graph

May 19, 2024

[266]:

```
import os
import pandas as pd
import matplotlib.pyplot as plt

# List of district names in English
district_list = [
    "bilecik ort sıcaklık",
]

# Create folders for each district if they don't exist
for district in district_list:
    folder_name = district.lower().replace(" ", "_")
    if not os.path.exists(folder_name):
        os.makedirs(folder_name)

# Repeat the operations for each district
for district in district_list:
    # Create a filename specific to the district
    filename = f'{district}.csv'

    # Load the data
    df = pd.read_csv(filename, delimiter=';', parse_dates=[0], dayfirst=True)

    # Define the column names
    df.columns = ['Date', 'Temperature']

    # Ensure the Temperature column is numeric
    df['Temperature'] = pd.to_numeric(df['Temperature'].str.replace(',', '.'),␣
 ↪errors='coerce')

    # Add Year and Month columns
    df['Year'] = df['Date'].dt.year
    df['Month'] = df['Date'].dt.month
```

[267]:

```
# Calculate the total temperature for each year and month
monthly_totals = df.groupby(['Year', 'Month'])['Temperature'].sum().unstack()
print(monthly_totals)
```

```
Month         1          2          3          4          5          6          7  \
Year
1994   128.0505    45.4727   170.4419   386.7641   482.1101   536.8775   650.2115
1995    91.1666   119.6716   179.3557   268.3099   458.6182   607.8366   600.9979
1996     0.4085    91.2993    62.9744   234.2690   521.7894   533.3410   650.1499
1997    76.9062    -3.8195    62.0167   176.0251   489.4204   549.2706   626.5099
1998    57.3271    94.3850    87.4575   359.9962   441.2862   539.7306   657.5190
1999   108.0991    72.6457   192.0888   334.2359   480.6935   547.1426   674.6858
2000   -99.6891    32.5044   100.1366   356.2832   434.4697   532.5729   698.4926
2001   102.5028    79.7526   306.2236   328.4610   427.2100   580.8492   703.9738
2002   -91.5673   135.7581   222.5778   274.6243   444.3219   553.3657   705.8451
2003   148.2665   -34.4178    46.1615   237.4941   534.4040   604.8901   659.2739
2004    10.4014    26.0679   186.8927   310.7556   434.9249   555.8554   654.7440
2005    86.4435    77.1912   158.1688   304.7813   469.0797   524.1019   668.7975
2006   -29.9126     3.3386   191.5699   334.2289   461.9601   581.7002   647.8296
2007    70.7743    88.7575   192.4000   250.9857   553.6848   628.2757   736.5836
2008   -10.2706    28.4658   262.8924   364.8831   450.6556   611.3714   680.2486
2009    66.7686    94.2468   146.0233   302.3539   464.9201   626.2637   679.5598
2010    90.8528   172.1197   210.6975   311.4003   502.8806   563.7376   688.5377
2011    60.5018    68.4212   143.9160   239.8419   421.6868   539.5697   710.8683
2012   -47.3834   -51.3676   110.7656   372.7860   482.4411   632.0530   743.8796
2013    95.9974   158.6223   238.4024   345.2896   565.2031   602.9281   667.2671
2014   141.0173   156.2529   230.2098   370.4248   468.2433   563.6818   695.3163
2015    -2.3621    78.1104   191.8664   245.5218   499.1743   520.4372   680.0544
2016    46.5403   205.6293   237.7644   408.3972   453.8295   626.4332   698.2868
2017   -60.5487    87.5417   235.2956   312.1300   462.2581   584.6594   698.1486
2018    72.1374   165.9796   297.0562   432.3799   520.5122   599.8111   692.2599
2019    54.9482    89.0489   192.7555   300.4971   532.0724   608.8810   661.8122
2020    26.6500   106.9425   230.6237   295.4801   491.5365   572.0890   697.0659
2021    93.7239   108.3767   130.1207   308.6162   527.7544   537.0426   708.2079
2022   -18.9753    85.7253    41.5064   347.4604   489.1801   571.6245   646.4453
2023     2.2354        NaN        NaN        NaN        NaN        NaN        NaN

Month         8          9         10         11         12
Year
1994   665.7758   638.6726   484.4090   167.1177    38.7991
1995   624.8729   519.6891   326.0051    95.2102   118.7705
1996   633.3656   479.5075   337.4188   240.3702   183.7096
1997   559.4404   429.4001   361.1423   224.7996   101.8461
1998   691.8841   529.3282   435.6079   250.6804    97.2876
1999   678.6295   529.6745   417.3745   215.9078   168.1555
2000   639.1384   529.2213   369.5235   265.6983    94.0796
2001   676.3881   559.4254   396.6148   192.9676    17.2465
2002   642.7243   519.7432   408.3460   241.2018     1.6592
2003   692.3397   492.4249   432.2210   213.9778    59.3239
2004   645.3224   552.3605   456.4300   211.2191   117.6296
2005   694.9774   533.3822   340.7695   200.9564   120.3476
2006   767.5101   531.6644   423.9261   175.9133    61.9798
```

```
2007   730.2030  563.3299  442.2117  197.1551   68.7997
2008   728.2381  538.3213  393.9209  277.3368  107.2448
2009   656.5516  518.2496  492.2828  233.4709  178.5632
2010   789.9900  578.3921  361.5855  346.6571  180.9400
2011   647.1809  584.2723  331.3565  107.0494   94.3193
2012   689.1521  612.4468  509.2946  305.4828  133.5830
2013   707.6486  544.6098  351.1147  269.1201   14.4424
2014   718.5578  535.5930  405.2249  227.6841  173.3171
2015   712.8767  648.7786  425.1481  284.9992   58.2836
2016   708.8008  546.5455  412.6025  204.8283  -17.4404
2017   673.8185  615.1197  370.9278  220.3743  143.2393
2018   712.7176  566.2849  440.0009  272.1263   75.2745
2019   682.7382  564.1016  494.4409  330.8378  134.3447
2020   714.8888  642.7517  530.9062  221.4252  203.6346
2021   726.2829  507.0919  387.9455  286.6350  143.2613
2022   695.6671  546.0261  401.5139  291.4686  183.7961
2023        NaN       NaN       NaN       NaN       NaN
```

[268]:

```python
# Calculate the total temperature for each year
yearly_totals = df.groupby('Year')['Temperature'].sum()
print(yearly_totals)
```

```
Year
1994    4394.7025
1995    4010.5043
1996    3968.6032
1997    3652.9579
1998    4242.4898
1999    4419.3332
2000    3952.4314
2001    4371.6154
2002    4058.6001
2003    4086.3596
2004    4162.6035
2005    4178.9970
2006    4151.7084
2007    4523.1610
2008    4433.3082
2009    4459.2543
2010    4797.7909
2011    3948.9841
2012    4493.1336
2013    4560.6456
2014    4685.5231
2015    4342.8886
2016    4532.2174
2017    4342.9643
2018    4846.5405
```

```
2019    4646.4785
2020    4733.9942
2021    4465.0590
2022    4281.4385
2023       2.2354
Name: Temperature, dtype: float64
```

[269]:

```python
# Convert the horizontal table to a string and add borders
table_horizontal = monthly_totals.style.set_table_styles(
    [{'selector': 'th, td', 'props': [('border', '1px solid black')]}]
)
```

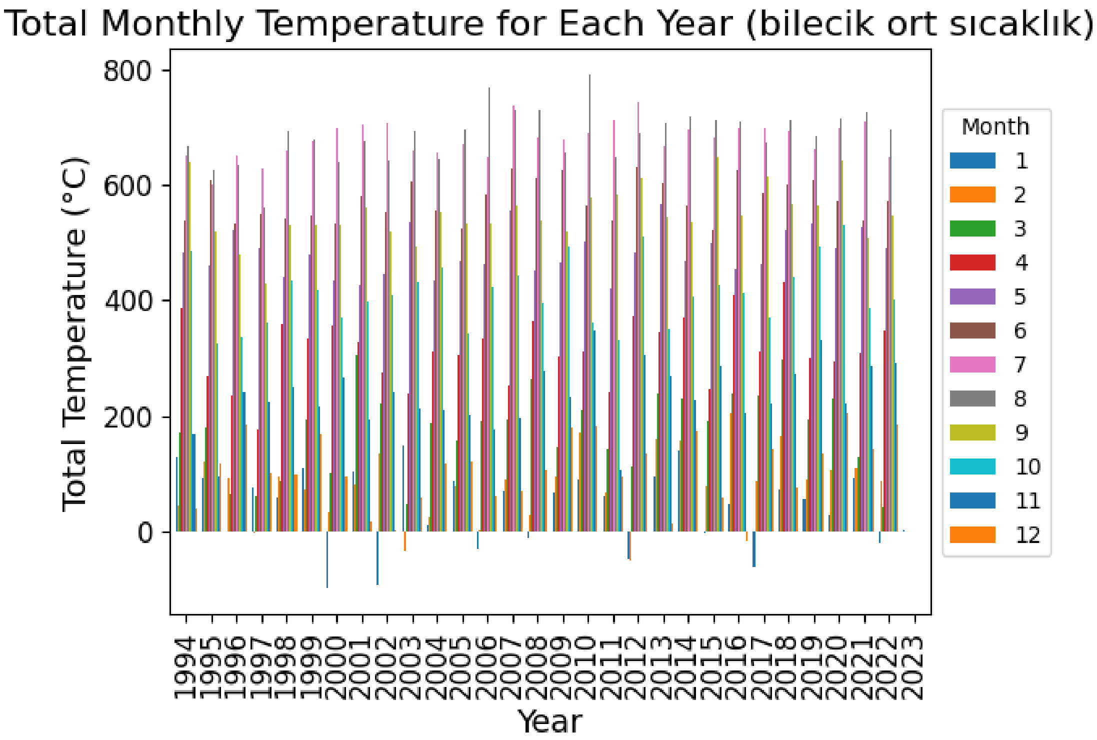
[270]:

```python
# Convert the vertical table to a string and add borders
table_vertical = yearly_totals.to_frame().style.set_table_styles(
    [{'selector': 'th, td', 'props': [('border', '1px solid black')]}]
)
```

[271]:

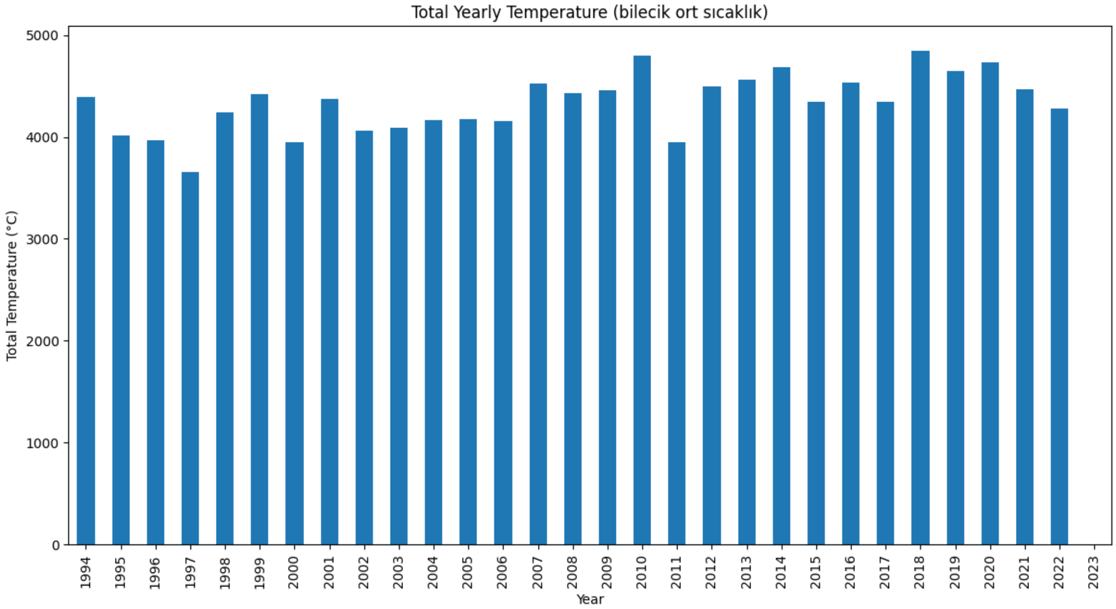
```python
# Create a bar chart (total temperature for each month of each year)
plt.figure(figsize=(16, 10))
monthly_totals.plot(kind='bar', width=0.8)
plt.title(f'Total Monthly Temperature for Each Year ({district})', fontsize=16)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Total Temperature (°C)', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.legend(title='Month', fontsize=10, loc='center left', bbox_to_anchor=(1, 0.
 ↪5))
plt.tight_layout()
plt.savefig(f'{folder_name}/{folder_name}_monthly_totals.png')
plt.show()
plt.close()
```

```
<Figure size 1600x1000 with 0 Axes>
```



[272]:

```
# Create a bar chart (total temperature for each year)
plt.figure(figsize=(14, 7))
yearly_totals.plot(kind='bar')
plt.title(f'Total Yearly Temperature ({district})')
plt.xlabel('Year')
plt.ylabel('Total Temperature (°C)')
plt.savefig(f'{folder_name}/{folder_name}_yearly_totals.png')
plt.show()
plt.close()
```



[273]:

```
# Save the monthly totals to a CSV file
monthly_totals.to_csv(f'{folder_name}/{folder_name}_monthly_totals.csv')

# Save the yearly totals to a CSV file
yearly_totals.to_csv(f'{folder_name}/{folder_name}_yearly_totals.csv')
```