ELEC 390 Lab 5

Liam Salass (20229595)

Charlotte Lombard (20232888)

Mile Stosic (20233349)

Thursday, March 23rd

Question 1

a) There were 48 ‘NaN’ values found and 10 ‘-’ values.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

data_set = pd.read_csv('Lab 5/unclean-wine-quality.csv')


# Question 1

#drop first column
data_set = data_set.drop(data_set.columns[0], axis=1)


labels = data_set.iloc[:,11]
data = data_set.iloc[:,0:11]

# use np.where to and sum to count number of - and NaN values
print('\nQuestion 1: \n')
print('number of - values: ', np.sum(np.where(data == '-', 1, 0)))
print ('Number of NaN values: ', np.sum(np.where(data.isnull(), 1, 0)))

# Print the indicies of the - values in the dataset
print('Indicies of - values: \n', np.where(data == '-'))
print ('Indicies of NaN values: \n', np.where(data.isnull()))

# Change all - values to NaN
print('\nReplacing all - values with NaN')
data = data.replace('-', np.nan)


# Print number of null values in the dataset
print ('Number of NaN values: ', np.sum(np.where(data.isnull(), 1, 0)))

# Change all values in the data set to float64
data = data.astype('float64')
```

Output:

```
Question 1:

number of - values:  10
Number of NaN values:  48
Indicies of - values:
 (array([   4,   97,  117,  162,  183,  234,  583,  920, 1024, 1087],
      dtype=int64), array([3, 3, 4, 5, 5, 5, 4, 2, 2, 2], dtype=int64))
Indicies of NaN values:
 (array([  17,   33,   54,   79,   86,   98,  139,  174,  179,  221,  224,
        229,  249,  267,  268,  368,  380,  438,  440,  518,  521,  587,
        589,  621,  623,  697,  747,  812,  830,  909,  972, 1077, 1079,
       1079, 1591, 2894, 2902, 2902, 4892, 4895, 6320, 6321, 6428, 6428,
       6429, 6429, 6486, 6493], dtype=int64), array([0, 3, 8, 3, 1, 4, 8, 0, 3, 5, 9, 3, 0, 0, 2, 0, 3, 3, 8, 0, 1, 8,
       5, 1, 2, 8, 4, 1, 5, 2, 8, 2, 0, 1, 2, 1, 0, 9, 8, 1, 2, 9, 0, 8,
       0, 8, 1, 9], dtype=int64))

Replacing all - values with NaN
Number of NaN values:  58
```

Question 2

a) There were 0 NaN values after replacement.

```
# Question 2
# Filling missing values with a constant value
print ('\nQuestion 2: \n')
data2 = data
data2 = data2.fillna({'fixed acidity': 0})
data2 = data2.fillna({'volatile acidity': 0})
data2 = data2.fillna({'citric acid': 0})
data2 = data2.fillna({'residual sugar': 0})
data2 = data2.fillna({'chlorides': 1})
data2 = data2.fillna({'free sulfur dioxide': 0})
data2 = data2.fillna({'total sulfur dioxide': 0})
data2 = data2.fillna({'density': 0})
data2 = data2.fillna({'pH': 1})
data2 = data2.fillna({'sulphates': 1})
data2 = data2.fillna({'alcohol': 0})

print ('Number of NaN values after replacement: ', np.sum(np.where(data2.isnull(), 1, 0)))
```

Output:

```
Question 2:

Number of NaN values after replacement:  0
```

Question 3

```
# Question 3

print('\nQuestion 3: \n')
data3 = data.fillna(method='ffill')
print('Sample-and-hold filling: \n', data3.iloc[16:19,0])
```

Output:

```
Question 3:

Sample-and-hold filling:
 16    6.3
17    6.3
18    7.4
Name: fixed acidity, dtype: float64
```

Question 4

a) The value that replaces [17,0] is the average of the value at [16,0] and [18,0]. This value is 6.85.

```
# Question 4
print('\nQuestion 4: \n')
data4 = data.interpolate(method='linear')
print('Linear interpolation: \n', data4.iloc[16:19,0])
```

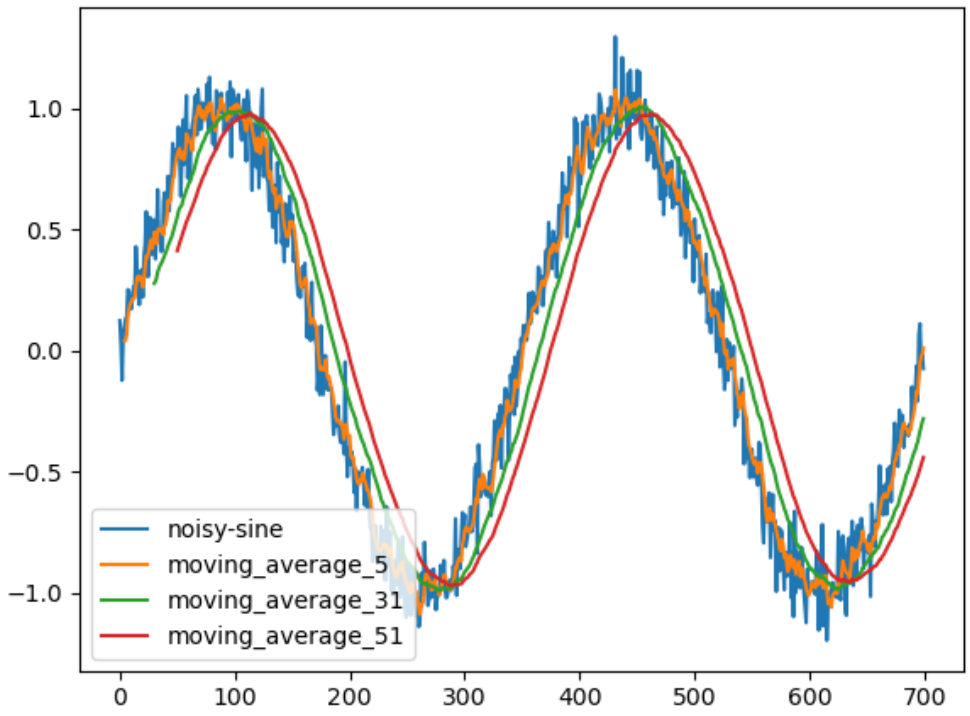
Output:

