020 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2020, date_2020, csales_2020)
plt.title('방배2동 전기차 등록대수 2016-2020 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2021, date_2021, sales_2021)
plt.title('방배2동 전기차 등록대수 2016-2021 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2021, date_2021, csales_2021)
plt.title('방배2동 전기차 등록대수 2016-2021 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

#결과값
pd.DataFrame([sales_cdf_2019, sales_cdf_2020, sales_cdf_2021], columns=date_forecast
```


방배2동 전기차 등록대수 2016-2019 pdf
예측등록대수
실제 등록대수
0
2
4
6
8
10
12
14
2020
2025
2030
2035


방배2동 전기차 등록대수 2016-2019 cdf
예측등록대수
실제 등록대수
0
5
10
15
20
25
2020
2025
2030
2035


방배2동 전기차 등록대수 2016-2020 pdf
예측등록대수
실제 등록대수
0
1000
2000
3000
4000
5000
6000
2020
2025
2030
2035


방배2동 전기차 등록대수 2016-2020 cdf
예측등록대수
실제 등록대수
0
5000
10000
15000
20000
25000
2020
2025
2030
2035



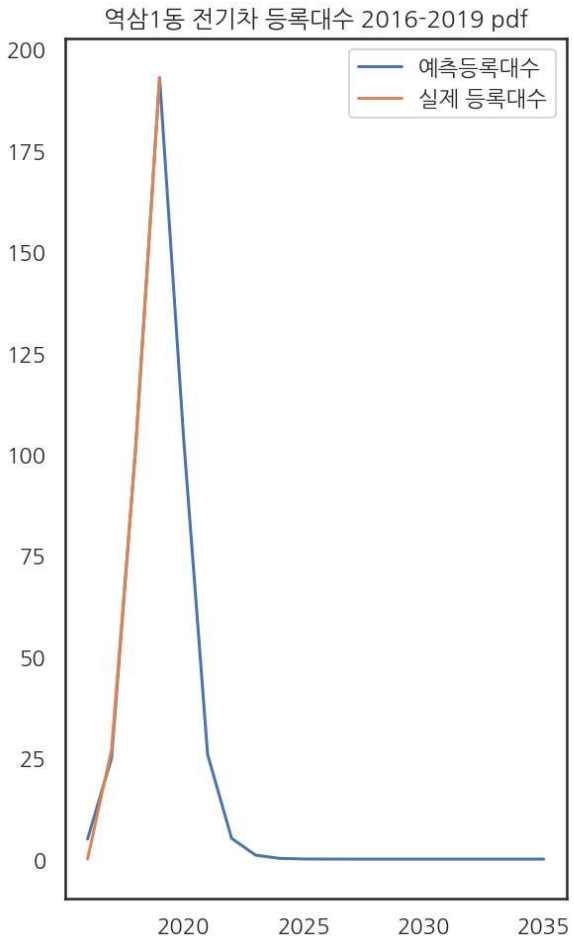
Out[14]:

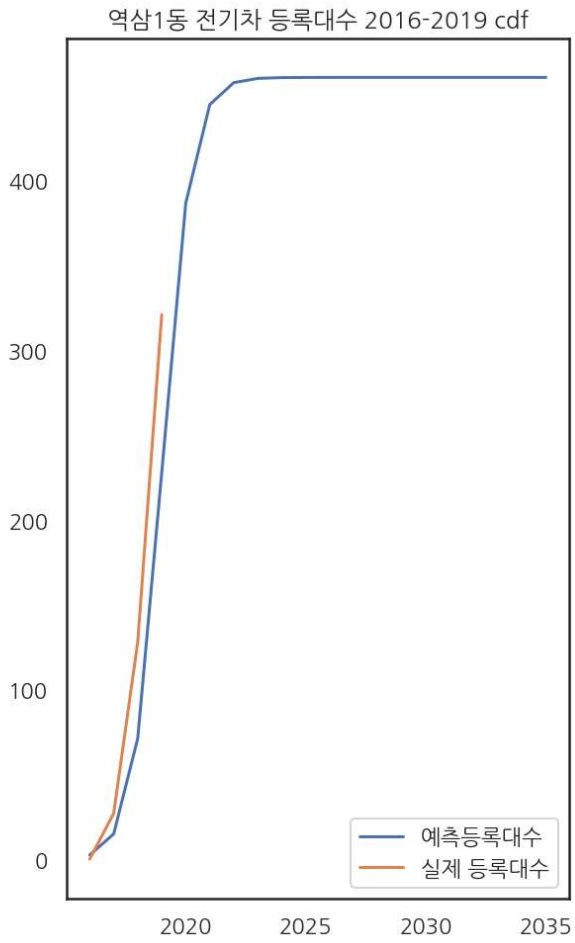
	2016	2017	2018	2019	2020	2021	2022	2023	
2016-2019	0.092719	1.261427	10.158945	21.464258	23.420632	23.582033	23.594243	23.595160	23
2016-2020	2.189857	5.713144	11.381435	20.499678	35.165197	58.746443	96.647214	157.520563	255
2016-2021	2.148547	5.624912	11.249139	20.346836	35.059395	58.842269	97.261476	159.257061	259

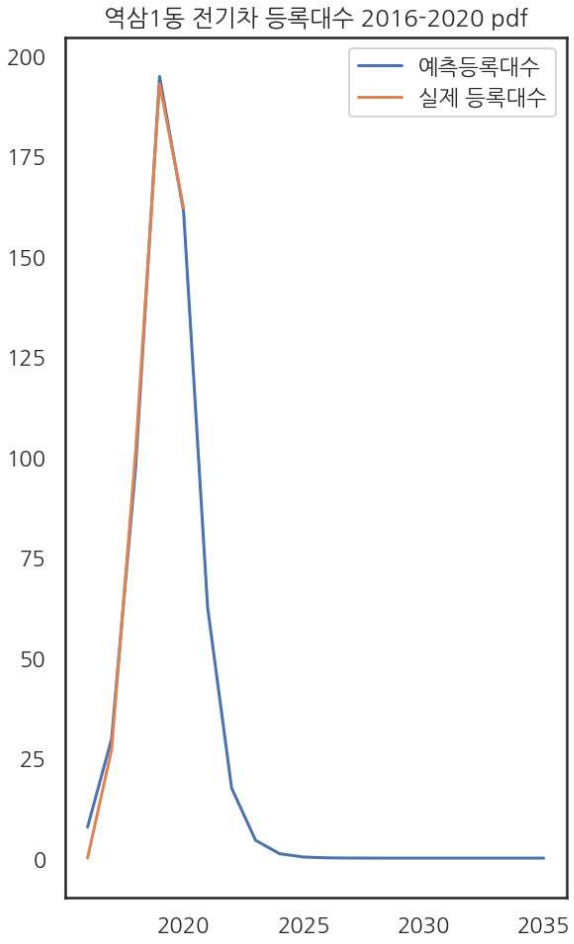
역삼1동

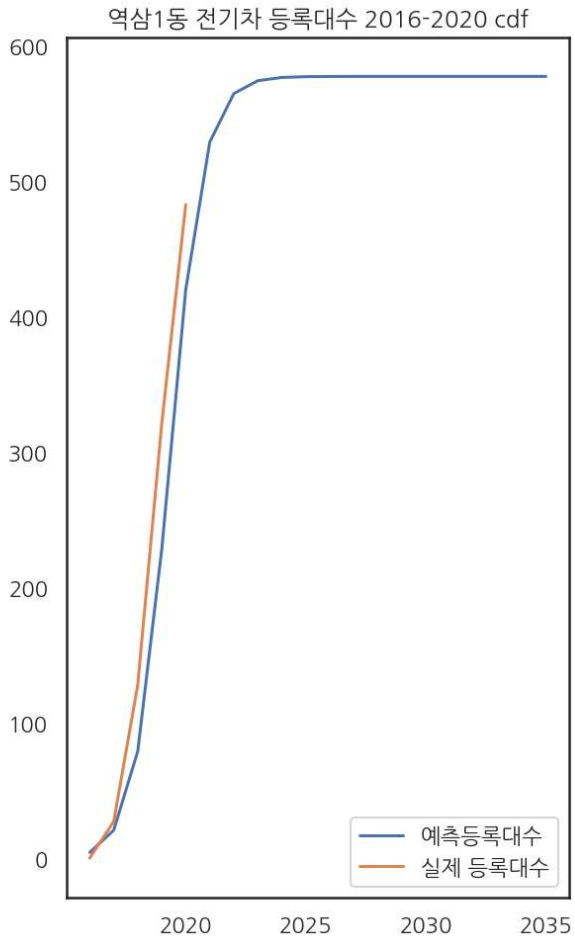
In [15]:
```
# initial variables(M, P & Q)
vars = [1, 0.08,0.58]

