Analysis Q4(d) Plotting

August 28, 2019

1 DATA420-19S1 Assignment 1 Peng Shen(57408055)

2 Analysis Q4(d)

Plot the time series of TMIN and TMAX on the same axis for each station in New Zealand

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import matplotlib.dates as mdates
        import os
        from collections import OrderedDict
```

```
In [2]: # Load the parquet file as pandas dataframe
        df = pd.read_parquet('./daily_temp_NZ.parquet', engine='pyarrow')
```

```
In [3]: df.head()
```

Out[3]:

	ID	DATE	ELEMENT	VALUE	MEASUREMENT_FLAG	QUALITY_FLAG	SOURCE_FLAG	OBSERVATION_TIME	LATITUDE	LONGITUDE	...	HCN/CRN_FLAG	WMO_ID	COUNTRY_CODE	COUNTRY_NAME	STATE_NAME	CORE_ELEMENT_COUNT	OTHER_ELEMENT_COUNT
0	NZ000936150	20100101	TMAX	324	None	None	S	None	-42.717	170.983	...		93781	NZ	New Zealand	None	3	1
1	NZM00093678	20100101	TMAX	242	None	None	S	None	-42.417	173.700	...		93678	NZ	New Zealand	None	3	1
2	NZM00093678	20100101	TMIN	94	None	None	S	None	-42.417	173.700	...		93678	NZ	New Zealand	None	3	1
3	NZ000933090	20100101	TMAX	197	None	None	S	None	-39.017	174.183	...		93309	NZ	New Zealand	None	3	1
4	NZ000933090	20100101	TMIN	82	None	None	S	None	-39.017	174.183	...		93309	NZ	New Zealand	None	3	1

```
4           NZ  New Zealand       None                  3                   1

  FIRSTYEAR_ACTIVE LASTYEAR_ACTIVE  YEAR
0             1964            2017  2010
1             1997            2017  2010
2             1997            2017  2010
3             1944            2017  2010
4             1944            2017  2010

[5 rows x 24 columns]
```

```
In [4]: df.DATE = pd.to_datetime(df.DATE, format='%Y-%m-%d')
        df.VALUE = df.VALUE/10
```

```
In [5]: df.head()
```

```
Out[5]:            ID       DATE ELEMENT  VALUE MEASUREMENT_FLAG QUALITY_FLAG  \
0  NZ000936150 2010-01-01    TMAX   32.4             None         None
1  NZM00093678 2010-01-01    TMAX   24.2             None         None
2  NZM00093678 2010-01-01    TMIN    9.4             None         None
3  NZ000933090 2010-01-01    TMAX   19.7             None         None
4  NZ000933090 2010-01-01    TMIN    8.2             None         None

  SOURCE_FLAG OBSERVATION_TIME  LATITUDE  LONGITUDE  ...  HCN/CRN_FLAG WMO_ID  \
0           S             None   -42.717    170.983  ...                93781
1           S             None   -42.417    173.700  ...                93678
2           S             None   -42.417    173.700  ...                93678
3           S             None   -39.017    174.183  ...                93309
4           S             None   -39.017    174.183  ...                93309

  COUNTRY_CODE COUNTRY_NAME STATE_NAME CORE_ELEMENT_COUNT OTHER_ELEMENT_COUNT  \
0           NZ  New Zealand       None                  3                   1
1           NZ  New Zealand       None                  3                   1
2           NZ  New Zealand       None                  3                   1
3           NZ  New Zealand       None                  3                   1
4           NZ  New Zealand       None                  3                   1

  FIRSTYEAR_ACTIVE LASTYEAR_ACTIVE  YEAR
0             1964            2017  2010
1             1997            2017  2010
2             1997            2017  2010
3             1944            2017  2010
4             1944            2017  2010

[5 rows x 24 columns]
```

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 447017 entries, 0 to 447016
```

```
Data columns (total 24 columns):
ID                    447017 non-null object
DATE                  447017 non-null datetime64[ns]
ELEMENT               447017 non-null object
VALUE                 447017 non-null float64
MEASUREMENT_FLAG      0 non-null object
QUALITY_FLAG          34 non-null object
SOURCE_FLAG           447017 non-null object
OBSERVATION_TIME      0 non-null object
LATITUDE              447017 non-null float64
LONGITUDE             447017 non-null float64
ELEVATION             447017 non-null float64
STATE                 447017 non-null object
STATION_NAME          447017 non-null object
GSN_FLAG              447017 non-null object
HCN/CRN_FLAG          447017 non-null object
WMO_ID                447017 non-null object
COUNTRY_CODE          447017 non-null object
COUNTRY_NAME          447017 non-null object
STATE_NAME            0 non-null object
CORE_ELEMENT_COUNT    447017 non-null int64
OTHER_ELEMENT_COUNT   447017 non-null int64
FIRSTYEAR_ACTIVE      447017 non-null int32
LASTYEAR_ACTIVE       447017 non-null int32
YEAR                  447017 non-null object
dtypes: datetime64[ns](1), float64(4), int32(2), int64(2), object(15)
memory usage: 78.4+ MB
```

```
In [7]: df.VALUE.describe()
```

```
Out[7]: count    447017.000000
        mean         12.489415
        std           6.751964
        min         -19.500000
        25%           7.700000
        50%          12.600000
        75%          17.400000
        max          38.100000
        Name: VALUE, dtype: float64
```

```
In [8]: df.DATE.describe()
```

```
Out[8]: count                  447017
        unique                  28292
        top       2003-05-16 00:00:00
        freq                       28
        first     1940-03-08 00:00:00
```

```
last      2017-09-08 00:00:00
Name: DATE, dtype: object
```

In [9]:
```
# Create a set of station IDs
stations = set(df.ID.tolist())
stations
```

Out[9]:
```
{'NZ000093012',
 'NZ000093292',
 'NZ000093417',
 'NZ000093844',
 'NZ000093994',
 'NZ000933090',
 'NZ000936150',
 'NZ000937470',
 'NZ000939450',
 'NZ000939870',
 'NZM00093110',
 'NZM00093439',
 'NZM00093678',
 'NZM00093781',
 'NZM00093929'}
```

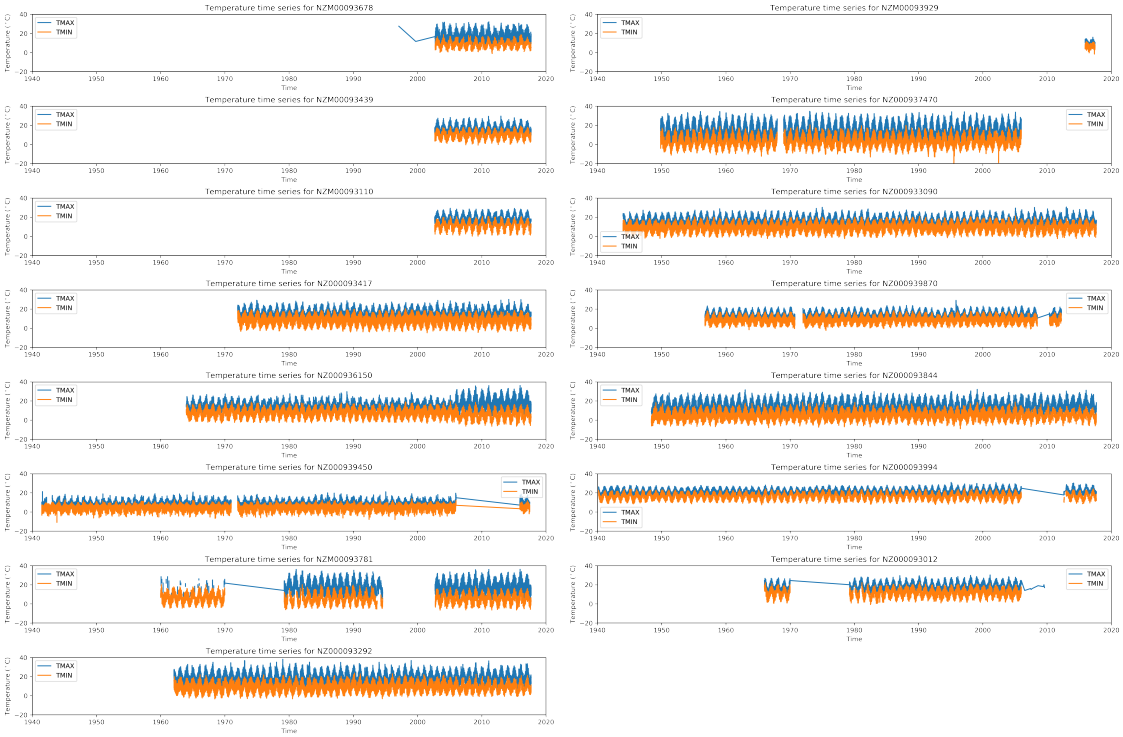
In [10]:
```
# Plot the time series of TMIN and TMAX on the same axis for each station in New Zeal
f, axes = plt.subplots(8, 2, dpi=300, figsize=(30, 20))  # affects output resolution
f.subplots_adjust(hspace=0.6,wspace=0.1)
axes = axes.flatten()  # squeeze grid of (8, 2) axes into a linear array for iterating

for i, station in enumerate(stations):  # generate one plot for each tag
    print(station)

    # Load
    data = df[df.ID==station][["DATE", "ELEMENT", "VALUE"]]
    data = data.pivot_table(index='DATE',columns='ELEMENT',values='VALUE')

    # Plots
    a = axes[i]
    a.plot(data)  # assign label to include in legend
    a.set_ylim([-20, 40])  # exapnd axes slightly beyond [0, 10]
    a.set_xlim("1940", "2020")

    # Legend
    a.legend(data.columns,loc="best")

    # Labels
    a.set_title(f"Temperature time series for {station}")
    a.set_xlabel("Time")
    a.set_ylabel("Temperature ($^\circ$C)")
```

```
        # Remove empty subplots
        for a in axes[(i + 1):]:
            f.delaxes(a)
```

```
NZM00093678
NZM00093929
NZM00093439
NZ000937470
NZM00093110
```

```
/Users/dylan/anaconda3/lib/python3.7/site-packages/pandas/plotting/_converter.py:129: FutureWa

To register the converters:
        >>> from pandas.plotting import register_matplotlib_converters
        >>> register_matplotlib_converters()
  warnings.warn(msg, FutureWarning)
```

```
NZ000933090
NZ000093417
NZ000939870
NZ000936150
NZ000093844
NZ000939450
NZ000093994
NZM00093781
NZ000093012
NZ000093292
```



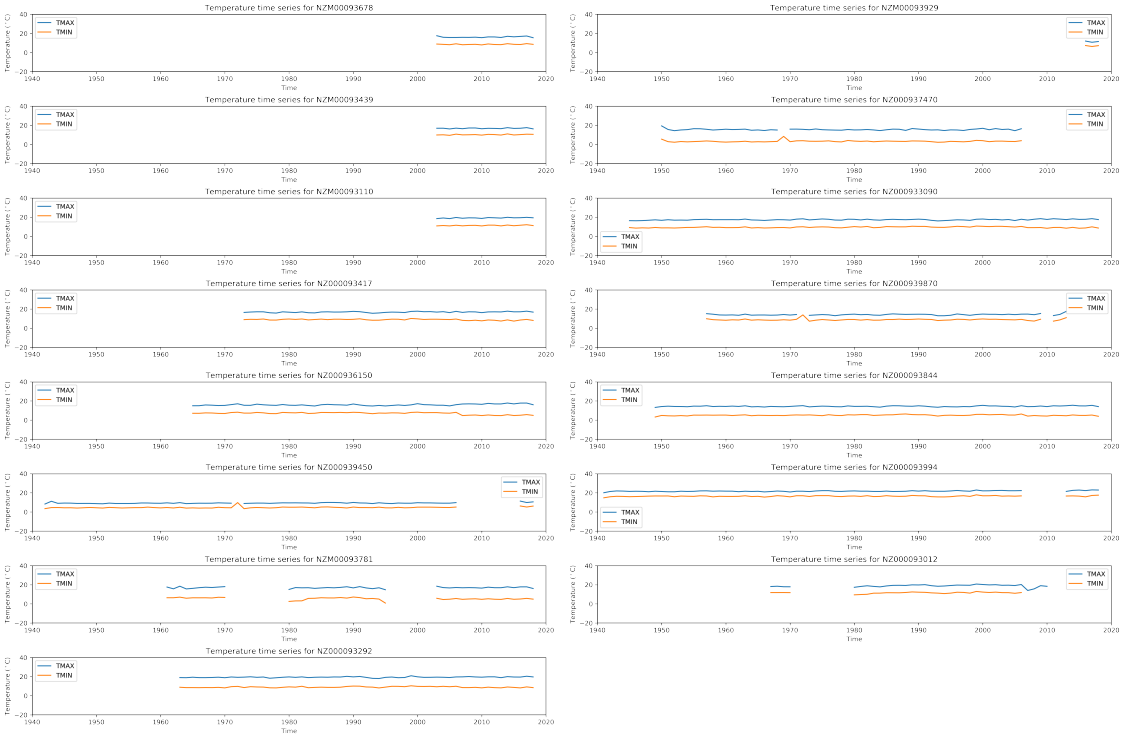
In [11]:
```
# Outputs
output_path = os.path.expanduser("~/Documents/plots")  # M:/plots on windows
if not os.path.exists(output_path):
    os.makedirs(output_path)

# Save
plt.tight_layout()  # reduce whitespace
f.savefig(os.path.join(output_path, f"Time series of TMIN and TMAX for NZ stations.png
plt.close(f)
```

```
<Figure size 432x288 with 0 Axes>
```

In [12]:
```
# Plot the time series of TMIN and TMAX on the same axis for each station in New Zeal
f, axes = plt.subplots(8, 2, dpi=300, figsize=(30, 20))  # affects output resolution
f.subplots_adjust(hspace=0.6,wspace=0.1)
axes = axes.flatten()  # squeeze grid of (8, 2) axes into a linear array for iterating

for i, station in enumerate(stations):  # generate one plot for each tag
    print(station)

    # Load

    data = df[df.ID==station][["DATE", "ELEMENT", "VALUE"]]
```

```
            data = data.pivot_table(index='DATE',columns='ELEMENT',values='VALUE')
            data = data.resample('Y').mean()

            # Plots

            a = axes[i]
            a.plot(data)  # assign label to include in legend
            a.set_ylim([-20, 40])  # exapnd axes slightly beyond [0, 10]
            a.set_xlim("1940", "2020")

            # Legend
            a.legend(data.columns,loc="best")

            # Labels
            a.set_title(f"Temperature time series for {station}")
            a.set_xlabel("Time")
            a.set_ylabel("Temperature ($^\circ$C)")

        # Remove empty subplots
        for a in axes[(i + 1):]:
            f.delaxes(a)
```

```
NZM00093678
NZM00093929
NZM00093439
NZ000937470
NZM00093110
NZ000933090
NZ000093417
NZ000939870
NZ000936150
NZ000093844
NZ000939450
NZ000093994
NZM00093781
NZ000093012
NZ000093292
```




```
In [13]: # Outputs
         output_path = os.path.expanduser("~/Documents/plots")  # M:/plots on windows
         if not os.path.exists(output_path):
             os.makedirs(output_path)

         # Save
         plt.tight_layout()  # reduce whitespace
         f.savefig(os.path.join(output_path, f"Time series of TMIN and TMAX for NZ stations(2)
         plt.close(f)
```

```
<Figure size 432x288 with 0 Axes>
```

2.1 Plot the average time series for the entire country

```
In [14]: data = (
             df[["DATE", "ELEMENT", "VALUE"]]
             .groupby(['ELEMENT','DATE'])
             .agg({"VALUE": ['mean']})
             .rename(columns={'mean': 'AVG_TEMP'})
         )
         data.columns = data.columns.droplevel(0)
         data.head()
```

Out[14]:
```
                     AVG_TEMP
ELEMENT DATE
TMAX    1940-03-08      26.1
        1940-03-09      24.3
        1940-03-10      25.0
        1940-03-11      26.1
        1940-03-12      23.3
```

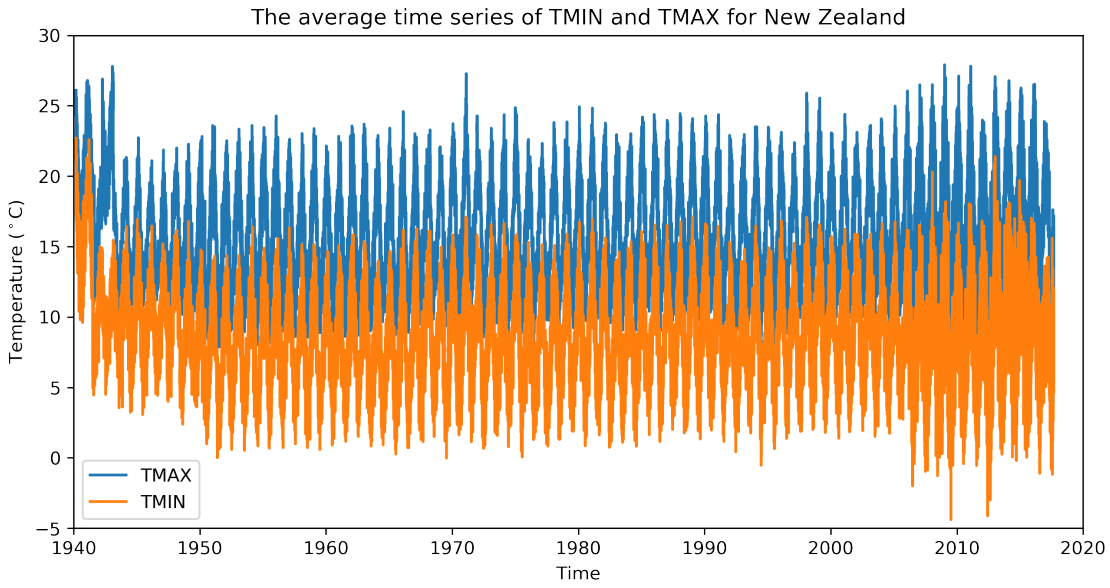
In [15]:
```
data = data.pivot_table(index='DATE',columns='ELEMENT',values='AVG_TEMP')
```

In [16]:
```
data.describe()
```

Out[16]:
```
ELEMENT          TMAX          TMIN
count    28283.000000  28203.000000
mean        16.391645      8.534455
std          3.563729      3.525118
min          6.711111     -4.400000
25%         13.500000      5.882576
50%         16.300000      8.512500
75%         19.120000     11.130000
max         27.911111     22.700000
```

In [17]:
```
# Plot average time series of TMIN and TMAX for entire New Zealand
f, a = plt.subplots(dpi=300, figsize=(10, 5))  # affects output resolution (dpi) and
a.plot(data, label=data.columns)  # assign label to include in legend
a.set_ylim([-5, 30])  # exapnd axes slightly beyond [0, 10]
a.set_xlim("1940", "2020")

# Legend
a.legend(data.columns)

# Labels
a.set_title(f"The average time series of TMIN and TMAX for New Zealand")
a.set_xlabel("Time")
a.set_ylabel("Temperature ($^\circ$C)")
```

Out[17]:
```
Text(0, 0.5, 'Temperature ($^\\circ$C)')
```

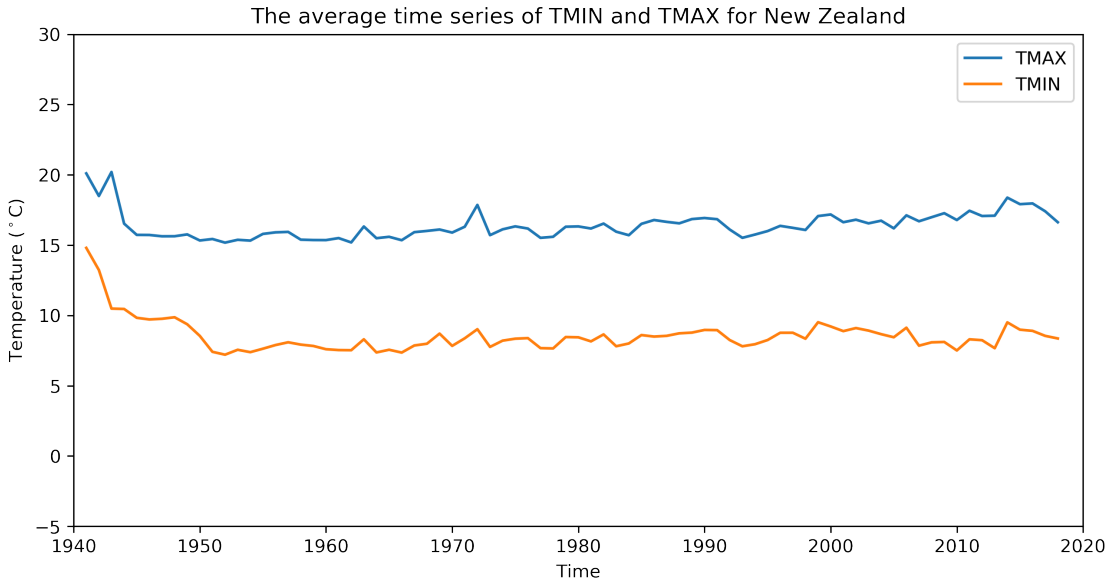


In [18]:
```
# Outputs
output_path = os.path.expanduser("~/Documents/plots")  # M:/plots on windows
if not os.path.exists(output_path):
    os.makedirs(output_path)

# Save
plt.tight_layout()  # reduce whitespace
f.savefig(os.path.join(output_path, f"Time series of TMIN and TMAX for entire New Zeal
plt.close(f)
```

```
<Figure size 432x288 with 0 Axes>
```

In [19]:
```
# Plot the average time series of TMIN and TMAX for entire New Zealand with resamplin
f, a = plt.subplots(dpi=300, figsize=(10, 5))  # affects output resolution (dpi) and
data = data.resample('Y').mean()
a.plot(data, label=data.columns)  # assign label to include in legend
a.set_ylim([-5, 30])  # exapnd axes slightly beyond [0, 10]
a.set_xlim("1940", "2020")

# Legend
a.legend(data.columns)

# Labels
a.set_title(f"The average time series of TMIN and TMAX for New Zealand")
a.set_xlabel("Time")
a.set_ylabel("Temperature ($^\circ$C)")
```

Out[19]: Text(0, 0.5, 'Temperature ($^\\circ$C)')



In [20]:
```
# Outputs
output_path = os.path.expanduser("~/Documents/plots")  # M:/plots on windows
if not os.path.exists(output_path):
    os.makedirs(output_path)

# Save
plt.tight_layout()  # reduce whitespace
f.savefig(os.path.join(output_path, f"The average time series of TMIN and TMAX for en
plt.close(f)
```

```
<Figure size 432x288 with 0 Axes>
```

In [ ]: