ai4water datasets

Ather Abbas

Feb 03, 2023

SCRIPTS

CHAPTER

ONE

SCRIPTS

Scripts describing datasets from ai4water

1.1 beach water quality

```
from ai4water.eda import EDA
from ai4water.datasets import busan_beach
from ai4water.utils.utils import get_version_info

# sphinx_gallery_thumbnail_number = 7

for k,v in get_version_info().items():
    print(f"{k} version: {v}")
```

```
/home/docs/checkouts/readthedocs.org/user_builds/ai4water-datasets/envs/latest/lib/
↪python3.7/site-packages/sklearn/experimental/enable_hist_gradient_boosting.py:17:␣
↪UserWarning: Since version 1.0, it is not needed to import enable_hist_gradient_
↪boosting anymore. HistGradientBoostingClassifier and HistGradientBoostingRegressor are␣
↪now stable and can be normally imported from sklearn.ensemble.
  "Since version 1.0, "

**********Tensorflow models could not be imported **********

python version: 3.7.9 (default, Oct 19 2020, 15:13:17)
[GCC 7.5.0]
os version: posix
ai4water version: 1.06
easy_mpl version: 0.21.2
SeqMetrics version: 1.3.4
numpy version: 1.21.6
pandas version: 1.2.3
matplotlib version: 3.5.3
joblib version: 1.2.0
```

```
data = busan_beach(target=['ecoli', 'sul1_coppml', 'aac_coppml',
                           'tetx_coppml', 'blaTEM_coppml'])
print(data.shape)
```

```
(1446, 18)
```

```
data.head()
```

```
data.isna().sum()
```

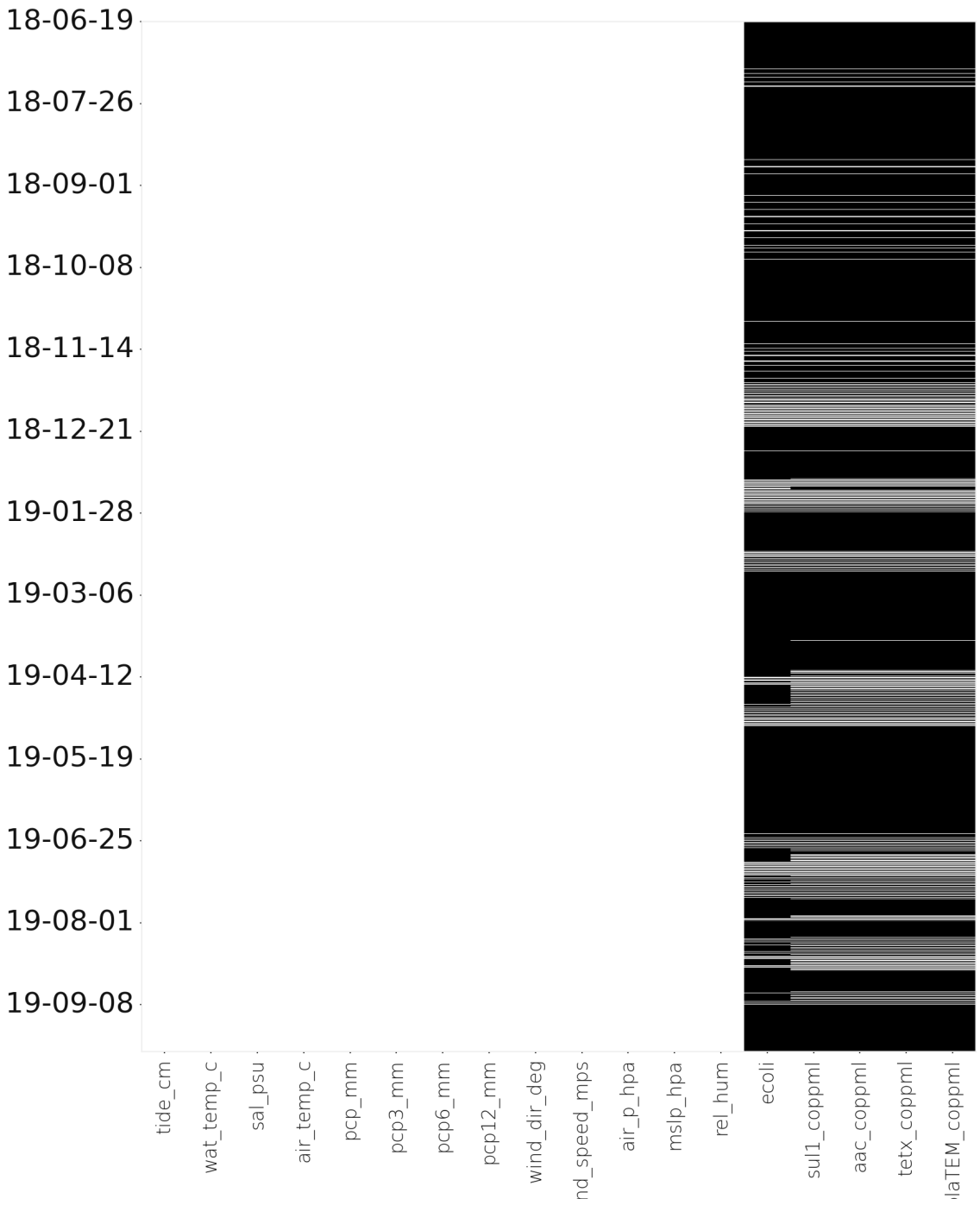
```
tide_cm              0
wat_temp_c           0
sal_psu              0
air_temp_c           0
pcp_mm               0
pcp3_mm              0
pcp6_mm              0
pcp12_mm             0
wind_dir_deg         0
wind_speed_mps       0
air_p_hpa            0
mslp_hpa             0
rel_hum              0
ecoli             1279
sul1_coppml       1228
aac_coppml        1228
tetx_coppml       1228
blaTEM_coppml     1228
dtype: int64
```

```
data.isna().sum()
```

```
tide_cm              0
wat_temp_c           0
sal_psu              0
air_temp_c           0
pcp_mm               0
pcp3_mm              0
pcp6_mm              0
pcp12_mm             0
wind_dir_deg         0
wind_speed_mps       0
air_p_hpa            0
mslp_hpa             0
rel_hum              0
ecoli             1279
sul1_coppml       1228
aac_coppml        1228
tetx_coppml       1228
blaTEM_coppml     1228
dtype: int64
```

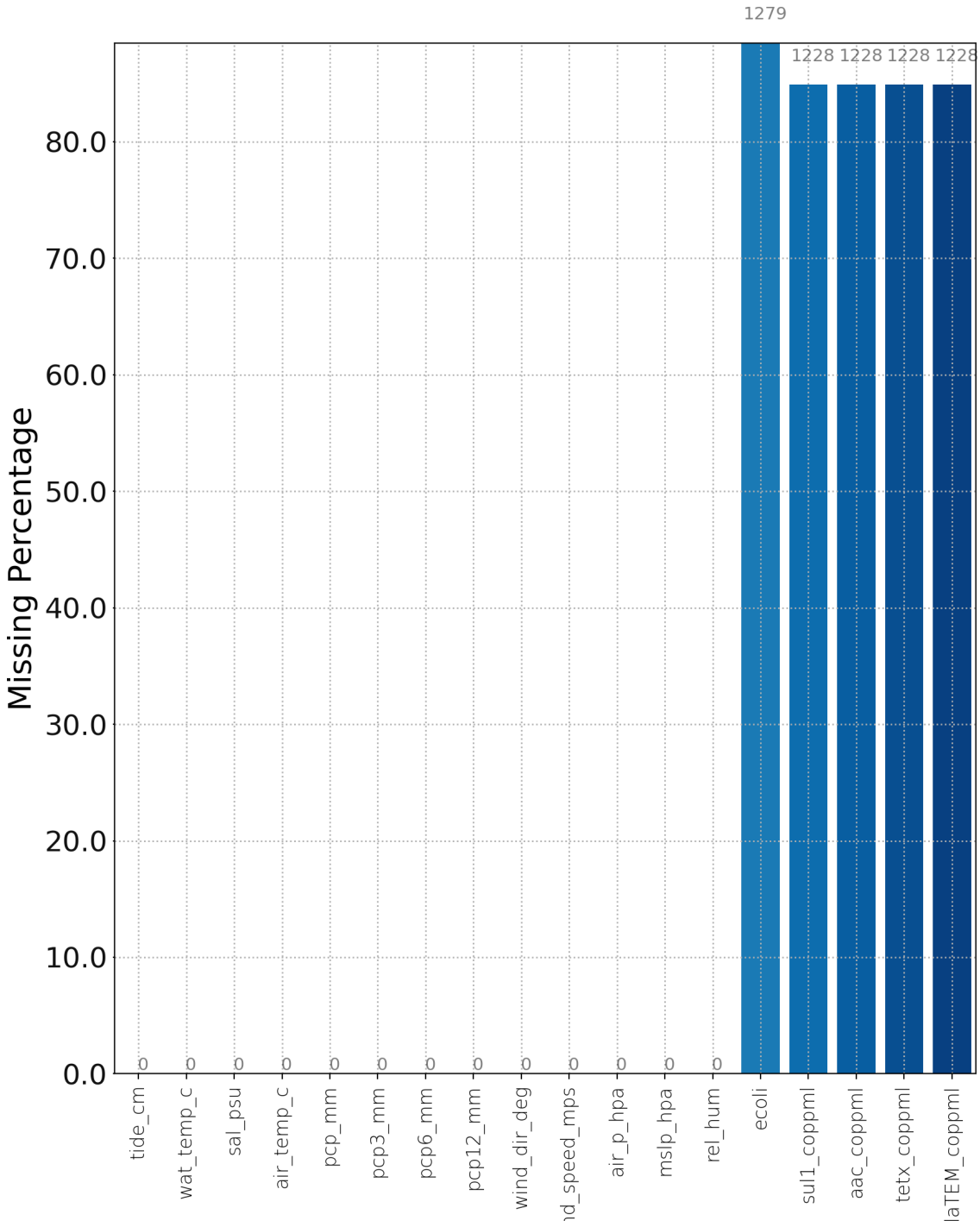
```
eda = EDA(data, save=False)
```

```
eda.heatmap()
```




```
<AxesSubplot:ylabel='Examples'>
```

```
_ = eda.plot_missing()
```



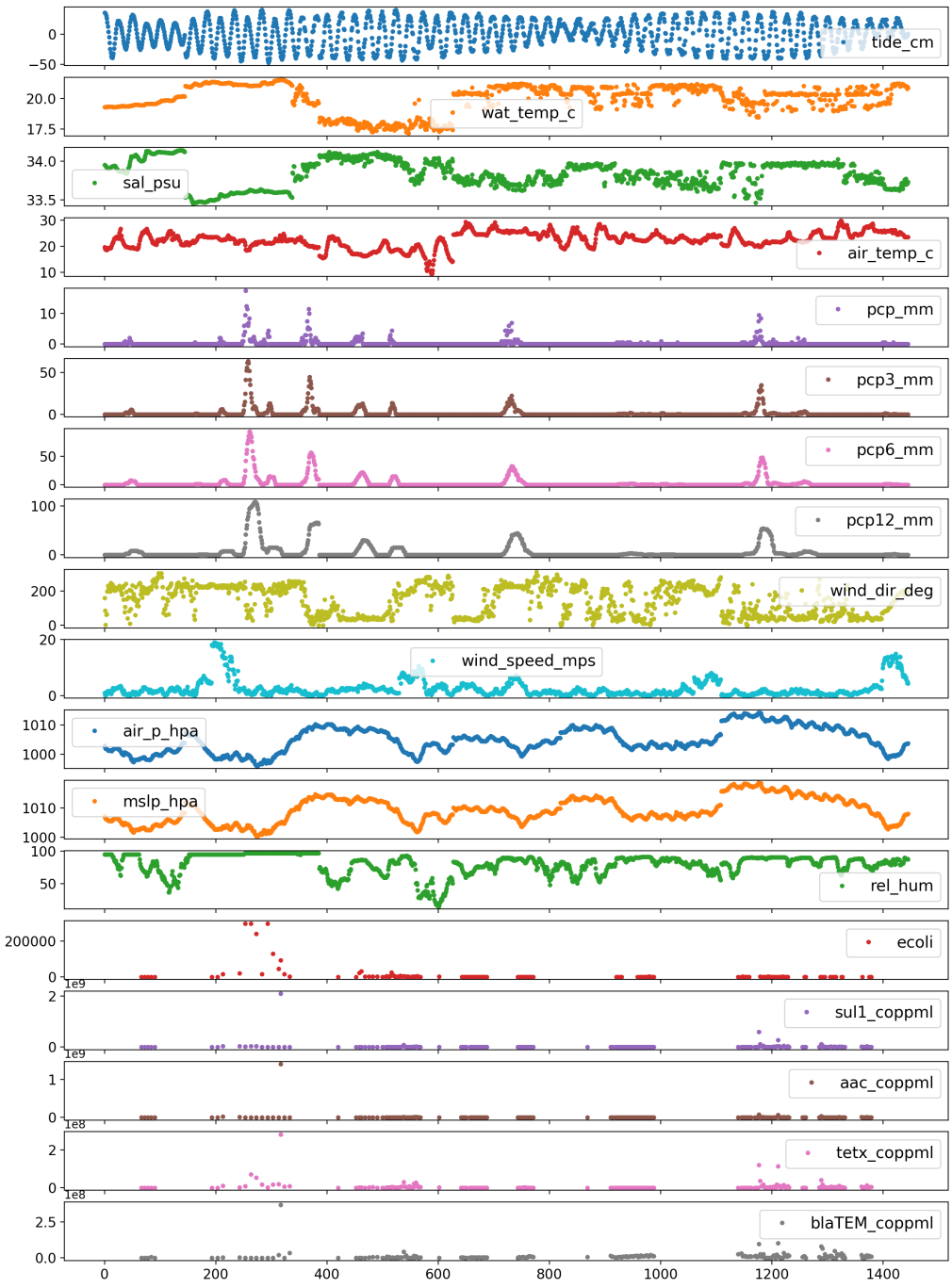

```
/home/docs/checkouts/readthedocs.org/user_builds/ai4water-datasets/envs/latest/lib/
```

(continues on next page)

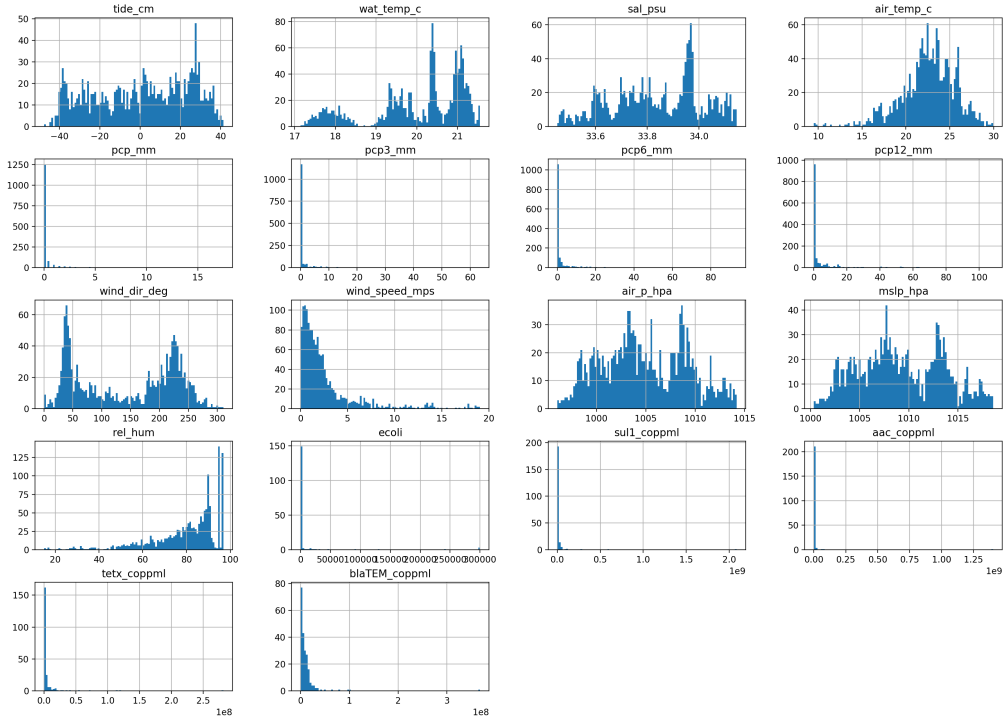
(continued from previous page)

```
↪python3.7/site-packages/ai4water/eda/_main.py:377: UserWarning: FixedFormatter should␣
↪only be used together with FixedLocator
  ax1.set_yticklabels(ax1.get_yticks(), fontsize="18")
```

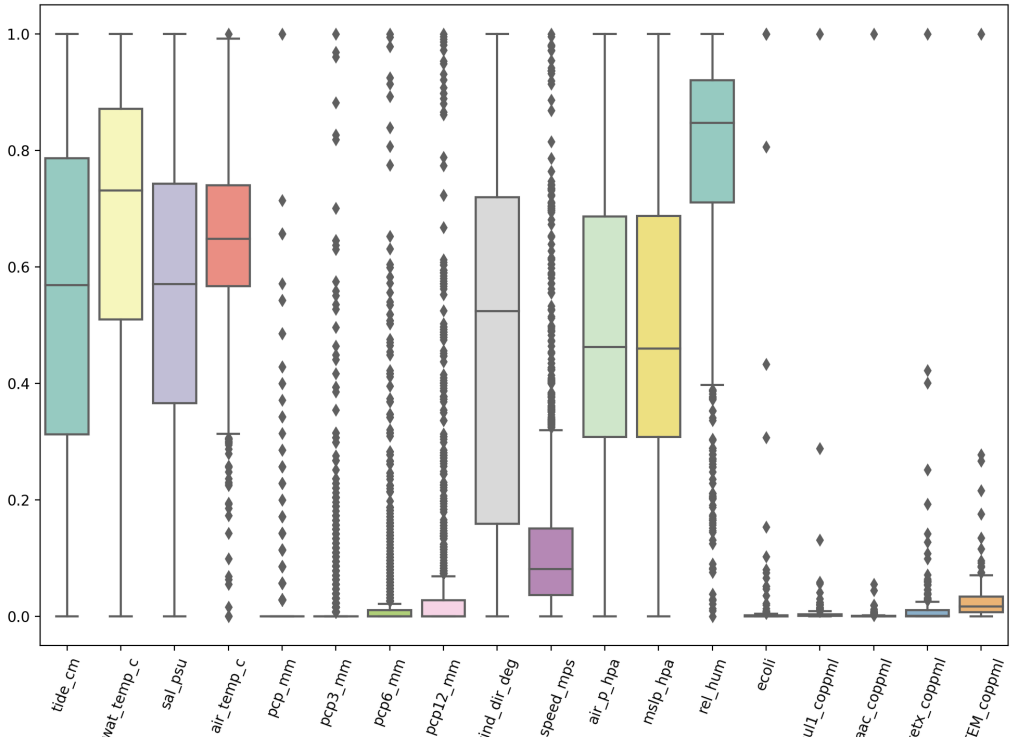
```
# _ = eda.plot_data(subplots=True, max_cols_in_plot=20, figsize=(14, 20))
#
# ############################################################

eda.plot_data(subplots=True, max_cols_in_plot=20, figsize=(14, 20),
              ignore_datetime_index=True)
```


tide_cm
wat_temp_c
sal_psu
air_temp_c
pcp_mm
pcp3_mm
pcp6_mm
pcp12_mm
wind_dir_deg
wind_speed_mps
air_p_hpa
mslp_hpa
rel_hum
ecoli
sul1_coppml
aac_coppml
tetx_coppml
blaTEM_coppml
1e9
1e8
0
200
400
600
800
1000
1200
1400

```
_ = eda.plot_histograms()
```

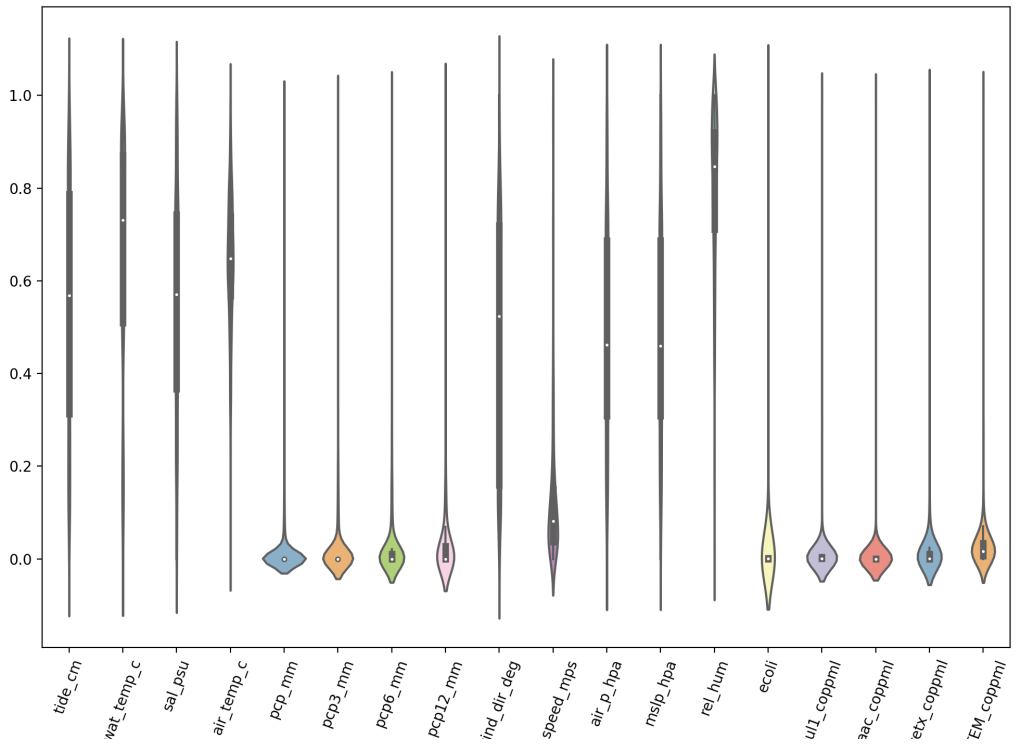



```
eda.box_plot(max_features=18, palette="Set3")
```



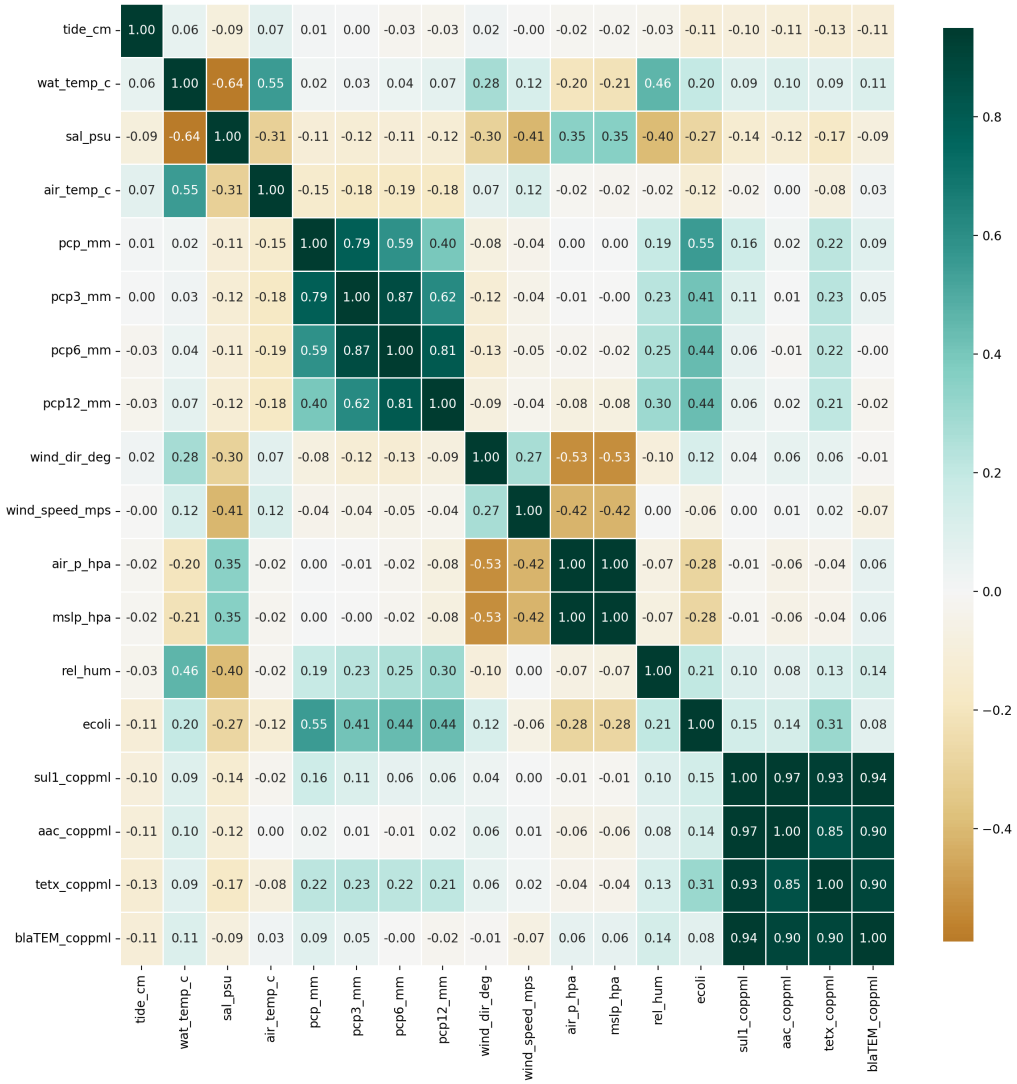

```
<AxesSubplot:>
```

```
eda.box_plot(max_features=18, palette="Set3", violen=True)
```



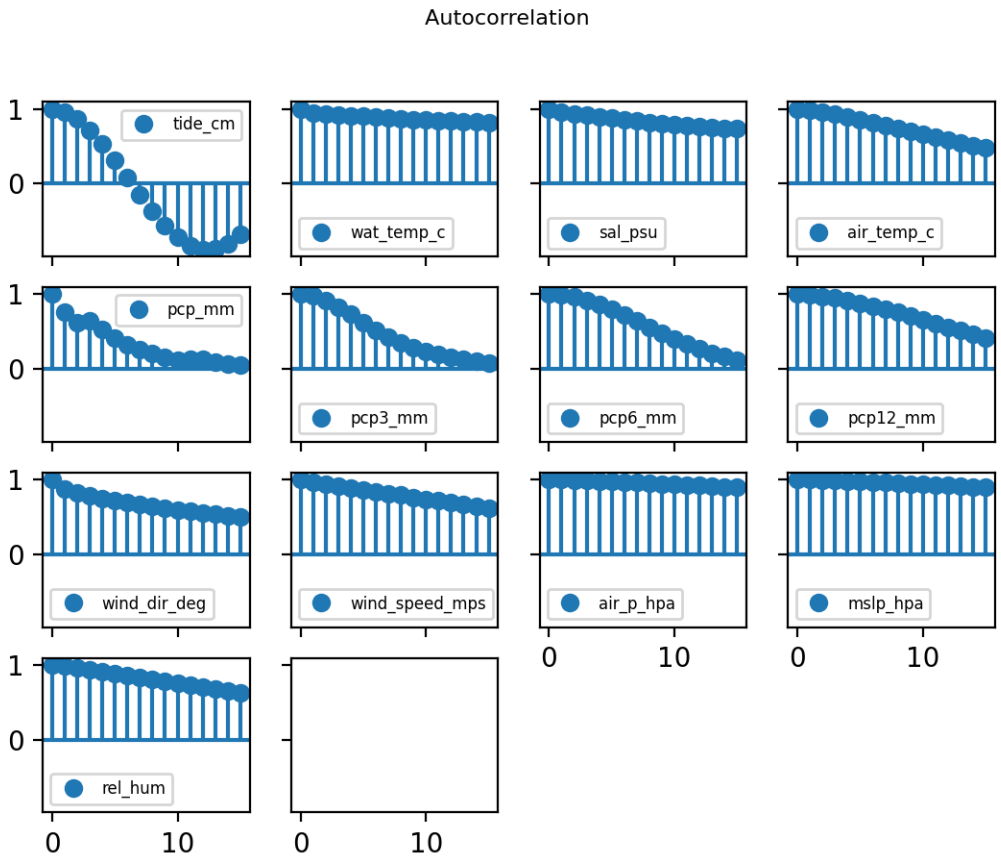

```
<AxesSubplot:>
```

```
eda.correlation(figsize=(14, 14))

# ############################################################
#
#
# eda.grouped_scatter(max_subplots=18)
```



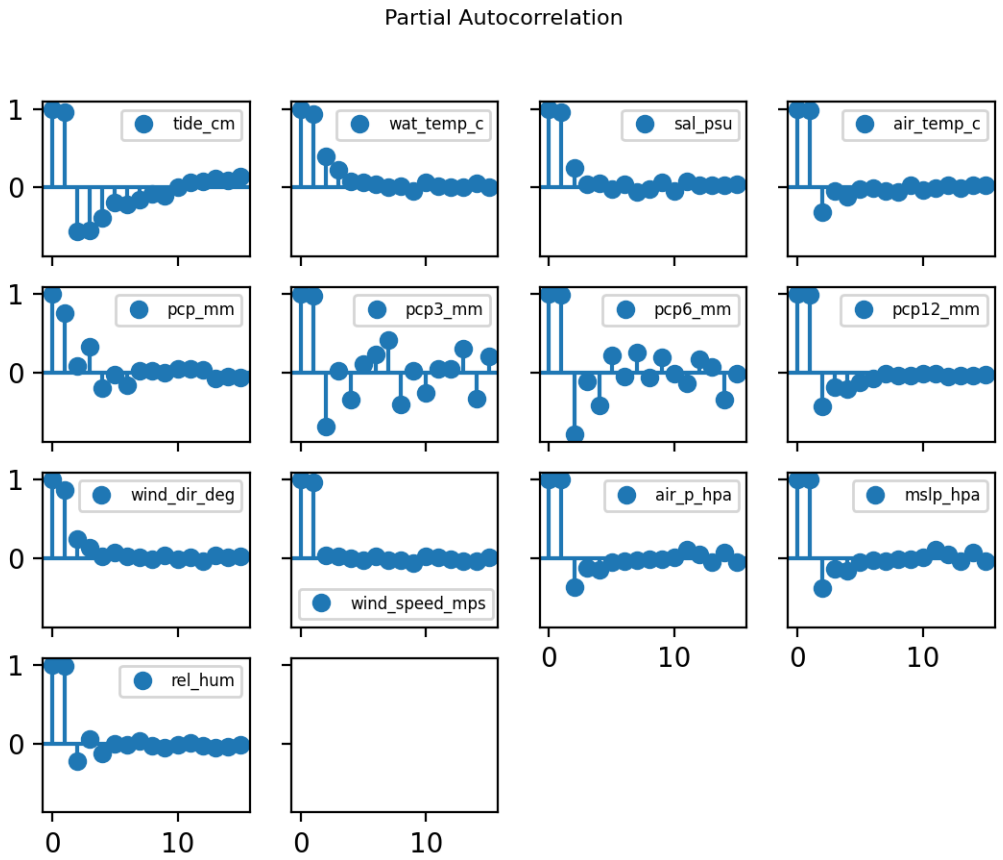

```
<AxesSubplot:>
```

```
_ = eda.autocorrelation(n_lags=15)
```



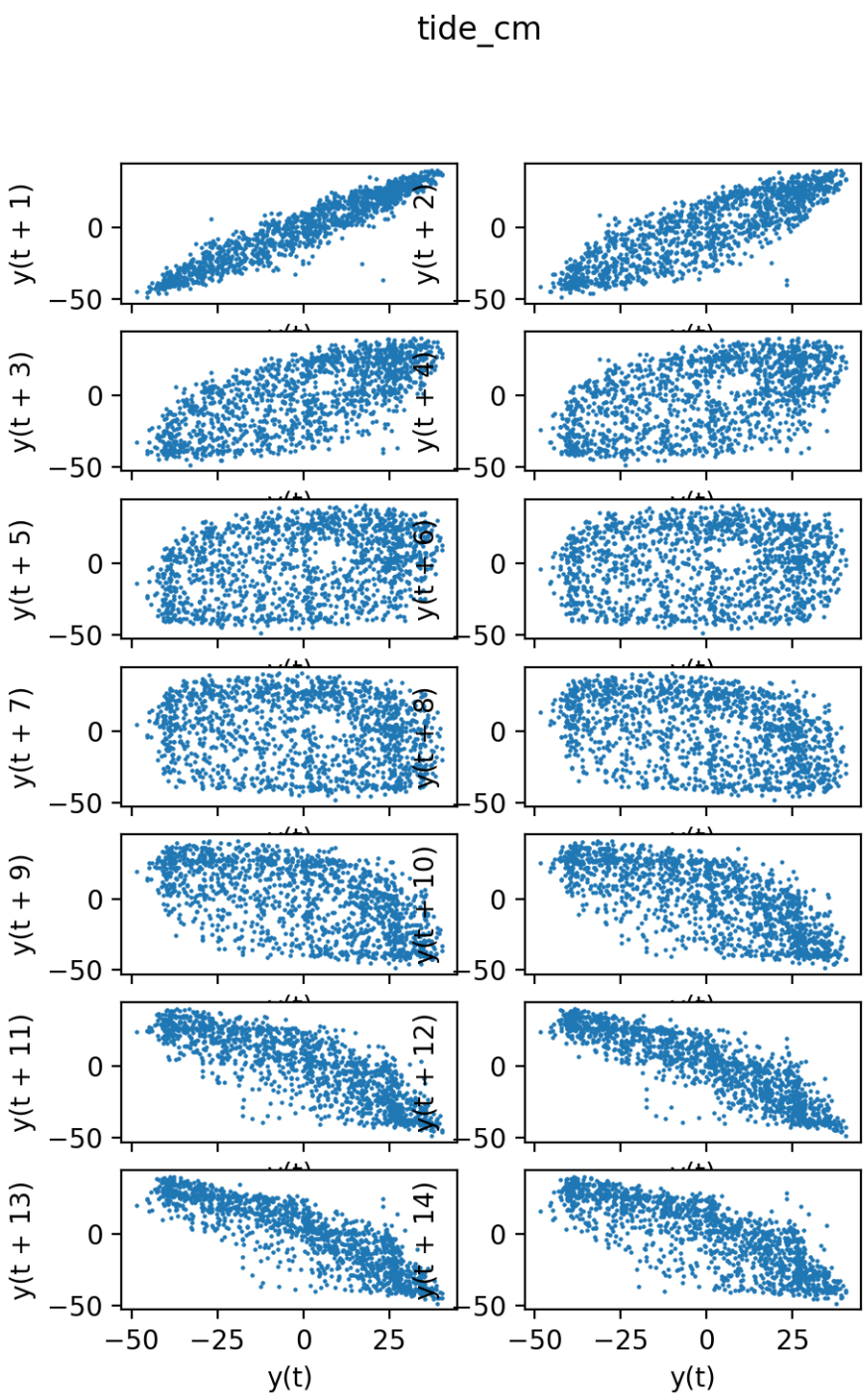

```
cannot plot autocorrelation for ecoli feature
cannot plot autocorrelation for sul1_coppml feature
cannot plot autocorrelation for aac_coppml feature
```

```
_ = eda.partial_autocorrelation(n_lags=15)
```

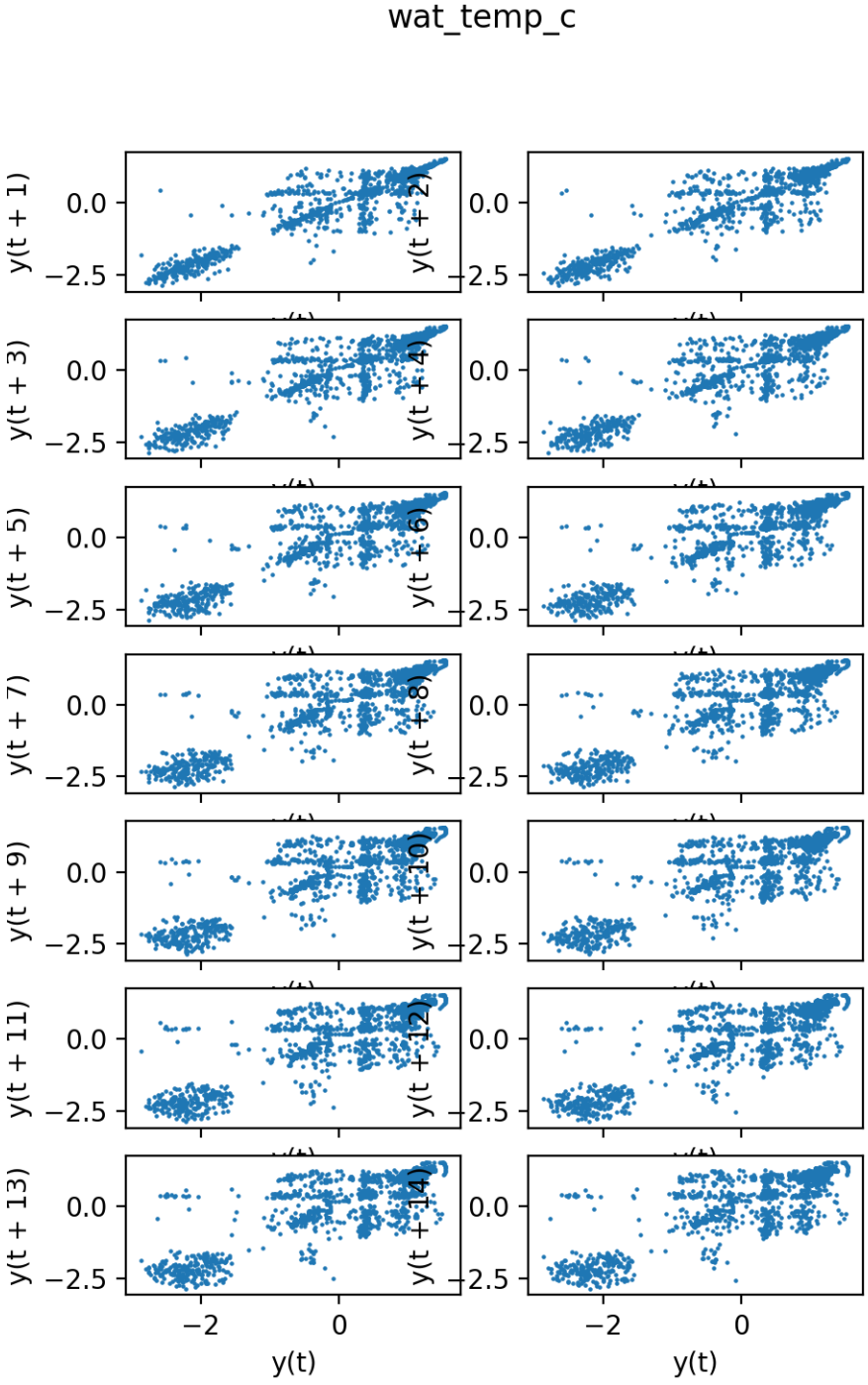


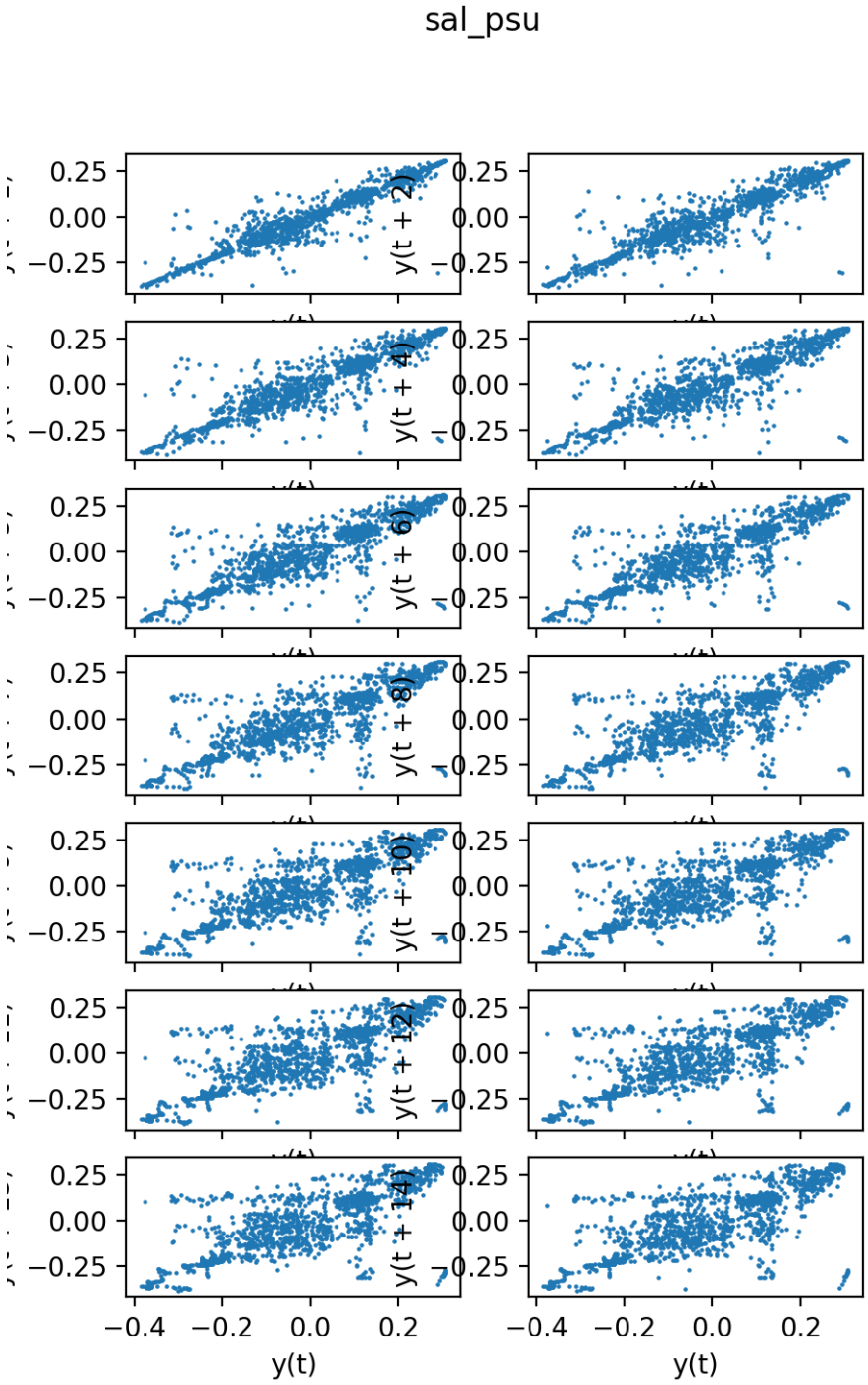

```
cannot plot autocorrelation for ecoli feature
cannot plot autocorrelation for sul1_coppml feature
cannot plot autocorrelation for aac_coppml feature
```

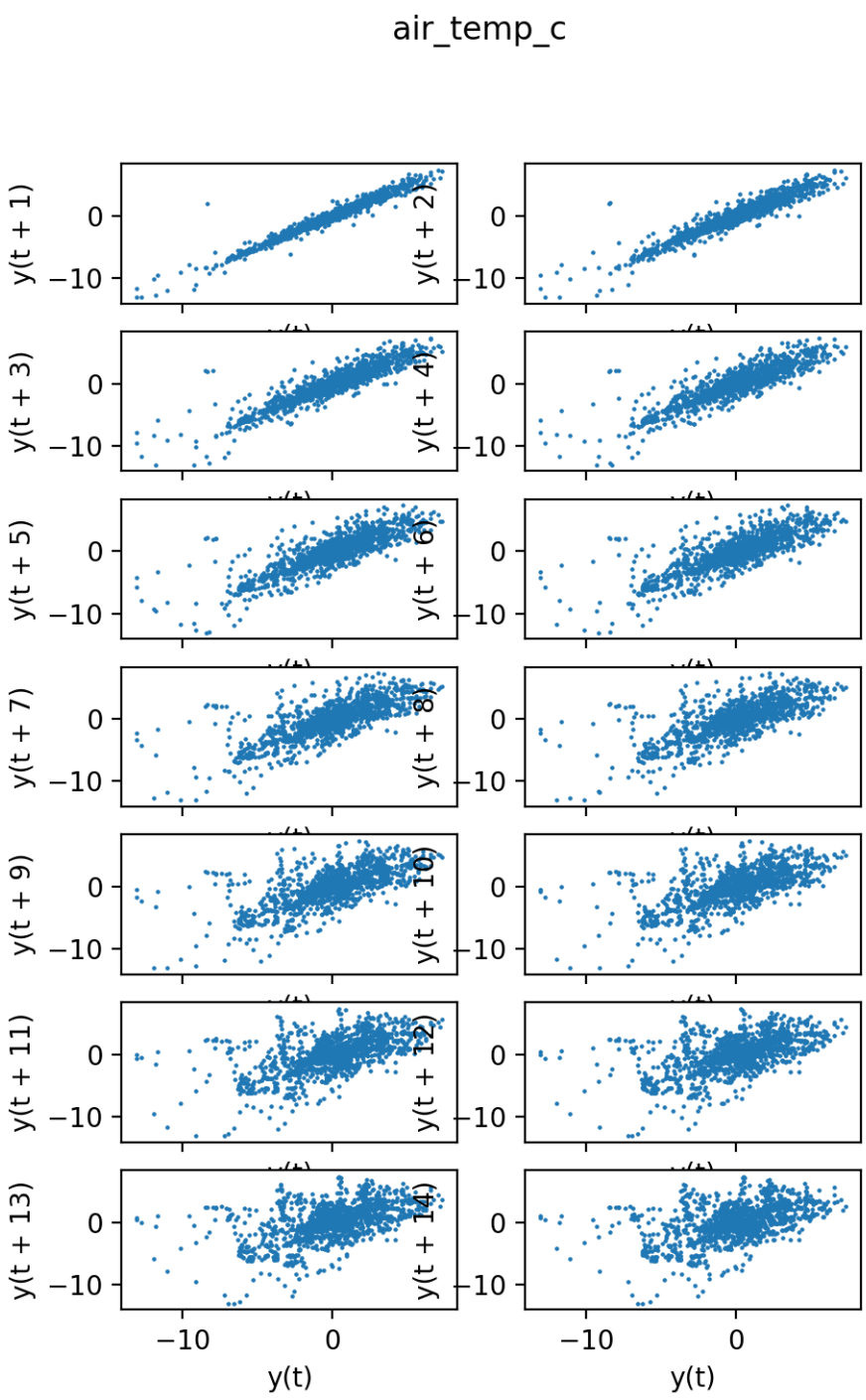
```
_ = eda.lag_plot(n_lags=14, s=0.4)
```

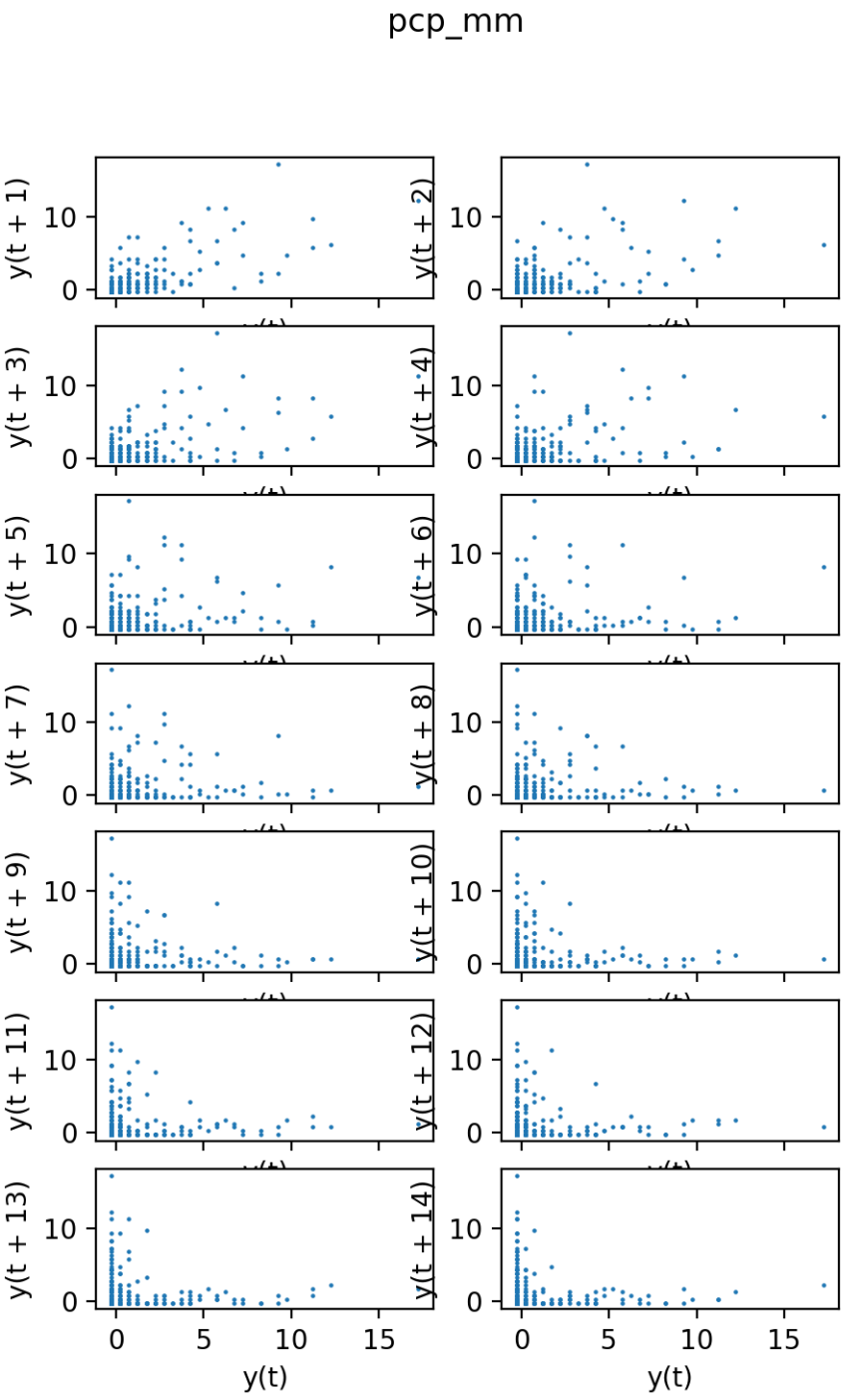

tide_cm
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0
−50
−50
−25
0
25
y(t)
y(t)

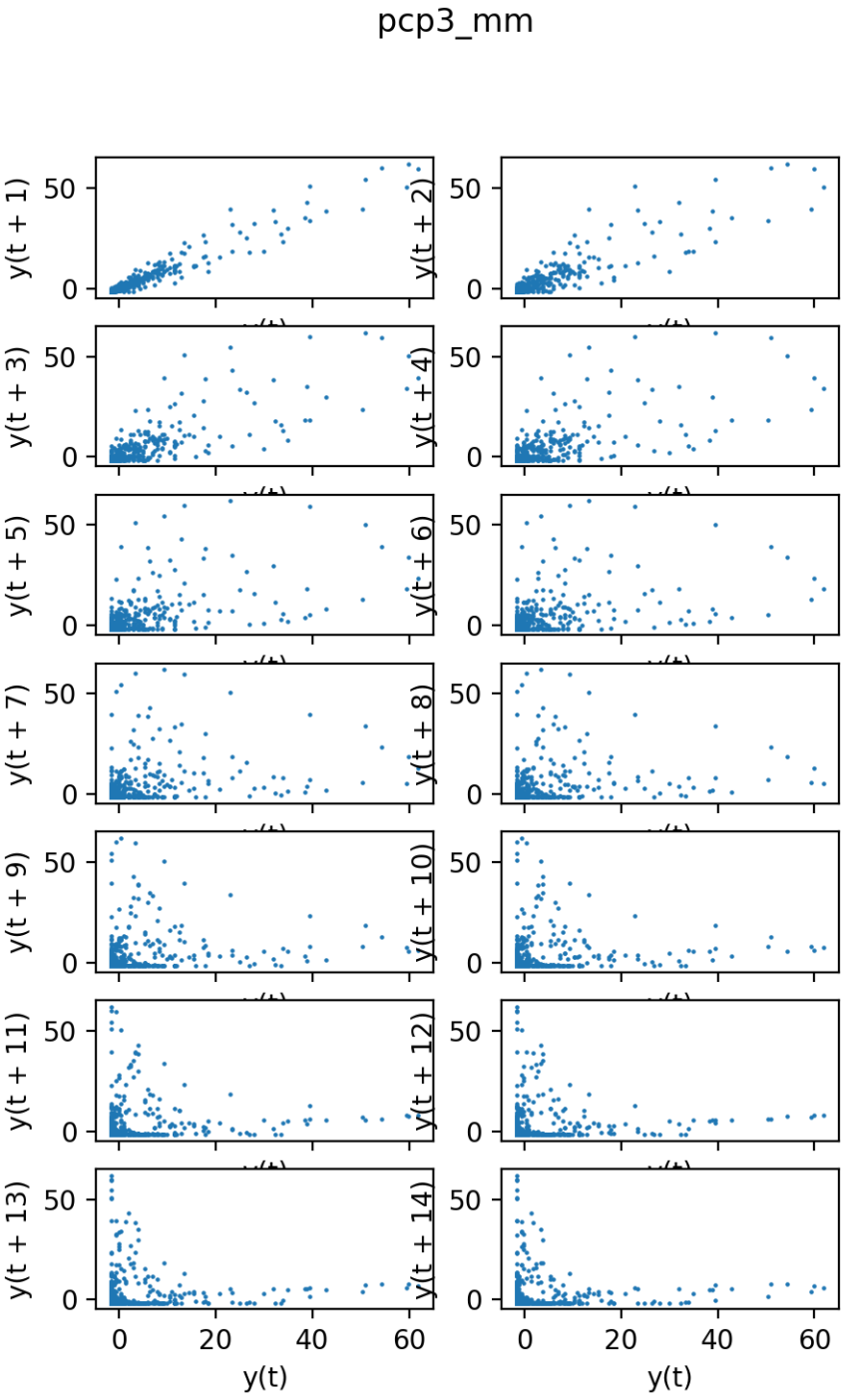
•

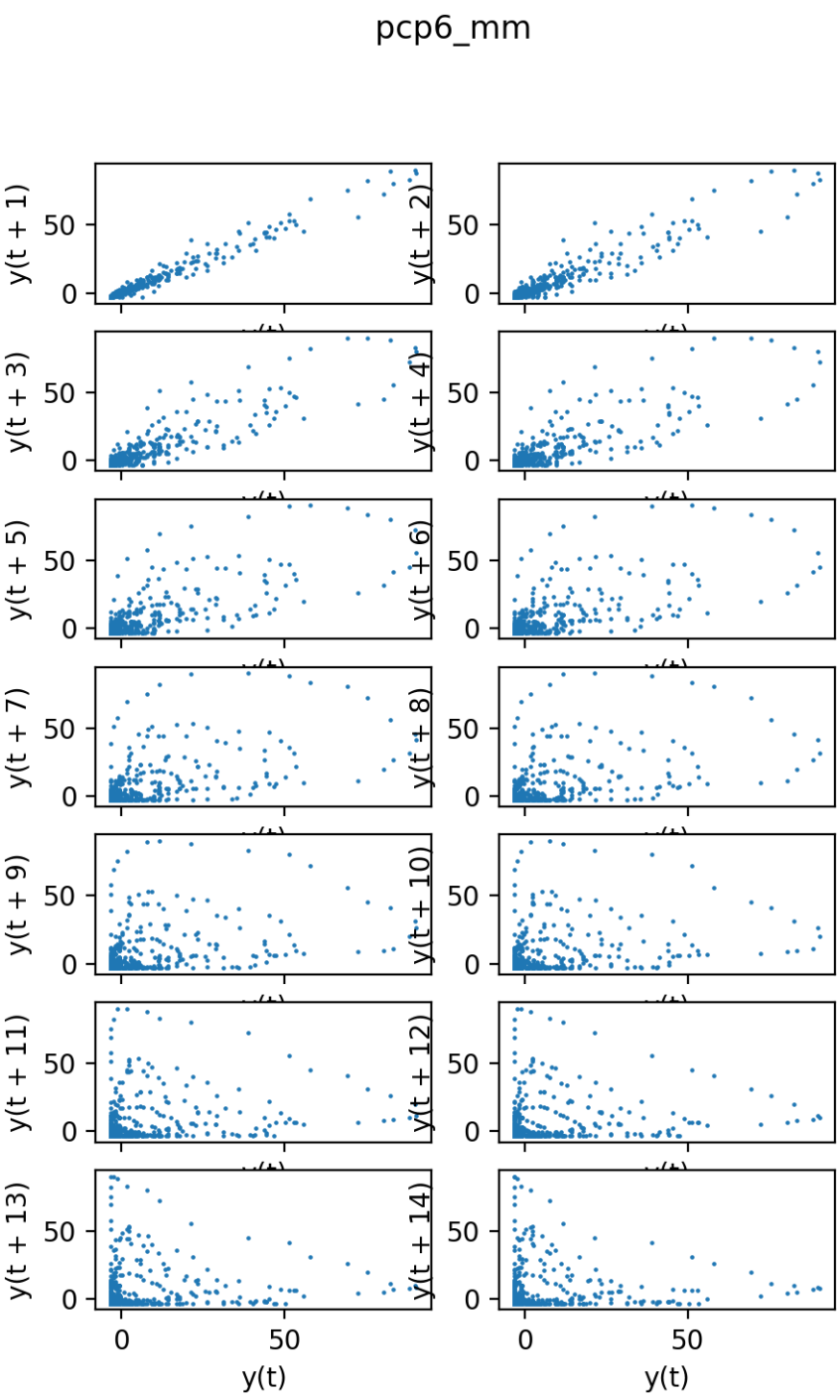

wat_temp_c
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0.0
−2.5
−2
0
y(t)

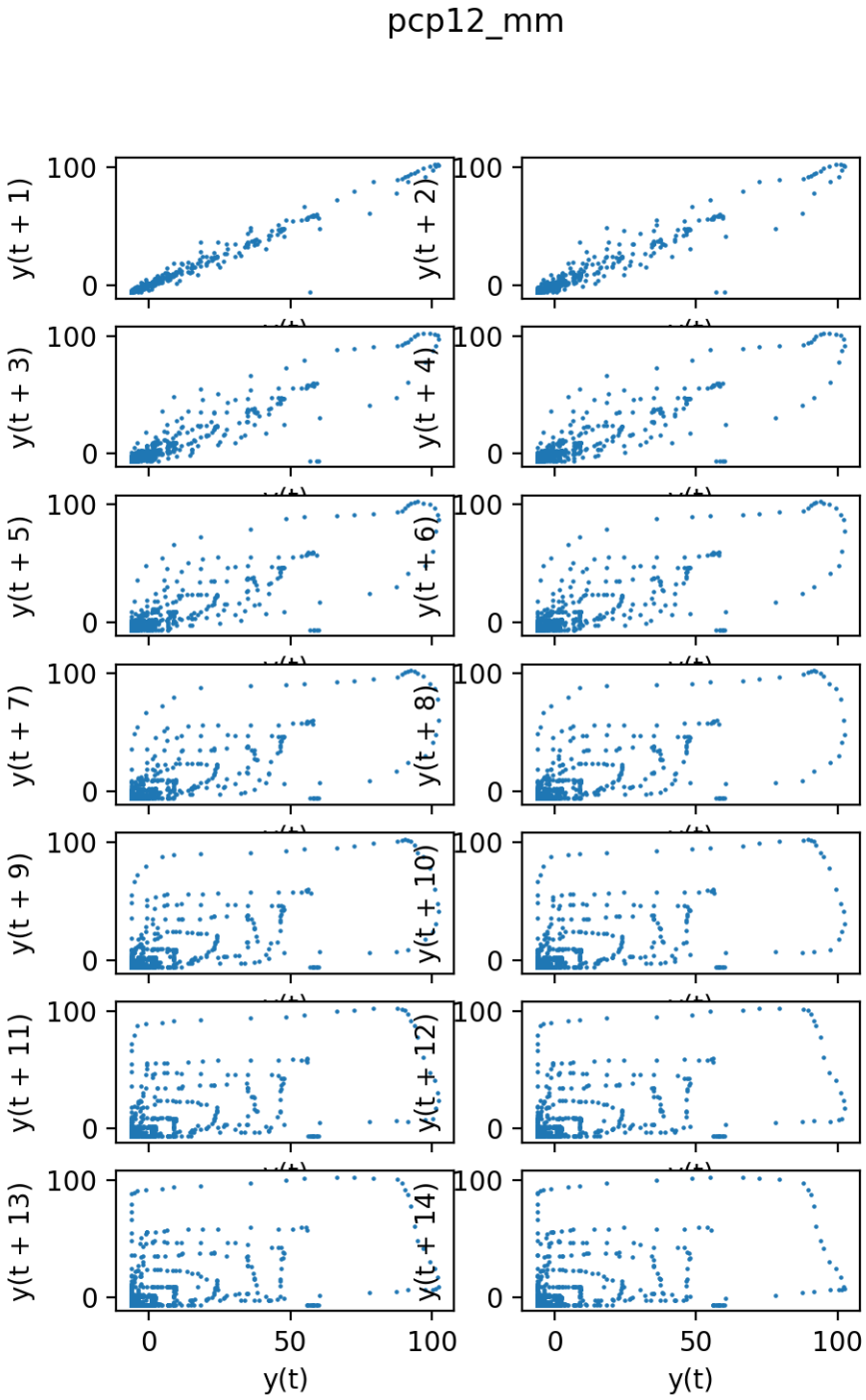

sal_psu
y(t + 2)
y(t + 4)
y(t + 6)
y(t + 8)
y(t + 10)
y(t + 12)
y(t + 14)
0.25
0.00
−0.25
−0.4
−0.2
0.0
0.2
y(t)

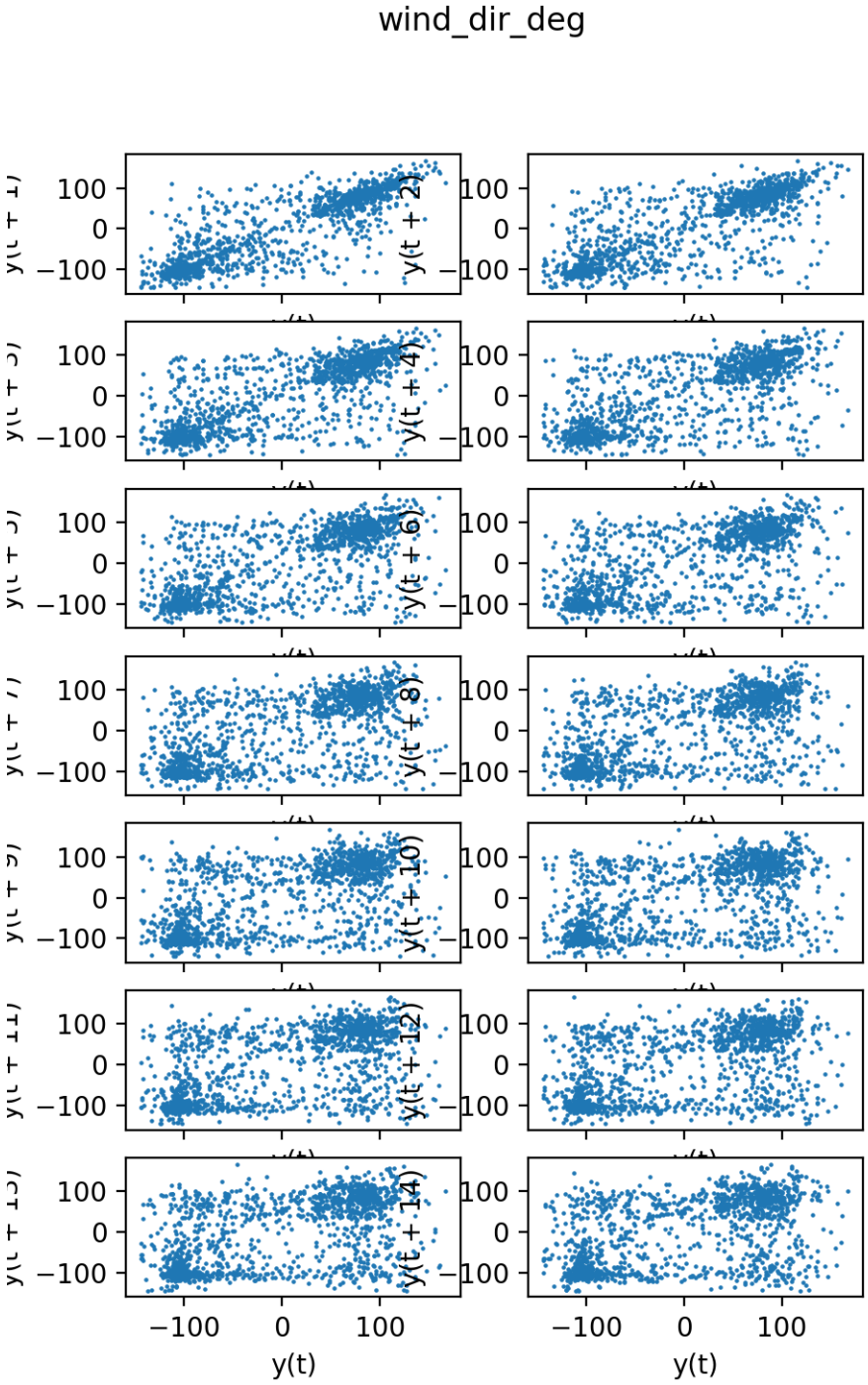

air_temp_c
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0
−10
y(t)

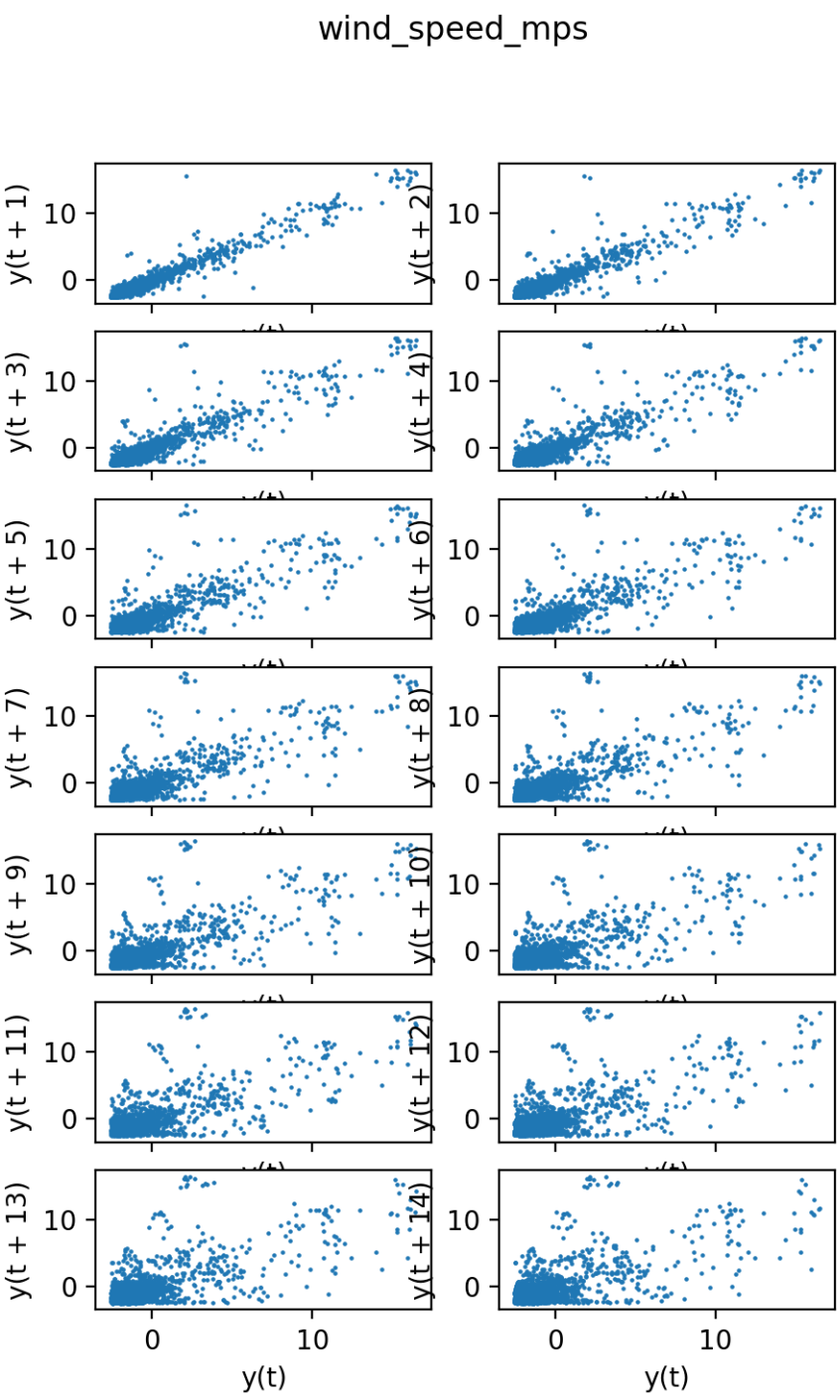

pcp_mm
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0
10
0
5
10
15
y(t)
y(t)

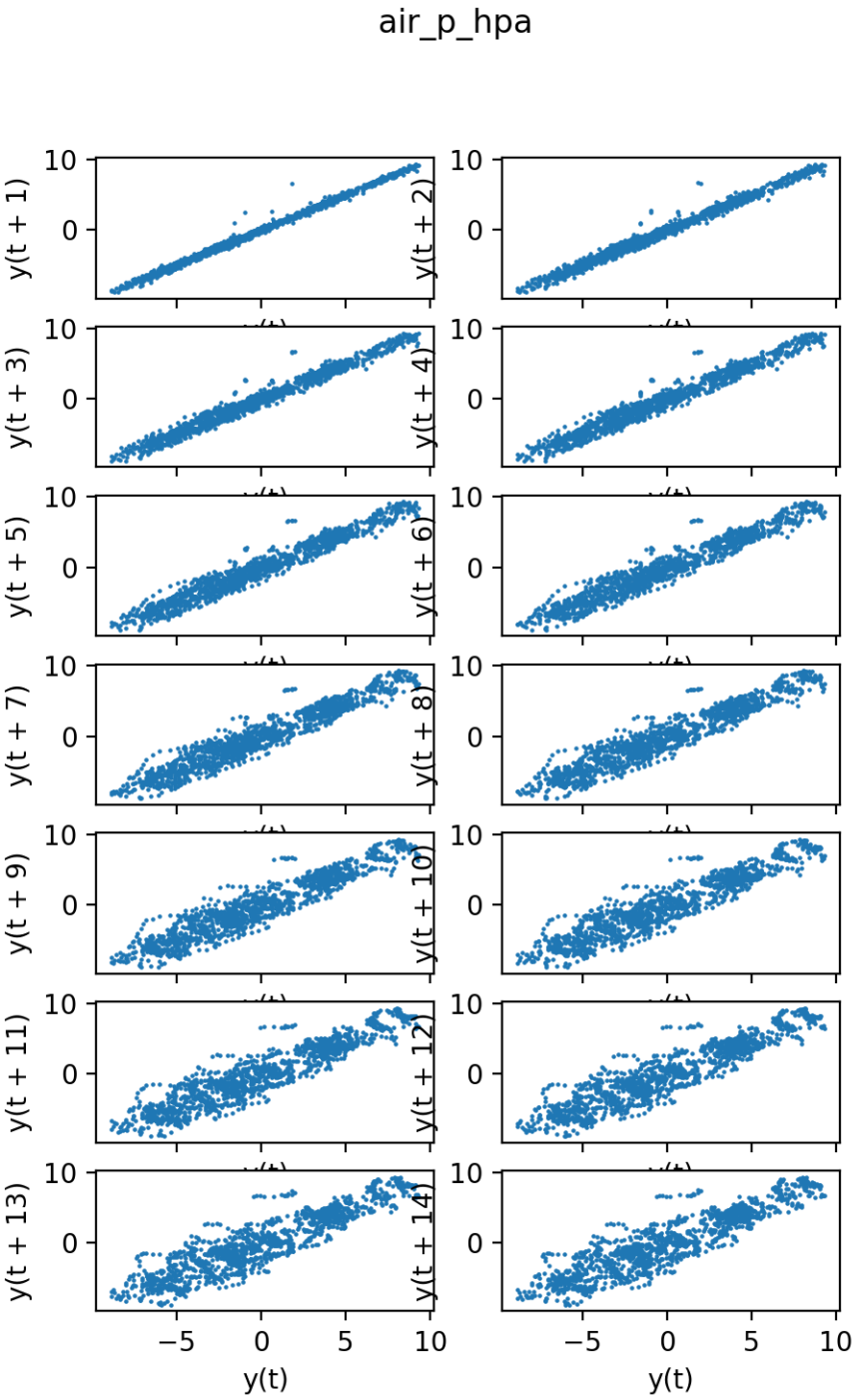

pcp3_mm
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0
50
0
20
40
60
y(t)
y(t)


pcp6_mm
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0
50
y(t)

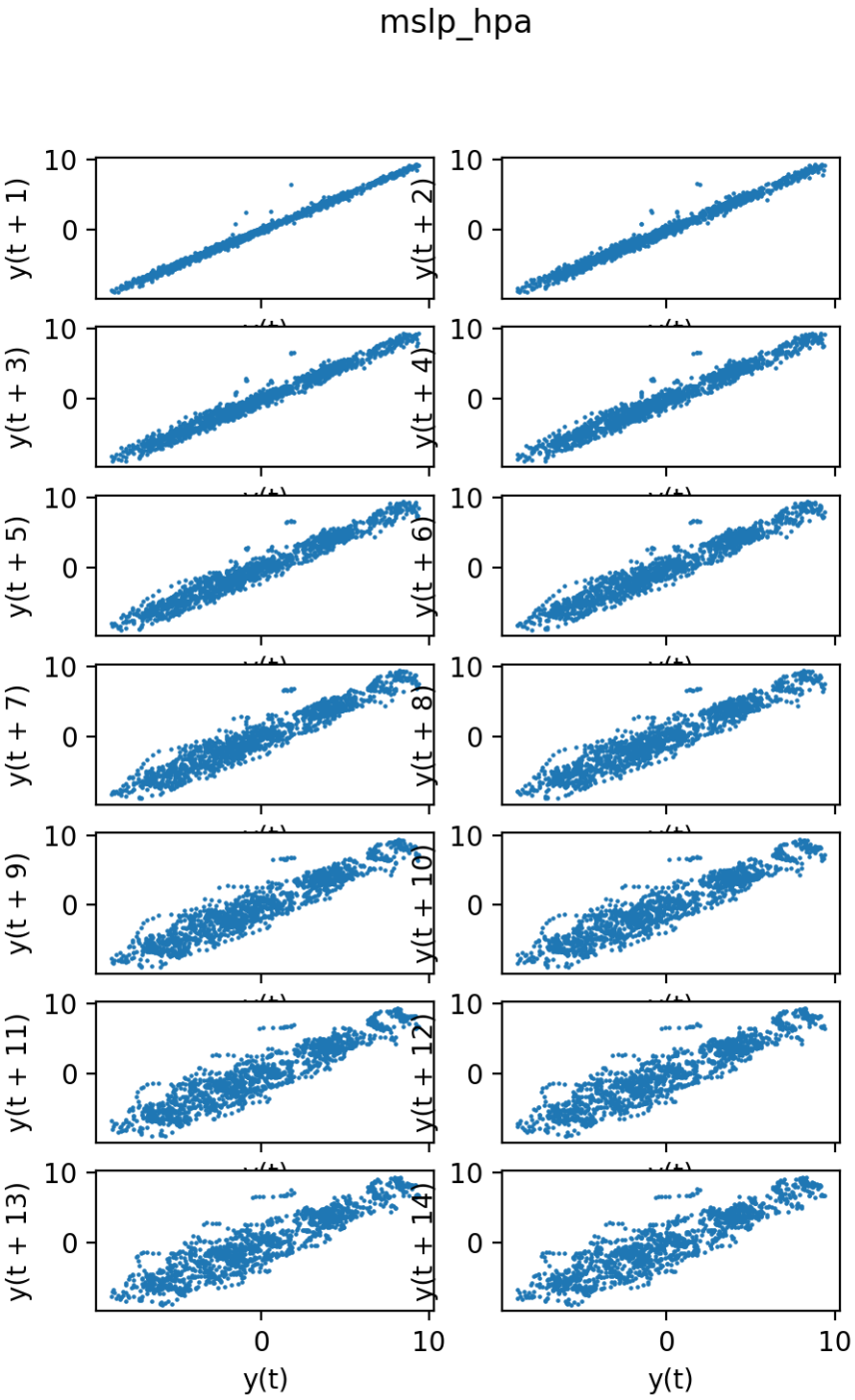

pcp12_mm
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0
50
100
y(t)

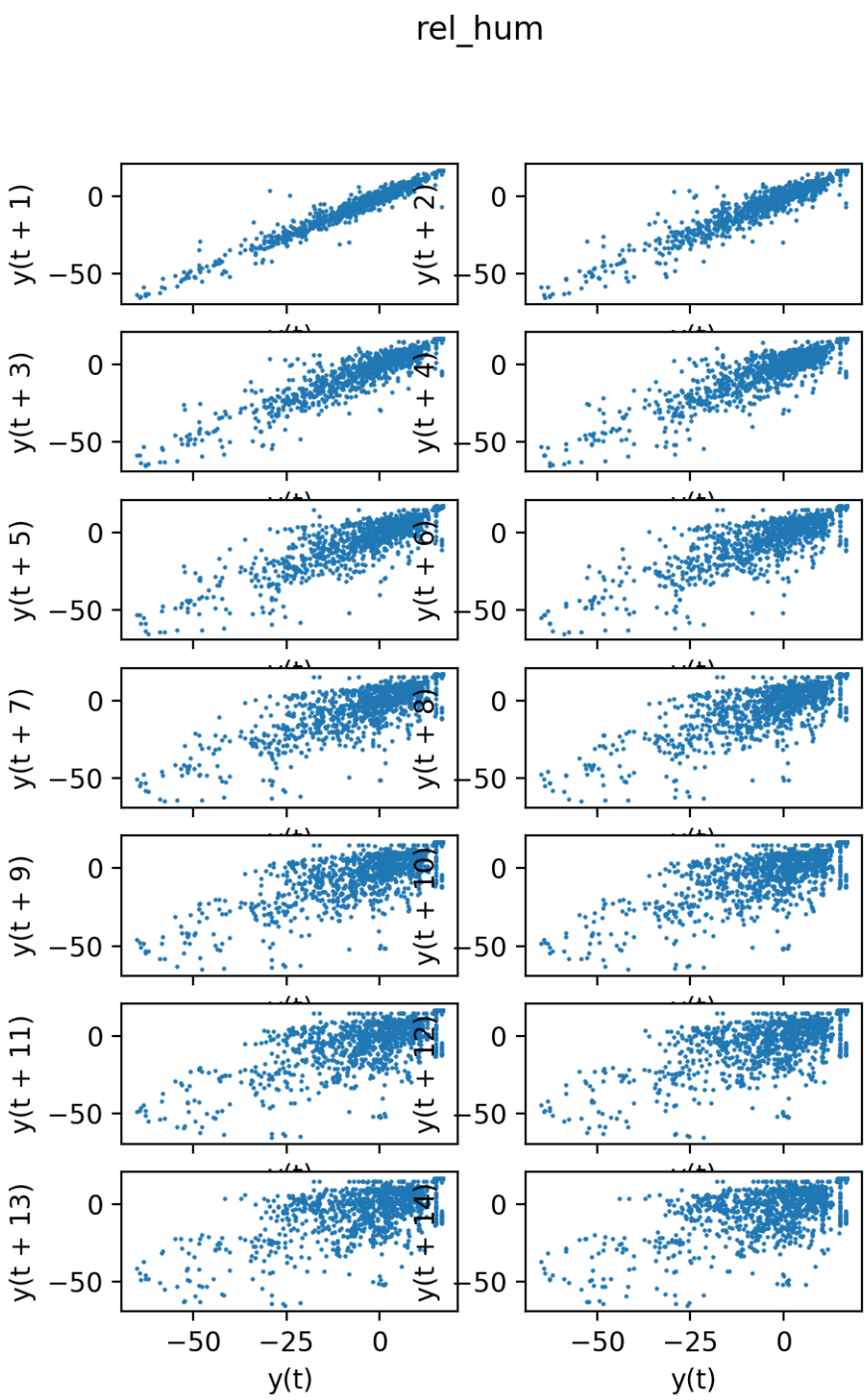

wind_dir_deg
y(t + 21)
y(t + 4)
y(t + 6)
y(t + 8)
y(t + 10)
y(t + 12)
y(t + 14)
100
0
−100
y(t)

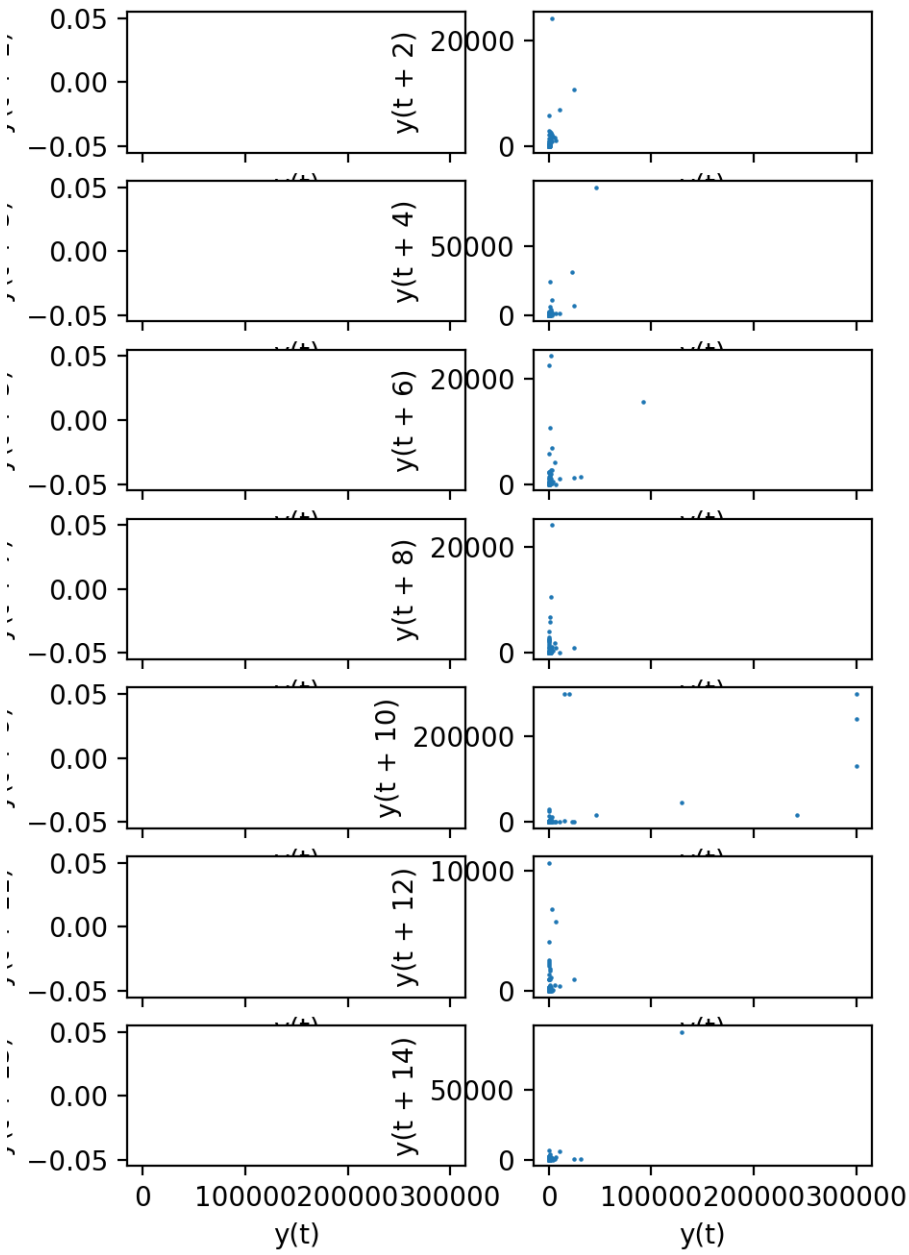

wind_speed_mps
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0
10
y(t)

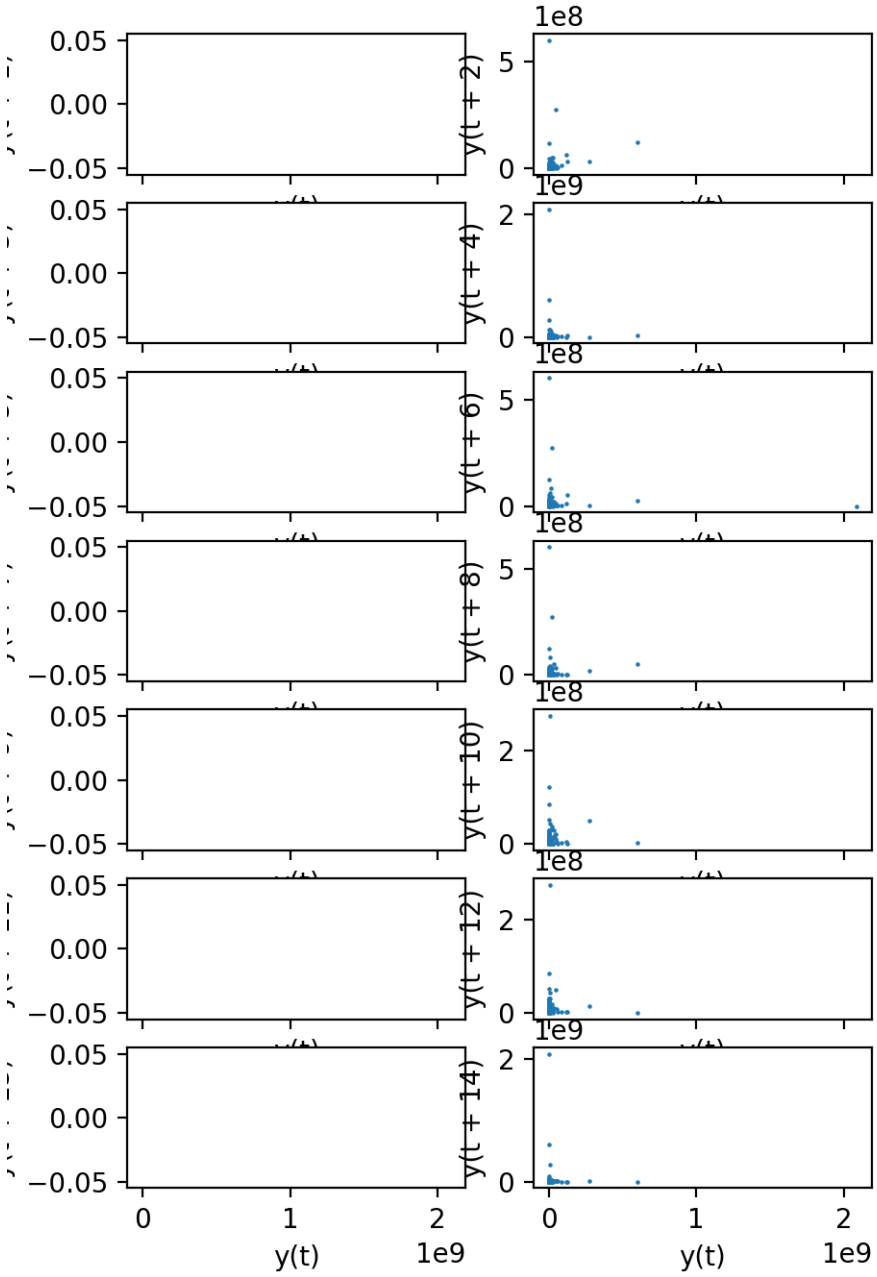

air_p_hpa
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
10
0
−5
0
5
10
y(t)

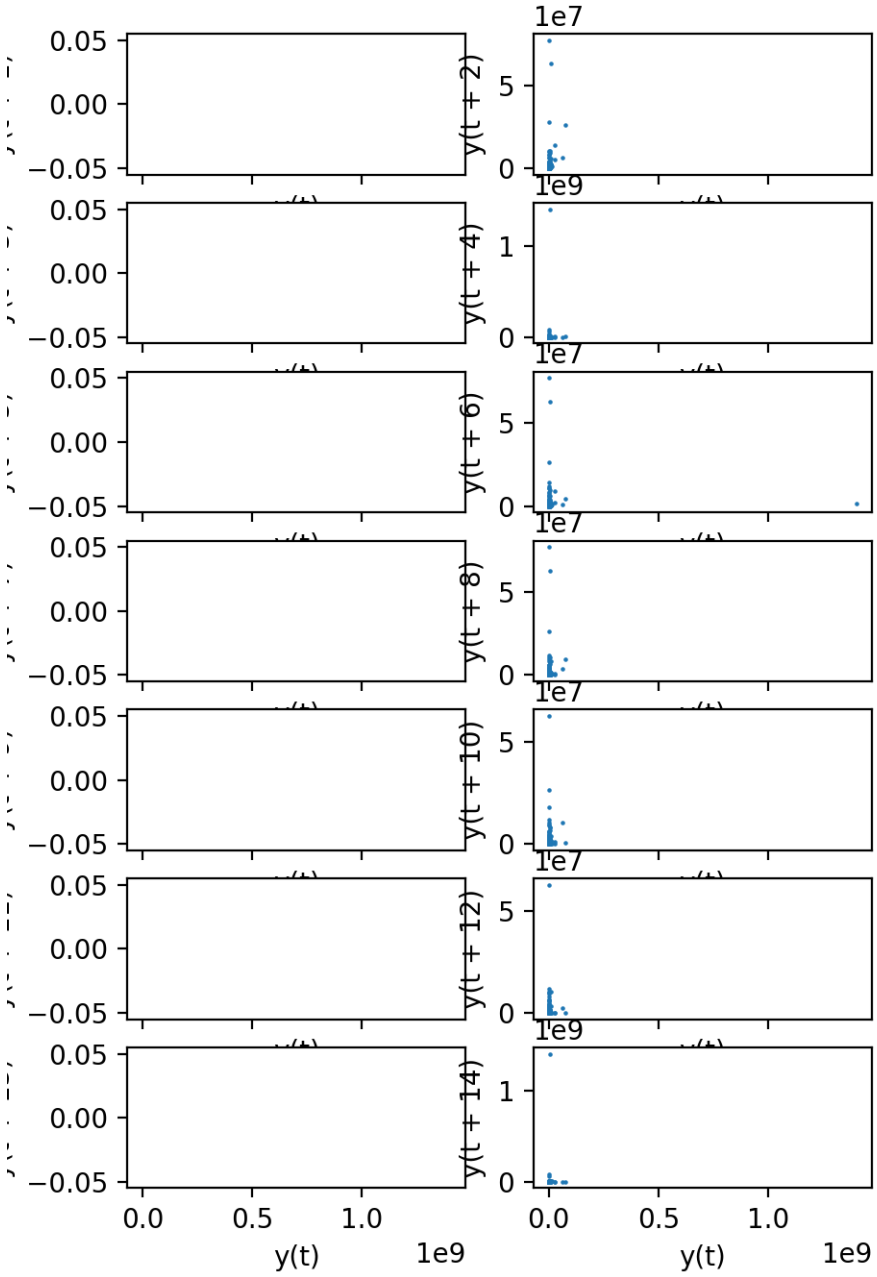
•

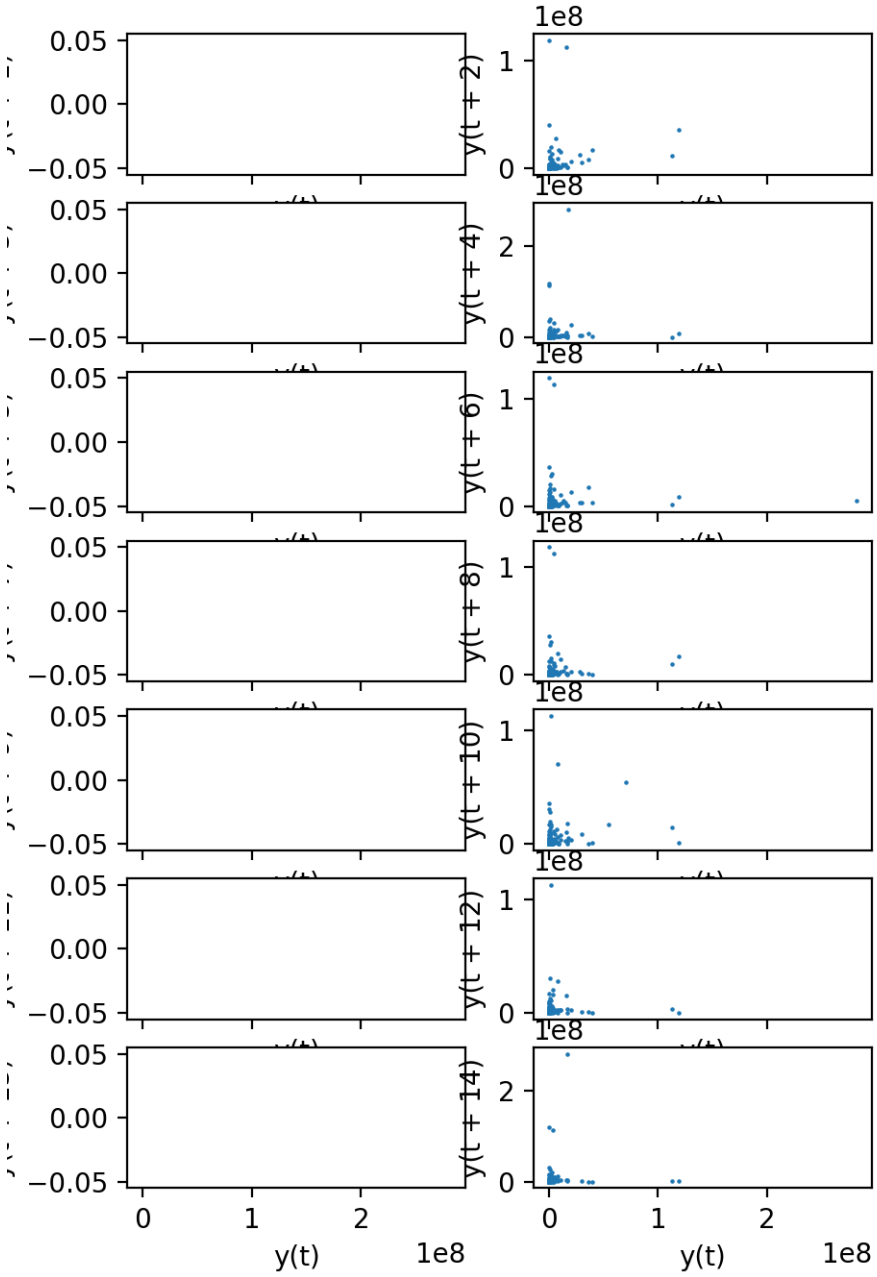

mslp_hpa
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
10
0
y(t)

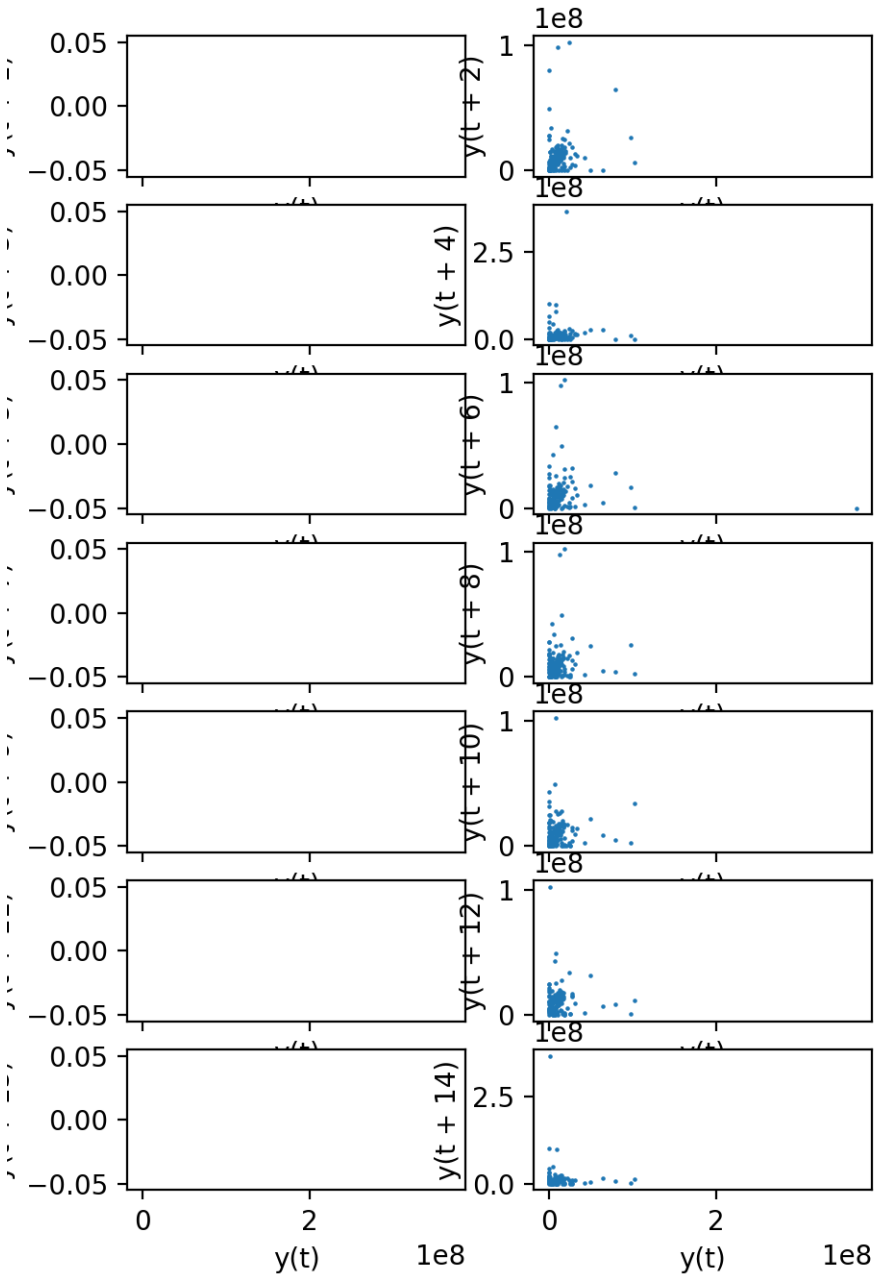

rel_hum
y(t + 1)
y(t + 2)
y(t + 3)
y(t + 4)
y(t + 5)
y(t + 6)
y(t + 7)
y(t + 8)
y(t + 9)
y(t + 10)
y(t + 11)
y(t + 12)
y(t + 13)
y(t + 14)
0
−50
−25
y(t)


ecoli
y(t + 2)
y(t + 4)
y(t + 6)
y(t + 8)
y(t + 10)
y(t + 12)
y(t + 14)
0.05
0.00
−0.05
20000
50000
200000
10000
0
100000200000300000
y(t)

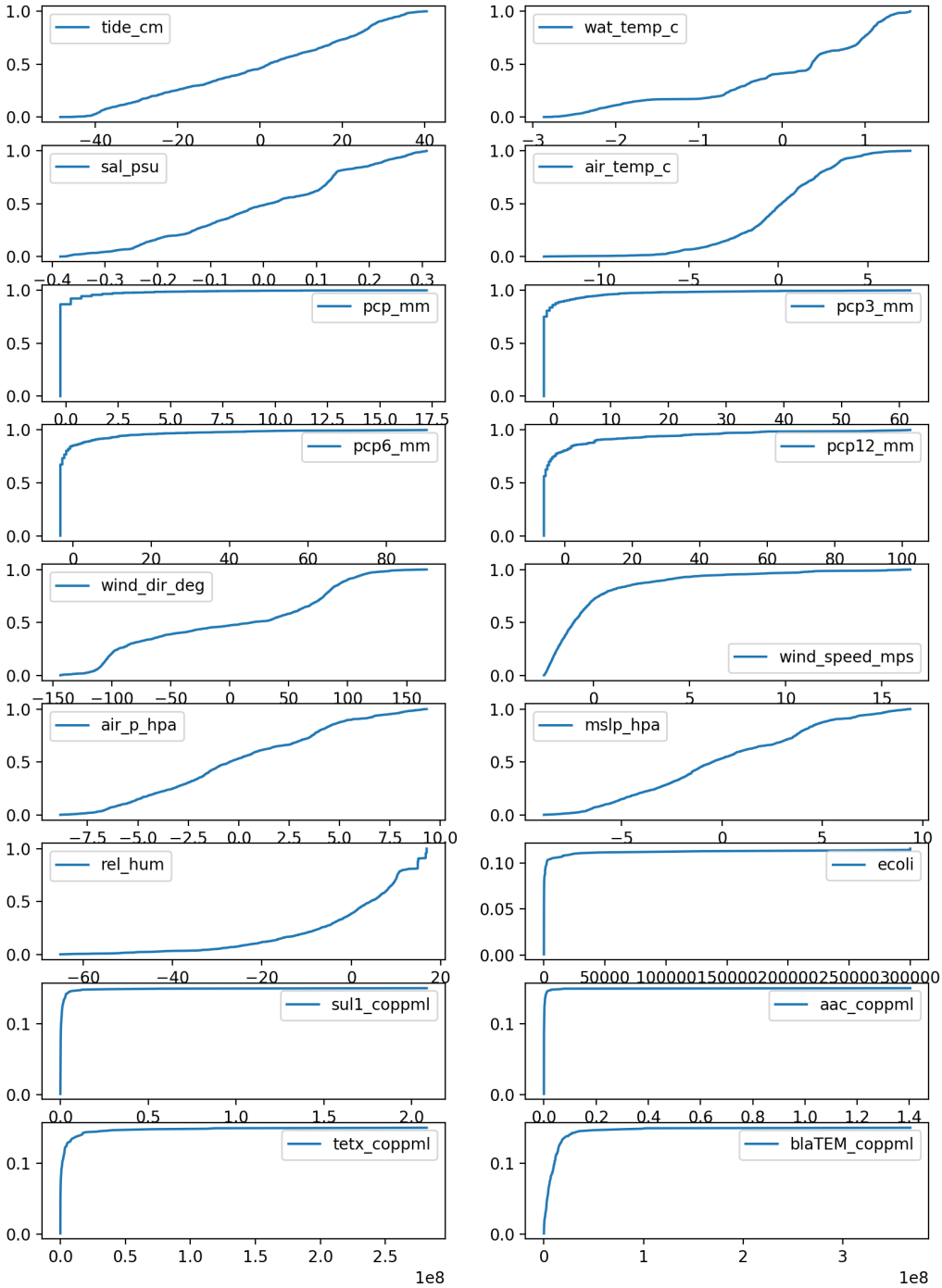

sul1_coppml
y(t + 2)
y(t + 4)
y(t + 6)
y(t + 8)
y(t + 10)
y(t + 12)
y(t + 14)
y(t)
1e8
1e9


aac_coppml
y(t)
y(t + 2)
y(t + 4)
y(t + 6)
y(t + 8)
y(t + 10)
y(t + 12)
y(t + 14)
1e7
1e9
0.05
0.00
−0.05
0.0
0.5
1.0
0
1
5

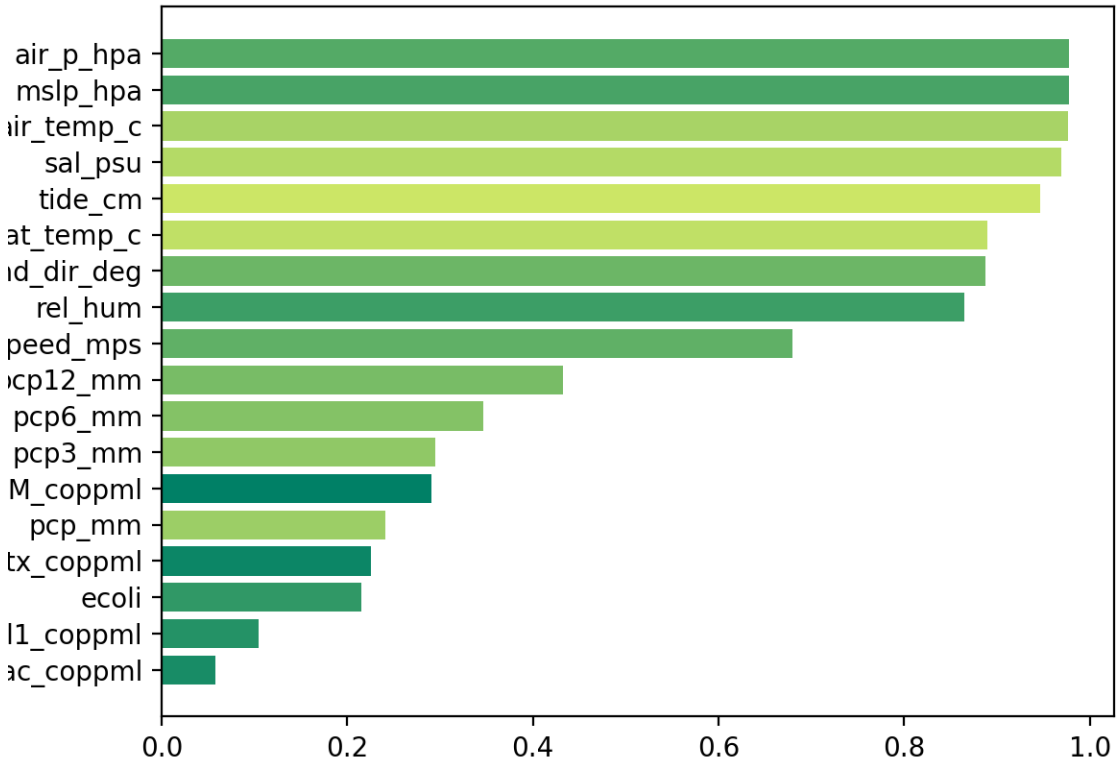

tetx_coppml
y(t)
y(t + 2)
y(t + 4)
y(t + 6)
y(t + 8)
y(t + 10)
y(t + 12)
y(t + 14)
1e8




```
_ = eda.plot_ecdf(figsize=(10, 14))
```


tide_cm
wat_temp_c
sal_psu
air_temp_c
pcp_mm
pcp3_mm
pcp6_mm
pcp12_mm
wind_dir_deg
wind_speed_mps
air_p_hpa
mslp_hpa
rel_hum
ecoli
sul1_coppml
aac_coppml
tetx_coppml
blaTEM_coppml
1e8
1e8

```
eda.normality_test()
```



Total running time of the script: (0 minutes 38.103 seconds)

1.2 Quadica dataset

```
# sphinx_gallery_thumbnail_number = 3

import pandas as pd
import matplotlib.pyplot as plt
from easy_mpl import hist, ridge
from ai4water.datasets import Quadica
from easy_mpl.utils import create_subplots
from ai4water.utils.utils import get_version_info
```

```
for k,v in get_version_info().items():
    print(k, v)
```

```
python 3.7.9 (default, Oct 19 2020, 15:13:17)
[GCC 7.5.0]
os posix
```

(continues on next page)

(continued from previous page)

```
ai4water 1.06
easy_mpl 0.21.2
SeqMetrics 1.3.4
numpy 1.21.6
pandas 1.2.3
matplotlib 3.5.3
joblib 1.2.0
```

```
dataset = Quadica()

avg_temp = dataset.avg_temp()
print(avg_temp.shape)
```

```
0% of 38.49 MB downloaded
100% of 38.49 MB downloaded
0% of 0.03 MB downloaded
100% of 0.03 MB downloaded
0% of 1.77 MB downloaded
100% of 1.77 MB downloaded
unzipping /home/docs/checkouts/readthedocs.org/user_builds/ai4water-datasets/envs/latest/
↪lib/python3.7/site-packages/ai4water/datasets/data/Quadica/quadica.zip to /home/docs/
↪checkouts/readthedocs.org/user_builds/ai4water-datasets/envs/latest/lib/python3.7/site-
↪packages/ai4water/datasets/data/Quadica/quadica
(828, 1386)
```

```
avg_temp.head()
```

1.2.1 pet

```
pet = dataset.pet()
print(pet.shape)
```

```
(828, 1386)
```

1.2.2 precipitation

```
pcp = dataset.precipitation()
print(pcp.shape)
```

```
(828, 1386)
```

1.2.3 monthly median values

```
mon_medians = dataset.monthly_medians()
print(mon_medians.shape)
```

```
(16629, 18)
```

```
mon_medians.head()
```

```
wrtds_mon = dataset.wrtds_monthly()
print(wrtds_mon.shape)
```

```
(50186, 47)
```

1.2.4 catchment attributes

```
cat_attrs = dataset.catchment_attributes()
print(cat_attrs.shape)
```

```
(1386, 113)
```

```
print(cat_attrs.columns)
```

```
Index(['OBJECTID', 'Station', 'Area_km2', 'f_AreaGer', 'dem.mean',
       'dem.median', 'slo.mean', 'slo.median', 'twi.mean', 'twi.med',
       ...
       'flashi', 'BFI', 'P_mm', 'P_SIsw', 'P_SI', 'P_lambda', 'P_alpha',
       'PET_mm', 'AI', 'T_mean'],
      dtype='object', length=113)
```

```
dataset.catchment_attributes(stations=[1,2,3])
```

1.2.5 monthly data

```
dyn, cat = dataset.fetch_monthly(max_nan_tol=None)
print(dyn.shape)
```

```
(29484, 33)
```

```
dyn['OBJECTID'].unique()
```

```
array([ 333,  334,  335,  336,  337,  340,  341,  342,  345,  346,  347,
        348,  349,  350,  352,  355,  358,  359,  360,  362,  363,  364,
        365,  368,  370,  373,  374,  376,  380,  381,  391,  393,  637,
        663,  667,  673,  678,  686,  687,  688,  690,  692,  696,  701,
        705,  711,  716,  718,  722,  723,  728,  730,  734,  735,  736,
```

(continues on next page)

(continued from previous page)

```
        737,  739,  740,  742,  744,  745,  746,  750,  752,  754,  769,
        773,  774,  775,  776,  778,  782,  783,  785,  786,  787,  789,
        796,  797,  874,  885,  899,  985,  986,  991, 1011, 1016, 1017,
       1019, 1082, 1113, 1186, 1237, 1238, 1255, 1270, 1271, 1275, 1287,
       1303, 1332, 1467, 1473, 1482, 1495, 1570, 1571, 1573, 1672, 1677,
       1678, 1679, 1680, 1683, 1688, 1690, 1691])
```

```
print(dyn.columns)
```

```
Index(['mean_Flux_NMin', 'median_C_NO3', 'median_FNC_NMin', 'median_FNC_PO4',
       'mean_Flux_PO4', 'median_C_NMin', 'mean_Flux_TOC', 'mean_FNFlux_TN',
       'mean_Flux_NO3', 'median_Q', 'mean_Flux_TN', 'mean_FNFlux_TOC',
       'mean_FNFlux_TP', 'median_FNC_TOC', 'median_FNC_TN', 'median_FNC_TP',
       'mean_FNFlux_DOC', 'median_C_TN', 'mean_FNFlux_NO3', 'median_C_TP',
       'median_FNC_DOC', 'mean_FNFlux_PO4', 'median_C_DOC', 'mean_Flux_DOC',
       'mean_FNFlux_NMin', 'median_C_TOC', 'median_C_PO4', 'mean_Flux_TP',
       'median_FNC_NO3', 'OBJECTID', 'avg_temp', 'precip', 'pet'],
      dtype='object')
```

```
print(dyn.isna().sum())
```

```
mean_Flux_NMin       9161
median_C_NO3         2691
median_FNC_NMin      9161
median_FNC_PO4       1988
mean_Flux_PO4        1988
median_C_NMin        9161
mean_Flux_TOC       15456
mean_FNFlux_TN      18880
mean_Flux_NO3        2691
median_Q               13
mean_Flux_TN        18880
mean_FNFlux_TOC     15469
mean_FNFlux_TP       1819
median_FNC_TOC      15469
median_FNC_TN       18880
median_FNC_TP        1819
mean_FNFlux_DOC     16361
median_C_TN         18880
mean_FNFlux_NO3      2709
median_C_TP          1819
median_FNC_DOC      16361
mean_FNFlux_PO4      1988
median_C_DOC        16361
mean_Flux_DOC       16361
mean_FNFlux_NMin     9161
median_C_TOC        15456
median_C_PO4         1988
mean_Flux_TP         1819
median_FNC_NO3       2709
OBJECTID                0
```

(continues on next page)

(continued from previous page)

```
avg_temp                0
precip                  0
pet                     0
dtype: int64
```

```
print(cat.shape)
```

```
(29484, 113)
```

1.2.6 monthly TN

```
dyn, cat = dataset.fetch_monthly(features="TN", max_nan_tol=0)
print(dyn.shape)
```

```
(6300, 9)
```

```
dyn.head()
```

```
dyn.tail()
```

```
print(dyn.isna().sum())
```

```
median_Q          0
mean_Flux_TN      0
median_FNC_TN     0
mean_FNFlux_TN    0
median_C_TN       0
OBJECTID          0
avg_temp          0
precip            0
pet               0
dtype: int64
```

```
dyn['OBJECTID'].unique()
```

```
array([ 663,  673,  678,  686,  687,  688,  690,  728,  730,  734,  744,
        745,  746,  750,  754,  782,  783,  785,  786,  985,  986,  991,
       1016, 1017, 1019])
```

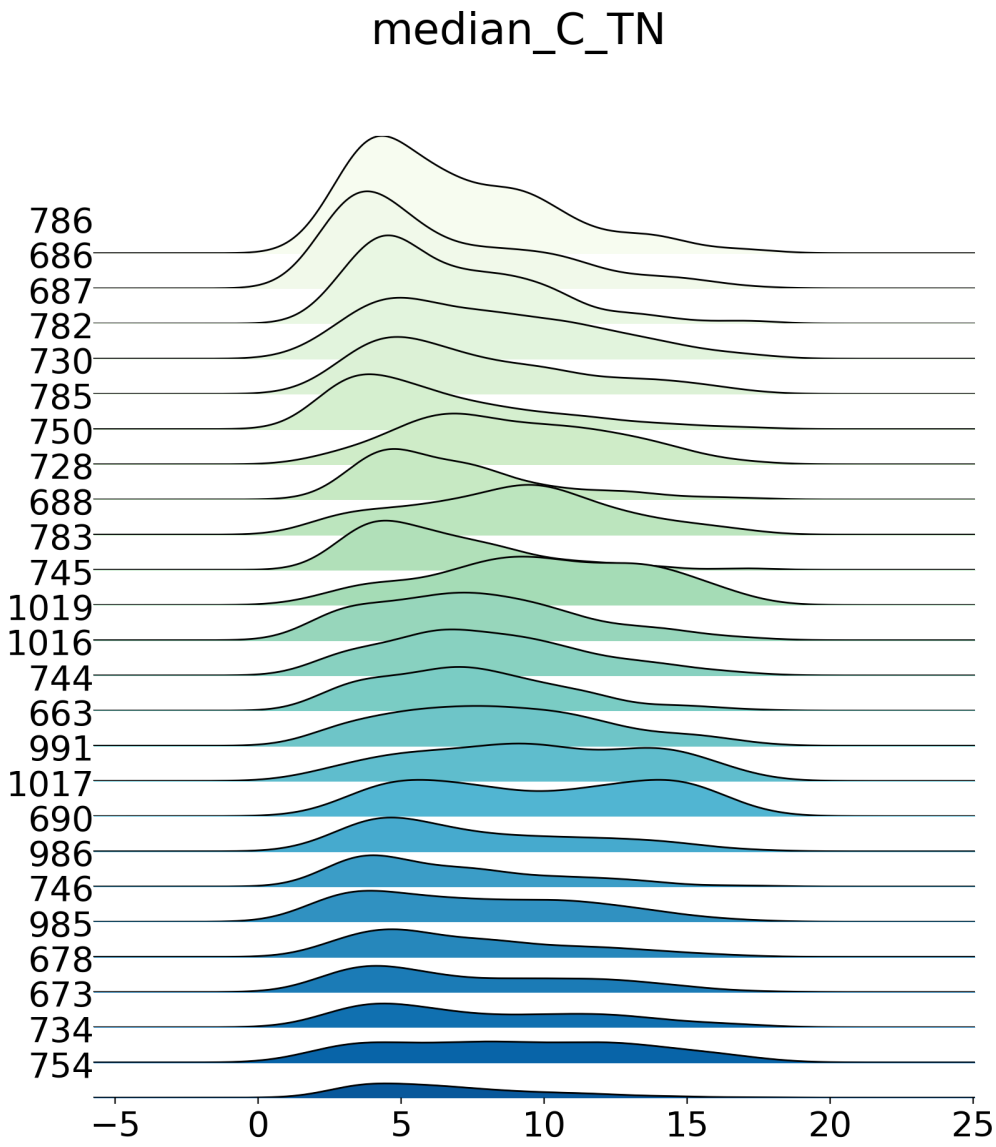
```
print(len(dyn['OBJECTID'].unique()))
```

```
25
```

```
print(cat.shape)
```

```
(6300, 113)
```

```
df = pd.concat([grp['median_C_TN'] for idx,grp in dyn.groupby('OBJECTID')], axis=1)
df.columns = dyn['OBJECTID'].unique()
ridge(df, figsize=(10, 10), color="GnBu", title="median_C_TN")
```




```
[<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
↪<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
↪<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
↪<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
↪<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>]
```

1.2.7 monthly TP

```
dyn, cat = dataset.fetch_monthly(features="TP", max_nan_tol=0)
print(dyn.shape)
```

```
(21420, 9)
```

```
dyn['OBJECTID'].unique()
```

```
array([ 334,  335,  336,  337,  340,  341,  342,  345,  347,  350,  352,
        355,  358,  359,  360,  362,  363,  364,  365,  368,  370,  374,
        376,  380,  381,  391,  663,  673,  678,  686,  687,  688,  690,
        692,  696,  701,  705,  711,  716,  718,  722,  723,  728,  730,
        734,  735,  736,  737,  739,  740,  742,  744,  745,  746,  750,
        754,  769,  773,  776,  778,  782,  783,  785,  786,  874,  885,
        899,  985,  986,  991, 1016, 1017, 1019, 1082, 1113, 1186, 1271,
       1275, 1570, 1571, 1573, 1677, 1678, 1680, 1683])
```

```
print(len(dyn['OBJECTID'].unique()))
```

```
85
```

```
dyn.head()
```

```
dyn.tail()
```

```
print(dyn.isna().sum())
```

```
median_Q          0
median_C_TP       0
mean_Flux_TP      0
mean_FNFlux_TP    0
median_FNC_TP     0
OBJECTID          0
avg_temp          0
precip            0
pet               0
dtype: int64
```

```
print(cat.shape)
```

```
(21420, 113)
```

1.2.8 monthly TOC

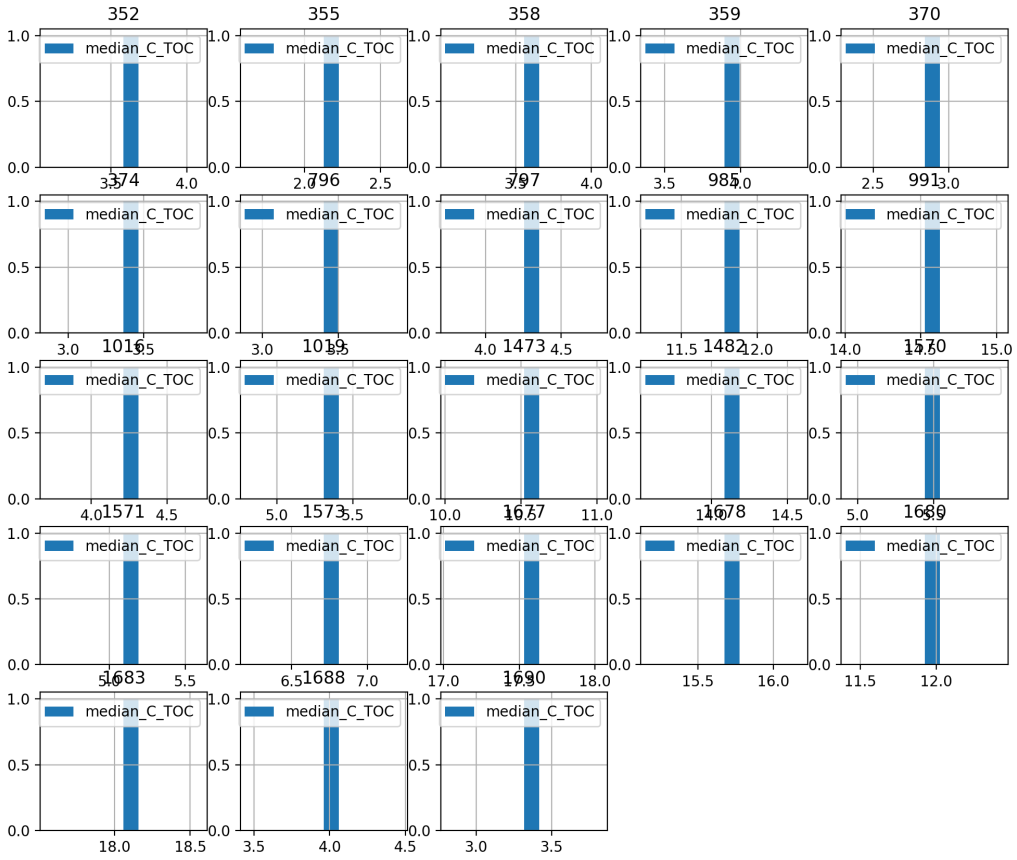
```
dyn, cat = dataset.fetch_monthly(features="TOC", max_nan_tol=0)
print(dyn.shape)
```

```
(5796, 9)
```

```
dyn['OBJECTID'].unique()
```

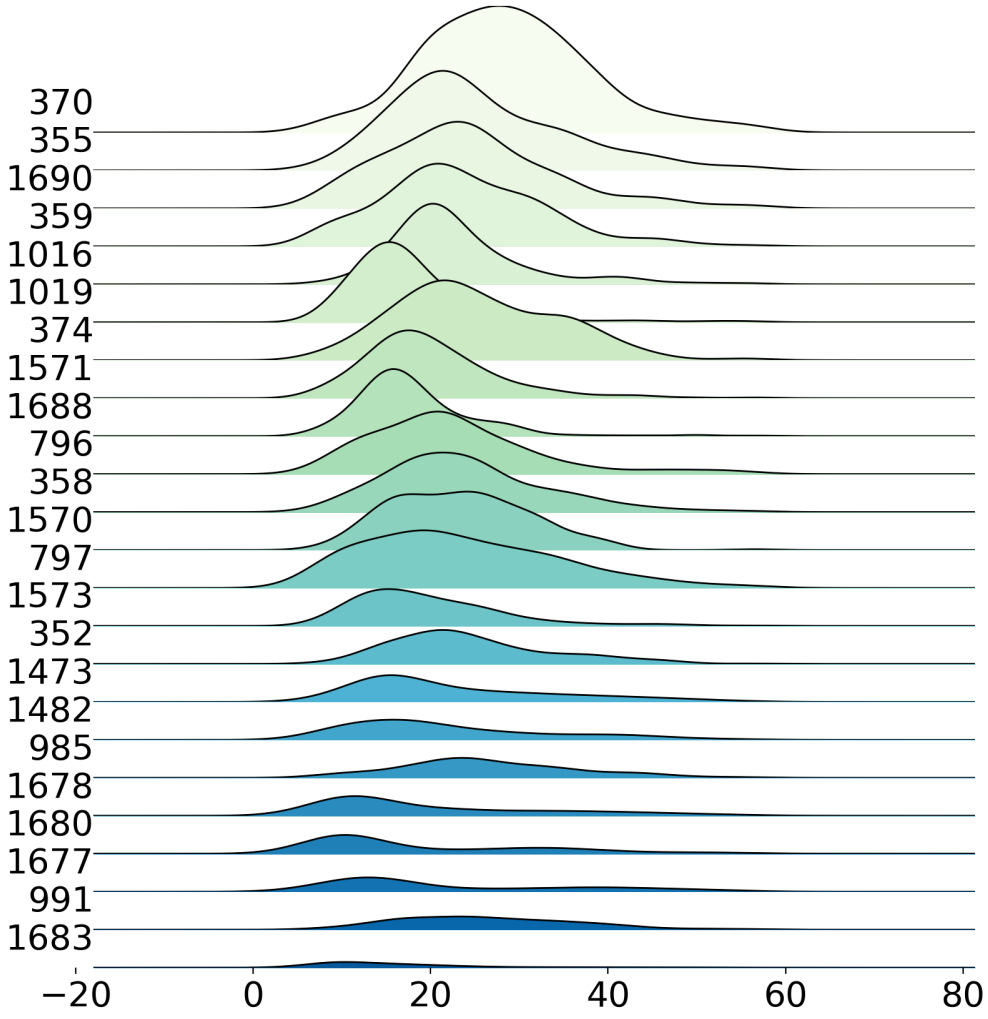
```
array([ 352,  355,  358,  359,  370,  374,  796,  797,  985,  991, 1016,
       1019, 1473, 1482, 1570, 1571, 1573, 1677, 1678, 1680, 1683, 1688,
       1690])
```

```
print(len(dyn['OBJECTID'].unique()))

grouper = dyn.groupby("OBJECTID")



fig, axes = create_subplots(grouper.ngroups, figsize=(12, 10))
for (idx, grp), ax in zip(grouper, axes.flat):
    hist(grp['median_C_TOC'], ax=ax, show=False, ax_kws=dict(title=idx))
plt.show()
```



23

```
df = pd.concat([grp['median_C_TOC'] for idx,grp in dyn.groupby('OBJECTID')], axis=1)
df.columns = dyn['OBJECTID'].unique()

ridge(df, figsize=(10, 10), color="GnBu", title="median_C_TOC")
```




```
[<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
↪<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
↪<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
↪<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
↪<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>]
```

```
dyn.head()
```

```
dyn.tail()
```

```
print(dyn.isna().sum())
```

```
median_C_TOC       0
median_Q           0
mean_FNFlux_TOC    0
median_FNC_TOC     0
mean_Flux_TOC      0
OBJECTID           0
avg_temp           0
precip             0
pet                0
dtype: int64
```

```
print(cat.shape)
```

```
(5796, 113)
```

1.2.9 monthly DOC

```
dyn, cat = dataset.fetch_monthly(features="DOC", max_nan_tol=0)
print(dyn.shape)
```

```
(6804, 9)
```

```
dyn['OBJECTID'].unique()
```

```
array([ 663,  678,  690,  696,  701,  705,  711,  718,  722,  723,  728,
        734,  744,  745,  746,  750,  754,  776,  782,  783,  785,  786,
       1016, 1017, 1019, 1082, 1271])
```

```
print(len(dyn['OBJECTID'].unique()))
```

```
27
```

```
dyn.head()
```

```
dyn.tail()
```

```
print(dyn.isna().sum())
```

```
median_Q           0
median_FNC_DOC     0
median_C_DOC       0
mean_Flux_DOC      0
mean_FNFlux_DOC    0
OBJECTID           0
avg_temp           0
precip             0
pet                0
dtype: int64
```

```
print(cat.shape)
```

```
(6804, 113)
```

Total running time of the script: (0 minutes 21.327 seconds)

CHAPTER

TWO

GALLERY OF EXAMPLES

[ ]:

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search