# sales vector
sales_2019=np.array([0, 27, 101, 193])
sales_2020=np.array([0, 27, 101, 193, 162])
sales_2021=np.array([0, 27, 101, 193, 162, 231])

# cumulatice sales
csales_2019=np.cumsum(sales_2019)
csales_2020=np.cumsum(sales_2020)
csales_2021=np.cumsum(sales_2021)

# non linear least square fitting
varfinal_2019,success_2019 = leastsq(residual, vars, args=(t_2019, sales_2019))
varfinal_2020,success_2020 = leastsq(residual, vars, args=(t_2020, sales_2020))
varfinal_2021,success_2021 = leastsq(residual, vars, args=(t_2021, sales_2021))

# estimated coefficients
```

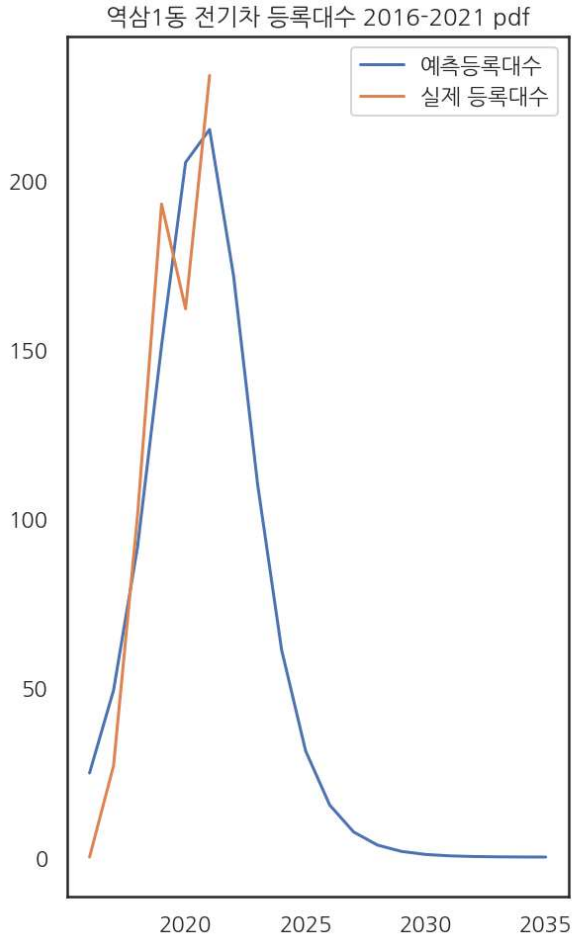
```
m1 = varfinal_2019[0]
p1 = varfinal_2019[1]
q1 = varfinal_2019[2]
m2 = varfinal_2020[0]
p2 = varfinal_2020[1]
q2 = varfinal_2020[2]
m3 = varfinal_2021[0]
p3 = varfinal_2021[1]
q3 = varfinal_2021[2]

#sales plot (pdf)
cofactor_2019= np.exp(-(p1+q1) * tp)
sales_pdf_2019= m1*(((p1+q1)**2/p1)*cofactor_2019)/(1+(q1/p1)*cofactor_2019)**2
cofactor_2020= np.exp(-(p2+q2) * tp)
sales_pdf_2020= m2*(((p2+q2)**2/p2)*cofactor_2020)/(1+(q2/p2)*cofactor_2020)**2
cofactor_2021= np.exp(-(p3+q3) * tp)
sales_pdf_2021= m3*(((p3+q3)**2/p3)*cofactor_2021)/(1+(q3/p3)*cofactor_2021)**2

# Cumulative sales (cdf)
sales_cdf_2019= m1*(1-cofactor_2019)/(1+(q1/p1)*cofactor_2019)
sales_cdf_2020= m2*(1-cofactor_2020)/(1+(q2/p2)*cofactor_2020)
sales_cdf_2021= m3*(1-cofactor_2021)/(1+(q3/p3)*cofactor_2021)

# 시각화
plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2019, date_2019, sales_2019)
plt.title('역삼1동 전기차 등록대수 2016-2019 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2019, date_2019, csales_2019)
plt.title('역삼1동 전기차 등록대수 2016-2019 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2020, date_2020, sales_2020)
plt.title('역삼1동 전기차 등록대수 2016-2020 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2020, date_2020, csales_2020)
plt.title('역삼1동 전기차 등록대수 2016-2020 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

plt.subplot(1,2,1)
plt.plot(date_forecast, sales_pdf_2021, date_2021, sales_2021)
plt.title('역삼1동 전기차 등록대수 2016-2021 pdf')
plt.legend(['예측등록대수', '실제 등록대수'])

plt.subplot(1,2,2)
plt.plot(date_forecast, sales_cdf_2021, date_2021, csales_2021)
plt.title('역삼1동 전기차 등록대수 2016-2021 cdf')
plt.legend(['예측등록대수', '실제 등록대수'])
plt.show()

#결과값
pd.DataFrame([sales_cdf_2019, sales_cdf_2020, sales_cdf_2021], columns=date_forecast
```

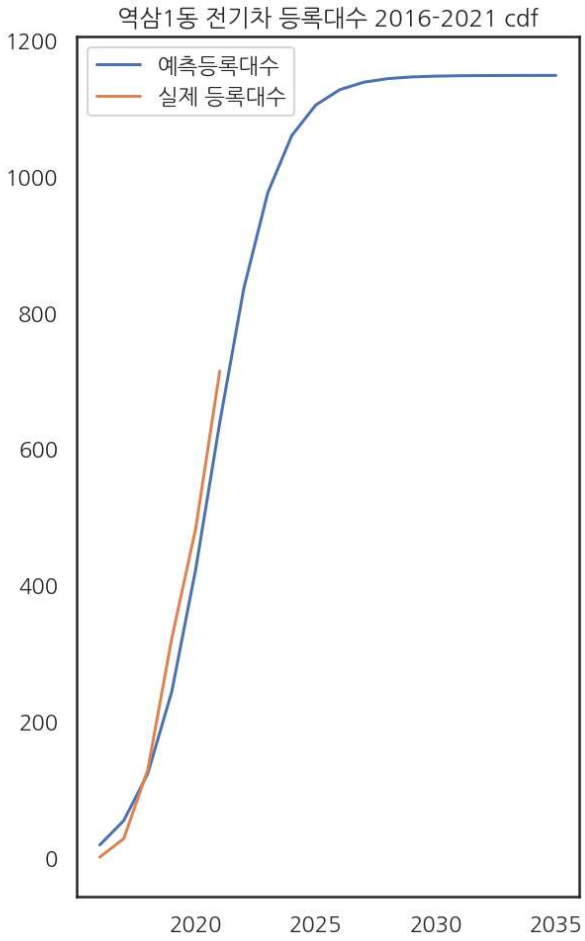

역삼1동 전기차 등록대수 2016-2019 pdf
예측등록대수
실제 등록대수
200
175
150
125
100
75
50
25
0
2020
2025
2030
2035


역삼1동 전기차 등록대수 2016-2019 cdf
예측등록대수
실제 등록대수
400
300
200
100
0
2020
2025
2030
2035


역삼1동 전기차 등록대수 2016-2020 pdf
예측등록대수
실제 등록대수
200
175
150
125
100
75
50
25
0
2020
2025
2030
2035


역삼1동 전기차 등록대수 2016-2020 cdf
예측등록대수
실제 등록대수
600
500
400
300
200
100
0
2020
2025
2030
2035





Out[15]:

	2016	2017	2018	2019	2020	2021	2022	20
2016-2019	2.412967	14.852454	71.178085	227.904866	386.746953	444.885277	457.801585	460.3102
2016-2020	4.174450	20.600001	78.890409	228.137311	420.321715	529.211548	565.058264	574.5951
2016-2021	17.752620	53.584668	122.424625	242.959822	423.308840	638.373635	835.086536	975.8938

In [ ]:

In [ ]: