reg_year_r_random_points_new_200

March 5, 2024

0.1 Importing

[ ]:
```
import xarray as xr
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn import preprocessing

from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import BaggingRegressor

from sklearn.metrics import mean_squared_error as mse

from tqdm.auto import tqdm

import dill
import random

import salishsea_tools.viz_tools as sa_vi
```

0.2 Datasets Preparation

[ ]:
```
def datasets_preparation(dataset):

    drivers = np.stack([np.ravel(dataset['Temperature_(0m-15m)']),
        np.ravel(dataset['Temperature_(15m-100m)']), np.
 ↪ravel(dataset['Salinity_(0m-15m)']),
        np.ravel(dataset['Salinity_(15m-100m)'])])
    indx = np.where(~np.isnan(drivers).any(axis=0))
    drivers = drivers[:,indx[0]]

    diat = np.ravel(dataset['Diatom'])
    diat = diat[indx[0]]

    return(drivers, diat, indx)
```

0.3 Regressor

[ ]:
```
def regressor (inputs, targets):

    inputs = inputs.transpose()

    # Regressor
    scale = preprocessing.StandardScaler()
    inputs = scale.fit_transform(inputs)
    X_train, _, y_train, _ = train_test_split(inputs, targets, train_size=0.35)

    drivers = None
    diat = None

    inputs = None
    targets = None

    model =MLPRegressor()
    regr = BaggingRegressor(model, n_estimators=12, n_jobs=4).fit(X_train,␣
 ↪y_train)

    return (regr)
```

0.4 Regressor 2

[ ]:
```
def regressor2 (inputs, targets, variable_name):

    inputs = inputs.transpose()

    # Regressor
    scale = preprocessing.StandardScaler()
    inputs2 = scale.fit_transform(inputs)

    outputs_test = regr.predict(inputs2)

    m = scatter_plot(targets, outputs_test, variable_name)
    r = np.round(np.corrcoef(targets, outputs_test)[0][1],3)
    rms = mse(targets, outputs_test)

    return (r, rms, m)
```

0.5 Regressor 3

[ ]:
```
def regressor3 (inputs, targets):

    inputs = inputs.transpose()
```

```
    # Regressor
    scale = preprocessing.StandardScaler()
    inputs2 = scale.fit_transform(inputs)

    outputs_test = regr.predict(inputs2)

    # compute slope m and intercept b
    m, b = np.polyfit(targets, outputs_test, deg=1)

    r = np.round(np.corrcoef(targets, outputs_test)[0][1],3)
    rms = mse(targets, outputs_test)

    return (r, rms, m)
```

0.6 Regressor 4

[ ]:
```
def regressor4 (inputs, targets, variable_name):

    inputs = inputs.transpose()

    # Regressor
    scale = preprocessing.StandardScaler()
    inputs2 = scale.fit_transform(inputs)

    outputs = regr.predict(inputs2)

    # Post processing
    indx2 = np.full((len(diat_i.y)*len(diat_i.x)),np.nan)
    indx2[indx[0]] = outputs
    model = np.reshape(indx2,(len(diat_i.y),len(diat_i.x)))

    m = scatter_plot(targets, outputs, variable_name + str(dates[i].date()))

    # Preparation of the dataarray
    model = xr.DataArray(model,
        coords = {'y': diat_i.y, 'x': diat_i.x},
        dims = ['y','x'],
        attrs=dict( long_name = variable_name + "Concentration",
        units="mmol m-2"),)

    plotting3(targets, model, diat_i, variable_name)
```

0.7 Printing

[ ]:
```
def printing (targets, outputs, m):

    print ('The amount of data points is', outputs.size)
    print ('The slope of the best fitting line is ', np.round(m,3))
    print ('The correlation coefficient is:', np.round(np.corrcoef(targets,␣
 ↪outputs)[0][1],3))
    print (' The mean square error is:', np.round(mse(targets,outputs),5))
```

0.8 Scatter Plot

[ ]:
```
def scatter_plot(targets, outputs, variable_name):

    # compute slope m and intercept b
    m, b = np.polyfit(targets, outputs, deg=1)

    printing(targets, outputs, m)

    fig, ax = plt.subplots(2, figsize=(5,10), layout='constrained')

    ax[0].scatter(targets,outputs, alpha = 0.2, s = 10)

    lims = [np.min([ax[0].get_xlim(), ax[0].get_ylim()]),
        np.max([ax[0].get_xlim(), ax[0].get_ylim()])]

    # plot fitted y = m*x + b
    ax[0].axline(xy1=(0, b), slope=m, color='r')

    ax[0].set_xlabel('targets')
    ax[0].set_ylabel('outputs')
    ax[0].set_xlim(lims)
    ax[0].set_ylim(lims)
    ax[0].set_aspect('equal')

    ax[0].plot(lims, lims,linestyle = '--',color = 'k')

    h = ax[1].hist2d(targets,outputs, bins=100, cmap='jet',
        range=[lims,lims], cmin=0.1, norm='log')

    ax[1].plot(lims, lims,linestyle = '--',color = 'k')

    # plot fitted y = m*x + b
    ax[1].axline(xy1=(0, b), slope=m, color='r')

    ax[1].set_xlabel('targets')
    ax[1].set_ylabel('outputs')
```

```
    ax[1].set_aspect('equal')

    fig.colorbar(h[3],ax=ax[1], location='bottom')

    fig.suptitle(variable_name)

    plt.show()

    return (m)
```

0.9 Plotting

[ ]:
```
def plotting(variable, name):

    plt.plot(years,variable, marker = '.', linestyle = '')
    plt.legend(['diatom','flagellate'])
    plt.xlabel('Years')
    plt.ylabel(name)
    plt.show()
```

0.10 Plotting 2

[ ]:
```
def plotting2(variable,title):

    fig, ax = plt.subplots()

    scatter= ax.scatter(dates,variable, marker='.', c=pd.DatetimeIndex(dates).
 ↪month)

    ax.legend(handles=scatter.legend_elements()[0],␣
 ↪labels=['February','March','April'])
    fig.suptitle('Daily ' + title + ' (15 Feb - 30 Apr)')

    fig.show()
```

0.11 Plotting 3

[ ]:
```
def plotting3(targets, model, variable, variable_name):

    fig, ax = plt.subplots(2,2, figsize = (10,15))

    cmap = plt.get_cmap('cubehelix')
    cmap.set_bad('gray')

    variable.plot(ax=ax[0,0], cmap=cmap, vmin = targets.min(), vmax =targets.
 ↪max(), cbar_kwargs={'label': variable_name + ' Concentration  [mmol m-2]'})
```

```
    model.plot(ax=ax[0,1], cmap=cmap, vmin = targets.min(), vmax = targets.
↪max(), cbar_kwargs={'label': variable_name + ' Concentration  [mmol m-2]'})
    ((variable-model) / variable * 100).plot(ax=ax[1,0], cmap=cmap,␣
↪cbar_kwargs={'label': variable_name + ' Concentration  [percentage]'})

    plt.subplots_adjust(left=0.1,
        bottom=0.1,
        right=0.95,
        top=0.95,
        wspace=0.35,
        hspace=0.35)

    sa_vi.set_aspect(ax[0,0])
    sa_vi.set_aspect(ax[0,1])
    sa_vi.set_aspect(ax[1,0])


    ax[0,0].title.set_text(variable_name + ' (targets)')
    ax[0,1].title.set_text(variable_name + ' (outputs)')
    ax[1,0].title.set_text('targets - outputs')
    ax[1,1].axis('off')

    fig.suptitle(str(dates[i].date()))

    plt.show()
```

0.12 Training (Random Points)

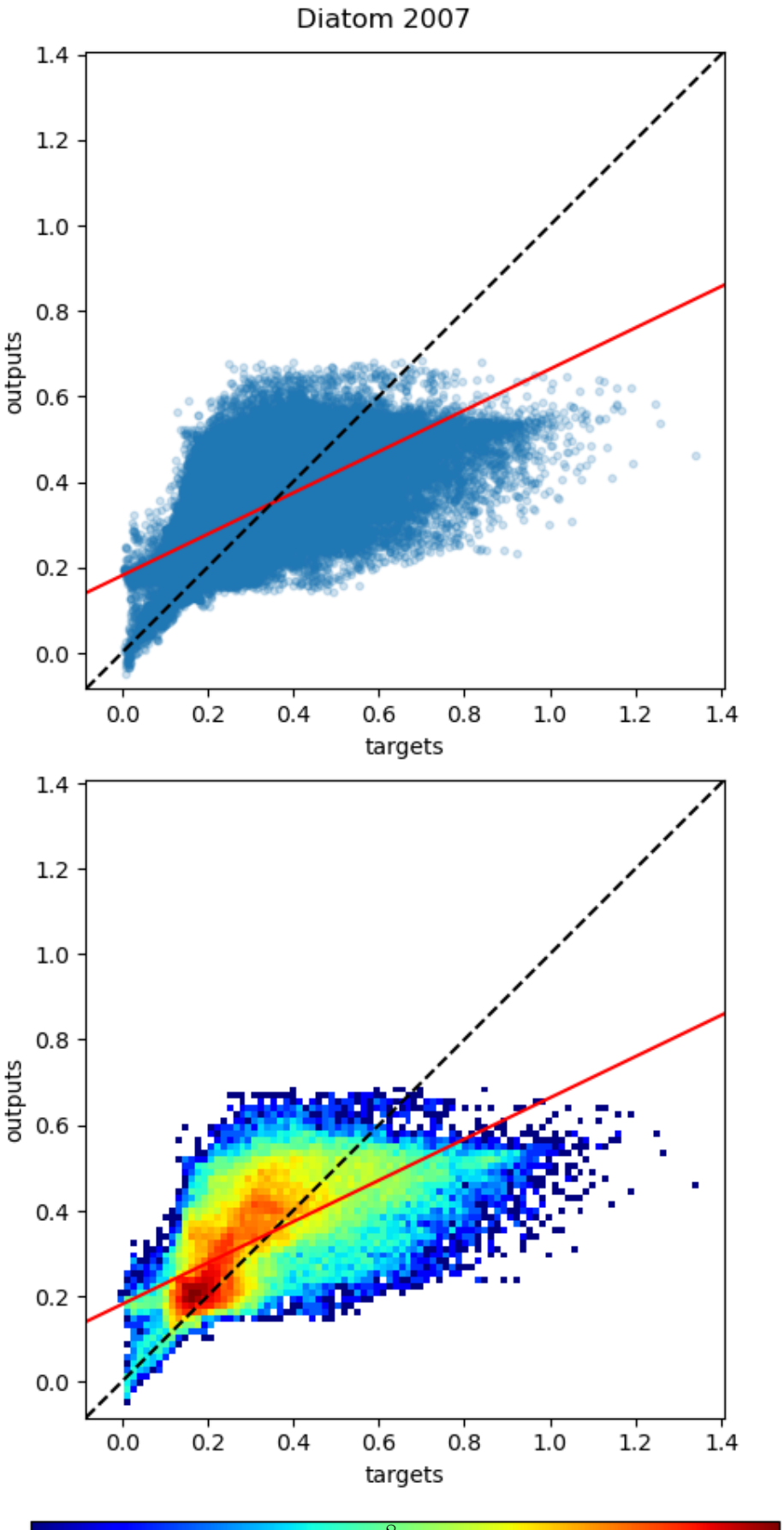
[ ]:
```
ds = xr.open_dataset('/data/ibougoudis/MOAD/files/integrated_model_var_old.nc')

ds = ds.isel(time_counter = (np.arange(0, len(ds.Diatom.time_counter),2)),
    y=(np.arange(ds.y[0], ds.y[-1], 5)),
    x=(np.arange(ds.x[0], ds.x[-1], 5)))

dates = pd.DatetimeIndex(ds['time_counter'].values)

drivers, diat, _ = datasets_preparation(ds)

regr = regressor(drivers, diat)
```

0.13 Other Years (Anually)

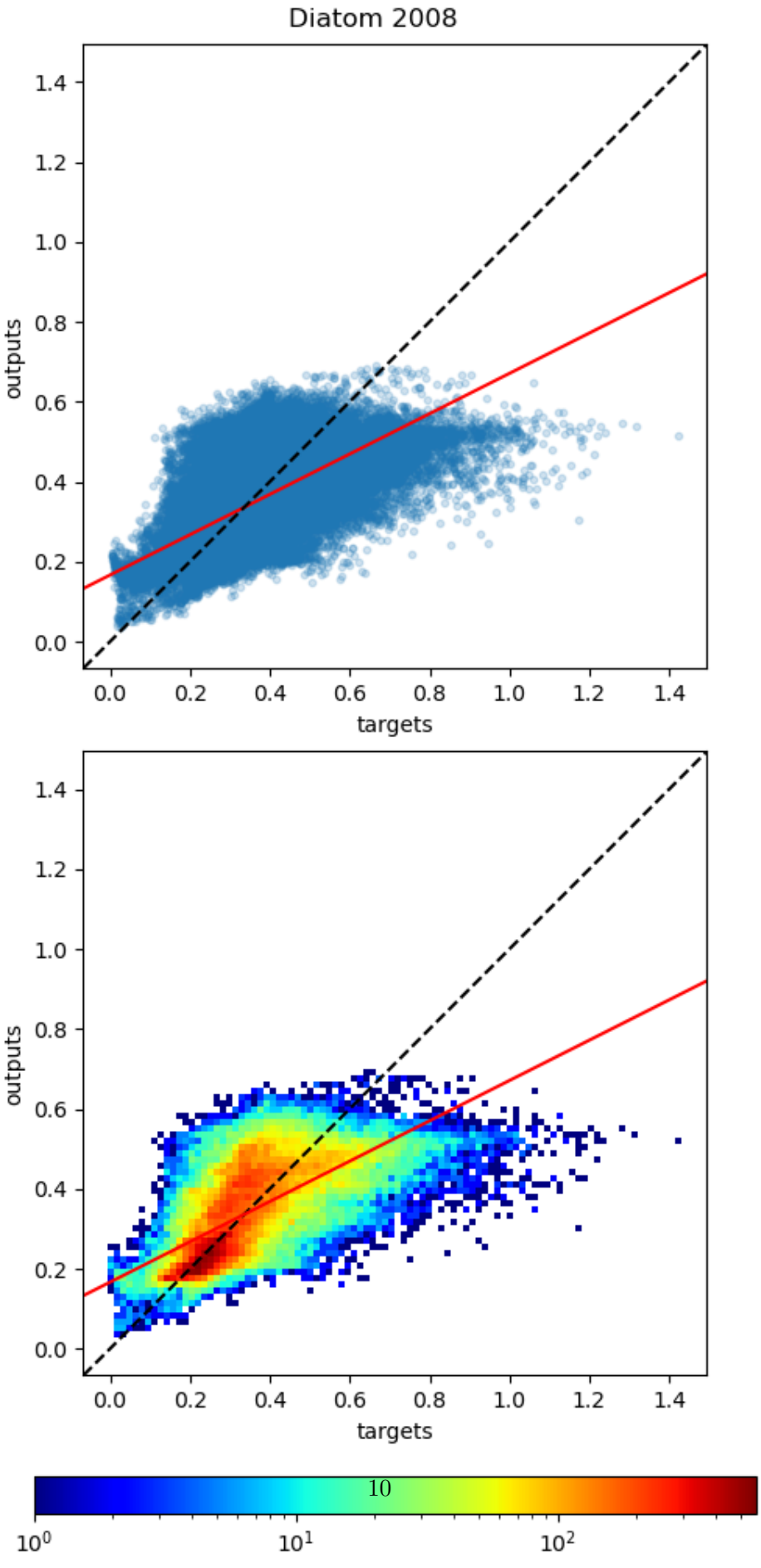
[ ]:
```
years = range (2007,2024)

r_all = []
rms_all = []
slope_all = []

for year in tqdm(range (2007,2024)):

    dataset = ds.sel(time_counter=str(year))

    drivers, diat, _ = datasets_preparation(dataset)

    r, rms, m = regressor2(drivers, diat, 'Diatom ' + str(year))

    r_all.append(r)
    rms_all.append(rms)
    slope_all.append(m)

plotting(np.transpose(r_all), 'Correlation Coefficient')
plotting(np.transpose(rms_all), 'Mean Square Error')
plotting (np.transpose(slope_all), 'Slope of the best fitting line')
```

```
  0%|          | 0/17 [00:00<?, ?it/s]
```

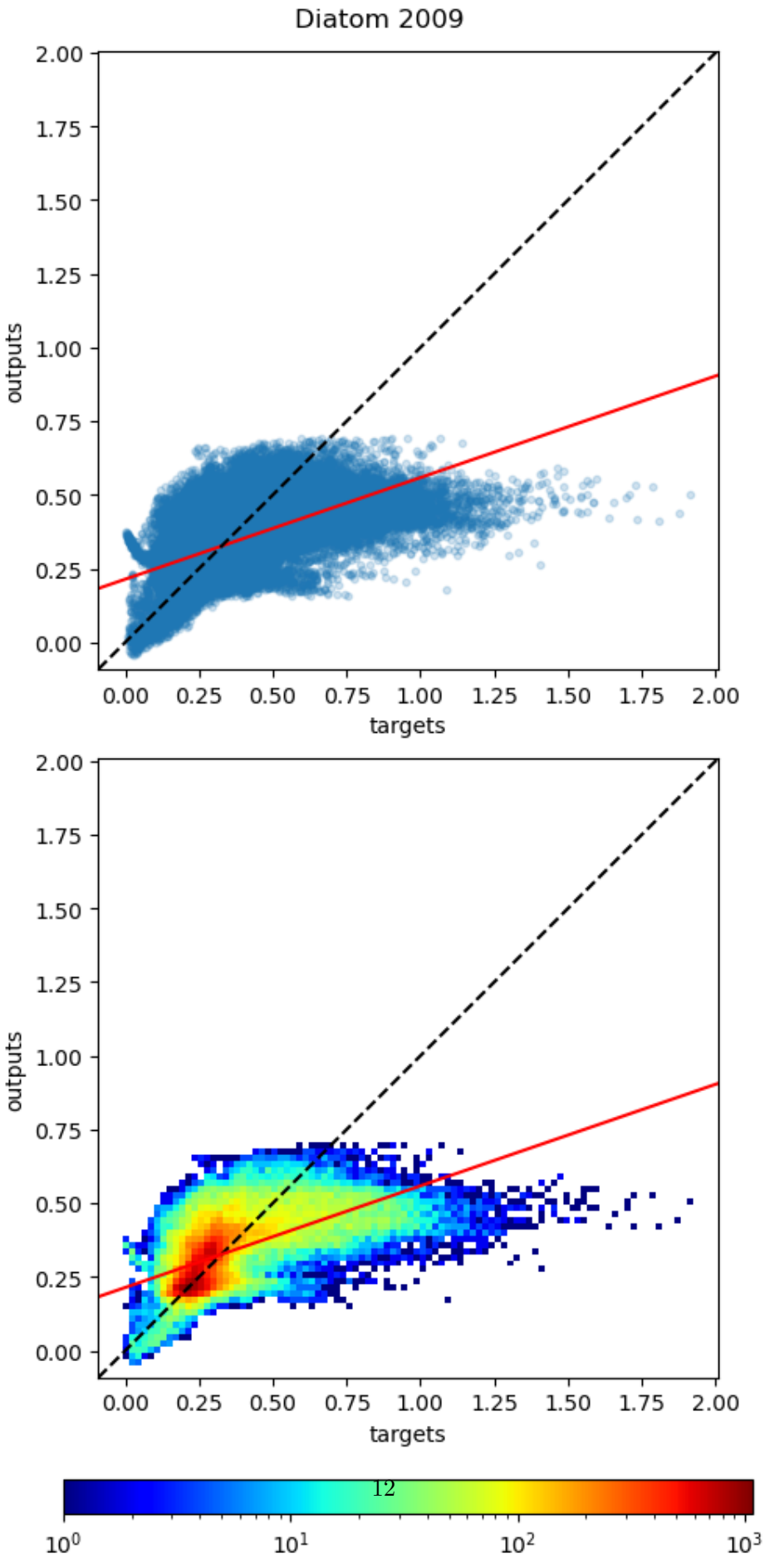
```
The amount of data points is 70794
The slope of the best fitting line is  0.483
The correlation coefficient is: 0.666
 The mean square error is: 0.01471
```


Diatom 2007
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
8
10^0
10^1
10^2

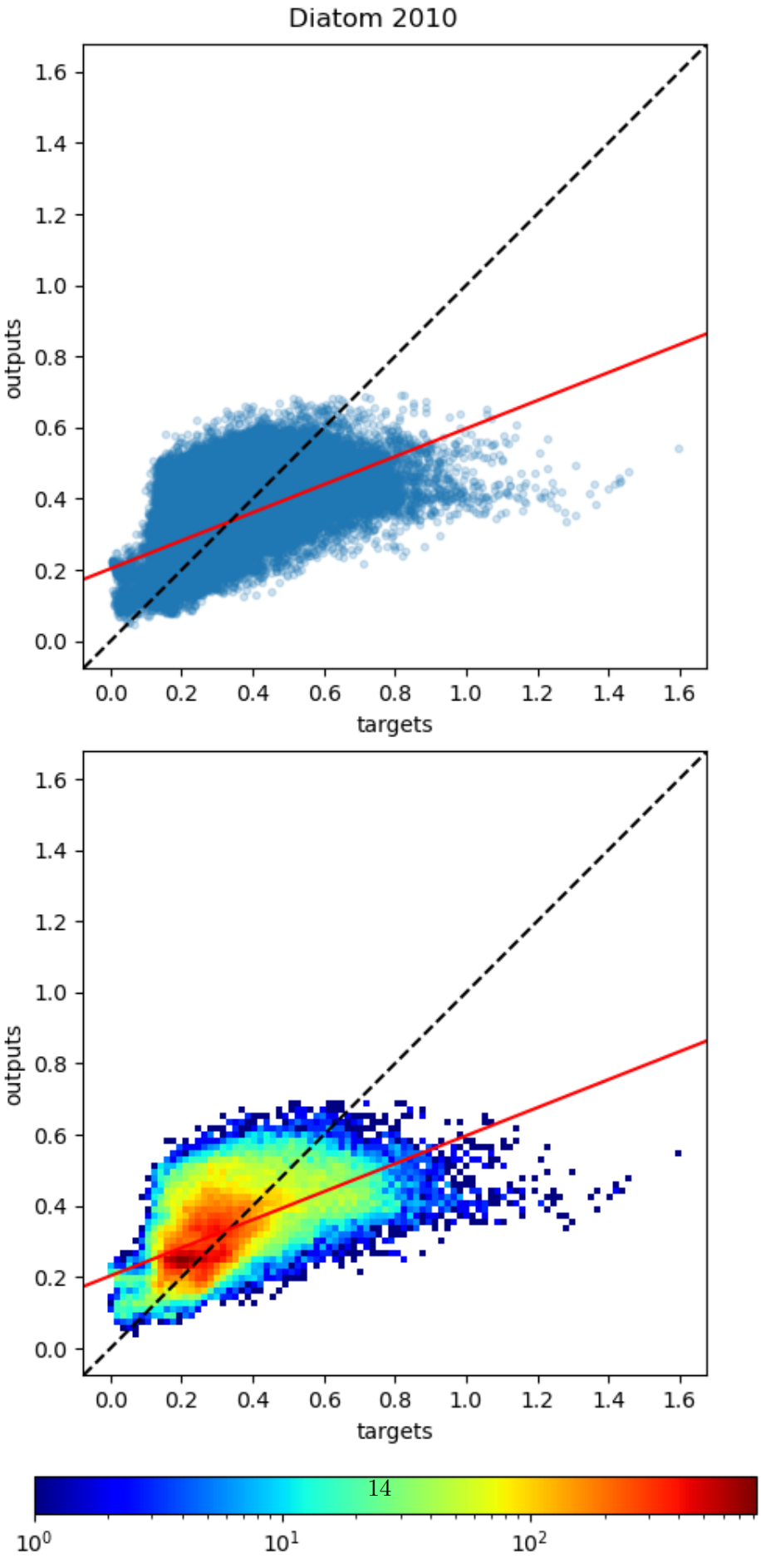
```
The amount of data points is 70794
The slope of the best fitting line is  0.504
The correlation coefficient is: 0.655
 The mean square error is: 0.01234
```


Diatom 2008
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
10
10^0
10^1
10^2

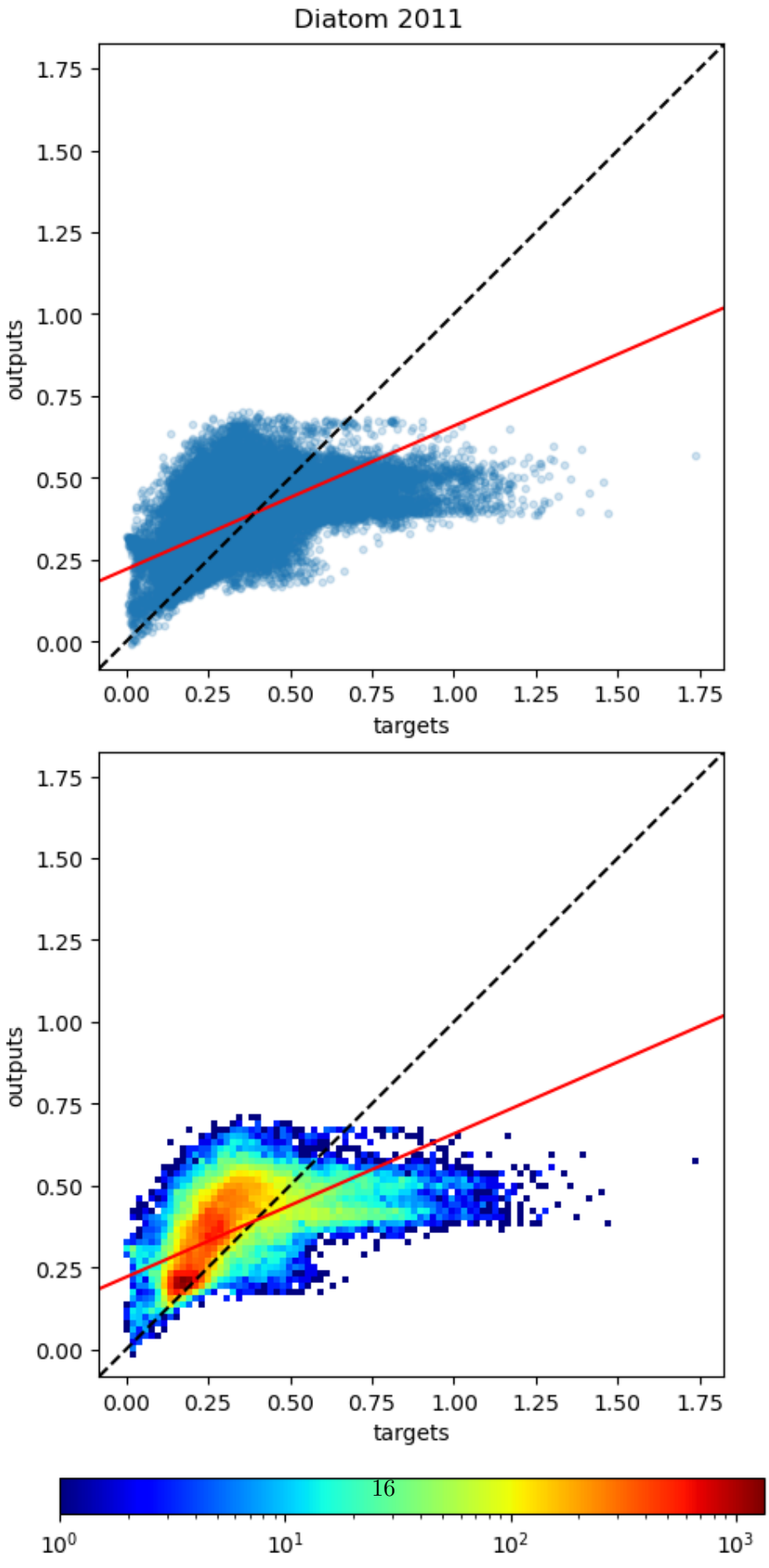
```
The amount of data points is 68931
The slope of the best fitting line is  0.345
The correlation coefficient is: 0.6
 The mean square error is: 0.02483
```


Diatom 2009
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
12
10^0
10^1
10^2
10^3

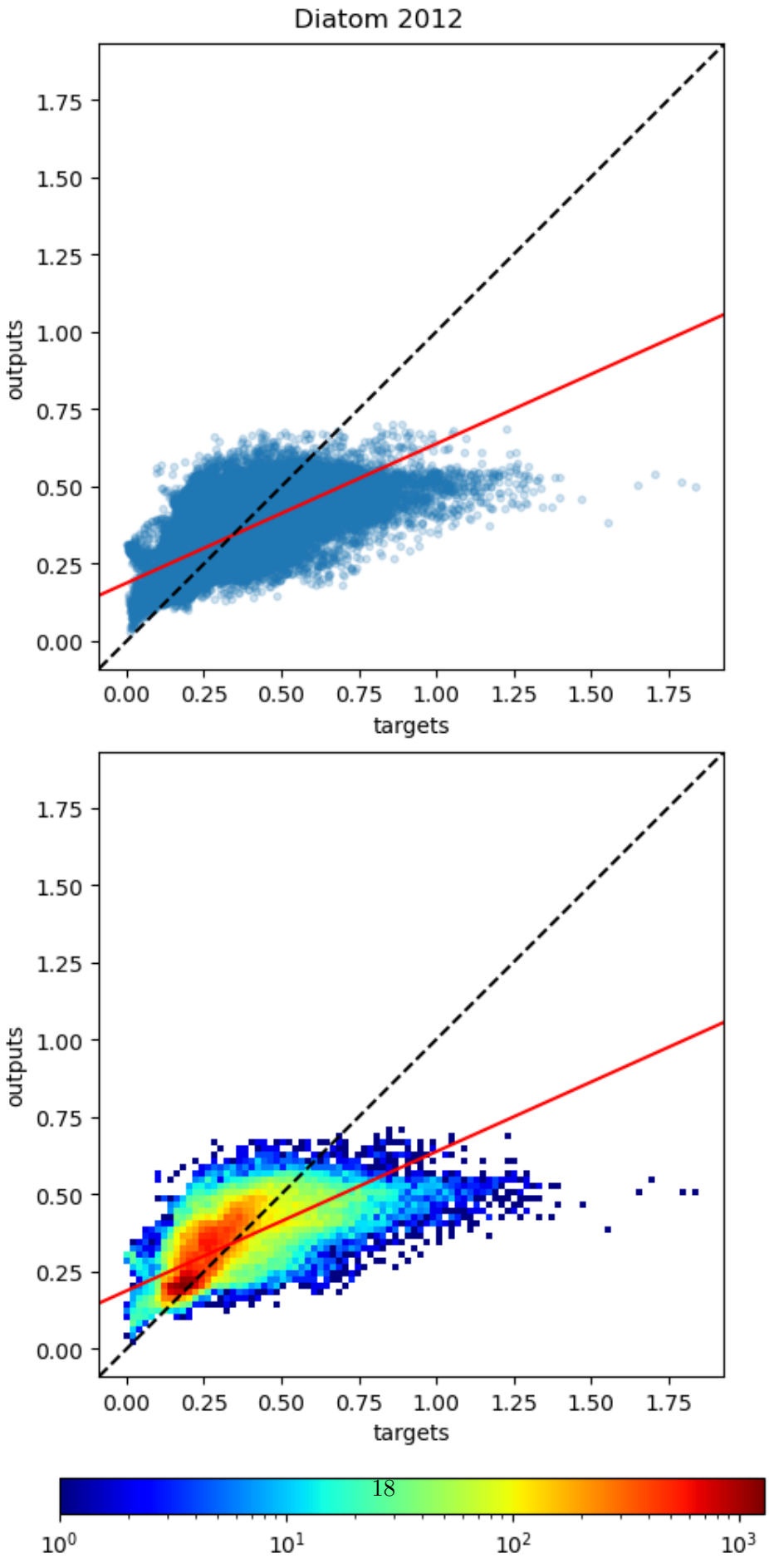
```
The amount of data points is 70794
The slope of the best fitting line is  0.394
The correlation coefficient is: 0.576
 The mean square error is: 0.01444
```


Diatom 2010
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
outputs
targets
14
10^0
10^1
10^2

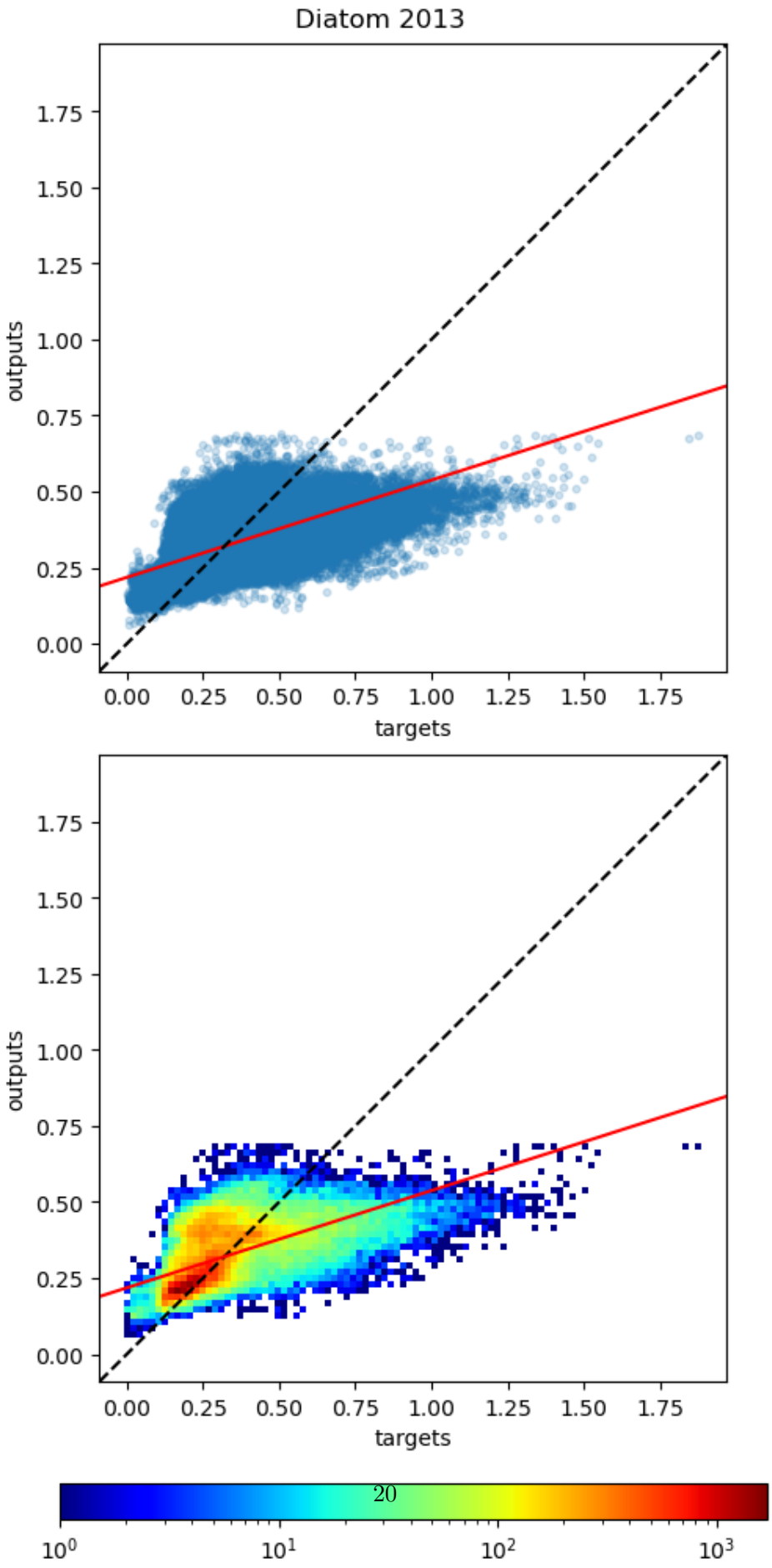
```
The amount of data points is 68931
The slope of the best fitting line is  0.438
The correlation coefficient is: 0.606
 The mean square error is: 0.0183
```


Diatom 2011
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
16
10^0
10^1
10^2
10^3

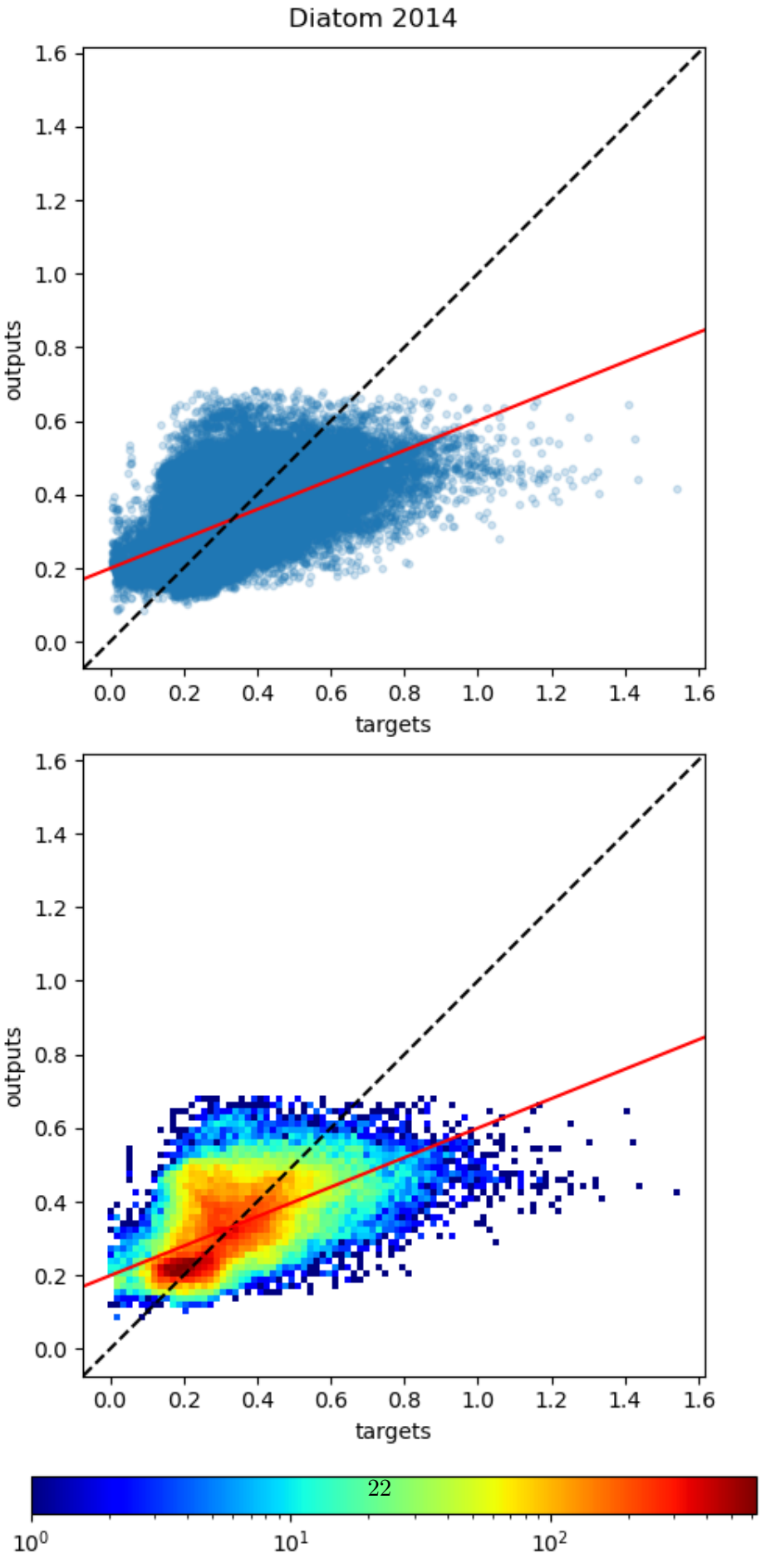
```
The amount of data points is 70794
The slope of the best fitting line is  0.45
The correlation coefficient is: 0.689
 The mean square error is: 0.01361
```


Diatom 2012
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
18
10^0
10^1
10^2
10^3

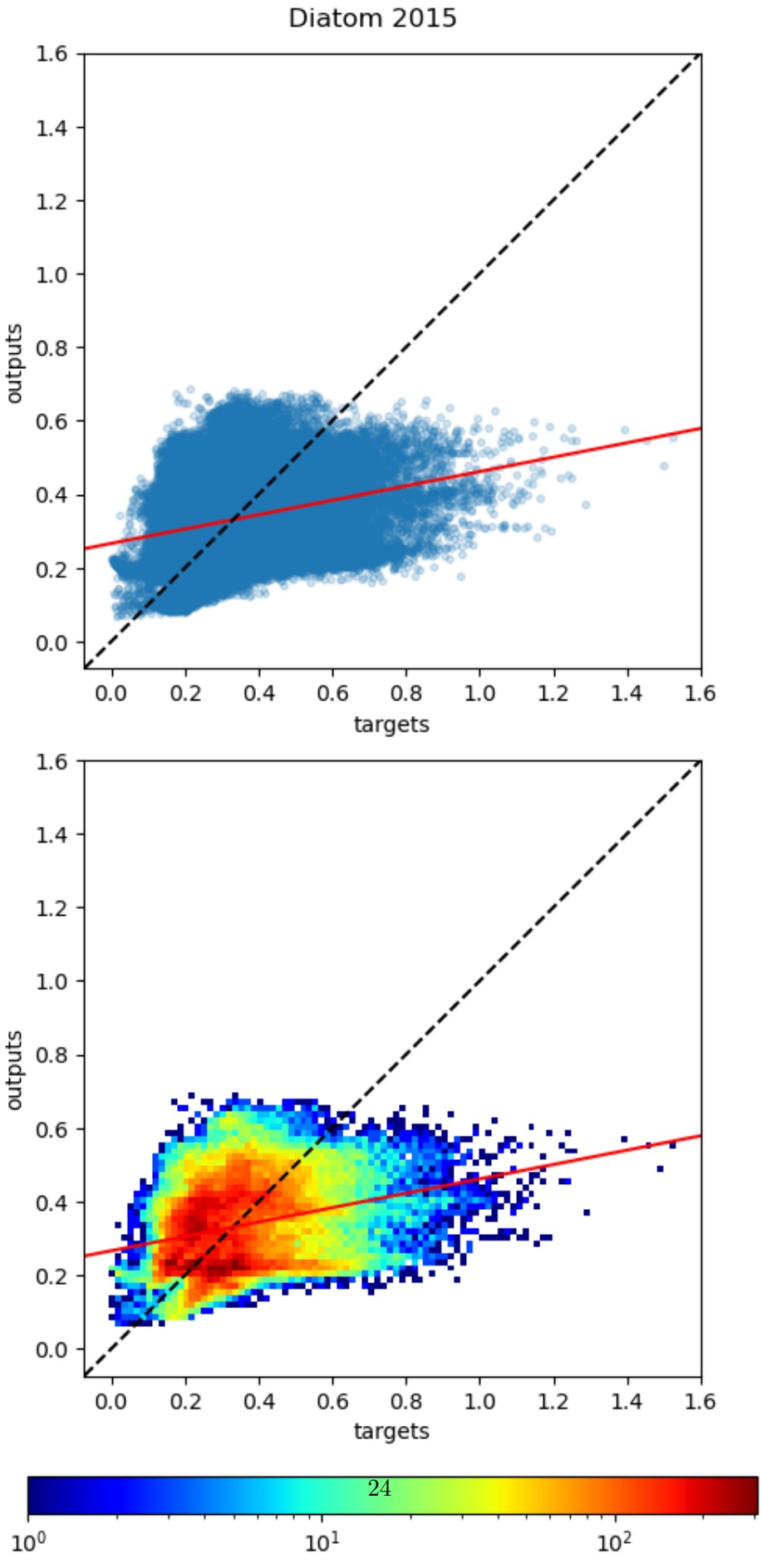
```
The amount of data points is 70794
The slope of the best fitting line is  0.32
The correlation coefficient is: 0.588
 The mean square error is: 0.02118
```


Diatom 2013
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
20
10^0
10^1
10^2
10^3

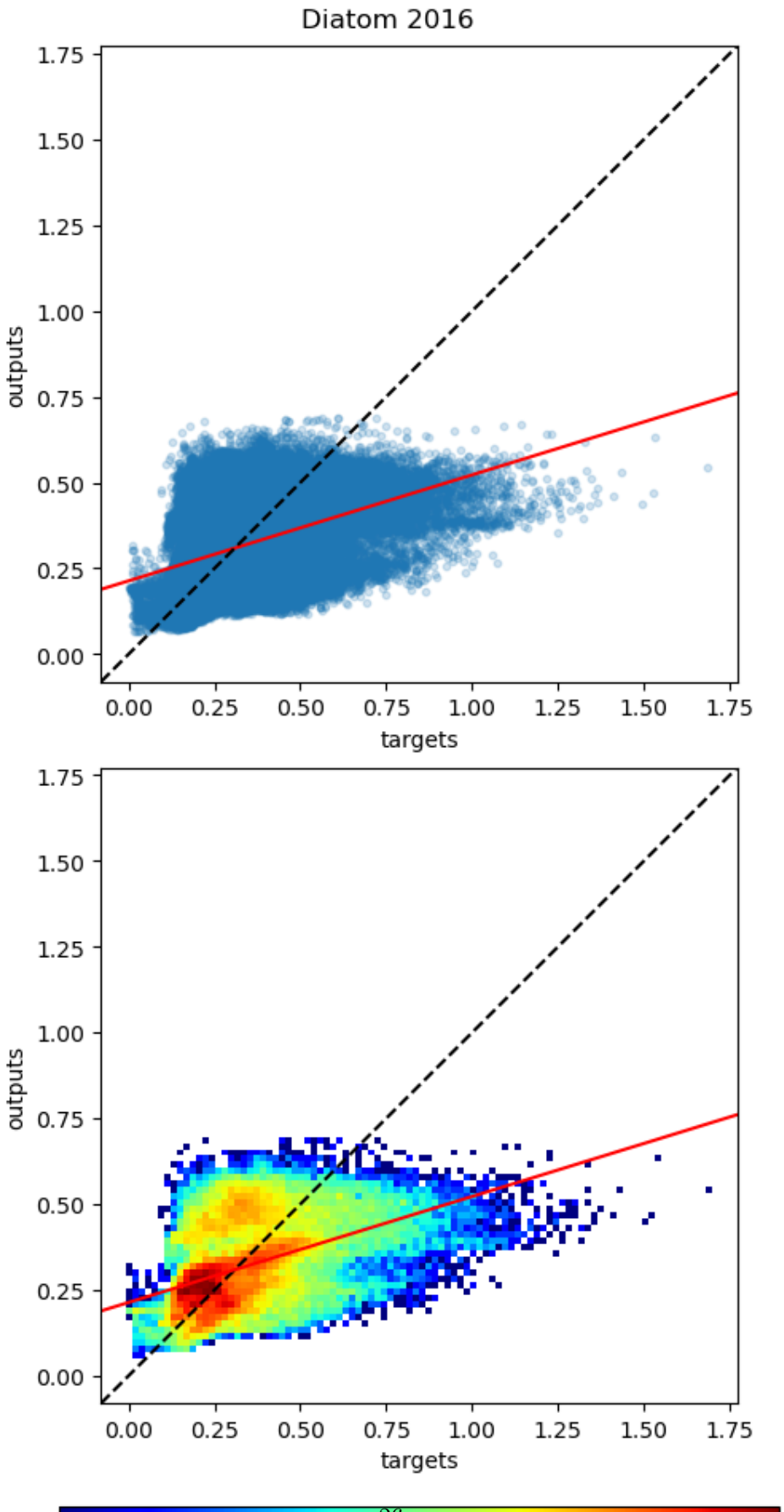
```
The amount of data points is 68931
The slope of the best fitting line is  0.401
The correlation coefficient is: 0.577
 The mean square error is: 0.01432
```


Diatom 2014
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
22
10^0
10^1
10^2

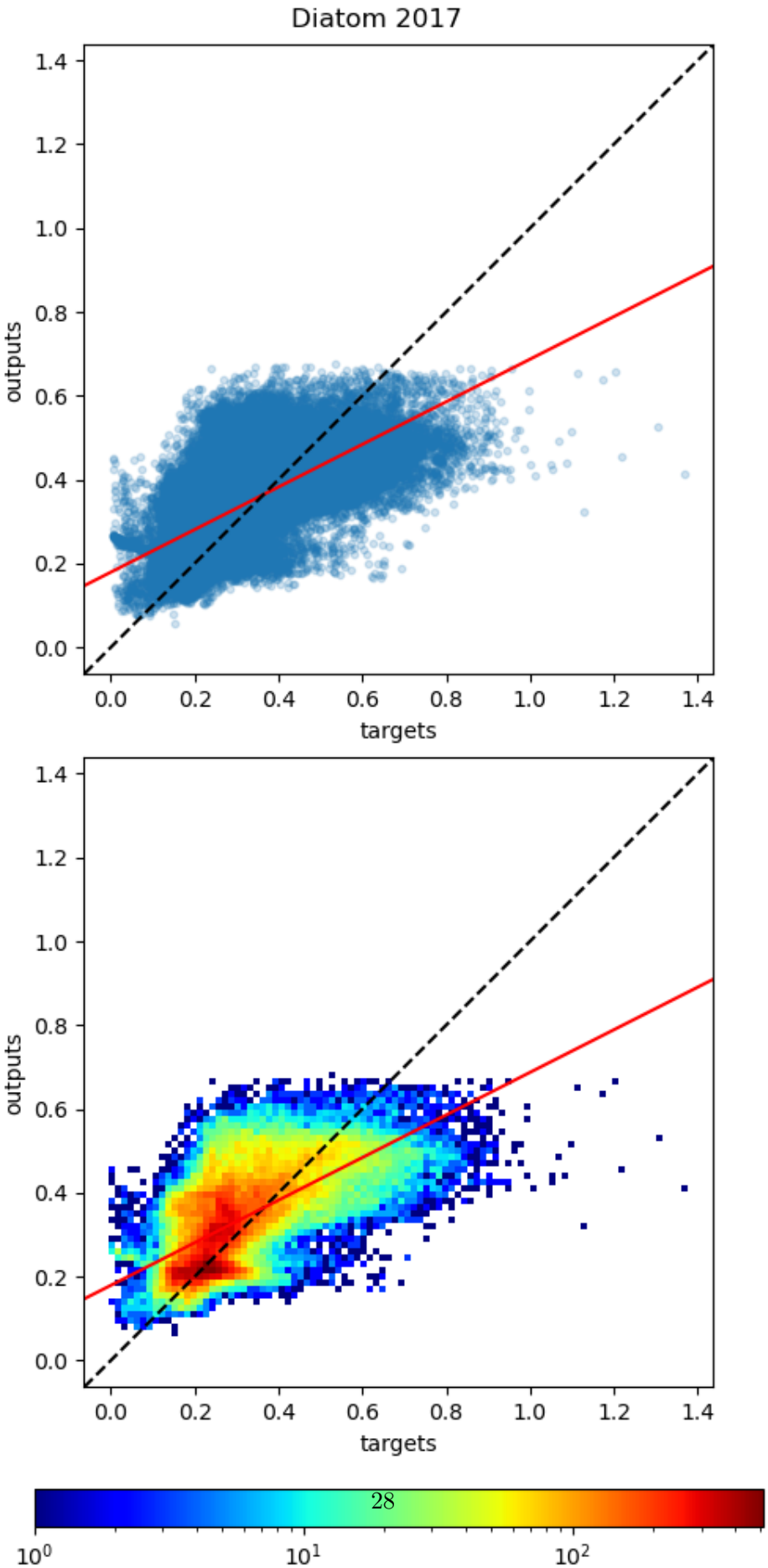
```
The amount of data points is 70794
The slope of the best fitting line is  0.195
The correlation coefficient is: 0.271
 The mean square error is: 0.02633
```


Diatom 2015
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
24
10^0
10^1
10^2

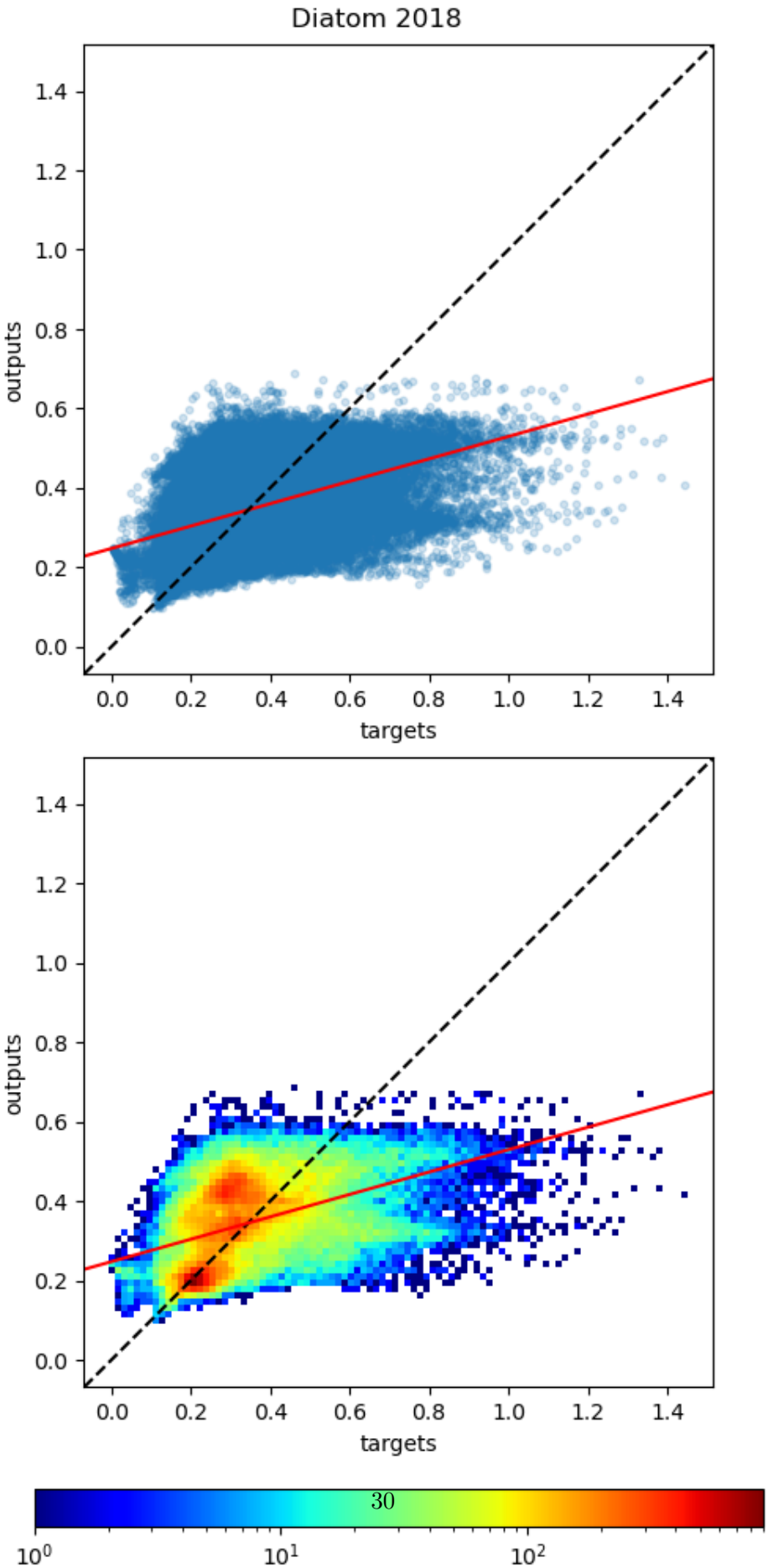
```
The amount of data points is 70794
The slope of the best fitting line is  0.309
The correlation coefficient is: 0.455
 The mean square error is: 0.02419
```


Diatom 2016
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
26
10^0
10^1
10^2

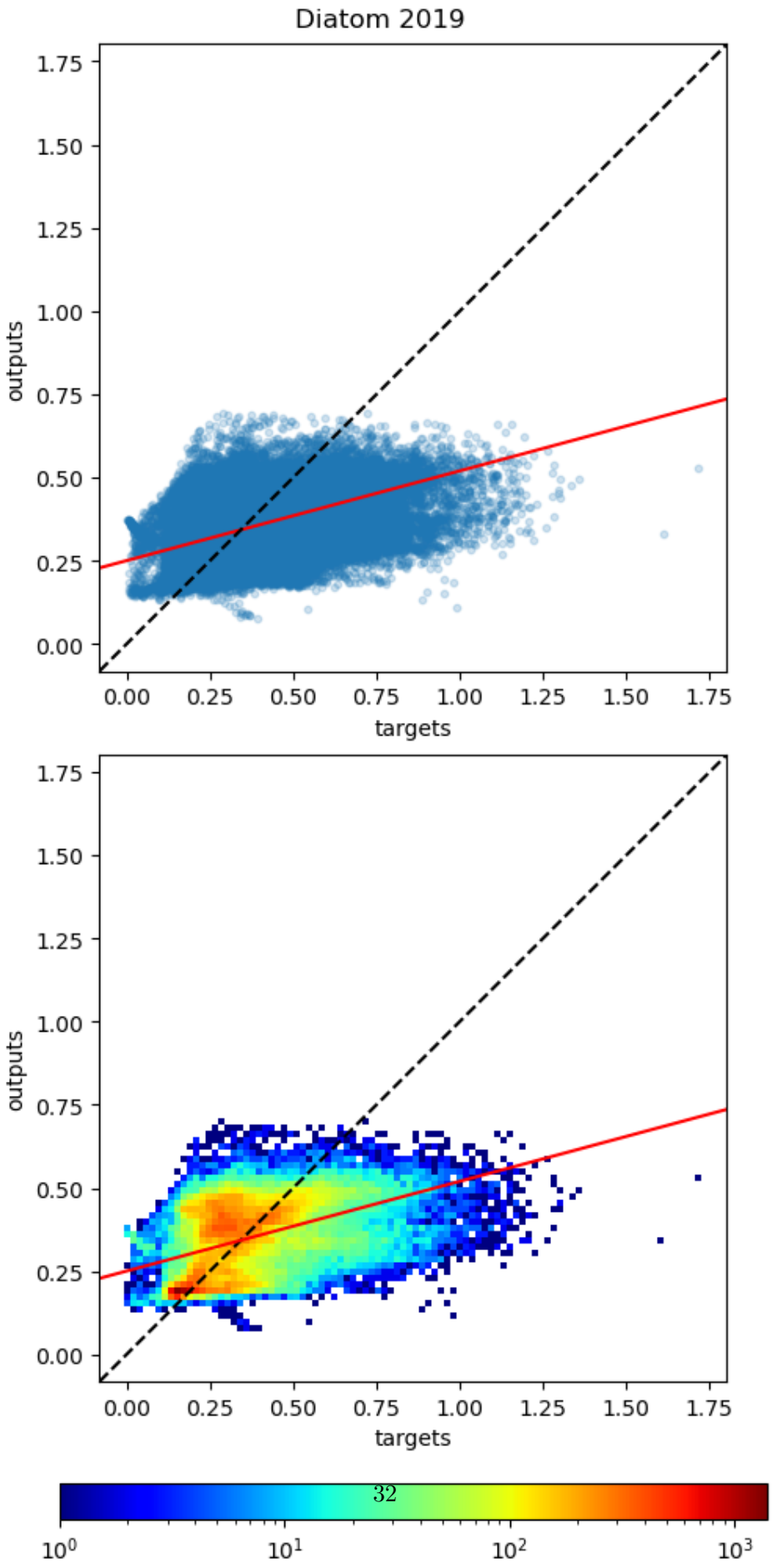
```
The amount of data points is 68931
The slope of the best fitting line is  0.508
The correlation coefficient is: 0.633
 The mean square error is: 0.01202
```


Diatom 2017
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
28
10^0
10^1
10^2

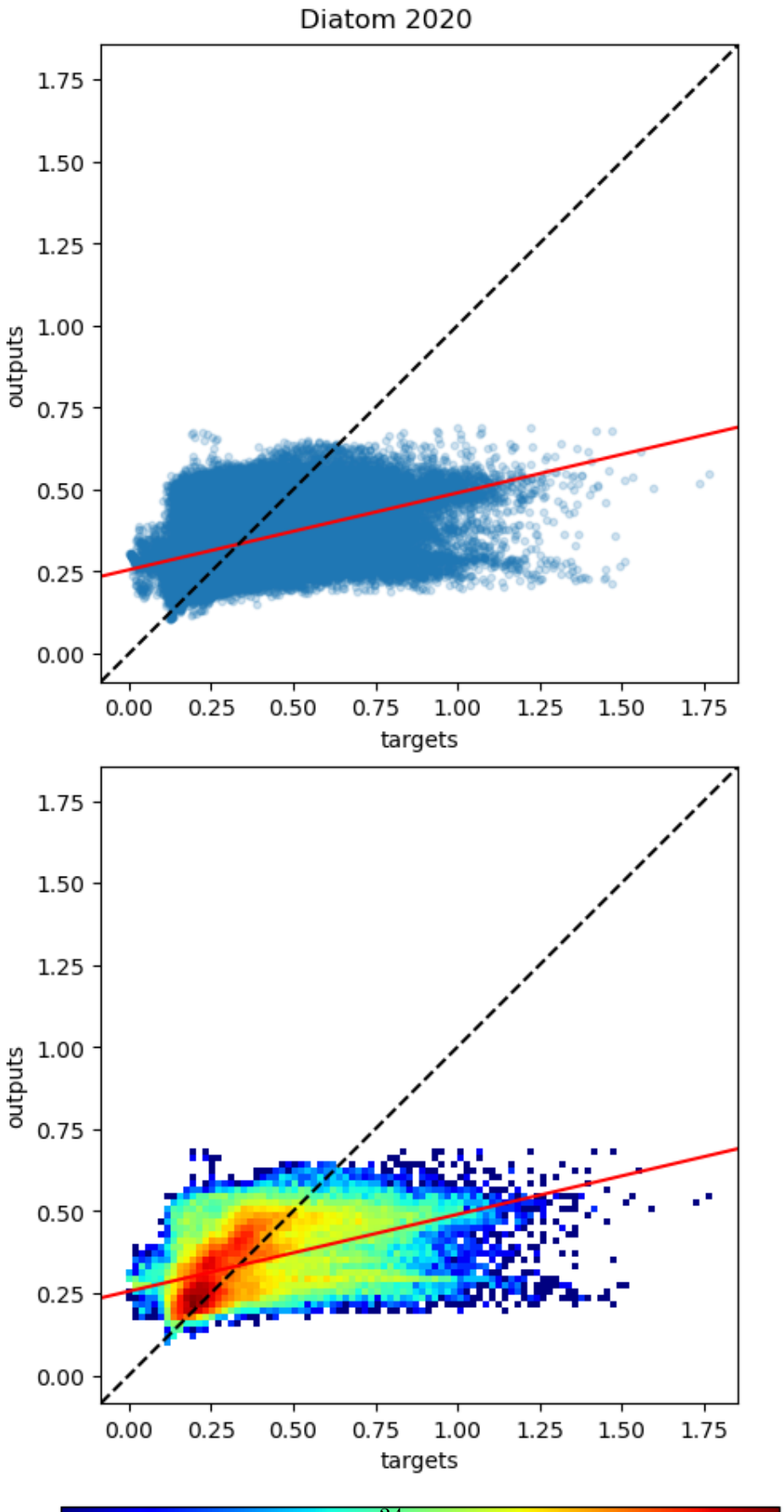
```
The amount of data points is 70794
The slope of the best fitting line is  0.282
The correlation coefficient is: 0.432
 The mean square error is: 0.02141
```


Diatom 2018
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
30
10^0
10^1
10^2

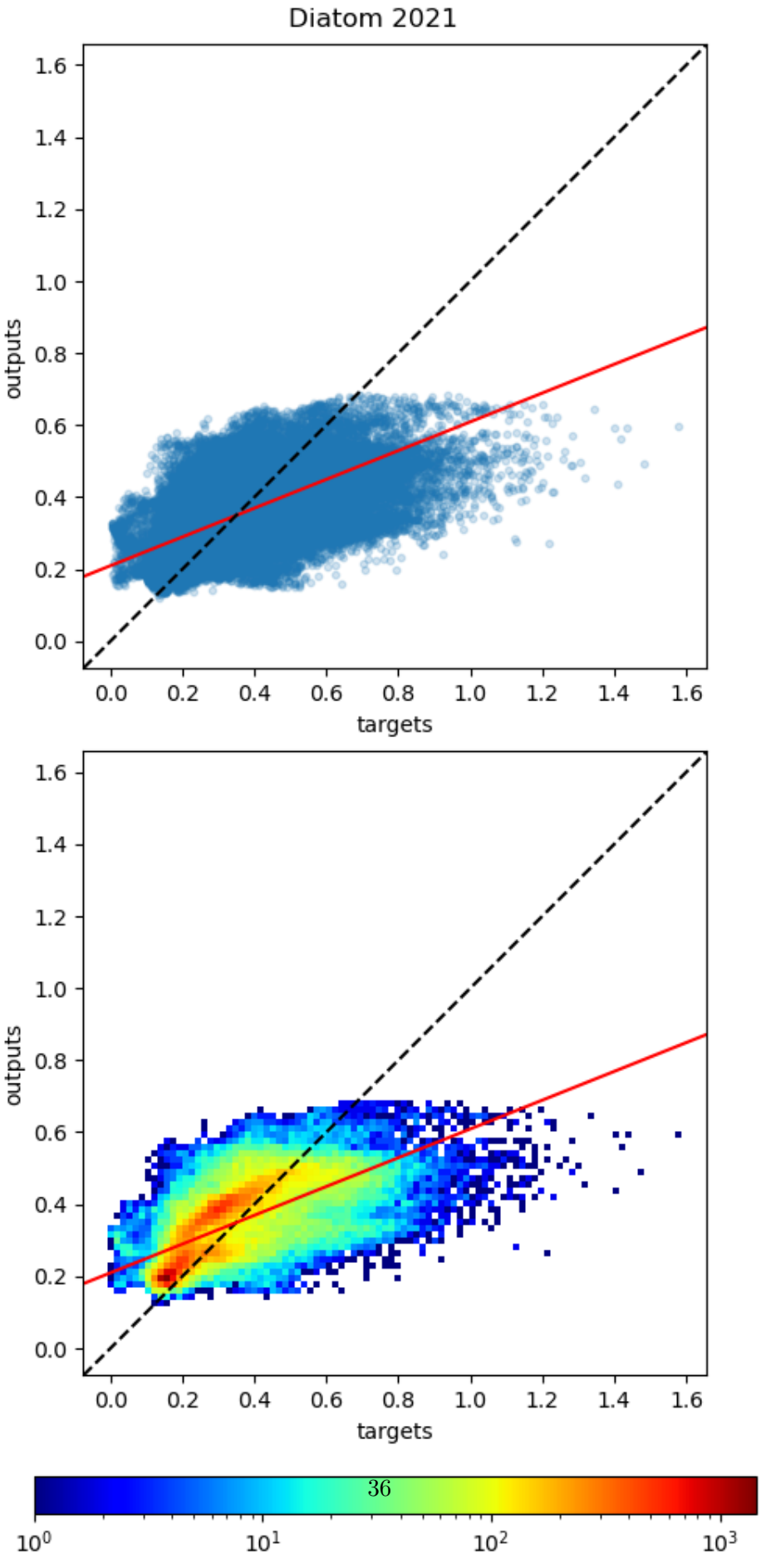
```
The amount of data points is 68931
The slope of the best fitting line is  0.27
The correlation coefficient is: 0.446
 The mean square error is: 0.02494
```


Diatom 2019
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
32
10^0
10^1
10^2
10^3

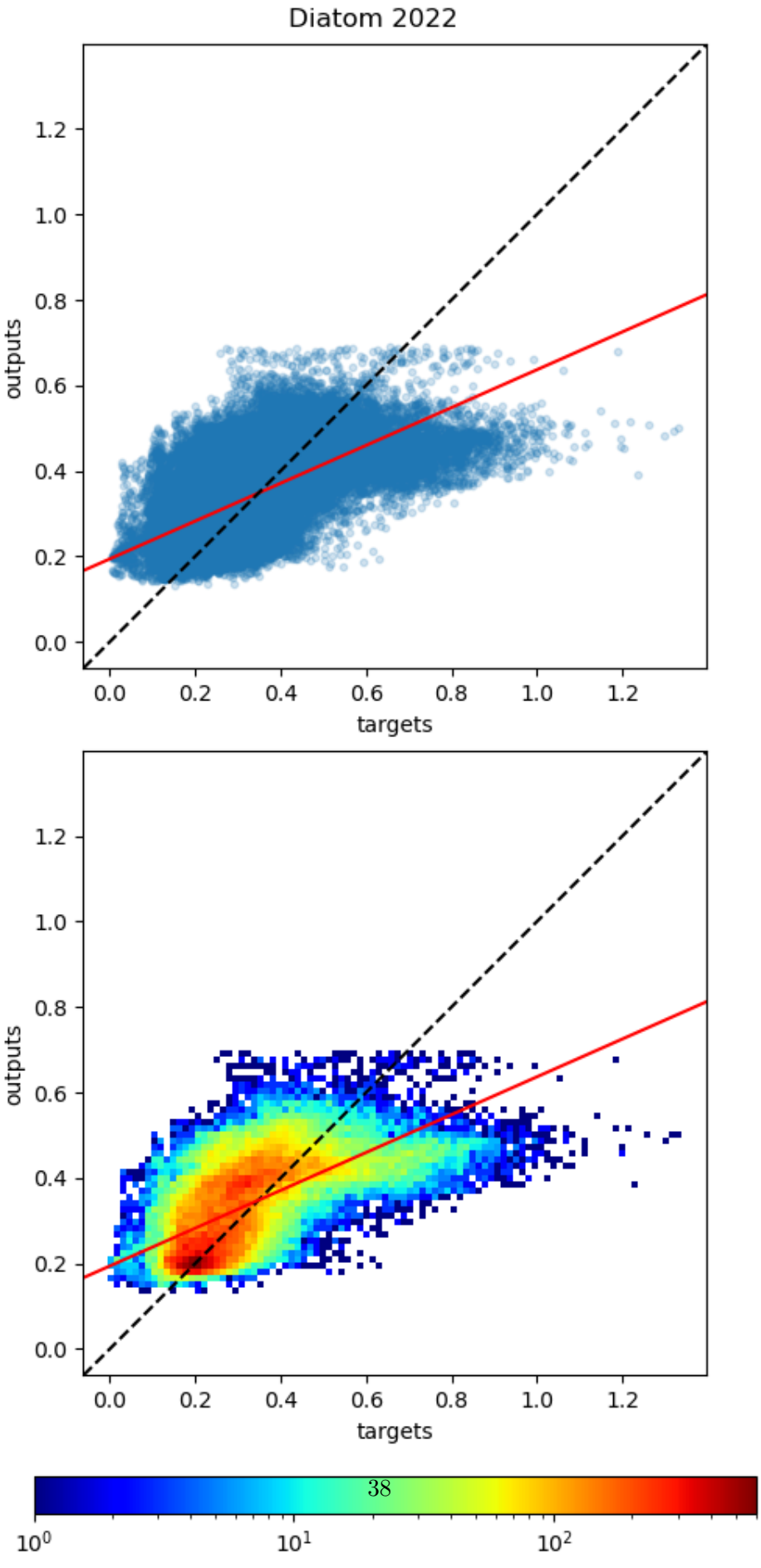
```
The amount of data points is 70794
The slope of the best fitting line is  0.234
The correlation coefficient is: 0.475
 The mean square error is: 0.03263
```


Diatom 2020
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
34
10^0
10^1
10^2

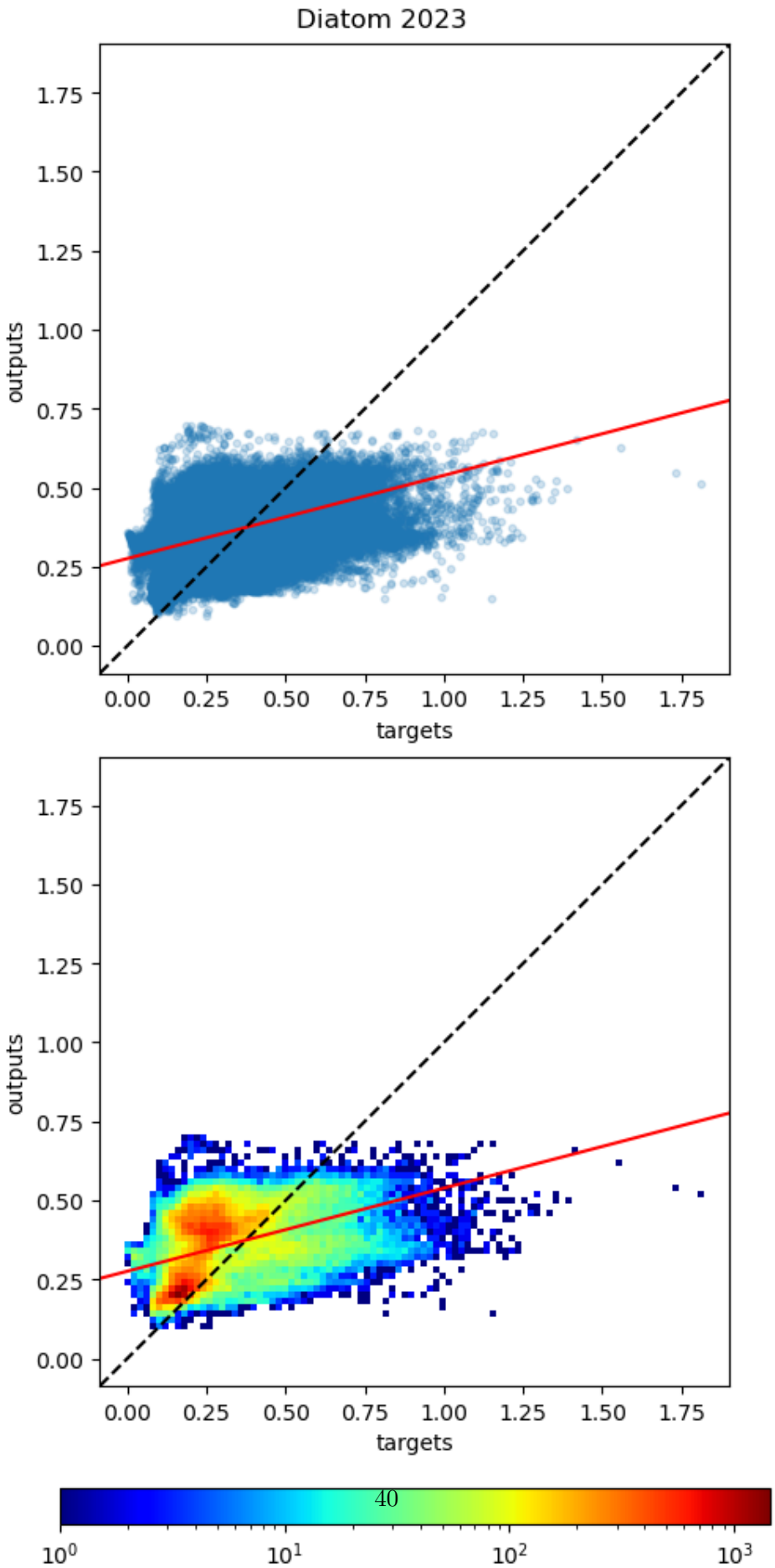
```
The amount of data points is 70794
The slope of the best fitting line is  0.4
The correlation coefficient is: 0.646
 The mean square error is: 0.01751
```

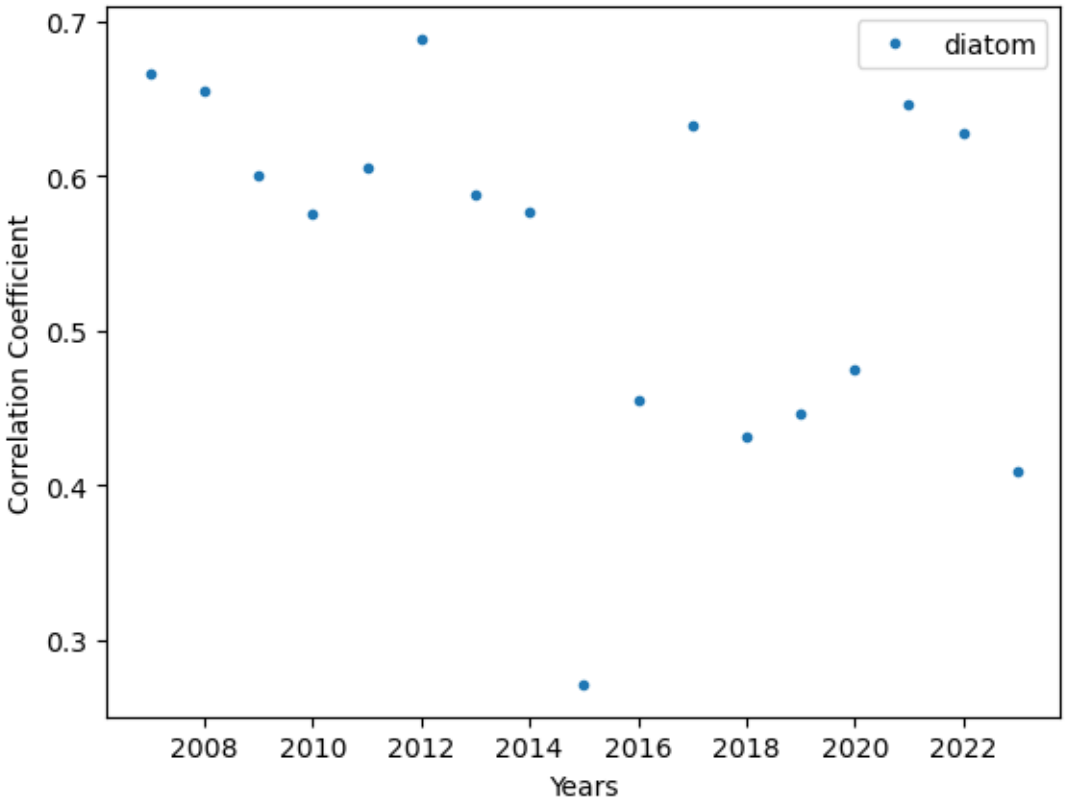

Diatom 2021
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
outputs
targets
36
10^0
10^1
10^2
10^3

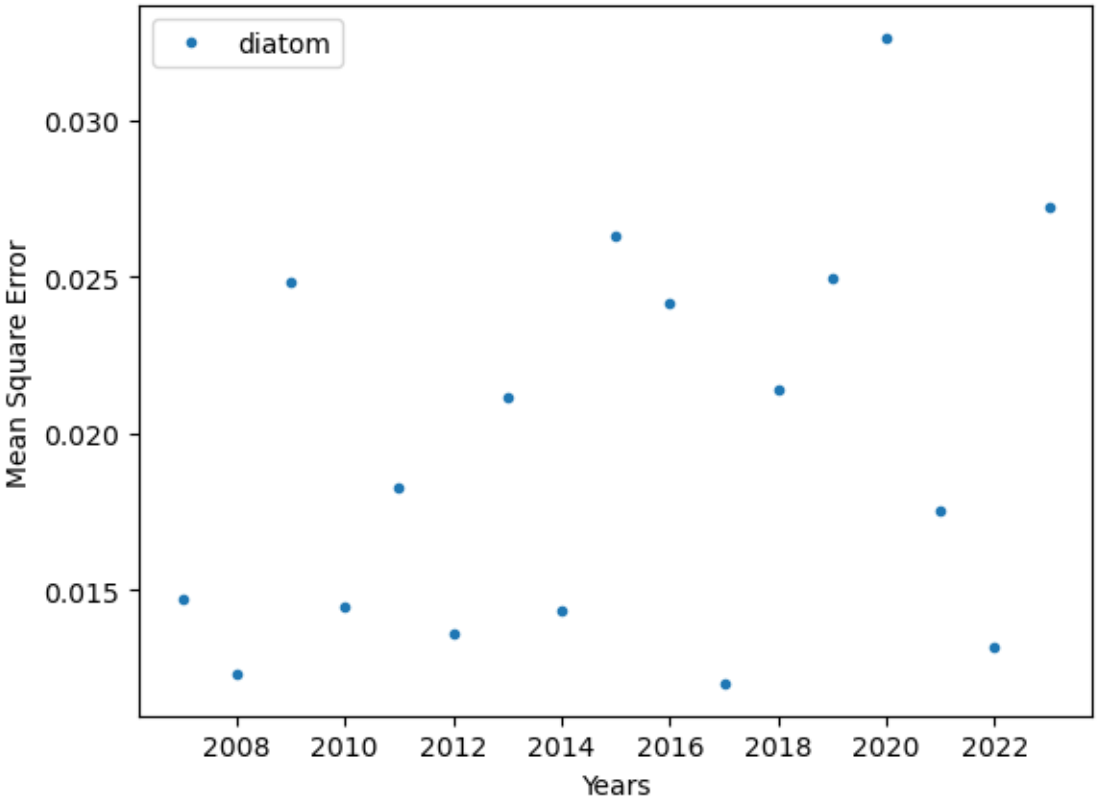
```
The amount of data points is 68931
The slope of the best fitting line is  0.443
The correlation coefficient is: 0.628
 The mean square error is: 0.01321
```

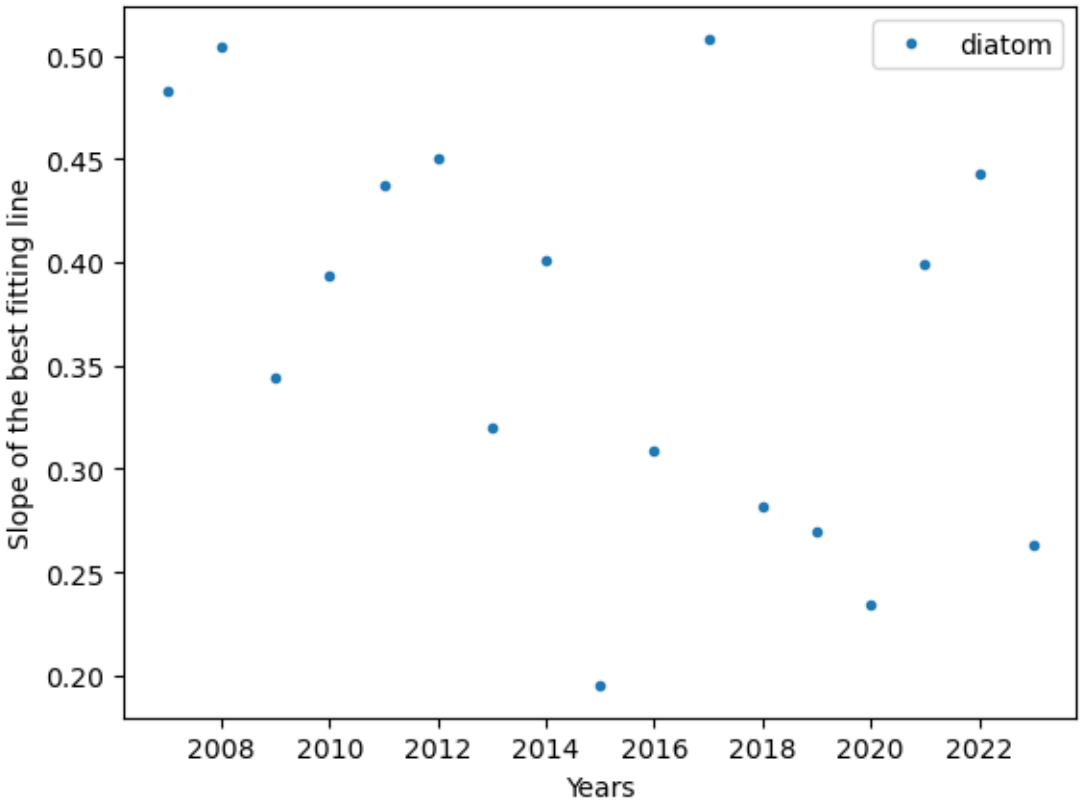

Diatom 2022
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
outputs
targets
38
10^0
10^1
10^2

```
The amount of data points is 70794
The slope of the best fitting line is  0.263
The correlation coefficient is: 0.409
 The mean square error is: 0.02721
```


Diatom 2023
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
40
10^0
10^1
10^2
10^3


diatom
0.7
0.6
0.5
0.4
0.3
Correlation Coefficient
2008
2010
2012
2014
2016
2018
2020
2022
Years

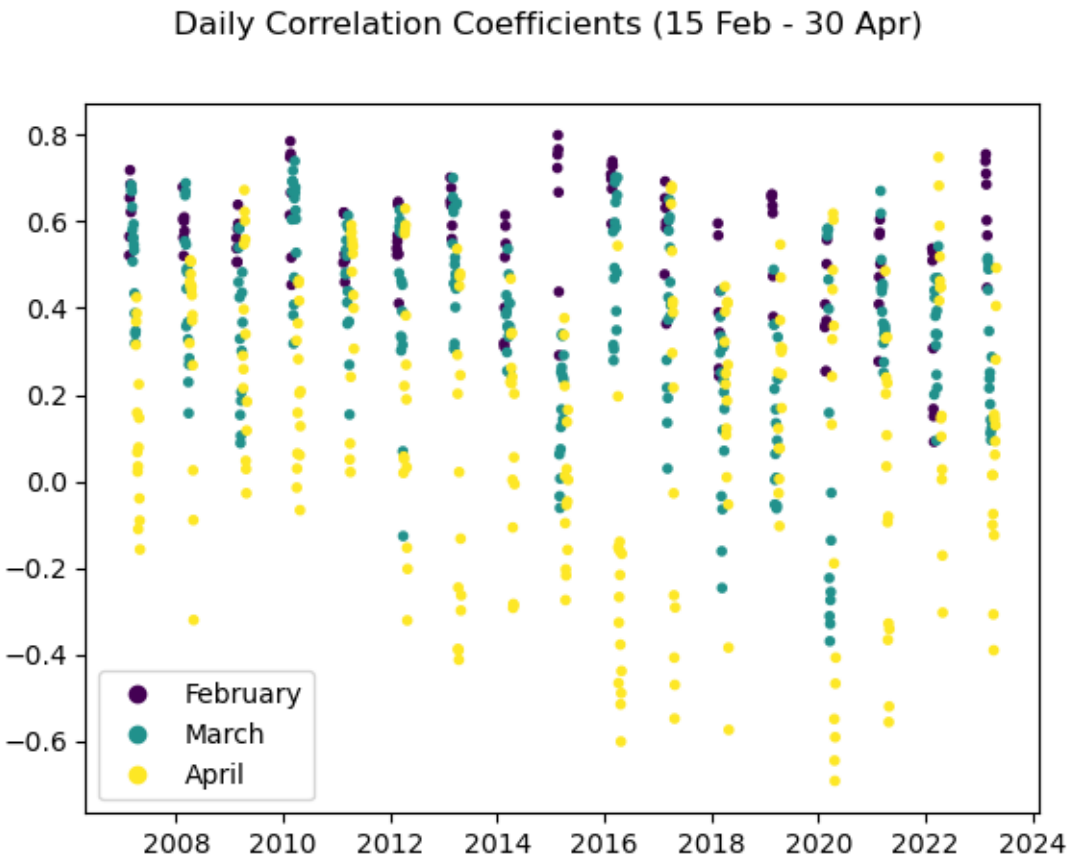

diatom
0.030
0.025
0.020
0.015
Mean Square Error
2008
2010
2012
2014
2016
2018
2020
2022
Years

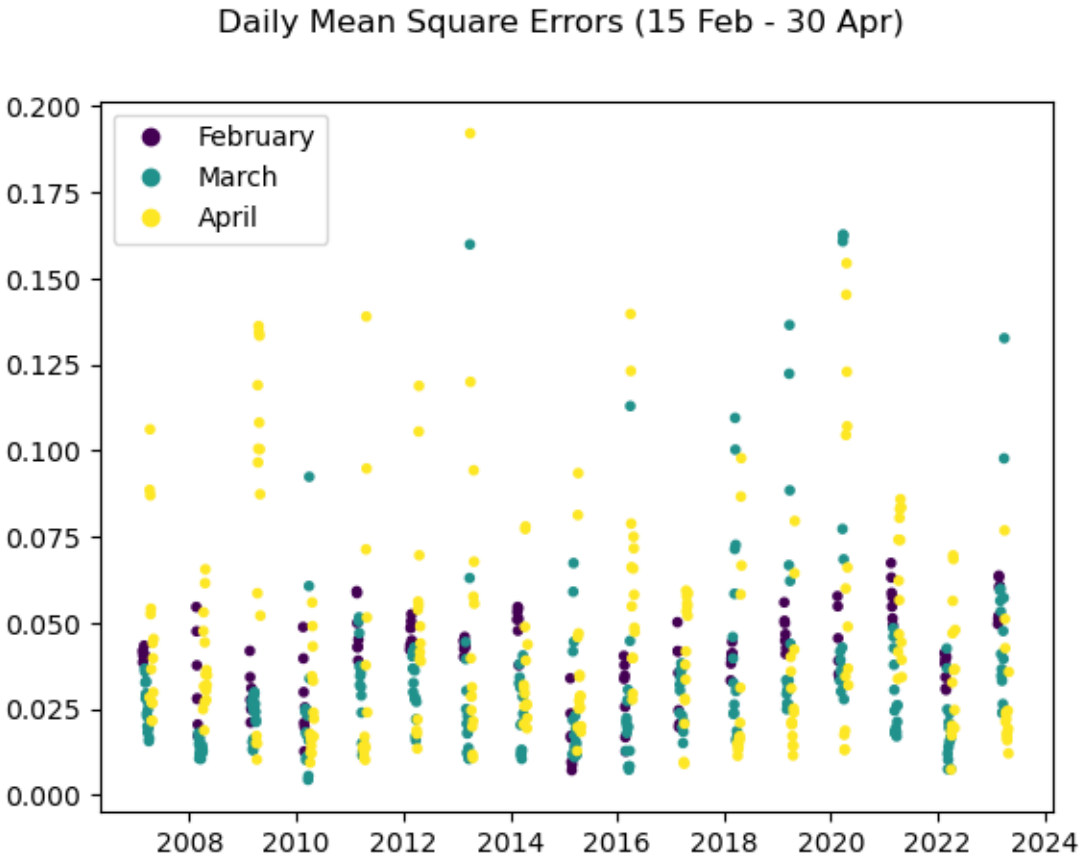


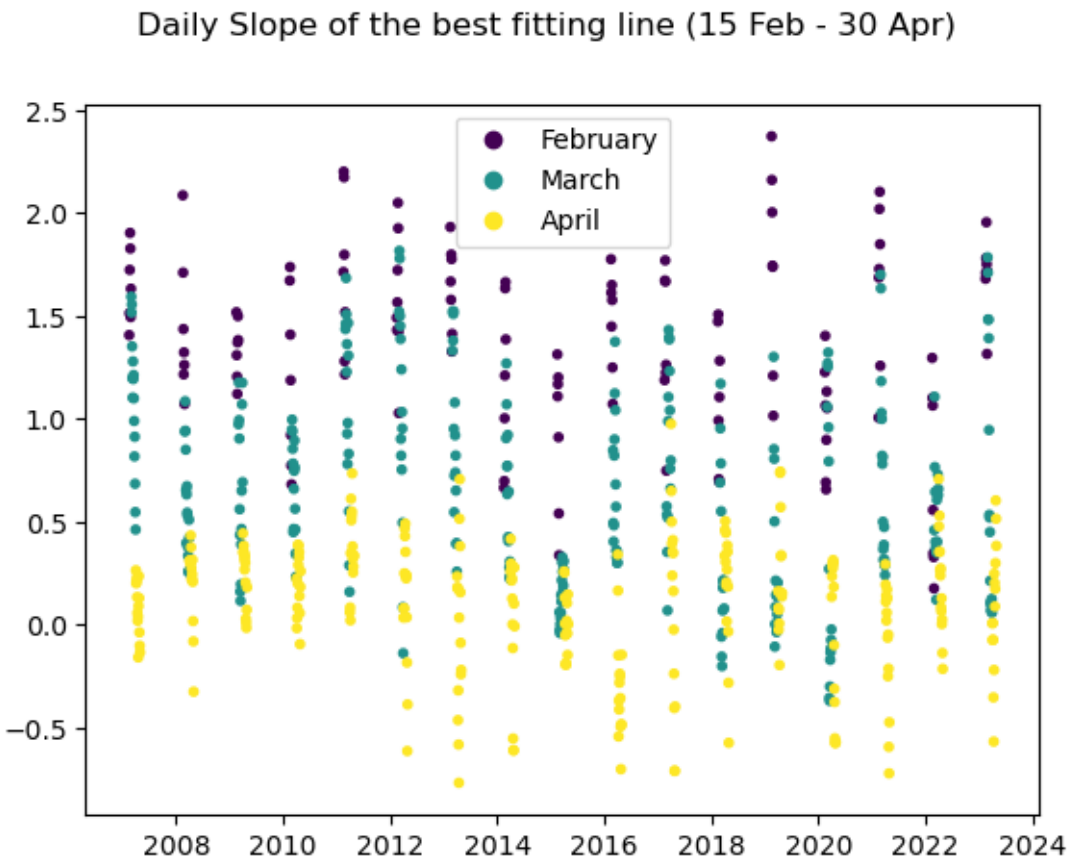
0.14 Other Years (Daily)

[ ]:
```
r_all2 = np.array([])
rms_all2 = np.array([])
slope_all2 = np.array([])

for i in tqdm(range (0, len(ds.time_counter))):

    dataset = ds.isel(time_counter=i)

    drivers, diat, _ = datasets_preparation(dataset)

    r, rms, m = regressor3(drivers, diat)

    r_all2 = np.append(r_all2,r)
    rms_all2 = np.append(rms_all2,rms)
    slope_all2 = np.append(slope_all2,m)

plotting2(r_all2, 'Correlation Coefficients')
plotting2(rms_all2, 'Mean Square Errors')
plotting2(slope_all2, 'Slope of the best fitting line')
```

```
  0%|          | 0/640 [00:00<?, ?it/s]
```



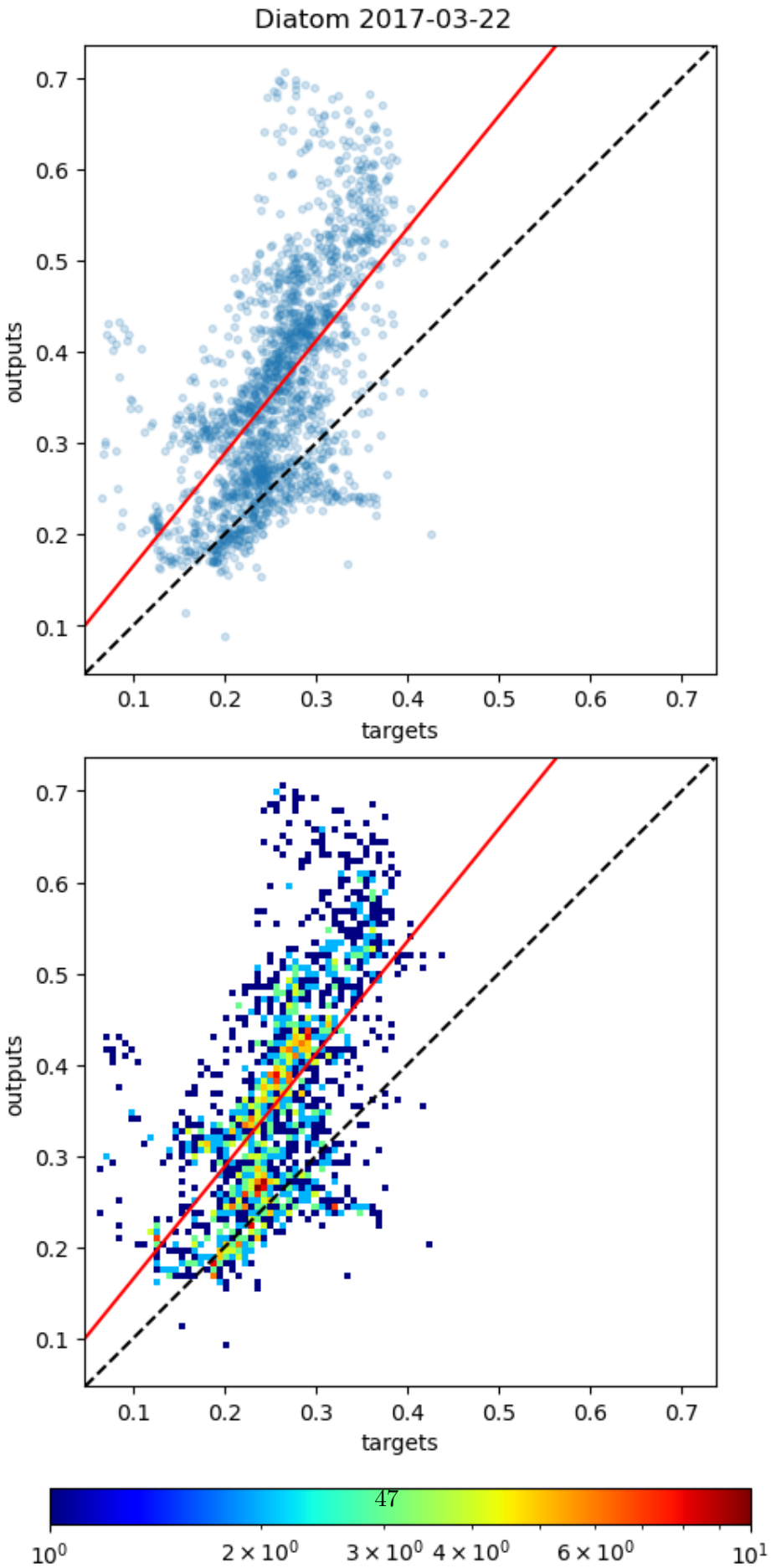

Daily Mean Square Errors (15 Feb - 30 Apr)
February
March
April
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
2008
2010
2012
2014
2016
2018
2020
2022
2024

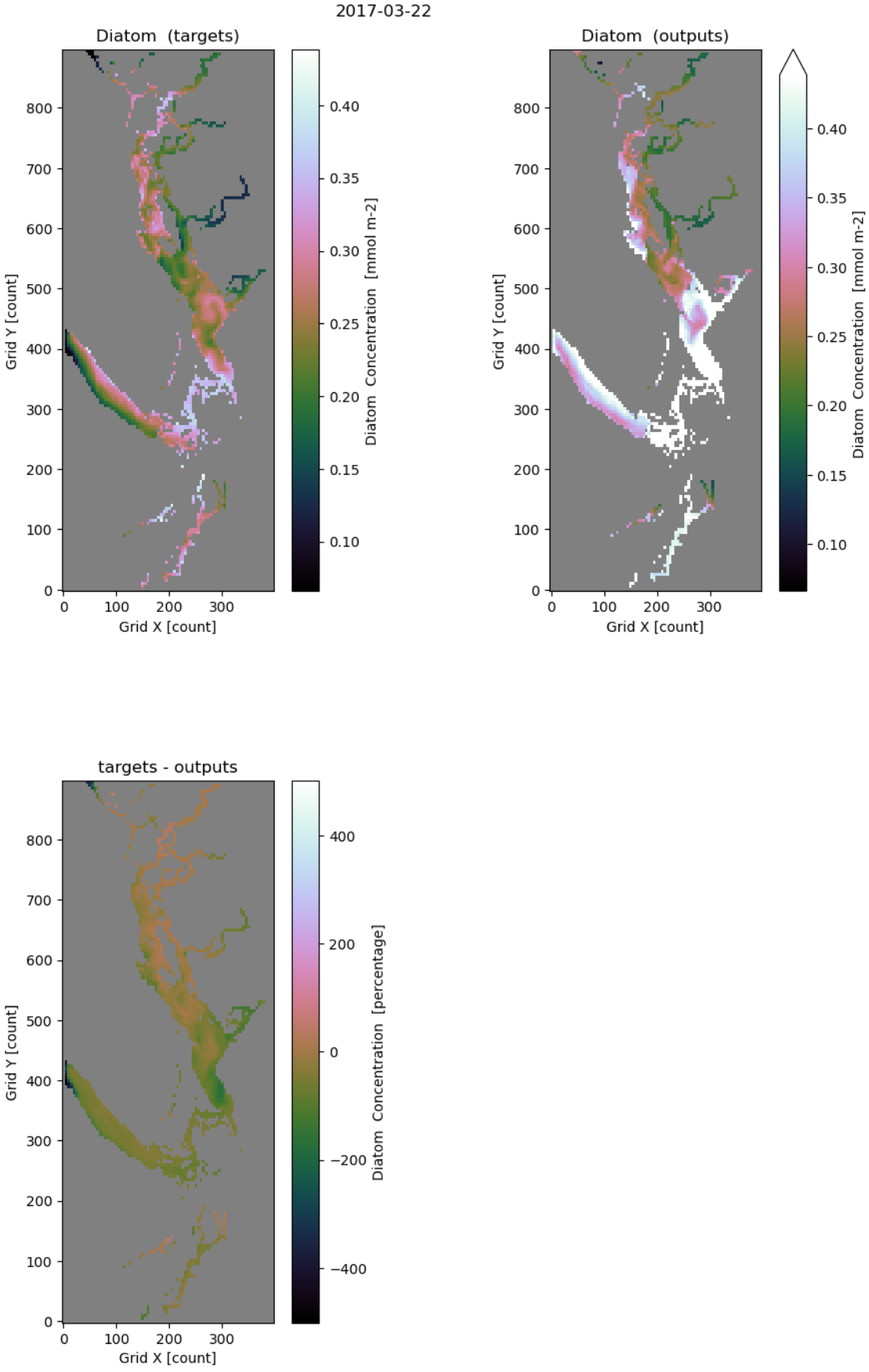


1 Daily Maps

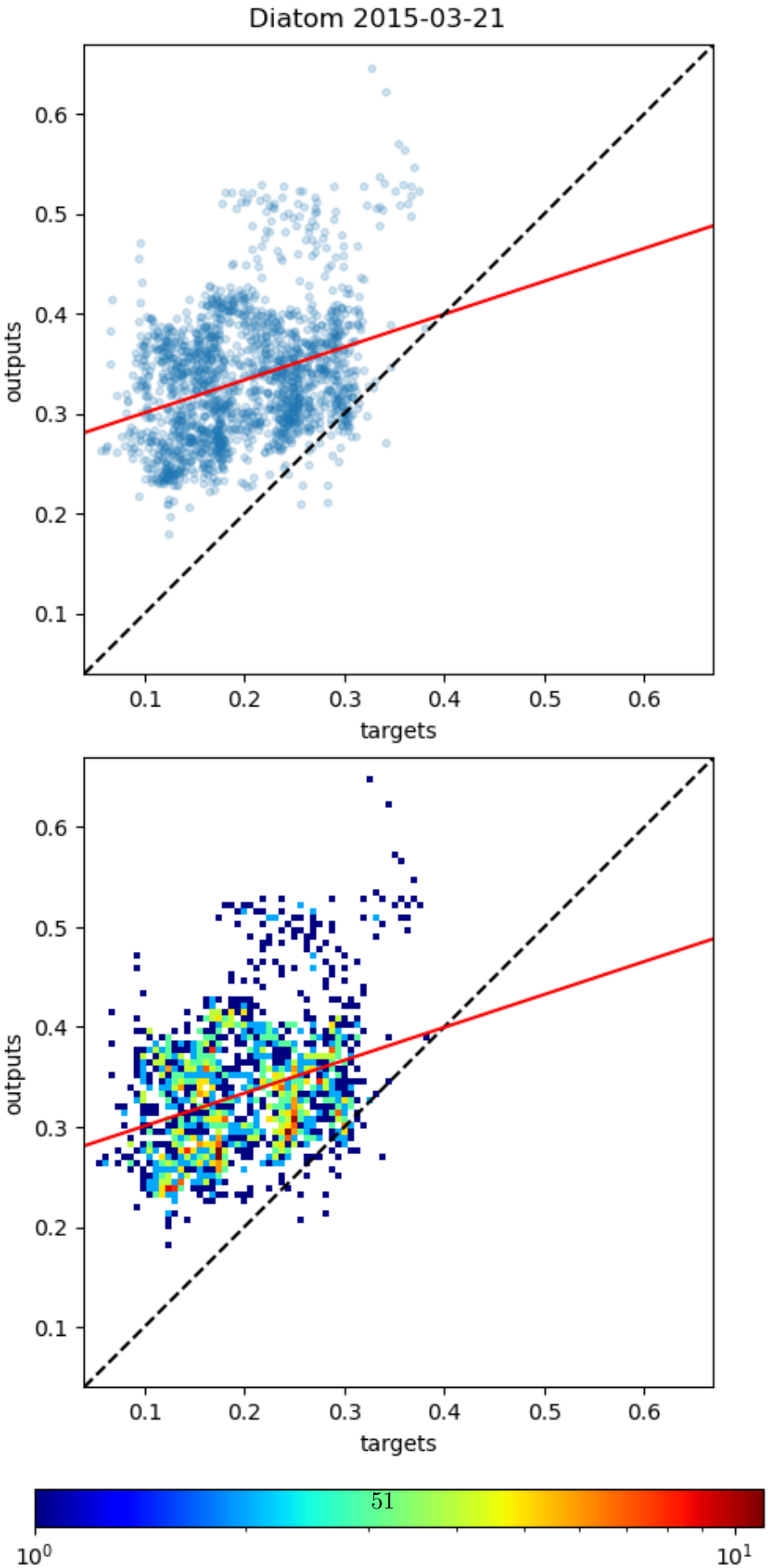
[ ]:
```
maps = random.sample(range(0,len(ds.time_counter)),10)

for i in tqdm(maps):

    dataset = ds.isel(time_counter=i)
    drivers, diat, indx = datasets_preparation(dataset)

    diat_i = dataset['Diatom']

    regressor4(drivers, diat, 'Diatom ')
```

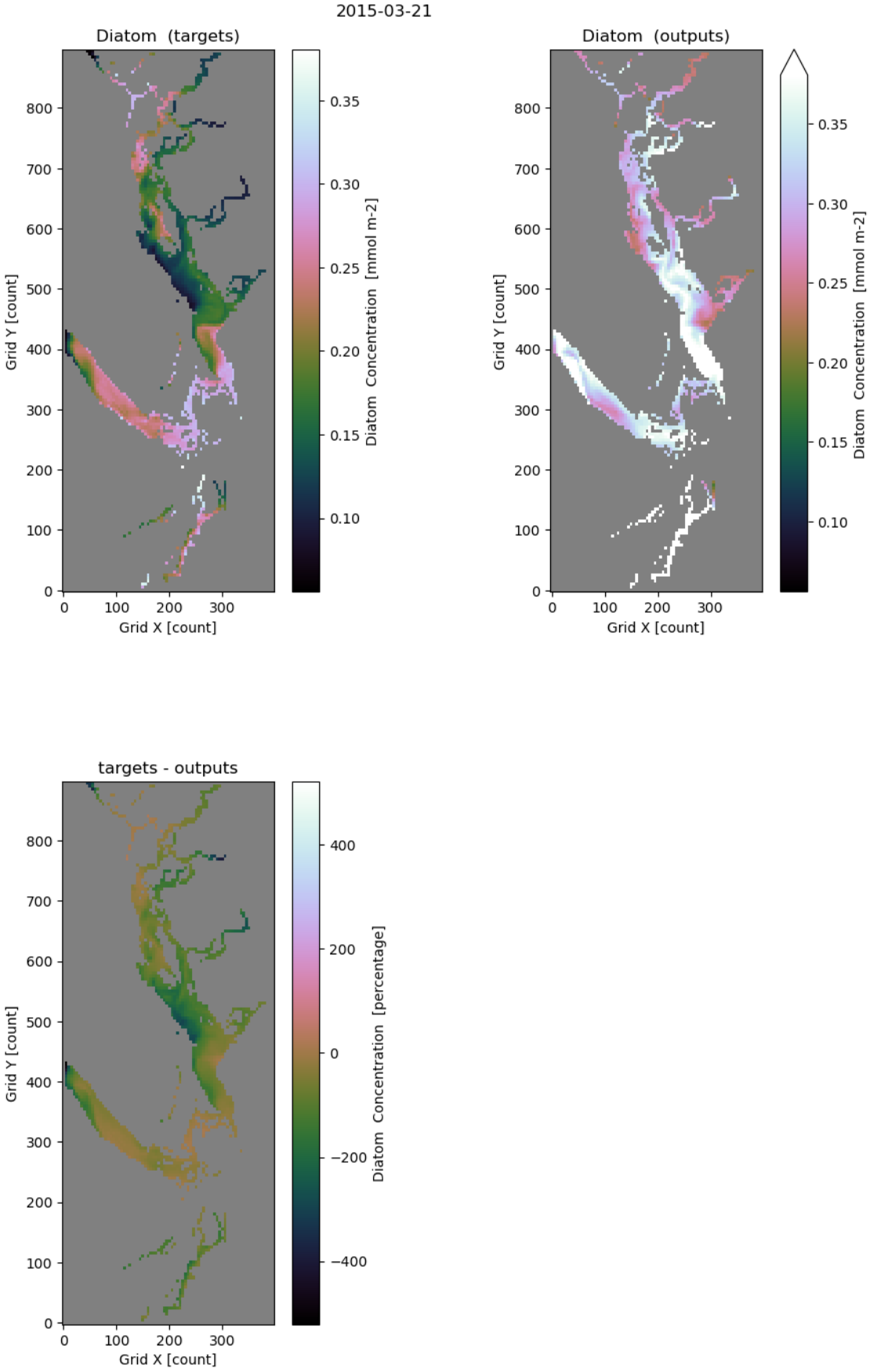
```
  0%|          | 0/10 [00:00<?, ?it/s]
The amount of data points is 1863
The slope of the best fitting line is  1.234
The correlation coefficient is: 0.611
 The mean square error is: 0.01974
```


Diatom 2017-03-22
outputs
targets
0.1
0.2
0.3
0.4
0.5
0.6
0.7
10^0
2×10^0
3×10^0
4×10^0
6×10^0
10^1

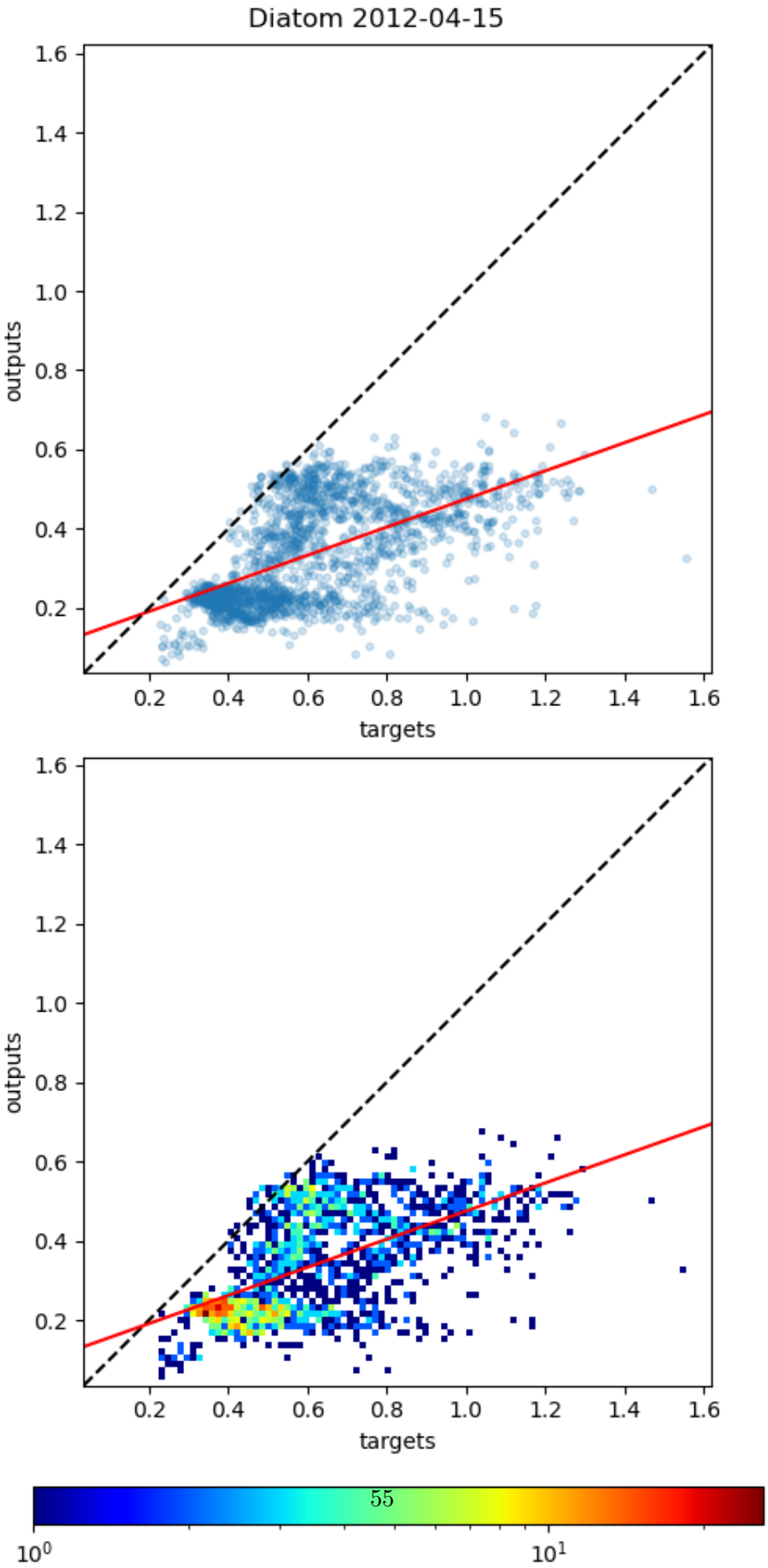

2017-03-22
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]
0
100
200
300
400
500
600
700
800
0.10
0.15
0.20
0.25
0.30
0.35
0.40
−400
−200
0
200
400

```
The amount of data points is 1863
The slope of the best fitting line is  0.329
The correlation coefficient is: 0.336
 The mean square error is: 0.02305
```

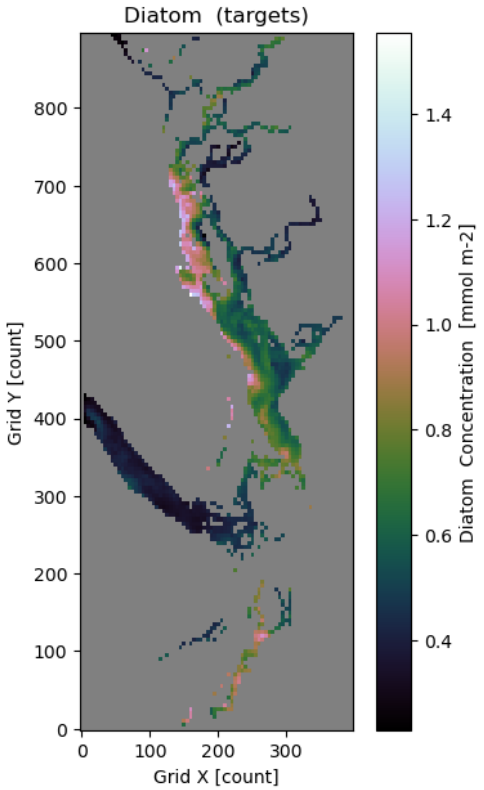

Diatom 2015-03-21
outputs
targets
0.1
0.2
0.3
0.4
0.5
0.6
51
10^0
10^1

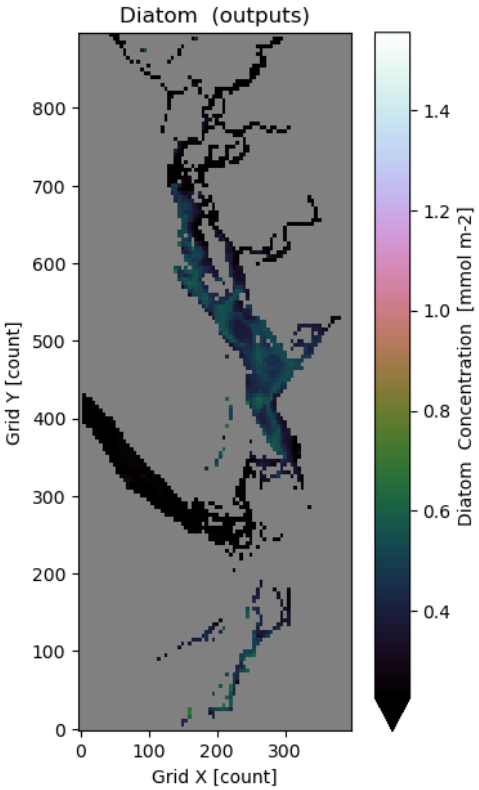

2015-03-21
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]
0
100
200
300
400
500
600
700
800
0.10
0.15
0.20
0.25
0.30
0.35
−400
−200
0
200
400

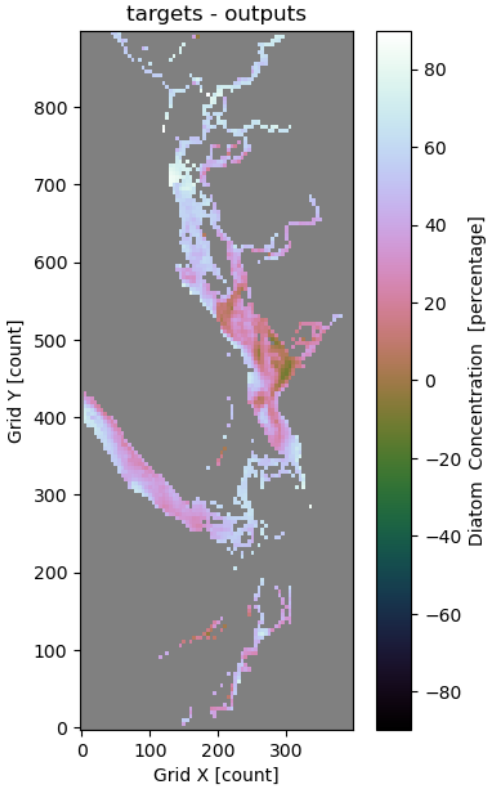
```
The amount of data points is 1863
The slope of the best fitting line is  0.355
The correlation coefficient is: 0.592
 The mean square error is: 0.10548
```


Diatom 2012-04-15
outputs
targets
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
outputs
targets
55
10^0
10^1

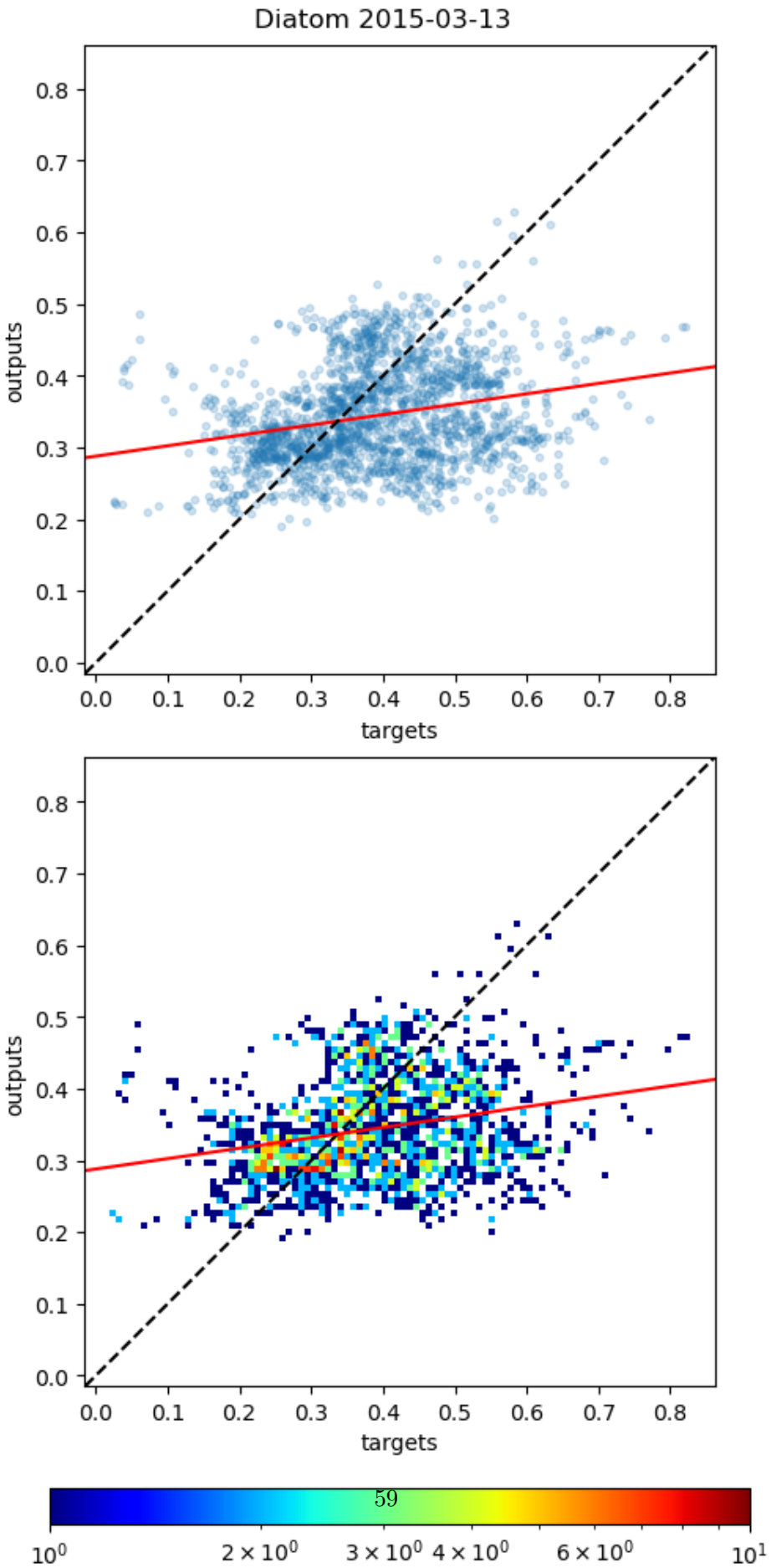
2012-04-15

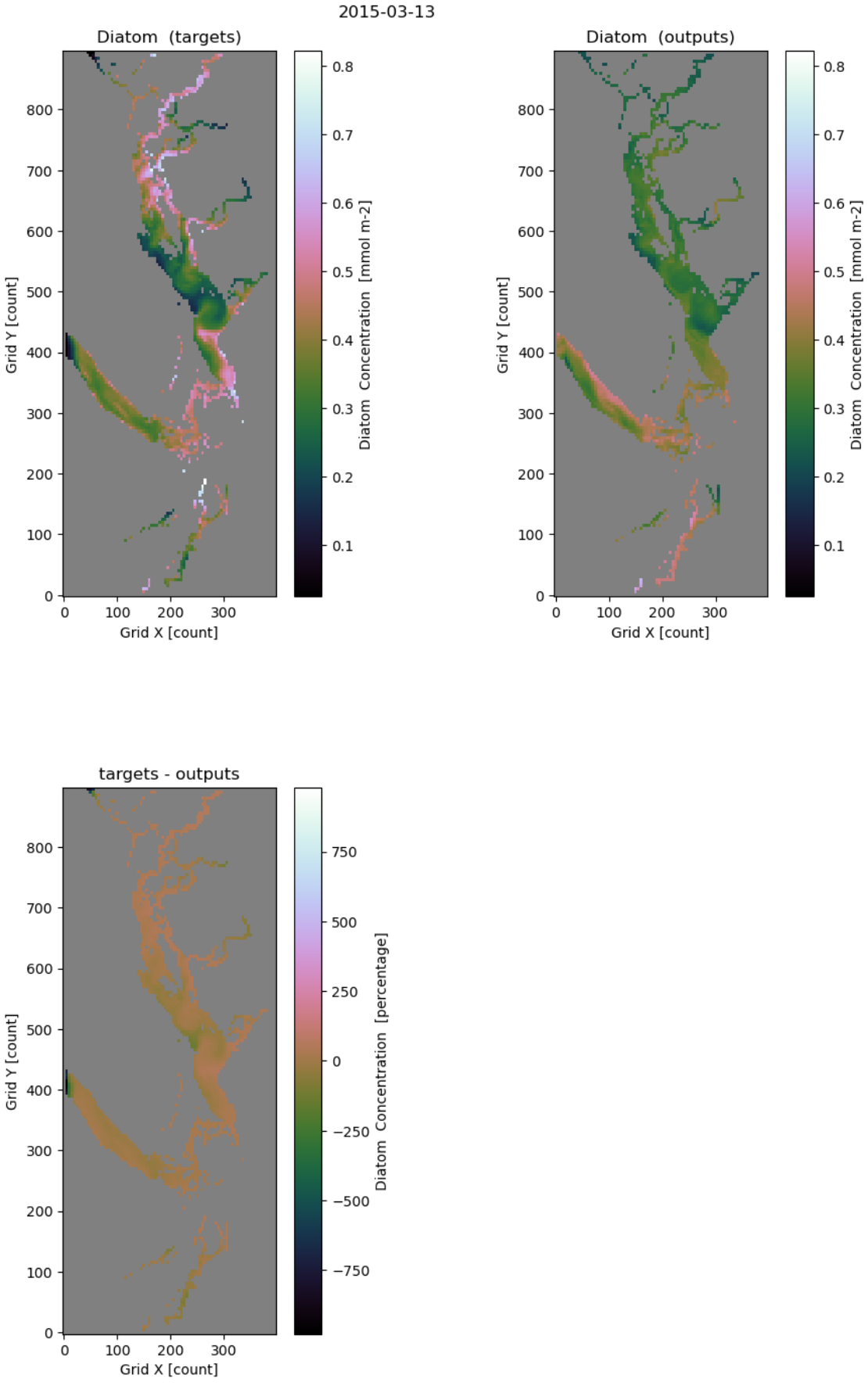

Diatom (targets)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]


Diatom (outputs)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]

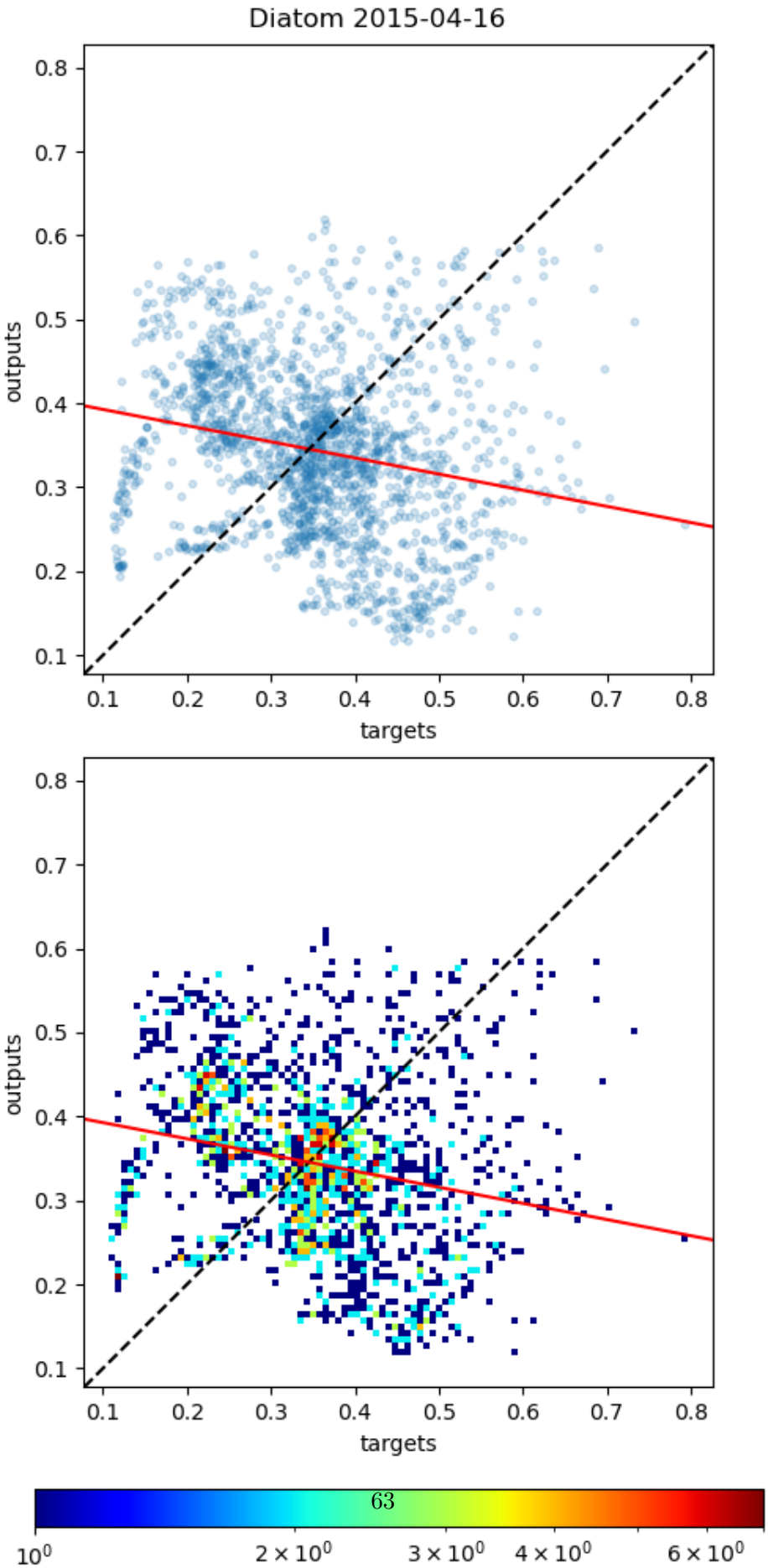

targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [percentage]

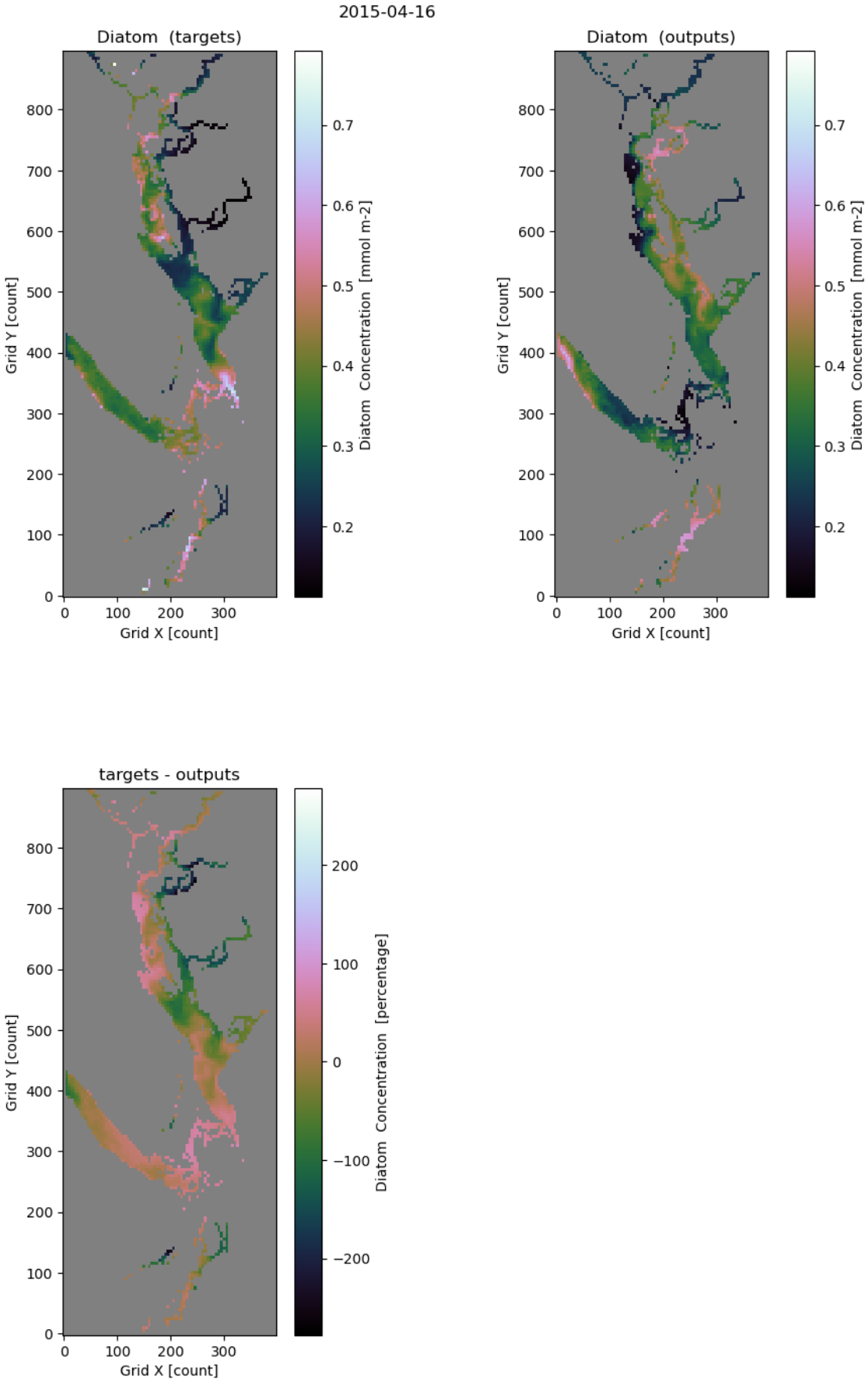
```
The amount of data points is 1863
The slope of the best fitting line is  0.145
The correlation coefficient is: 0.252
 The mean square error is: 0.01726
```


Diatom 2015-03-13
outputs
targets
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
59
10^0
2×10^0
3×10^0
4×10^0
6×10^0
10^1

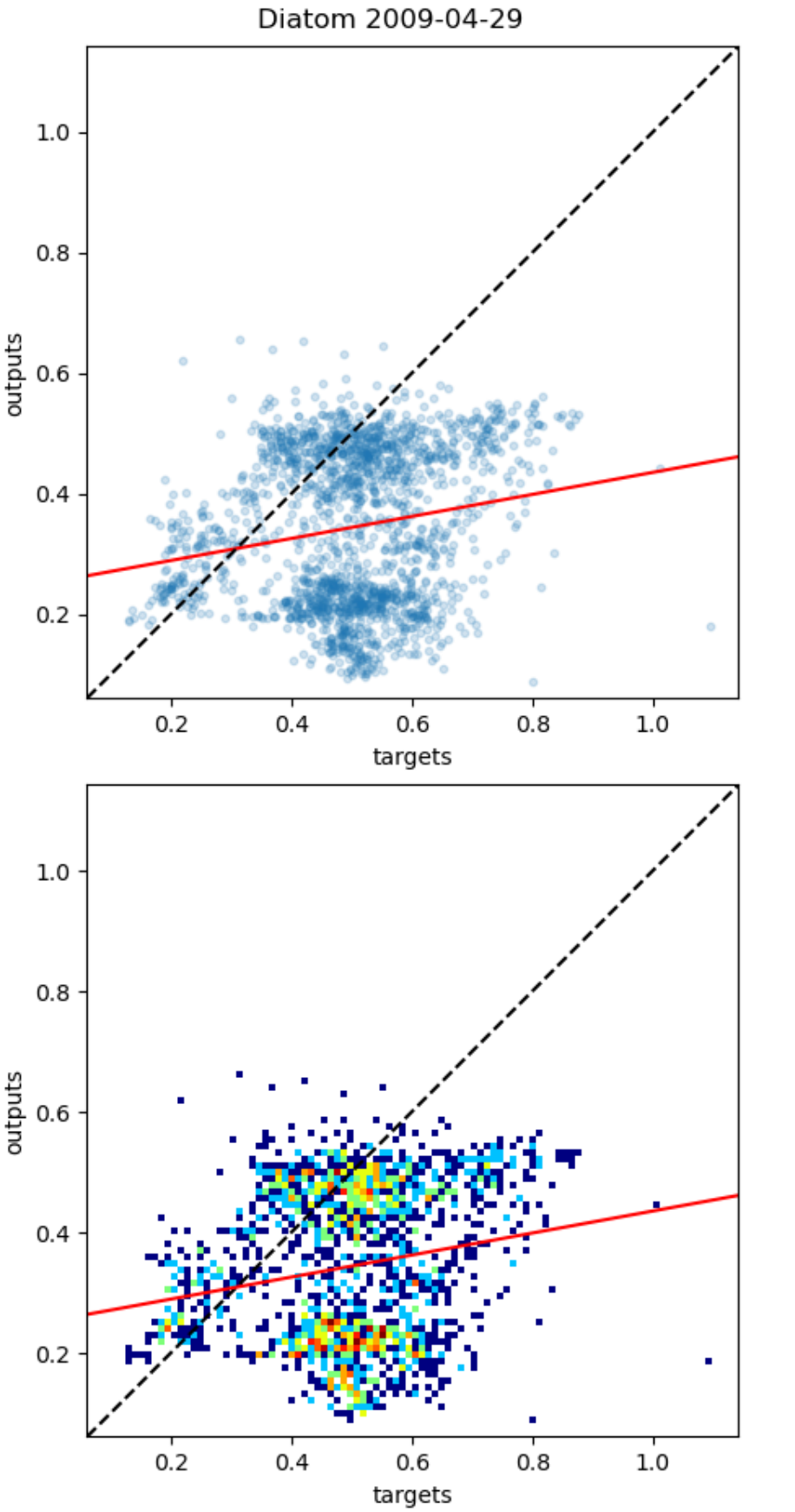

2015-03-13
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

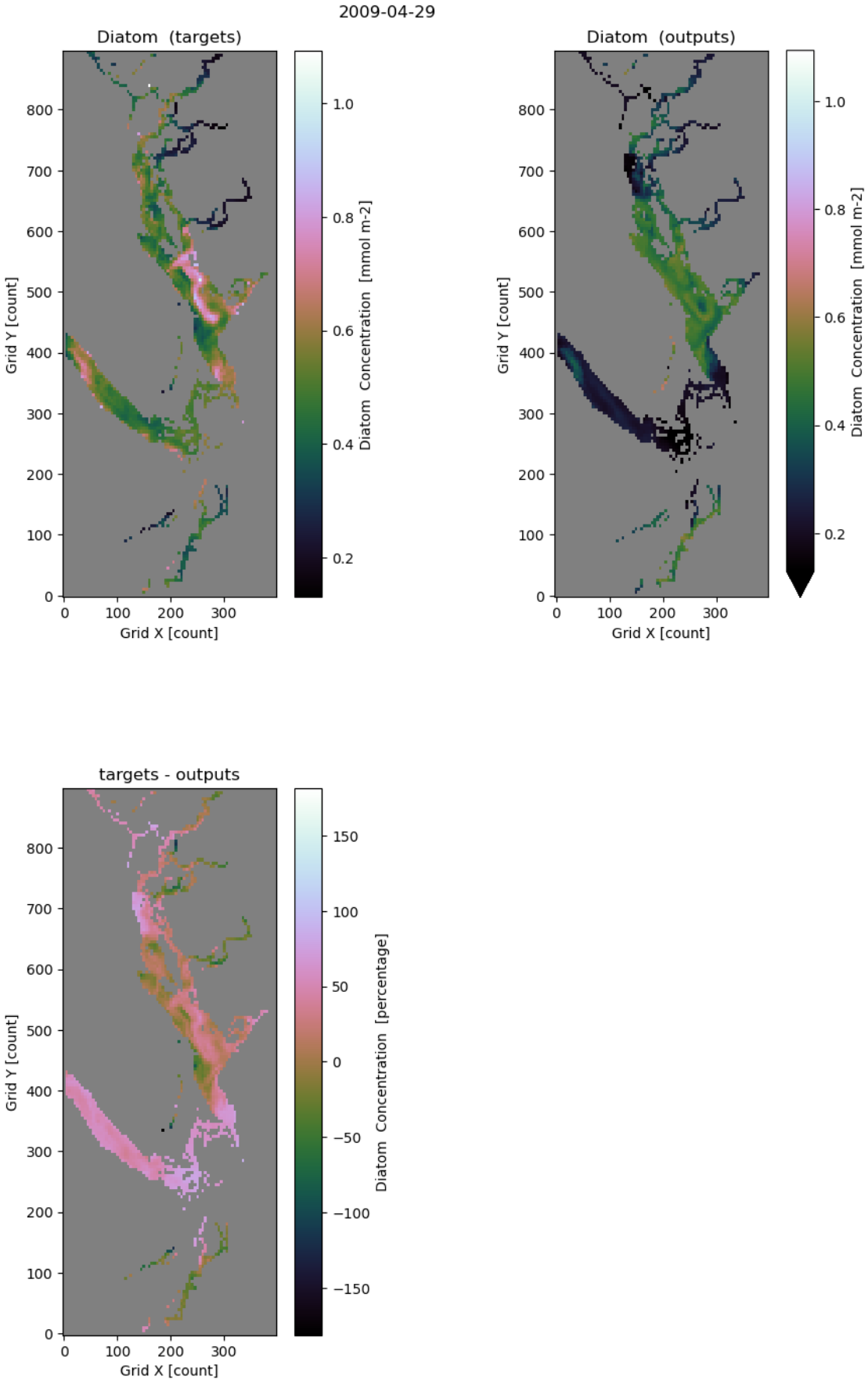
```
The amount of data points is 1863
The slope of the best fitting line is  -0.192
The correlation coefficient is: -0.218
 The mean square error is: 0.02733
```


Diatom 2015-04-16
outputs
targets
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
63
10^0
2×10^0
3×10^0
4×10^0
6×10^0

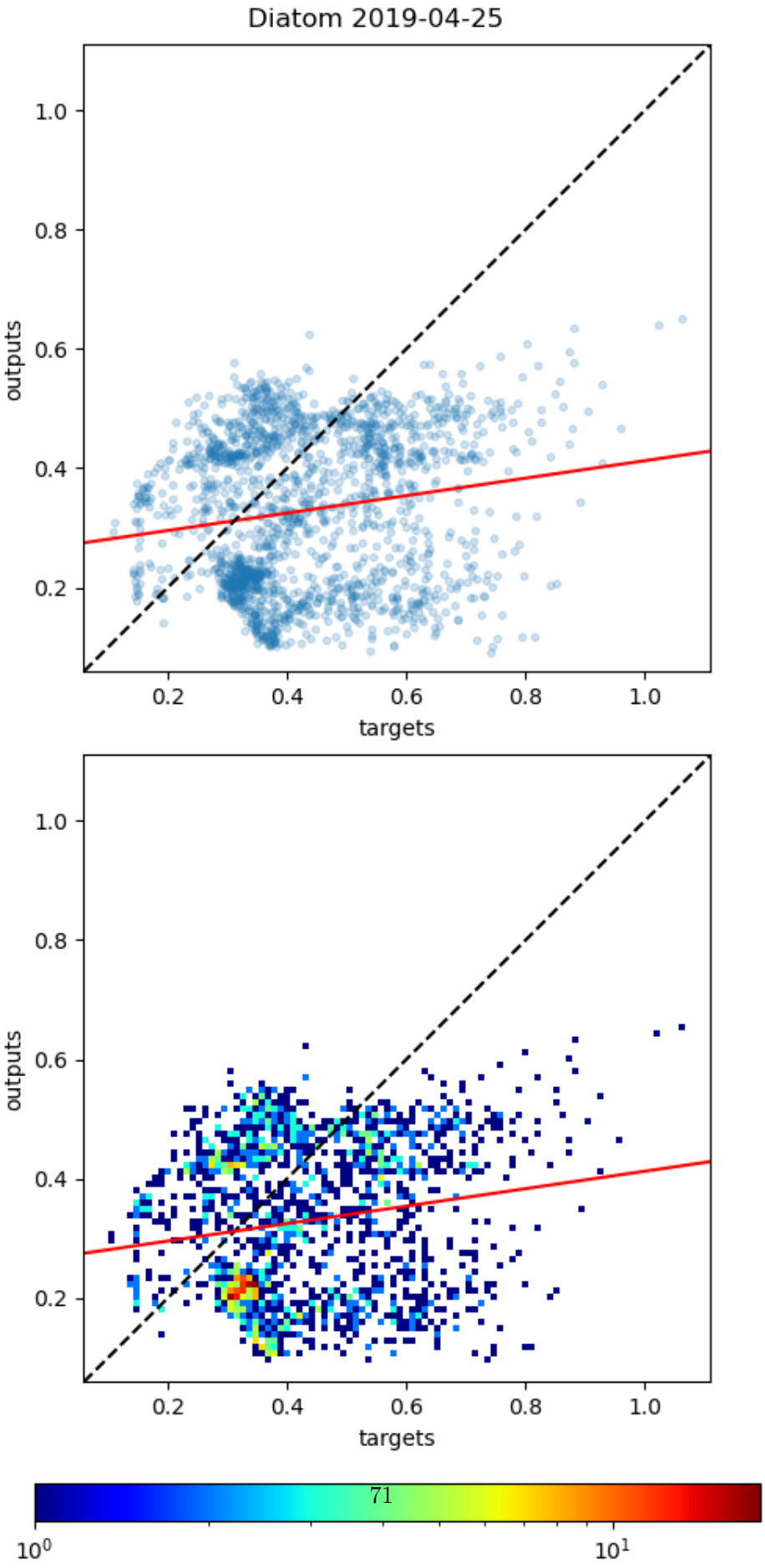

2015-04-16
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

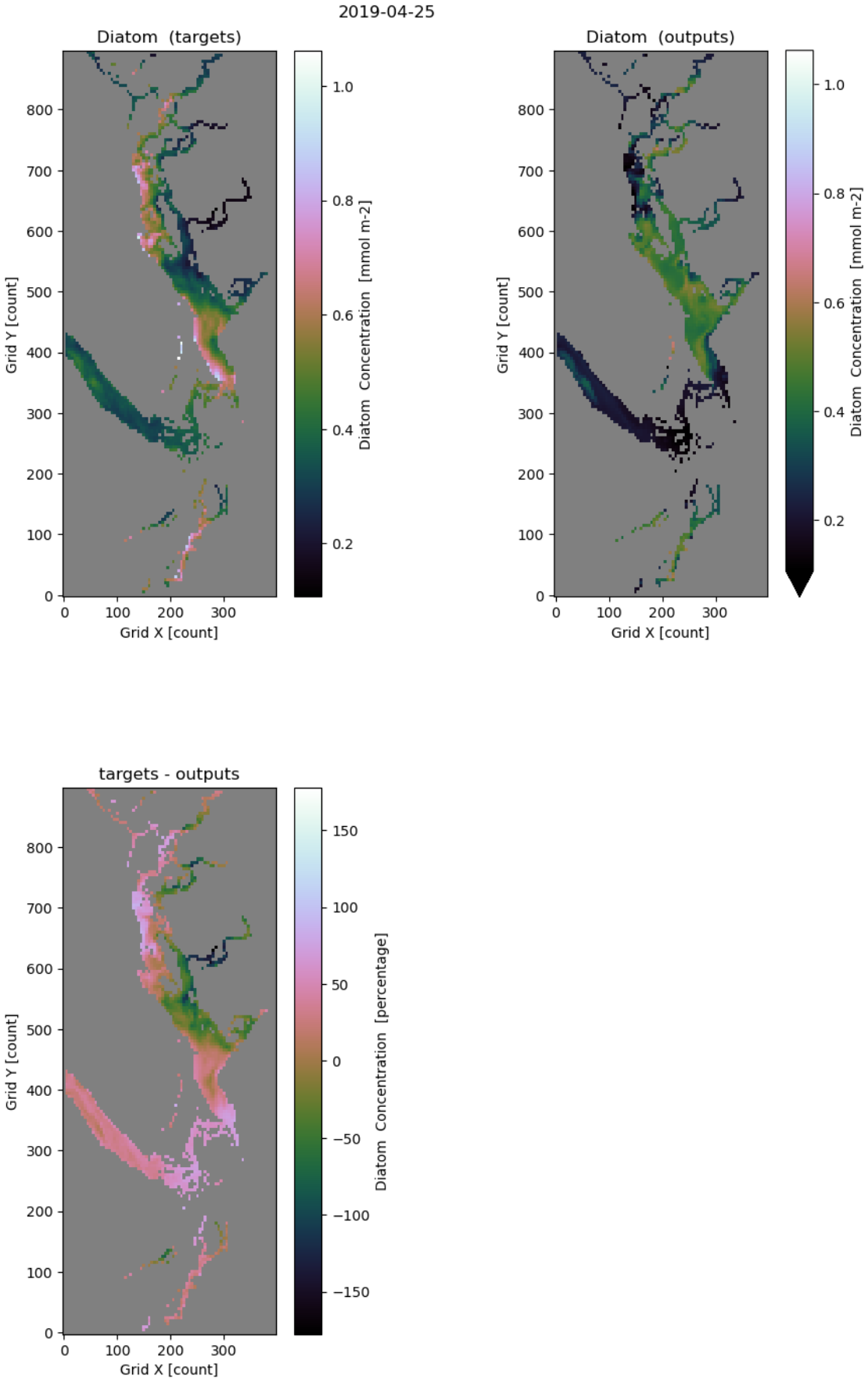
```
The amount of data points is 1863
The slope of the best fitting line is  0.183
The correlation coefficient is: 0.184
 The mean square error is: 0.05203
```


Diatom 2009-04-29
outputs
targets
1.0
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1.0
10^0
2×10^0
3×10^0
4×10^0
6×10^0

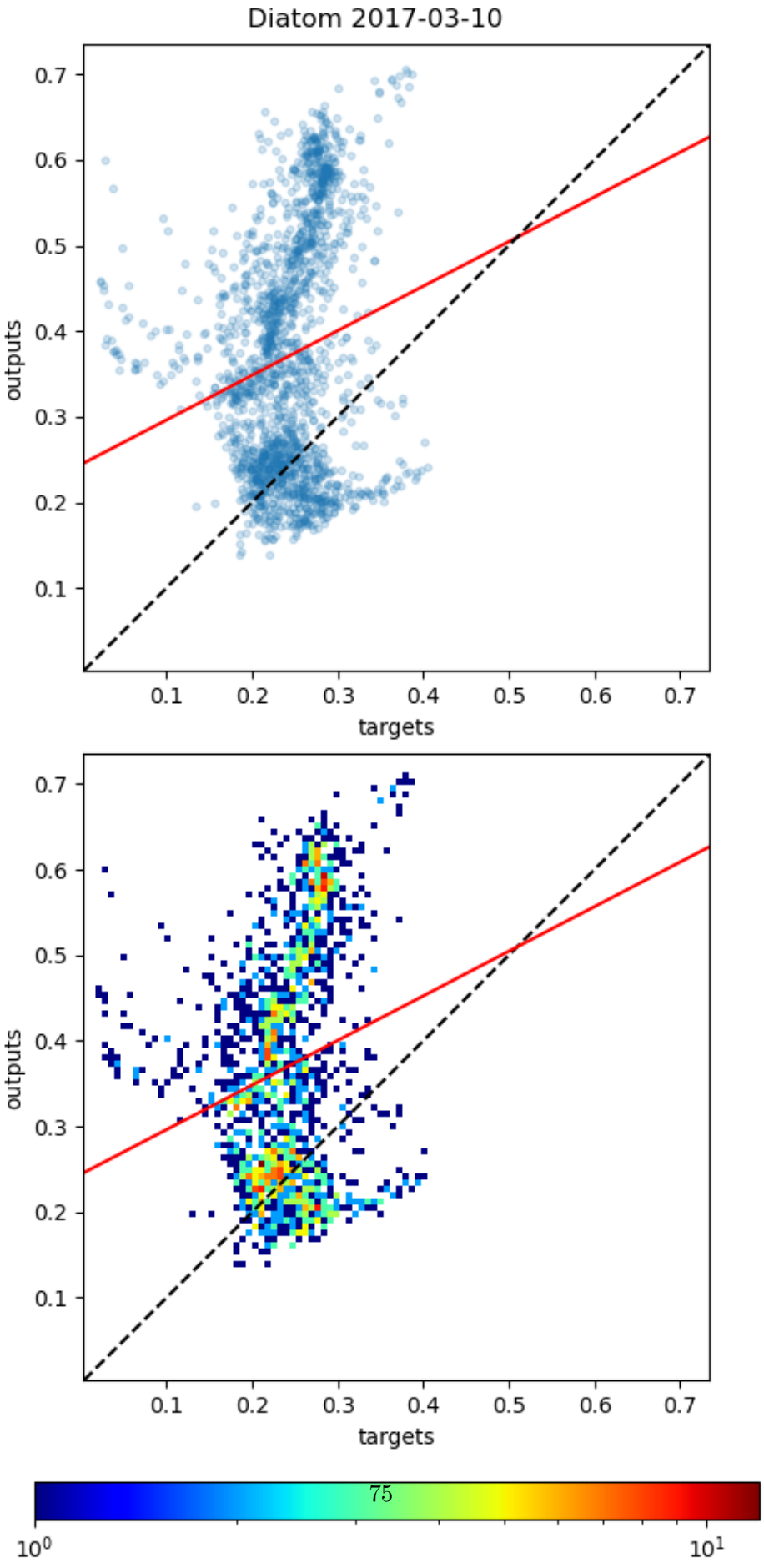

2009-04-29
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

```
The amount of data points is 1863
The slope of the best fitting line is  0.146
The correlation coefficient is: 0.169
 The mean square error is: 0.04219
```

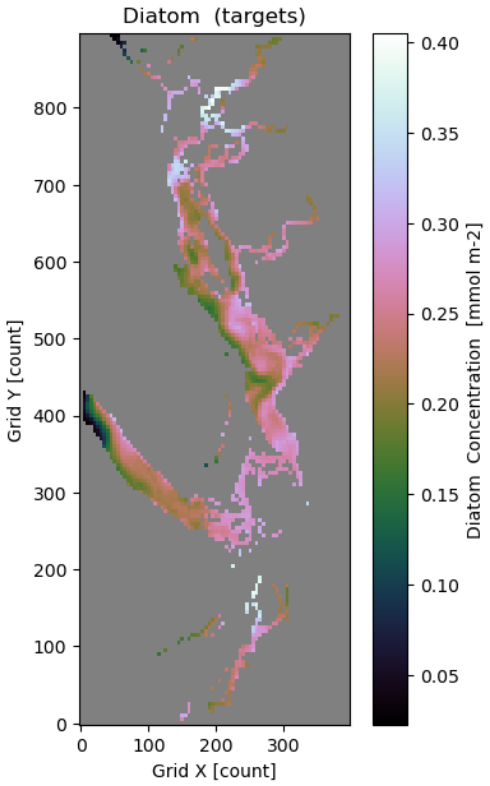

Diatom 2019-04-25
outputs
targets
1.0
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1.0
71
10^0
10^1

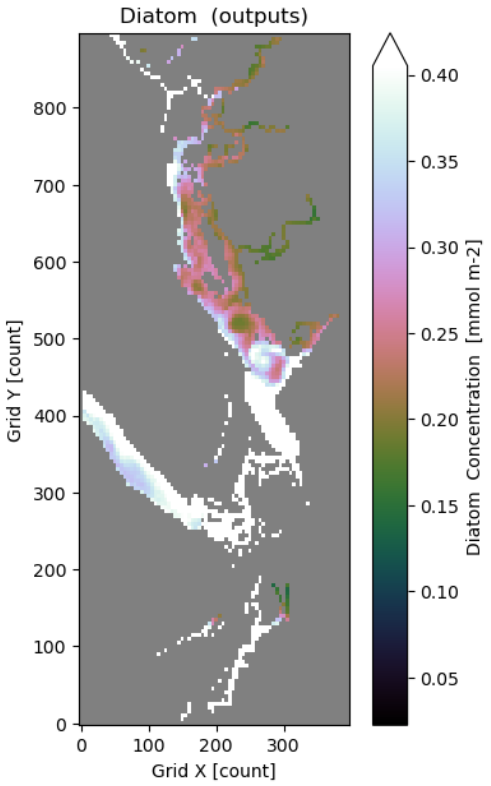

2019-04-25
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

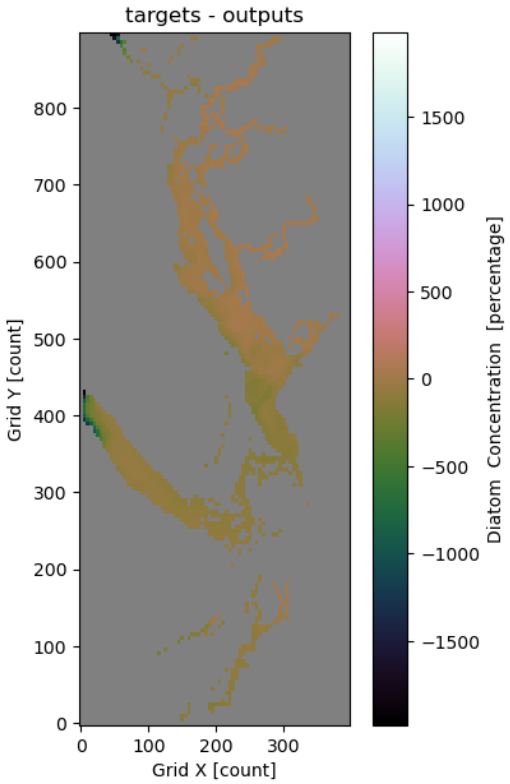
```
The amount of data points is 1863
The slope of the best fitting line is  0.521
The correlation coefficient is: 0.192
 The mean square error is: 0.03589
```


Diatom 2017-03-10
outputs
targets
0.1
0.2
0.3
0.4
0.5
0.6
0.7
75
10^0
10^1

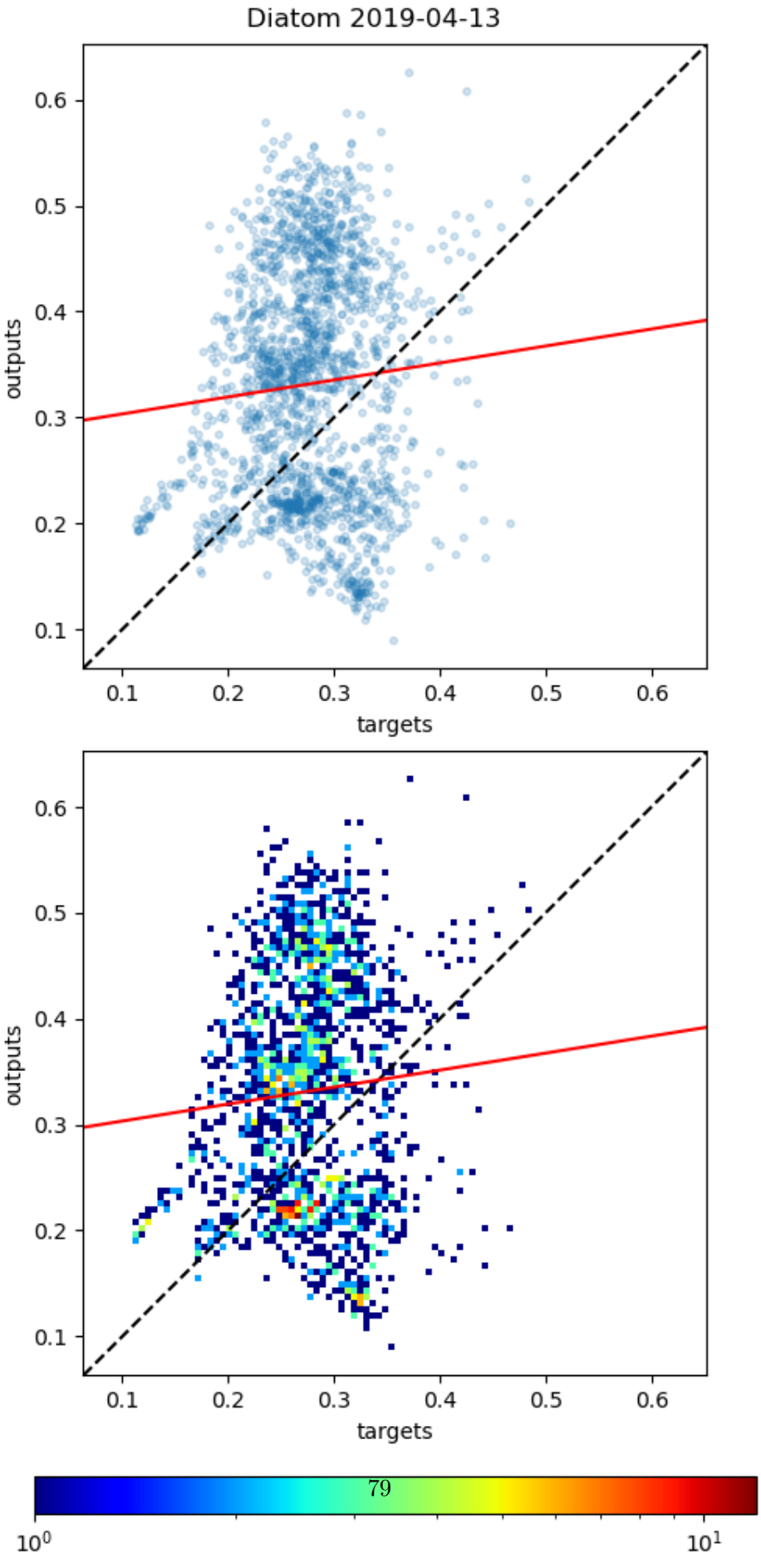
2017-03-10

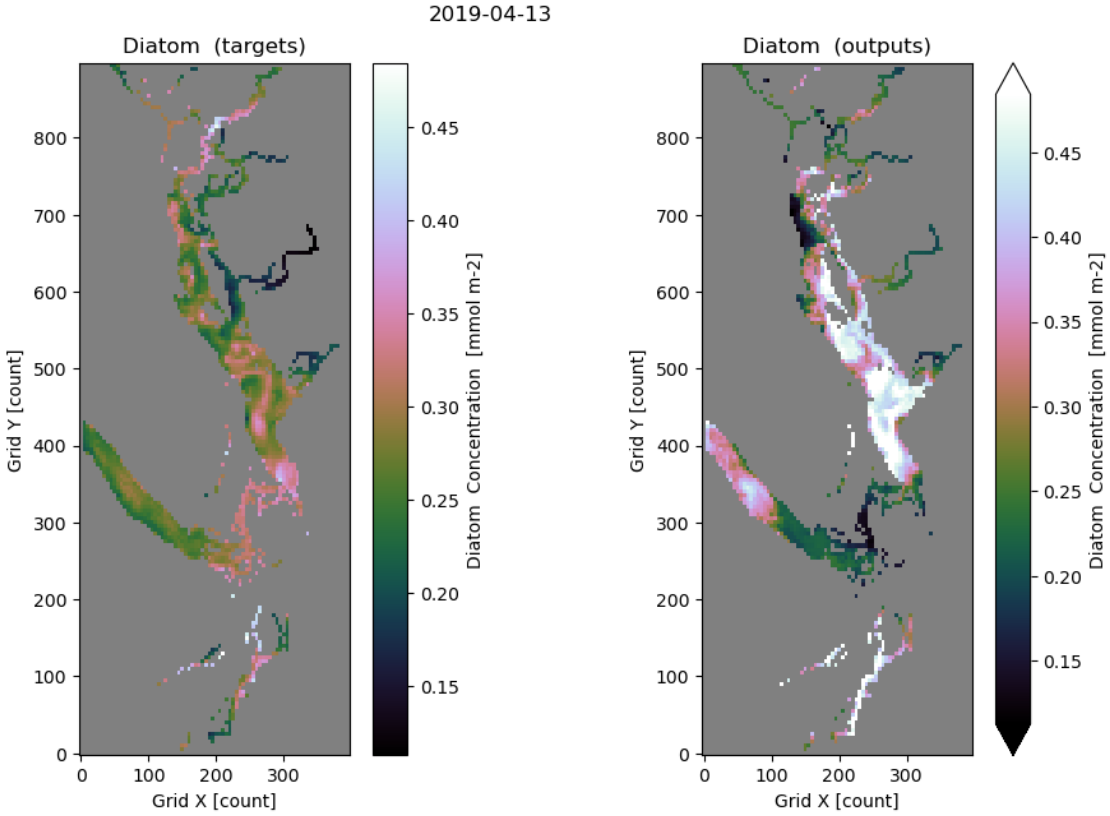

Diatom (targets)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05

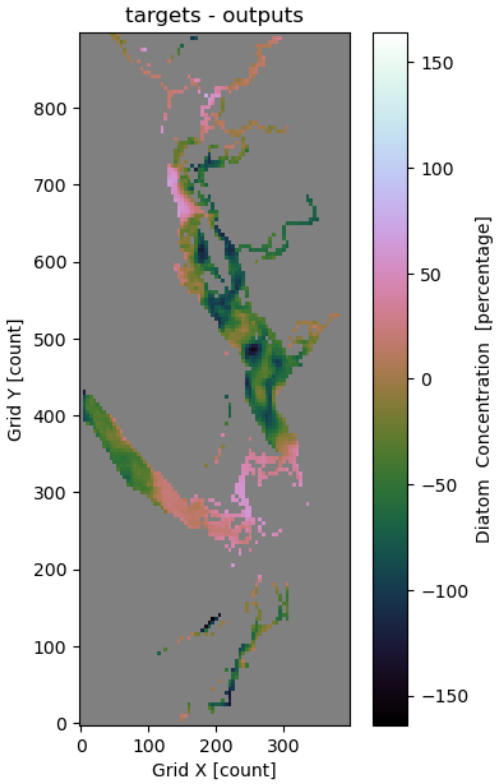

Diatom (outputs)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05


targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [percentage]
1500
1000
500
0
−500
−1000
−1500

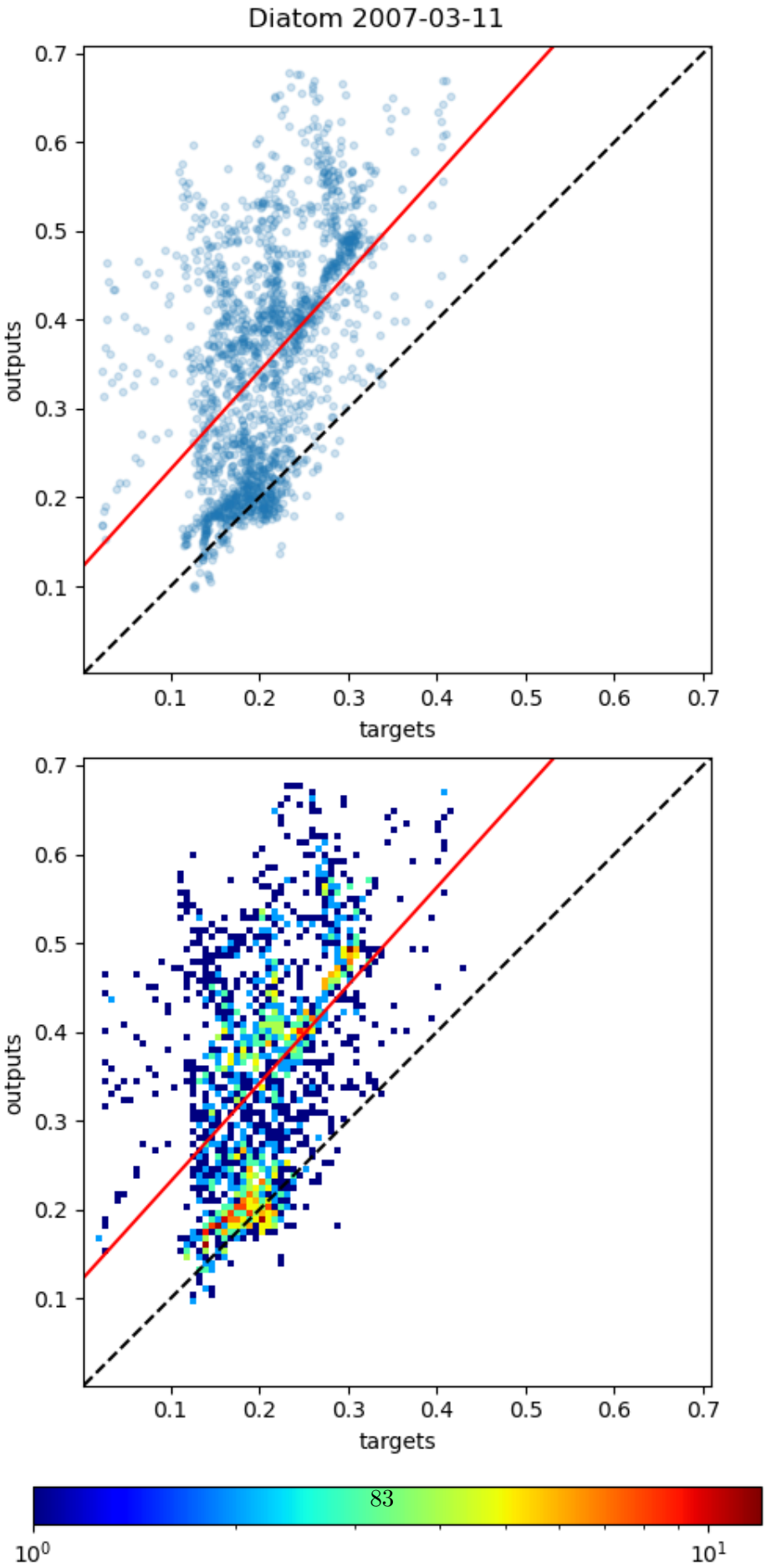
```
The amount of data points is 1863
The slope of the best fitting line is  0.16
The correlation coefficient is: 0.076
 The mean square error is: 0.01705
```

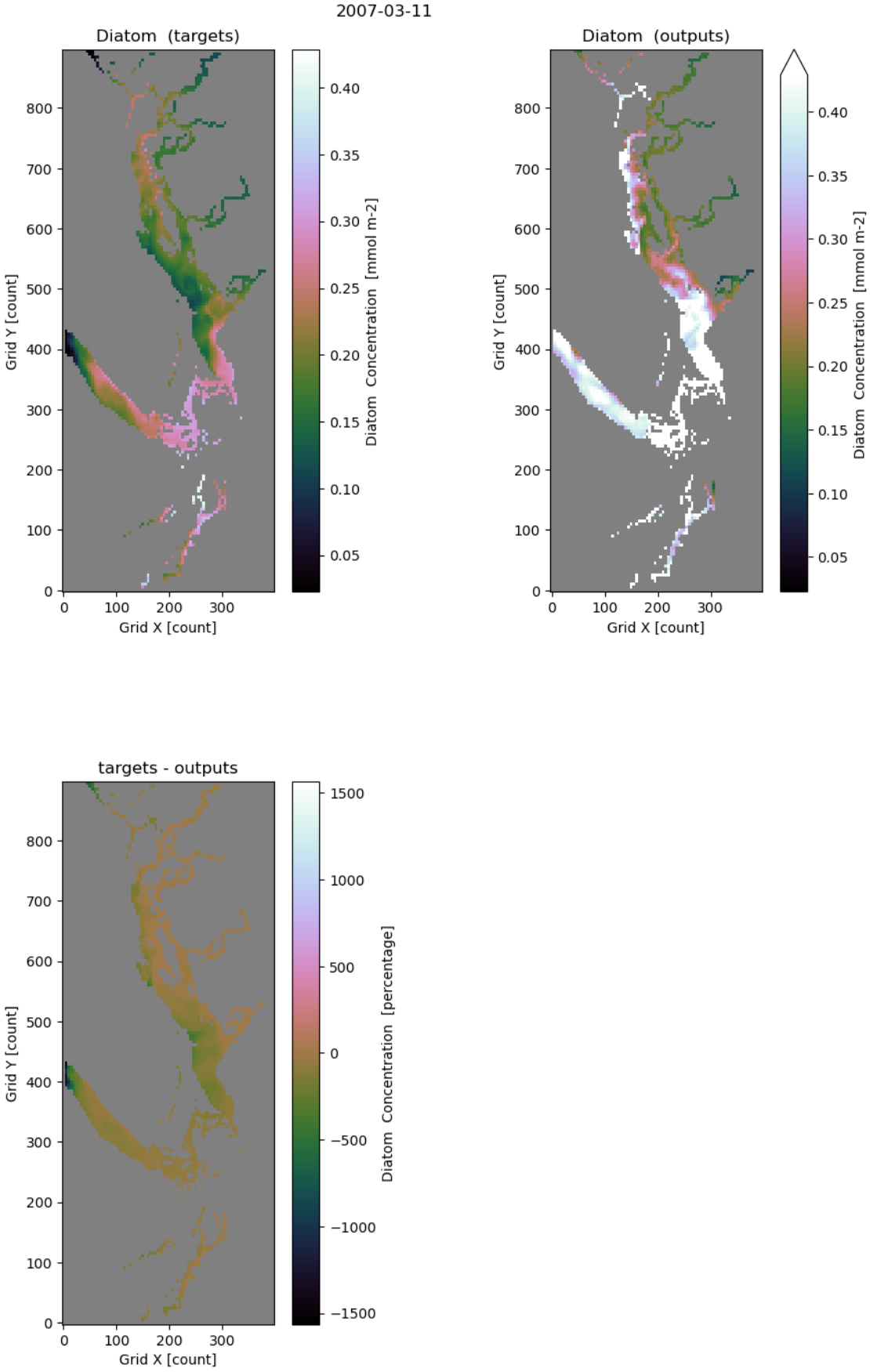

Diatom 2019-04-13
outputs
targets
0.1
0.2
0.3
0.4
0.5
0.6
79
10^0
10^1


2019-04-13
Diatom (targets)
Diatom (outputs)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
0
100
200
300
400
500
600
700
800
0.15
0.20
0.25
0.30
0.35
0.40
0.45


targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [percentage]
0
100
200
300
400
500
600
700
800
−150
−100
−50
0
50
100
150

```
The amount of data points is 1863
The slope of the best fitting line is  1.104
The correlation coefficient is: 0.508
 The mean square error is: 0.03292
```


Diatom 2007-03-11
outputs
targets
0.1
0.2
0.3
0.4
0.5
0.6
0.7
83
10^0
10^1


2007-03-11
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

[ ]: