Section 1.3

October 1, 2023

Figures for Section 1.3

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

[2]:
```
# read dat file
def reaDat(filename):
    # read dat to a list of lists
    data = [i.strip().split() for i in open("../../data/"+filename).readlines()]
    data_df = pd.DataFrame(data)

    # change datatype from str to int
    data_df = data_df.astype({0:'float'})

    return data_df
```

[3]:
```
# time series plot
def plotDat(ts, xlim):
    fig, ax = plt.subplots()
    ax.plot(xlim,ts,'o-')
```

Figure 1.7

Annual water levels of Lake Huron (left panel) and the residual plot obtained from fitting a linear trend to the data (right panel).

[4]:
```
lake = reaDat("lake.dat")
```

[5]:
```
#find line of best fit
x = range(1,len(lake)+1)
x_1875 = range(1875,1973)
a, b = np.polyfit(x, lake, 1)
fitted = np.polyval([a,b],x)
```

[6]:
```
fig, axs = plt.subplots(1, 2, figsize=(9,4),constrained_layout = True)
axs[0].plot(x_1875,lake)
axs[0].plot(x_1875,a*x+b)
```

```
axs[1].plot(x_1875,lake[0]-fitted)
```

[6]:
```
[<matplotlib.lines.Line2D at 0x7fccd8d74510>]
```

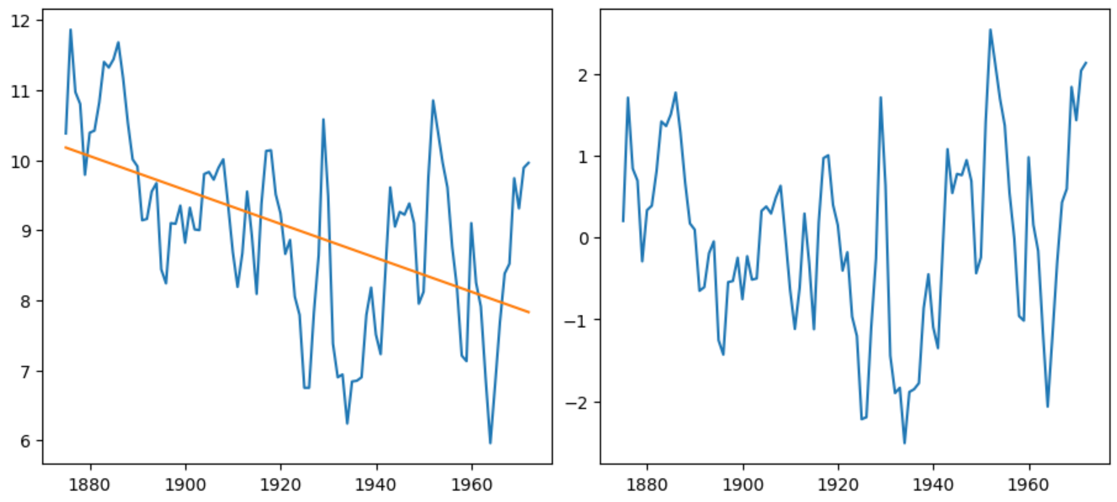


Figure 1.8

The two-sided moving average filters Wt for the Lake Huron data (upper panel) and their residuals (lower panel) with bandwidth q = 2 (left), q = 10 (middle) and q = 35 (right).

[25]:
```
# create moving windows
def MA(data,q):
    # create an empty array
    l = len(data)
    ma = np.zeros(len(data))

    # initialize the previous and
    # latter q entries with NaNs
    ma[0:q] = np.nan
    ma[l-q:l] = np.nan
    for i in range(q,l-q):
        # calcualte the mean values between
        # the previous q and latter q entries
        ma[i] = data[i-q:i+q+1].mean()

    # calcualte the fitted
    x_range = range(l-2*q)
    data_fit = data[q:l-q]
    a,b = np.polyfit(x_range, data_fit,1)
    fitted = np.polyval([a,b],x_range)
    return (ma, fitted, data_fit)
```

```
[29]: ma2, fit2, d_fit2 = MA(lake[0],2)
      ma10, fit10, d_fit10 = MA(lake[0],10)
      ma35, fit35, d_fit35 = MA(lake[0], 35)

      # reset the x_ticks for residuals
      range_fit2 = range(1875+2,1973-2)
      range_fit10 = range(1875+10,1973-10)
      range_fit35 = range(1875+35,1973-35)
```

```
[31]: fig, axs = plt.subplots(2, 3, figsize=(9,4),constrained_layout = True)
      axs[0,0].plot(x_1875,ma2, label='Moving Average (window=2)')
      axs[0,0].plot(range_fit2,fit2)

      axs[0,1].plot(x_1875,ma10, label='Moving Average (window=10)')
      axs[0,1].plot(range_fit10,fit10)

      axs[0,2].plot(x_1875,ma2, label='Moving Average (window=35)')
      axs[0,2].plot(range_fit35,fit35)

      axs[1,0].plot(range_fit2, d_fit2-fit2)
      axs[1,1].plot(range_fit10, d_fit10-fit10)
      axs[1,2].plot(range_fit35, d_fit35-fit35)
```

```
[31]: [<matplotlib.lines.Line2D at 0x7fccd9474310>]
```

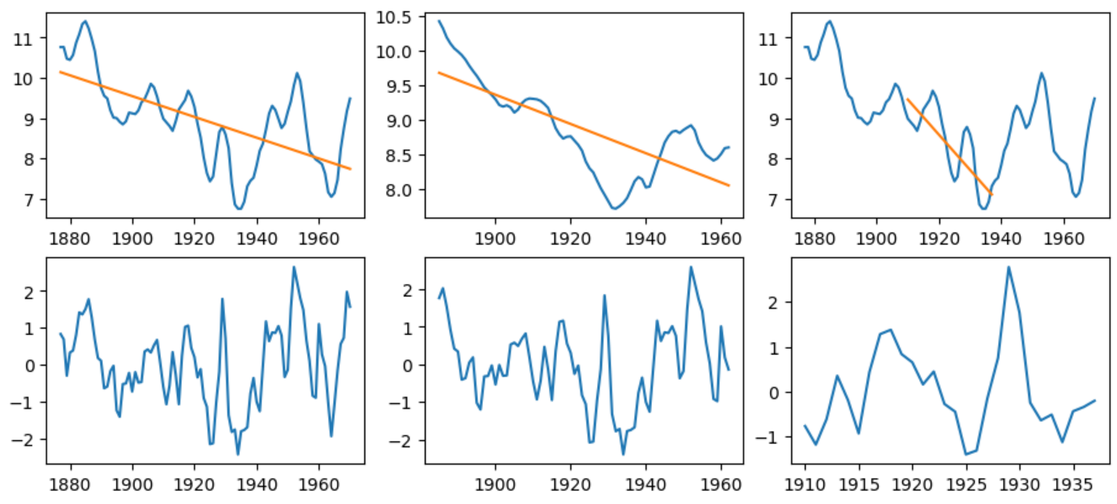


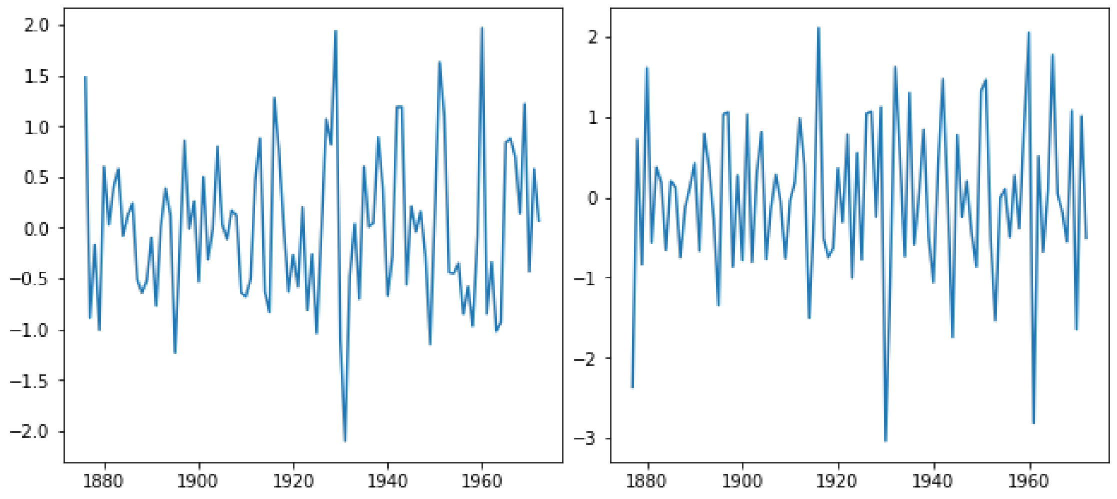
Figure 1.9

Time series plots of the observed sequences (∇x_t) in the left panel and $(\nabla^2 x_t)$ in the right panel of the differenced Lake Huron data described in Example 1.3.1.

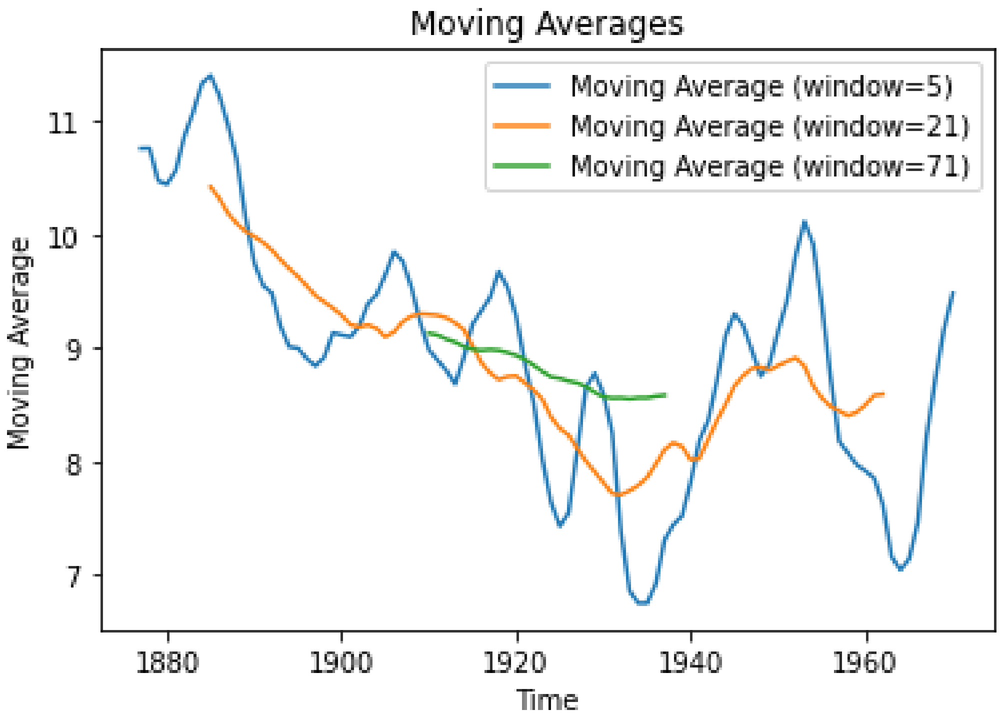
[10]:
```
df1 = np.diff(lake[0])
df2 = np.diff(df1)
```

[11]:
```
fig, axs = plt.subplots(1, 2, figsize=(9,4),constrained_layout = True)
axs[0].plot(range(1876,1973),df1)
# plt.show()
axs[1].plot(range(1877,1973),df2)
```

[11]:
```
[<matplotlib.lines.Line2D at 0x7fa0710cb0d0>]
```



[145]:
```
# Generate 't' as a sequence from 1 to the length of 'lake'
t = np.arange(1, len(lake) + 1)

# Define the moving averages
ma2 = pd.Series(lake[0]).rolling(window=5, center=True).mean()
ma10 = pd.Series(lake[0]).rolling(window=21, center=True).mean()
ma35 = pd.Series(lake[0]).rolling(window=71, center=True).mean()

a_2, b_2 = np.polyfit(x, ma2, 1)
fitted_2 = np.polyval([a_2,b_2],x)


# Plot the moving averages
plt.plot(x_1875, ma2, label='Moving Average (window=5)')
# plt.plot(x_1875,fitted_2)
plt.plot(x_1875, ma10, label='Moving Average (window=21)')
plt.plot(x_1875, ma35, label='Moving Average (window=71)')

# Set the plot labels and title
plt.xlabel('Time')
plt.ylabel('Moving Average')
plt.title('Moving Averages')
plt.legend()

# Display the plot
plt.show()
```


Moving Averages
Moving Average (window=5)
Moving Average (window=21)
Moving Average (window=71)
Moving Average
Time