• Pandas와 Matplotlib를 활용한 남북한발전전력량 데이터 분석

In [1]:
```
import pandas as pd

file_path = '남북한발전전력량.xlsx'

df = pd.read_excel(file_path, header=None)
# header=None을 사용하면 첫번째 행이 데이터로 간주되며 레이블이 출력되지 않습니다.
```

In [2]:
```
df
```

Out[2]:

	0	1	2	3	4	5	6	7	8	9	...	19	20	21	22	23	24	25	26	27	28
0	전력량 (억kWh)	발전 전력별	1990	1991	1992	1993	1994	1995	1996	1997	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
1	남한	합계	1077	1186	1310	1444	1650	1847	2055	2244	...	4031	4224	4336	4747	4969	5096	5171	5220	5281	5404
2	NaN	수력	64	51	49	60	41	55	52	54	...	50	56	56	65	78	77	84	78	58	66
3	NaN	화력	484	573	696	803	1022	1122	1264	1420	...	2551	2658	2802	3196	3343	3430	3581	3427	3402	3523
4	NaN	원자력	529	563	565	581	587	670	739	771	...	1429	1510	1478	1486	1547	1503	1388	1564	1648	1620
5	NaN	신재생	-	-	-	-	-	-	-	-	...	-	-	-	-	-	86	118	151	173	195
6	북한	합계	277	263	247	221	231	230	213	193	...	236	255	235	237	211	215	221	216	190	239
7	NaN	수력	156	150	142	133	138	142	125	107	...	133	141	125	134	132	135	139	130	100	128
8	NaN	화력	121	113	105	88	93	88	88	86	...	103	114	110	103	79	80	82	86	90	111
9	NaN	원자력	-	-	-	-	-	-	-	-	...	-	-	-	-	-	-	-	-	-	-

10 rows × 29 columns

In [3]:
```
df = pd.read_excel('남북한발전전력량.xlsx')
# header=None을 생략하면 첫번째 행은 레이블로 출력됩니다.
```

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

Out[4]:

	전력량 (억kWh)	발전 전력별	1990	1991	1992	1993	1994	1995	1996	1997	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
0	남한	합계	1077	1186	1310	1444	1650	1847	2055	2244	...	4031	4224	4336	4747	4969	5096	5171	5220	5281	5404
1	NaN	수력	64	51	49	60	41	55	52	54	...	50	56	56	65	78	77	84	78	58	66
2	NaN	화력	484	573	696	803	1022	1122	1264	1420	...	2551	2658	2802	3196	3343	3430	3581	3427	3402	3523
3	NaN	원자력	529	563	565	581	587	670	739	771	...	1429	1510	1478	1486	1547	1503	1388	1564	1648	1620
4	NaN	신재생	-	-	-	-	-	-	-	-	...	-	-	-	-	-	86	118	151	173	195
5	북한	합계	277	263	247	221	231	230	213	193	...	236	255	235	237	211	215	221	216	190	239
6	NaN	수력	156	150	142	133	138	142	125	107	...	133	141	125	134	132	135	139	130	100	128
7	NaN	화력	121	113	105	88	93	88	88	86	...	103	114	110	103	79	80	82	86	90	111
8	NaN	원자력	-	-	-	-	-	-	-	-	...	-	-	-	-	-	-	-	-	-	-

9 rows × 29 columns

In [5]:
```
1 df.columns  # 열(column) 레이블이 출력됩니다.
```

Out[5]:
```
Index(['전력량 (억kWh)', '발전 전력별', '1990', '1991', '1992', '1993', '1994', '1995',
       '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
       '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013',
       '2014', '2015', '2016'],
      dtype='object')
```

In [6]:
```
1 df.index   # 행(row) 레이블에 해당하는 첫번째 열 인덱스입니다. 0번 행부터 9번 행까지 1씩 증가학 있음을 의미
```

Out[6]:
```
RangeIndex(start=0, stop=9, step=1)
```

In [7]:
```
1 df.iloc[[0],]   # 남한 합계에 해당하는 0번 row 데이터를 출력합니다.
```

Out[7]:

	전력량 (억kWh)	발전 전력별	1990	1991	1992	1993	1994	1995	1996	1997	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
0	남한	합계	1077	1186	1310	1444	1650	1847	2055	2244	...	4031	4224	4336	4747	4969	5096	5171	5220	5281	5404

1 rows × 29 columns

In [8]:
```
df.iloc[[1],]  # 남한 수력에 해당하는 1번 row 데이터를 출력합니다.
```

Out[8]:

	전력량 (억kWh)	발전 전력별	1990	1991	1992	1993	1994	1995	1996	1997	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
1	NaN	수력	64	51	49	60	41	55	52	54	...	50	56	56	65	78	77	84	78	58	66

1 rows × 29 columns

In [9]:
```
df.iloc[[5],]   # 북한 합계에 해당하는 5번 row 데이터를 출력합니다.
```

Out[9]:

	전력량 (억kWh)	발전 전력별	1990	1991	1992	1993	1994	1995	1996	1997	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
5	북한	합계	277	263	247	221	231	230	213	193	...	236	255	235	237	211	215	221	216	190	239

1 rows × 29 columns

In [10]:
```
df_sn = df.iloc[[0, 5], 2:]

# 남한, 북한 발전량 합계에 해당하는 0번과 5번 row,
# 1990년 ~ 2016년 데이터에 해당하는 2번부터 마지막 column까지 추출합니다.
```

In [11]:
```
df_sn
```

Out[11]:

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
0	1077	1186	1310	1444	1650	1847	2055	2244	2153	2393	...	4031	4224	4336	4747	4969	5096	5171	5220	5281	5404
5	277	263	247	221	231	230	213	193	170	186	...	236	255	235	237	211	215	221	216	190	239

