Labolatory 1. python review

Bartłomiej Gondek

Importing essential libraries:

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

Reading (first column containing date-time stamp is set as index) and presenting data:

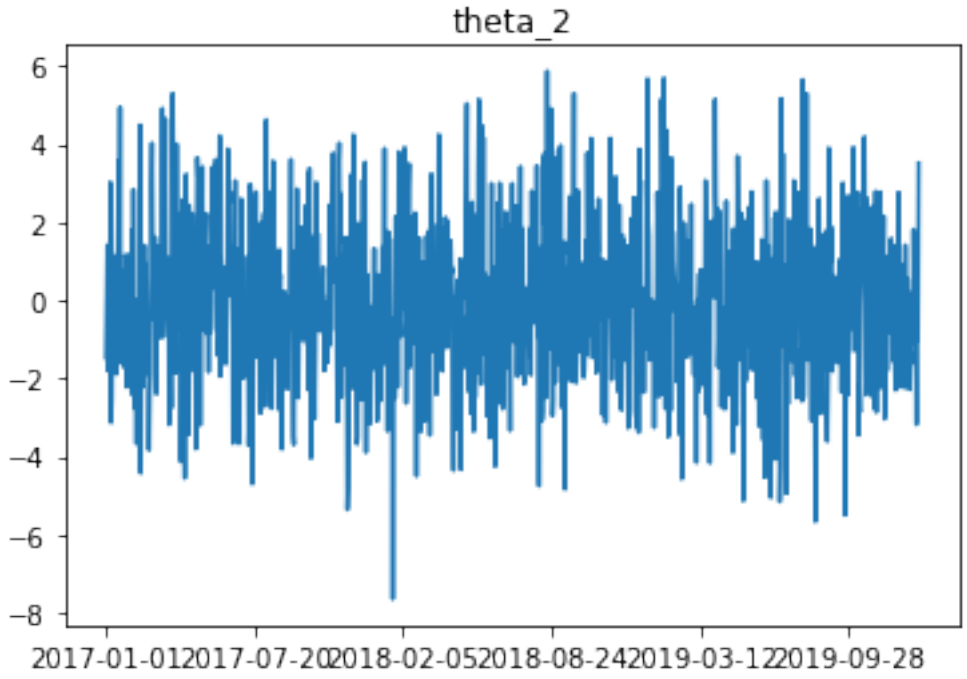
[2]:
```
df = pd.read_csv('../working_data/Data1.csv',index_col=0)
df.head()
```

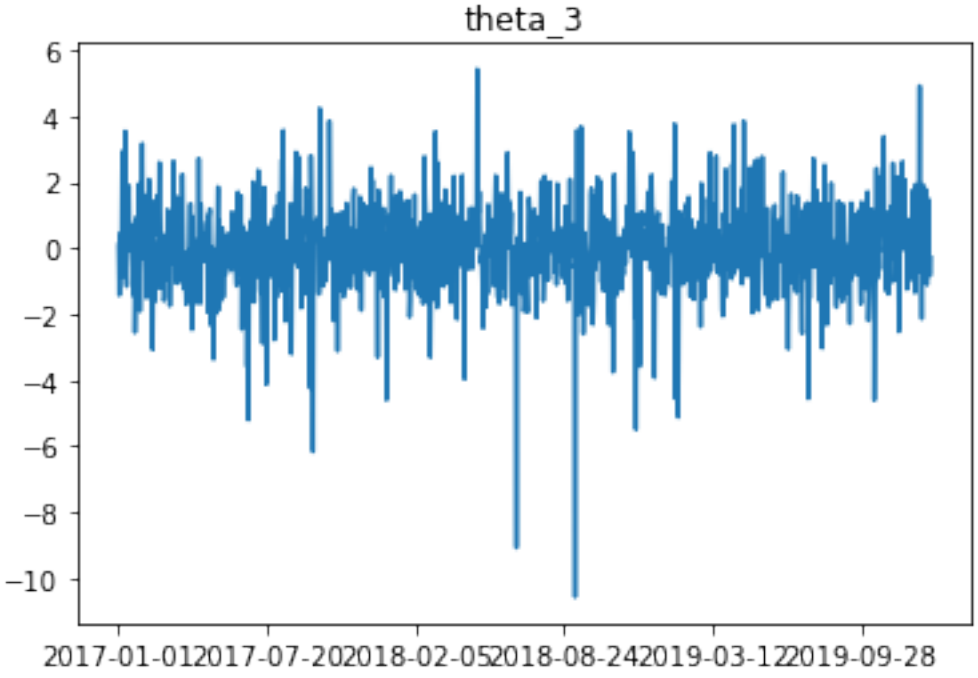
[2]:

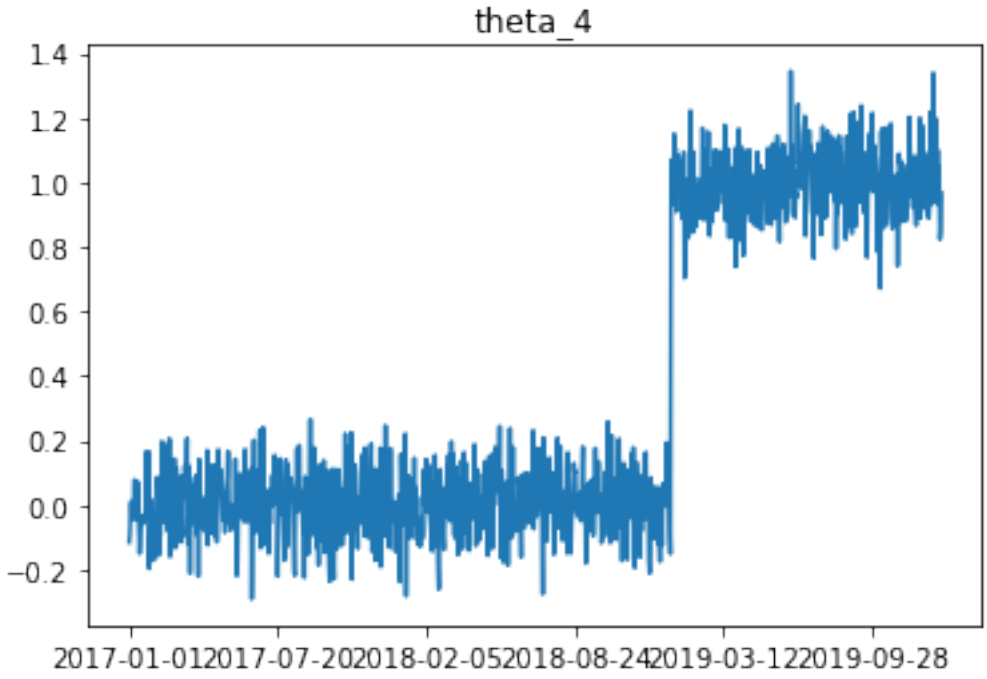
	theta_1	theta_2	theta_3	theta_4	theta_5	theta_6
2017-01-01	0.756936	-1.467790	0.096136	-0.115306	-0.447908	0.902579
2017-01-02	0.767089	0.185797	-1.428536	-0.086443	-0.954288	1.930909
2017-01-03	0.404544	1.415887	0.443466	0.000200	-0.892351	2.449691
2017-01-04	1.313957	-1.804471	-0.836986	0.011785	-1.012518	1.182085
2017-01-05	0.209862	1.315868	0.140993	-0.046473	-1.417092	1.742433

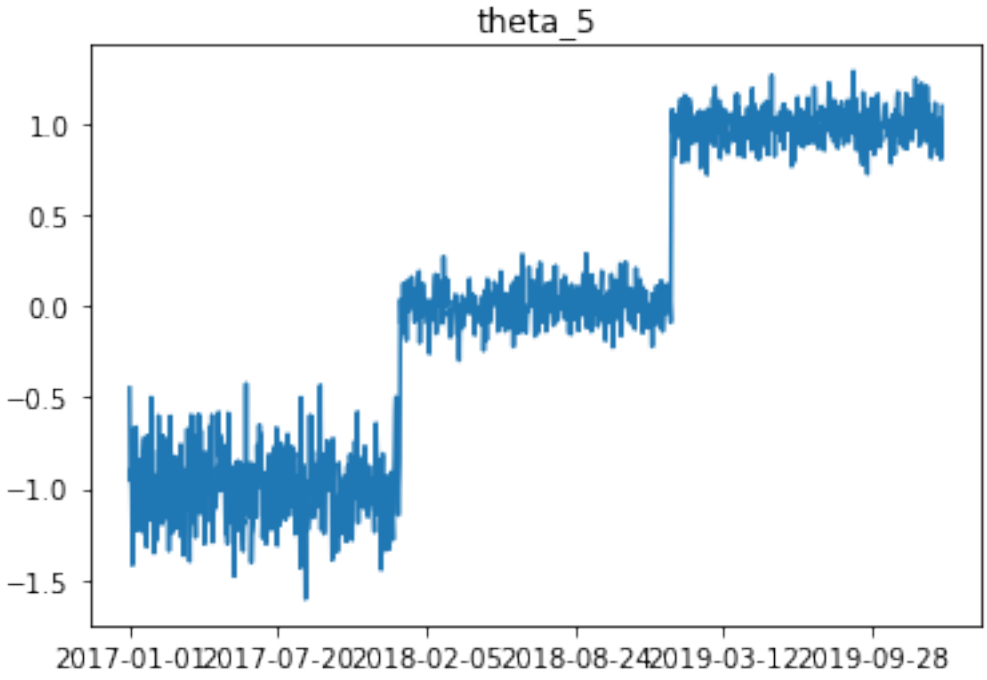
Plotting each column as timeseries with respect to index column (date-time stamp), each on seperate plot:

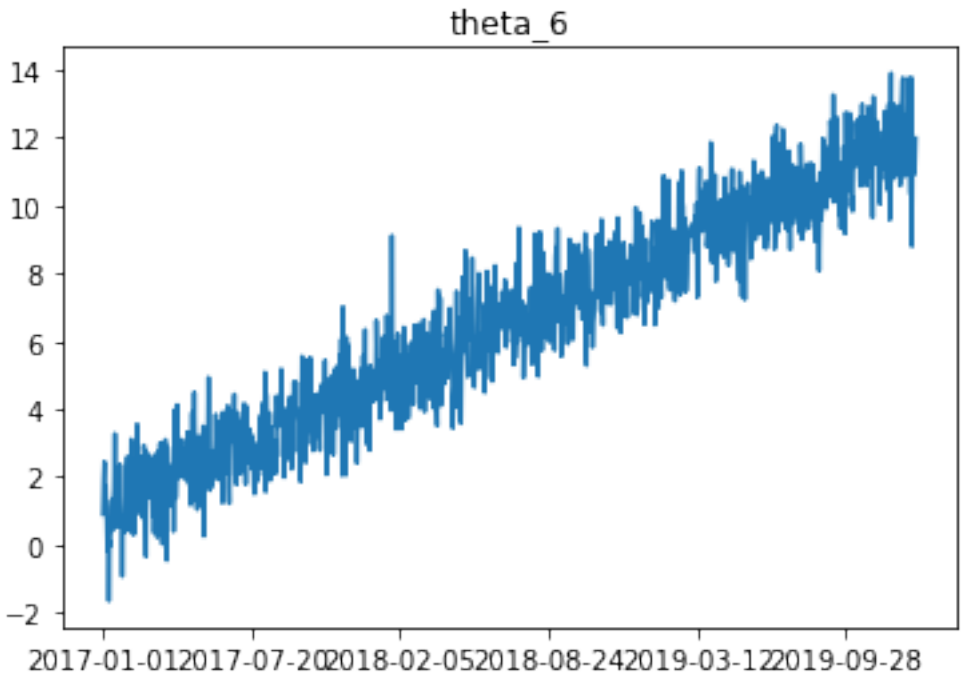
[3]:
```
for c in df.columns[1:]:
    df[c].plot()
    plt.title(c)
    plt.show()
```


theta_2
6
4
2
0
−2
−4
−6
−8
2017-01-01
2017-07-20
2018-02-05
2018-08-24
2019-03-12
2019-09-28


theta_3
6
4
2
0
−2
−4
−6
−8
−10
2017-01-01
2017-07-20
2018-02-05
2018-08-24
2019-03-12
2019-09-28

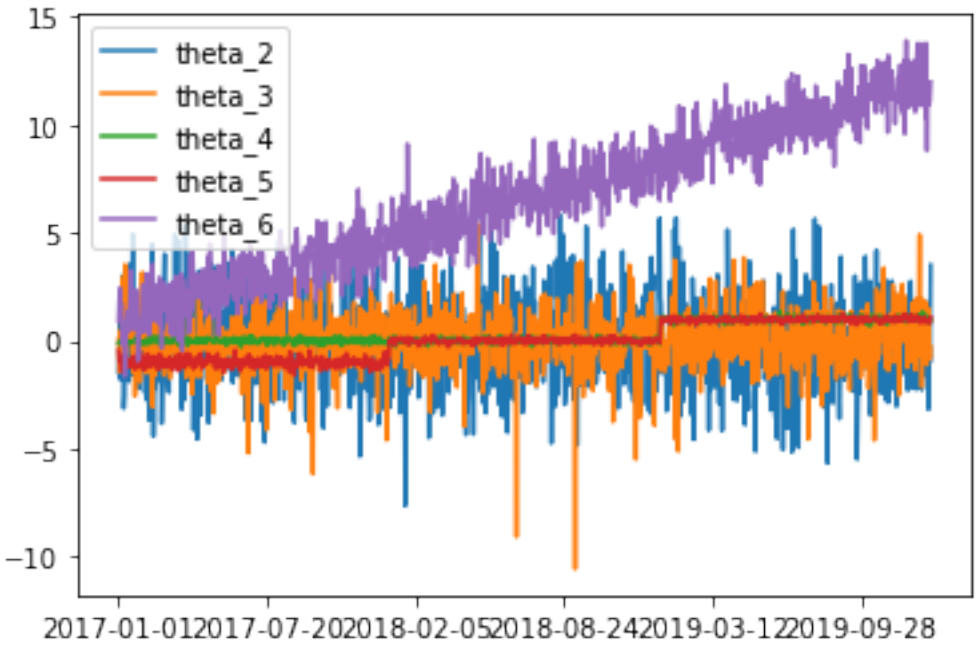

theta_4
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
2017-01-01
2017-07-20
2018-02-05
2018-08-24
2019-03-12
2019-09-28


theta_5
1.0
0.5
0.0
−0.5
−1.0
−1.5
2017-01-01
2017-07-20
2018-02-05
2018-08-24
2019-03-12
2019-09-28



Plotting each column as timeseries with respect to index column (date-time stamp) on the same plots:

[4]:
```
for c in df.columns[1:]:
    df[c].plot()
plt.legend(df.columns[1:])
plt.show()
```



Melting dataframe from form of:

index, θ_1, θ_2, θ_3, θ_4, θ_5, θ_6

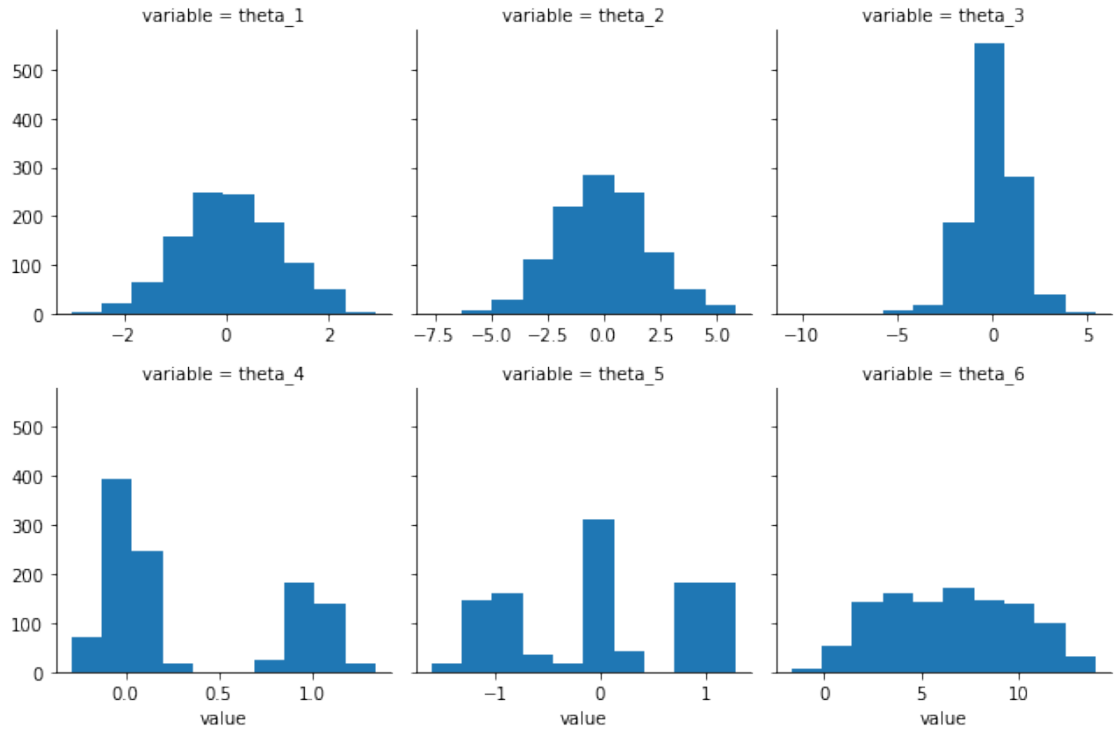
into:

index, θ, value

[5]:
```
dfm = df.melt()
```

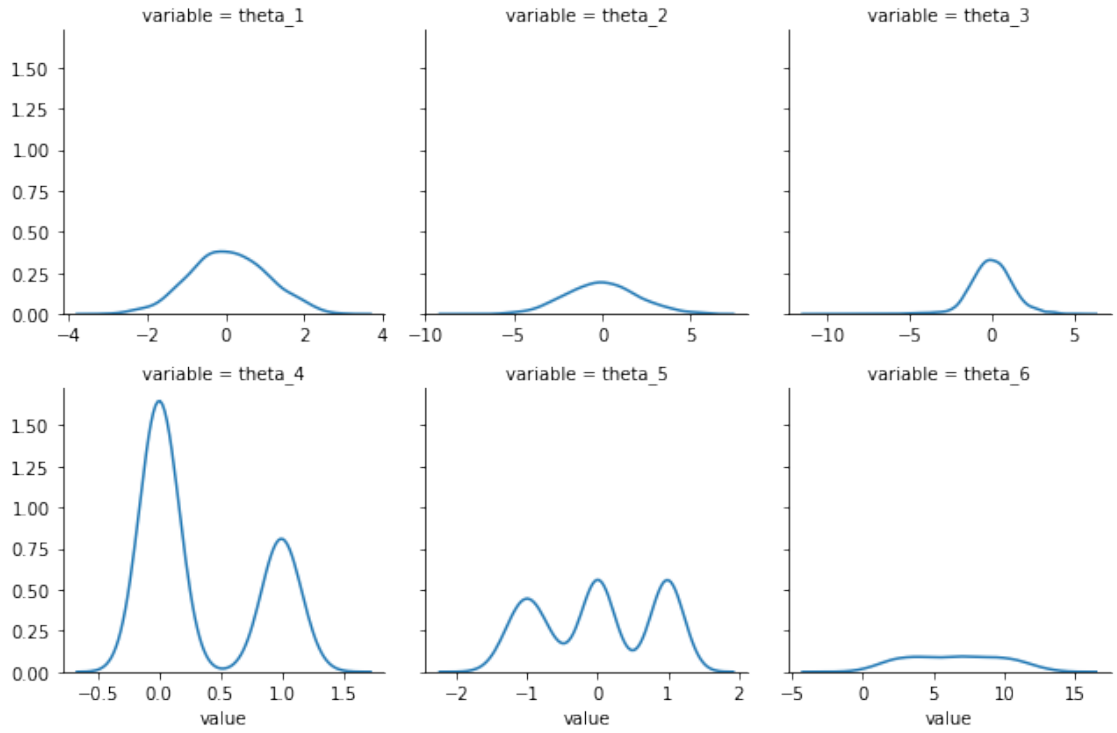
Creating a facet grid and creating histograms for each variable in different cell:

[6]:
```
g = sns.FacetGrid(dfm, col='variable', col_wrap=3, sharex = False)
g = g.map(plt.hist, 'value')
```



Creating a facet grid with *KernalDensityEstimator* plots:

[7]:
```
f = sns.FacetGrid(dfm, col='variable', col_wrap=3, sharex = False)
f = f.map(sns.kdeplot, 'value')
```



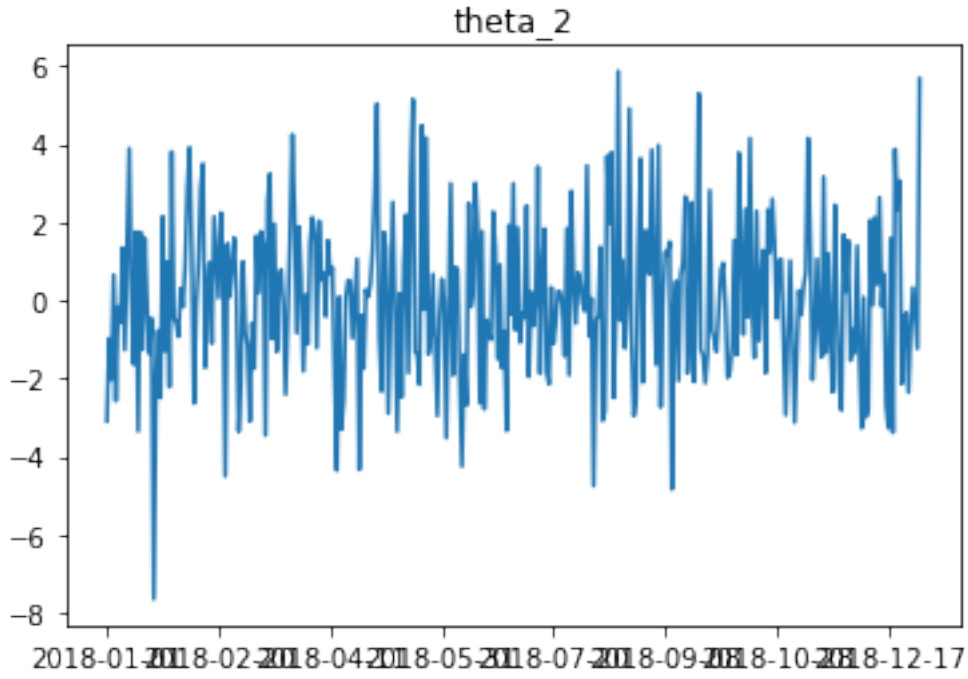
Limiting dataframe to contain only variables θ_1 to θ_4 noted only in year 2018 and repeating whole process presented above for limited data:

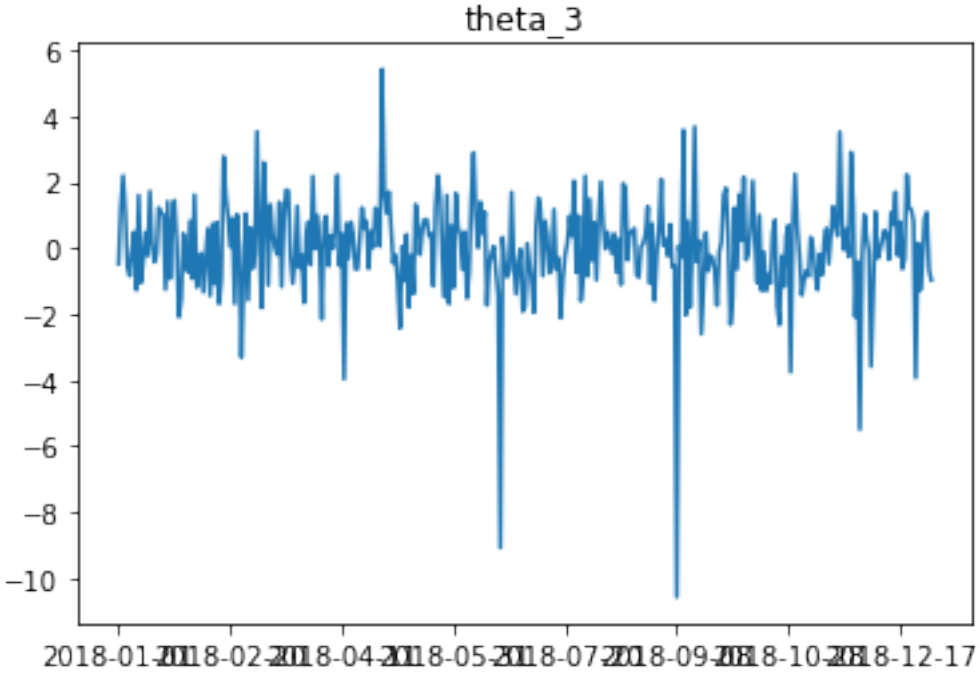
[8]:
```
df2 = df.loc['2018-01-01':'2018-12-31','theta_1':'theta_4']
df2.head()
```

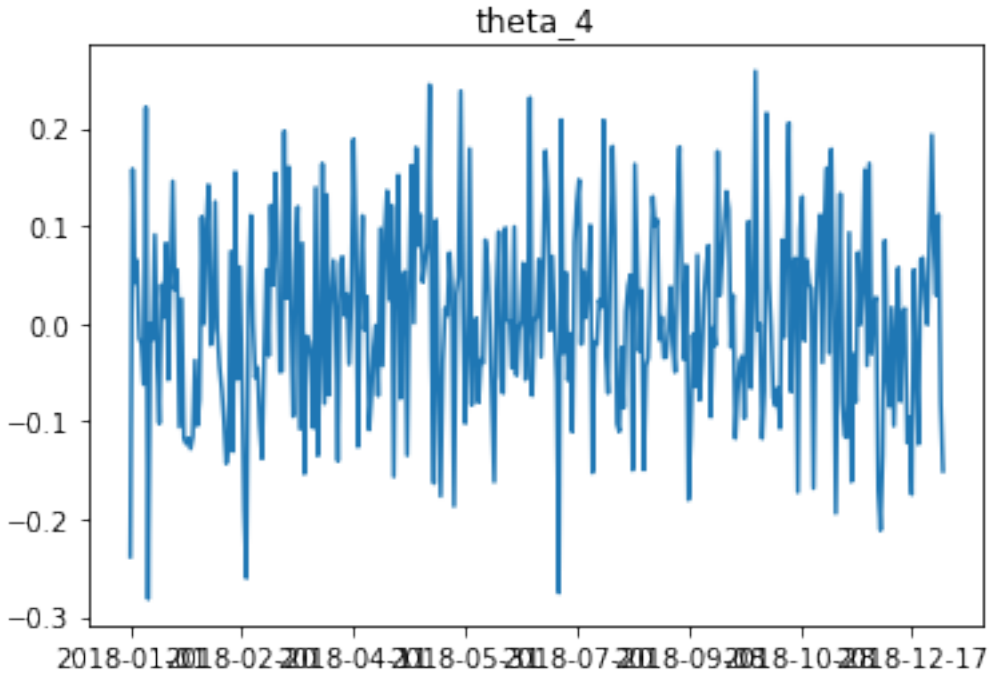
[8]:

	theta_1	theta_2	theta_3	theta_4
2018-01-01	0.682693	-3.091767	-0.475717	-0.238530
2018-01-02	-0.283107	-0.979955	1.233933	0.158031
2018-01-03	1.572221	-2.033528	2.196317	0.041347
2018-01-04	-1.042981	0.651530	1.060125	0.064832
2018-01-05	-1.392614	-2.570905	-0.600063	-0.015025

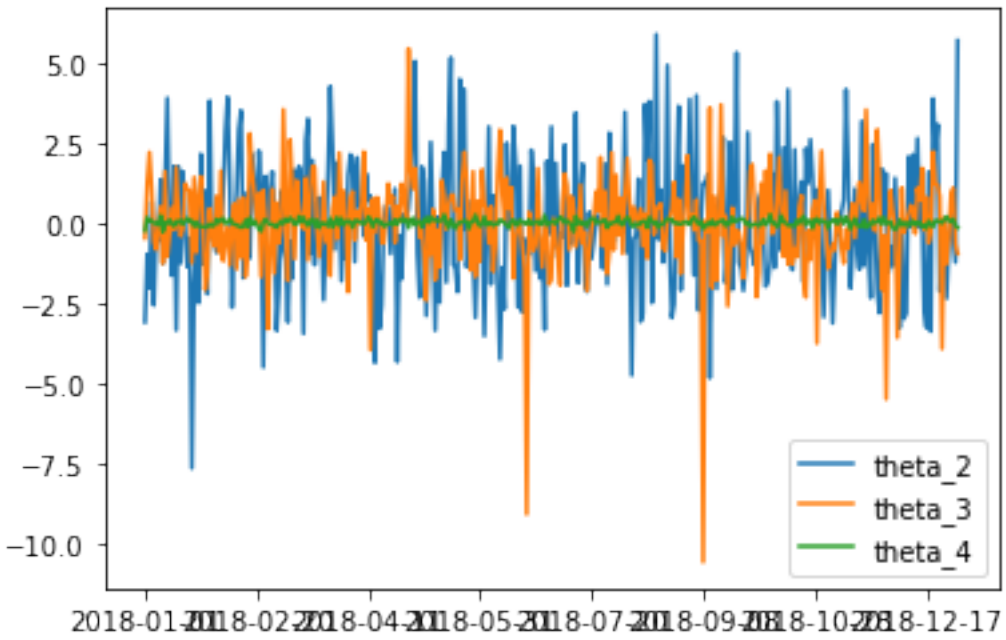
[9]:
```
for c in df2.columns[1:]:
    df2[c].plot()
    plt.title(c)
    plt.show()
```


theta_2
6
4
2
0
−2
−4
−6
−8
2018-01-
2018-02-
2018-04-
2018-05-
2018-07-
2018-09-
2018-10-
2018-12-17


theta_3
6
4
2
0
−2
−4
−6
−8
−10
2018-01-
2018-02-
2018-04-
2018-05-
2018-07-
2018-09-
2018-10-
2018-12-17

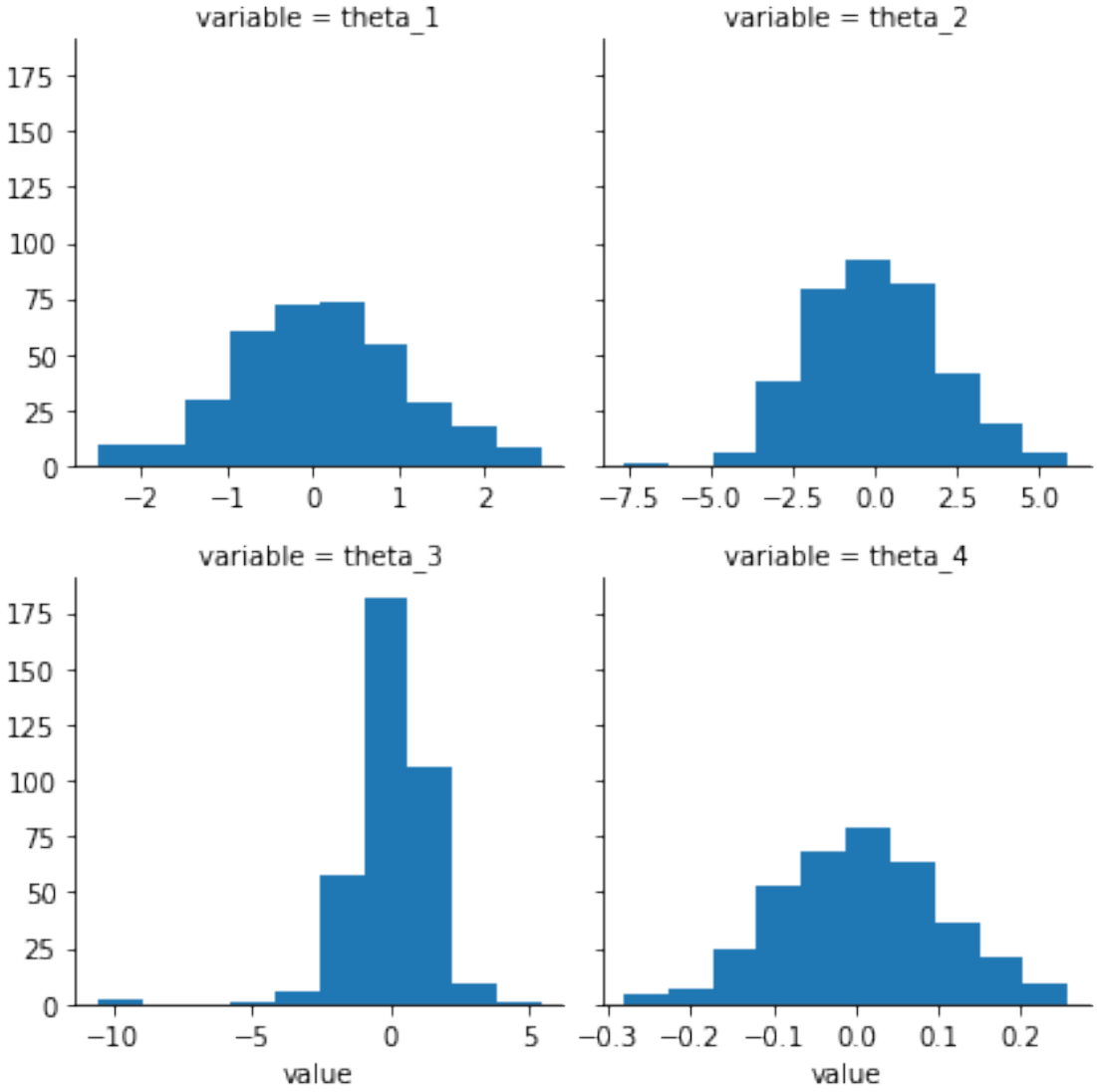


[10]:
```
for c in df2.columns[1:]:
    df2[c].plot()
plt.legend(df.columns[1:])
plt.show()
```

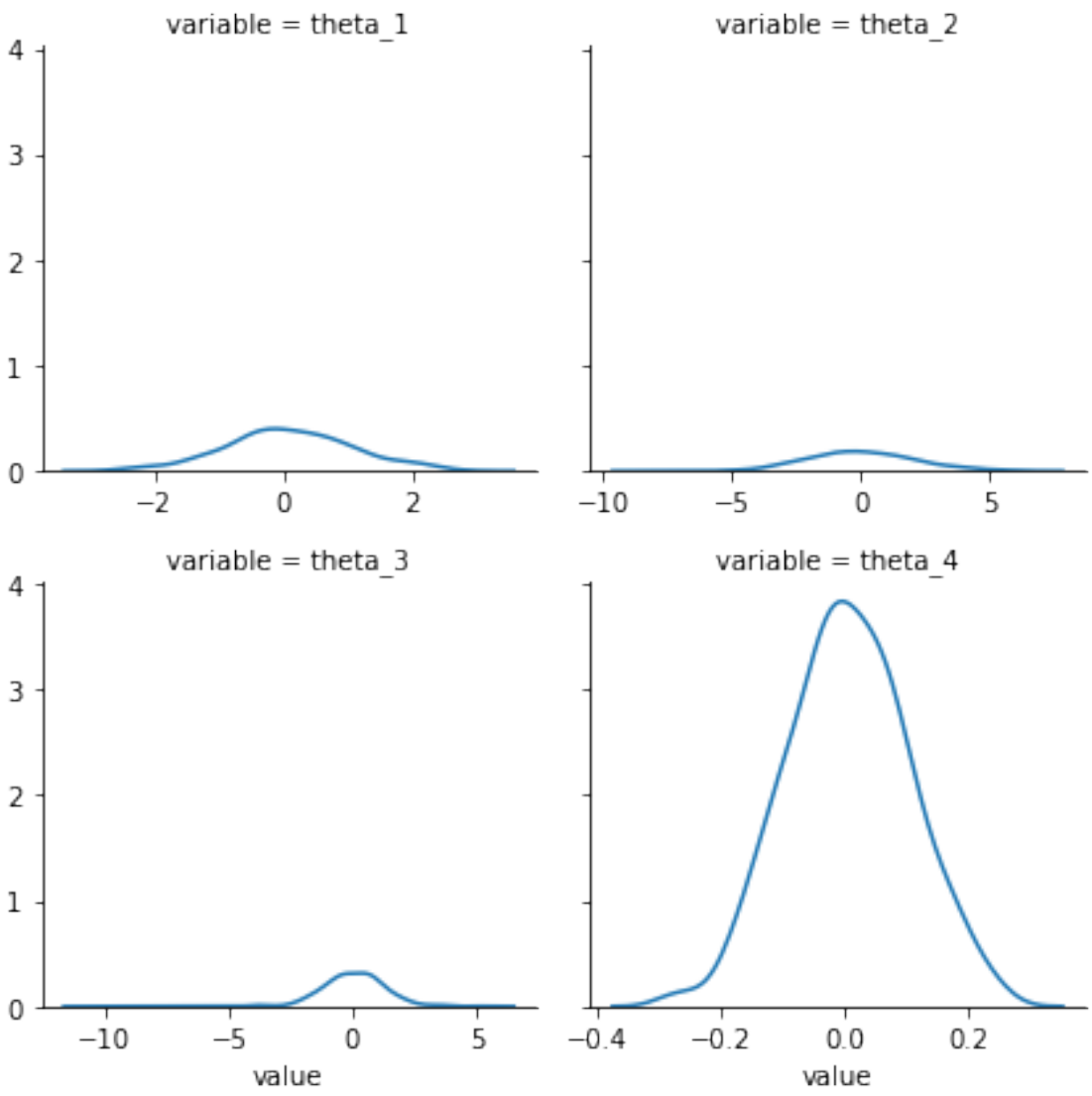


[11]:
```
df2m = df2.melt()
```

[12]:
```
g = sns.FacetGrid(df2m, col='variable', col_wrap=2, sharex = False)
g = g.map(plt.hist, 'value')
```



[13]:
```
f = sns.FacetGrid(df2m, col='variable', col_wrap=2, sharex = False)
f = f.map(sns.kdeplot, 'value')
```


variable = theta_1
variable = theta_2
variable = theta_3
variable = theta_4
value
value