prova

February 9, 2023

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

[2]:
```
import fit_arima
```

[3]:
```
import open_data
df_temp = open_data.open()
df = df_temp[0]
```

[4]:
```
ritorno = fit_arima.compute(2, 1, catene=4, samples_per_chain=1250, burnin=1000)
```

```
16:43:28 - cmdstanpy - INFO - CmdStan start processing
chain 1 |          | 00:00 Status
chain 2 |          | 00:00 Status
chain 3 |          | 00:00 Status
chain 4 |          | 00:00 Status


16:54:53 - cmdstanpy - INFO - CmdStan done processing.
16:54:53 - cmdstanpy - WARNING - Non-fatal error during sampling:
Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in '/home/b
r1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/code.stan'
, line 157, column 4 to column 34)
        Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/home/br1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/co
de.stan', line 157, column 4 to column 34)
        Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/home/br1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/co
de.stan', line 157, column 4 to column 34)
        Exception: code_model_namespace::log_prob: phi[1][1] is -nan, but must
be greater than or equal to -1.000000 (in '/home/br1/PythonProjects/PythonStats/
```

```
git_bayesian_copie/ARIMA/OnlyCovariates/code.stan', line 87, column 2 to column
47)
        Exception: code_model_namespace::log_prob: phi[2][5] is -nan, but must
be greater than or equal to -1.000000 (in '/home/br1/PythonProjects/PythonStats/
git_bayesian_copie/ARIMA/OnlyCovariates/code.stan', line 87, column 2 to column
47)
Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in '/home/b
r1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/code.stan'
, line 157, column 4 to column 34)
        Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/home/br1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/co
de.stan', line 157, column 4 to column 34)
        Exception: code_model_namespace::log_prob: theta[1] is -nan, but must be
greater than or equal to -1.000000 (in '/home/br1/PythonProjects/PythonStats/git
_bayesian_copie/ARIMA/OnlyCovariates/code.stan', line 86, column 2 to column 36)
        Exception: code_model_namespace::log_prob: phi[1][35] is 1, but must be
less than or equal to 1.000000 (in '/home/br1/PythonProjects/PythonStats/git_bay
esian_copie/ARIMA/OnlyCovariates/code.stan', line 87, column 2 to column 47)
Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in '/home/b
r1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/code.stan'
, line 157, column 4 to column 34)
        Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/home/br1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/co
de.stan', line 157, column 4 to column 34)
        Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/home/br1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/co
de.stan', line 157, column 4 to column 34)
        Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/home/br1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/co
de.stan', line 157, column 4 to column 34)
Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in '/home/b
r1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/code.stan'
, line 157, column 4 to column 34)
        Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'/home/br1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/co
de.stan', line 157, column 4 to column 34)
        Exception: code_model_namespace::log_prob: phi[1][1] is -nan, but must
be greater than or equal to -1.000000 (in '/home/br1/PythonProjects/PythonStats/
git_bayesian_copie/ARIMA/OnlyCovariates/code.stan', line 87, column 2 to column
47)
        Exception: code_model_namespace::log_prob: phi[1][18] is 1, but must be
less than or equal to 1.000000 (in '/home/br1/PythonProjects/PythonStats/git_bay
esian_copie/ARIMA/OnlyCovariates/code.stan', line 87, column 2 to column 47)
Consider re-running with show_console=True if the above output is unclear!


/home/br1/PythonProjects/PythonStats/git_bayesian_copie/ARIMA/OnlyCovariates/fit
_arima.py:41: FutureWarning: Indexing with a float is deprecated, and will raise
```

```
an IndexError in pandas 2.0. You can manually convert to an integer key instead.
  y_missing_list.append({'Stazione':df.columns[v[i]],'Data':df.index[u[i]],'Samp
les':inference_data.posterior.w.values[:,:,i].reshape(catene*samples_per_chain)}
)
```

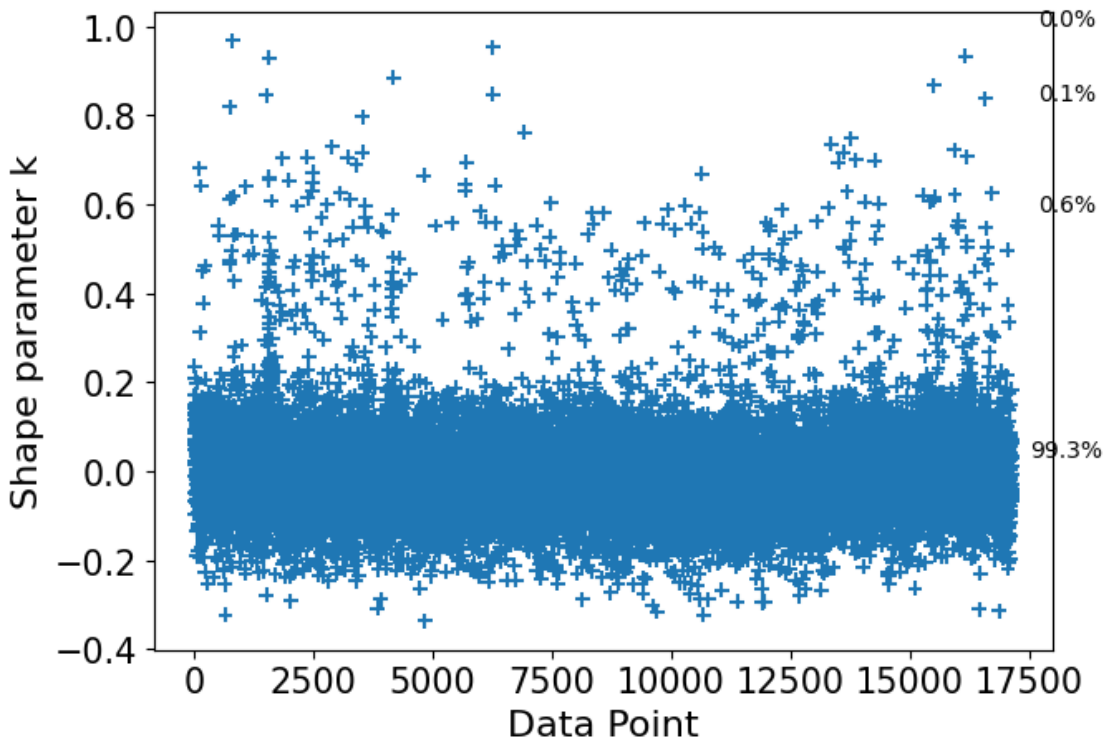
Tempo di computazione: 11:24

[5]:
```
res = az.loo(ritorno['inference_data'], var_name="log_lik", pointwise=True)
res
```

```
/home/br1/PythonProjects/PythonStats/.venv/lib/python3.10/site-
packages/arviz/stats/stats.py:803: UserWarning: Estimated shape parameter of
Pareto distribution is greater than 0.7 for one or more samples. You should
consider using a more robust model, this is because importance sampling is less
likely to work well if the marginal posterior and LOO posterior are very
different. This is more likely to happen with a non-robust model and highly
influential observations.
  warnings.warn(
```

[5]:
```
Computed from 5000 posterior samples and 17148 observations log-likelihood
matrix.

         Estimate       SE
elpd_loo  7232.91   155.38
p_loo      276.29        -

There has been a warning during the calculation. Please check the results.
------

Pareto k diagnostic values:
                         Count   Pct.
(-Inf, 0.5]   (good)     17023   99.3%
 (0.5, 0.7]   (ok)         102    0.6%
   (0.7, 1]   (bad)         23    0.1%
   (1, Inf)   (very bad)     0    0.0%
```

[6]:
```
aux_plt = az.plot_khat(res, show_bins=True, figsize=(7,5))
```

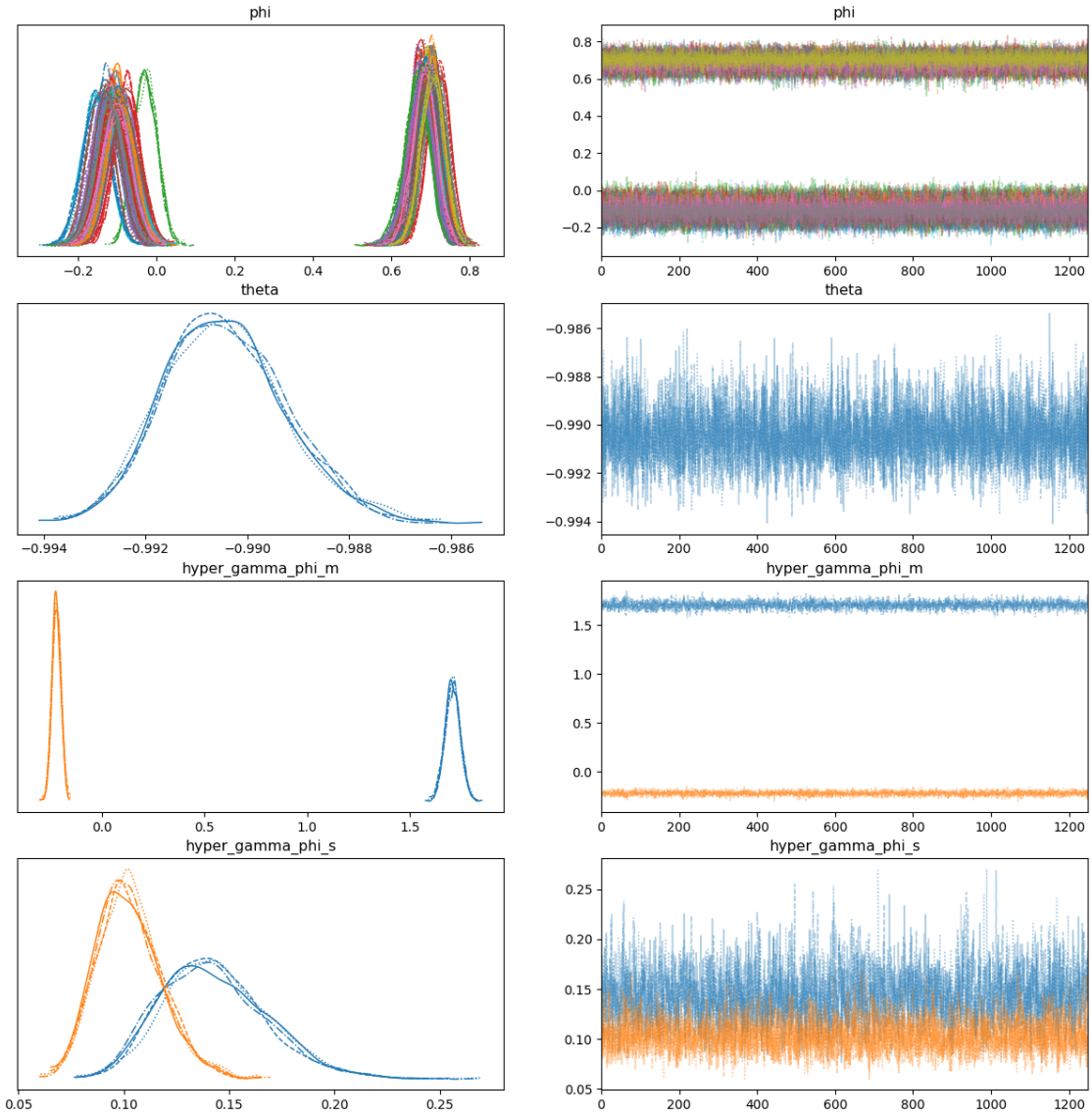


[7]:
```
res = az.waic(ritorno['inference_data'], var_name="log_lik")
res
```

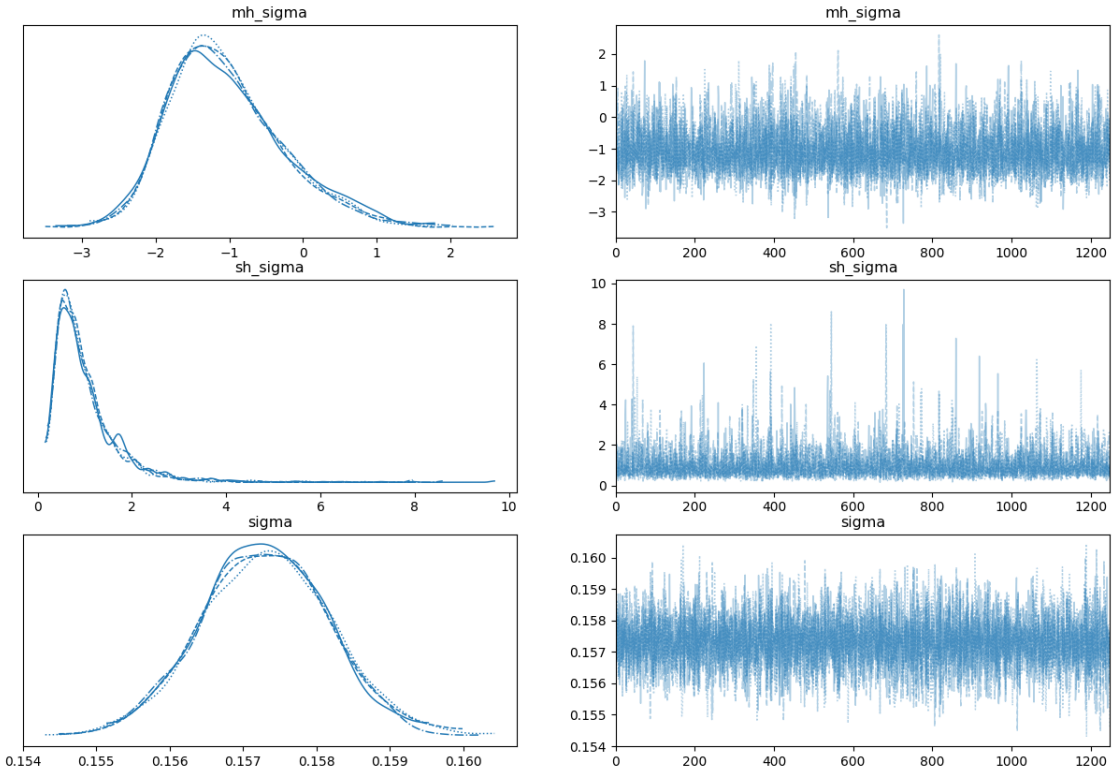
```
/home/br1/PythonProjects/PythonStats/.venv/lib/python3.10/site-
packages/arviz/stats/stats.py:1645: UserWarning: For one or more samples the
posterior variance of the log predictive densities exceeds 0.4. This could be
indication of WAIC starting to fail.
See http://arxiv.org/abs/1507.04544 for details
  warnings.warn(
```

[7]:
```
Computed from 5000 posterior samples and 17148 observations log-likelihood
matrix.

          Estimate       SE
elpd_waic  7246.67   155.23
p_waic      262.52        -

There has been a warning during the calculation. Please check the results.
```

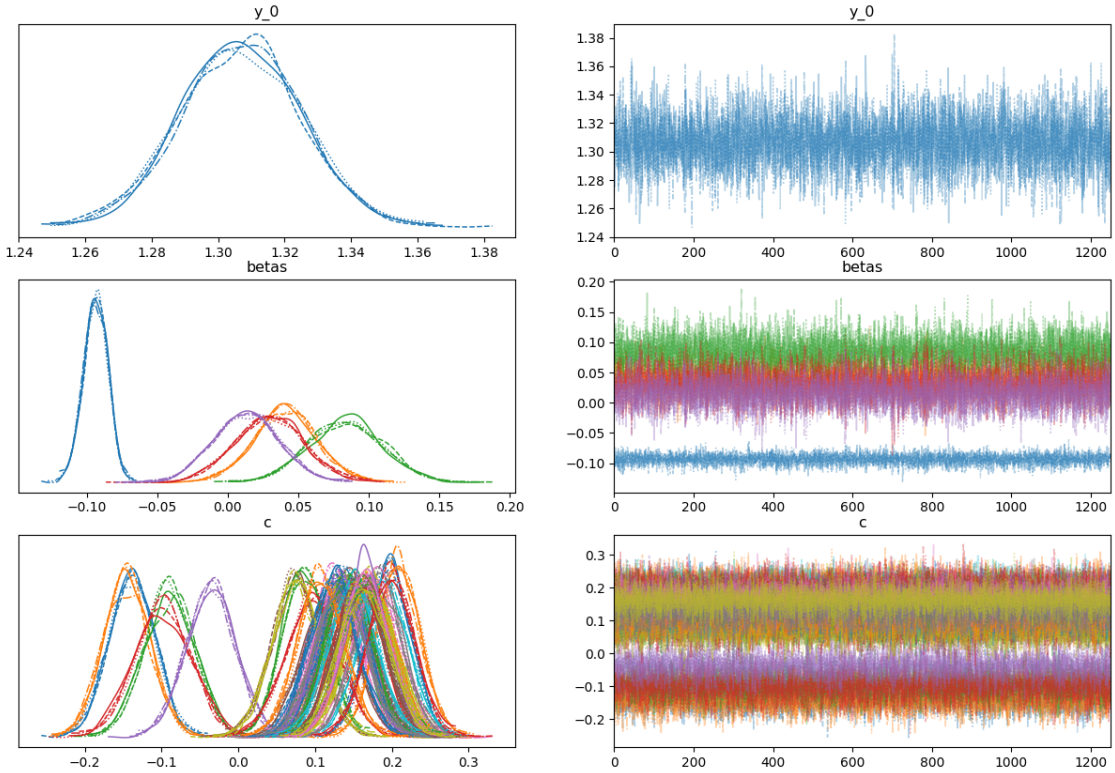
[8]:
```
plt_aux = az.plot_trace(ritorno['inference_data'],␣
 ↪var_names=['phi','theta','hyper_gamma_phi_m','hyper_gamma_phi_s'],␣
 ↪divergences=True, figsize=(15,15))
```



[9]:
```
plt_aux = az.plot_trace(ritorno['inference_data'],␣
↪var_names=['mh_sigma','sh_sigma','sigma'], divergences=True, figsize=(15,10))
```



[10]:
```
plt_aux = az.plot_trace(ritorno['inference_data'],␣
↪var_names=['y_0','betas','c'], divergences=True, figsize=(15,10))
```



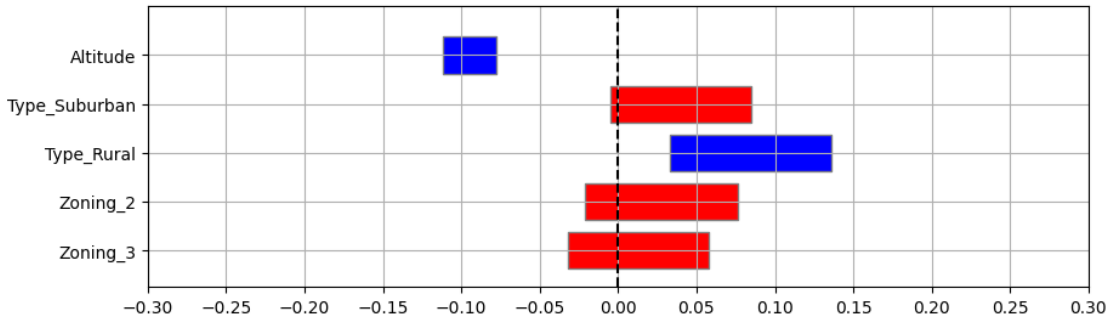
[11]:
```
df_risultati = az.summary(ritorno['inference_data'], var_names=['betas'],␣
 ↪hdi_prob=0.95)
df_risultati.loc[:,'Name'] =␣
 ↪['Altitude','Type_Suburban','Type_Rural','Zoning_2','Zoning_3'] #?????????␣
 ↪Come sono state fatte le covariate?
df_risultati.set_index('Name', inplace=True)
df_risultati
```

[11]:

Name	mean	sd	hdi_2.5%	hdi_97.5%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Altitude	-0.094	0.009	-0.112	-0.078	0.0	0.0	6128.0	4265.0	1.0
Type_Suburban	0.041	0.022	-0.005	0.085	0.0	0.0	11120.0	3589.0	1.0
Type_Rural	0.084	0.026	0.033	0.136	0.0	0.0	9262.0	4023.0	1.0
Zoning_2	0.030	0.025	-0.021	0.076	0.0	0.0	6029.0	3628.0	1.0
Zoning_3	0.013	0.023	-0.032	0.058	0.0	0.0	6029.0	3628.0	1.0

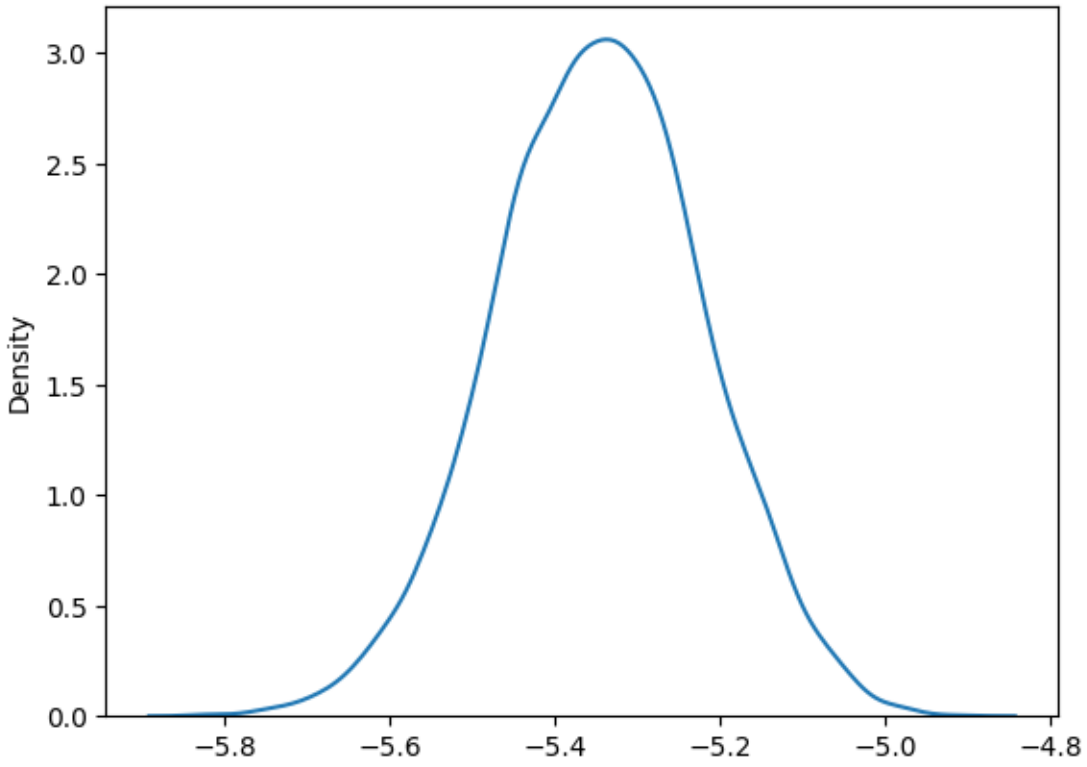
```
Zoning_3          6780.0     4381.0     1.0
```

[12]:
```
import time
from matplotlib.patches import Rectangle

plt.figure(figsize=(10,3))
ax = plt.subplot(1,1,1)
plt.xlim(left = -0.3, right = 0.3)
pos_rows = np.arange(0,len(df_risultati.index))
plt.ylim(bottom=-0.75, top=pos_rows[-1]+1.0)
i = 1
for row in df_risultati.index:
    lci = df_risultati.loc[row, 'hdi_2.5%']
    rci = df_risultati.loc[row, 'hdi_97.5%']
    len_ci = rci - lci
    if lci*rci <= 0:
        col = 'red'
    else:
        col = 'blue'
    ax.add_patch(Rectangle((lci, pos_rows[-i] - 0.75/2), len_ci, 0.75,
             edgecolor = 'grey',
             facecolor = col,
             fill=True,
             lw=1))
    i += 1


plt.axvline(0, 0, pos_rows[-1]+0.75, linestyle='--', color='black')
plt.grid(axis='both')
plt.yticks(ticks=np.flip(pos_rows), labels=df_risultati.index.to_list())
plt.xticks(ticks=np.linspace(-0.3,0.3,13))
plt.show()
```



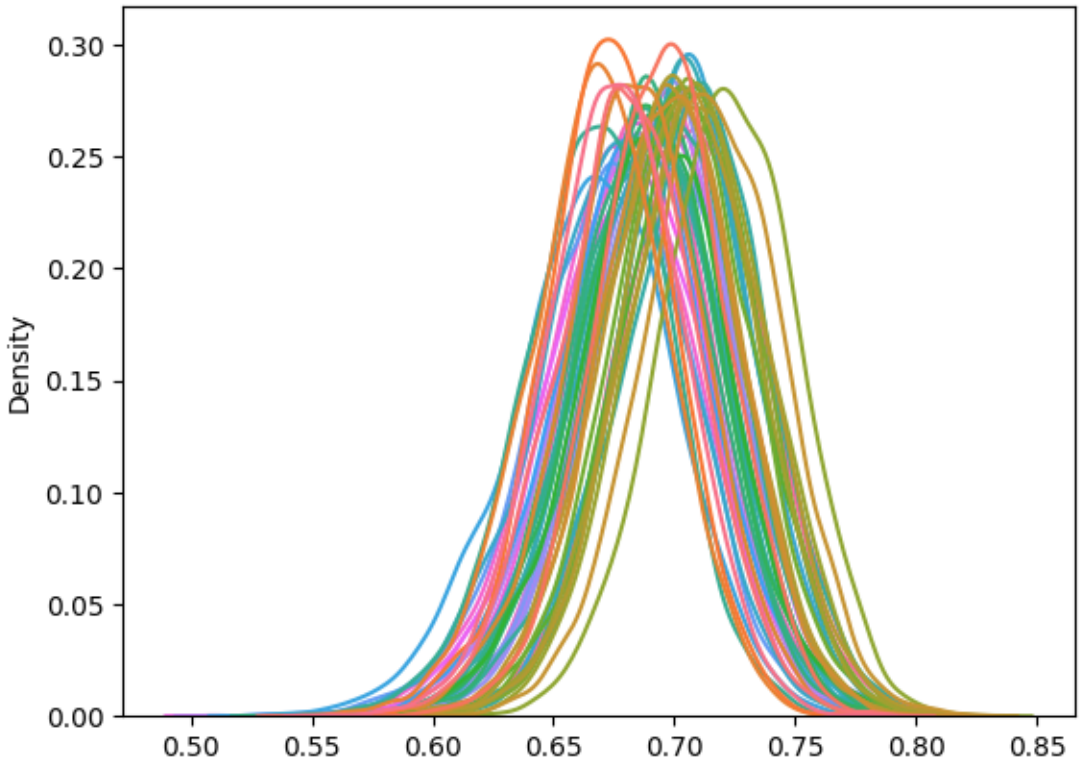
[13]:
```
aux_shape = ritorno['inference_data'].posterior.gamma_th.values.shape
sns.kdeplot(ritorno['inference_data'].posterior.gamma_th.values.
 ↪reshape((aux_shape[0]*aux_shape[1])), legend=False)
```

[13]: <AxesSubplot: ylabel='Density'>



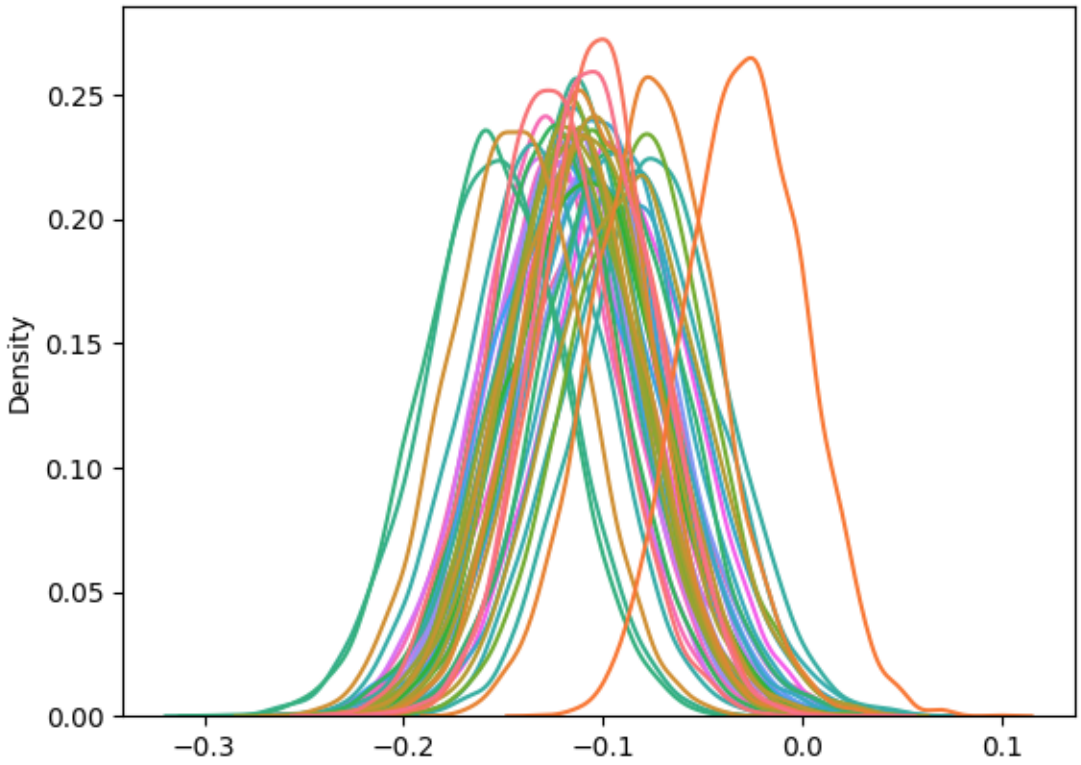
[14]:
```
aux_shape = ritorno['inference_data'].posterior.phi.values.shape
sns.kdeplot(ritorno['inference_data'].posterior.phi.values.
 ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2],aux_shape[3]))[:,0,:],␣
 ↪legend=False)
```

[14]: <AxesSubplot: ylabel='Density'>



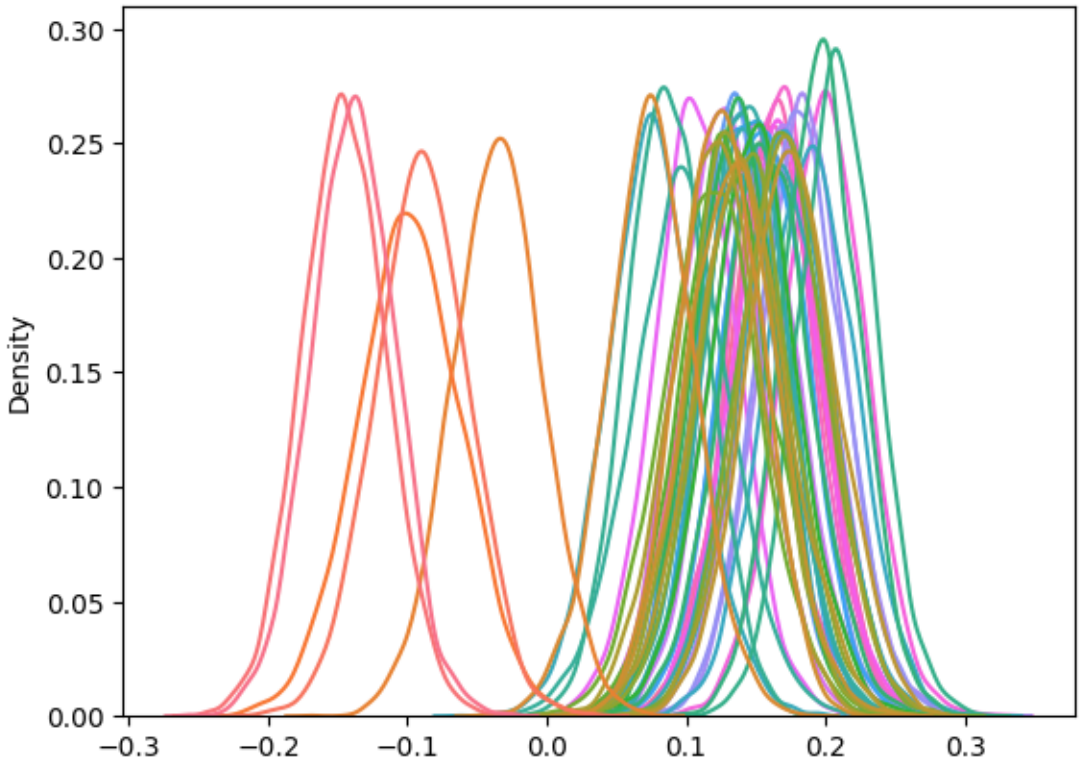
[15]:
```
aux_shape = ritorno['inference_data'].posterior.phi.values.shape
sns.kdeplot(ritorno['inference_data'].posterior.phi.values.
 ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2],aux_shape[3]))[:,1,:],␣
 ↪legend=False)
```

[15]:
```
<AxesSubplot: ylabel='Density'>
```



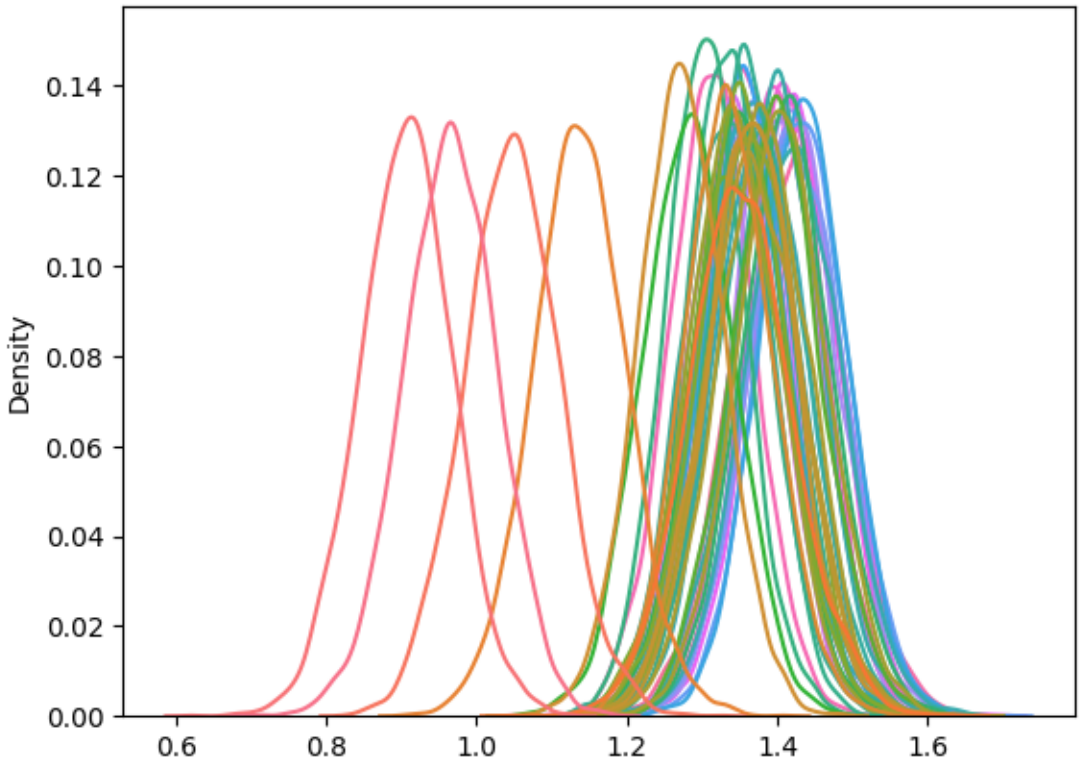
[16]:
```
aux_shape = ritorno['inference_data'].posterior.c.values.shape
sns.kdeplot(ritorno['inference_data'].posterior.c.values.
  ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2])), legend=False)
```

[16]: <AxesSubplot: ylabel='Density'>



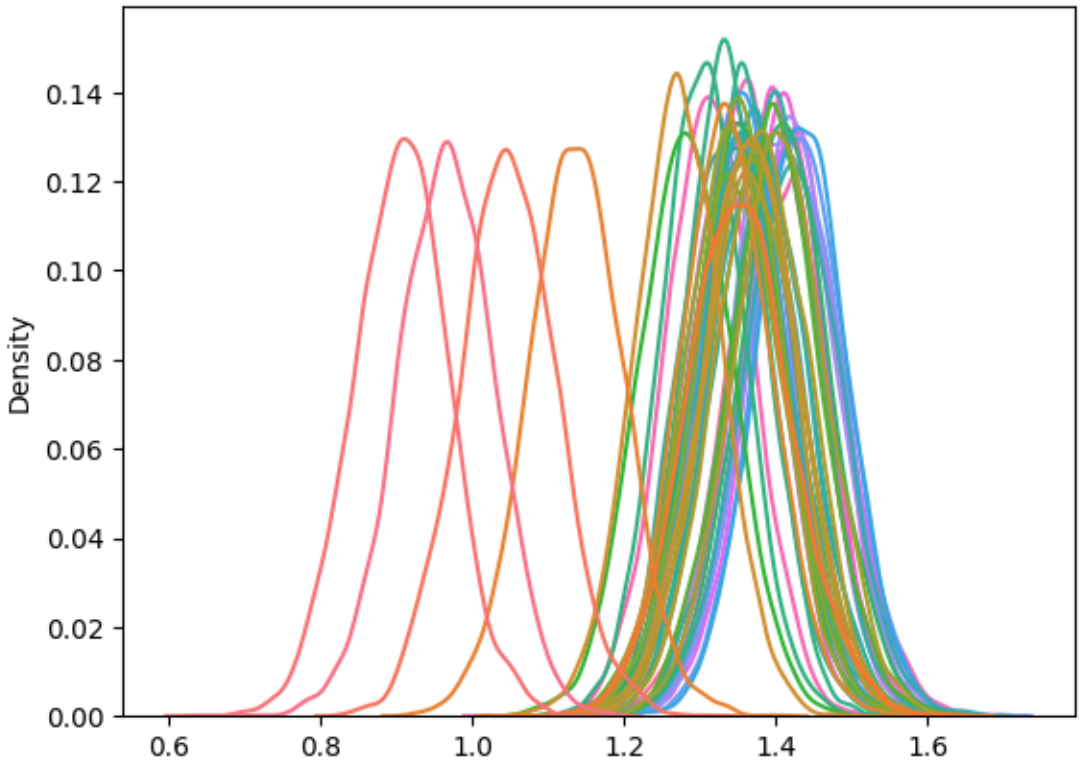
[17]:
```
aux_shape = ritorno['inference_data'].posterior.annual_mean.values.shape
sns.kdeplot(ritorno['inference_data'].posterior.annual_mean.values.
  ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2])), legend=False)
```

[17]:
```
<AxesSubplot: ylabel='Density'>
```



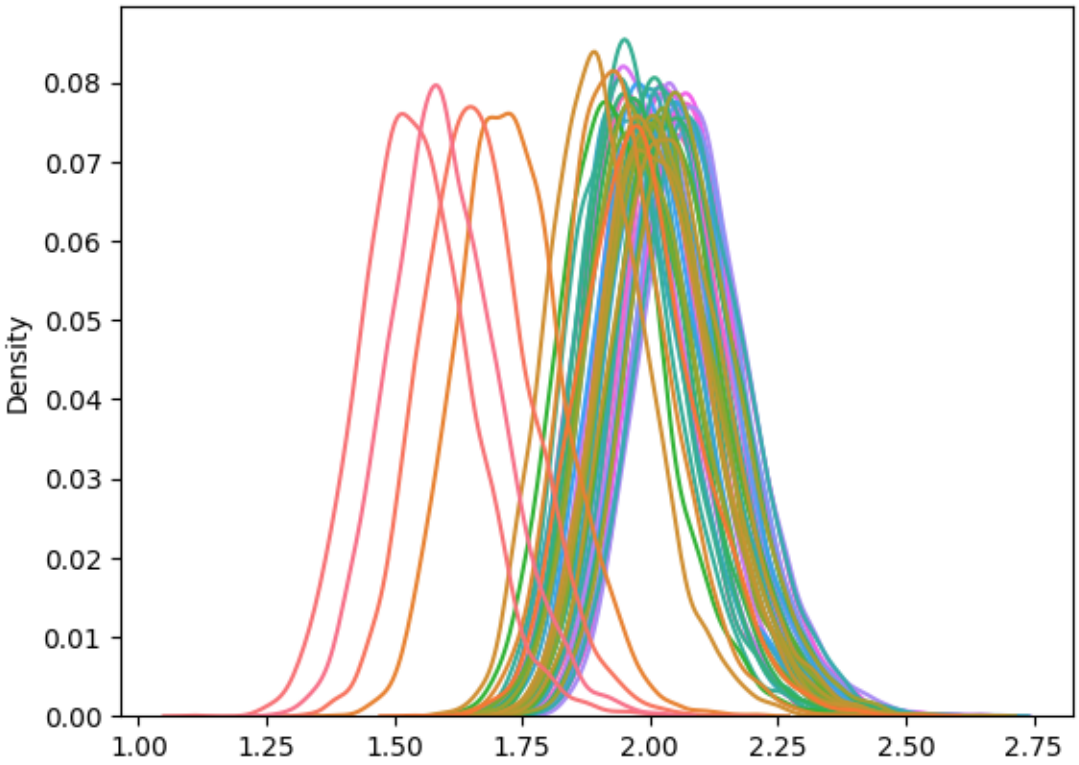
[18]:
```
aux_shape = ritorno['inference_data'].posterior.annual_median.values.shape
sns.kdeplot(ritorno['inference_data'].posterior.annual_median.values.
  ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2])), legend=False)
```

[18]: <AxesSubplot: ylabel='Density'>

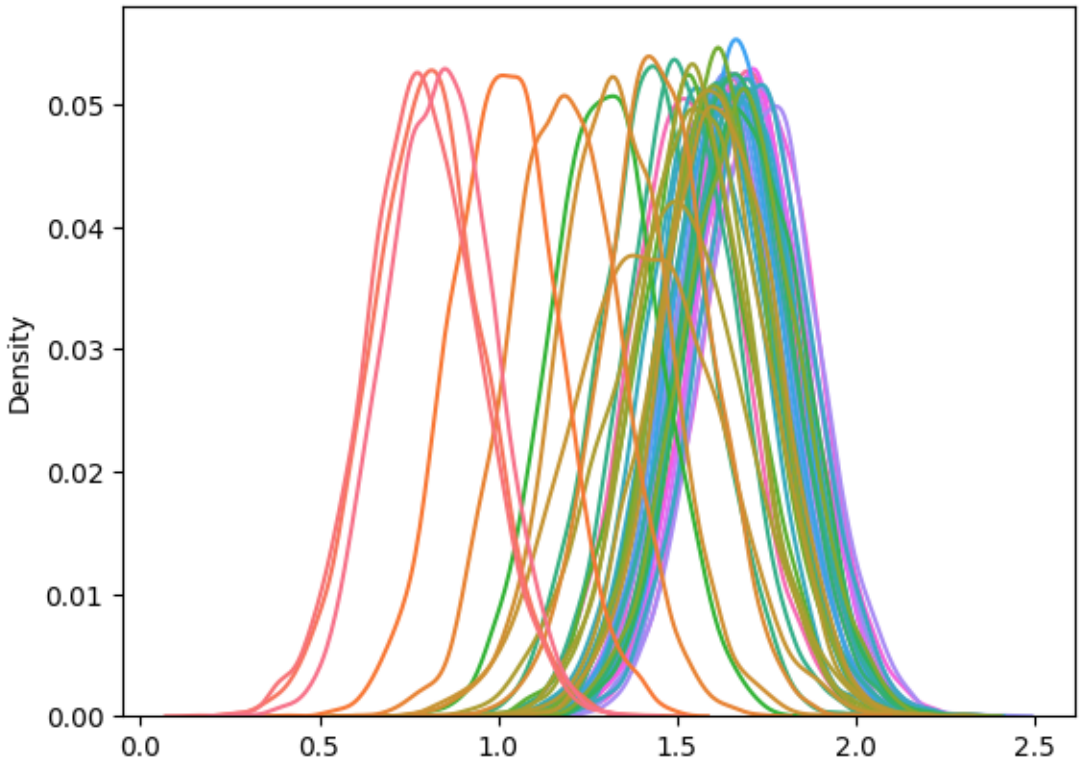


[19]:
```
aux_shape = ritorno['inference_data'].posterior.annual_max.values.shape
sns.kdeplot(ritorno['inference_data'].posterior.annual_max.values.
  ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2])), legend=False)
```

[19]: <AxesSubplot: ylabel='Density'>

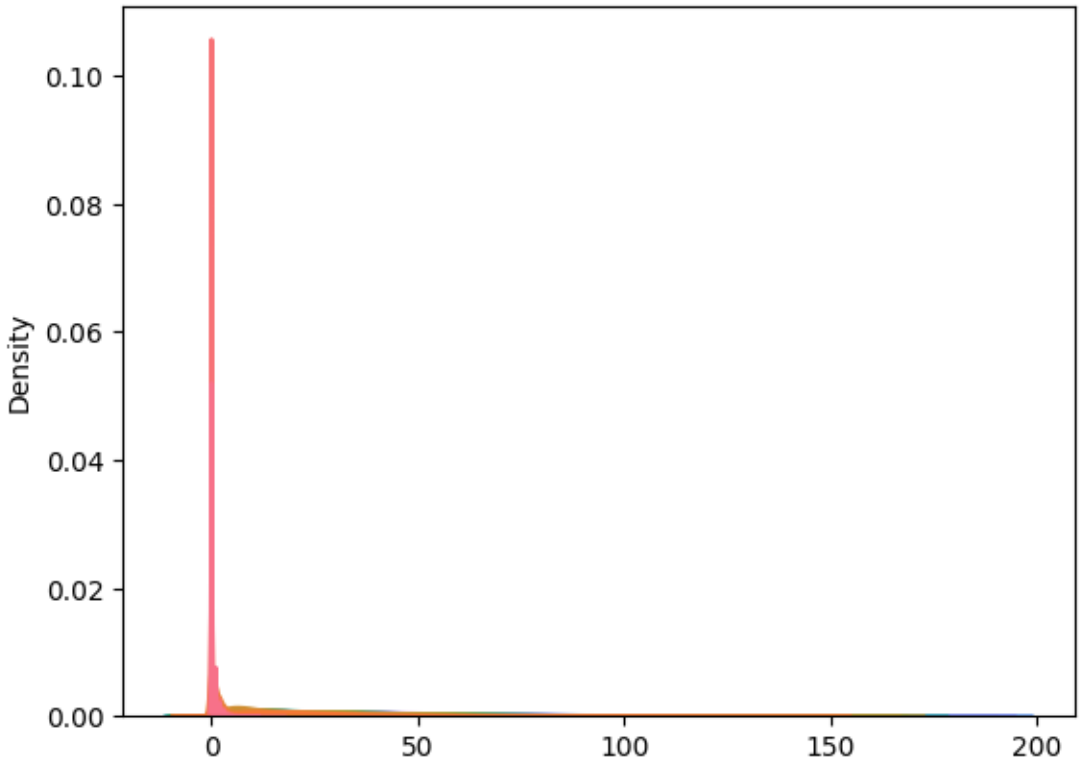


[20]:
```
valori_31_dic = []
for dizionario in ritorno['missing']:
    if dizionario['Data'] == '2018-12-31':
        valori_31_dic.append(dizionario['Samples'])

res_plot = sns.kdeplot(valori_31_dic, legend=False)
```



[21]:
```
aux_shape = ritorno['inference_data'].posterior.annual_days_over_threshold.
 ↪values.shape
sns.kdeplot(ritorno['inference_data'].posterior.annual_days_over_threshold.
 ↪values.reshape((aux_shape[0]*aux_shape[1],aux_shape[2])), legend=False)
```

[21]:
```
<AxesSubplot: ylabel='Density'>
```



[22]:
```
pd.set_option('display.max_columns', 500)
aux_results = pd.DataFrame(ritorno['inference_data'].posterior.
 ↪annual_days_over_threshold.values.
 ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2])), columns=df.columns).
 ↪describe()
aux_results.sort_values('mean', axis=1, ascending=True)
```

[22]:

	FEBBIO	CASTELLUCCIO	CORTE BRUGNATELLA	SAN LEO	BADIA	\
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	
mean	0.099800	0.264200	0.690600	1.664600	10.998200	
std	0.389192	0.695194	1.352345	2.506744	7.888158	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	5.000000	
50%	0.000000	0.000000	0.000000	1.000000	9.000000	
75%	0.000000	0.000000	1.000000	2.000000	15.000000	
max	5.000000	11.000000	17.000000	39.000000	69.000000	

	LUGAGNANO	VERUCCHIO	BESENZONE	SAN FRANCESCO	CASTELLARANO	\
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	
mean	14.312600	17.251400	18.501800	19.526200	20.589400	
std	9.739335	11.710714	11.927148	11.461622	11.221123	
min	0.000000	0.000000	0.000000	0.000000	0.000000	

```
25%       7.000000     9.000000    10.000000      11.000000     12.000000
50%      12.000000    15.000000    16.000000      17.000000     18.000000
75%      19.000000    22.000000    24.000000      25.000000     27.000000
max      72.000000   106.000000   105.000000      86.000000     84.000000

          GHERARDI     S. ROCCO  DELTA CERVIA  VIA CHIARINI     GAVELLO  \
count  5000.000000  5000.000000   5000.000000   5000.000000  5000.00000
mean     21.930800    22.077000     22.438200     23.898200    23.91660
std      14.494483    13.166189     13.271451     13.673143    13.78881
min       0.000000     0.000000      0.000000      0.000000     0.00000
25%      12.000000    13.000000     13.000000     14.000000    14.00000
50%      19.000000    20.000000     20.000000     21.000000    21.00000
75%      29.000000    29.000000     29.000000     31.000000    31.00000
max     127.000000   134.000000    118.000000     98.000000   123.00000

       SAN PIETRO CAPOFIUME  SAVIGNANO DI RIGO  PARCO MONTECUCCO      SARAGAT  \
count           5000.000000         5000.00000       5000.000000  5000.000000
mean              24.678800           24.73260         25.963200    26.549000
std               15.752449           17.46889         14.938503    14.009998
min                0.000000            0.00000          0.000000     0.000000
25%               13.000000           12.00000         15.000000    16.000000
50%               21.000000           21.00000         23.000000    24.000000
75%               32.000000           33.00000         34.000000    34.000000
max              150.000000          145.00000        111.000000   113.000000

        S. LAZZARO  PARCO EDILCARANI   CITTADELLA  GIARDINI MARGHERITA  \
count  5000.000000       5000.000000  5000.000000          5000.000000
mean     27.678600         27.900800    28.435600            28.612000
std      14.683066         13.344414    15.052892            15.922982
min       0.000000          0.000000     1.000000             0.000000
25%      17.000000         18.000000    18.000000            17.000000
50%      25.000000         26.000000    26.000000            26.000000
75%      36.000000         35.000000    36.000000            37.000000
max     120.000000         99.000000   131.000000           114.000000

         SAVIGNANO     REMESINA  FRANCHINI-ANGELONI   MARECCHIA  \
count  5000.000000  5000.000000         5000.000000  5000.00000
mean     29.114600    30.602000           30.839400    30.99320
std      15.445257    15.677218           16.745748    16.66553
min       1.000000     1.000000            0.000000     1.00000
25%      18.000000    19.000000           19.000000    19.00000
50%      26.000000    28.000000           28.000000    28.00000
75%      37.000000    39.000000           40.000000    40.00000
max     118.000000   121.000000          127.000000   123.00000

       PARCO BERTOZZI        CENTO  VILLA FULVIA       CAORLE  \
count     5000.000000  5000.000000   5000.000000  5000.000000
```

```
mean        31.276000    31.753800     32.183200    32.671200
std         17.586747    16.549247     16.857309    18.020434
min          1.000000     0.000000      0.000000     0.000000
25%         19.000000    20.000000     20.000000    20.000000
50%         28.000000    29.000000     29.000000    29.000000
75%         41.000000    41.000000     41.250000    42.000000
max        157.000000   123.000000    133.000000   140.000000

       PARCO RESISTENZA  PARCO FERRARI  SAN LAZZARO  PORTA SAN FELICE  \
count       5000.000000    5000.000000  5000.000000       5000.000000
mean          33.960800      35.033400    36.740600         37.555800
std           17.273846      16.956703    18.990773         18.037471
min            1.000000       0.000000     1.000000          1.000000
25%           21.000000      23.000000    23.000000         24.000000
50%           31.000000      32.000000    34.000000         35.000000
75%           43.000000      45.000000    46.000000         48.000000
max          163.000000     131.000000   148.000000        137.000000

        MONTEBELLO  GIORDANI-FARNESE       TIMAVO   MALCANTONE         ROMA  \
count  5000.000000       5000.000000  5000.000000  5000.000000  5000.000000
mean     39.698600         39.752800    40.293000    40.804800    43.384200
std      18.656306         18.438733    19.358866    21.029673    20.539128
min       1.000000          2.000000     1.000000     2.000000     1.000000
25%      26.000000         26.000000    27.000000    26.000000    29.000000
50%      37.000000         37.000000    37.000000    37.000000    40.000000
75%      50.000000         50.000000    51.000000    52.000000    55.000000
max     140.000000        135.000000   142.000000   183.000000   157.000000

         DE AMICIS      GERBIDO    PARADIGNA     FLAMINIA     GIARDINI  \
count  5000.000000  5000.000000  5000.000000  5000.000000  5000.000000
mean     43.489800    45.099400    45.673200    45.792000    46.121800
std      20.042471    22.216972    21.254693    20.601612    19.670972
min       4.000000     2.000000     5.000000     2.000000     3.000000
25%      29.000000    29.000000    30.000000    31.000000    32.000000
50%      40.000000    41.000000    42.000000    43.000000    44.000000
75%      55.000000    57.000000    58.000000    58.000000    57.000000
max     139.000000   178.000000   187.000000   160.000000   162.000000

          BOGOLESE         CENO    ZALAMELLA       ISONZO
count  5000.000000  5000.000000  5000.000000  5000.000000
mean     46.480600    47.041000    48.187200    48.776600
std      21.129783    21.574652    22.154312    20.897513
min       3.000000     2.000000     1.000000     5.000000
25%      31.000000    32.000000    32.000000    34.000000
50%      43.000000    44.000000    45.000000    46.000000
75%      59.000000    59.000000    61.000000    61.000000
max     146.000000   158.000000   166.000000   149.000000
```

[23]:
```
pd.set_option('display.max_columns', 500)
aux_results = pd.DataFrame(ritorno['inference_data'].posterior.
 ↪is_over_daily_limit.values.
 ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2])), columns=df.columns).
 ↪describe()
aux_results.sort_values('mean', axis=1, ascending=True)
```

[23]:

	CASTELLUCCIO	FEBBIO	CORTE BRUGNATELLA	SAN LEO	BADIA	\
count	5000.0	5000.0	5000.0	5000.000000	5000.000000	
mean	0.0	0.0	0.0	0.000400	0.012800	
std	0.0	0.0	0.0	0.019998	0.112422	
min	0.0	0.0	0.0	0.000000	0.000000	
25%	0.0	0.0	0.0	0.000000	0.000000	
50%	0.0	0.0	0.0	0.000000	0.000000	
75%	0.0	0.0	0.0	0.000000	0.000000	
max	0.0	0.0	0.0	1.000000	1.000000	

	LUGAGNANO	VERUCCHIO	BESENZONE	SAN FRANCESCO	CASTELLARANO	\
count	5000.000000	5000.0000	5000.000000	5000.000000	5000.00000	
mean	0.037400	0.0768	0.089000	0.092200	0.10000	
std	0.189759	0.2663	0.284772	0.289337	0.30003	
min	0.000000	0.0000	0.000000	0.000000	0.00000	
25%	0.000000	0.0000	0.000000	0.000000	0.00000	
50%	0.000000	0.0000	0.000000	0.000000	0.00000	
75%	0.000000	0.0000	0.000000	0.000000	0.00000	
max	1.000000	1.0000	1.000000	1.000000	1.00000	

	S. ROCCO	DELTA CERVIA	GHERARDI	GAVELLO	VIA CHIARINI	\
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	
mean	0.135600	0.145400	0.150200	0.172000	0.177400	
std	0.342398	0.352539	0.357303	0.377418	0.382045	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	

	SAN PIETRO CAPOFIUME	SAVIGNANO DI RIGO	SARAGAT	PARCO MONTECUCCO	\
count	5000.000000	5000.000000	5000.000000	5000.000000	
mean	0.204800	0.209000	0.224000	0.224000	
std	0.403596	0.406635	0.416964	0.416964	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

	PARCO EDILCARANI	S. LAZZARO	CITTADELLA	GIARDINI MARGHERITA	\
count	5000.000000	5000.00000	5000.000000	5000.000000	
mean	0.247800	0.25180	0.262000	0.276000	
std	0.431778	0.43409	0.439767	0.447061	
min	0.000000	0.00000	0.000000	0.000000	
25%	0.000000	0.00000	0.000000	0.000000	
50%	0.000000	0.00000	0.000000	0.000000	
75%	0.000000	1.00000	1.000000	1.000000	
max	1.000000	1.00000	1.000000	1.000000	

	SAVIGNANO	REMESINA	FRANCHINI-ANGELONI	MARECCHIA	\
count	5000.000000	5000.000000	5000.000000	5000.000000	
mean	0.282000	0.325200	0.325200	0.337600	
std	0.450018	0.468496	0.468496	0.472939	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	1.000000	1.000000	1.000000	1.000000	
max	1.000000	1.000000	1.000000	1.000000	

	PARCO BERTOZZI	CENTO	VILLA FULVIA	CAORLE	\
count	5000.000000	5000.000000	5000.000000	5000.000000	
mean	0.345000	0.354400	0.357800	0.362000	
std	0.475416	0.478379	0.479401	0.480627	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	1.000000	1.000000	1.000000	1.000000	
max	1.000000	1.000000	1.000000	1.000000	

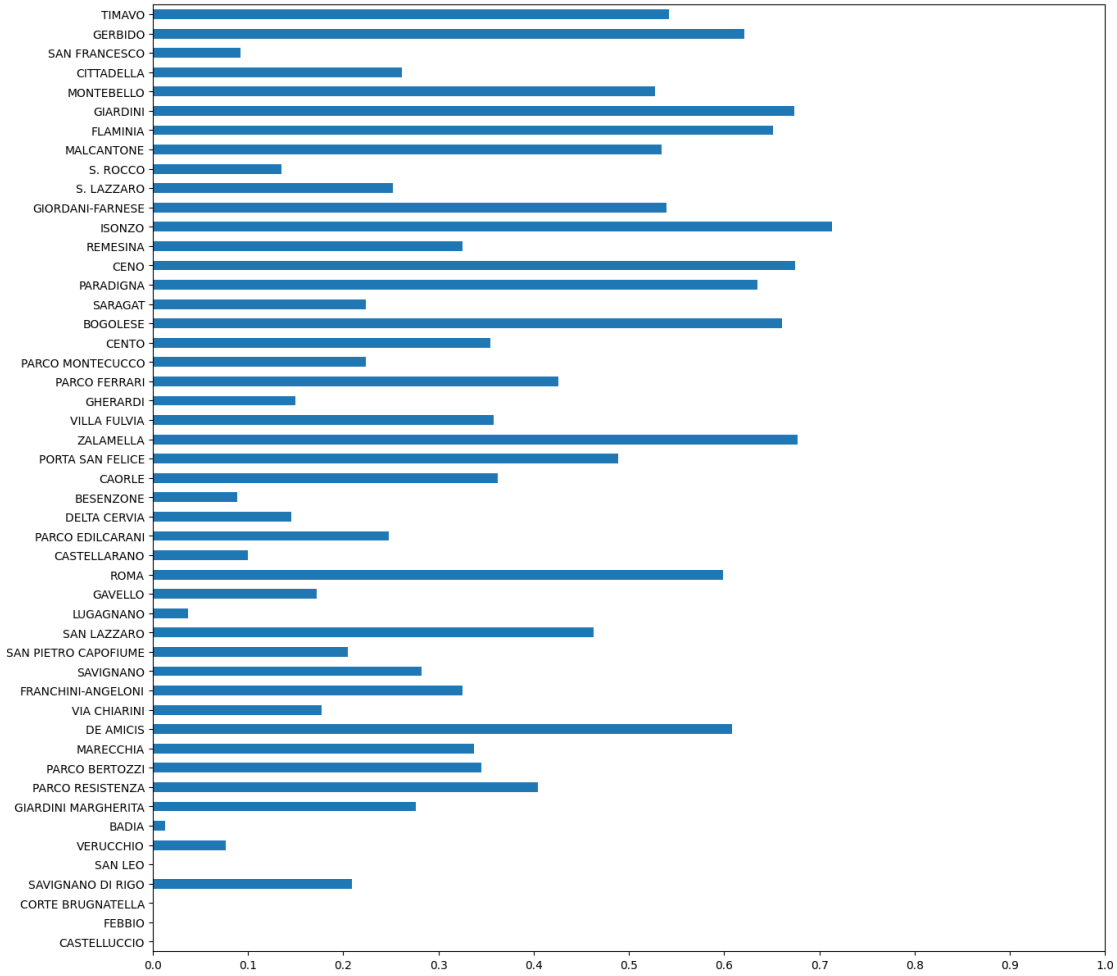
	PARCO RESISTENZA	PARCO FERRARI	SAN LAZZARO	PORTA SAN FELICE	\
count	5000.000000	5000.000000	5000.000000	5000.000000	
mean	0.404600	0.425600	0.462600	0.489000	
std	0.490864	0.494483	0.498649	0.499929	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	1.000000	1.000000	1.000000	1.000000	
max	1.000000	1.000000	1.000000	1.000000	

	MONTEBELLO	MALCANTONE	GIORDANI-FARNESE	TIMAVO	ROMA	\
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.00000	
mean	0.527800	0.534600	0.539200	0.542200	0.59920	
std	0.499276	0.498851	0.498511	0.498266	0.49011	
min	0.000000	0.000000	0.000000	0.000000	0.00000	
25%	0.000000	0.000000	0.000000	0.000000	0.00000	
50%	1.000000	1.000000	1.000000	1.000000	1.00000	

```
75%       1.000000     1.000000          1.000000     1.000000     1.00000
max       1.000000     1.000000          1.000000     1.000000     1.00000

         DE AMICIS      GERBIDO    PARADIGNA     FLAMINIA     BOGOLESE  \
count  5000.000000  5000.000000  5000.000000  5000.000000  5000.000000
mean      0.608600     0.621200     0.635200     0.651400     0.660600
std       0.488112     0.485137     0.481422     0.476575     0.473553
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       1.000000     1.000000     1.000000     1.000000     1.000000
75%       1.000000     1.000000     1.000000     1.000000     1.000000
max       1.000000     1.000000     1.000000     1.000000     1.000000

          GIARDINI         CENO    ZALAMELLA       ISONZO
count  5000.000000  5000.000000  5000.000000  5000.000000
mean      0.673600     0.674600     0.677400     0.713600
std       0.468942     0.468571     0.467518     0.452124
min       0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000
50%       1.000000     1.000000     1.000000     1.000000
75%       1.000000     1.000000     1.000000     1.000000
max       1.000000     1.000000     1.000000     1.000000
```

[24]:
```
aux_results.loc['mean',:].plot(kind='barh', figsize=(15,15), xticks=np.
 ↪linspace(0,1,11))
#res = plt.xticks(rotation = 30)
```

[24]:
```
<AxesSubplot: >
```



[25]:
```
pd.set_option('display.max_columns', 500)
aux_results = pd.DataFrame(ritorno['inference_data'].posterior.
 ↪is_over_annual_limit.values.
 ↪reshape((aux_shape[0]*aux_shape[1],aux_shape[2])), columns=df.columns).
 ↪describe()
aux_results.sort_values('mean', axis=1, ascending=True)
```

[25]:

	CASTELLUCCIO	PARCO MONTECUCCO	PARCO FERRARI	GHERARDI	REMESINA	\
count	5000.0	5000.0	5000.0	5000.0	5000.0	
mean	0.0	0.0	0.0	0.0	0.0	
std	0.0	0.0	0.0	0.0	0.0	
min	0.0	0.0	0.0	0.0	0.0	
25%	0.0	0.0	0.0	0.0	0.0	
50%	0.0	0.0	0.0	0.0	0.0	
75%	0.0	0.0	0.0	0.0	0.0	
max	0.0	0.0	0.0	0.0	0.0	

	BESENZONE	PARCO EDILCARANI	CASTELLARANO	GAVELLO	LUGAGNANO	CENTO	\
count	5000.0	5000.0	5000.0	5000.0	5000.0	5000.0	
mean	0.0	0.0	0.0	0.0	0.0	0.0	
std	0.0	0.0	0.0	0.0	0.0	0.0	
min	0.0	0.0	0.0	0.0	0.0	0.0	
25%	0.0	0.0	0.0	0.0	0.0	0.0	
50%	0.0	0.0	0.0	0.0	0.0	0.0	
75%	0.0	0.0	0.0	0.0	0.0	0.0	
max	0.0	0.0	0.0	0.0	0.0	0.0	

	SAVIGNANO	SARAGAT	FEBBIO	CORTE BRUGNATELLA	MARECCHIA	\
count	5000.0	5000.0	5000.0	5000.0	5000.0	
mean	0.0	0.0	0.0	0.0	0.0	
std	0.0	0.0	0.0	0.0	0.0	
min	0.0	0.0	0.0	0.0	0.0	
25%	0.0	0.0	0.0	0.0	0.0	
50%	0.0	0.0	0.0	0.0	0.0	
75%	0.0	0.0	0.0	0.0	0.0	
max	0.0	0.0	0.0	0.0	0.0	

	SAN FRANCESCO	SAN LEO	GIARDINI MARGHERITA	BADIA	VIA CHIARINI	\
count	5000.0	5000.0	5000.0	5000.0	5000.0	
mean	0.0	0.0	0.0	0.0	0.0	
std	0.0	0.0	0.0	0.0	0.0	
min	0.0	0.0	0.0	0.0	0.0	
25%	0.0	0.0	0.0	0.0	0.0	
50%	0.0	0.0	0.0	0.0	0.0	
75%	0.0	0.0	0.0	0.0	0.0	
max	0.0	0.0	0.0	0.0	0.0	

	VERUCCHIO	FRANCHINI-ANGELONI	VILLA FULVIA	DELTA CERVIA	S. ROCCO	\
count	5000.0	5000.000000	5000.000000	5000.000000	5000.000000	
mean	0.0	0.000200	0.000200	0.000200	0.000200	
std	0.0	0.014142	0.014142	0.014142	0.014142	
min	0.0	0.000000	0.000000	0.000000	0.000000	
25%	0.0	0.000000	0.000000	0.000000	0.000000	
50%	0.0	0.000000	0.000000	0.000000	0.000000	
75%	0.0	0.000000	0.000000	0.000000	0.000000	
max	0.0	1.000000	1.000000	1.000000	1.000000	

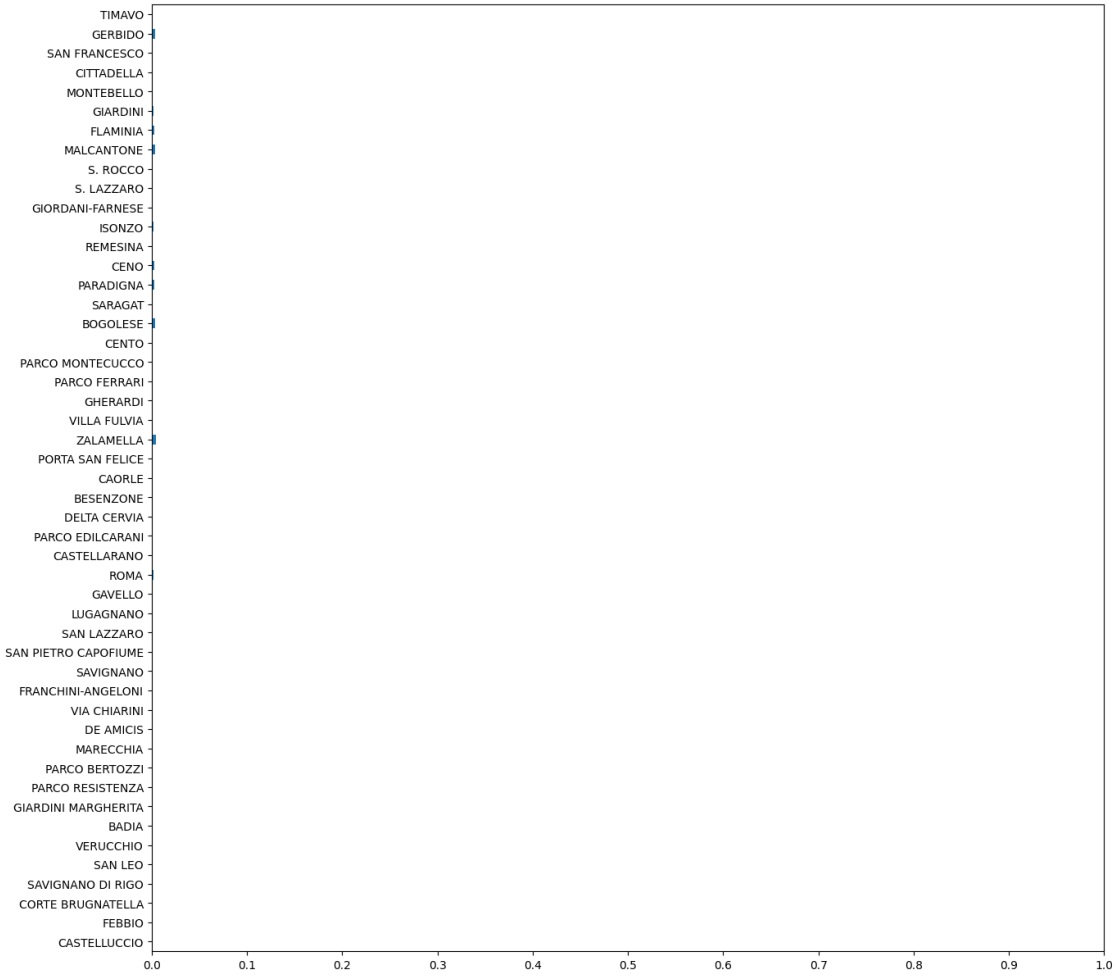
	PARCO RESISTENZA	PARCO BERTOZZI	S. LAZZARO	CITTADELLA	\
count	5000.000000	5000.000000	5000.000000	5000.000000	
mean	0.000200	0.000200	0.000200	0.000200	
std	0.014142	0.014142	0.014142	0.014142	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	

```
75%           0.000000        0.000000     0.000000     0.000000
max           1.000000        1.000000     1.000000     1.000000

       GIORDANI-FARNESE  SAN PIETRO CAPOFIUME    DE AMICIS      CAORLE  \
count       5000.000000           5000.000000  5000.000000  5000.00000
mean           0.000400              0.000400     0.000400     0.00060
std            0.019998              0.019998     0.019998     0.02449
min            0.000000              0.000000     0.000000     0.00000
25%            0.000000              0.000000     0.000000     0.00000
50%            0.000000              0.000000     0.000000     0.00000
75%            0.000000              0.000000     0.000000     0.00000
max            1.000000              1.000000     1.000000     1.00000

       PORTA SAN FELICE      TIMAVO   MONTEBELLO  SAVIGNANO DI RIGO  \
count        5000.00000  5000.00000  5000.000000        5000.000000
mean            0.00060     0.00060     0.000800           0.000800
std             0.02449     0.02449     0.028276           0.028276
min             0.00000     0.00000     0.000000           0.000000
25%             0.00000     0.00000     0.000000           0.000000
50%             0.00000     0.00000     0.000000           0.000000
75%             0.00000     0.00000     0.000000           0.000000
max             1.00000     1.00000     1.000000           1.000000

       SAN LAZZARO       ISONZO         ROMA     GIARDINI         CENO  \
count   5000.00000  5000.000000  5000.000000  5000.000000  5000.000000
mean       0.00100     0.001800     0.002000     0.002000     0.002400
std        0.03161     0.042392     0.044681     0.044681     0.048936
min        0.00000     0.000000     0.000000     0.000000     0.000000
25%        0.00000     0.000000     0.000000     0.000000     0.000000
50%        0.00000     0.000000     0.000000     0.000000     0.000000
75%        0.00000     0.000000     0.000000     0.000000     0.000000
max        1.00000     1.000000     1.000000     1.000000     1.000000

         PARADIGNA     FLAMINIA     BOGOLESE   MALCANTONE      GERBIDO  \
count  5000.000000  5000.000000  5000.000000  5000.000000  5000.000000
mean      0.002800     0.002800     0.003600     0.003800     0.004000
std       0.052846     0.052846     0.059898     0.061533     0.063125
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.000000     0.000000     0.000000
max       1.000000     1.000000     1.000000     1.000000     1.000000

         ZALAMELLA
count  5000.000000
mean      0.004200
std       0.064678
```

```
min       0.000000
25%       0.000000
50%       0.000000
75%       0.000000
max       1.000000
```

[26]:
```
aux_results.loc['mean',:].plot(kind='barh', figsize=(15,15), xticks=np.
 ↪linspace(0,1,11))
#res = plt.xticks(rotation = 30)
```

[26]: <AxesSubplot: >



[27]:
```
y_post_pred = ritorno['inference_data'].posterior.y_post_pred.to_numpy()
y_post_pred = y_post_pred.reshape(y_post_pred.shape[0]*y_post_pred.shape[1],␣
 ↪y_post_pred.shape[2], y_post_pred.shape[3])
y_post_pred.shape
```

[27]: (5000, 365, 49)

[28]:
```
import ipywidgets as widgets
from ipywidgets import HBox, VBox
from IPython.display import display
```

[29]:
```
@widgets.interact(stazione = df.columns)
def f(stazione):
    col_map = sns.light_palette((20, 75, 70), input='husl', as_cmap=True)

    plt.figure(figsize=(14,8))
    ax = plt.subplot(1,1,1)

    idx_stazione = df.columns.to_list().index(stazione)

    """
    sns.lineplot(np.transpose(y_post_pred[0:50,:,idx_stazione]))
    """

    lower_lim = np.zeros(len(df.index))
    upper_lim = np.zeros(len(df.index))
    for i in range(len(df.index)):
        lower_lim[i] = np.percentile(y_post_pred[:,i,idx_stazione],2.5)
        upper_lim[i] = np.percentile(y_post_pred[:,i,idx_stazione],97.5)


#    sns.lineplot(lower_lim)
#    sns.lineplot(upper_lim)

    for i in range(len(df.index)):
        ax.add_patch(mplt.patches.
 ↪Rectangle((i,lower_lim[i]),1,upper_lim[i]-lower_lim[i], fill=True,␣
 ↪color=col_map(180)))

    sns.lineplot(df[stazione])
    sns.lineplot(np.mean(y_post_pred[:,:,idx_stazione], axis=0))


    index = 0
    for line in ax.get_lines():
        if index == 0:
            col_map = sns.dark_palette((230,90,65), input='husl', as_cmap=True)
            line.set_c(col_map(155))
        else:
            col_map = sns.dark_palette((10,90,65), input='husl', as_cmap=True)
            line.set_c(col_map(175))
        index += 1
```

```
    plt.ylim(bottom=0,top=2.5)

    primo_giorno = datetime.date(2018,1,1)
    date_da_segnare = []
    date_da_segnare_posizioni = []

    for i in range(12):
        date_da_segnare.append(datetime.date(2018,i+1,1))
        date_da_segnare_posizioni.append((date_da_segnare[2*i] - primo_giorno).
↪days)
        date_da_segnare.append(datetime.date(2018,i+1,15))
        date_da_segnare_posizioni.append((date_da_segnare[2*i+1] -␣
↪primo_giorno).days)
        date_da_segnare[2*i] = date_da_segnare[2*i].isoformat()
        date_da_segnare[2*i+1] = date_da_segnare[2*i+1].isoformat()

    plt.xticks(date_da_segnare_posizioni,date_da_segnare,rotation=65)
    plt.tick_params(
        axis='x',          # changes apply to the x-axis
        which='both',      # both major and minor ticks are affected
        bottom=True,      # ticks along the bottom edge are off
        top=False,         # ticks along the top edge are off
        labelbottom=True)
    plt.grid()

    """
    ax.get_legend().remove()
    col_vals = np.linspace(1,255,num=len(df.columns))
    index = 0
    for line in ax.get_lines():
        if(index == len(ax.get_lines()) - 1):
            col_map = sns.dark_palette((230,90,65), input='husl', as_cmap=True)
            line.set_c(col_map(175))
            continue
        if(index == len(ax.get_lines()) - 2):
            col_map = sns.dark_palette((120,90,65), input='husl', as_cmap=True)
            line.set_c(col_map(175))
            index += 1
            continue

        #line.set_c(col_map(int(np.round(col_vals[index]))))
        line.set_c(col_map(220))
        line.set_alpha(0.3)
        index += 1
    """
```

```
    plt.show()
```

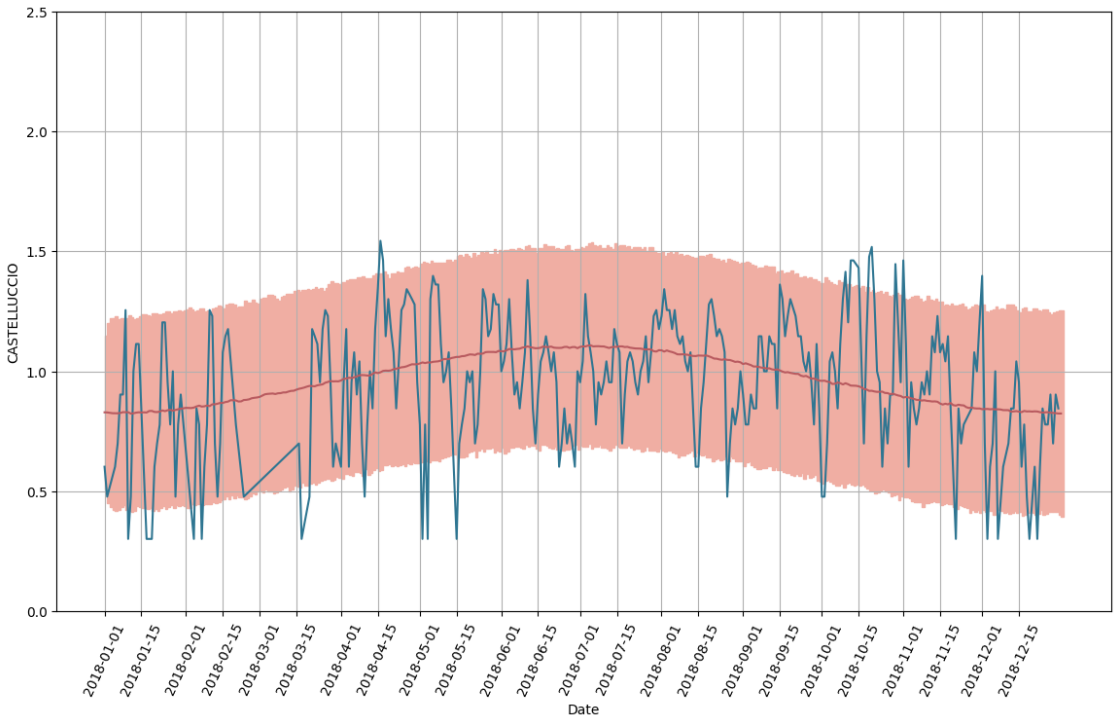
```
interactive(children=(Dropdown(description='stazione', options=('CASTELLUCCIO',␣
 ↪'FEBBIO', 'CORTE BRUGNATELLA',…
```

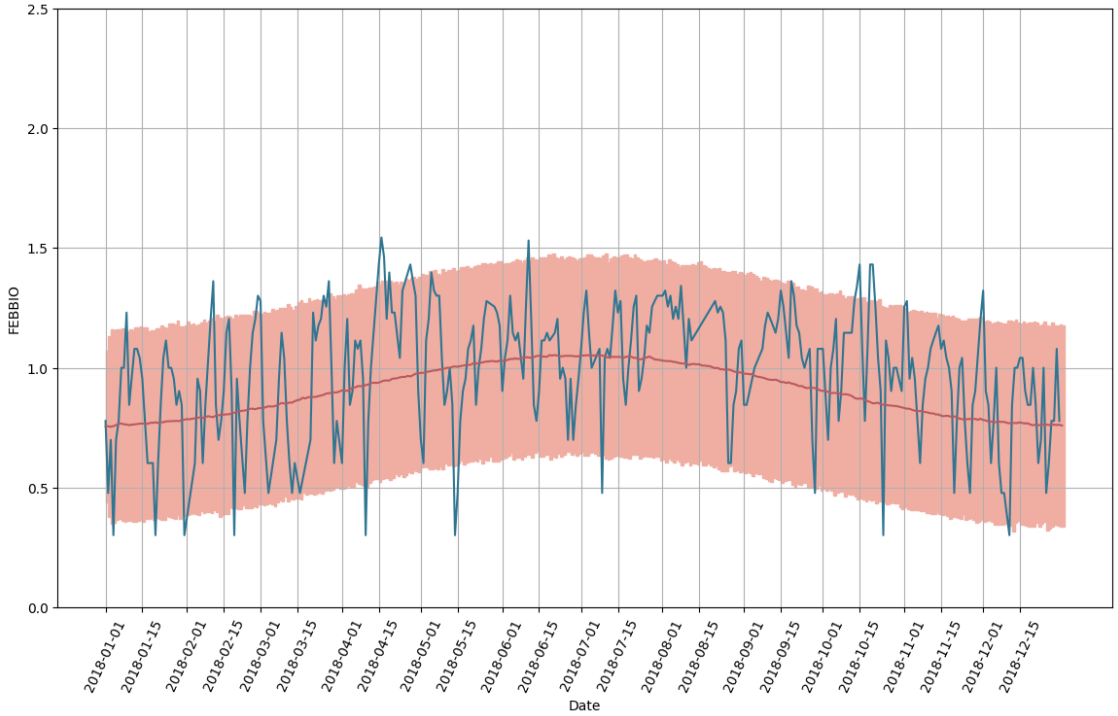
[30]:
```
def f(stazione):
    col_map = sns.light_palette((20, 75, 70), input='husl', as_cmap=True)

    plt.figure(figsize=(14,8))
    ax = plt.subplot(1,1,1)

    idx_stazione = df.columns.to_list().index(stazione)

    """
    sns.lineplot(np.transpose(y_post_pred[0:50,:,idx_stazione]))
    """

    lower_lim = np.zeros(len(df.index))
    upper_lim = np.zeros(len(df.index))
    for i in range(len(df.index)):
        lower_lim[i] = np.percentile(y_post_pred[:,i,idx_stazione],2.5)
        upper_lim[i] = np.percentile(y_post_pred[:,i,idx_stazione],97.5)


#    sns.lineplot(lower_lim)
#    sns.lineplot(upper_lim)

    for i in range(len(df.index)):
        ax.add_patch(mplt.patches.
 ↪Rectangle((i,lower_lim[i]),1,upper_lim[i]-lower_lim[i], fill=True,␣
 ↪color=col_map(180)))

    sns.lineplot(df[stazione])
    sns.lineplot(np.mean(y_post_pred[:,:,idx_stazione], axis=0))


    index = 0
    for line in ax.get_lines():
        if index == 0:
            col_map = sns.dark_palette((230,90,65), input='husl', as_cmap=True)
            line.set_c(col_map(155))
        else:
            col_map = sns.dark_palette((10,90,65), input='husl', as_cmap=True)
            line.set_c(col_map(175))
        index += 1
```

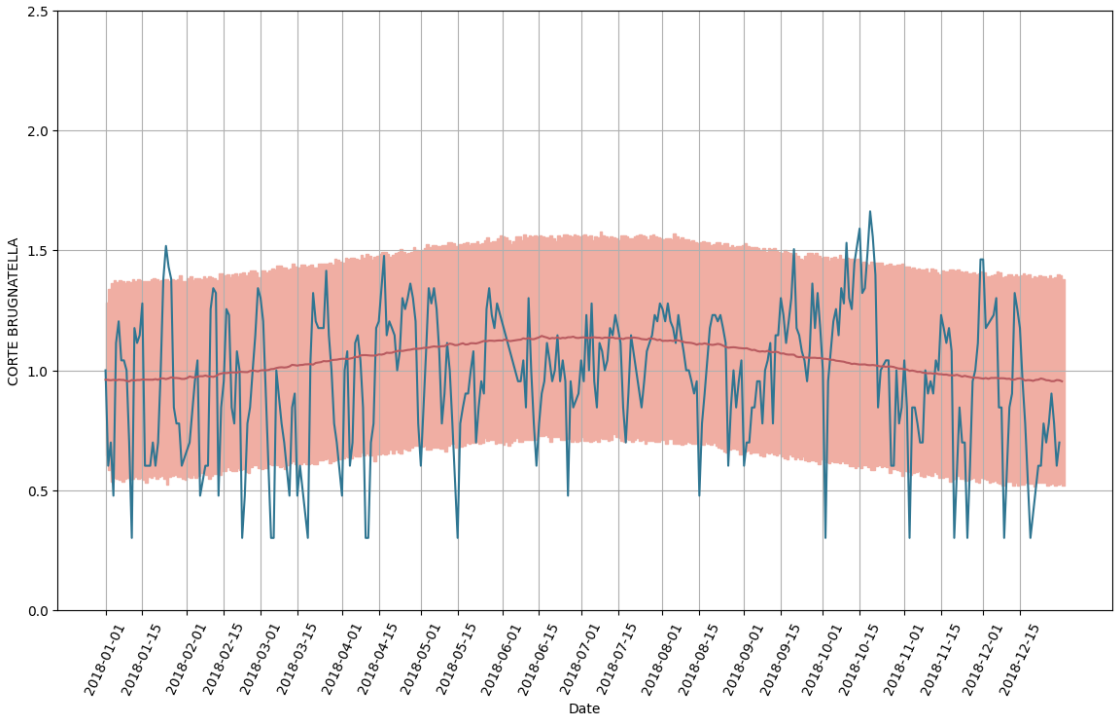
```
    plt.ylim(bottom=0,top=2.5)

    primo_giorno = datetime.date(2018,1,1)
    date_da_segnare = []
    date_da_segnare_posizioni = []

    for i in range(12):
        date_da_segnare.append(datetime.date(2018,i+1,1))
        date_da_segnare_posizioni.append((date_da_segnare[2*i] - primo_giorno).
↪days)
        date_da_segnare.append(datetime.date(2018,i+1,15))
        date_da_segnare_posizioni.append((date_da_segnare[2*i+1] -␣
↪primo_giorno).days)
        date_da_segnare[2*i] = date_da_segnare[2*i].isoformat()
        date_da_segnare[2*i+1] = date_da_segnare[2*i+1].isoformat()

    plt.xticks(date_da_segnare_posizioni,date_da_segnare,rotation=65)
    plt.tick_params(
        axis='x',          # changes apply to the x-axis
        which='both',      # both major and minor ticks are affected
        bottom=True,      # ticks along the bottom edge are off
        top=False,         # ticks along the top edge are off
        labelbottom=True)
    plt.grid()

    """
    ax.get_legend().remove()
    col_vals = np.linspace(1,255,num=len(df.columns))
    index = 0
    for line in ax.get_lines():
        if(index == len(ax.get_lines()) - 1):
            col_map = sns.dark_palette((230,90,65), input='husl', as_cmap=True)
            line.set_c(col_map(175))
            continue
        if(index == len(ax.get_lines()) - 2):
            col_map = sns.dark_palette((120,90,65), input='husl', as_cmap=True)
            line.set_c(col_map(175))
            index += 1
            continue

        #line.set_c(col_map(int(np.round(col_vals[index]))))
        line.set_c(col_map(220))
        line.set_alpha(0.3)
        index += 1
    """
```

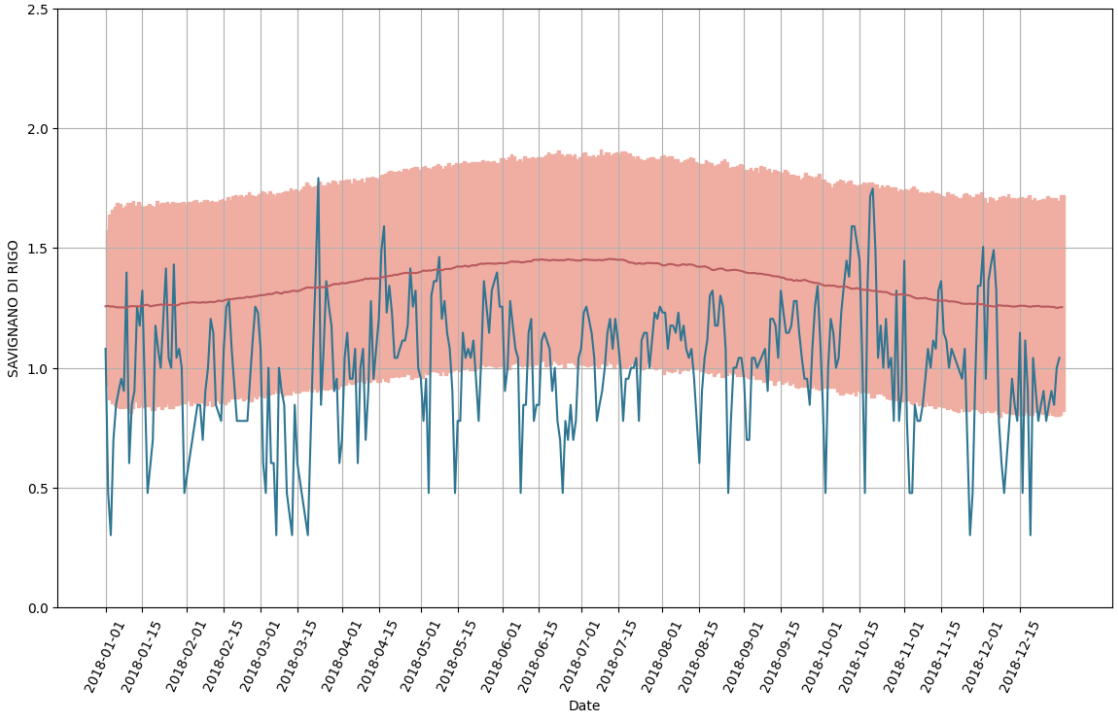
```
    plt.show()
```

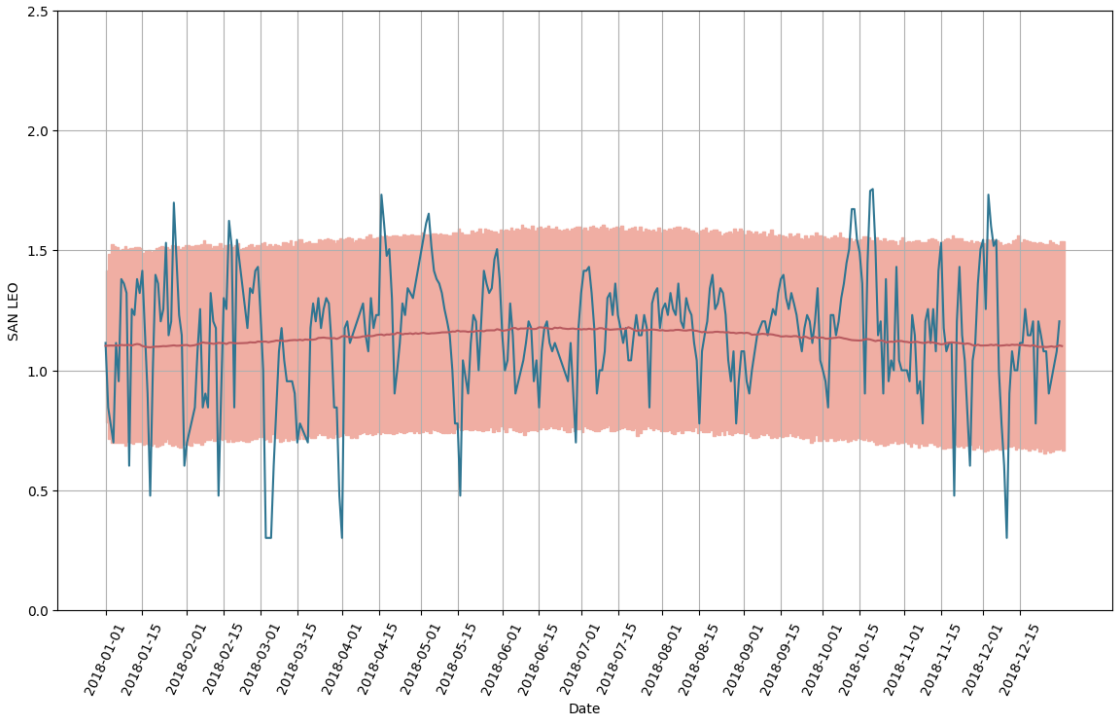
[31]:
```
for stazione in df.columns:
    f(stazione)
```

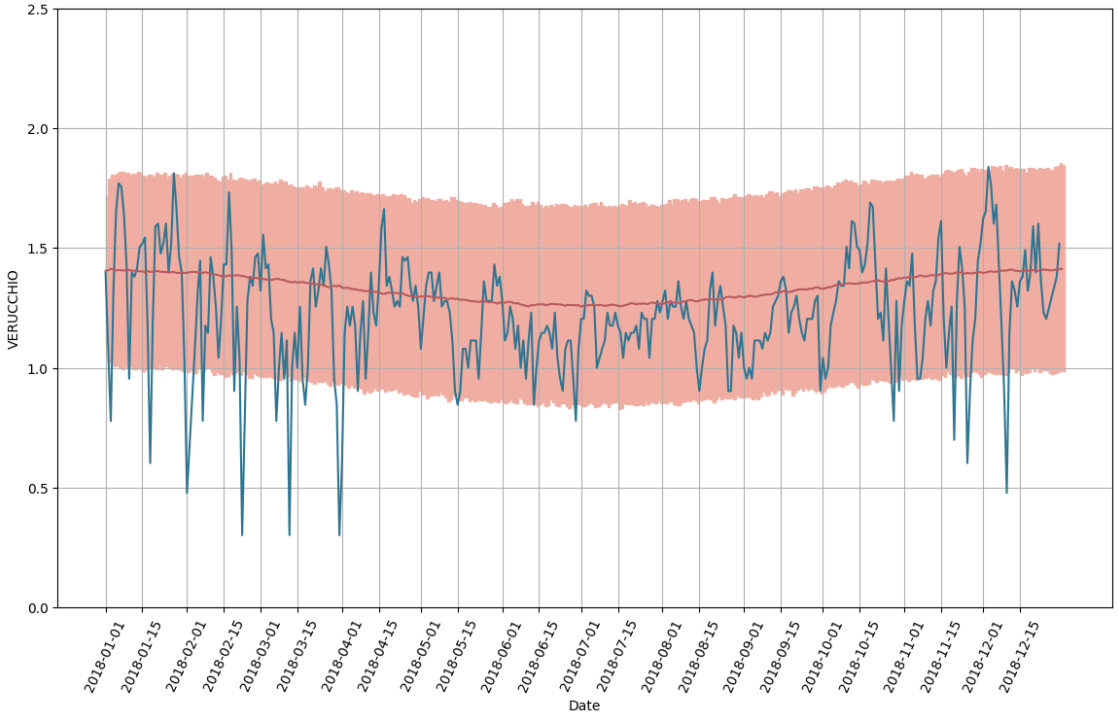


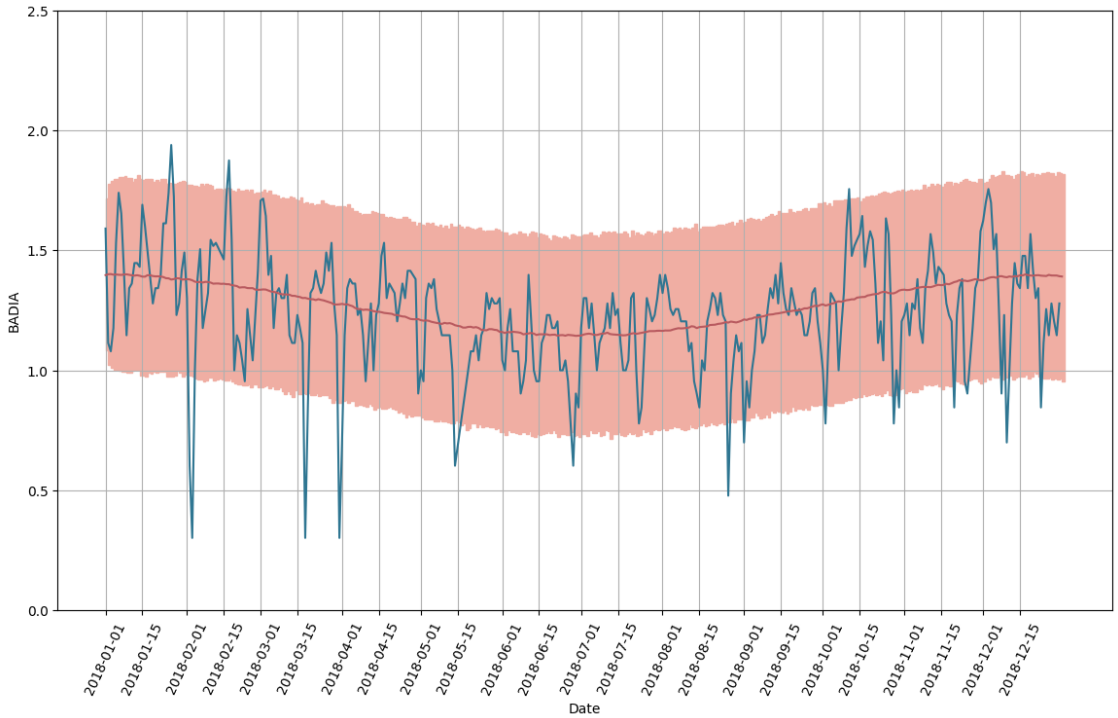

FEBBIO
Date

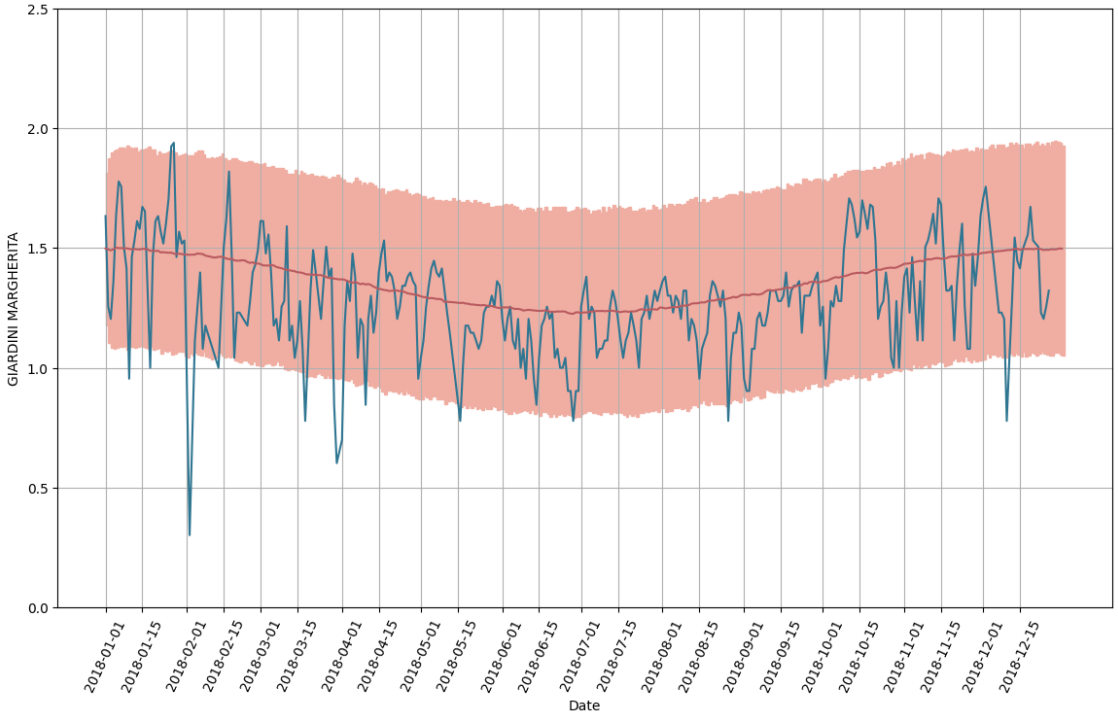

CORTE BRUGNATELLA
Date

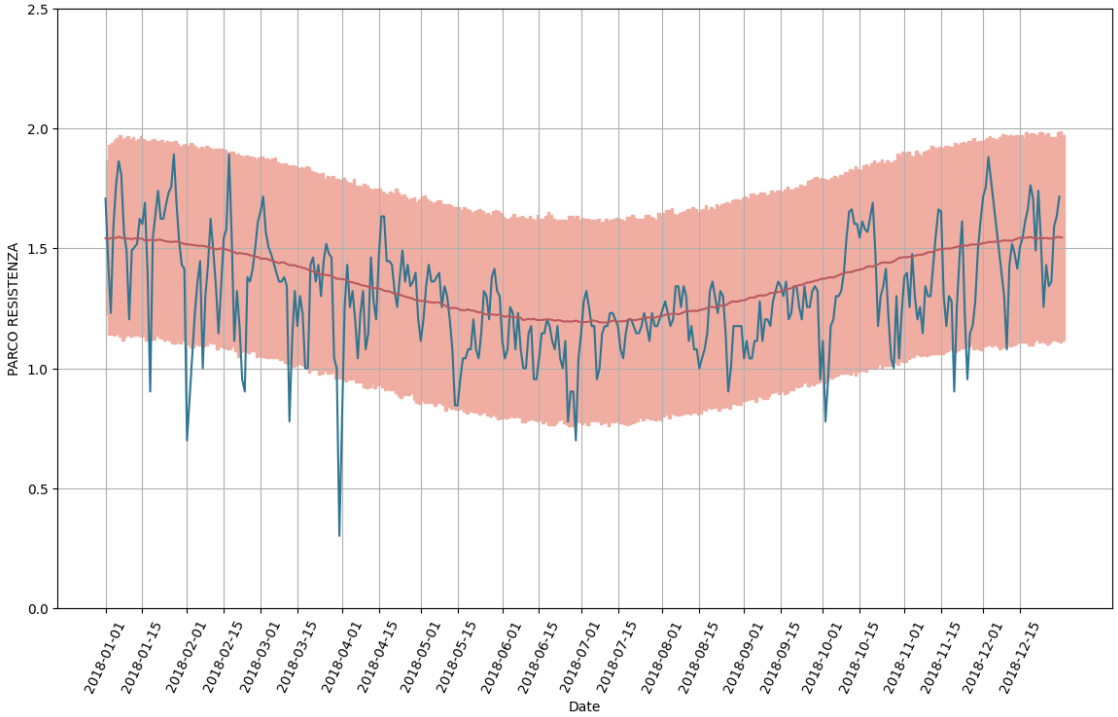

SAVIGNANO DI RIGO
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

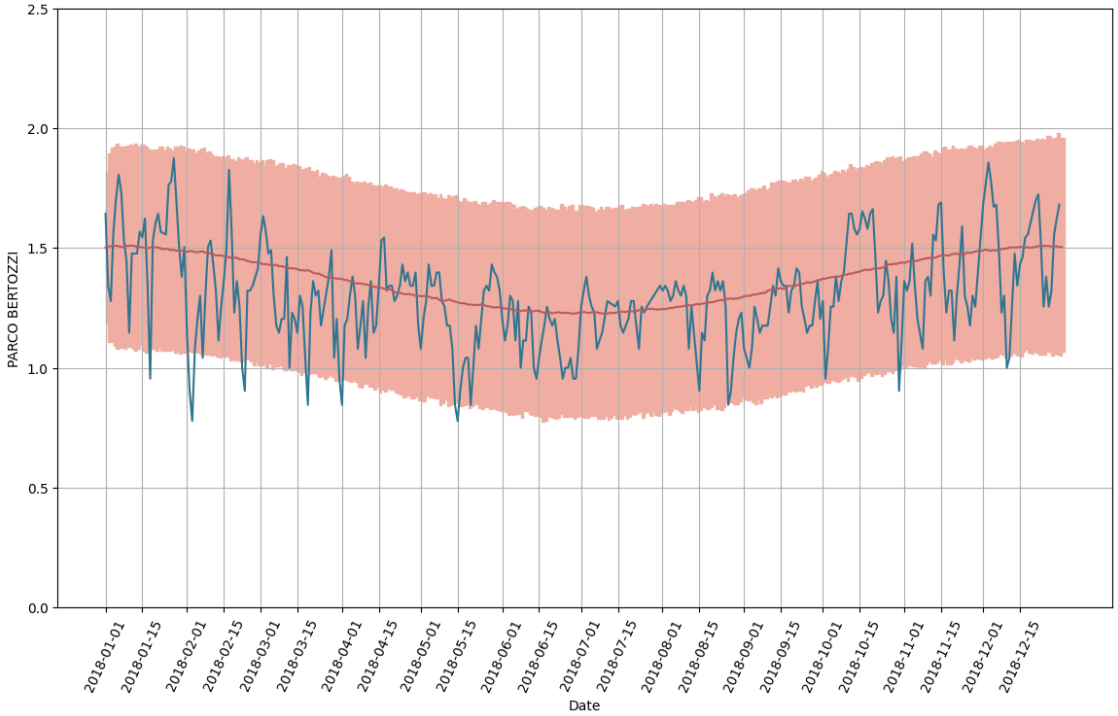

SAN LEO
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

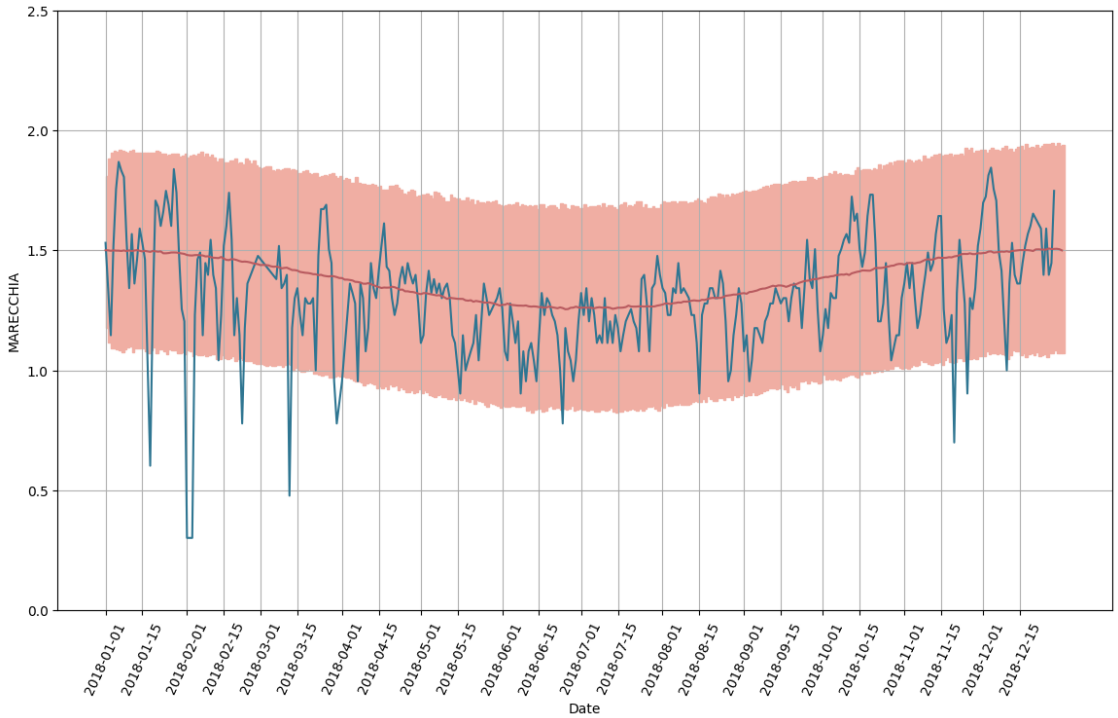

VERUCCHIO
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

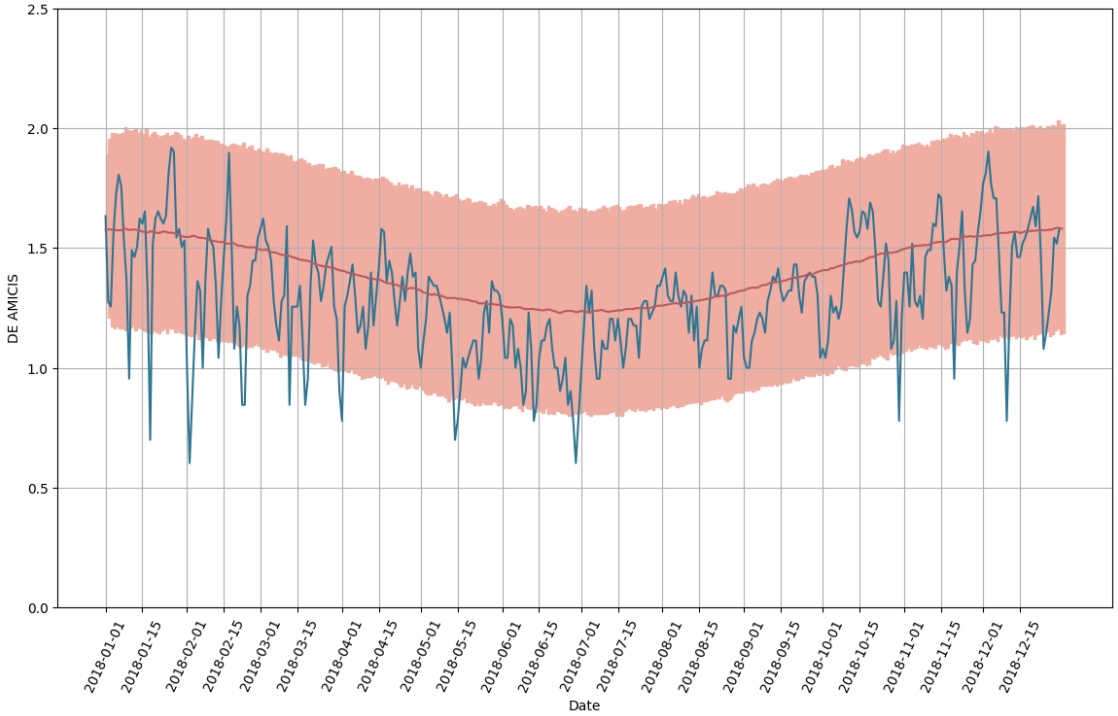

BADIA
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

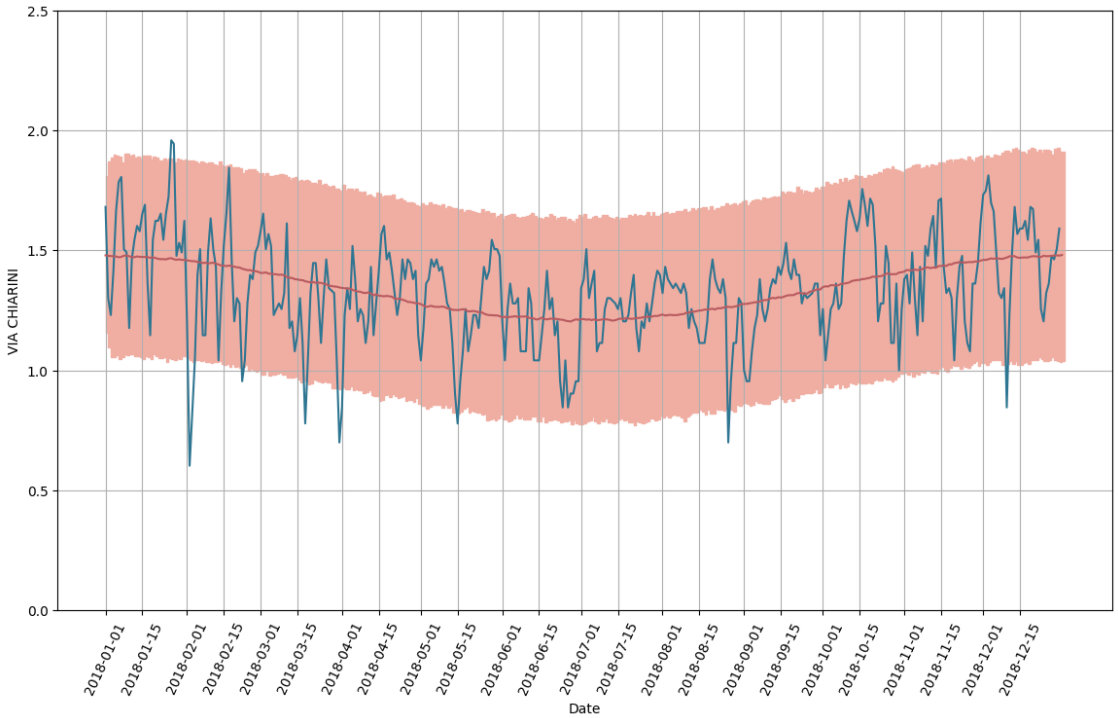

GIARDINI MARGHERITA
Date

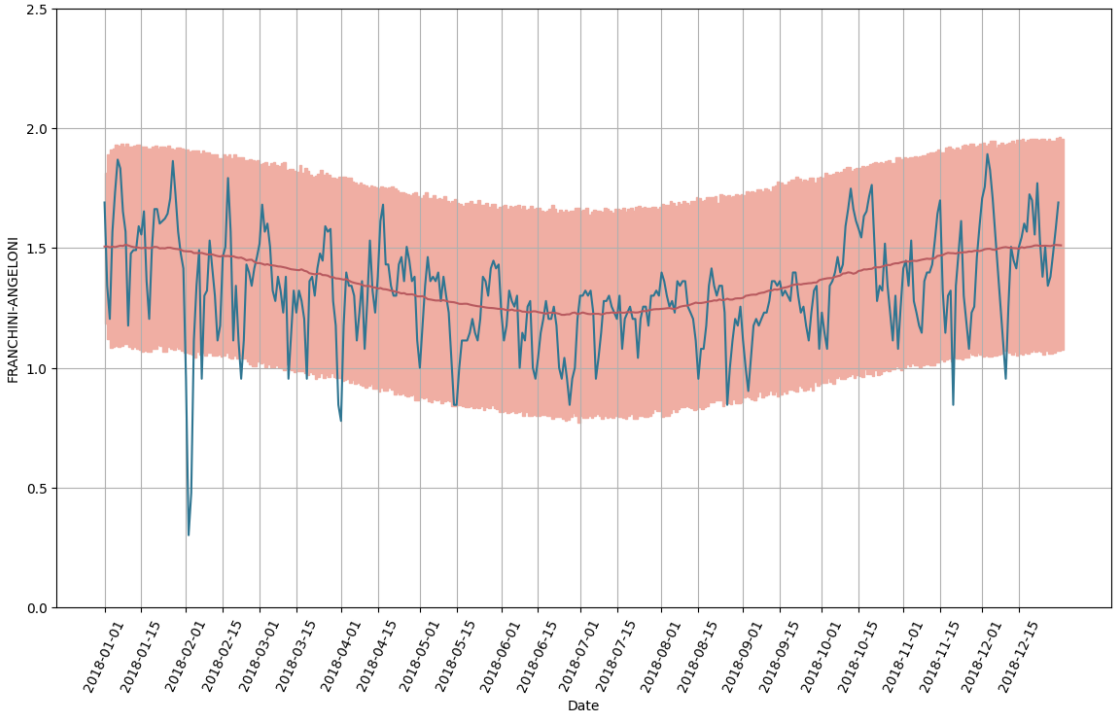

PARCO RESISTENZA
Date

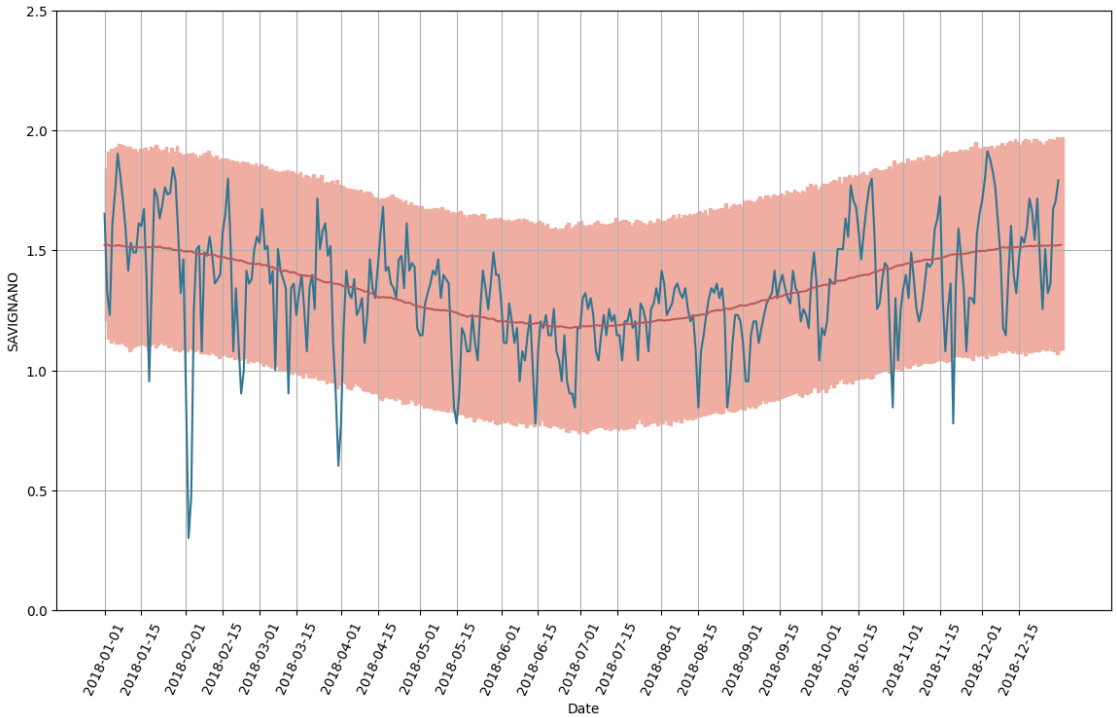

PARCO BERTOZZI
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

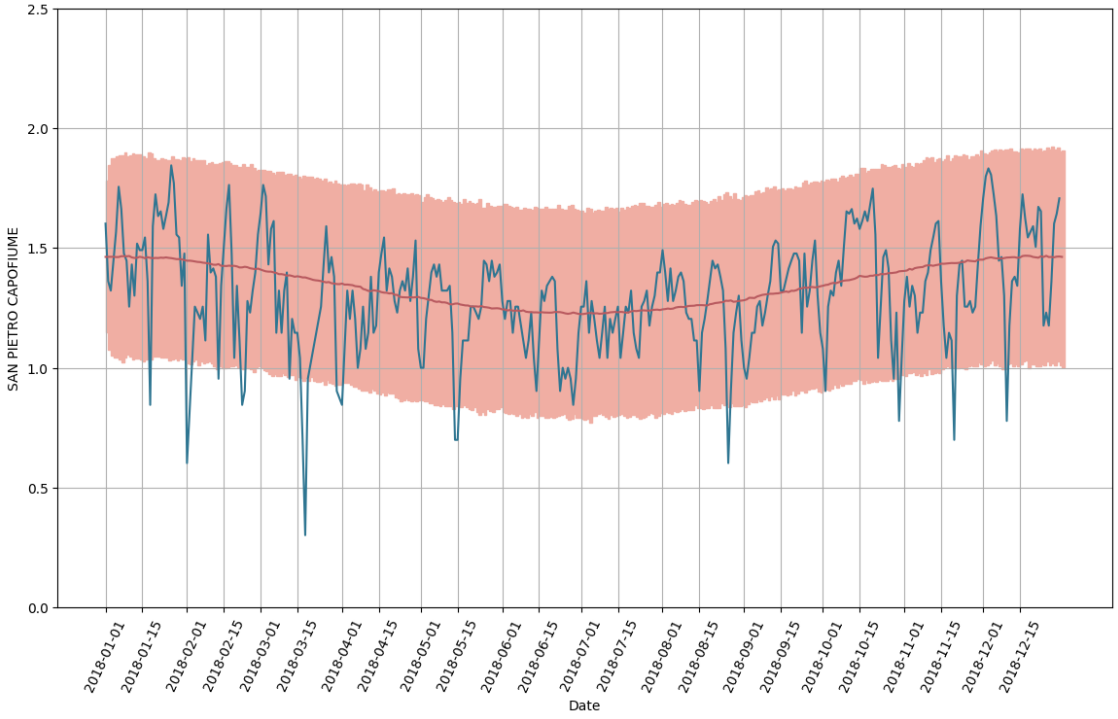

MARECCHIA
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

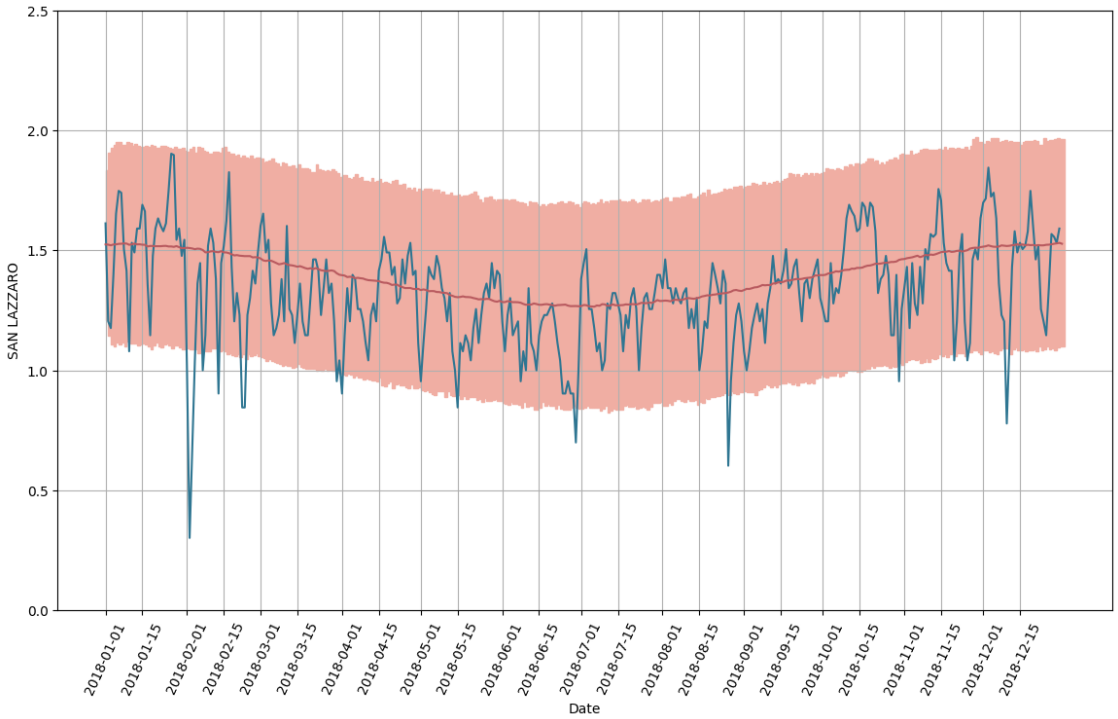

DE AMICIS
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

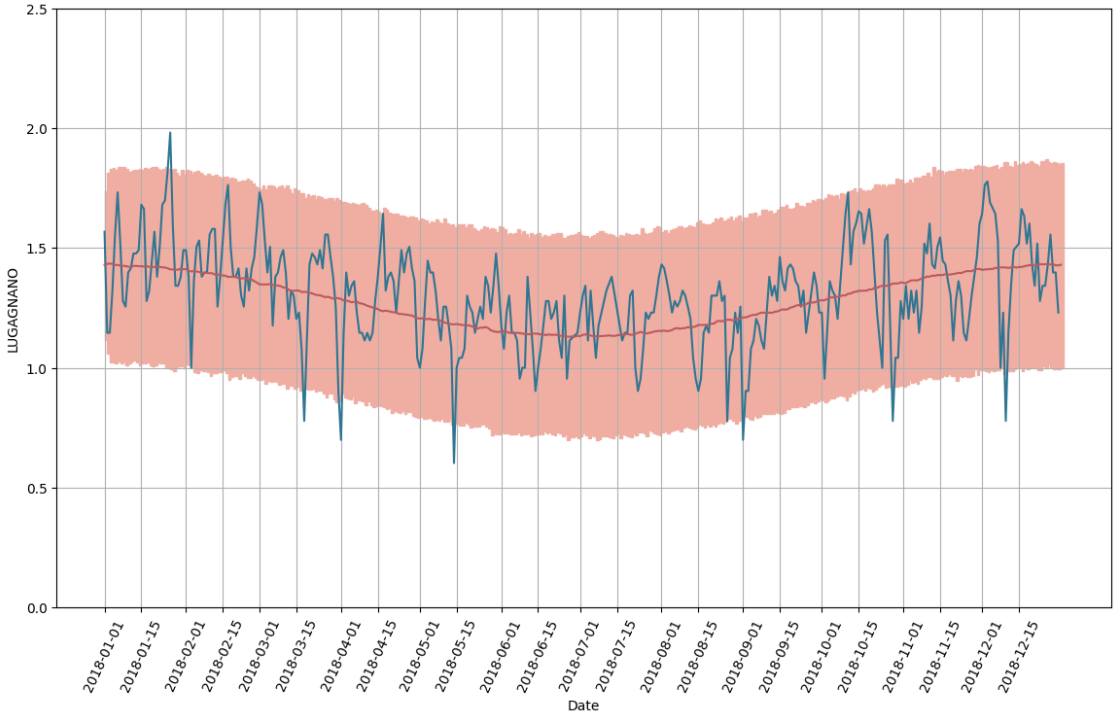

VIA CHIARINI
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

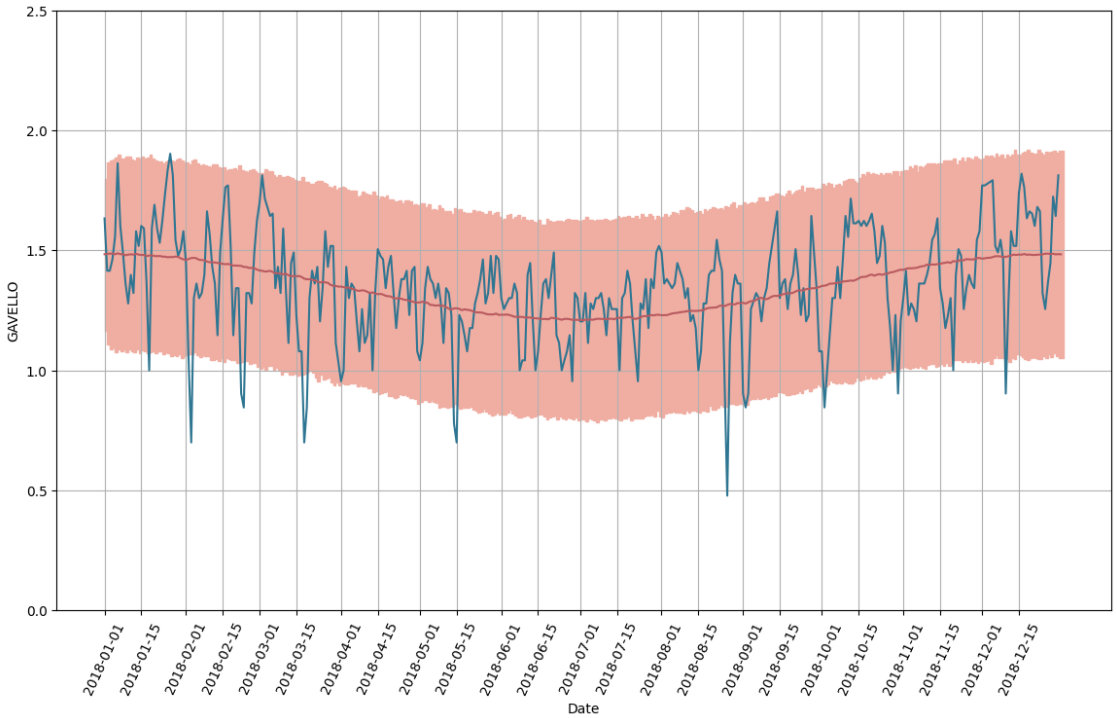

FRANCHINI-ANGELONI
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

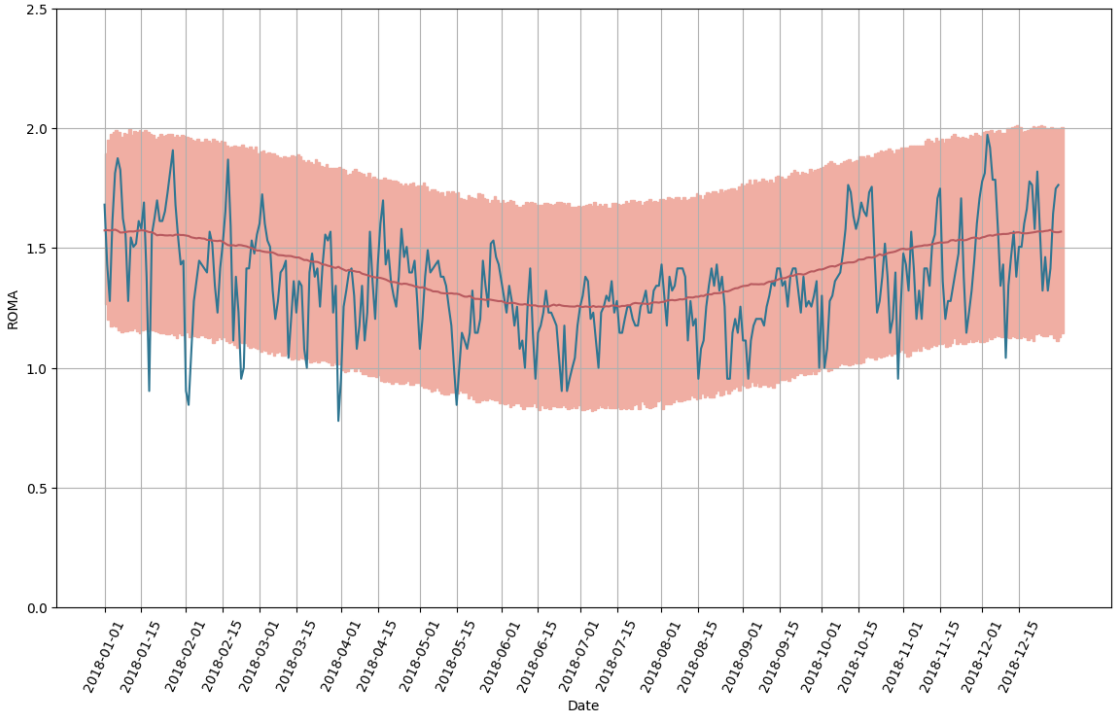

SAVIGNANO
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

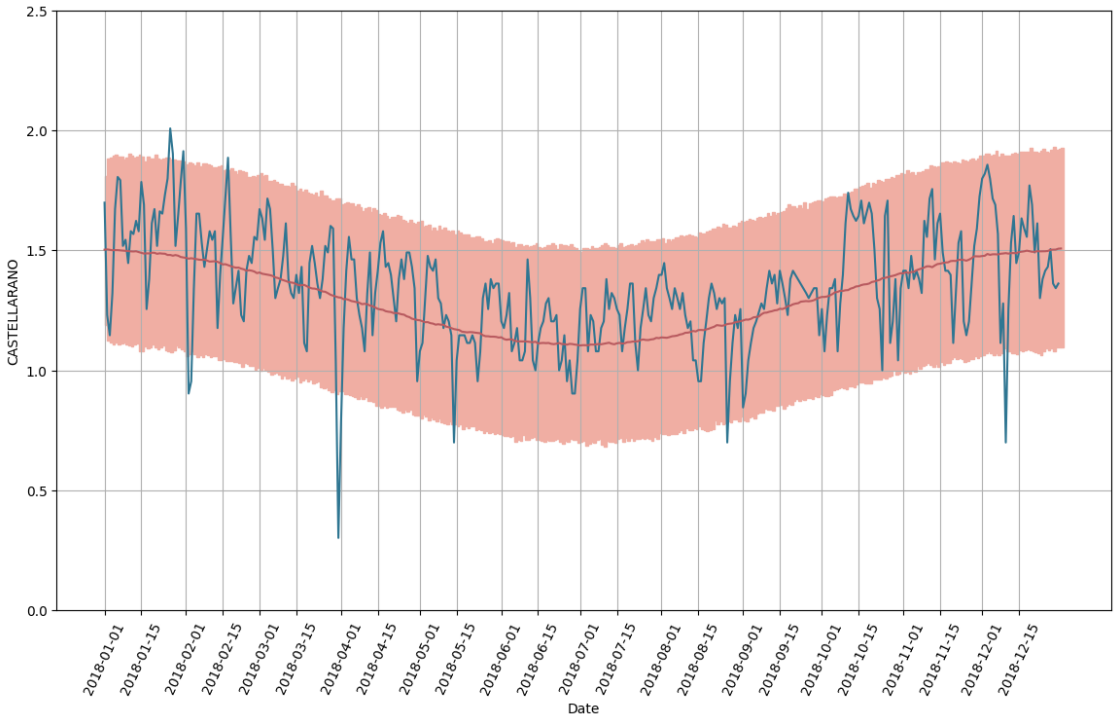

SAN PIETRO CAPOFIUME
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

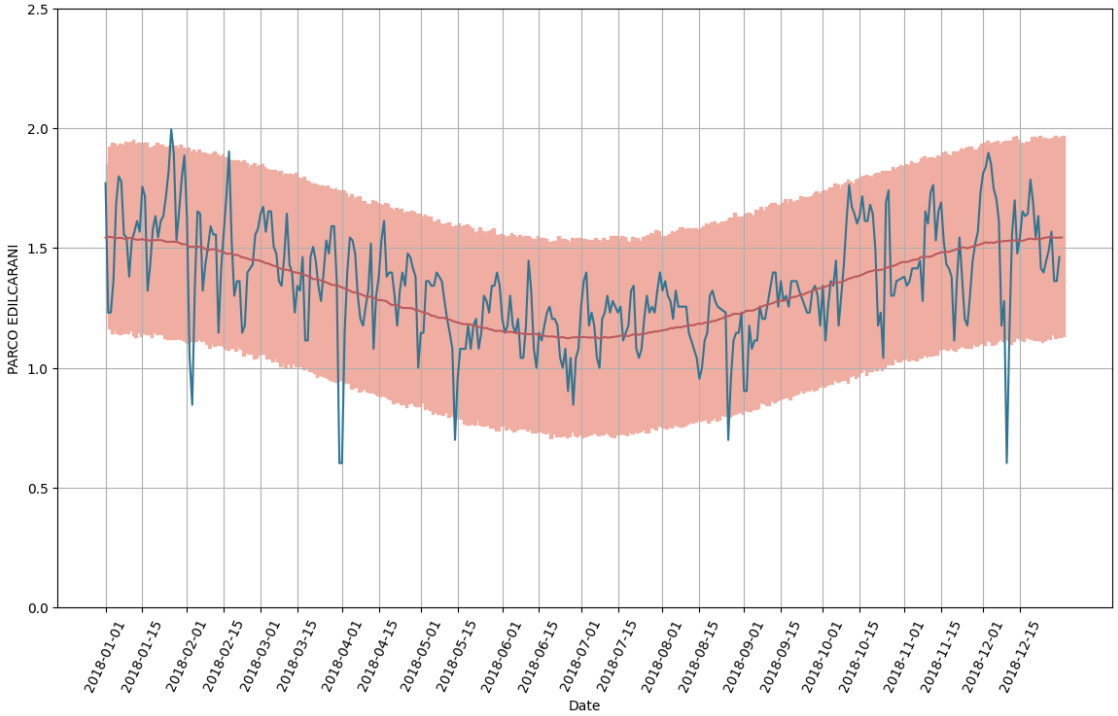

SAN LAZZARO
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

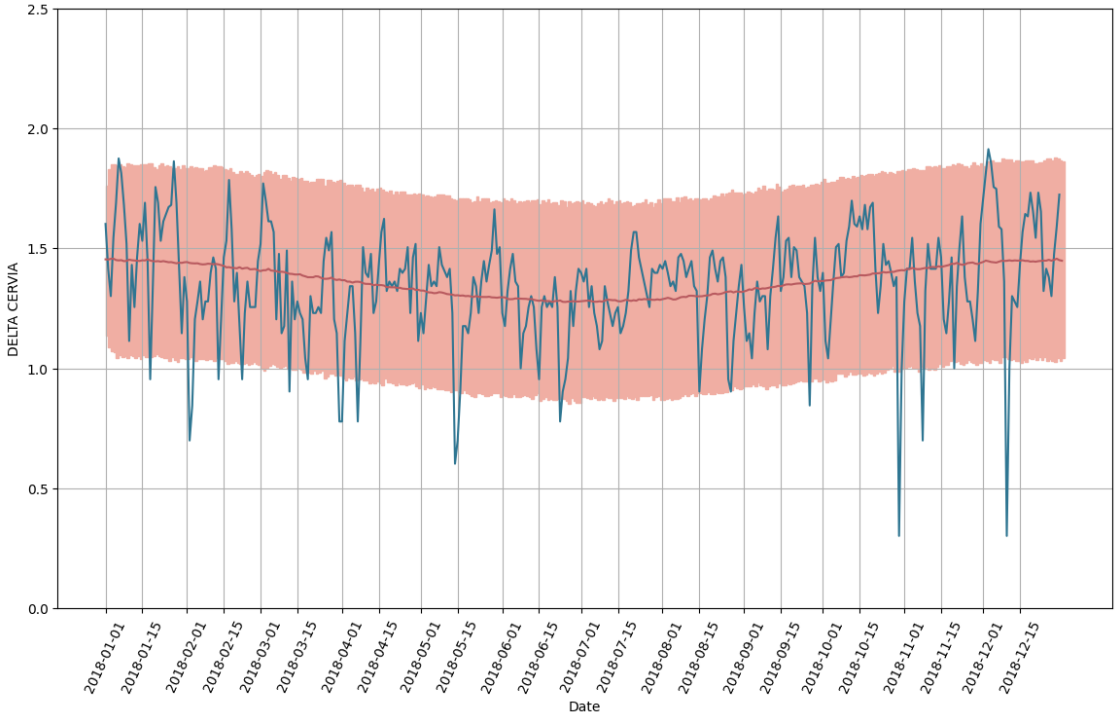

LUGAGNANO
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

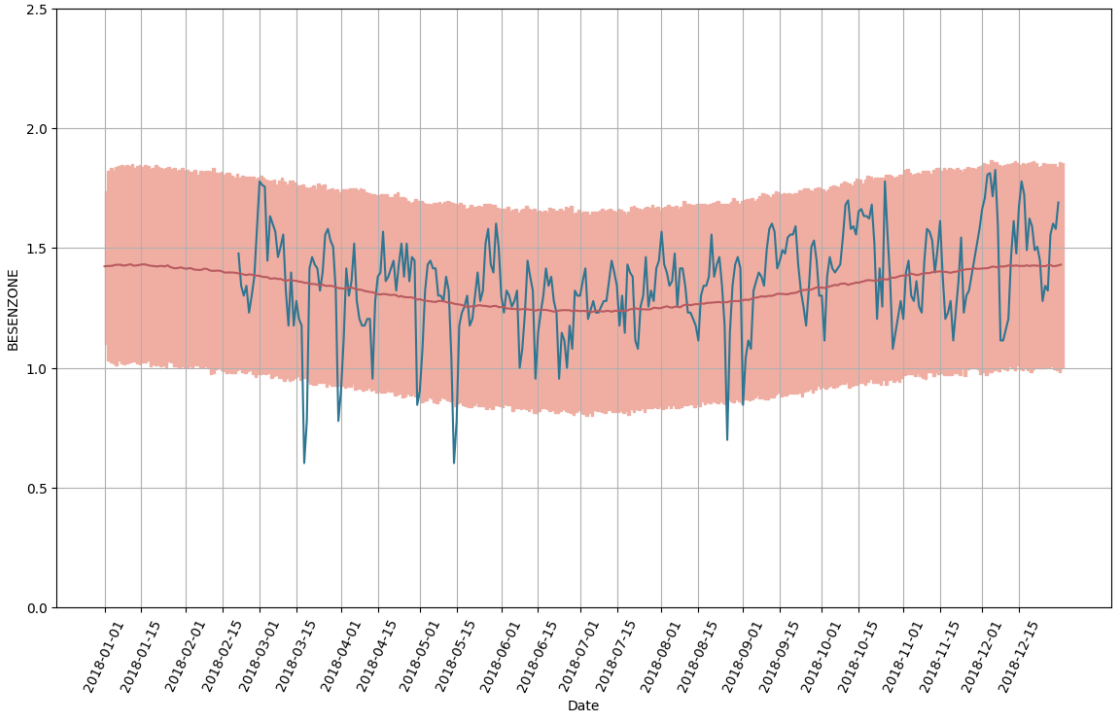

GAVELLO
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

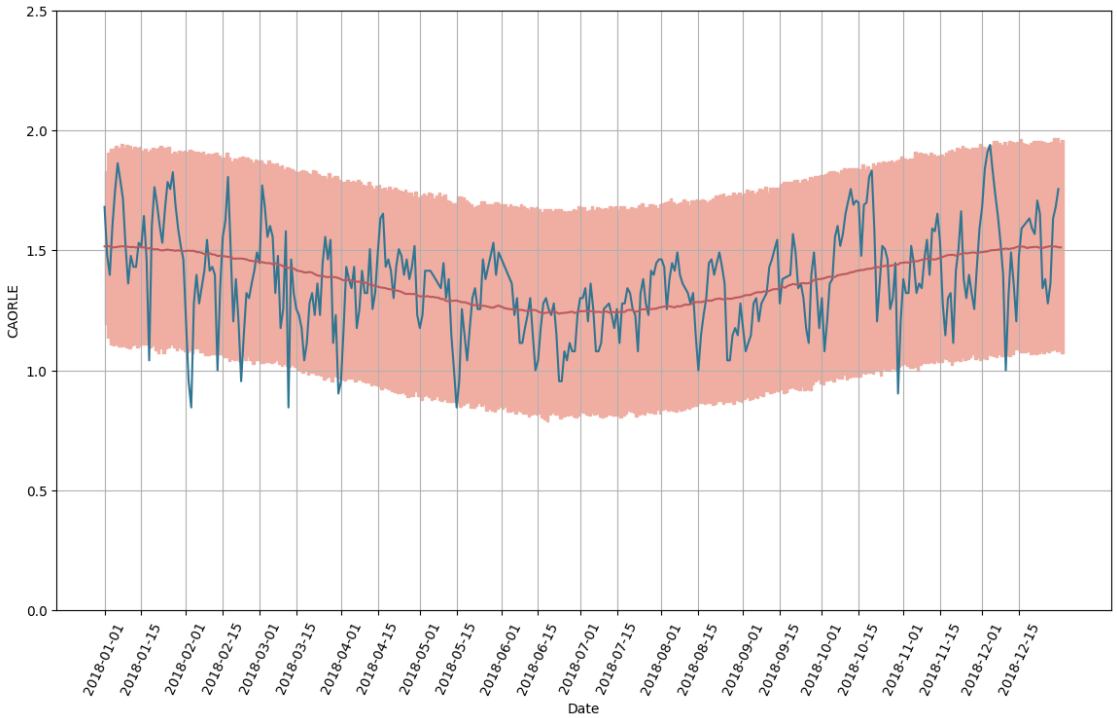

ROMA
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

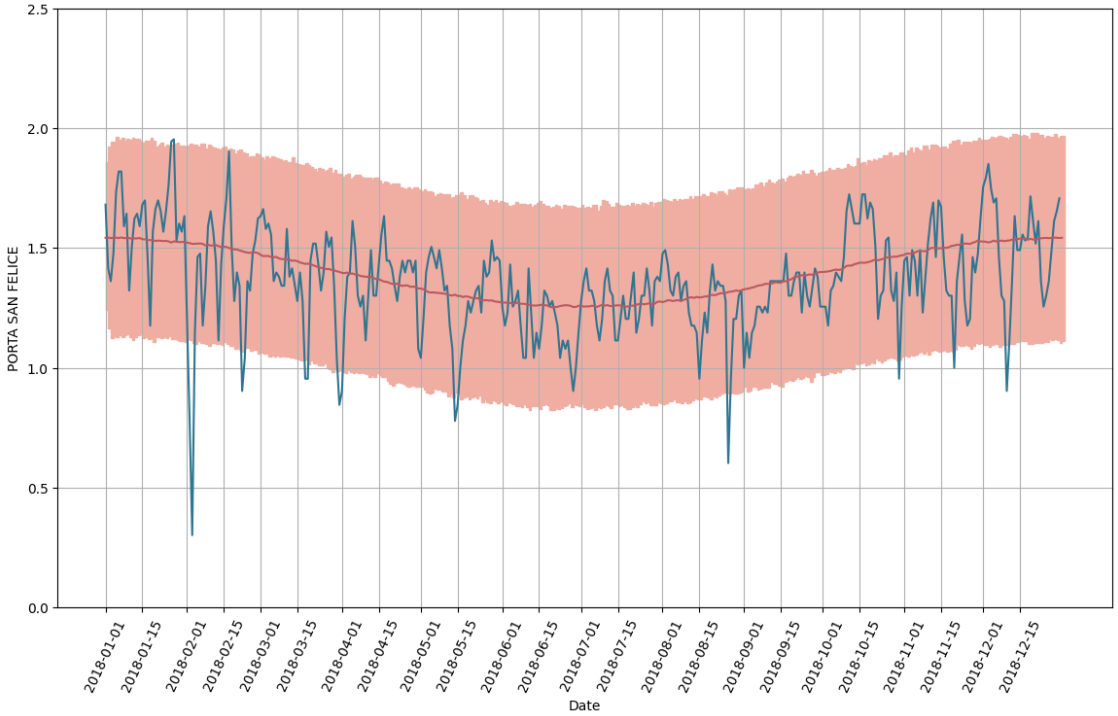

CASTELLARANO
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

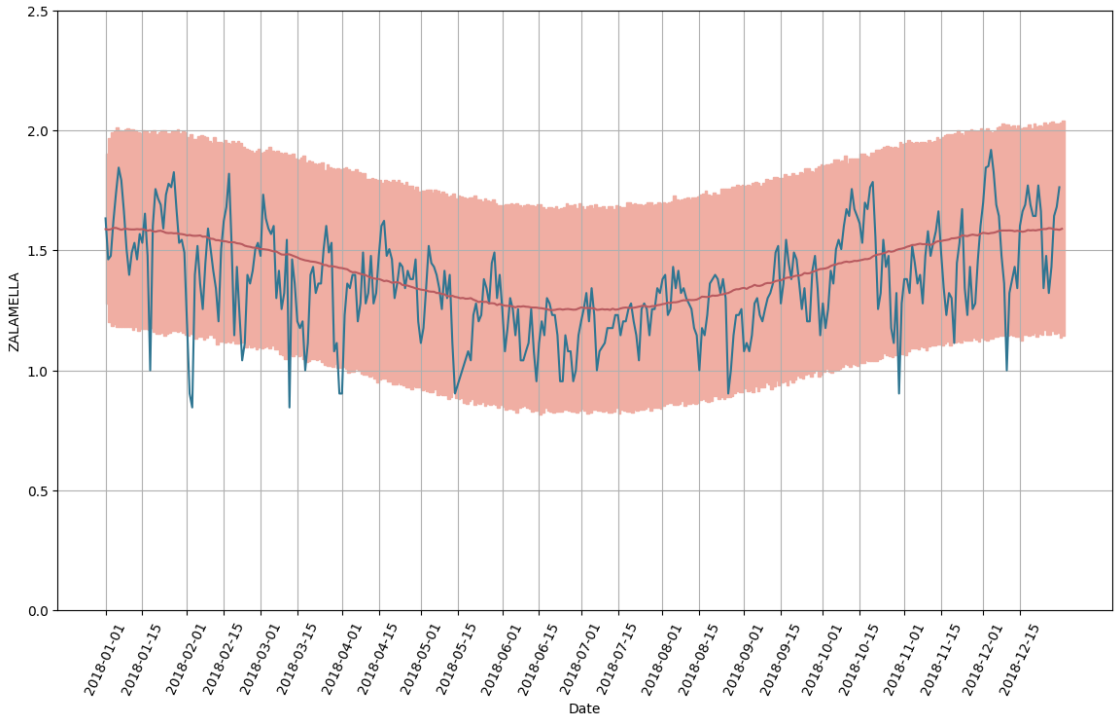

PARCO EDILCARANI
Date

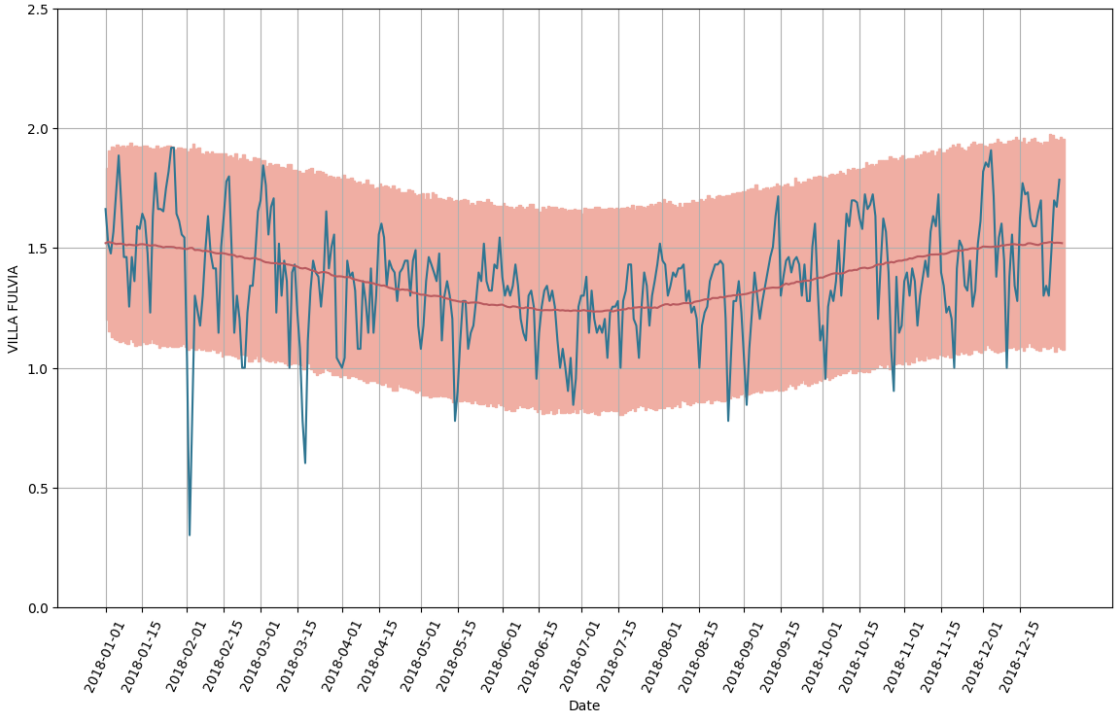

DELTA CERVIA
Date

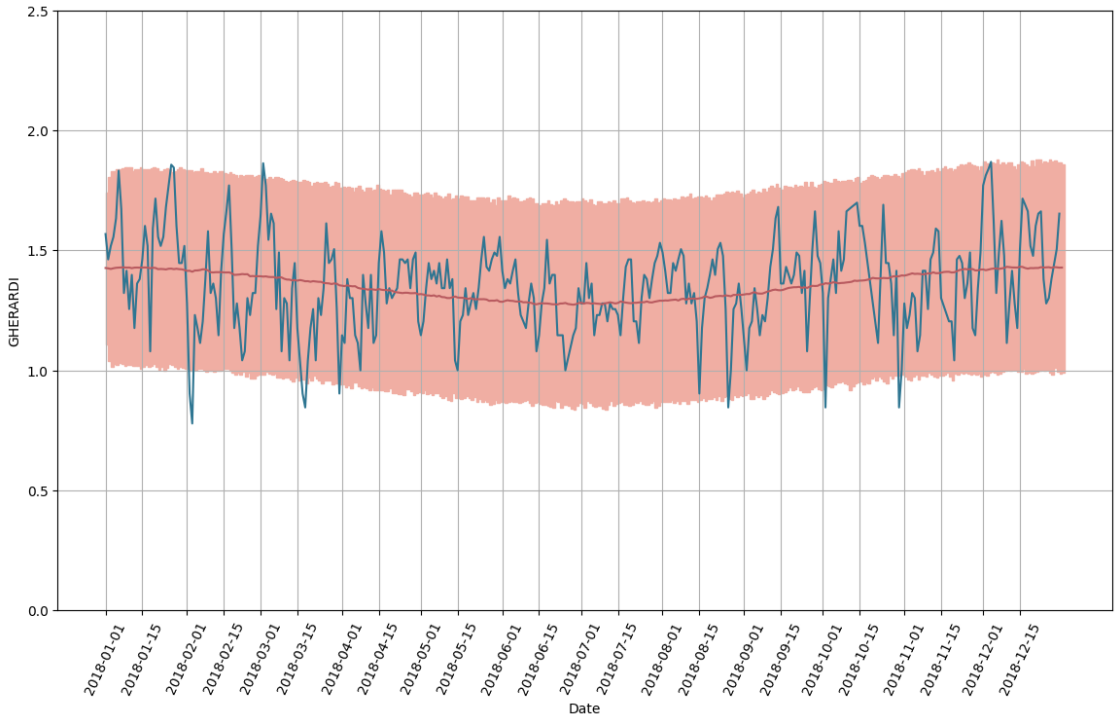

BESENZONE
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

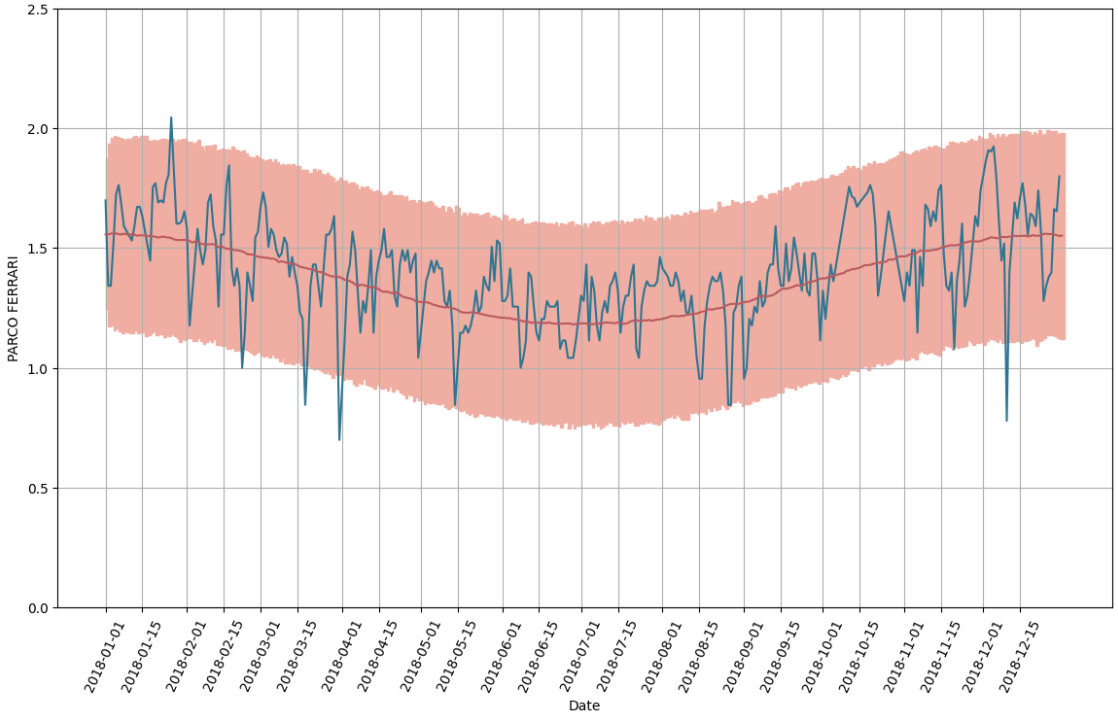

CAORLE
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

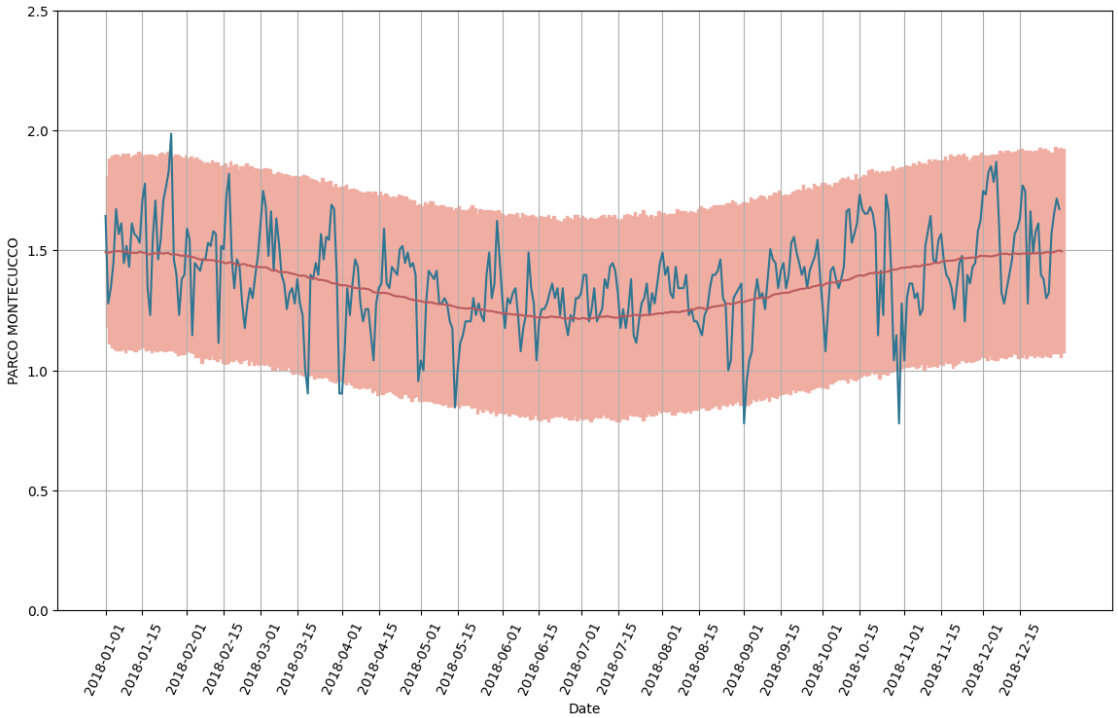

PORTA SAN FELICE
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

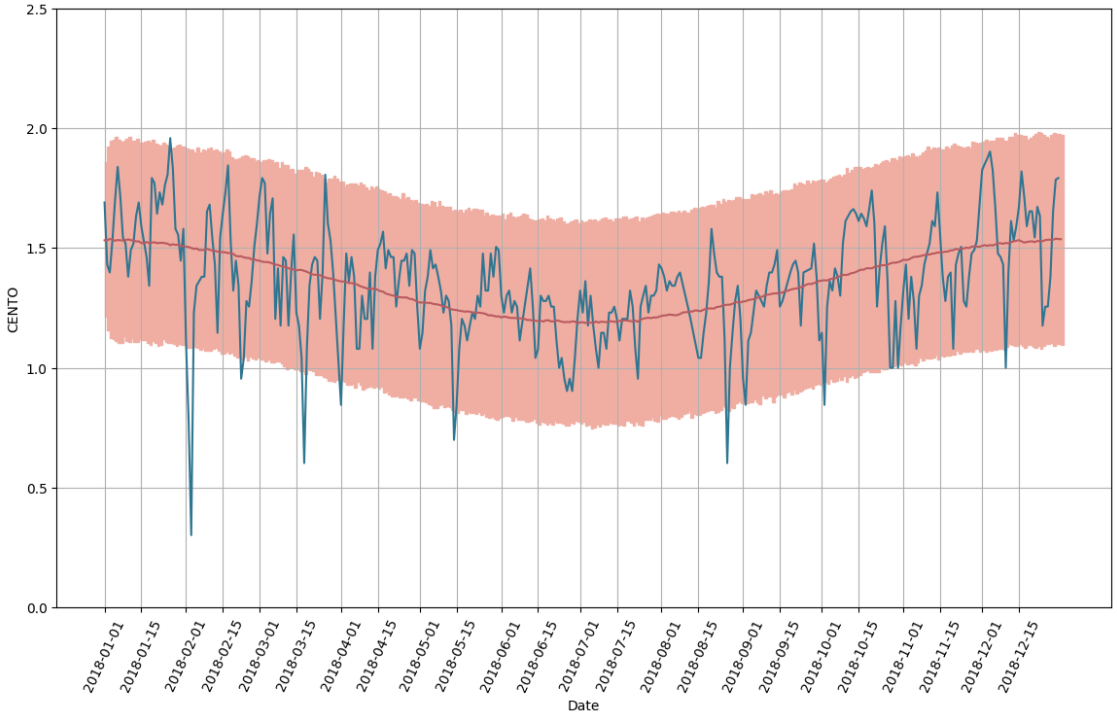

ZALAMELLA
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

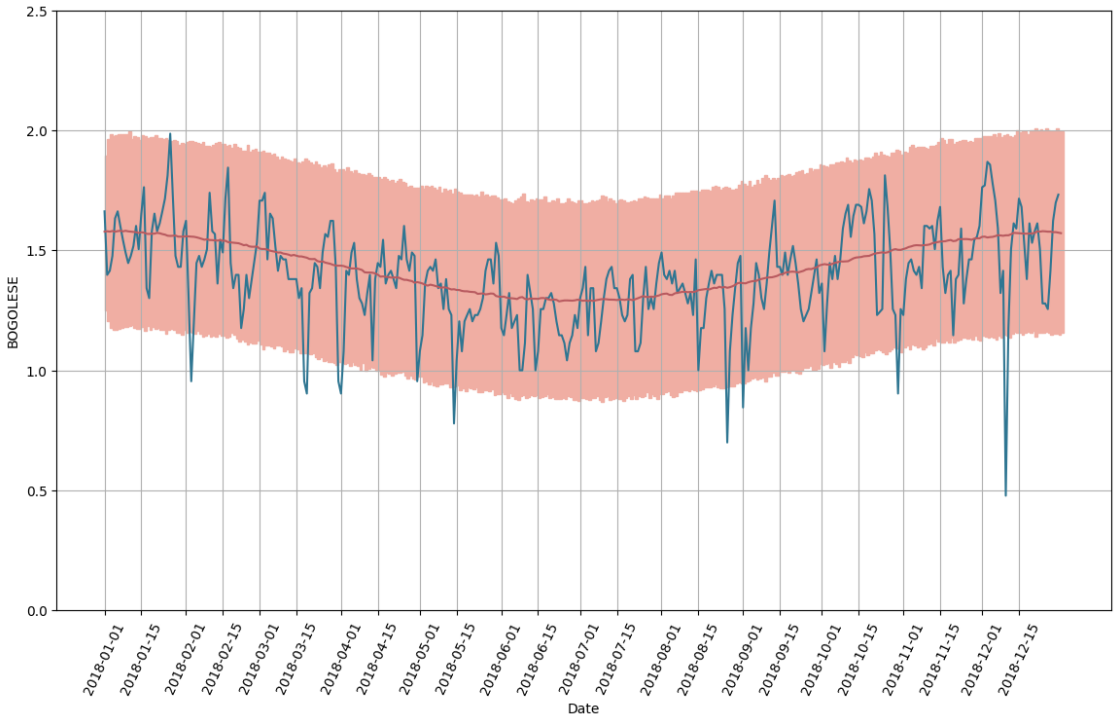

VILLA FULVIA
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

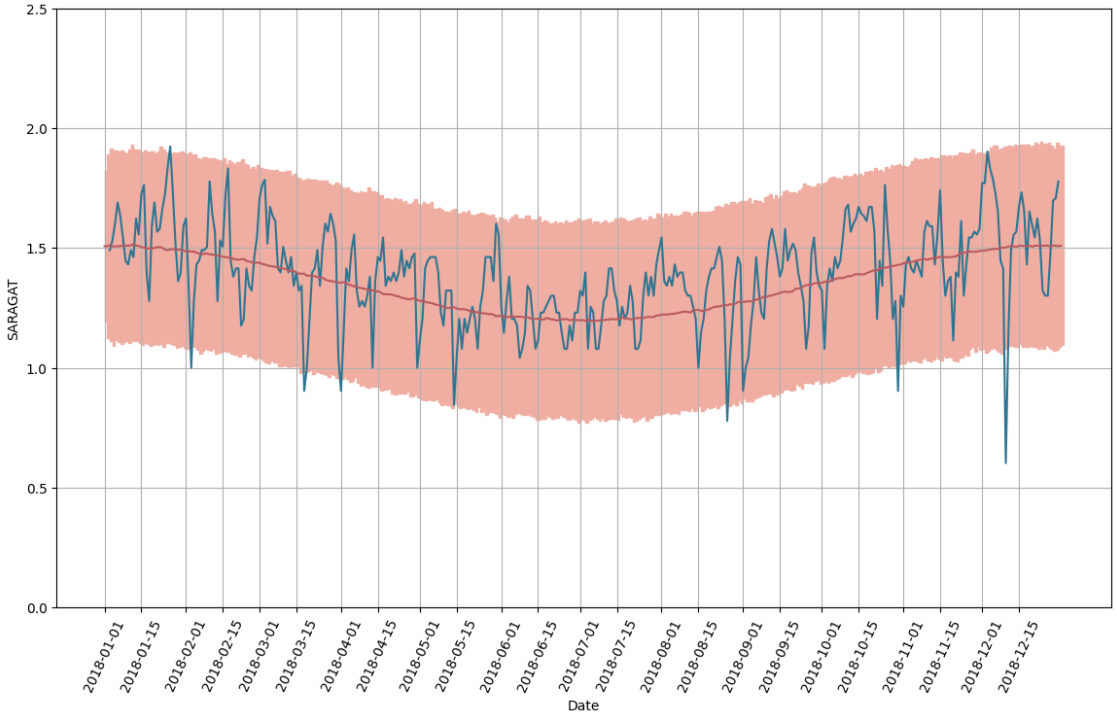

GHERARDI
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

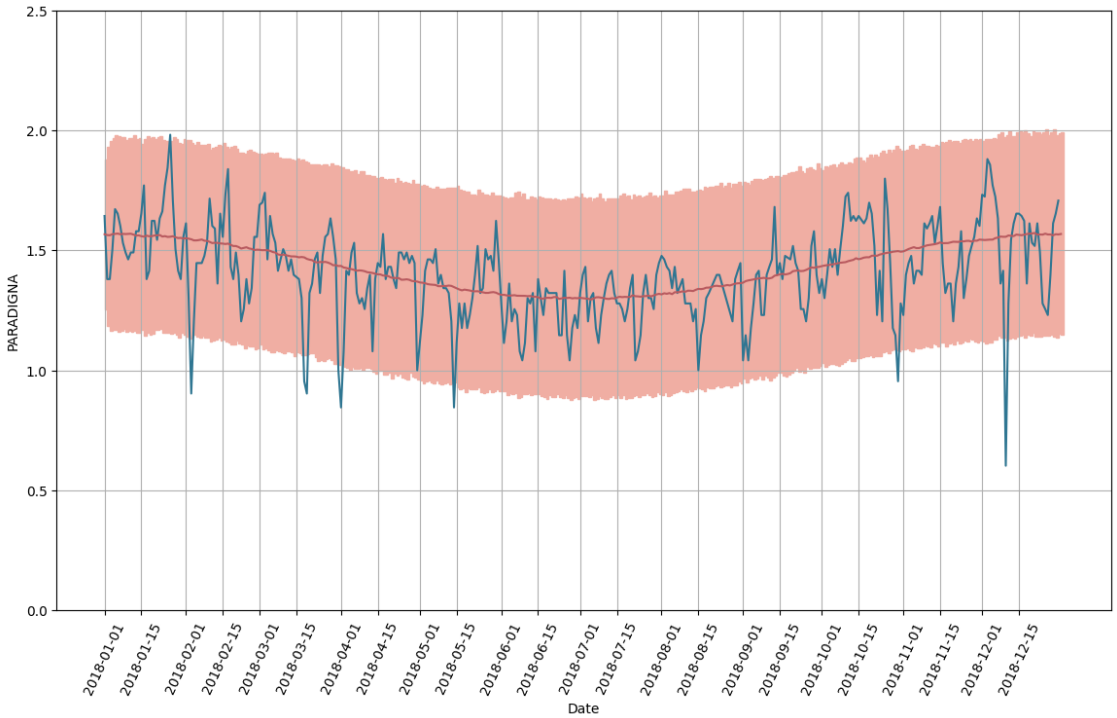

PARCO FERRARI
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

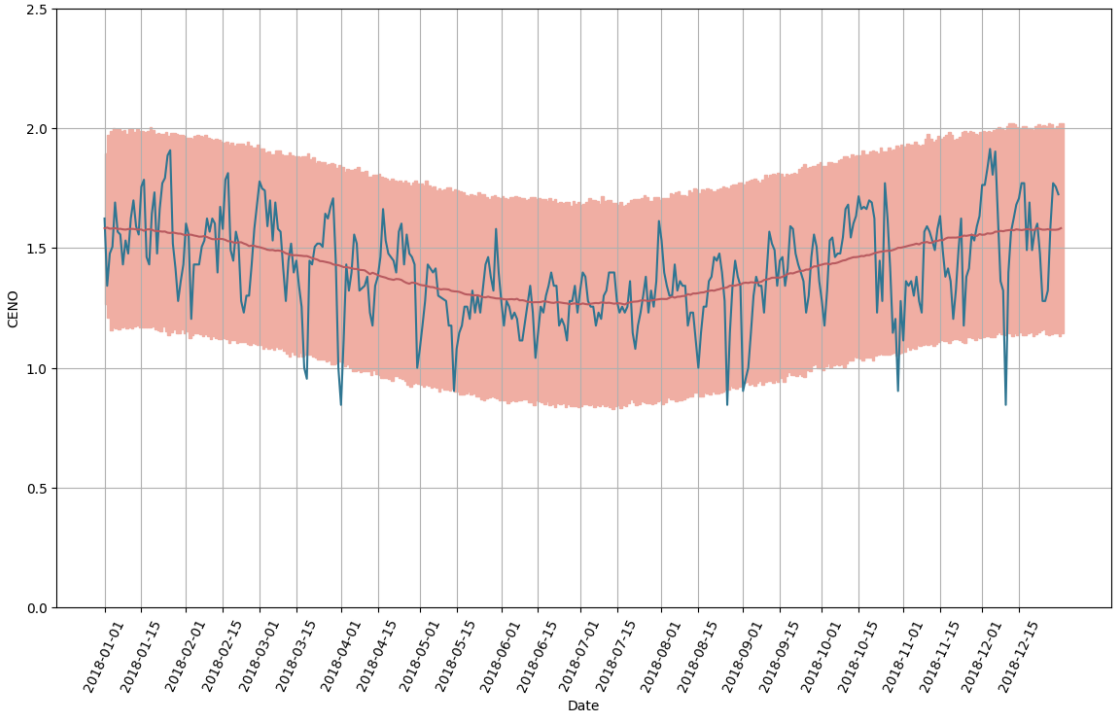

PARCO MONTECUCCO
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date

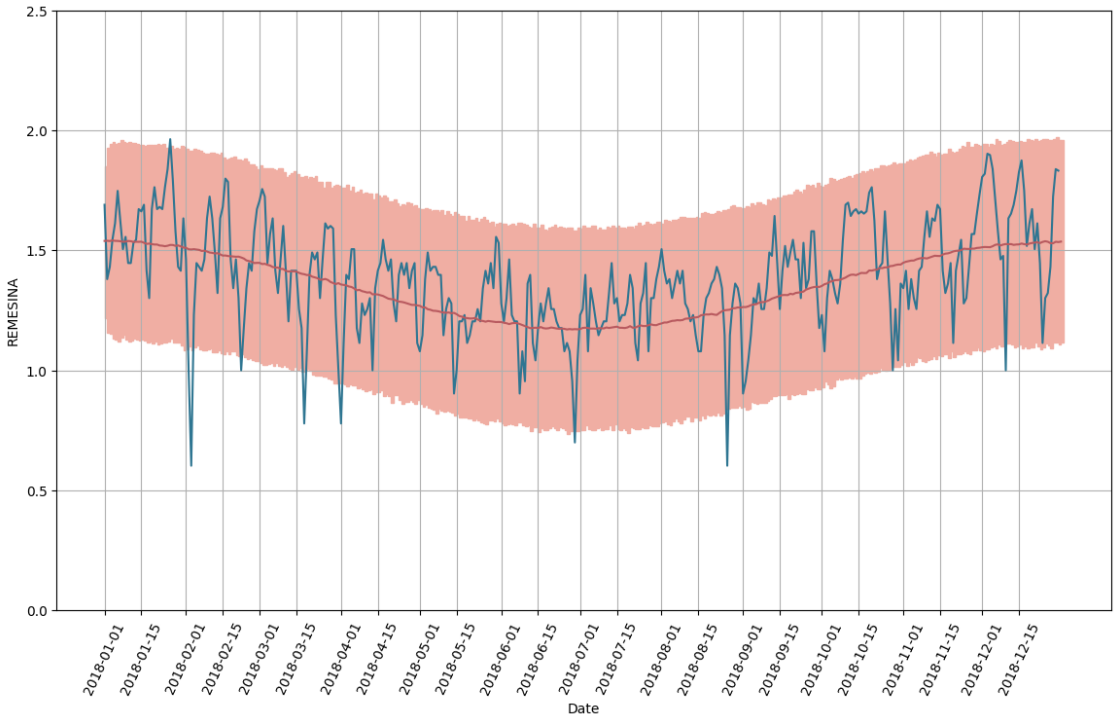

CENTO
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

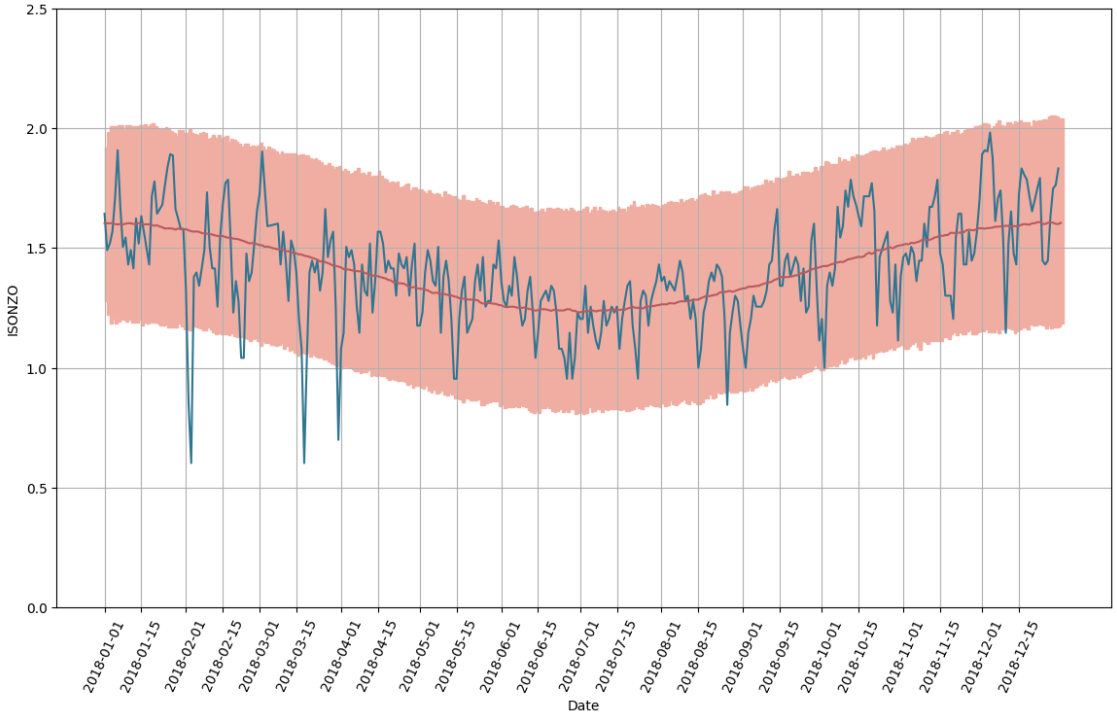

BOGOLESE
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

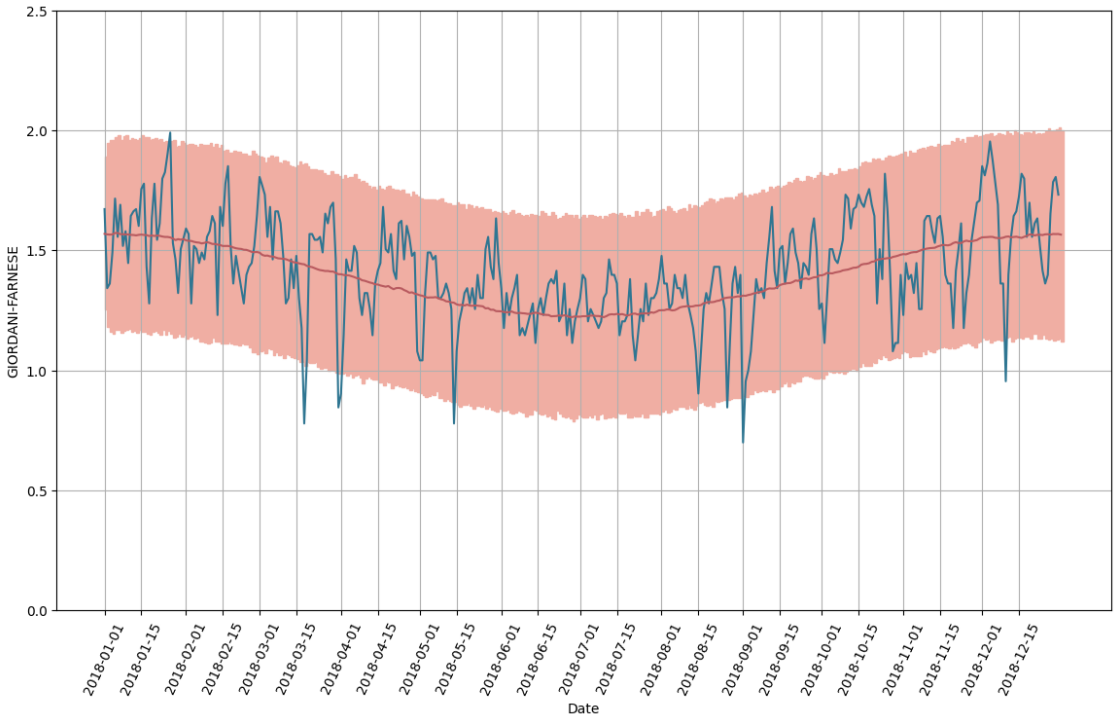

SARAGAT
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

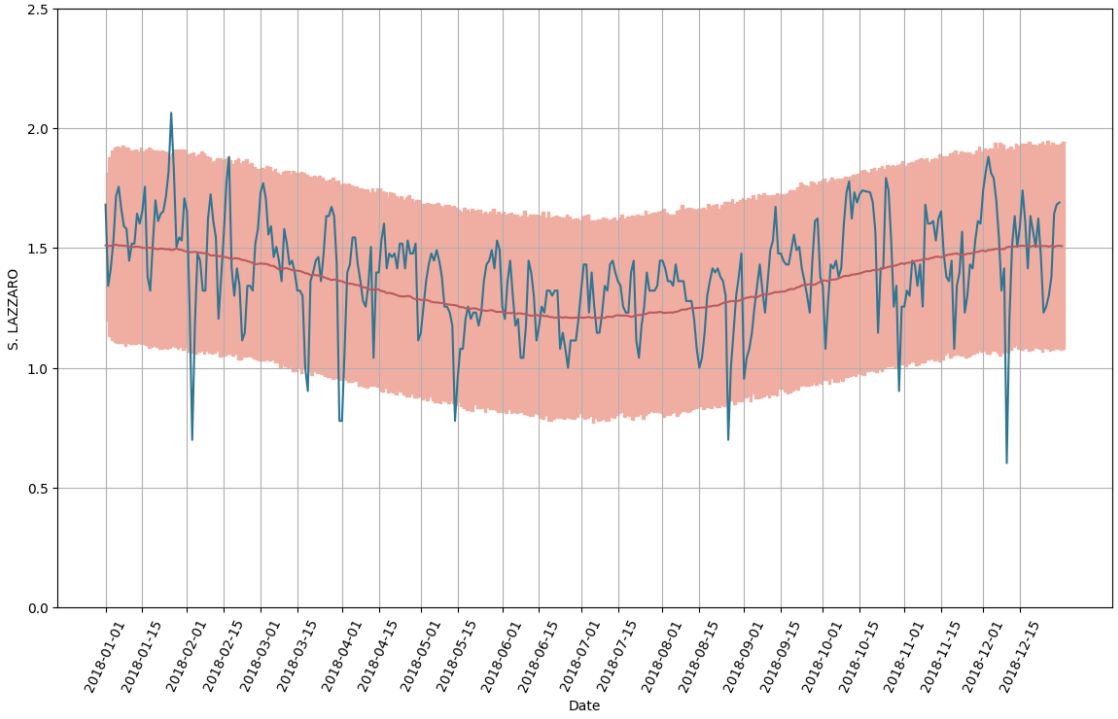

PARADIGNA
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

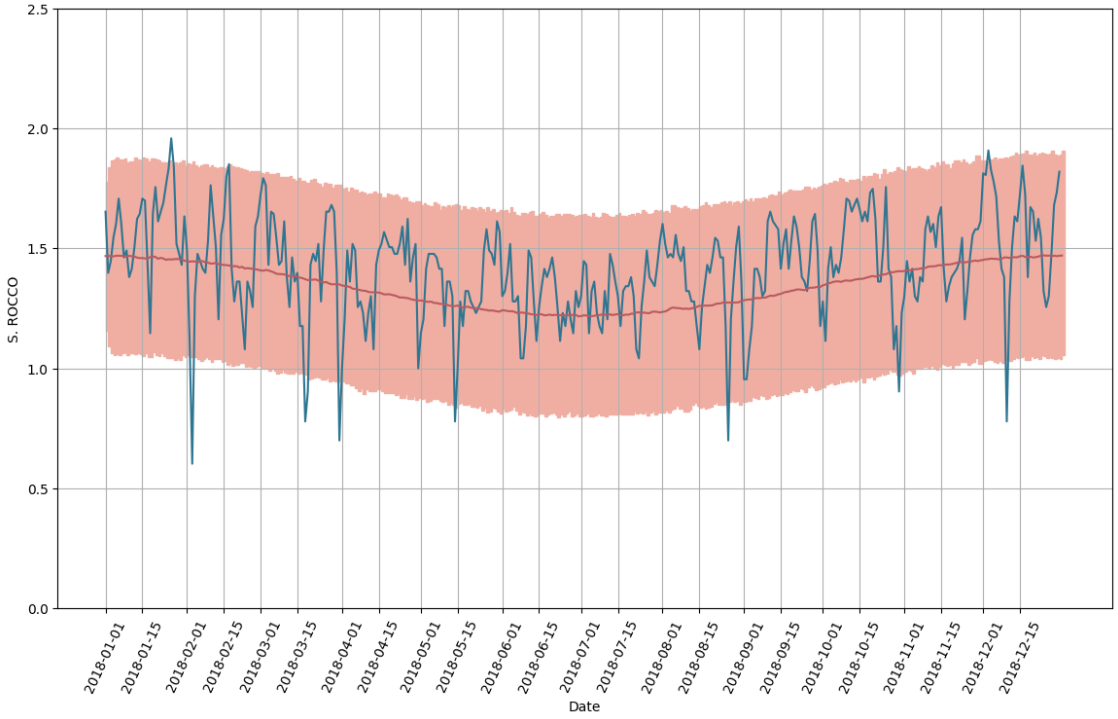

CENO
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

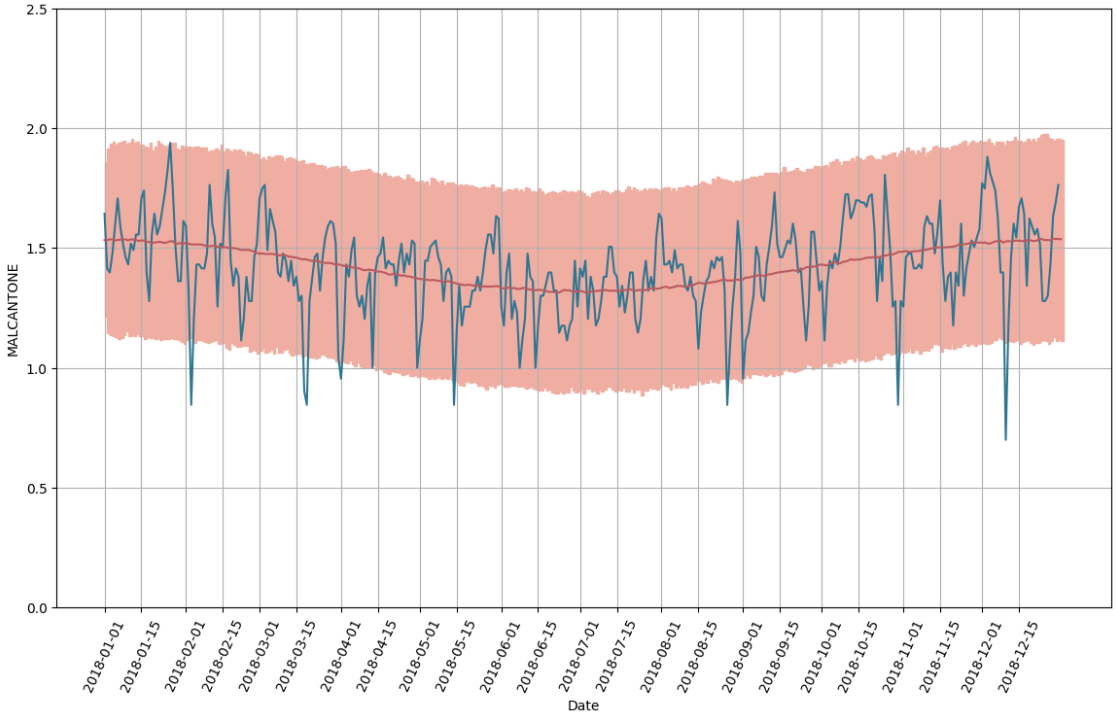

REMESINA
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

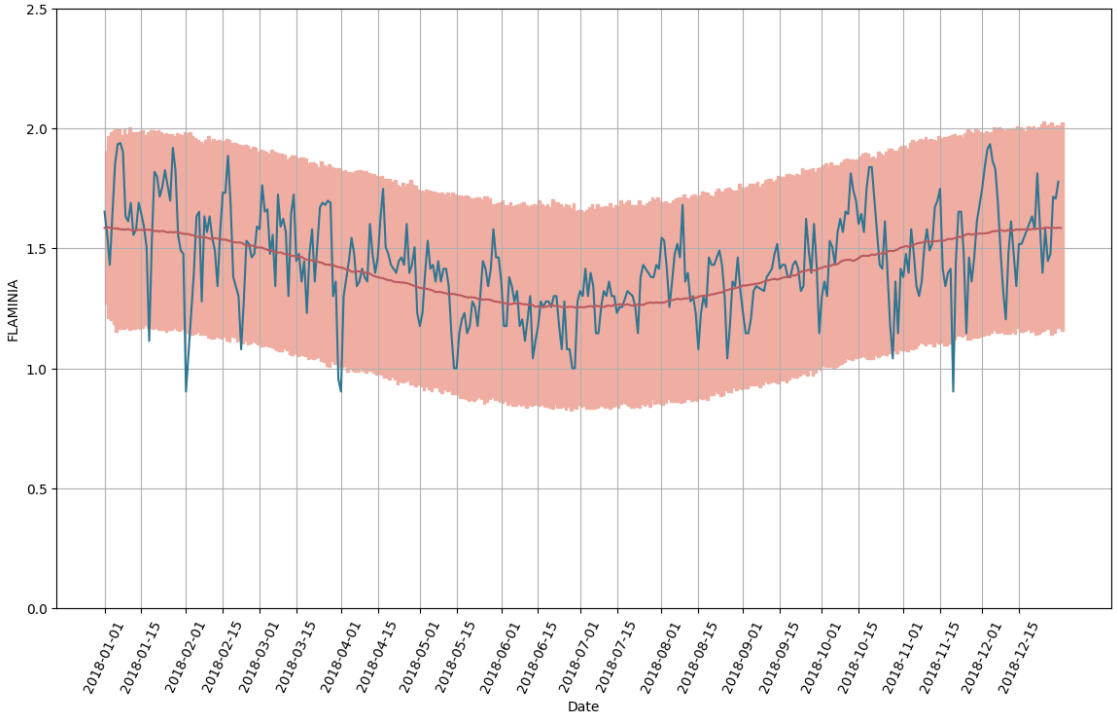

ISONZO
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

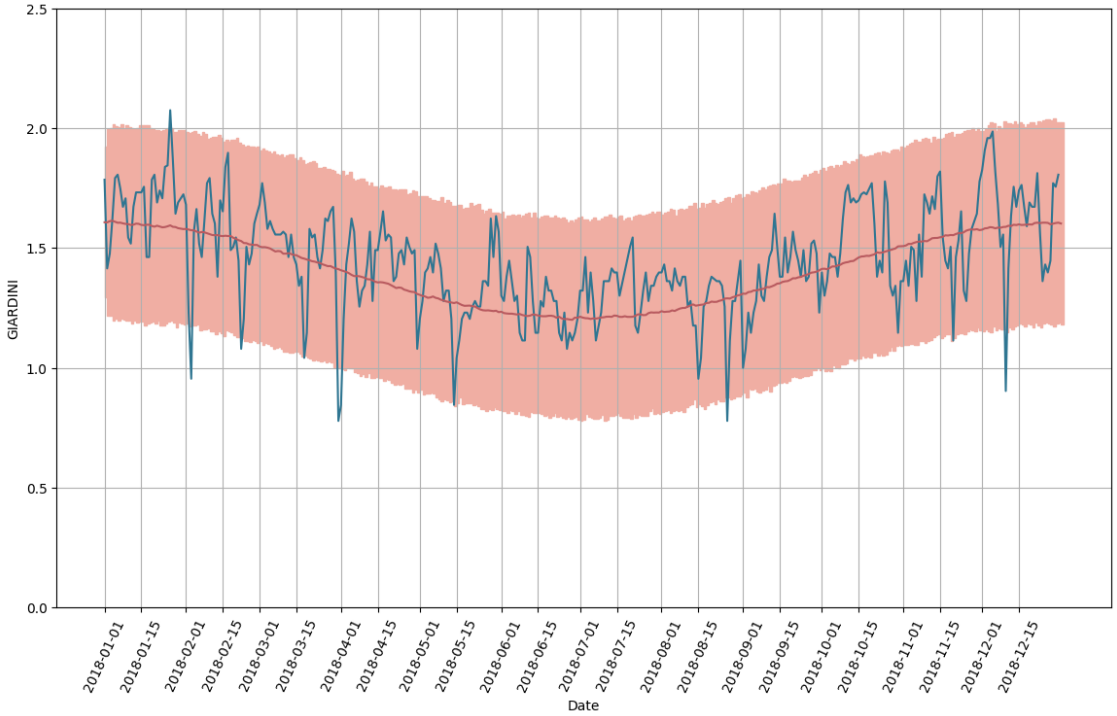

GIORDANI-FARNESE
Date
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15

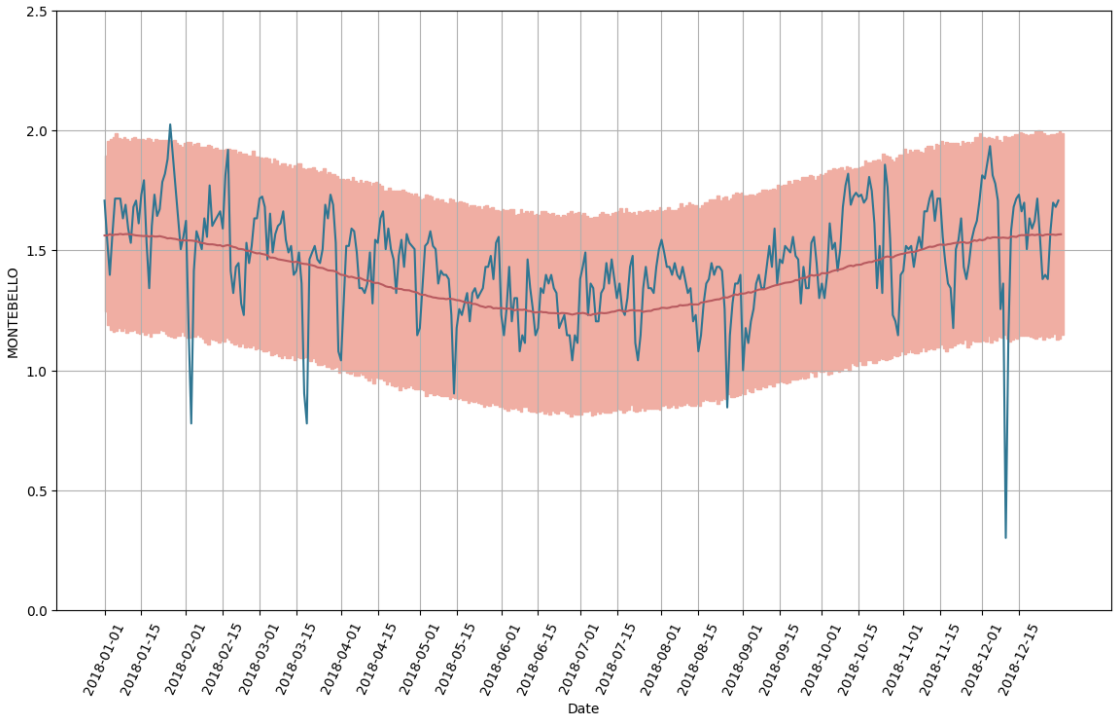

S. LAZZARO
Date

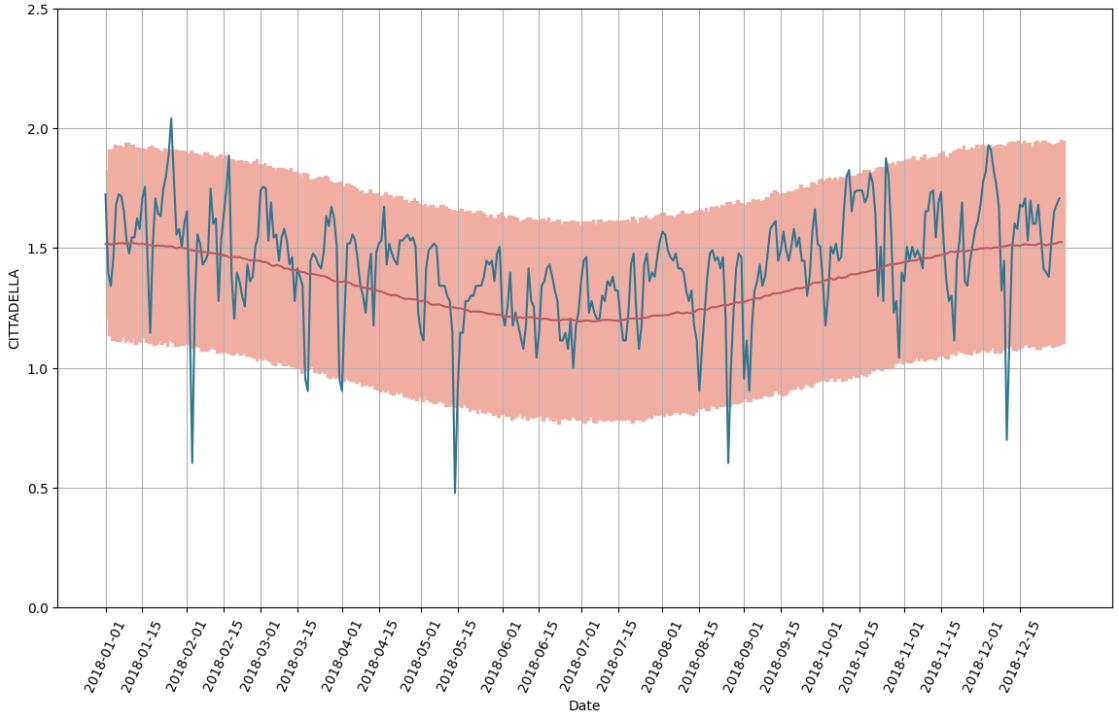

S. ROCCO
Date

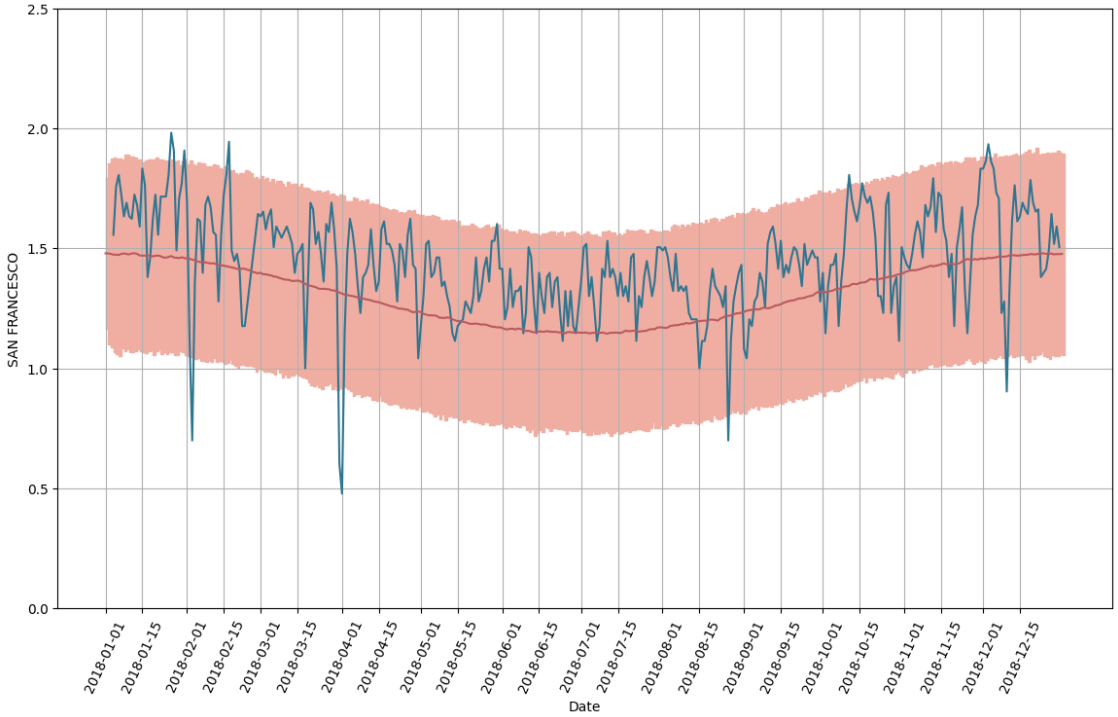

MALCANTONE
Date

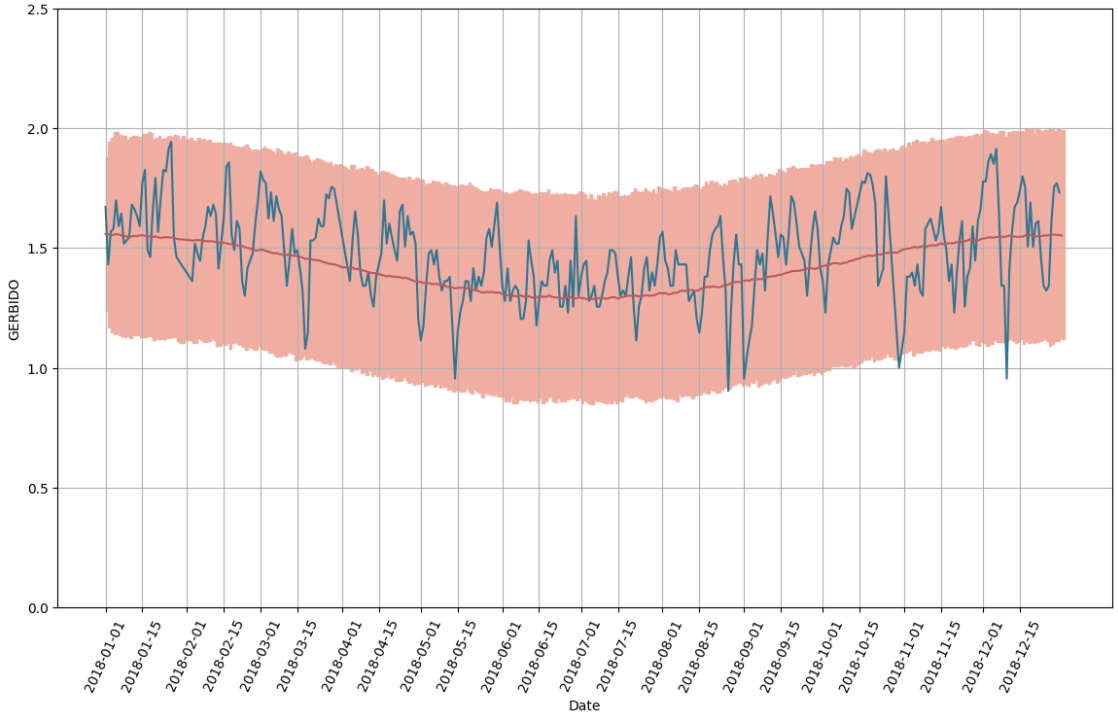

FLAMINIA
Date

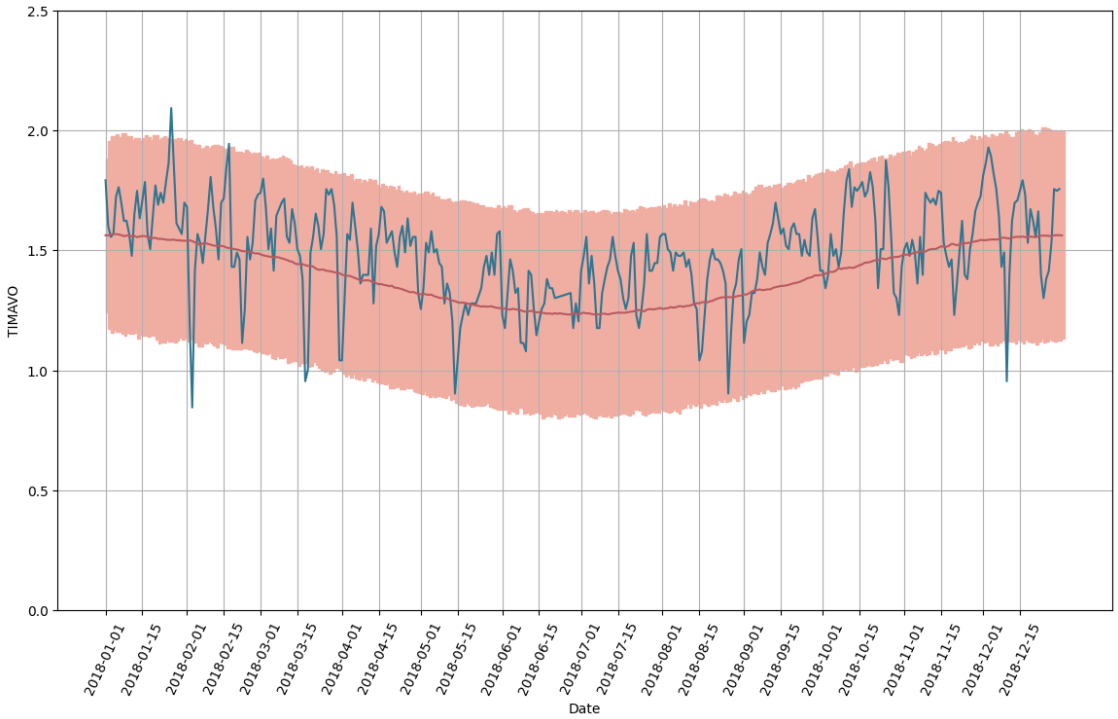

GIARDINI
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date


MONTEBELLO
2.5
2.0
1.5
1.0
0.5
0.0
2018-01-01
2018-01-15
2018-02-01
2018-02-15
2018-03-01
2018-03-15
2018-04-01
2018-04-15
2018-05-01
2018-05-15
2018-06-01
2018-06-15
2018-07-01
2018-07-15
2018-08-01
2018-08-15
2018-09-01
2018-09-15
2018-10-01
2018-10-15
2018-11-01
2018-11-15
2018-12-01
2018-12-15
Date


CITTADELLA
Date


SAN FRANCESCO
Date


GERBIDO
Date


TIMAVO
Date

[ ]:

[ ]: