Lab 2 Maryna Borovyk Task 2. Average temperature dataset

Import dataset

In [104]:
```
from pathlib import Path
import pandas as pd
filepath = Path('data.csv')
```

In [105]:
```
average_temp = pd.read_csv(filepath,
                           header=0,
                           parse_dates=True,
                           usecols=['Date',
                                    'Value',
                                    'Anomaly'])
average_temp.head()
```

Out[105]:

	Date	Value	Anomaly
0	194001	55.8	-0.5
1	194101	56.4	0.1
2	194201	57.7	1.4
3	194301	56.3	0.0
4	194401	56.1	-0.2

Locate missing values and change them to nan

In [106]:
```
import numpy as np
average_temp = average_temp.replace([-99: np.nan])
print(average_temp.isna())
```

```
    Date  Value  Anomaly
0  False  False    False
1  False  False    False
2  False  False    False
3  False  False    False
4  False  False    False
..   ...    ...      ...
78 False  False    False
79 False  False    False
80 False  False    False
81 False  False    False
82 False  False    False

[83 rows x 3 columns]
```

Use the interpolate function to put a value in the Nan's place

In [107]:
```
average_temp.interpolate(inplace = True)
print(average_temp)
```

```
    Date  Value  Anomaly
0  194001   55.8     -0.5
1  194101   56.4      0.1
2  194201   57.7      1.4
3  194301   56.3      0.0
4  194401   56.1     -0.2
..   ...    ...      ...
78 201801   59.4      3.1
79 201901   57.8      1.5
80 202001   57.9      1.6
81 202101   58.0      1.7
82 202201   58.0      2.3

[83 rows x 3 columns]
```

Convert the index to datetime format

In [108]:
```
average_temp['Date'] = pd.to_datetime(average_temp['Date'], format = '%Y%m')
average_temp.set_index('Date', inplace=True)
average_temp.head()
```

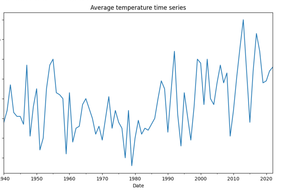
Out[107]:

	Value	Anomaly
Date		
1940-01-01	55.8	-0.5
1941-01-01	56.4	0.1
1942-01-01	57.7	1.4
1943-01-01	56.3	0.0
1944-01-01	56.1	-0.2

Plot the average temperature time series, the corresponding histogram, and kernel density plot

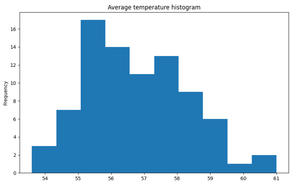
In [108]:
```
average_temp['Value'].plot(figsize=(10, 6), title = 'Average temperature time series')
```

Out[108]:
```
<Axes: title={'center': 'Average temperature time series'}, xlabel='Date'>
```



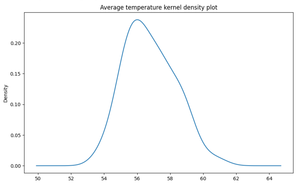
In [109]:
```
average_temp['Value'].plot.hist(figsize=(10, 6), title = 'Average temperature histogram')
```

Out[109]:
```
<Axes: title={'center': 'Average temperature histogram'}, ylabel='Frequency'>
```



In [110]:
```
average_temp['Value'].plot.kde(figsize=(10, 6), title = 'Average temperature kernel density plot')
```

Out[110]:
```
<Axes: title={'center': 'Average temperature kernel density plot'}, ylabel='Density'>
```



Generate descriptive statistics

In [111]:
```
average_temp.describe()
```

Out[111]:

	Value	Anomaly
count	83.000000	83.000000
mean	56.796265	0.466265
std	1.536331	1.536331
min	53.600000	-2.700000
25%	55.600000	-0.700000
50%	56.600000	0.300000
75%	57.800000	1.500000
max	61.000000	4.700000

Task 1. Vacation search results Set datetime index for each dataframe.

In [112]:
```
poland_df = pd.read_csv('poland.csv', parse_dates=[0], header=1)
uk_df = pd.read_csv('UK.csv', parse_dates=[0], header=1)
usa_df = pd.read_csv('USA.csv', parse_dates=[0], header=1)
```

In [113]:
```
poland_df.set_index('Month', inplace=True)
uk_df.set_index('Month', inplace=True)
usa_df.set_index('Month', inplace=True)
```

In [114]:
```
usa_df.head()
```

Out[114]:

	vacation: (United States)
Month	
2004-01-01	97
2004-02-01	87
2004-03-01	83
2004-04-01	79
2004-05-01	82

Rename the columns to the country name

In [115]:
```
usa_df.rename(columns={"vacation: (United States)": "US"}, inplace=True)
poland_df.rename(columns={"vacation: (Poland)": "PL"}, inplace=True)
uk_df.rename(columns={"vacation: (United Kingdom)": "UK"}, inplace=True)
```

In [116]:
```
usa_df.head()
```

Out[116]:

	US
Month	
2004-01-01	97
2004-02-01	87
2004-03-01	83
2004-04-01	79
2004-05-01	82

Combine the search counts in one dataframe

In [117]:
```
combined_df = pd.concat([usa_df, poland_df, uk_df], axis=1)
combined_df.head()
```

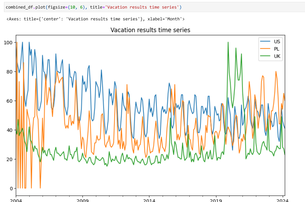
Out[117]:

	US	PL	UK
Month			
2004-01-01	97	0	40
2004-02-01	87	100	37
2004-03-01	83	0	47
2004-04-01	79	73	36
2004-05-01	82	0	38

Present the time series for all countries in one plot

In [118]:
```
combined_df.plot(figsize=(10, 6), title='Vacation results time series')
```

Out[118]:
```
<Axes: title={'center': 'Vacation results time series'}, xlabel='Month'>
```



Generate descriptive statistics

In [119]:
```
combined_df.describe()
```

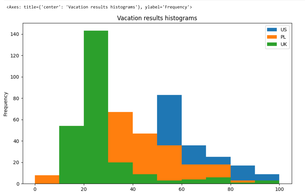
Out[119]:

	US	PL	UK
count	243.000000	243.000000	243.000000
mean	57.395062	42.510288	27.760494
std	15.221515	17.128014	14.386739
min	26.000000	0.000000	15.000000
25%	47.000000	30.500000	20.000000
50%	55.000000	40.000000	23.000000
75%	65.000000	51.500000	27.000000
max	100.000000	100.000000	100.000000

Show three histograms in one plot

In [120]:
```
combined_df.plot.hist(figsize=(10, 6), title='Vacation results histograms')
```

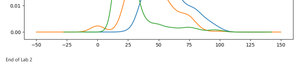
Out[120]:
```
<Axes: title={'center': 'Vacation results histograms'}, ylabel='Frequency'>
```



Show three kernel densities in one plot

In [121]:
```
combined_df.plot.kde(figsize=(10, 6), title='Vacation results kernel densities')
```

Out[121]:
```
<Axes: title={'center': 'Vacation results kernel densities'}, ylabel='Density'>
```



End of Lab 2