reg_year_r_random_points_new

March 2, 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(hidden_layer_sizes=400, alpha=0.002)
    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((898*398),np.nan)
    indx2[indx[0]] = outputs
    model = np.reshape(indx2,(898,398))

    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)
```

1 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))
```

1.1 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)
```

1.2 Plotting

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

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

1.3 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()
```

1.4 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()
```

1.5 Training (Random Points)

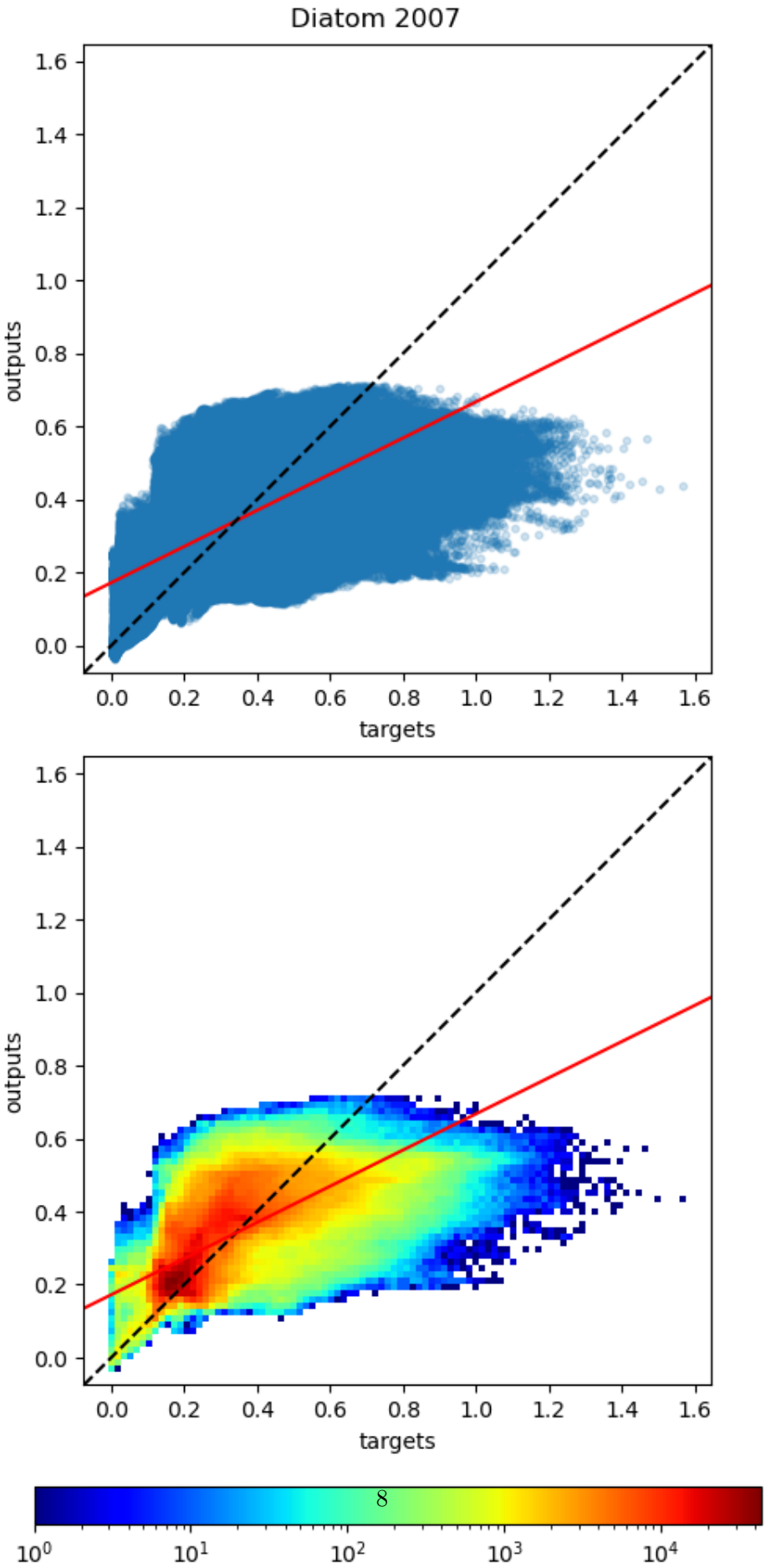
[ ]:
```
ds = xr.open_dataset('/data/ibougoudis/MOAD/files/integrated_model_var_old.nc')

# 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)
```

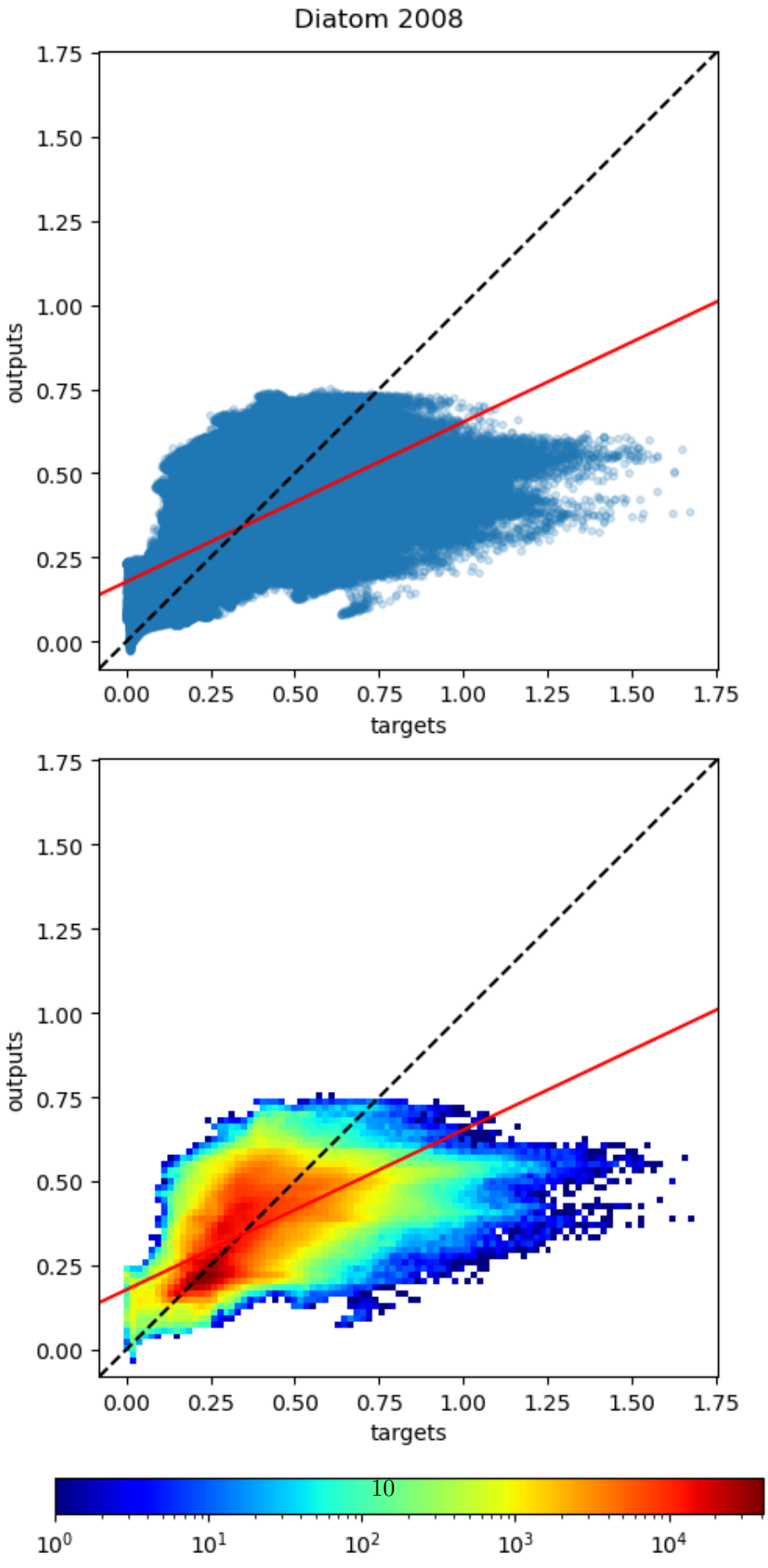
1.6 Other Years (Anually)

[ ]:
```
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')
```

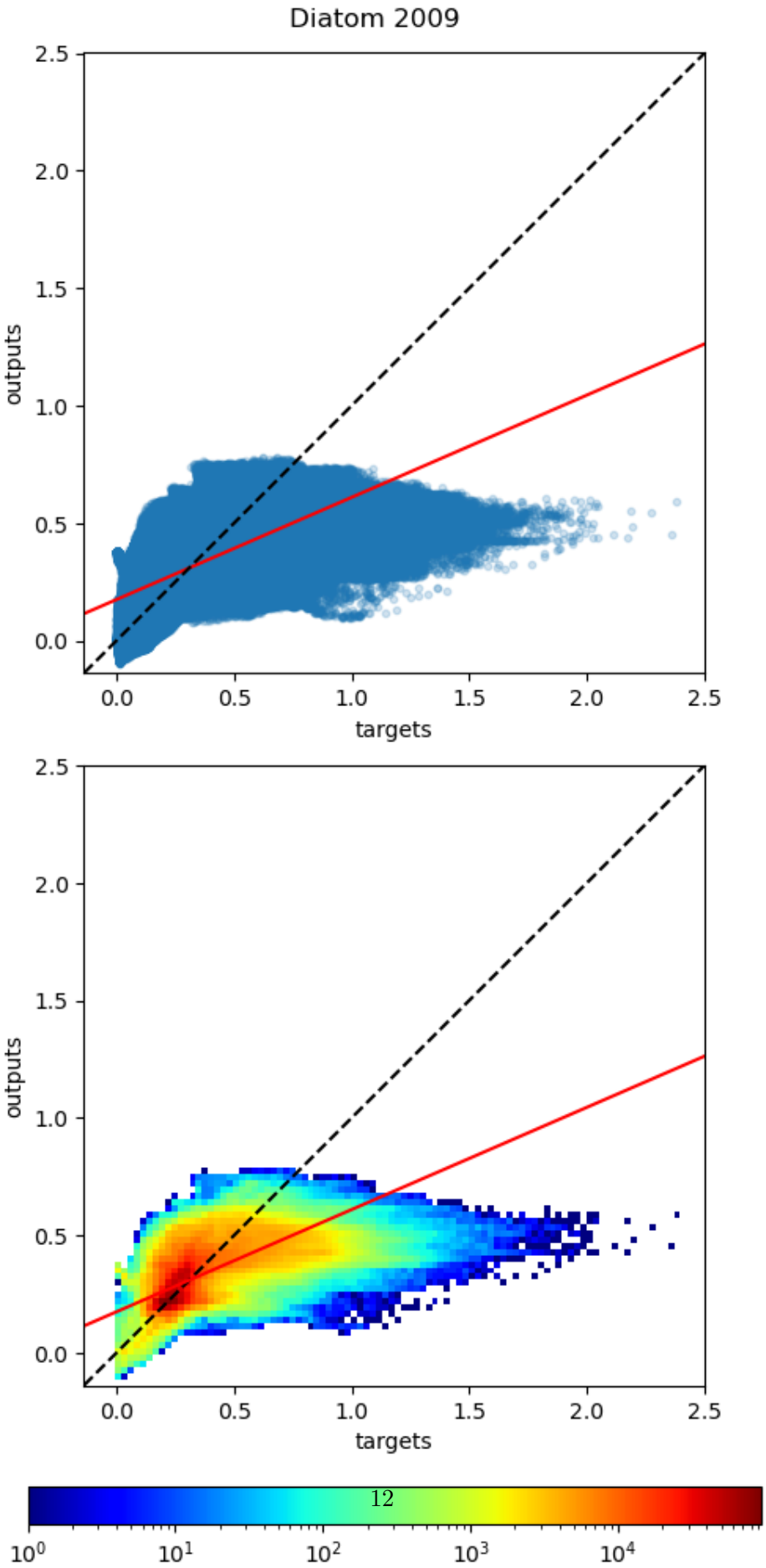
```
  0%|          | 0/17 [00:00<?, ?it/s]
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)
The amount of data points is 3485925
The slope of the best fitting line is  0.495
The correlation coefficient is: 0.655
 The mean square error is: 0.01527
```


Diatom 2007
outputs
targets
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
outputs
targets
8
10^0
10^1
10^2
10^3
10^4

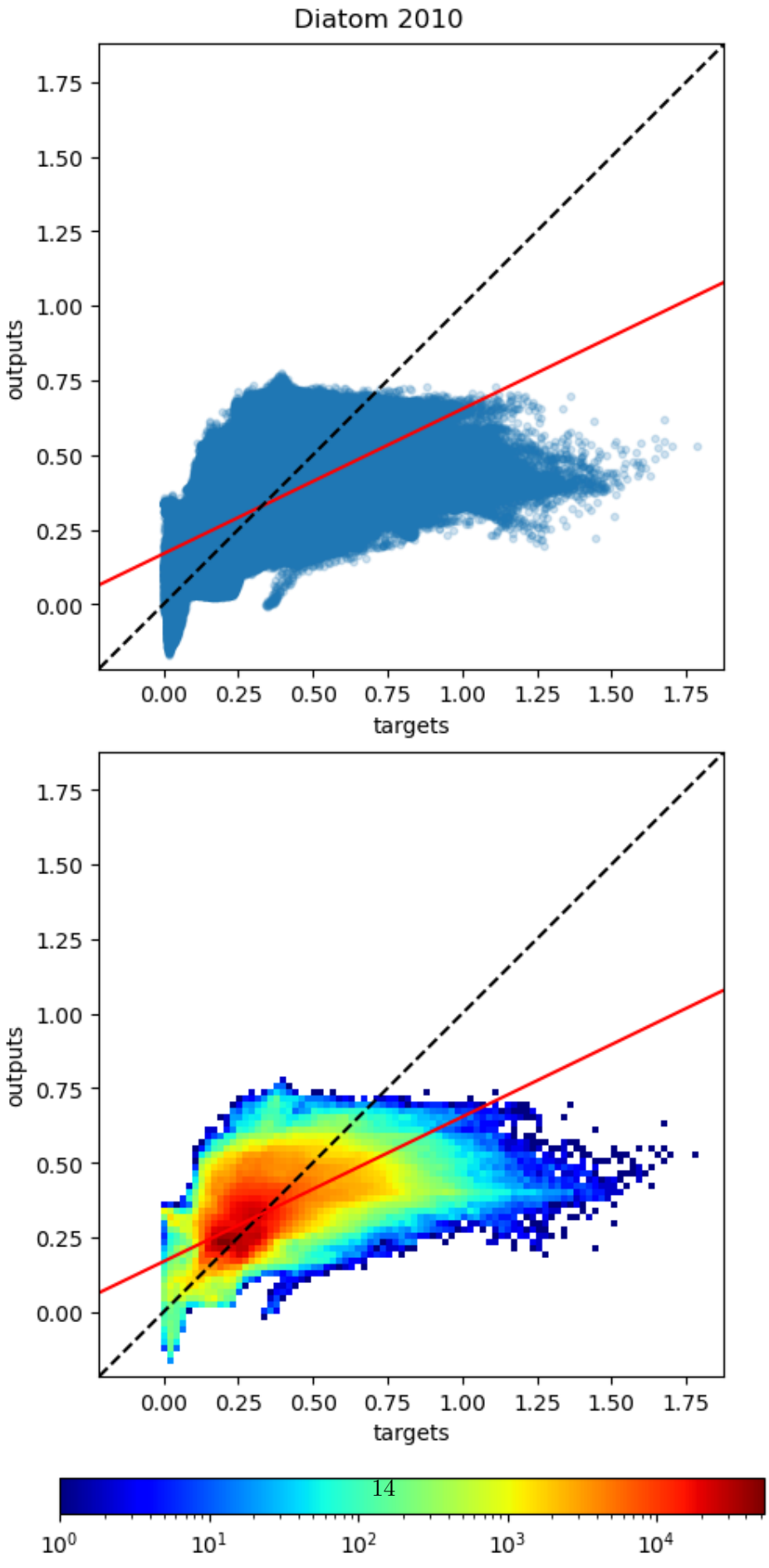
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3532404
The slope of the best fitting line is  0.475
The correlation coefficient is: 0.643
 The mean square error is: 0.01286
```


Diatom 2008
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
10
10^0
10^1
10^2
10^3
10^4

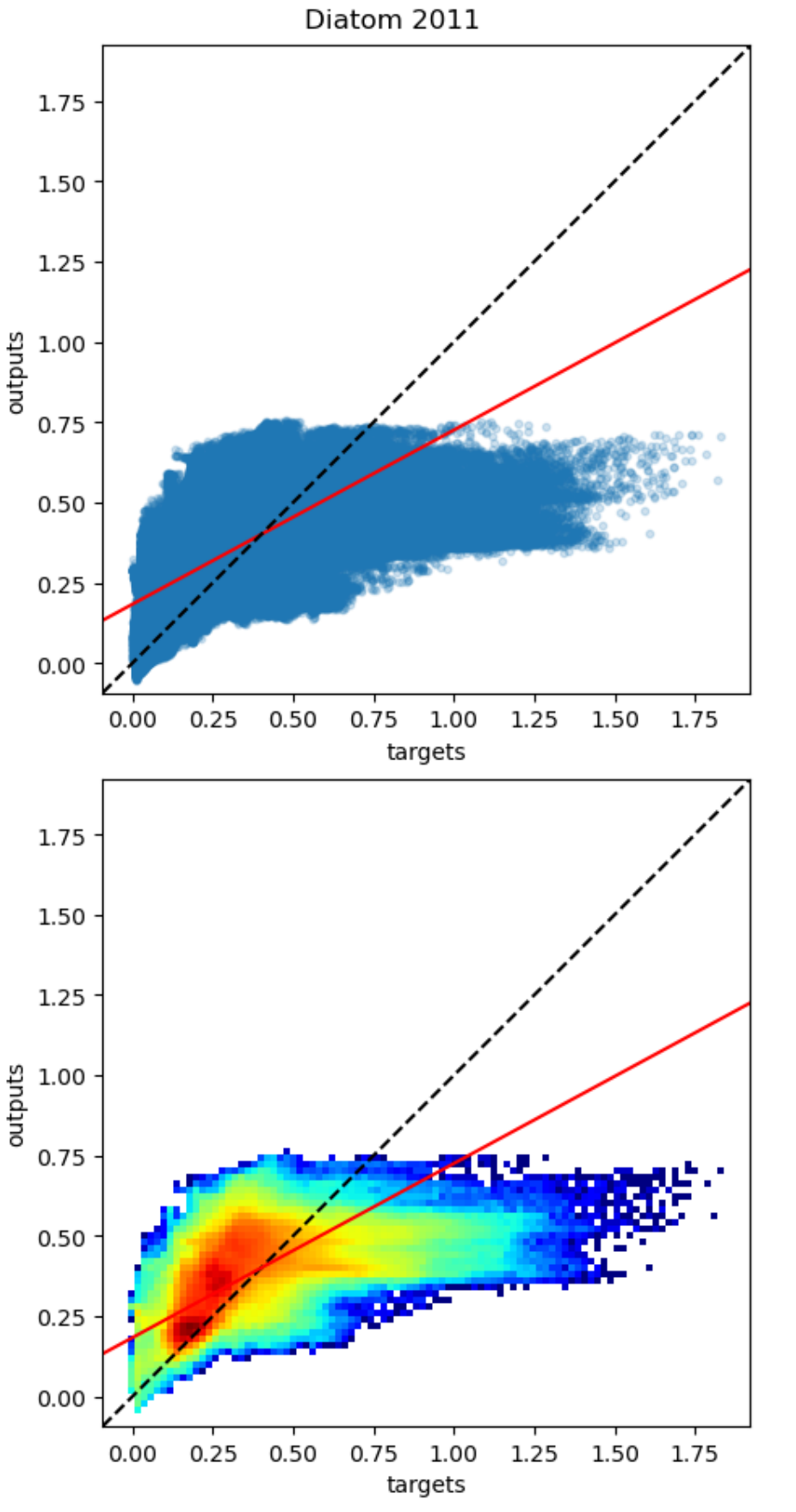
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.435
The correlation coefficient is: 0.605
 The mean square error is: 0.02475
```


Diatom 2009
outputs
targets
0.0
0.5
1.0
1.5
2.0
2.5
outputs
targets
12
10^0
10^1
10^2
10^3
10^4

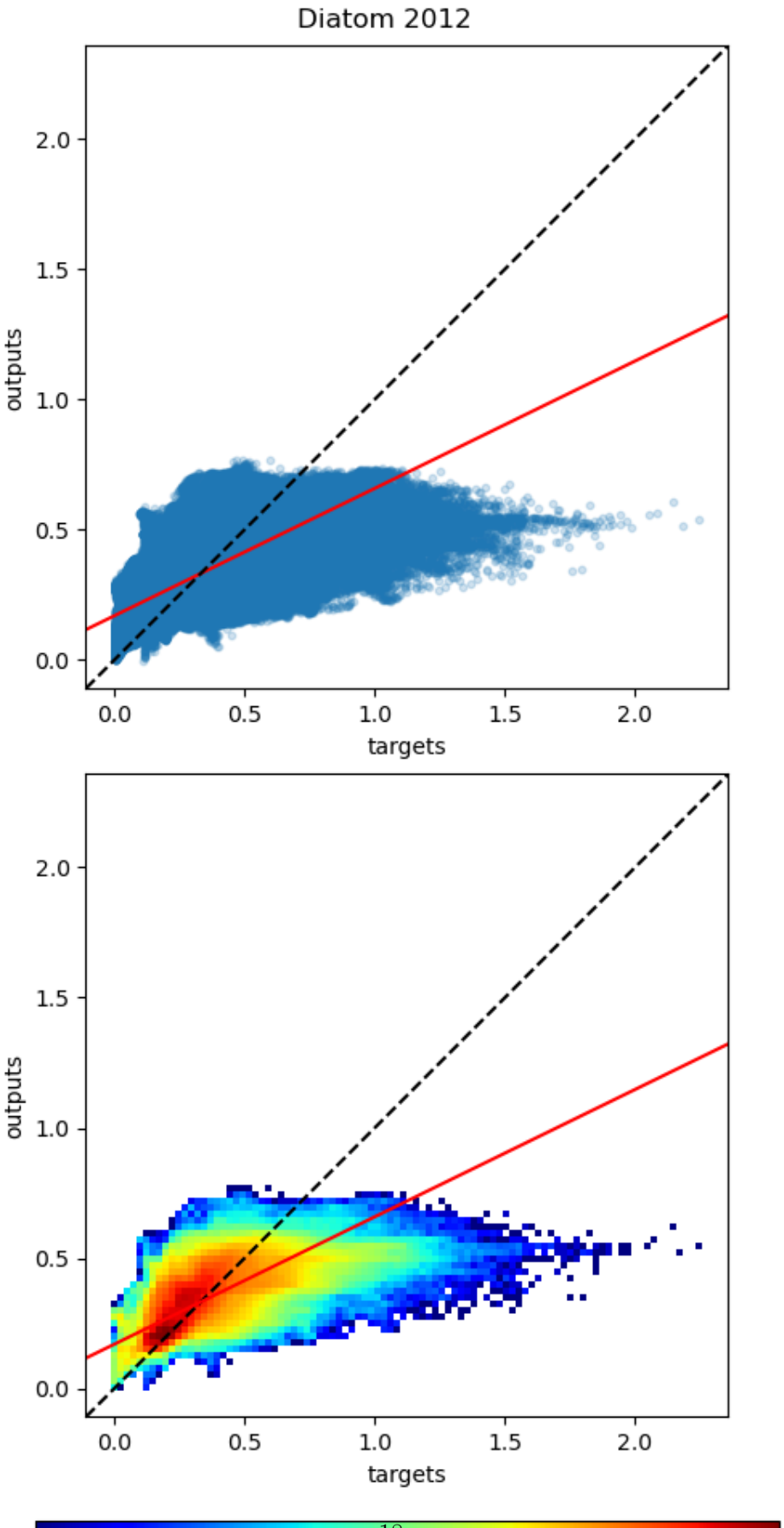
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.485
The correlation coefficient is: 0.564
 The mean square error is: 0.01499
```


Diatom 2010
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
14
10^0
10^1
10^2
10^3
10^4

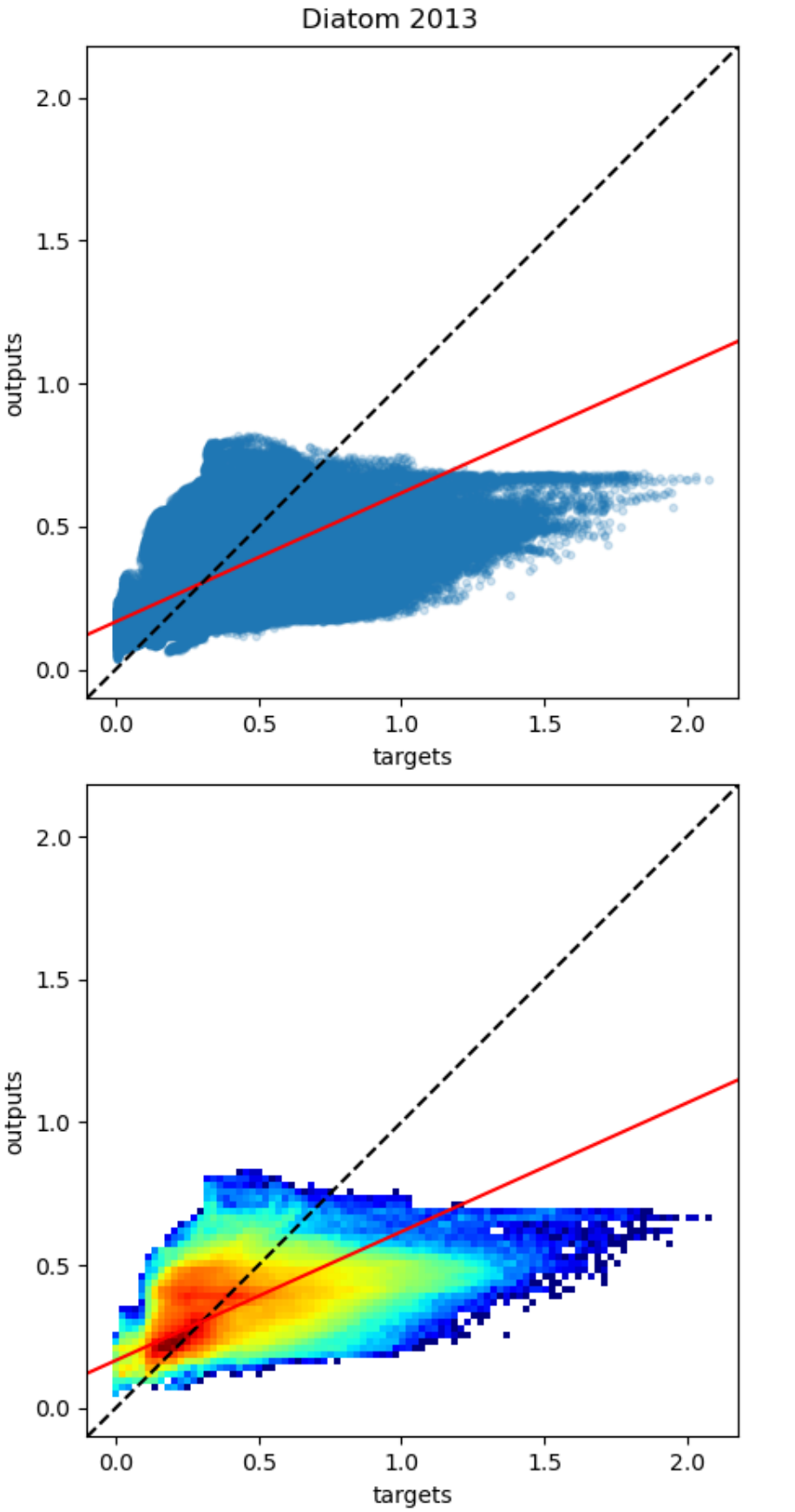
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.542
The correlation coefficient is: 0.612
 The mean square error is: 0.01803
```


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
10^4

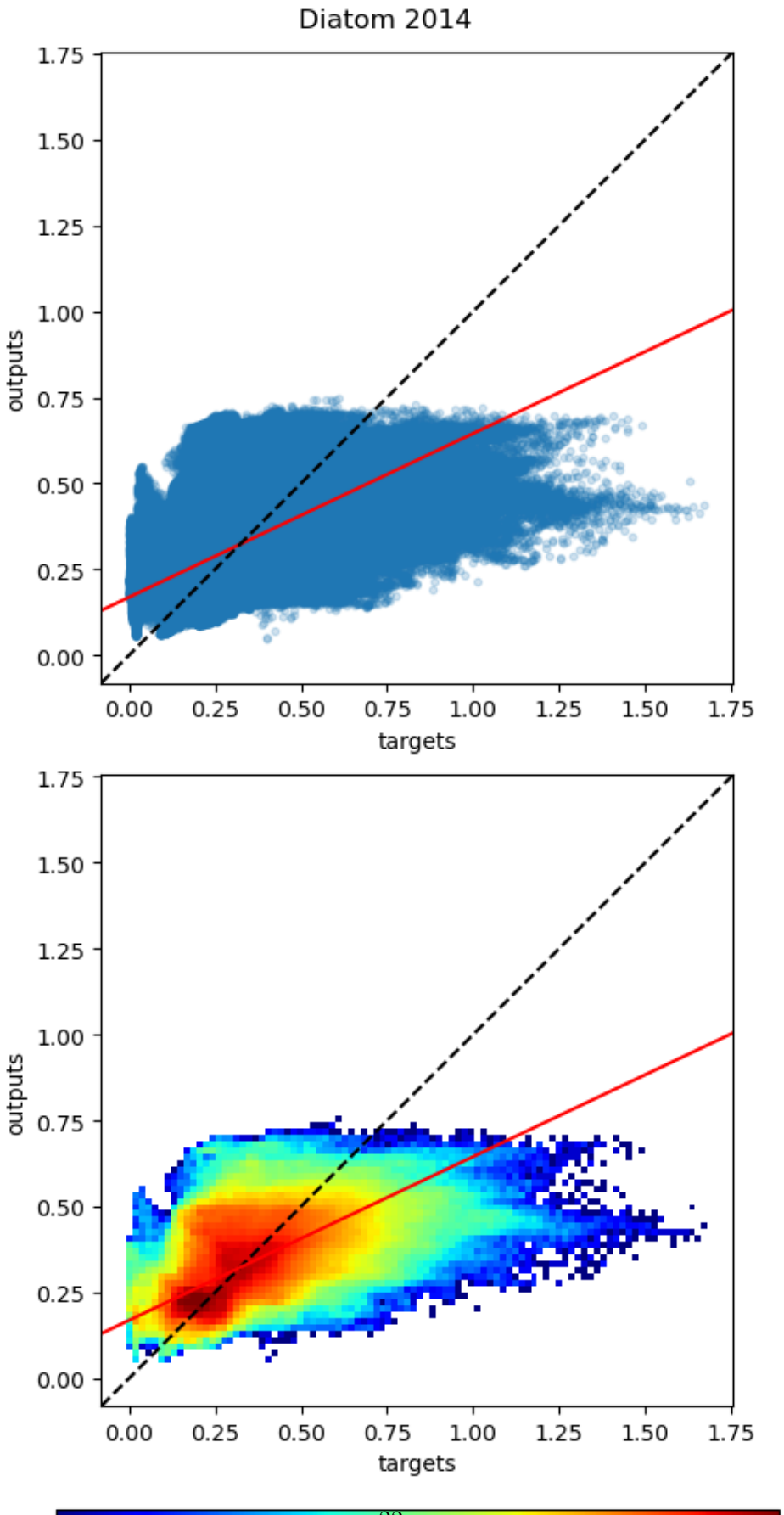
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3532404
The slope of the best fitting line is  0.489
The correlation coefficient is: 0.684
 The mean square error is: 0.01343
```


Diatom 2012
outputs
targets
0.0
0.5
1.0
1.5
2.0
outputs
targets
18
10^0
10^1
10^2
10^3
10^4

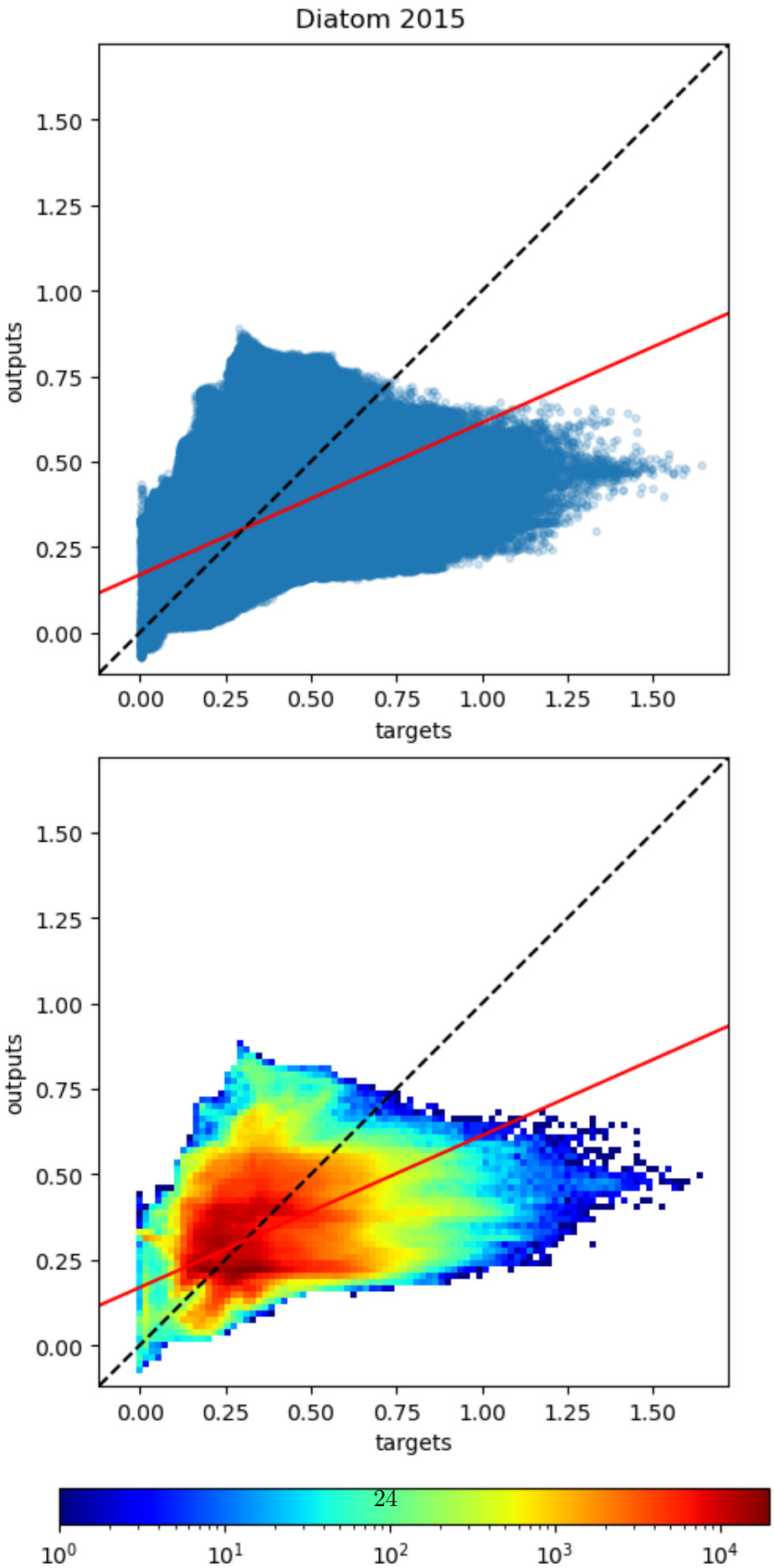
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.451
The correlation coefficient is: 0.569
 The mean square error is: 0.02148
```


Diatom 2013
outputs
targets
0.0
0.5
1.0
1.5
2.0
20
10^0
10^1
10^2
10^3
10^4

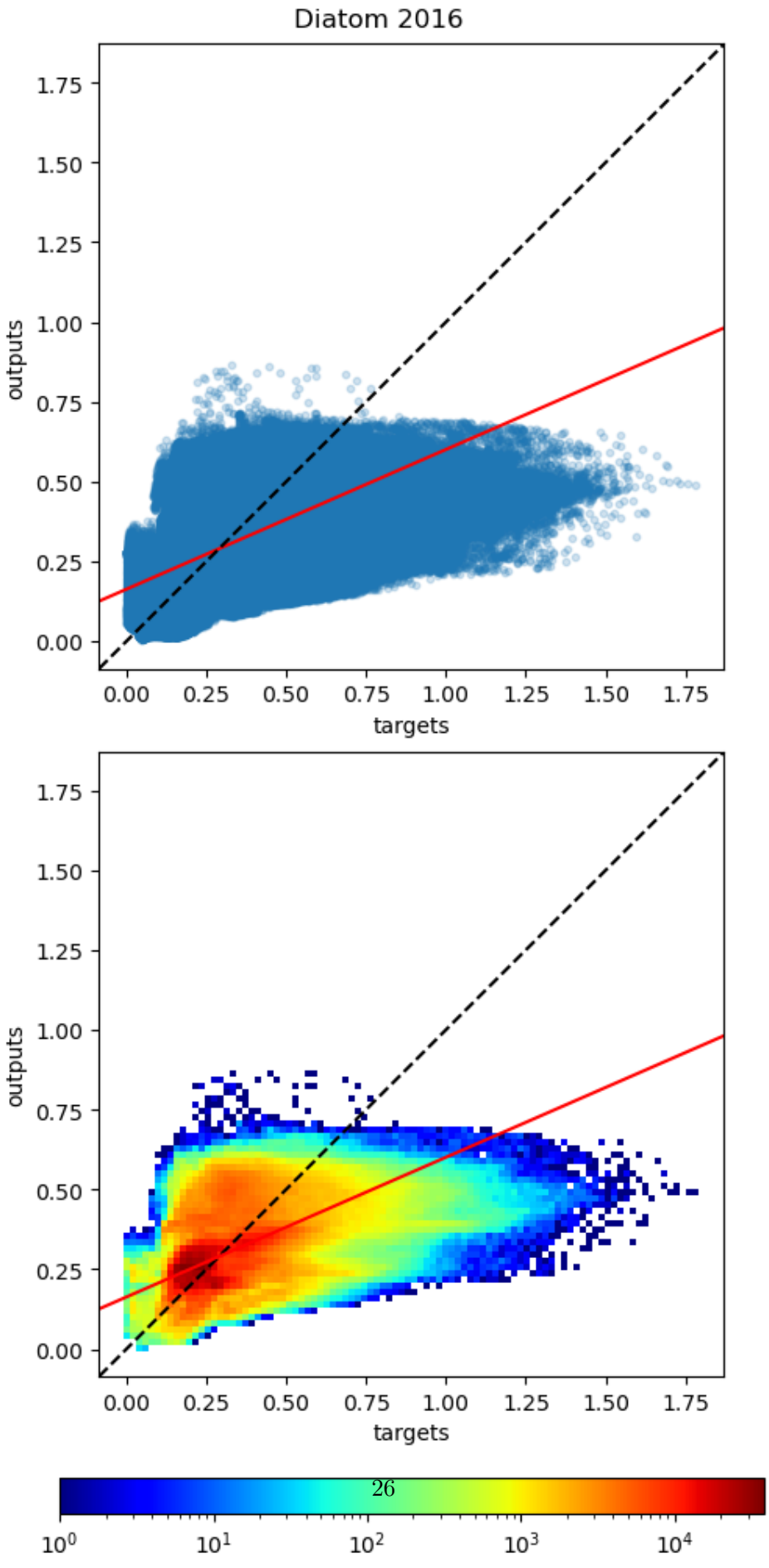
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.476
The correlation coefficient is: 0.571
 The mean square error is: 0.01449
```


Diatom 2014
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
22
10^0
10^1
10^2
10^3
10^4

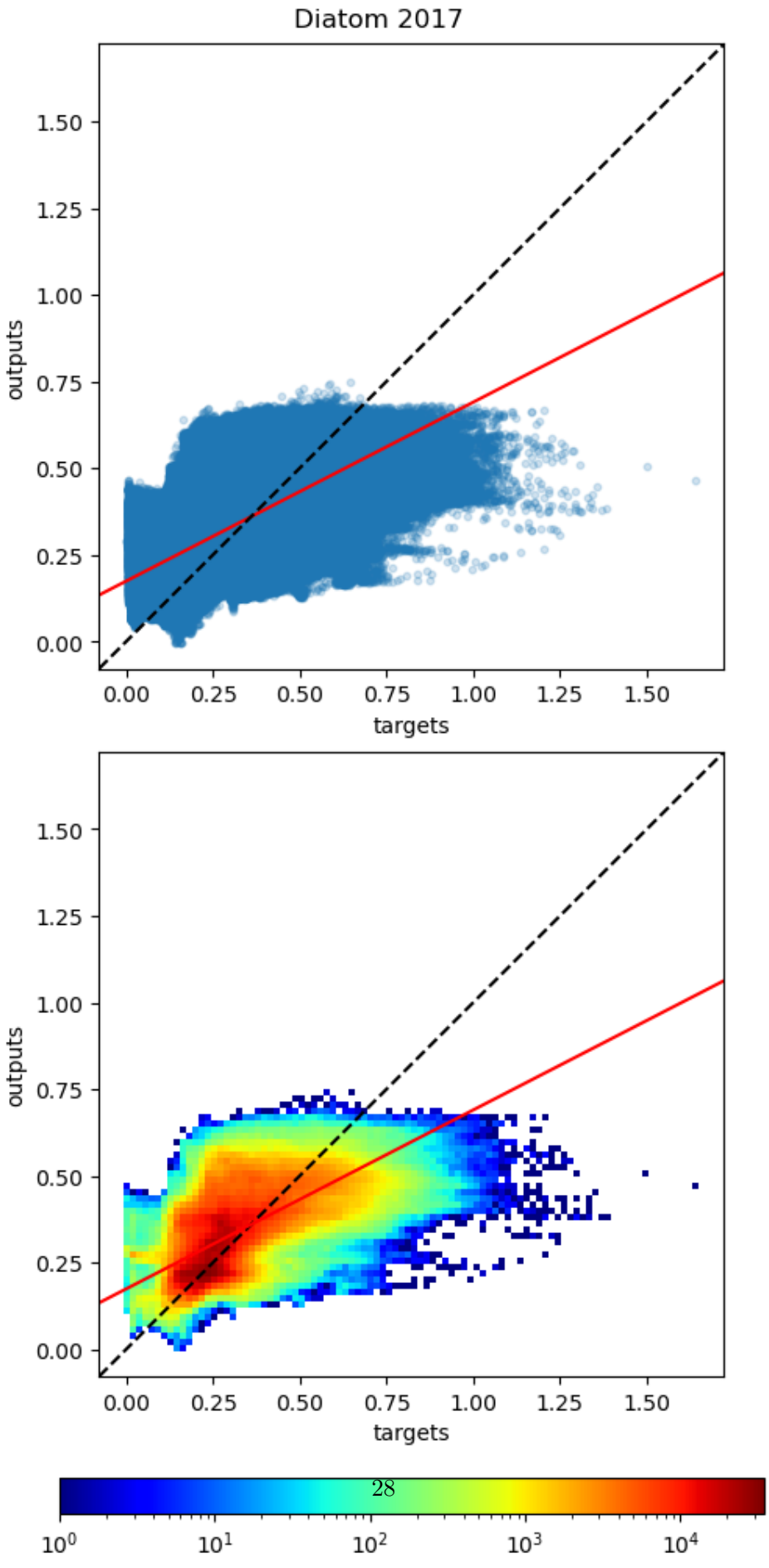
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.445
The correlation coefficient is: 0.267
 The mean square error is: 0.02705
```


Diatom 2015
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
24
10^0
10^1
10^2
10^3
10^4

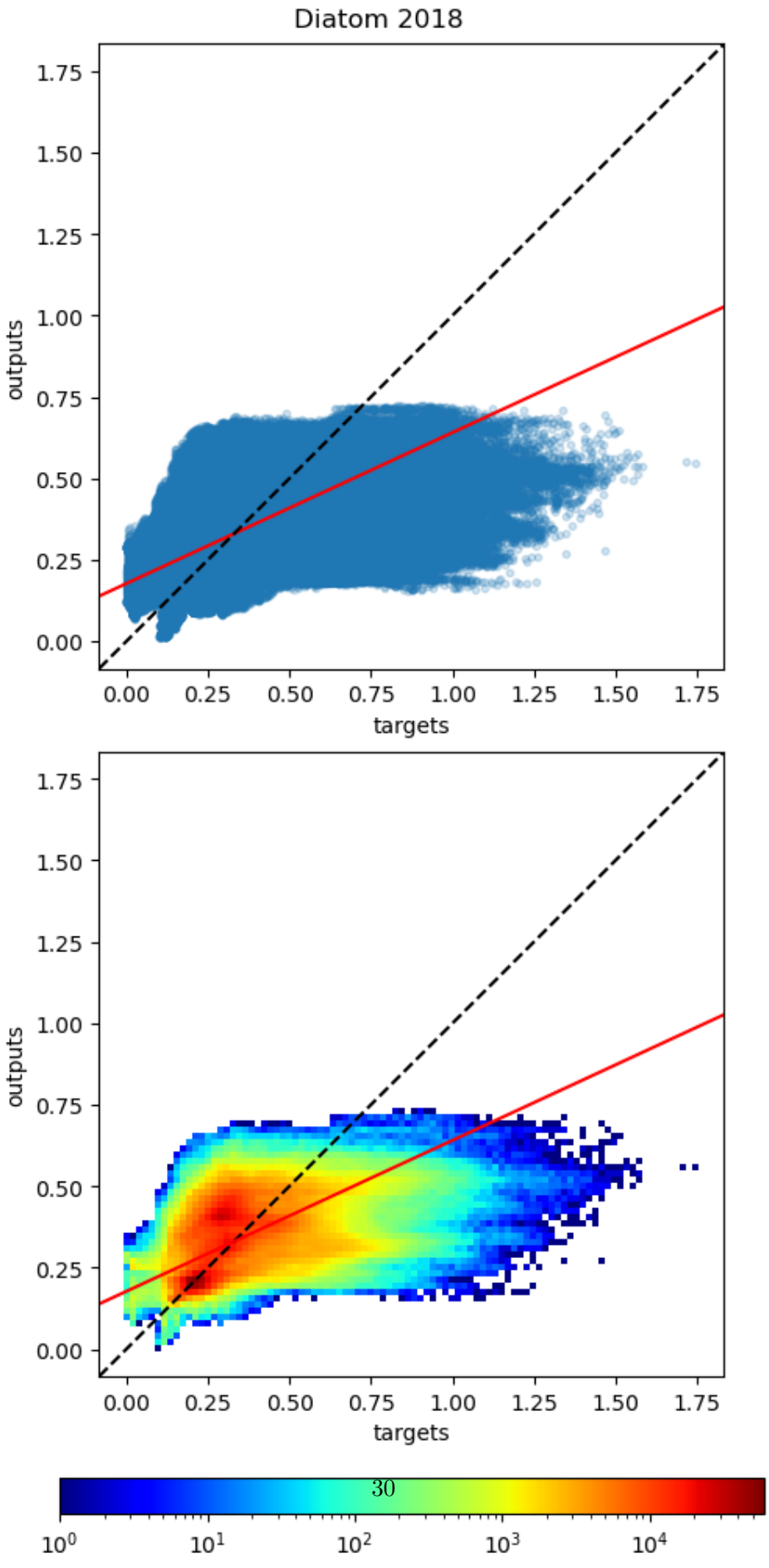
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3532404
The slope of the best fitting line is  0.438
The correlation coefficient is: 0.456
 The mean square error is: 0.02447
```


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
10^3
10^4

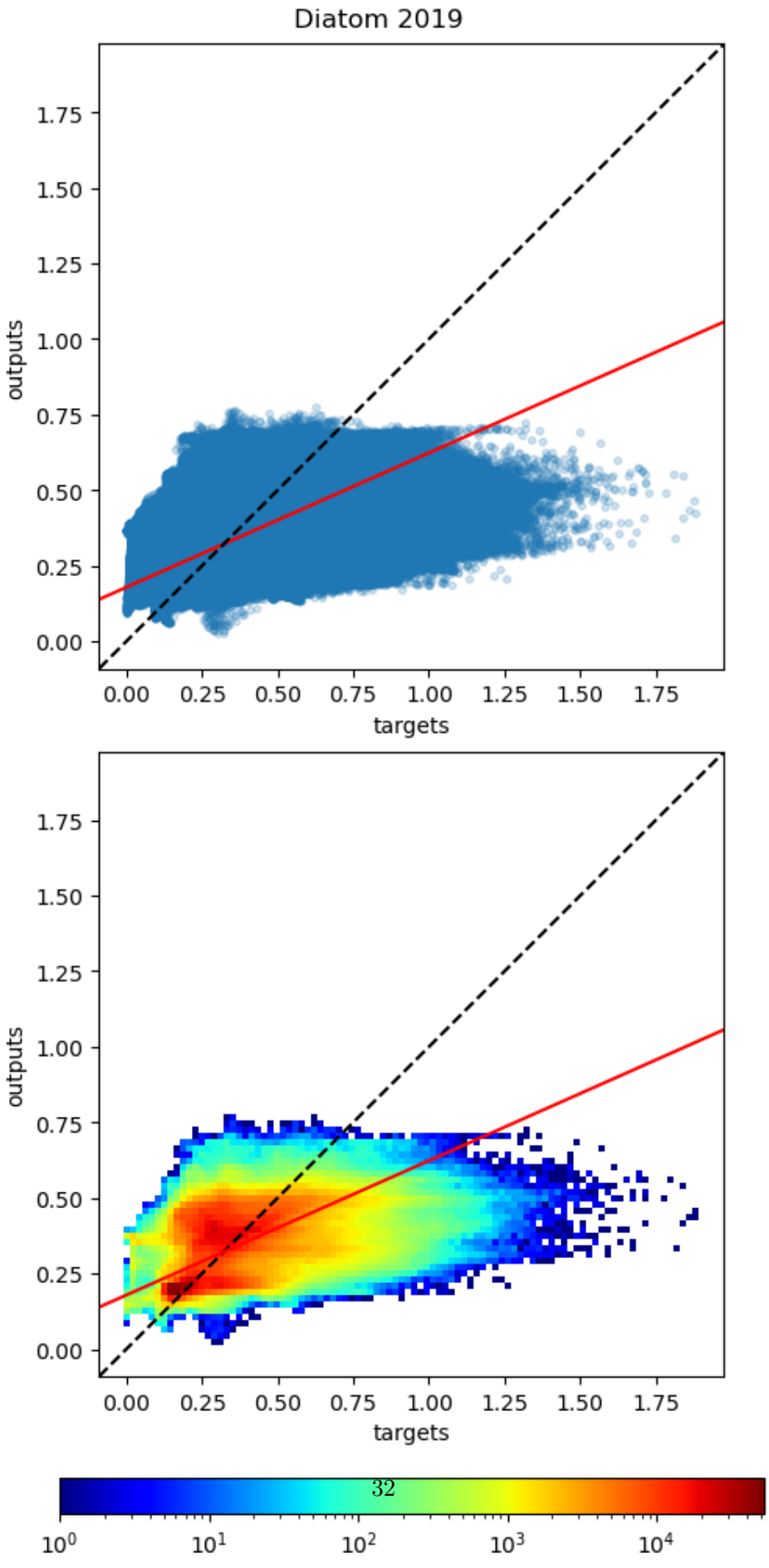
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.516
The correlation coefficient is: 0.62
 The mean square error is: 0.01238
```


Diatom 2017
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
28
10^0
10^1
10^2
10^3
10^4

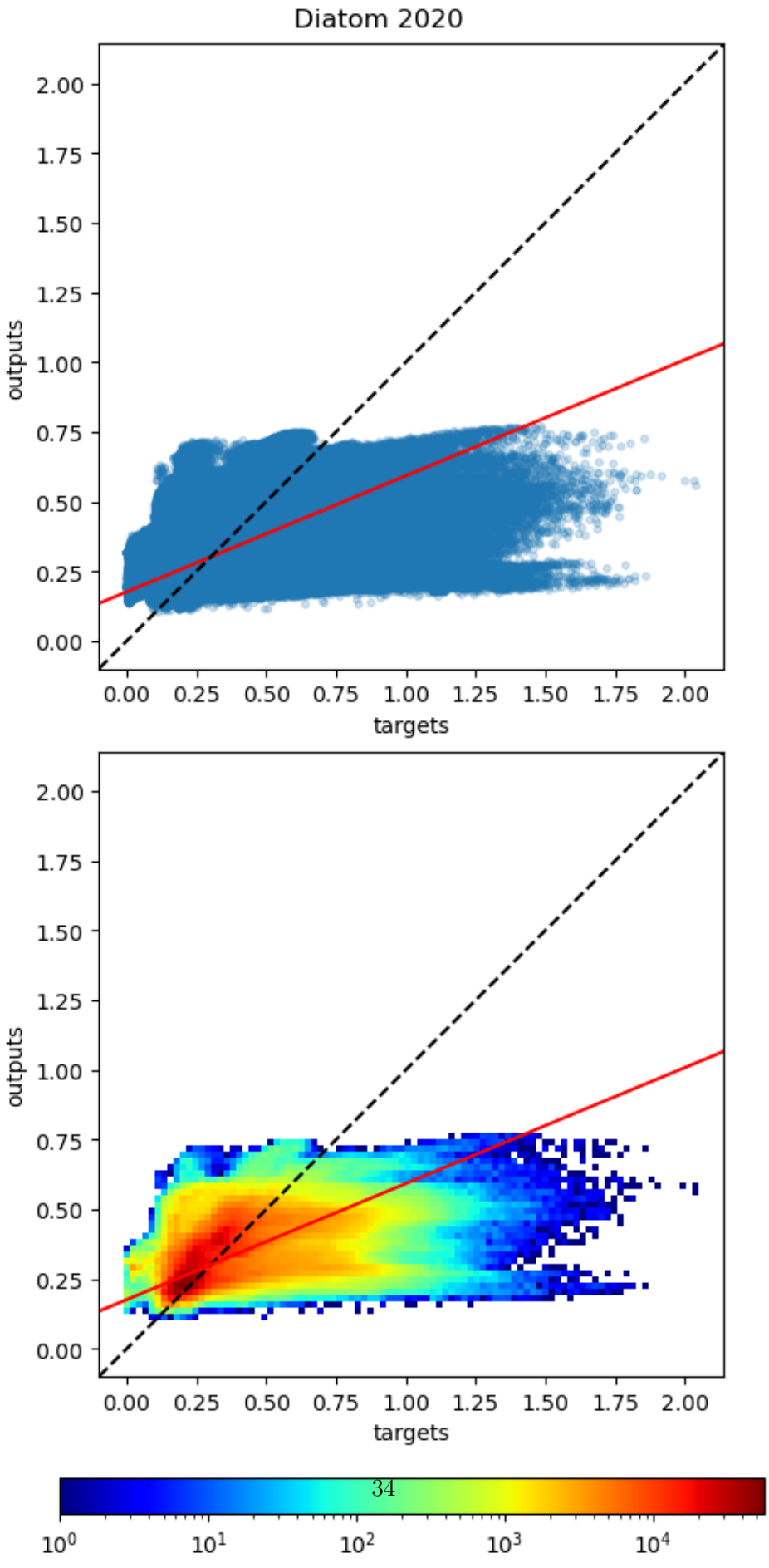
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.463
The correlation coefficient is: 0.429
 The mean square error is: 0.02133
```


Diatom 2018
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
outputs
targets
30
10^0
10^1
10^2
10^3
10^4

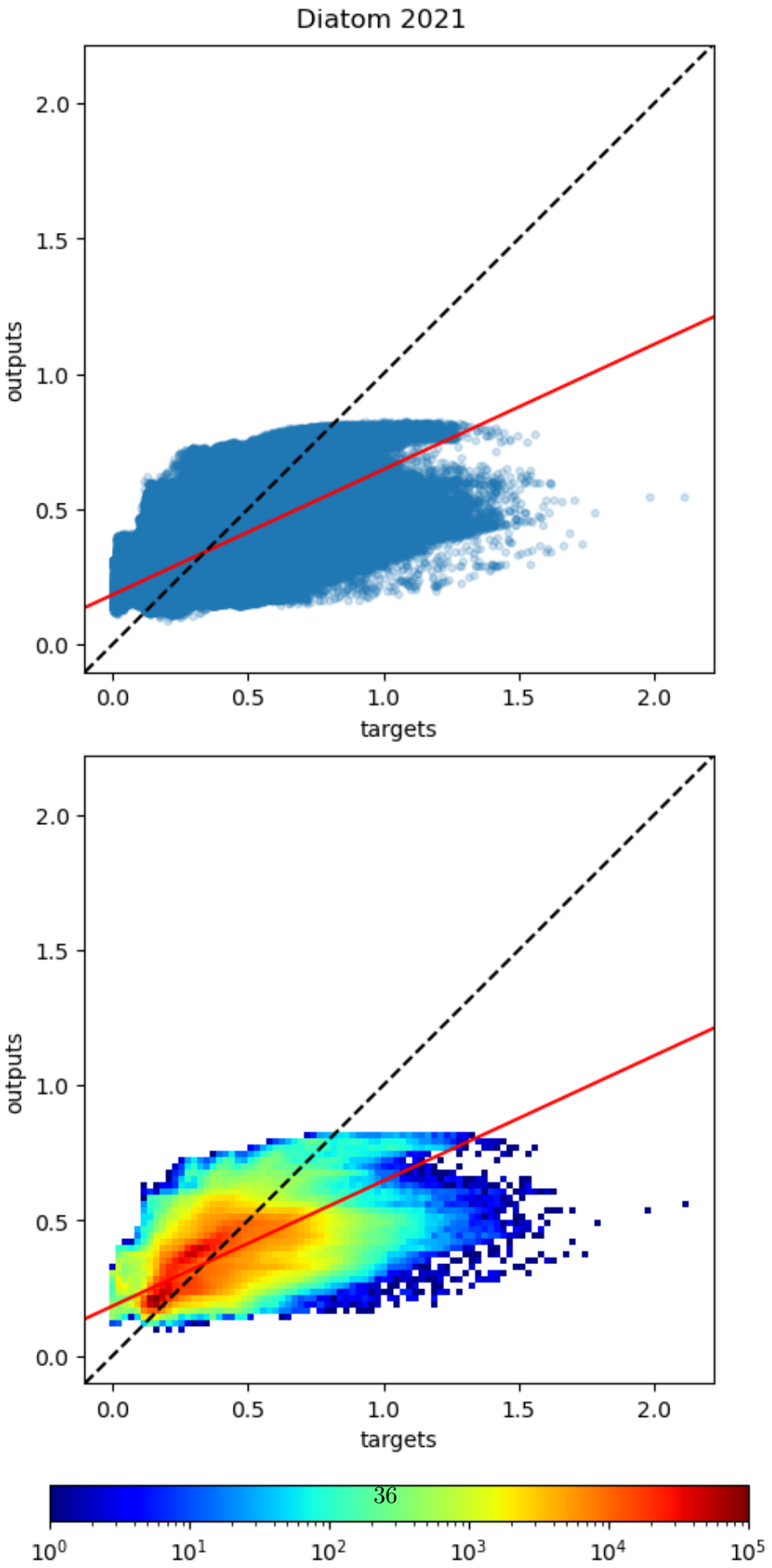
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.444
The correlation coefficient is: 0.45
 The mean square error is: 0.02527
```


Diatom 2019
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
outputs
targets
32
10^0
10^1
10^2
10^3
10^4

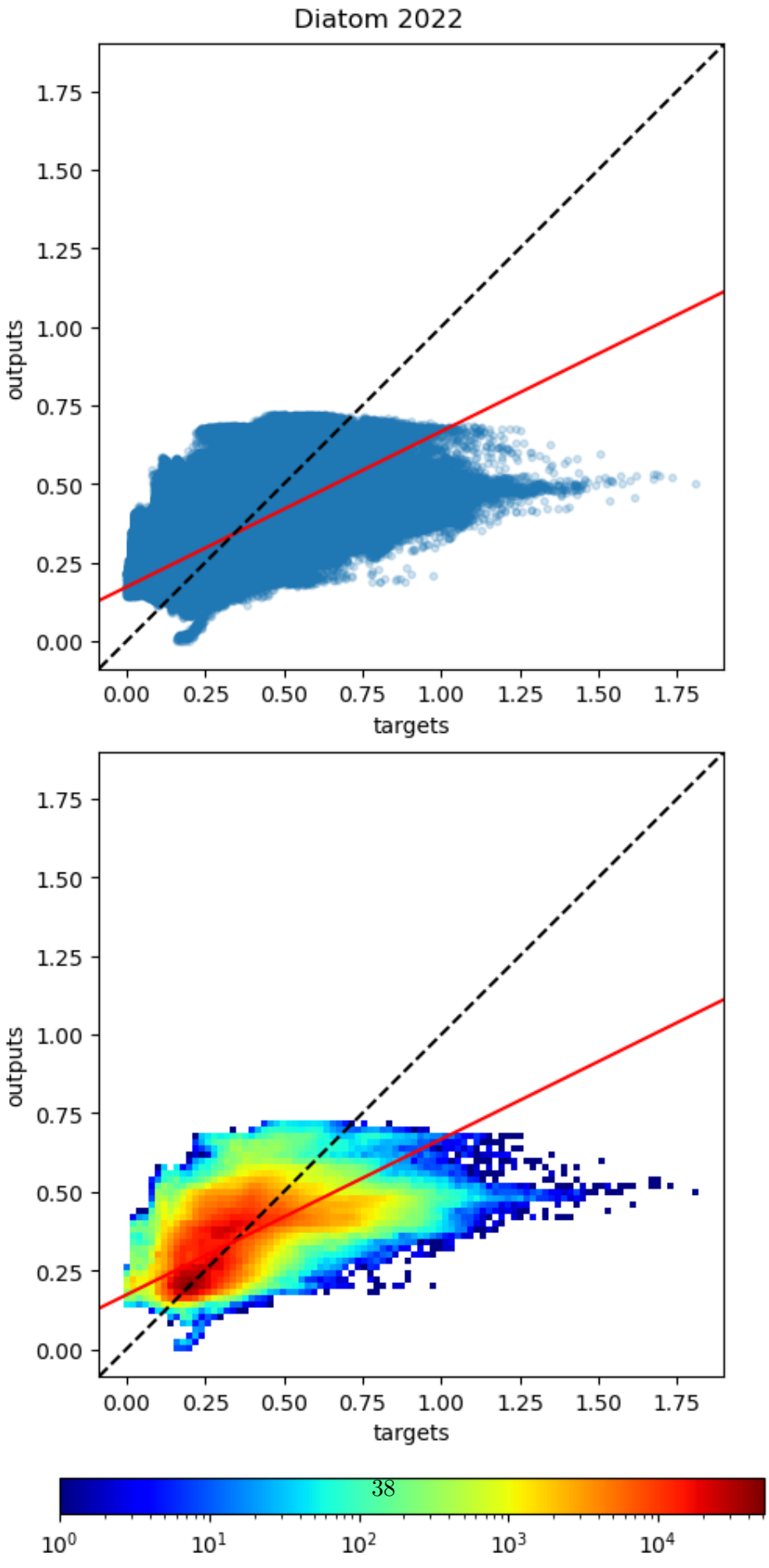
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3532404
The slope of the best fitting line is  0.416
The correlation coefficient is: 0.46
 The mean square error is: 0.03309
```


Diatom 2020
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
34
10^0
10^1
10^2
10^3
10^4

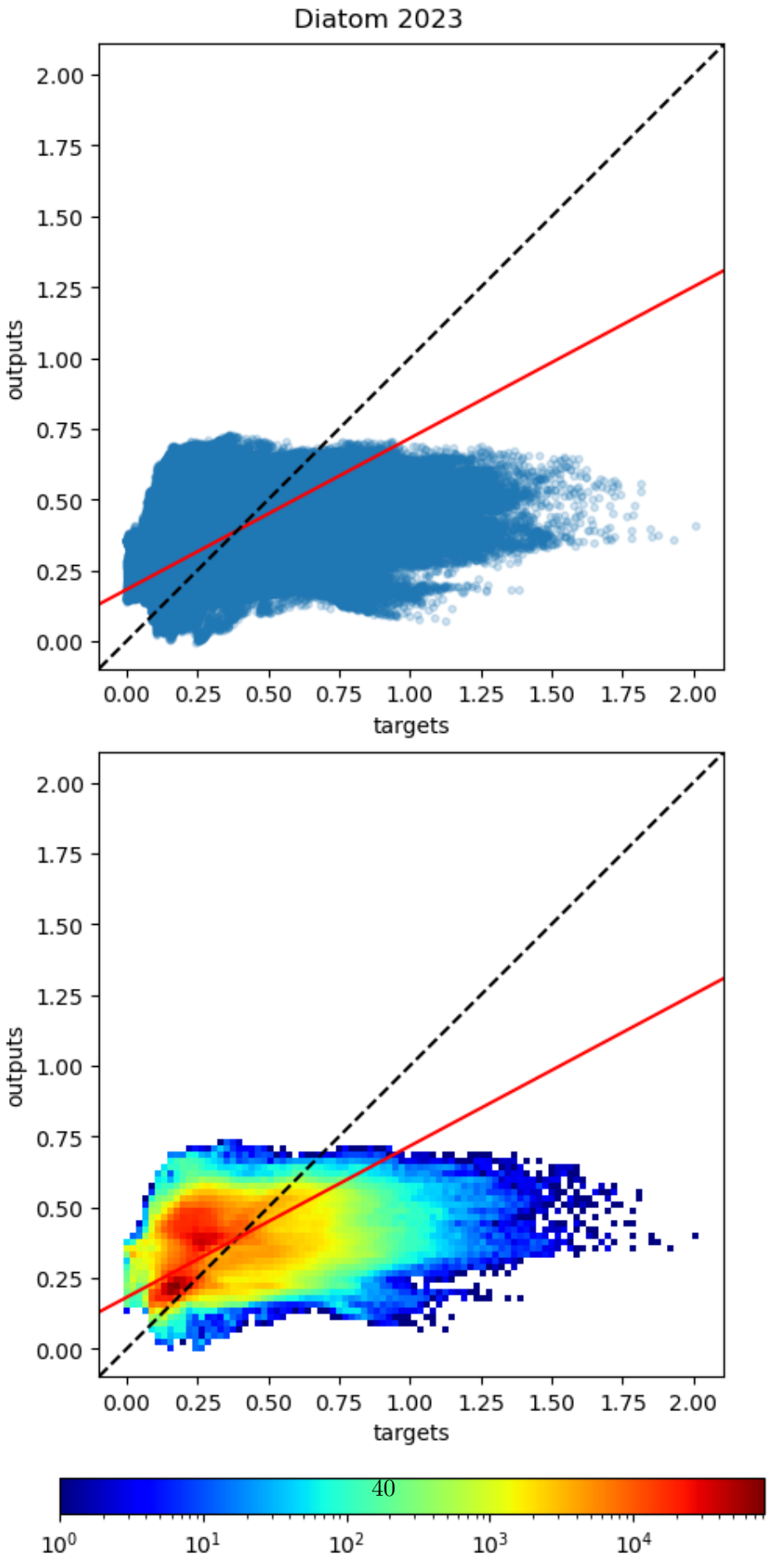
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.463
The correlation coefficient is: 0.637
 The mean square error is: 0.01838
```

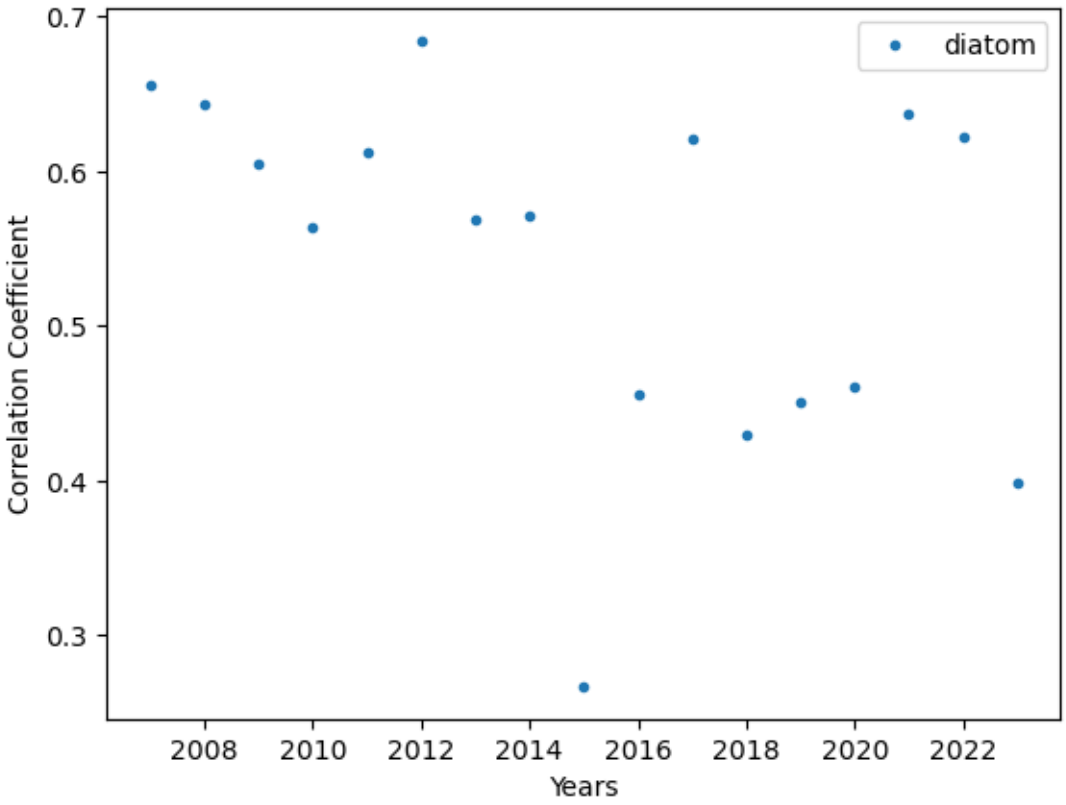

Diatom 2021
outputs
targets
0.0
0.5
1.0
1.5
2.0
outputs
targets
36
10^0
10^1
10^2
10^3
10^4
10^5

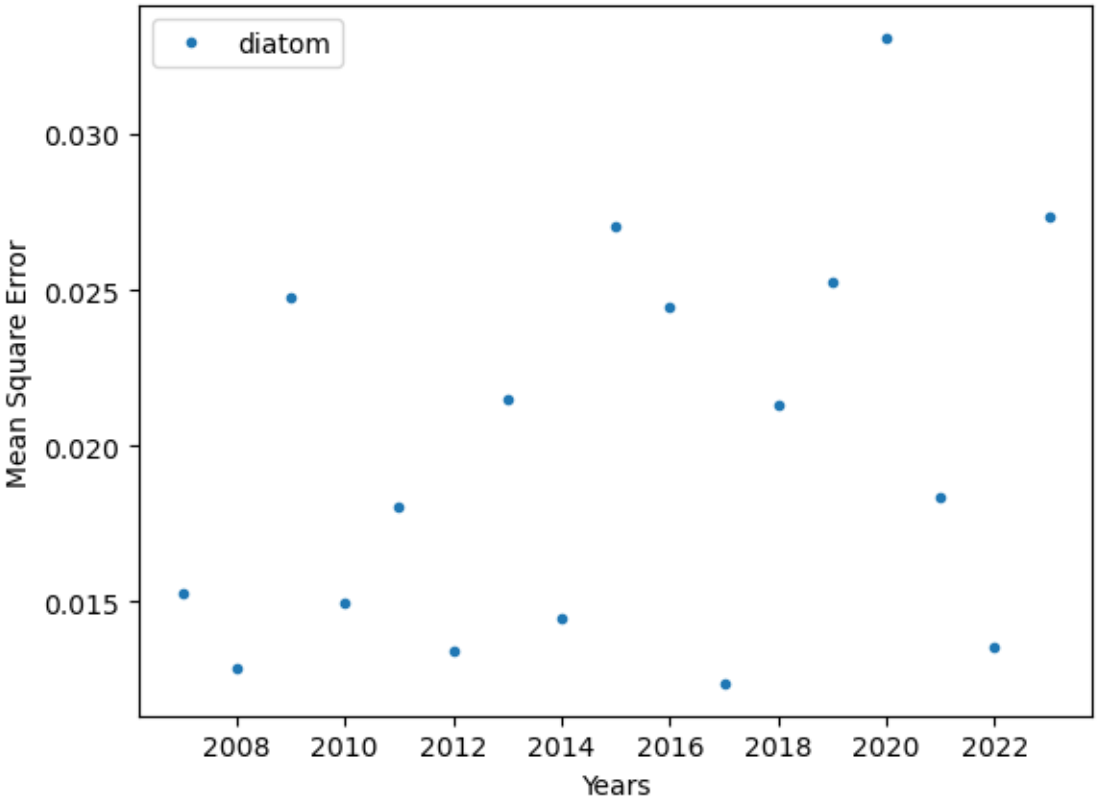
```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.495
The correlation coefficient is: 0.622
 The mean square error is: 0.01352
```

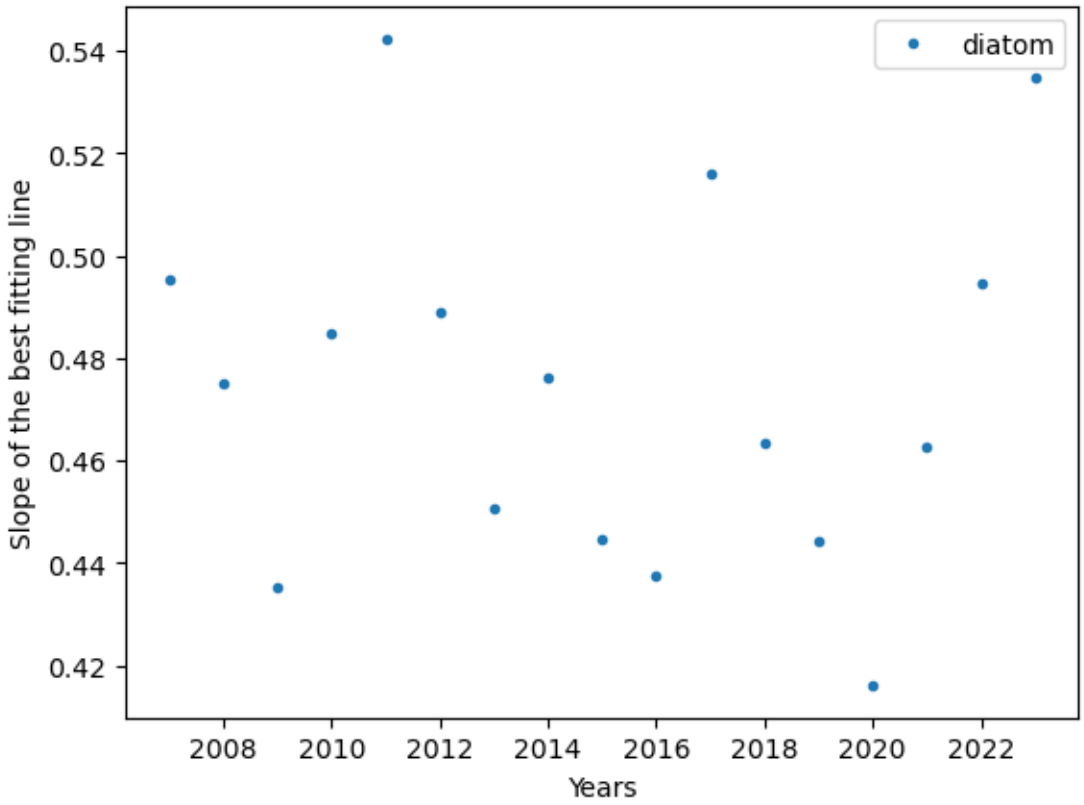

Diatom 2022
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
38
10^0
10^1
10^2
10^3
10^4

```
/tmp/ipykernel_20663/1980467486.py:4: RankWarning: Polyfit may be poorly
conditioned
  m, b = np.polyfit(targets, outputs, deg=1)

The amount of data points is 3485925
The slope of the best fitting line is  0.535
The correlation coefficient is: 0.398
 The mean square error is: 0.02733
```


Diatom 2023
outputs
targets
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
40
10^0
10^1
10^2
10^3
10^4


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


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

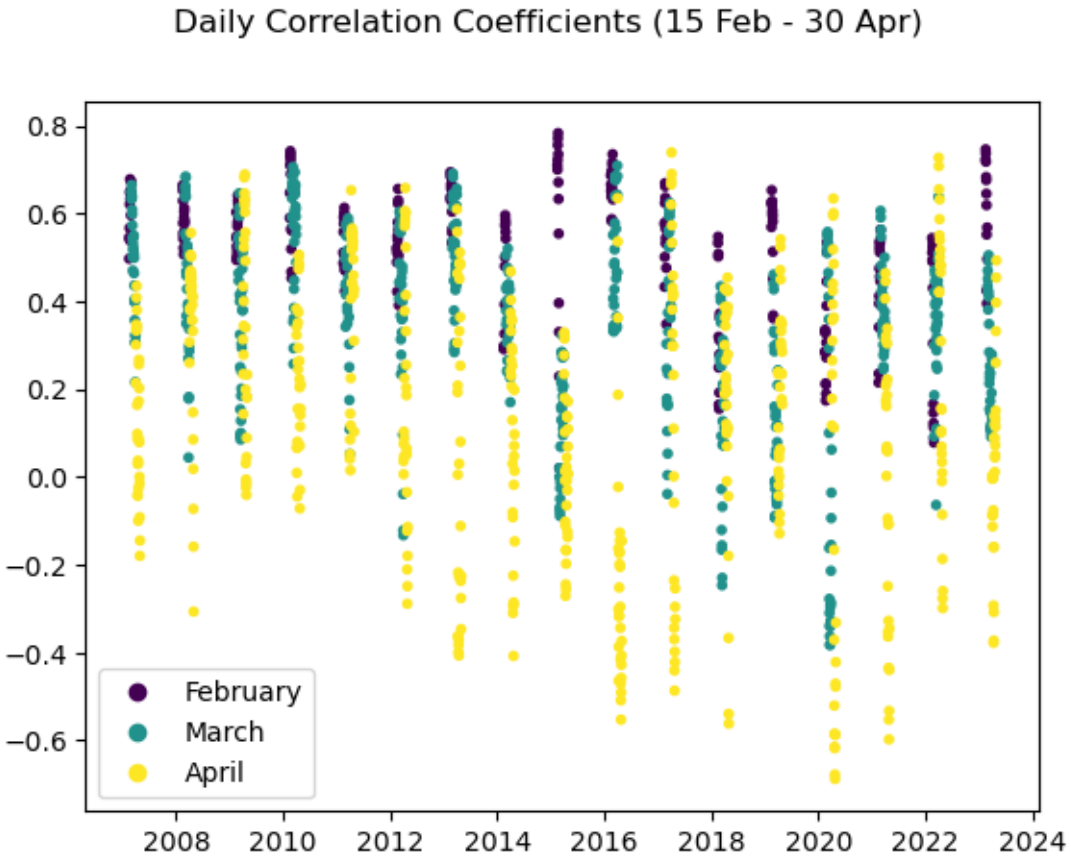


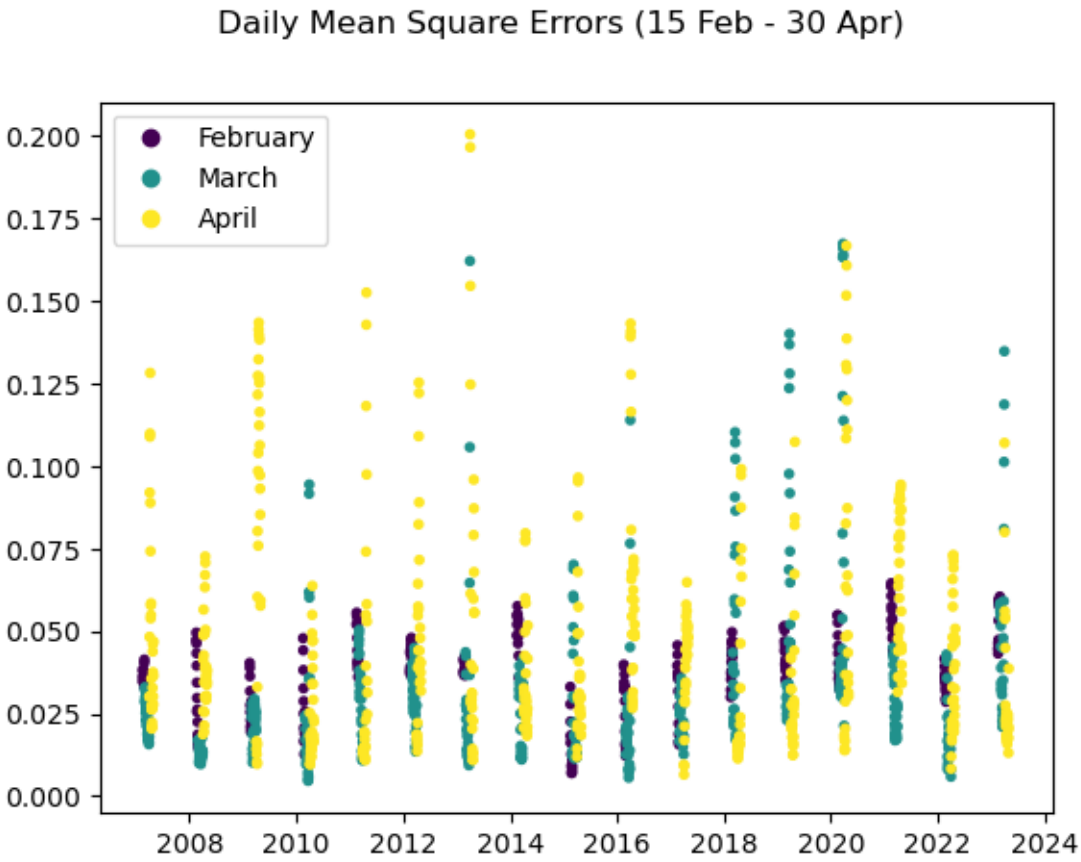
1.7 Other Years (Daily)

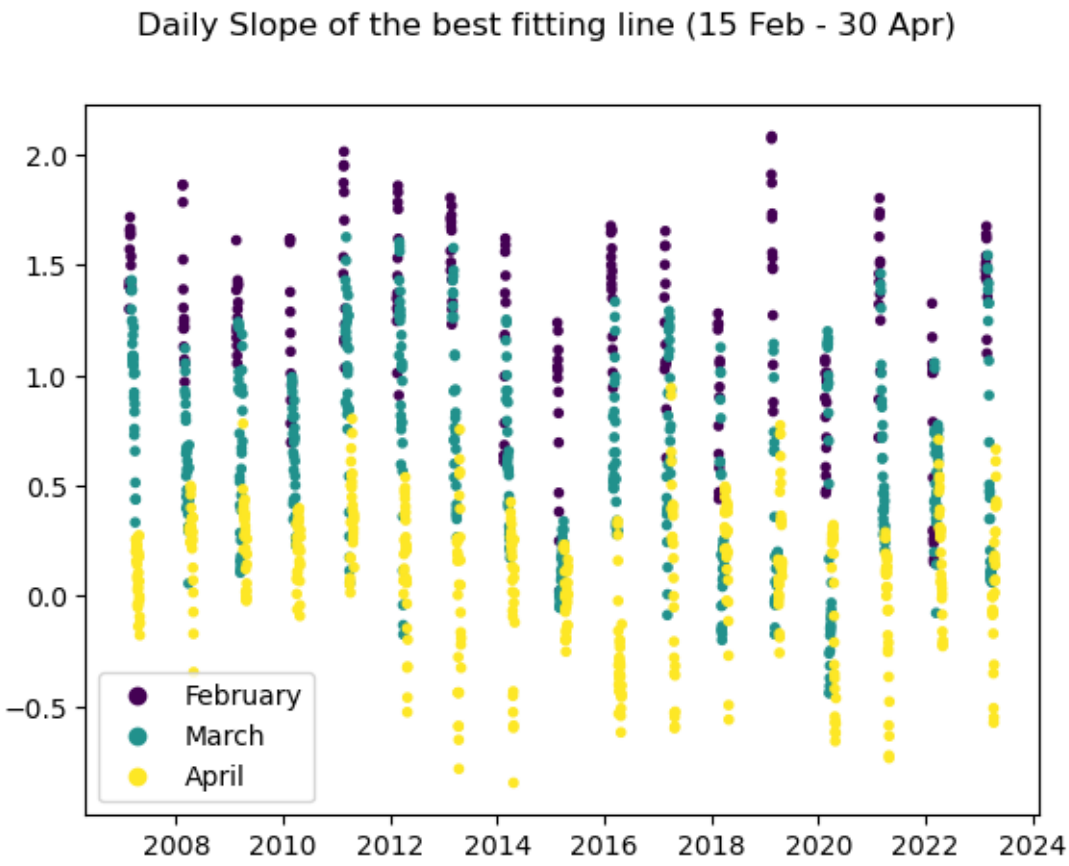
[ ]:
```
r_all2 = np.array([])
rms_all2 = np.array([])
slope_all2 = np.array([])

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%|          | 0/1279 [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

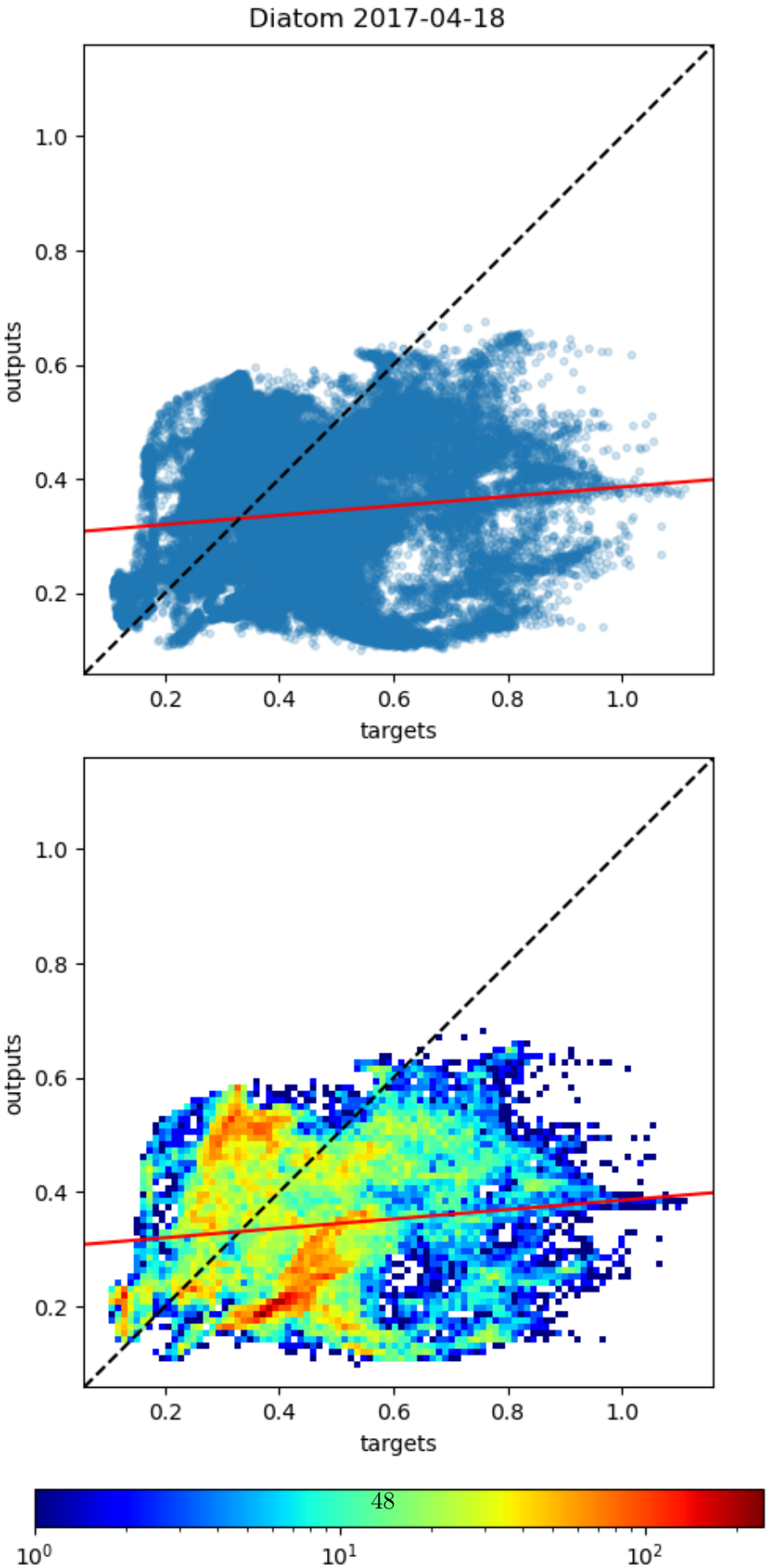


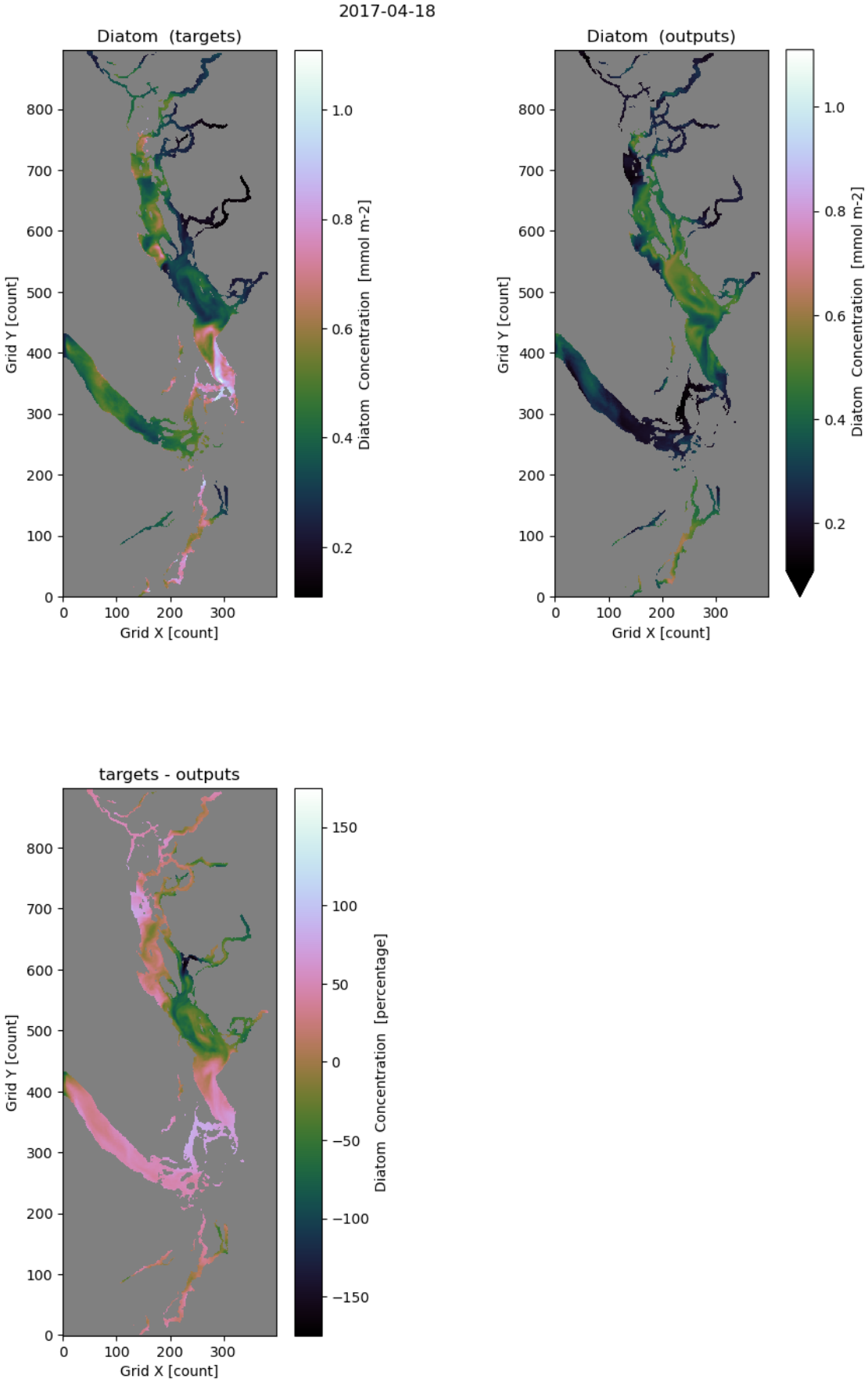
2 Daily Maps

[ ]:
```
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 ')
```

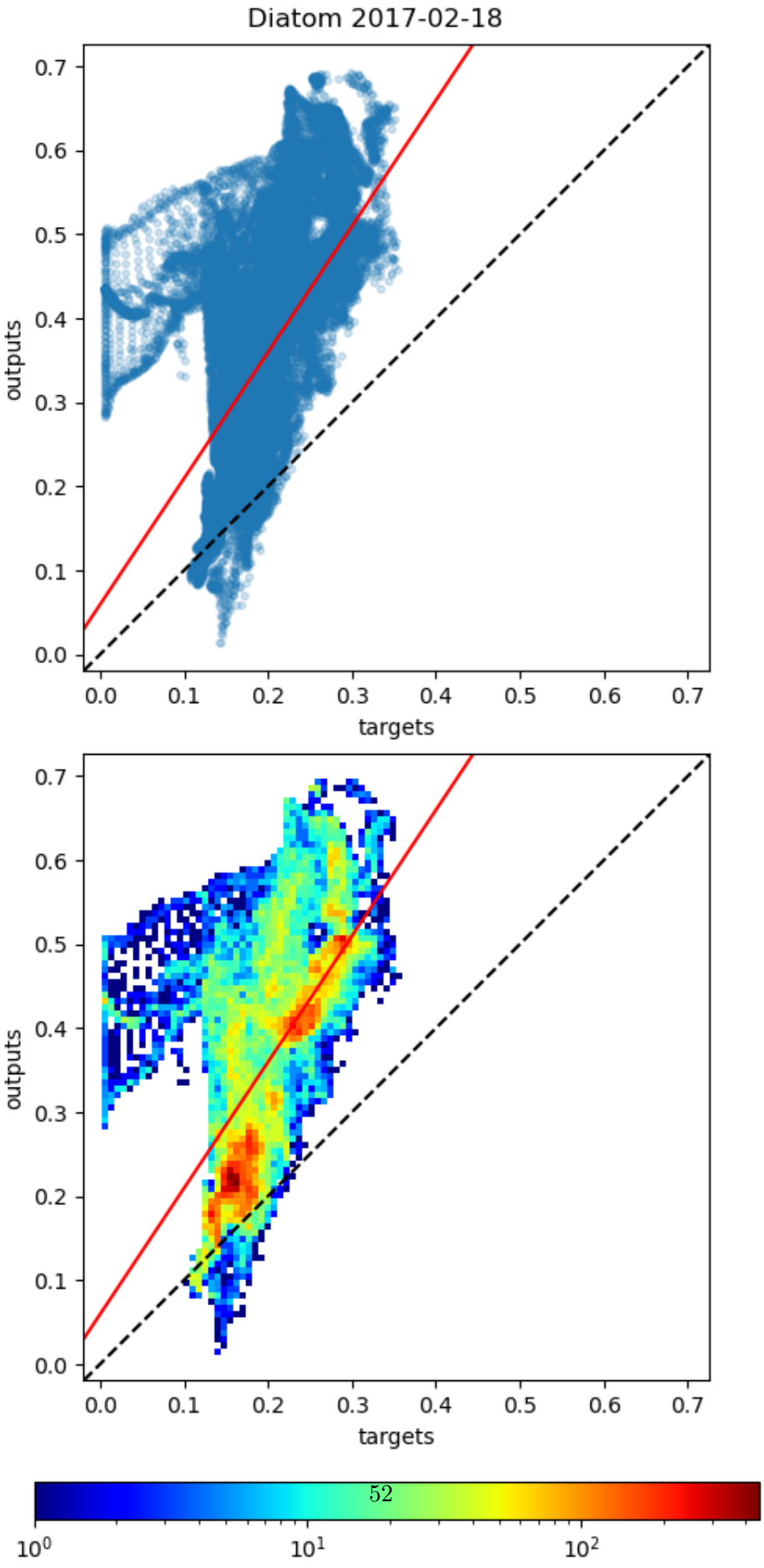
```
  0%|          | 0/10 [00:00<?, ?it/s]
The amount of data points is 46479
The slope of the best fitting line is  0.082
The correlation coefficient is: 0.11
```

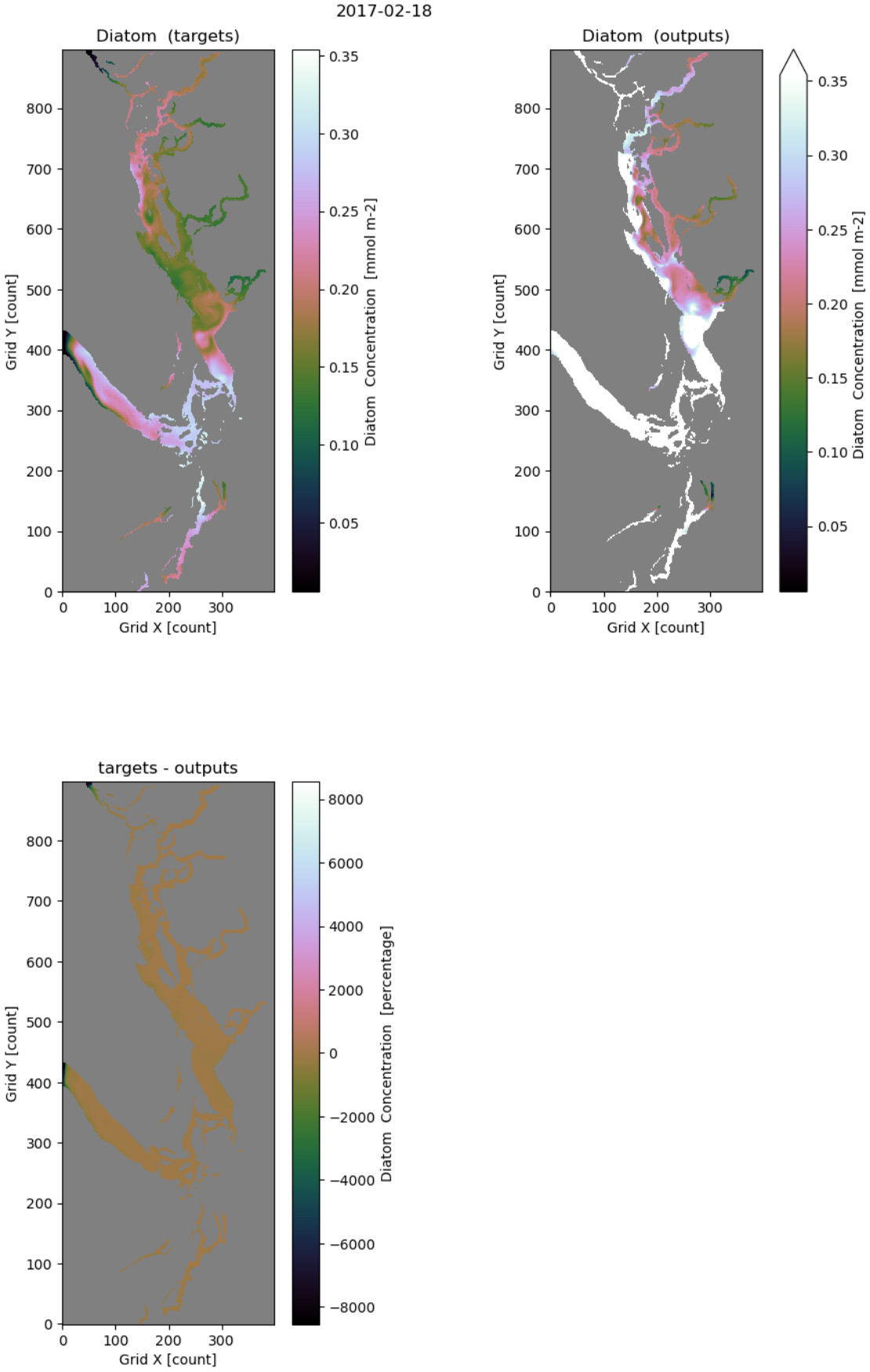
```
The mean square error is: 0.05047
```


Diatom 2017-04-18
outputs
targets
0.2
0.4
0.6
0.8
1.0
outputs
targets
48
10^0
10^1
10^2

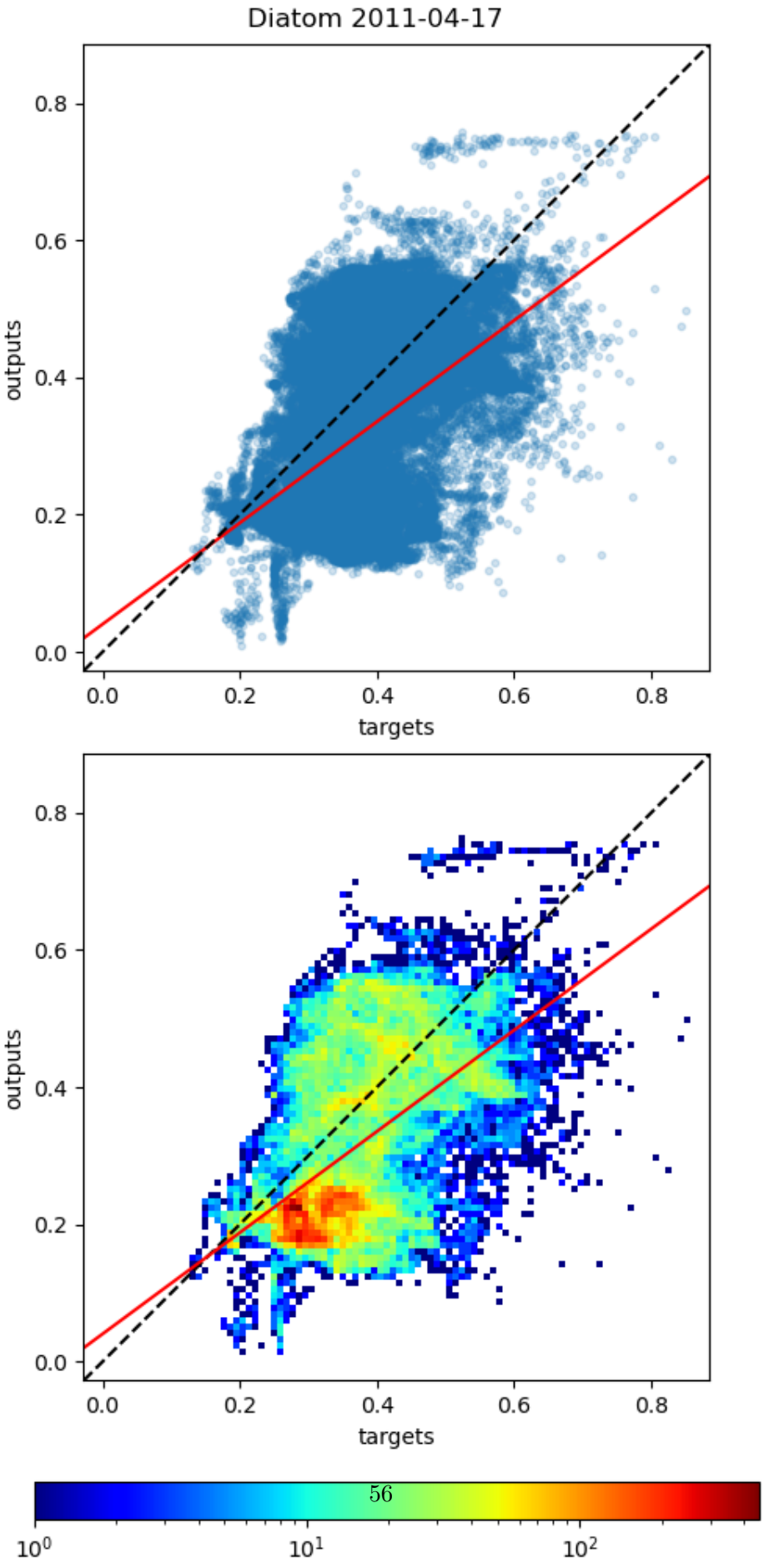

2017-04-18
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

```
The amount of data points is 46479
The slope of the best fitting line is  1.501
The correlation coefficient is: 0.624
 The mean square error is: 0.03707
```

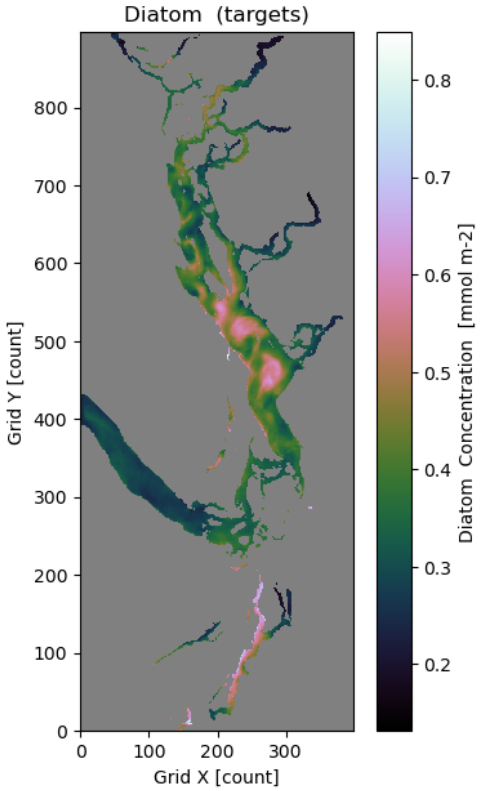

Diatom 2017-02-18
outputs
targets
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
52
10^0
10^1
10^2

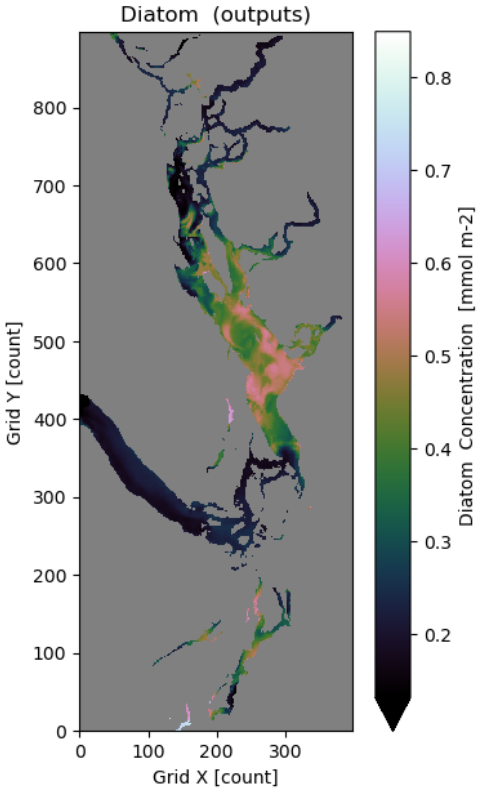

2017-02-18
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

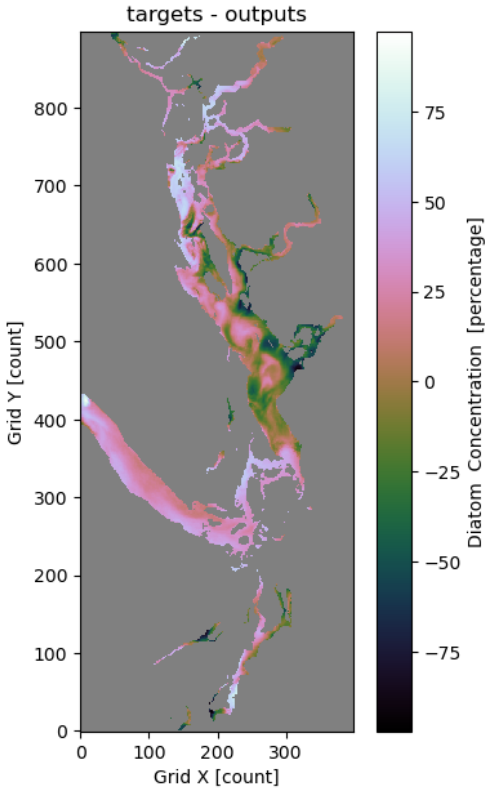
```
The amount of data points is 46479
The slope of the best fitting line is  0.738
The correlation coefficient is: 0.544
 The mean square error is: 0.01509
```


Diatom 2011-04-17
outputs
targets
0.0
0.2
0.4
0.6
0.8
outputs
targets
0.0
0.2
0.4
0.6
0.8
56
10^0
10^1
10^2

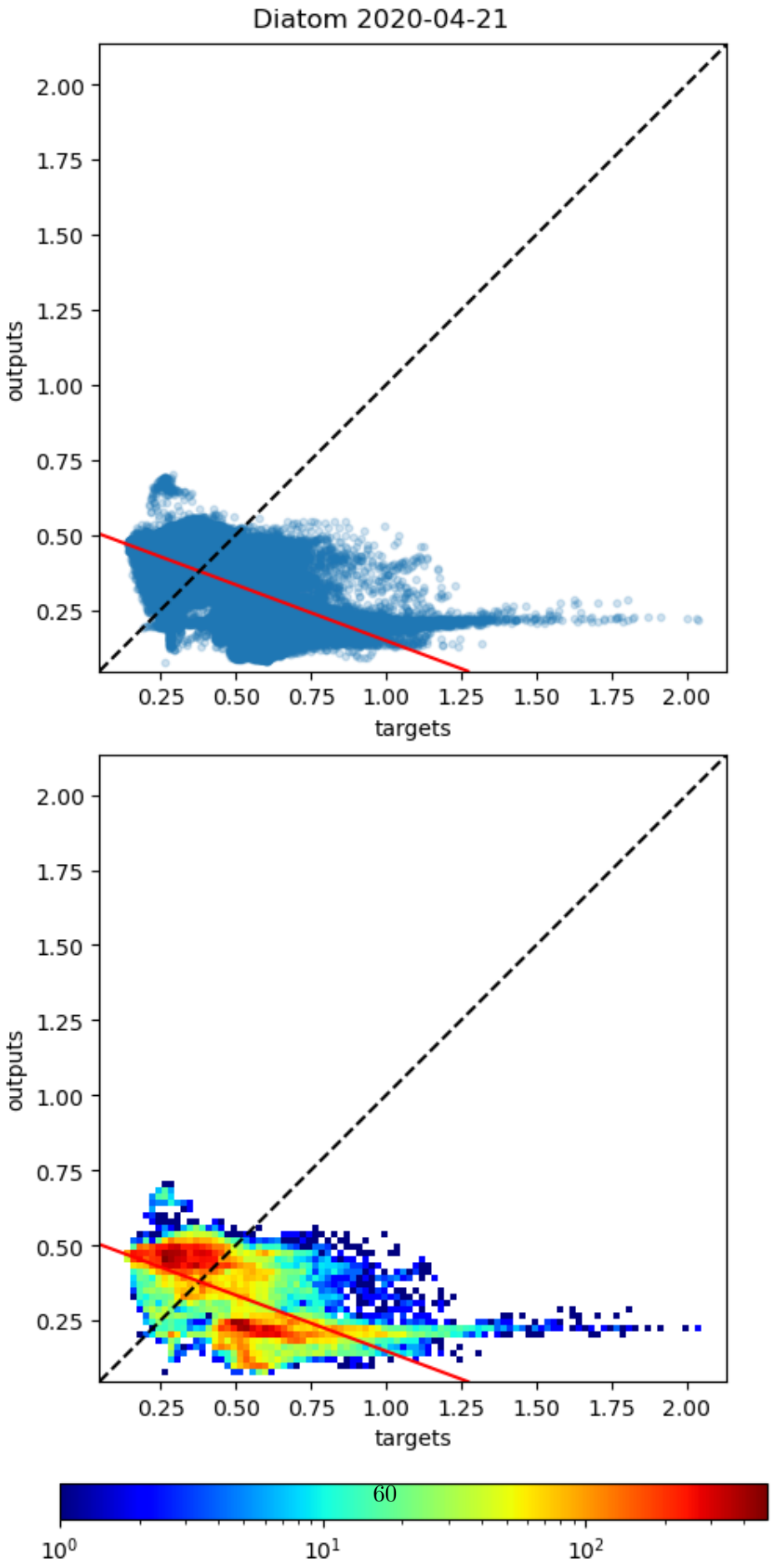
2011-04-17

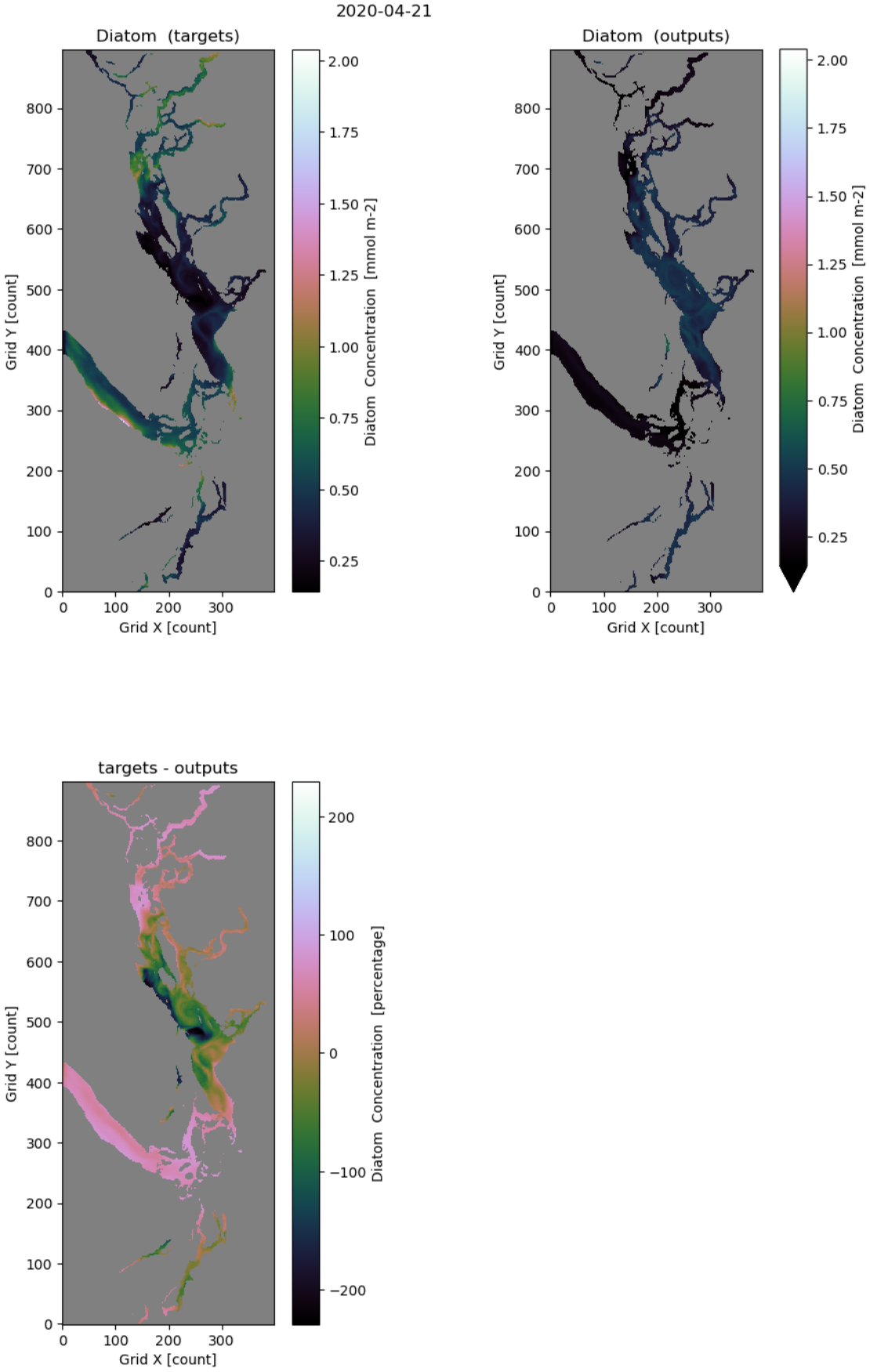

Diatom (targets)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
0
100
200
300
400
500
600
700
800
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Diatom (outputs)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
0
100
200
300
400
500
600
700
800
0.2
0.3
0.4
0.5
0.6
0.7
0.8

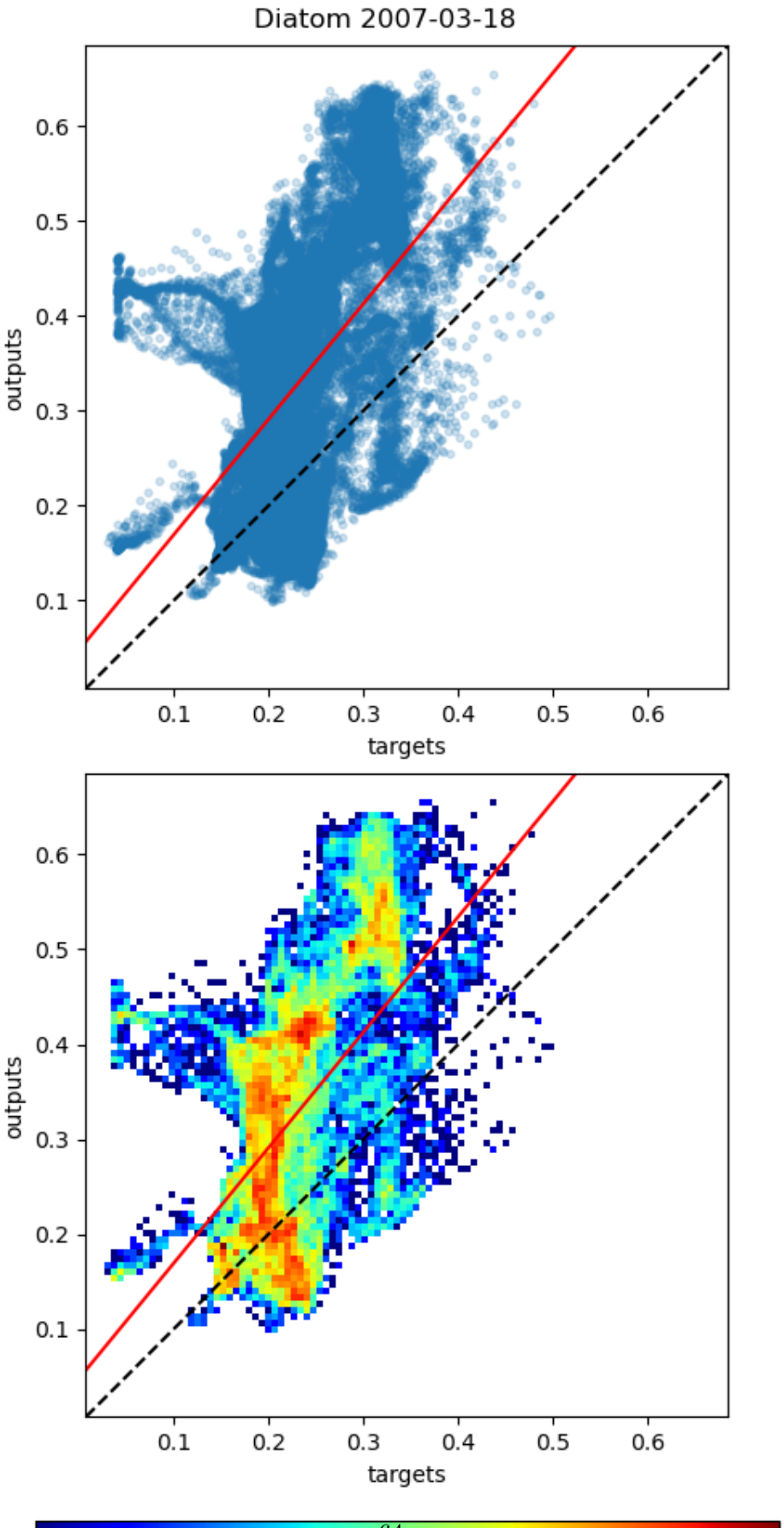

targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [percentage]
0
100
200
300
400
500
600
700
800
−75
−50
−25
0
25
50
75

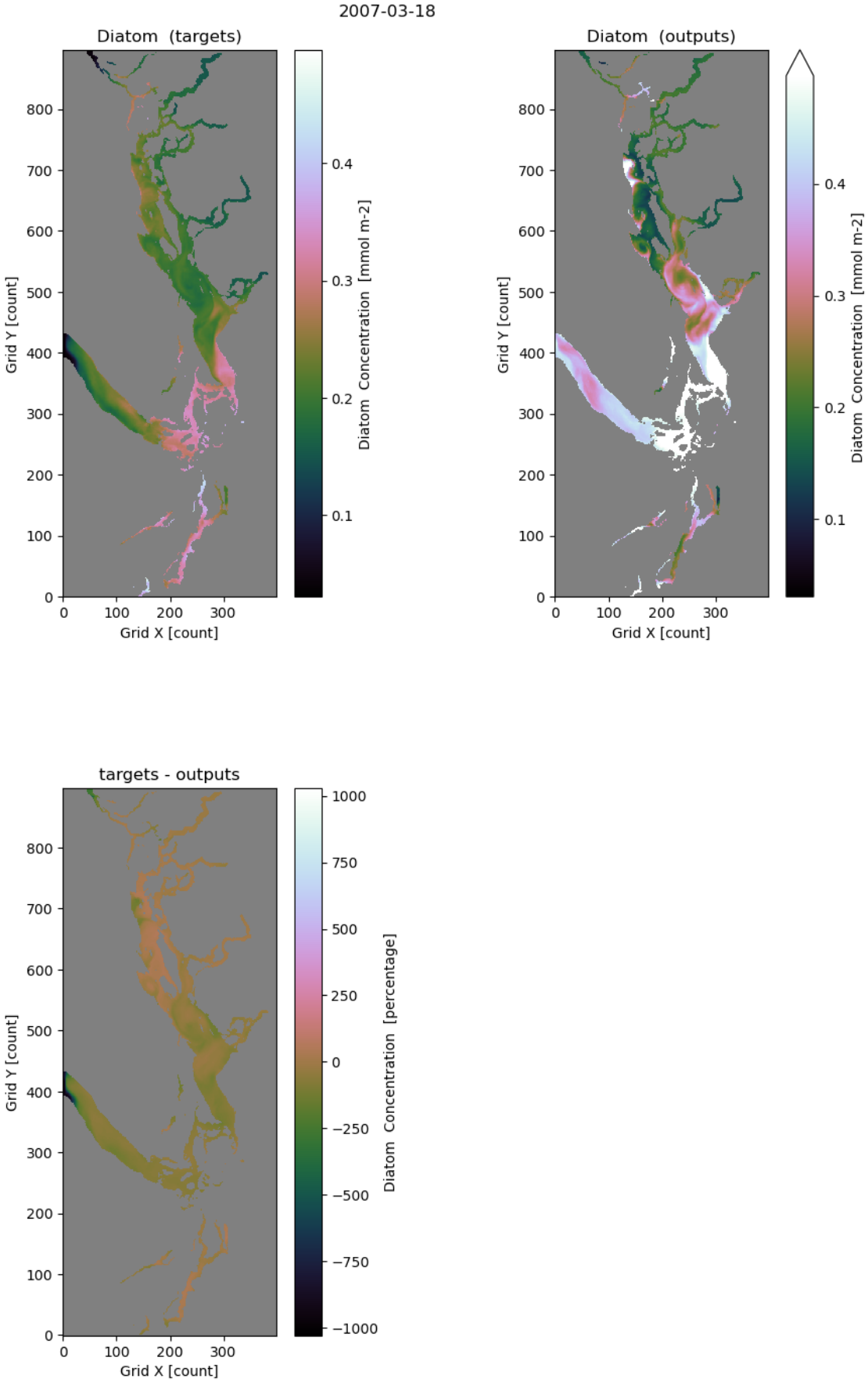
```
The amount of data points is 46479
The slope of the best fitting line is  -0.373
The correlation coefficient is: -0.615
 The mean square error is: 0.11102
```


Diatom 2020-04-21
outputs
targets
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
60
10^0
10^1
10^2

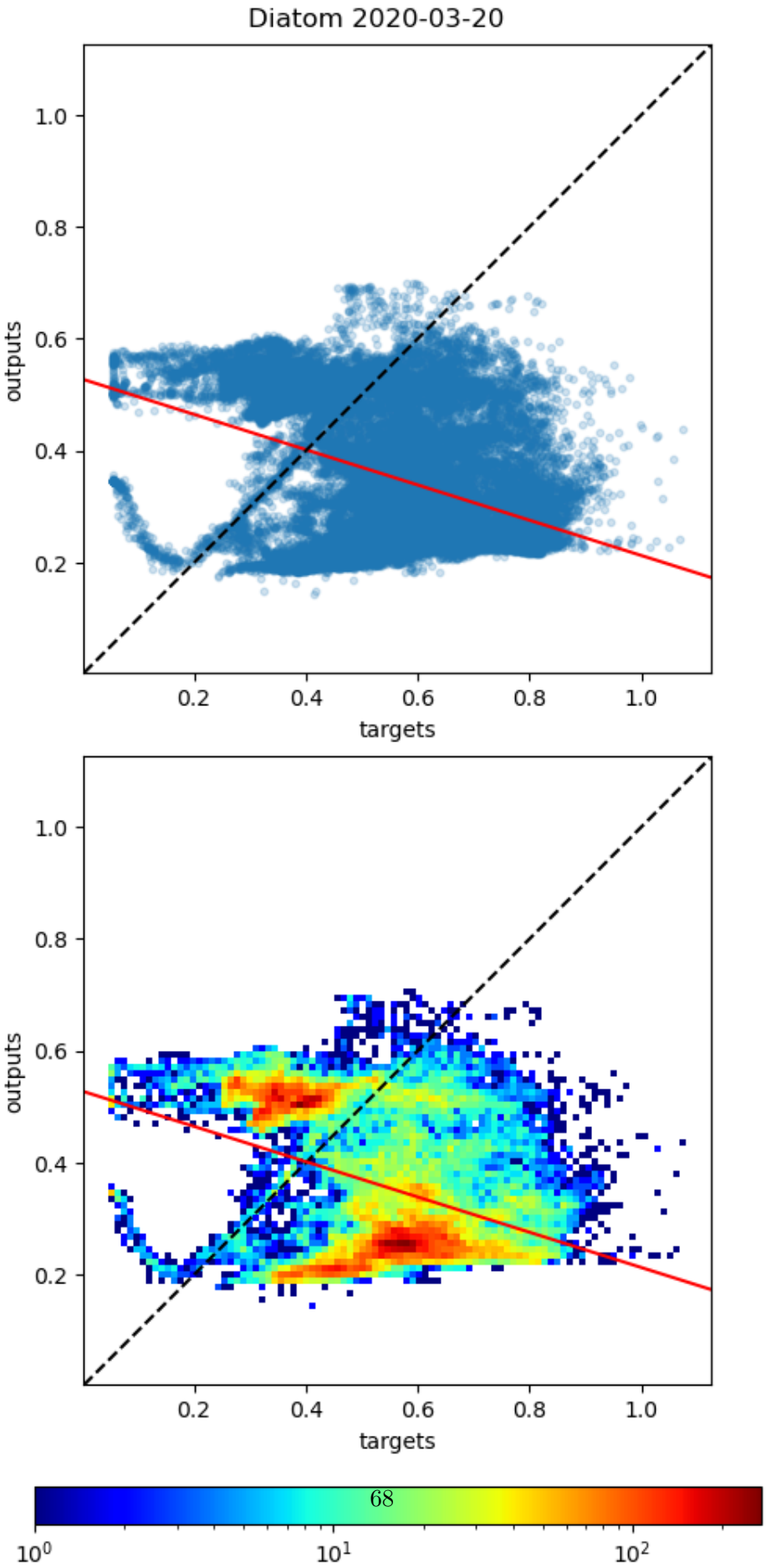

2020-04-21
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

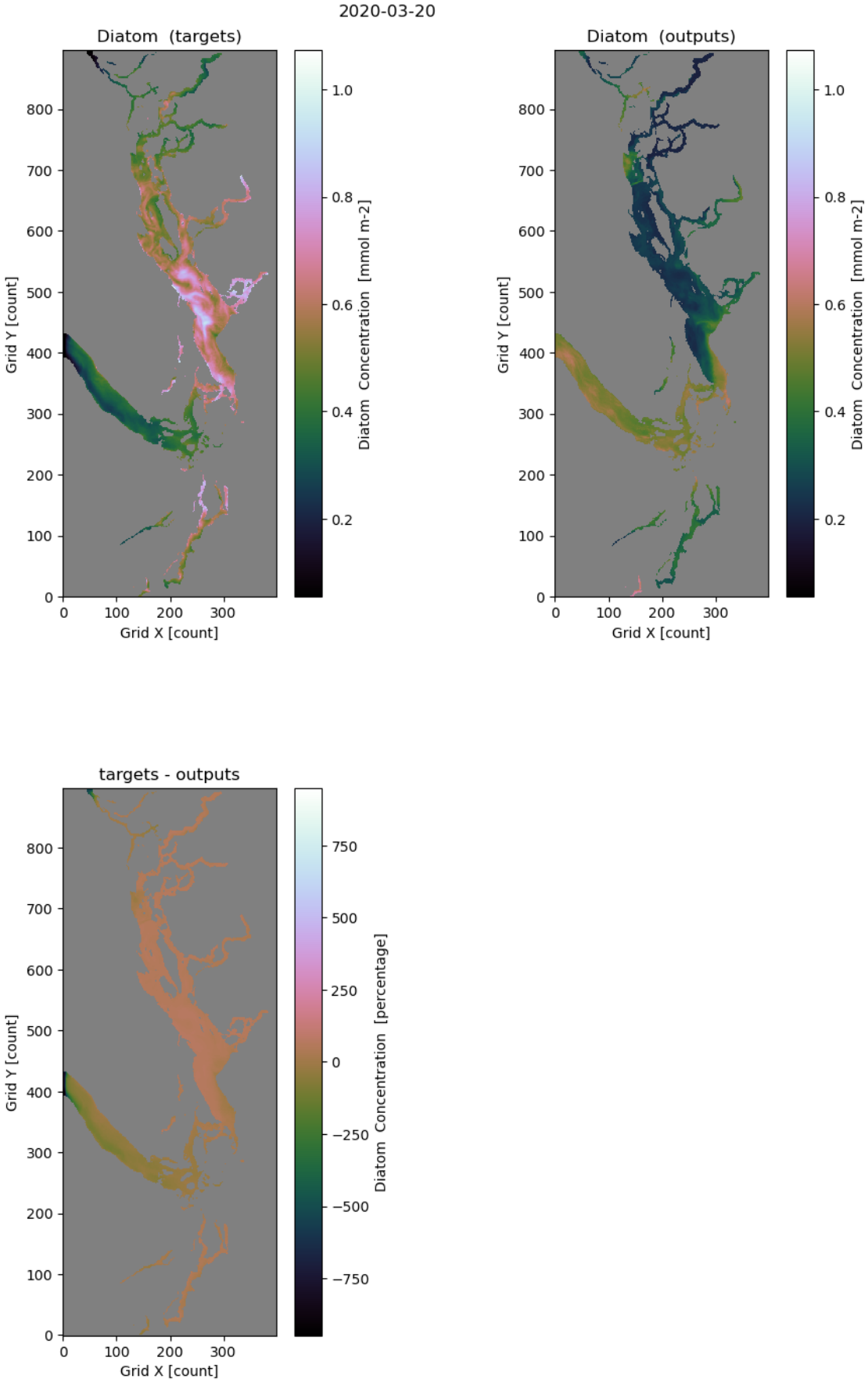
```
The amount of data points is 46479
The slope of the best fitting line is  1.217
The correlation coefficient is: 0.55
 The mean square error is: 0.02127
```


Diatom 2007-03-18
outputs
targets
0.1
0.2
0.3
0.4
0.5
0.6
64
10^0
10^1
10^2

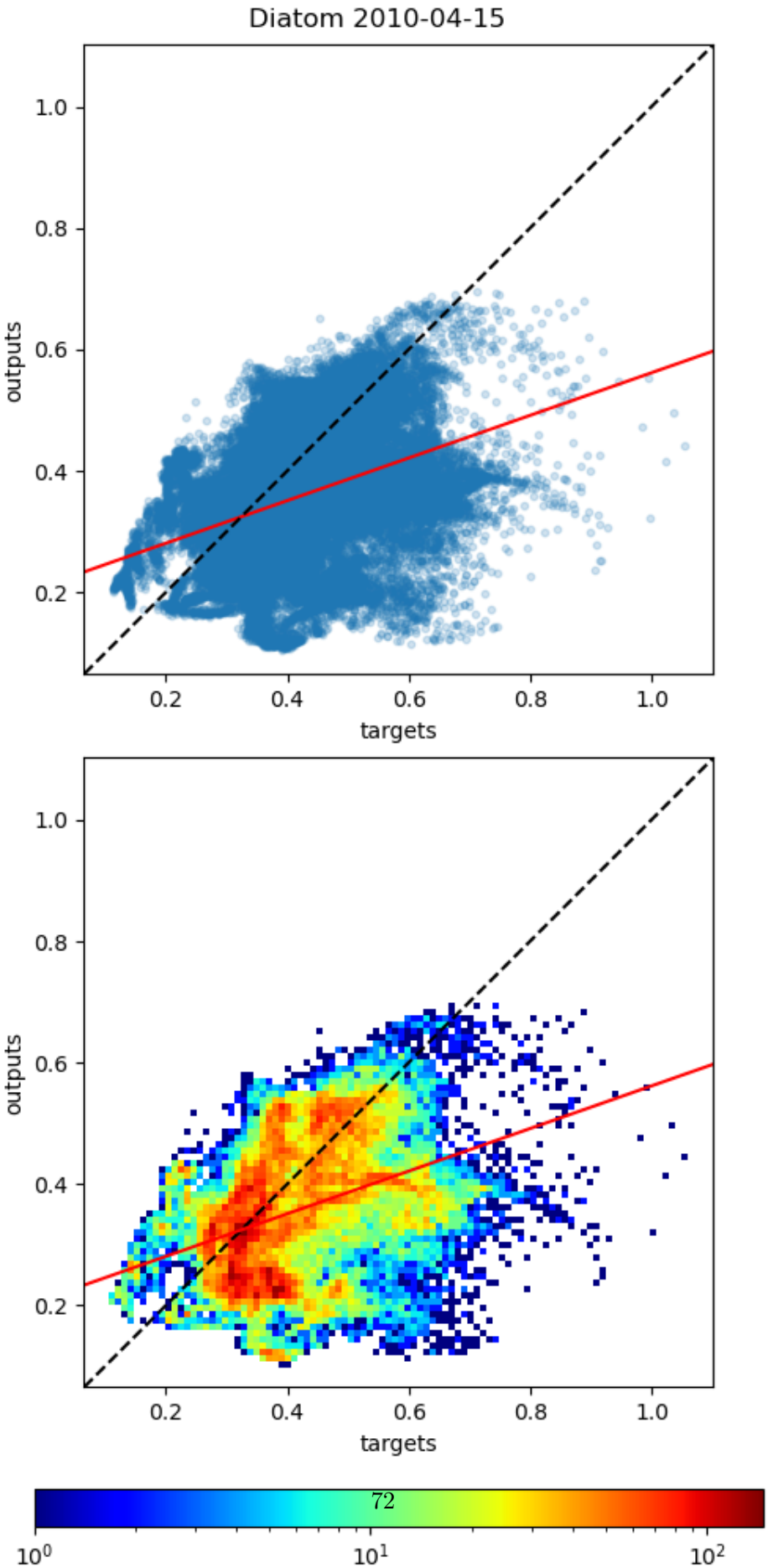

2007-03-18
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

```
The amount of data points is 46479
The slope of the best fitting line is  -0.315
The correlation coefficient is: -0.383
 The mean square error is: 0.07951
```

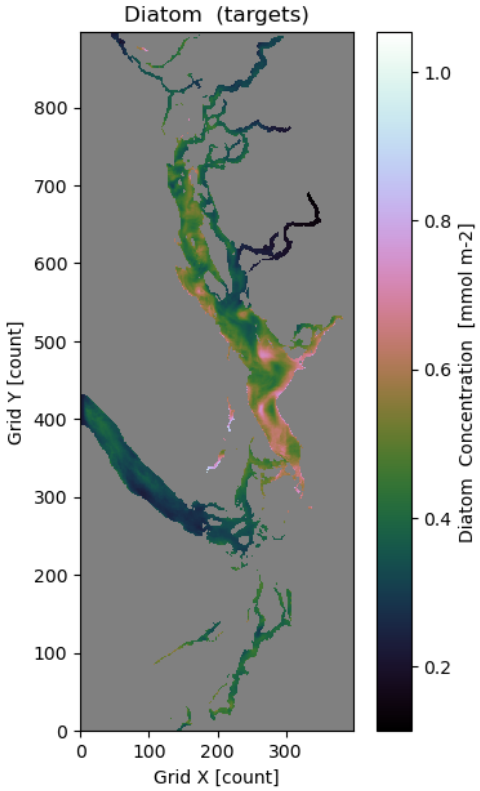

Diatom 2020-03-20
outputs
targets
1.0
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1.0
outputs
targets
1.0
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1.0
68
10^0
10^1
10^2

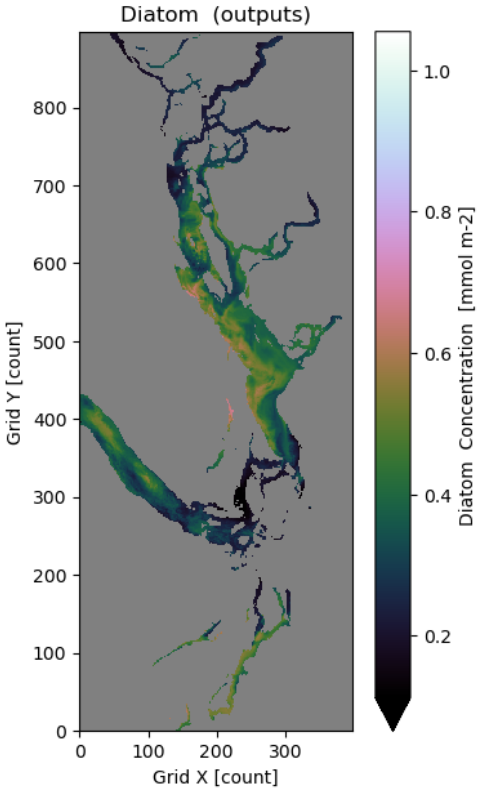

2020-03-20
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

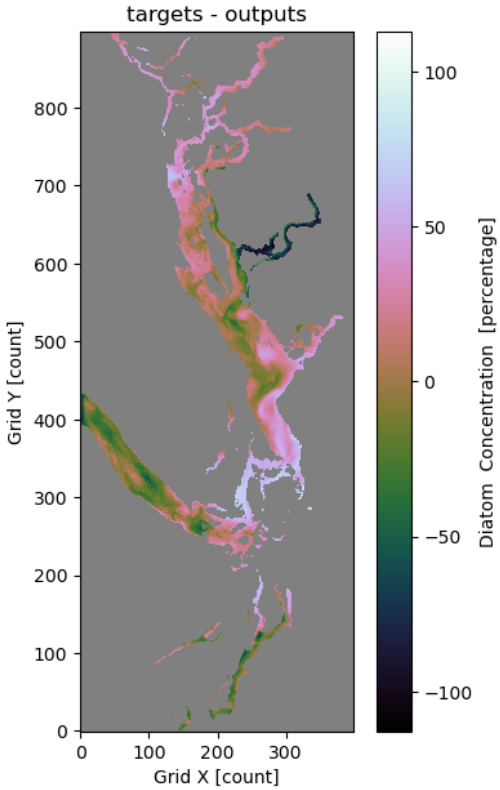
```
The amount of data points is 46479
The slope of the best fitting line is  0.351
The correlation coefficient is: 0.375
 The mean square error is: 0.01957
```


Diatom 2010-04-15
outputs
targets
1.0
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1.0
outputs
targets
1.0
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1.0
72
10^0
10^1
10^2

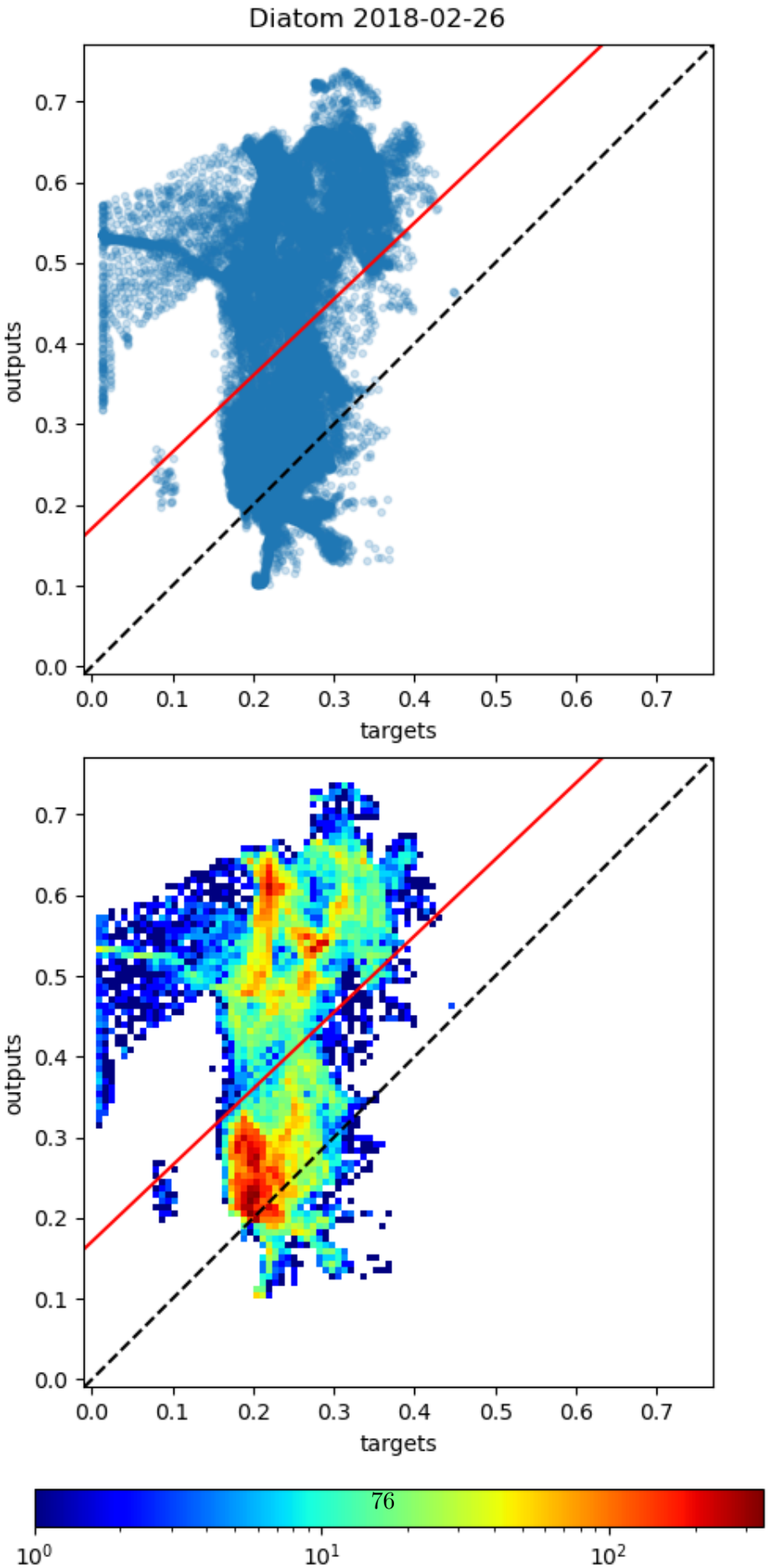
2010-04-15

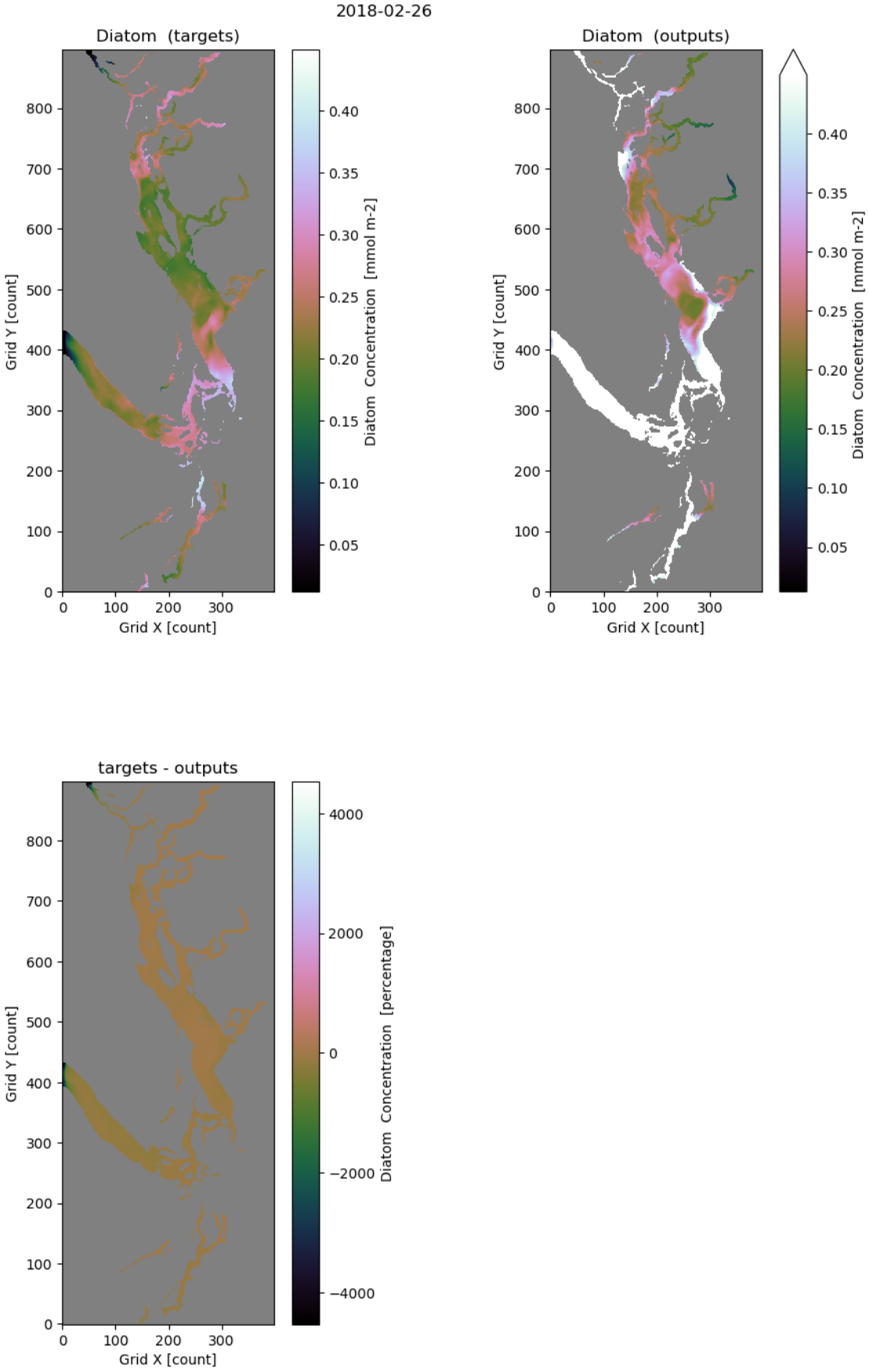

Diatom (targets)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
0
100
200
300
400
500
600
700
800
0.2
0.4
0.6
0.8
1.0


Diatom (outputs)
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
0
100
200
300
400
500
600
700
800
0.2
0.4
0.6
0.8
1.0

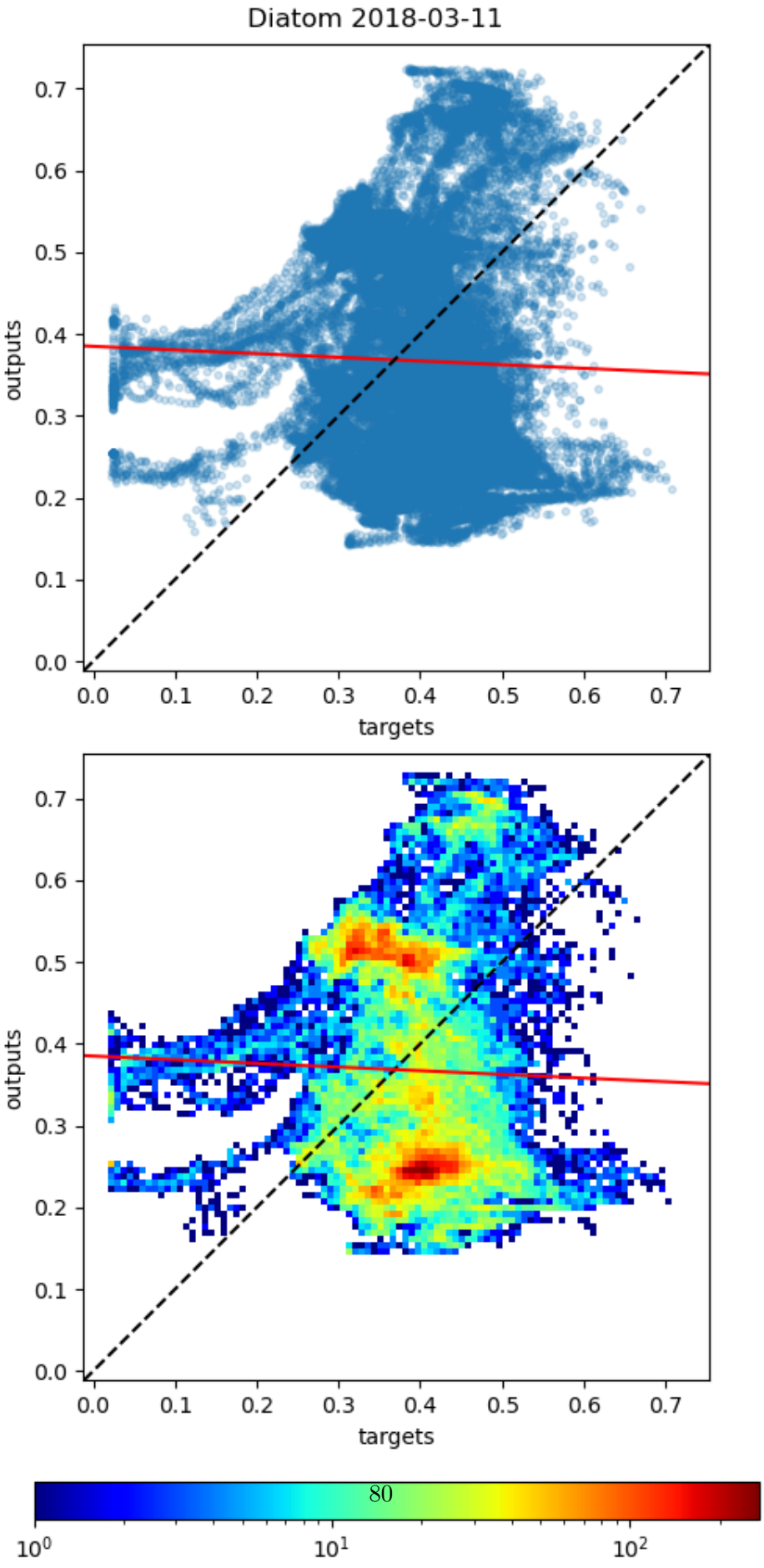

targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [percentage]
0
100
200
300
400
500
600
700
800
−100
−50
0
50
100

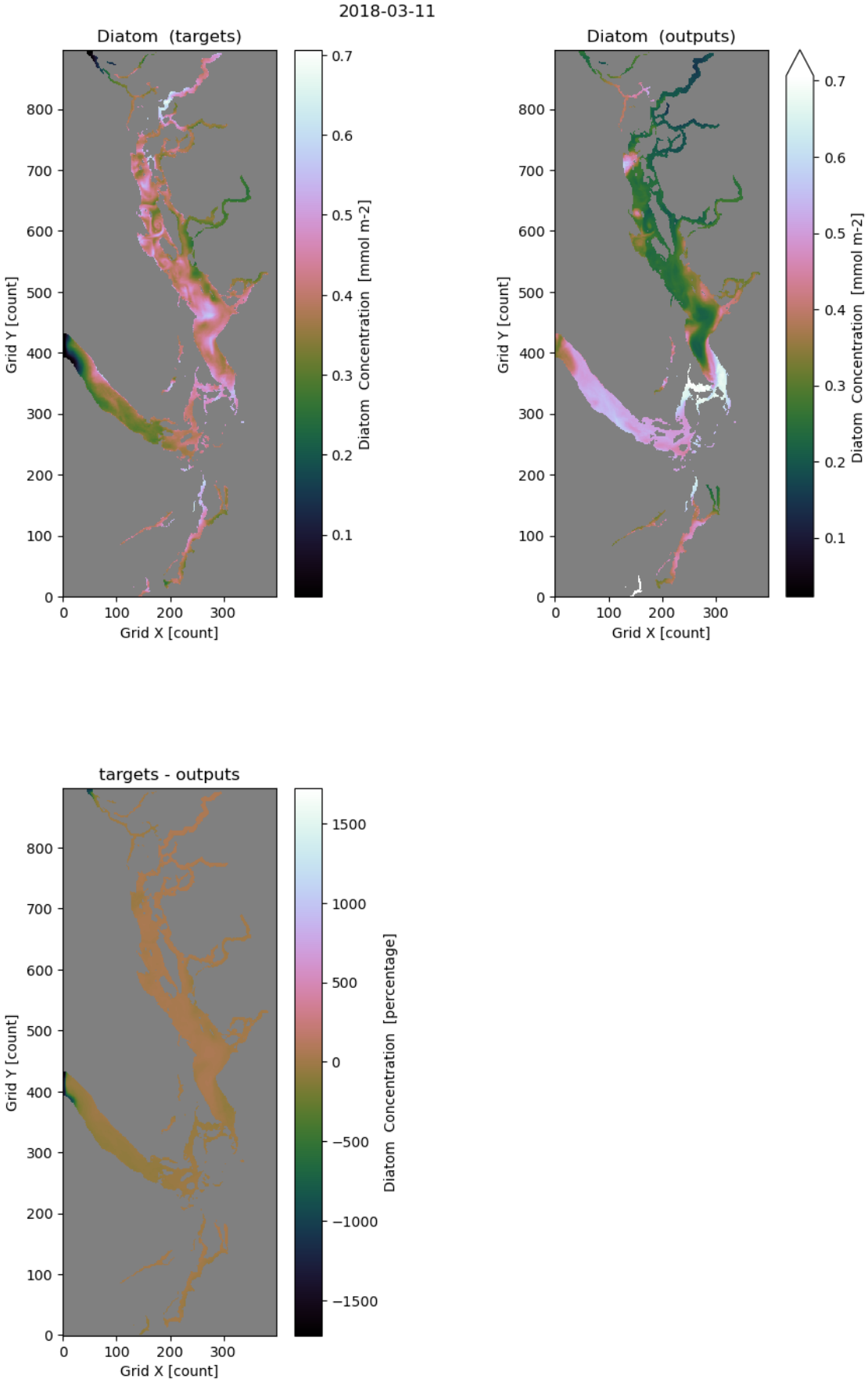
```
The amount of data points is 46479
The slope of the best fitting line is  0.947
The correlation coefficient is: 0.311
 The mean square error is: 0.04667
```


Diatom 2018-02-26
outputs
targets
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
76
10^0
10^1
10^2

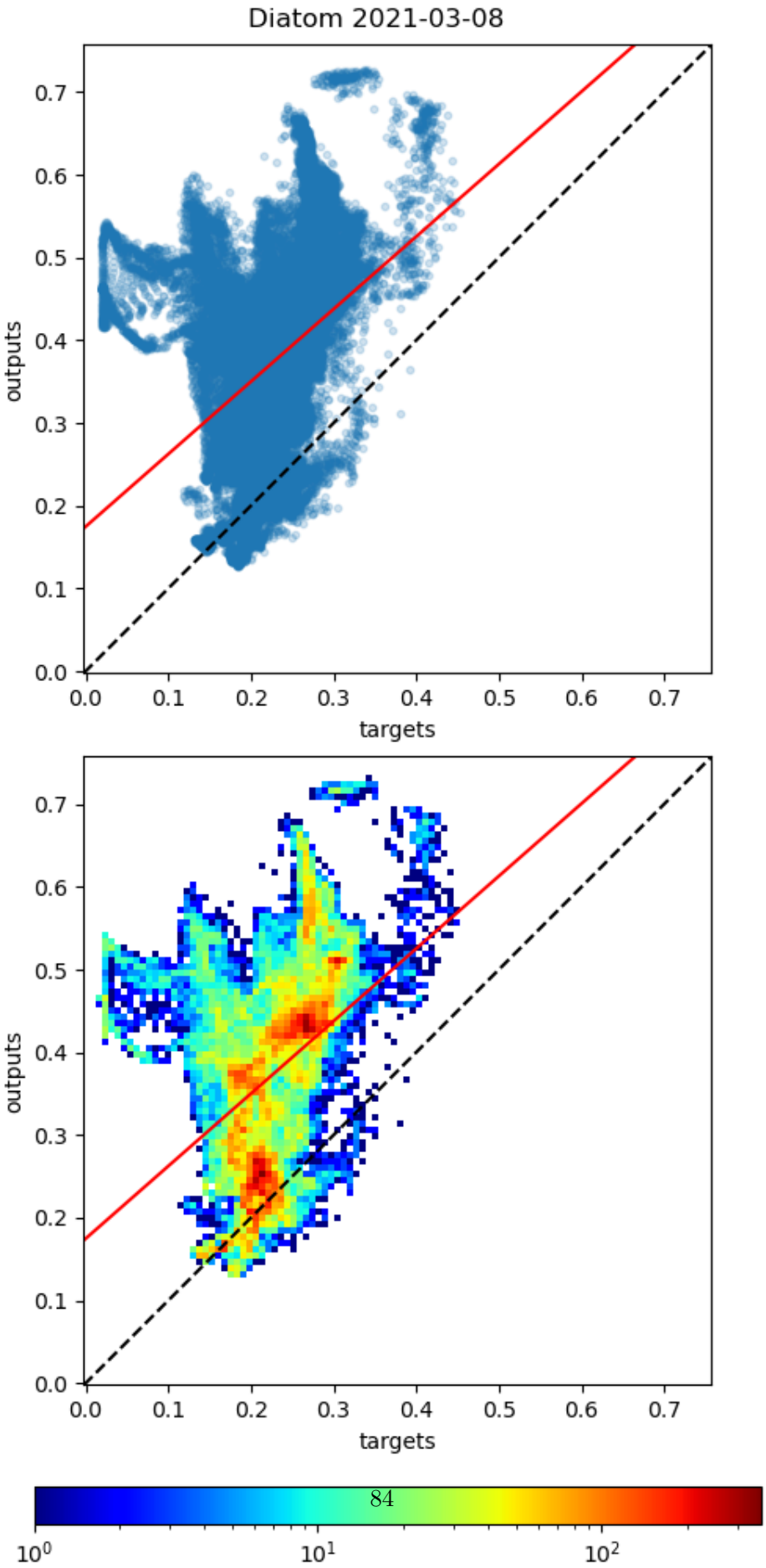

2018-02-26
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

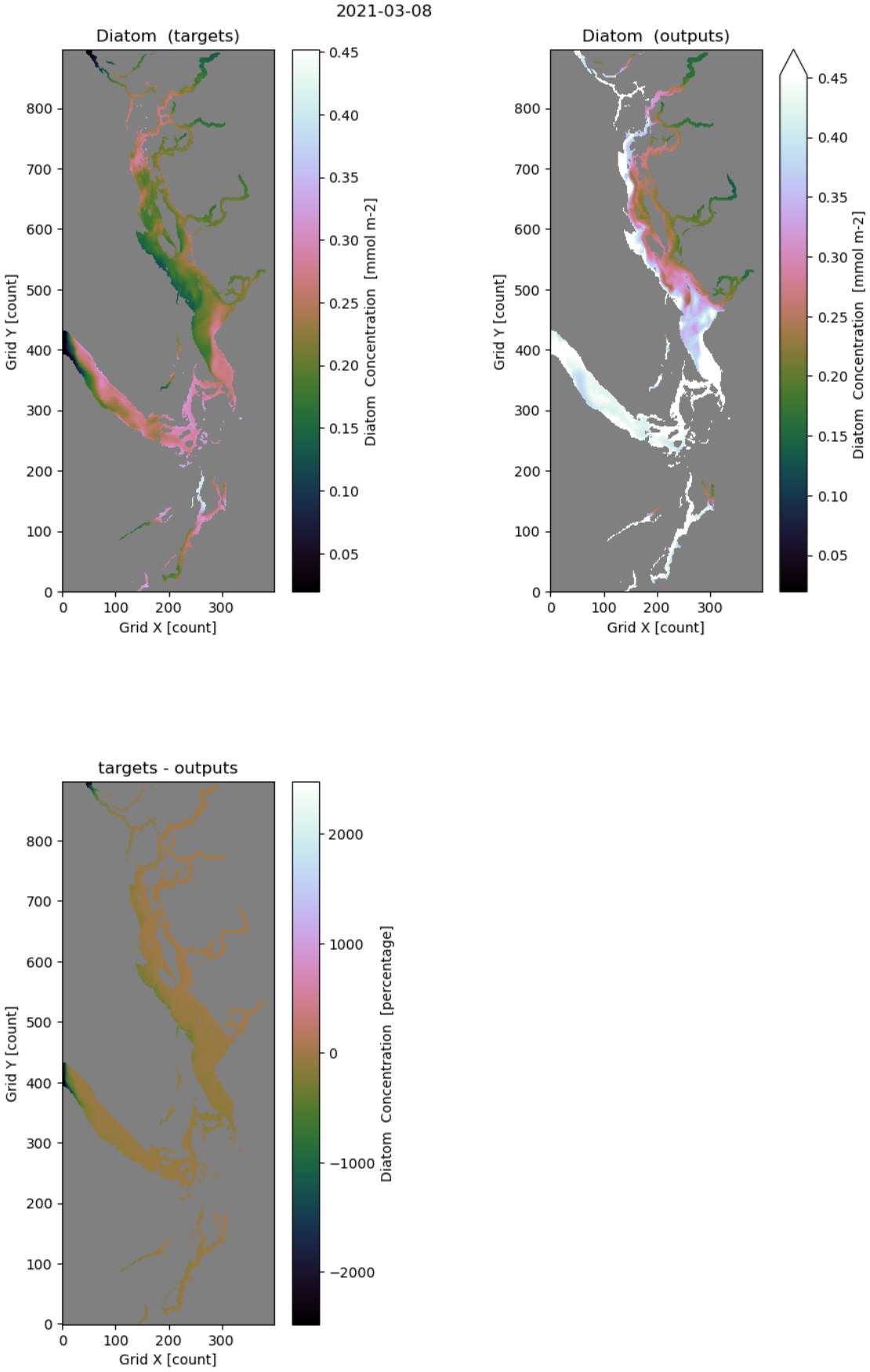
```
The amount of data points is 46479
The slope of the best fitting line is  -0.045
The correlation coefficient is: -0.028
 The mean square error is: 0.02643
```


Diatom 2018-03-11
outputs
targets
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
80
10^0
10^1
10^2


2018-03-11
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 46479
The slope of the best fitting line is  0.876
The correlation coefficient is: 0.396
 The mean square error is: 0.03383
```


Diatom 2021-03-08
outputs
targets
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
84
10^0
10^1
10^2


2021-03-08
Diatom (targets)
Diatom (outputs)
targets - outputs
Grid Y [count]
Grid X [count]
Diatom Concentration [mmol m-2]
Diatom Concentration [percentage]

[ ]: