

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from sklearn.preprocessing import Imputer
5
6 #southbound 5 minutes aggregation
7 data_west_5 = pd.read_csv('/Users/
artyomkholodkov/Downloads/1016/
aggregated_1016_west_5.csv')
8 data_east_5 = pd.read_csv('/Users/
artyomkholodkov/Downloads/1016/
aggregated_1016_east_5.csv')
9
10 def clean_data(dataset):
11     imputer = Imputer(strategy="median")
12     data_noindex = dataset.drop("TIME", axis
= 1)
13     imputer.fit(data_noindex)
14     filled_data = imputer.transform(
data_noindex)
15     data_transformed = pd.DataFrame(
filled_data, index=dataset.TIME)
16     data_transformed.columns = ["SPEED", "
HEADWAY", "SPACE_HEADWAY", "DENSITY", "FLOW"]
17     return data_transformed
18
19 def select_10_tickers(ts_array):
20     idx = np.round(np.linspace(0, len(
ts_array) - 1, 10)).astype(int)
21     return ts_array[idx]
22
23 def plot_dataset(dataset):
24     for column in dataset.columns:
25         ax=plt.subplot(len(dataset.columns),
1, dataset.columns.get_loc(column) + 1)
26         plt.title(column, y=0.5, loc='right')
27         ax.xaxis_date()
28         ax.xaxis.set_ticks(select_10_tickers(
dataset.index))
```

```

29         plt.plot(dataset.index, np.array(
dataset[column]))
30         plt.show()
31
32 from scipy.signal import lfilter
33 def denoise_data(cleaned_dataset):
34     n = 2
35     b = [1.0 / n] * n
36     a = 1
37     filtered_data = lfilter(b,a,
cleaned_dataset.FLOW)
38     return np.array(filtered_data)
39
40 cleaned_data_west_5 = clean_data(data_west_5)
41 dropped_flow_pd = cleaned_data_west_5.drop(['
FLOW'], axis = 1)
42 denoised_flow_west = denoise_data(
cleaned_data_west_5)
43 dropped_flow_pd["FLOW"] = denoised_flow_west
44 # dropped_flow_pd.to_csv('
denoised_data_1016_west.csv')
45
46 cleaned_data_east_5 = clean_data(data_east_5)
47 dropped_flow_pd_east = cleaned_data_east_5.
drop(['FLOW'], axis = 1)
48 denoised_flow_east = denoise_data(
cleaned_data_east_5)
49 dropped_flow_pd_east["FLOW"] =
denoised_flow_east
50 # dropped_flow_pd_east.to_csv('
denoised_data_1016_east.csv')
51
52
53
54
55 plt.figure()
56 plt.title("Actual data and denoised data ",
fontsize=14)
57 plt.plot(denoised_flow_west[-300:], 'bx-',

```

```
57 label = 'Denoised series')
58 plt.plot(cleaned_data_west_5.FLOW[-300:], 'r
    -', label = 'Original series')
59 plt.legend(loc="upper left")
60 plt.xlabel("Time (in 5 minutes intervals)")
61 plt.ylabel('Volume (veh/hour)')
62 plt.show()
63
64
```