2 rows × 27 columns

In [12]:
```
df_sn.index
```

Out[12]:
```
Int64Index([0, 5], dtype='int64')
```

In [13]:
```
df_sn.index = ['South SUM','North SUM']
# 남한과 북한 합계에 해당하는 row 인덱스 0과 5를 South, North로 변경합니다.
```

In [14]:
```
1 df_sn
```

Out[14]:

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
South SUM	1077	1186	1310	1444	1650	1847	2055	2244	2153	2393	...	4031	4224	4336	4747	4969	5096	5171	5220	5281	5404
North SUM	277	263	247	221	231	230	213	193	170	186	...	236	255	235	237	211	215	221	216	190	239

2 rows × 27 columns

In [15]:
```
1 df_sn.index
```

Out[15]:
```
Index(['South SUM', 'North SUM'], dtype='object')
```

In [16]:
```
1 df_sn.columns
```

Out[16]:
```
Index(['1990', '1991', '1992', '1993', '1994', '1995', '1996', '1997', '1998',
       '1999', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007',
       '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016'],
      dtype='object')
```

In [17]:
```
1 type(df_sn)
```

Out[17]:
```
pandas.core.frame.DataFrame
```

In [18]:
```
1 type(df_sn.index)
```

Out[18]:
```
pandas.core.indexes.base.Index
```

In [19]:
```
1 type(df_sn.columns)
```

Out[19]:
```
pandas.core.indexes.base.Index
```

In [20]:
```
1 df_sn
```

Out[20]:

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
South SUM	1077	1186	1310	1444	1650	1847	2055	2244	2153	2393	...	4031	4224	4336	4747	4969	5096	5171	5220	5281	5404
North SUM	277	263	247	221	231	230	213	193	170	186	...	236	255	235	237	211	215	221	216	190	239

2 rows × 27 columns

In [21]:
```
1 df_sn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2 entries, South SUM to North SUM
Data columns (total 27 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   1990    2 non-null      object
 1   1991    2 non-null      object
 2   1992    2 non-null      object
 3   1993    2 non-null      object
 4   1994    2 non-null      object
 5   1995    2 non-null      object
 6   1996    2 non-null      object
 7   1997    2 non-null      object
 8   1998    2 non-null      object
 9   1999    2 non-null      object
 10  2000    2 non-null      object
 11  2001    2 non-null      object
 12  2002    2 non-null      object
 13  2003    2 non-null      object
 14  2004    2 non-null      object
 15  2005    2 non-null      object
 16  2006    2 non-null      object
 17  2007    2 non-null      object
 18  2008    2 non-null      object
 19  2009    2 non-null      object
 20  2010    2 non-null      object
 21  2011    2 non-null      object
 22  2012    2 non-null      object
 23  2013    2 non-null      object
 24  2014    2 non-null      object
 25  2015    2 non-null      object
 26  2016    2 non-null      object
dtypes: object(27)
memory usage: 448.0+ bytes
```

In [22]:
```
df_sn.dtypes
```

Out[22]:
```
1990    object
1991    object
1992    object
1993    object
1994    object
1995    object
1996    object
1997    object
1998    object
1999    object
2000    object
2001    object
2002    object
2003    object
2004    object
2005    object
2006    object
2007    object
2008    object
2009    object
2010    object
2011    object
2012    object
2013    object
2014    object
2015    object
2016    object
dtype: object
```

In [23]:
```
df_sn = df_sn.astype('int') #object 타입을 int 타입으로 변환
```

In [24]:
```
1 df_sn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2 entries, South SUM to North SUM
Data columns (total 27 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   1990    2 non-null      int32
 1   1991    2 non-null      int32
 2   1992    2 non-null      int32
 3   1993    2 non-null      int32
 4   1994    2 non-null      int32
 5   1995    2 non-null      int32
 6   1996    2 non-null      int32
 7   1997    2 non-null      int32
 8   1998    2 non-null      int32
 9   1999    2 non-null      int32
 10  2000    2 non-null      int32
 11  2001    2 non-null      int32
 12  2002    2 non-null      int32
 13  2003    2 non-null      int32
 14  2004    2 non-null      int32
 15  2005    2 non-null      int32
 16  2006    2 non-null      int32
 17  2007    2 non-null      int32
 18  2008    2 non-null      int32
 19  2009    2 non-null      int32
 20  2010    2 non-null      int32
 21  2011    2 non-null      int32
 22  2012    2 non-null      int32
 23  2013    2 non-null      int32
 24  2014    2 non-null      int32
 25  2015    2 non-null      int32
 26  2016    2 non-null      int32
dtypes: int32(27)
memory usage: 232.0+ bytes
```

'남북한발전전략량' 파일 분석 후 그래프 출력

pandas에서 제공하는 DataFrame을 대상으로 plot() & matplotlib.pyplot 활용

In [25]:
```
1 df_sn #DataFrame
```

Out[25]:

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	...	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
South SUM	1077	1186	1310	1444	1650	1847	2055	2244	2153	2393	...	4031	4224	4336	4747	4969	5096	5171	5220	5281	5404
North SUM	277	263	247	221	231	230	213	193	170	186	...	236	255	235	237	211	215	221	216	190	239

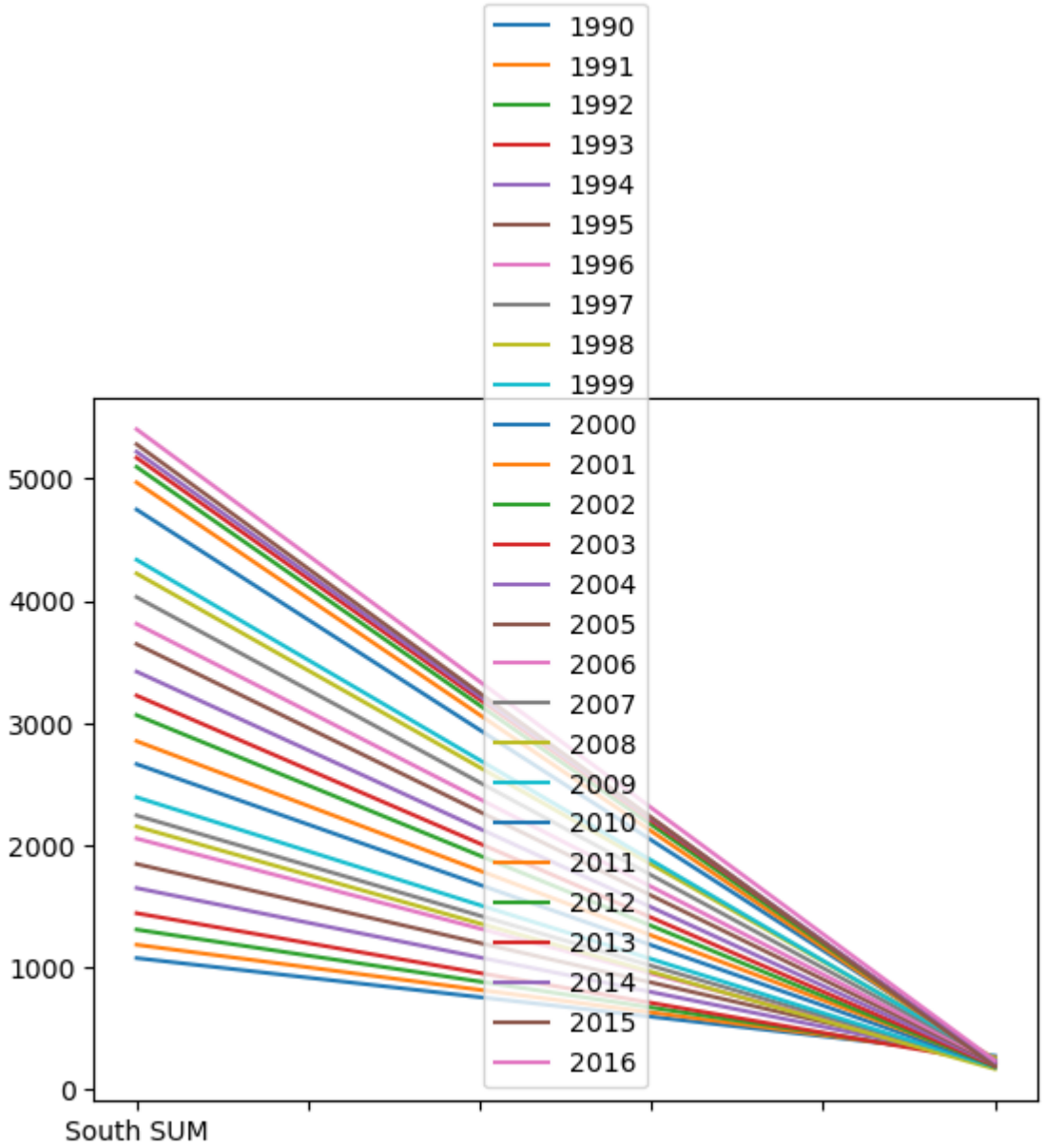
2 rows × 27 columns

In [26]:
```
1 type(df_sn)
```

Out[26]:
```
pandas.core.frame.DataFrame
```

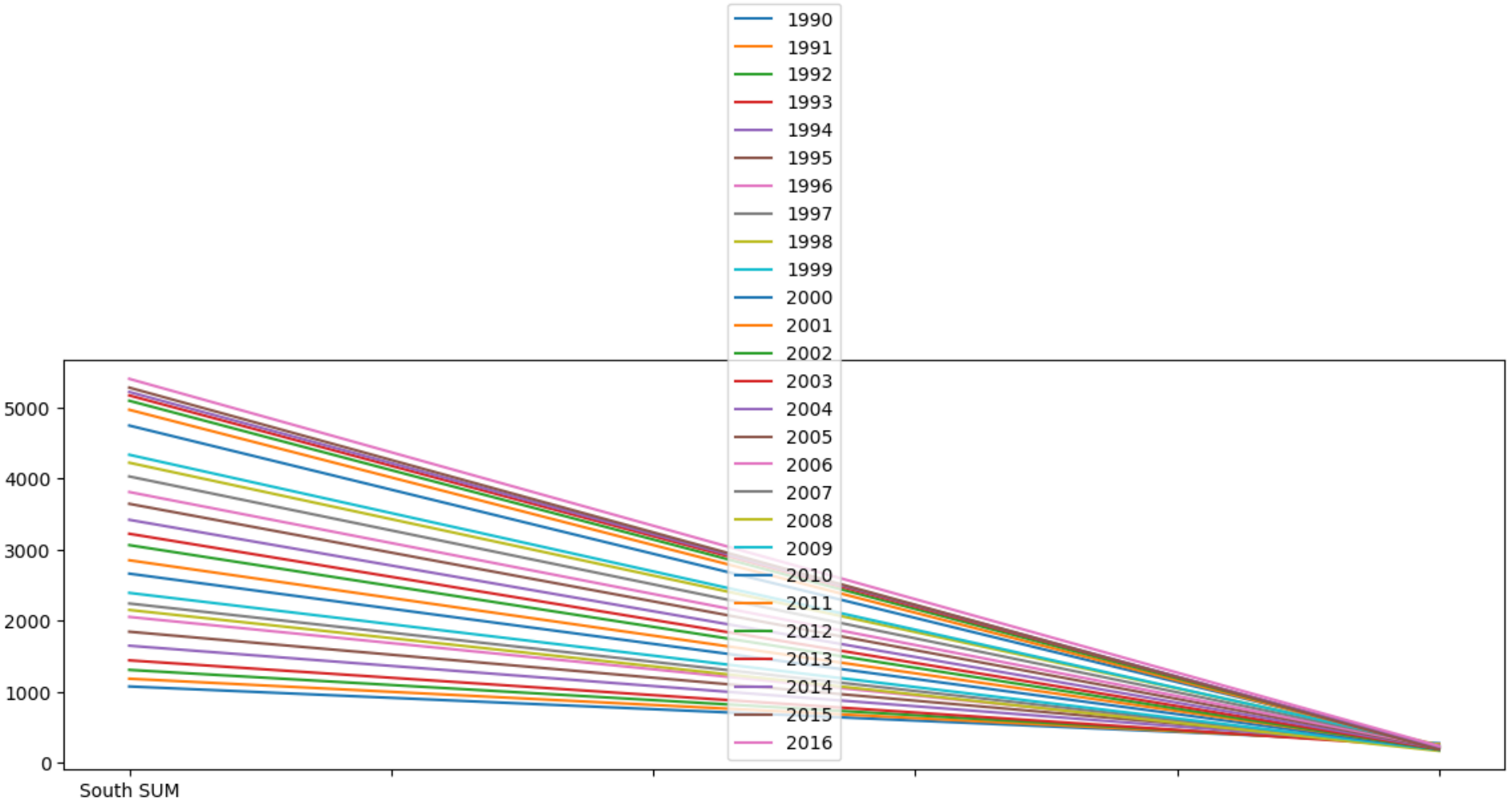
In [27]:
```
1 df_sn.plot()
```

Out[27]: <Axes: >



In [28]:
```
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = (14,4) #가로 너비 14인치, 세로 높이 4인치

#%matplotlib qt5

#pandas.core.frame.DataFrame 타입도 plot()를 이용한 그래프가 출력된다.
#각 열(column) 단위에 해당하는 년도별로 South SUM, North SUM 그래프가 출력
df_sn.plot()
```

Out[28]: <Axes: >



In [29]:

```
# 행, 열 전치하여 다시 그리기
# 출력된 그래프를 시간의 흐름에 따른 연도별 발전량 변화 추이를 보기 위해서는
# 연도 값을 x축에 표시하는 것이 좋습니다.
# 연도값을 x 축으로 바꾸기위해서 열레이블을 구성하고 있는 년도에 해당하는 값이
# 행레이블에 위치하도록 행렬을 전치합니다.

tdf_sn = df_sn.T  # T는 인덱스와 컬럼을 바꾼다.
tdf_sn
```

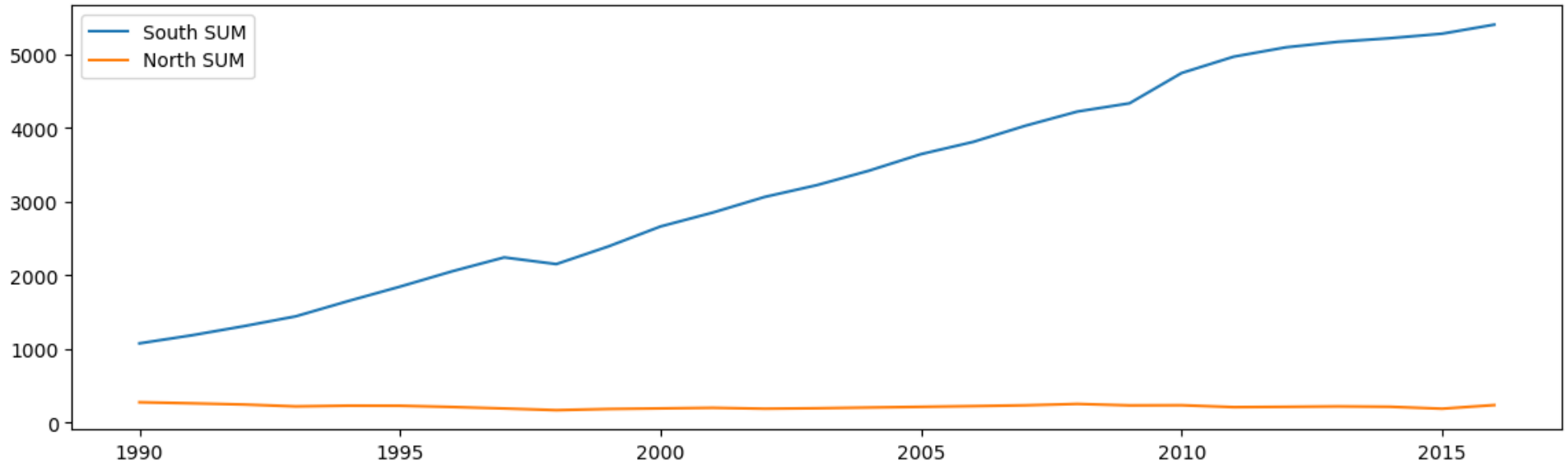
Out[29]:

	South SUM	North SUM
1990	1077	277
1991	1186	263
1992	1310	247
1993	1444	221
1994	1650	231
1995	1847	230
1996	2055	213
1997	2244	193
1998	2153	170
1999	2393	186
2000	2664	194
2001	2852	202
2002	3065	190
2003	3225	196
2004	3421	206
2005	3646	215
2006	3812	225
2007	4031	236
2008	4224	255
2009	4336	235
2010	4747	237
2011	4969	211
2012	5096	215
2013	5171	221
2014	5220	216
2015	5281	190
2016	5404	239

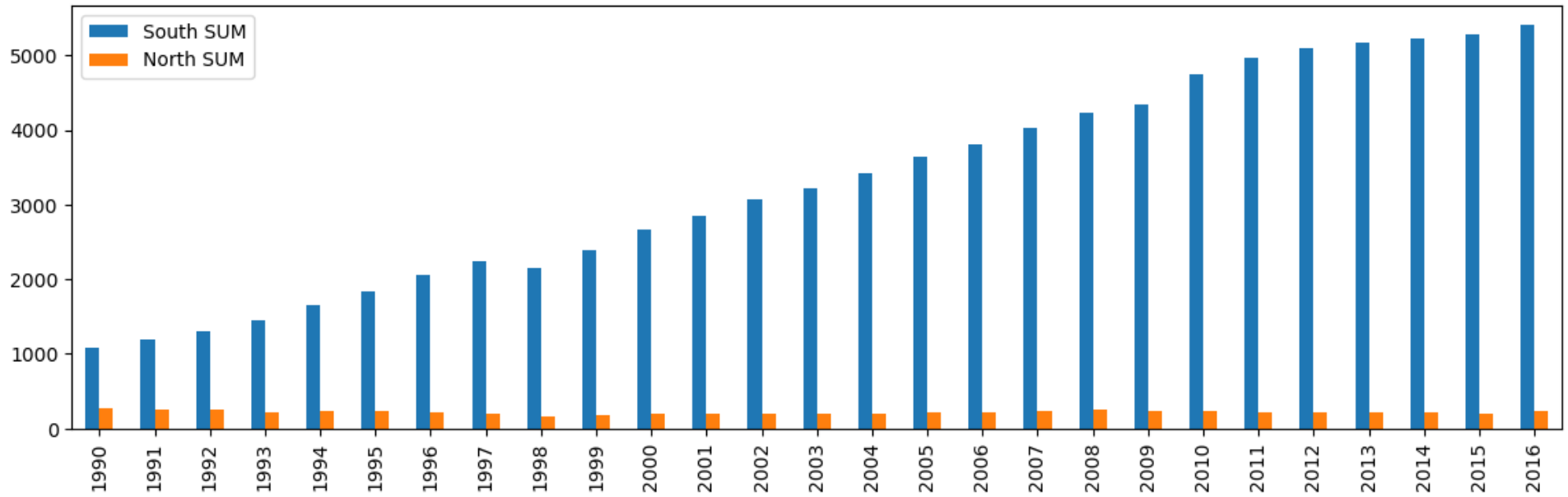
In [30]:
```
1 tdf_sn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 27 entries, 1990 to 2016
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   South SUM  27 non-null     int32
 1   North SUM  27 non-null     int32
dtypes: int32(2)
memory usage: 1.5+ KB
```

In [31]:
```
1 tdf_sn.plot()    # plot 그래프
2 plt.show()
```



In [32]:
```
1 tdf_sn.plot(kind='bar')   # 막대 그래프
2 plt.show()
```



In [33]:
```
1 tdf_sn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 27 entries, 1990 to 2016
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   South SUM  27 non-null     int32
 1   North SUM  27 non-null     int32
dtypes: int32(2)
memory usage: 1.5+ KB
```

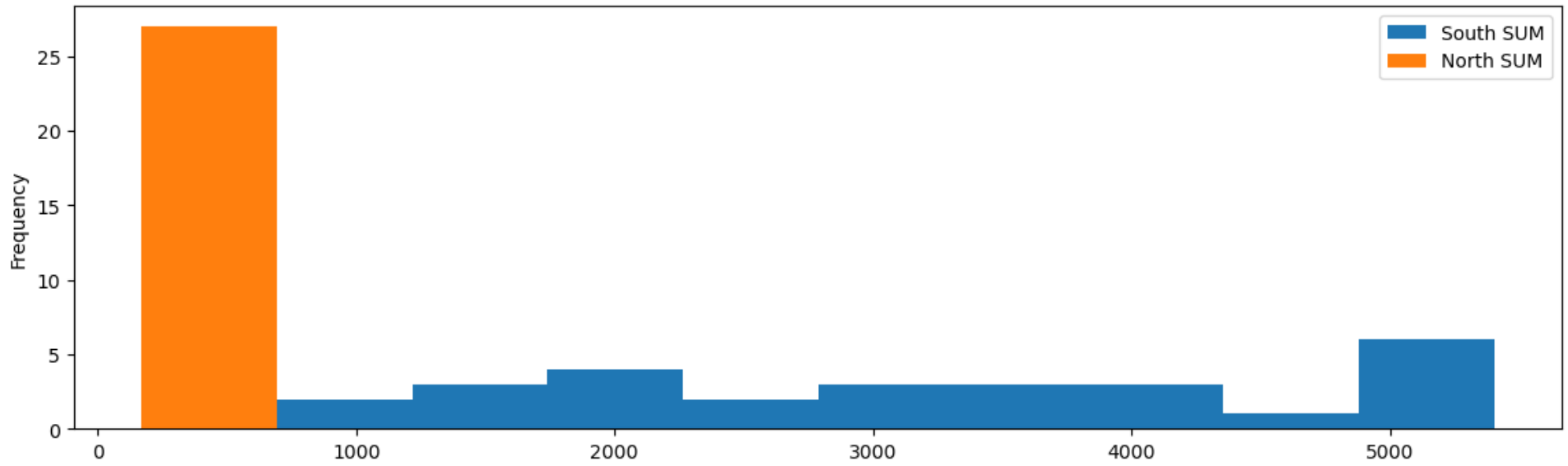
In [34]:
```
1 tdf_sn
```

Out[34]:

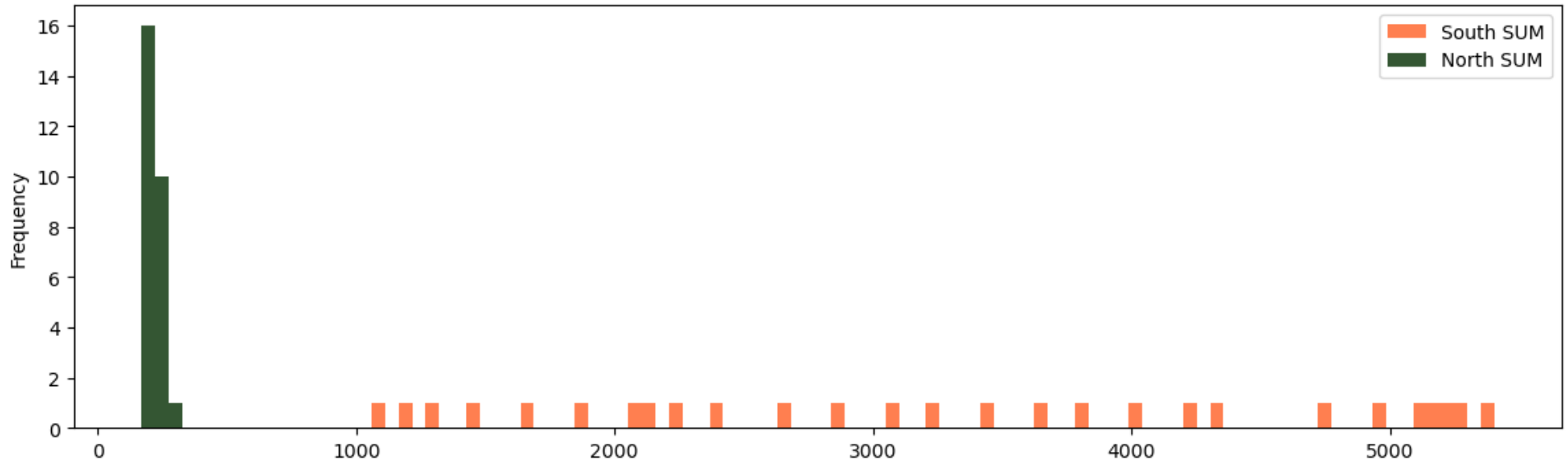
	South SUM	North SUM
1990	1077	277
1991	1186	263
1992	1310	247
1993	1444	221
1994	1650	231
1995	1847	230
1996	2055	213
1997	2244	193
1998	2153	170
1999	2393	186
2000	2664	194
2001	2852	202
2002	3065	190
2003	3225	196
2004	3421	206
2005	3646	215
2006	3812	225
2007	4031	236
2008	4224	255
2009	4336	235
2010	4747	237
2011	4969	211
2012	5096	215
2013	5171	221
2014	5220	216
2015	5281	190
2016	5404	239

In [35]:
```
1 tdf_sn.plot(kind='hist')
```

Out[35]: <Axes: ylabel='Frequency'>



In [36]:
```
1 tdf_sn.plot(kind='hist', bins=100, color=('coral','#345633'))   #히스토그램
2 plt.show()
3
4 # bins는 x축 구간의 개수를 의미하며 숫자가 커질수록 더 세밀하게 관찰할 수 있습니다.
5 # bins=1000 은 x축을 1000개로 쪼개서 분석했다는 뜻입니다.
```



matplotlib.font_manager 설정

In [37]:
```
import matplotlib.font_manager   # 한글 사용을 위해서 임포트합니다.

font_path = "C:/Windows/Fonts/HMFMMUEX.TTC"
font_name = matplotlib.font_manager.FontProperties(fname=font_path).get_name()
plt.rc('font', family=font_name)

# 위에서 import matplotlib.pyplot as plt 를 사용하지 않았다면 ...
# matplotlib.pyplot.rc('font', family=font_name) 으로 사용하면 됩니다.
```

In [38]:
```
tdf_sn
```

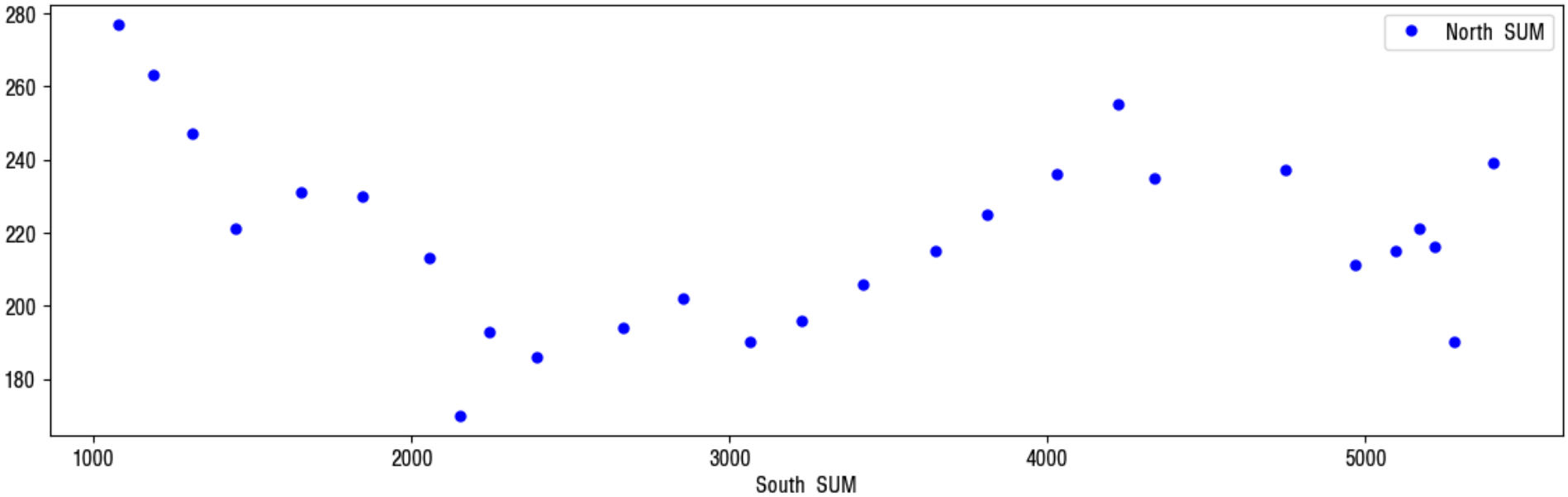
Out[38]:

	South SUM	North SUM
1990	1077	277
1991	1186	263
1992	1310	247
1993	1444	221
1994	1650	231
1995	1847	230
1996	2055	213
1997	2244	193
1998	2153	170
1999	2393	186
2000	2664	194

South SUM과 North SUM의 산점도(Scatter plot) 그래프

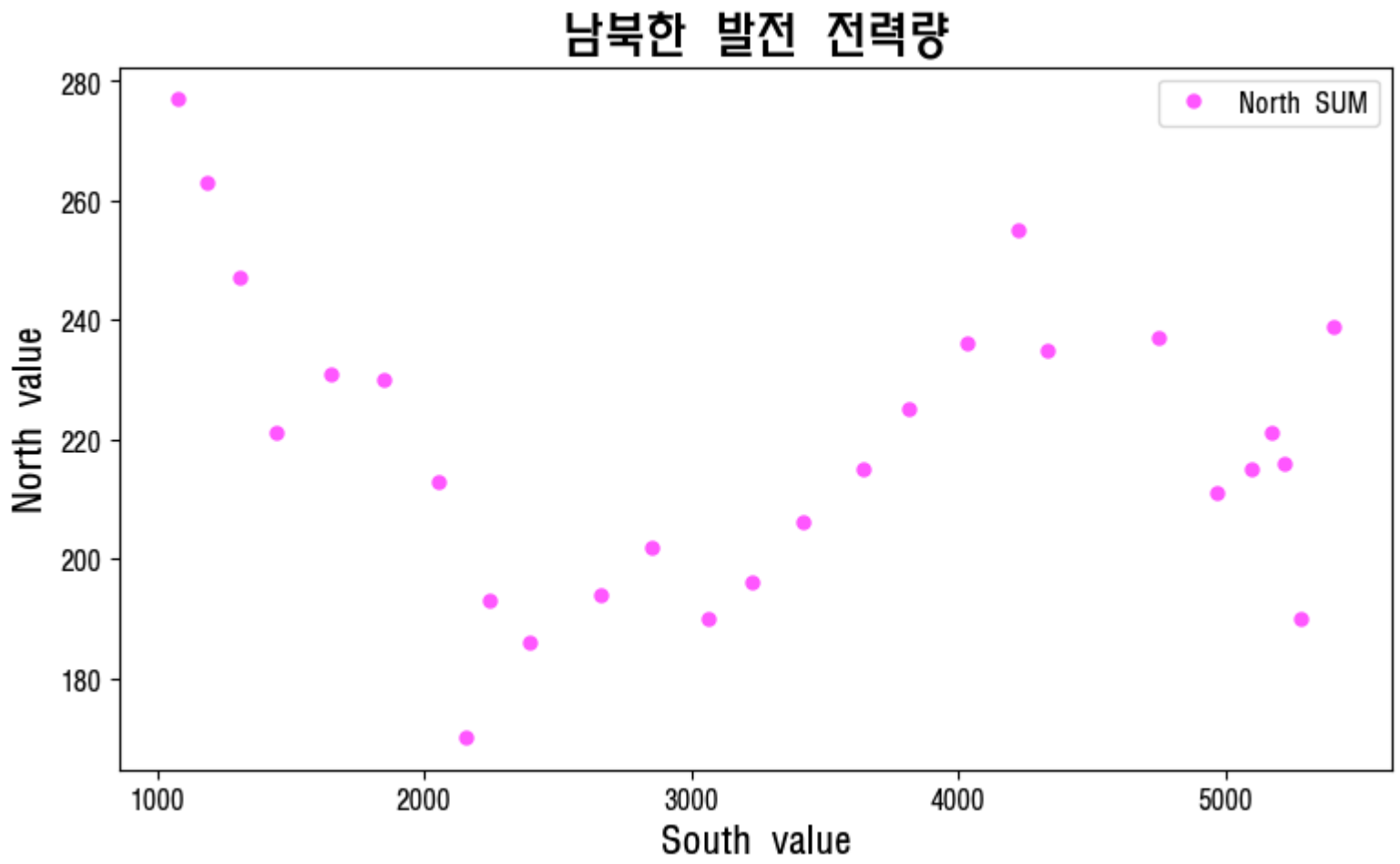
In [39]:

```
# pandas로 부터 그래프 출력
tdf_sn.plot('South SUM', 'North SUM',  # x축, y축을 의미함 -> 범례에 y축 레이블이 출력됨
         linestyle='none',     # linestyle='-', (실습 권장)
         marker='.',           # marker='x', (실습 권장)
         markersize=10,        # markersize=1, (실습 권장)
         color='blue',         # color='red', (실습 권장)
         alpha=1)              # alpha는 투명도를 의미함,  alpha=0.5 또는 alpha=0.1 (실습할 것)

plt.show()
```



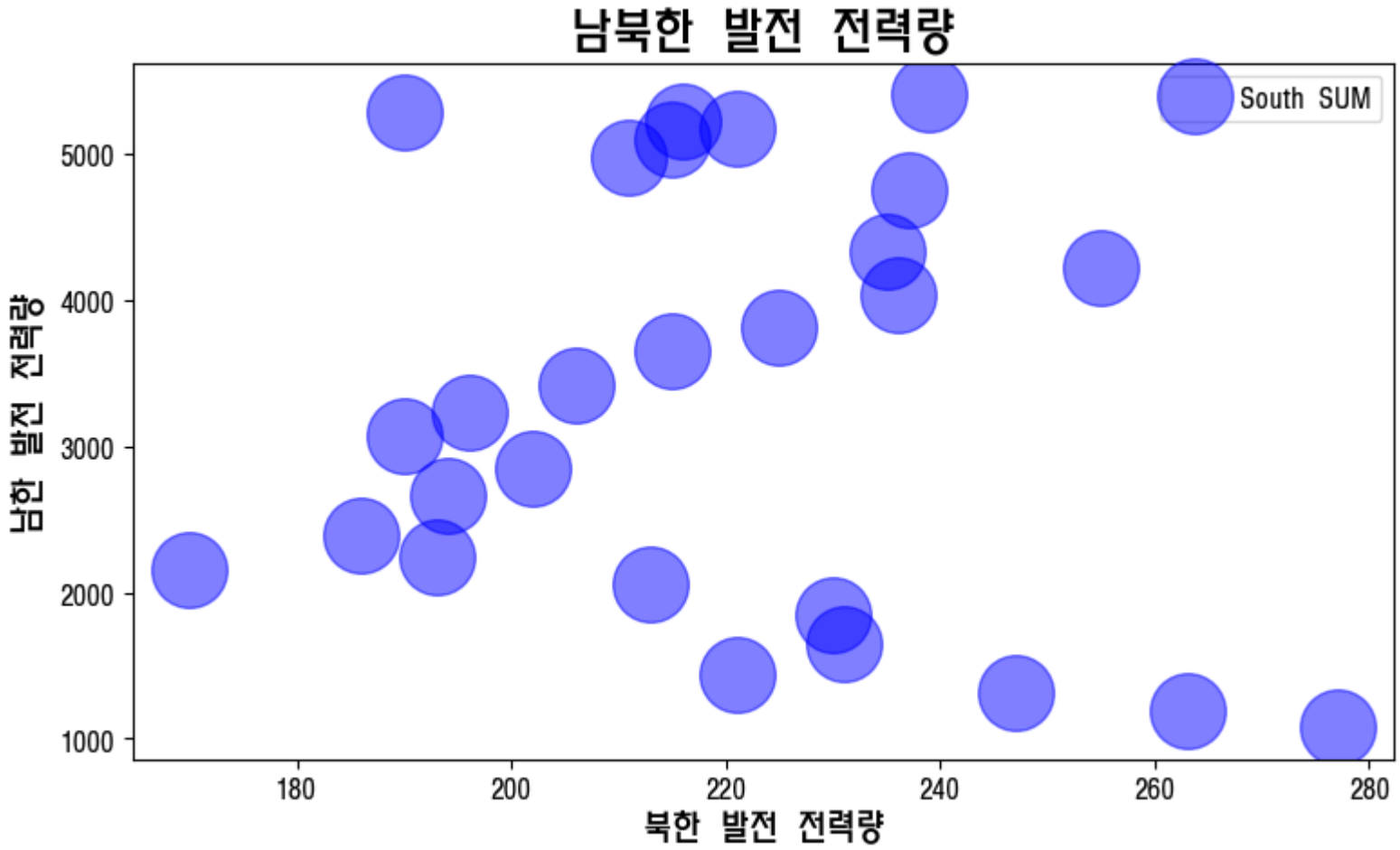
In [40]:

```
#pandas로 부터 그래프 출력

plt.rcParams["figure.figsize"] = (9,5) #가로 너비 9인치, 세로 높이 5인치

tdf_sn.plot('South SUM', 'North SUM',  # x축, y축을 의미함 -> 범례에 y축 레이블이 출력됨
        linestyle='none',        # linestyle='-', (실습 권장)
        marker='.',              # marker='x', (실습 권장)
        markersize=10,           # markersize=1, (실습 권장)
        color='#ff56ff',         # color='red', (실습 권장)
        alpha=1)                 # alpha는 투명도를 의미함,  alpha=0.5 또는 alpha=0.1 (실습할 것)

# pandas로 부터 출력된 그래프를 수정 -> import matplotlib.pyplot as plt 으로부터 plt를 사용
plt.title('남북한 발전 전력량', fontsize=20)
plt.xlabel('South value', fontsize=14)
plt.ylabel('North value', fontsize=14)
plt.show()
```



In [41]:

```
# pandas로 부터 그래프 출력
tdf_sn.plot('North SUM', 'South SUM',   # x축, y축을 의미함 -> 범례에 y축 레이블이 출력됨
        linestyle='none',
        marker='o',
        markersize=30,
        color='blue',
        alpha=0.5)

# pandas로 부터 출력된 그래프를 수정 -> import matplotlib.pyplot as plt 으로부터 plt를 사용
plt.title('남북한 발전 전력량', fontsize=20)
plt.xlabel('북한 발전 전력량',  fontsize=14)
plt.ylabel('남한 발전 전력량',  fontsize=14)
plt.show()
```



In [42]:
```
1 type(tdf_sn)
```

Out[42]: pandas.core.frame.DataFrame

In [43]:
```
1 type(plt)
```

Out[43]: module

In [ ]:
```
1
```