```
Question 4:

Linear interpolation:
 16    6.30
17    6.85
18    7.40
Name: fixed acidity, dtype: float64
```

Question 5

```
# Question 5
print('='*100)
print('\nQuestion 5: \n')
noisy_data = pd.read_csv('Lab 5/noisy-sine.csv')
# apply an moving average filter on the noisy_data with window size 5, 31 and 51
# then plot the original noisy_data along with result of the three moving average filters

# window size 5
noisy_data5 = noisy_data.rolling(window=5).mean()
# window size 31
noisy_data31 = noisy_data.rolling(window=31).mean()
# window size 51
noisy_data51 = noisy_data.rolling(window=51).mean()

# plot the original noisy_data along with result of the three moving average filters
plt.plot(noisy_data, label='noisy-sine')
plt.plot(noisy_data5, label='moving_average_5')
plt.plot(noisy_data31, label='moving_average_31')
plt.plot(noisy_data51, label='moving_average_51')
plt.legend()
plt.show()
```

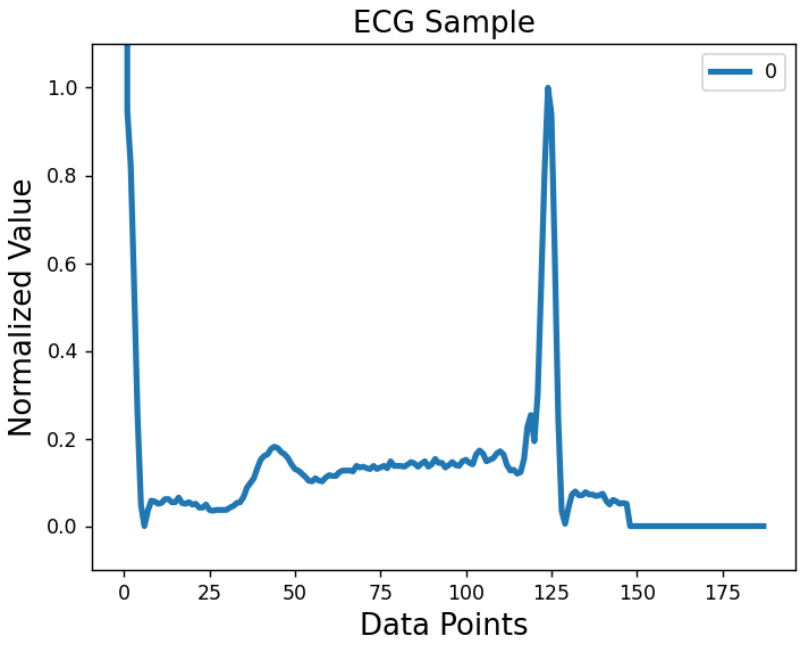
Output:



Question 6

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# i)
dataset = pd.read_csv("C:\\Users\\miles\\OneDrive - Queen's University\\Eng Year 3 - 2022-2023\\Sem 2\\ELEC 390\\Lab5

fig, ax = plt.subplots()
dataset.iloc[:].plot(ax=ax, linewidth=3)

ax.set_title('ECG Sample', fontsize=15)
ax.set_xlabel('Number of the window')
ax.set_ylabel('Value of the std')
ax.set_ylim(-0.005,0.31)
plt.show()

# ii)
features = pd.DataFrame(columns=['mean', 'std', 'max', 'min'])
window_size = 31
features['mean'] = dataset.iloc[:].rolling(window=window_size).mean()
features['std'] = dataset.iloc[:].rolling(window=window_size).std()
features['max'] = dataset.iloc[:].rolling(window=window_size).max()
features['min'] = dataset.iloc[:].rolling(window=window_size).min()
features = features.dropna()
print(features)

# iii)
features['std'].plot()
plt.show()

```

i)



ii)

	mean	std	max	min
30	0.123183	0.225509	0.947183	0.0
31	0.094105	0.165975	0.822183	0.0
32	0.069287	0.096427	0.545775	0.0
33	0.053442	0.038445	0.250000	0.0
34	0.047535	0.012655	0.066901	0.0
..	...	...	...	...
182	0.000000	0.000000	0.000000	0.0
183	0.000000	0.000000	0.000000	0.0
184	0.000000	0.000000	0.000000	0.0
185	0.000000	0.000000	0.000000	0.0
186	0.000000	0.000000	0.000000	0.0

[157 rows x 4 columns]

iii)