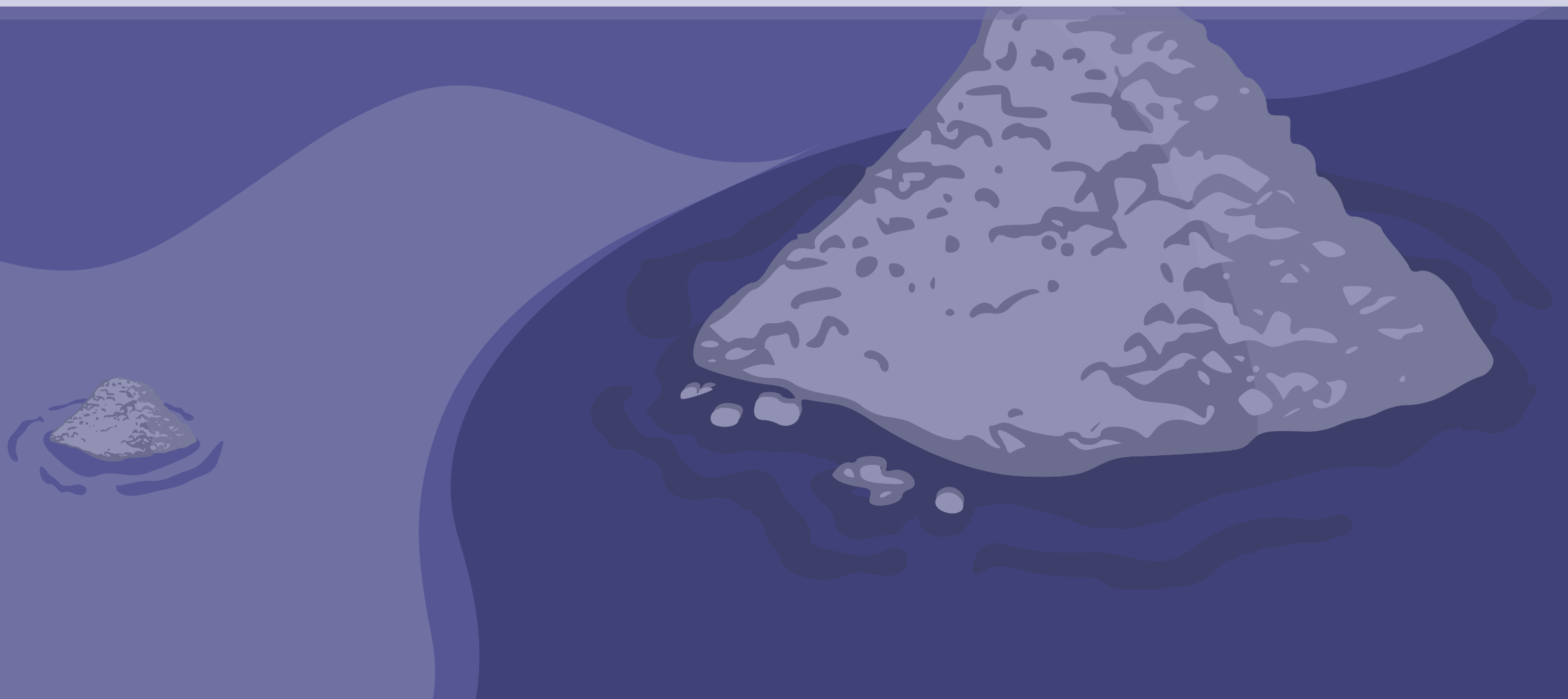




# MAKING OF SEA SURFACE SALINITY CONTRAST

Little Pictures of climate competition 2023

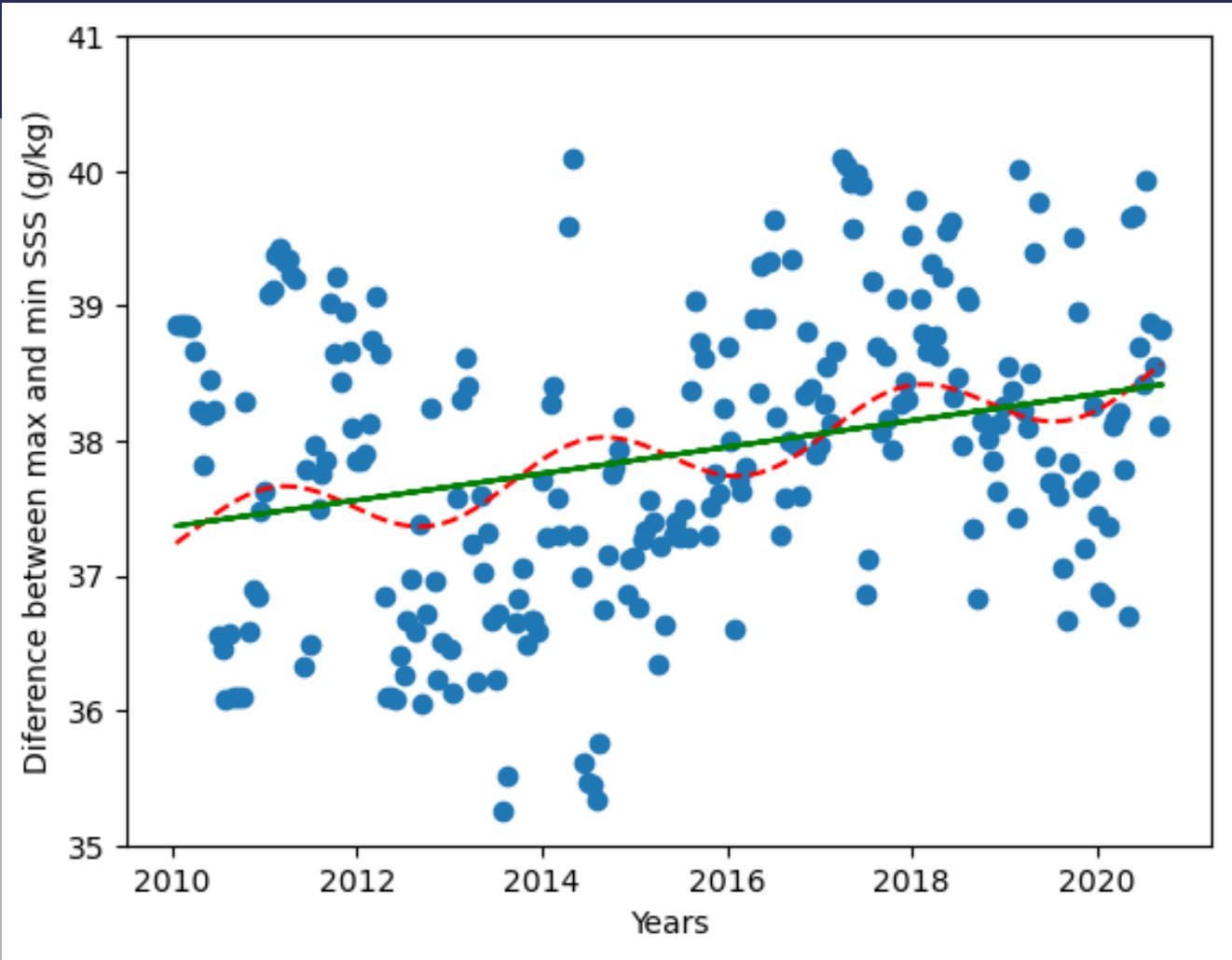
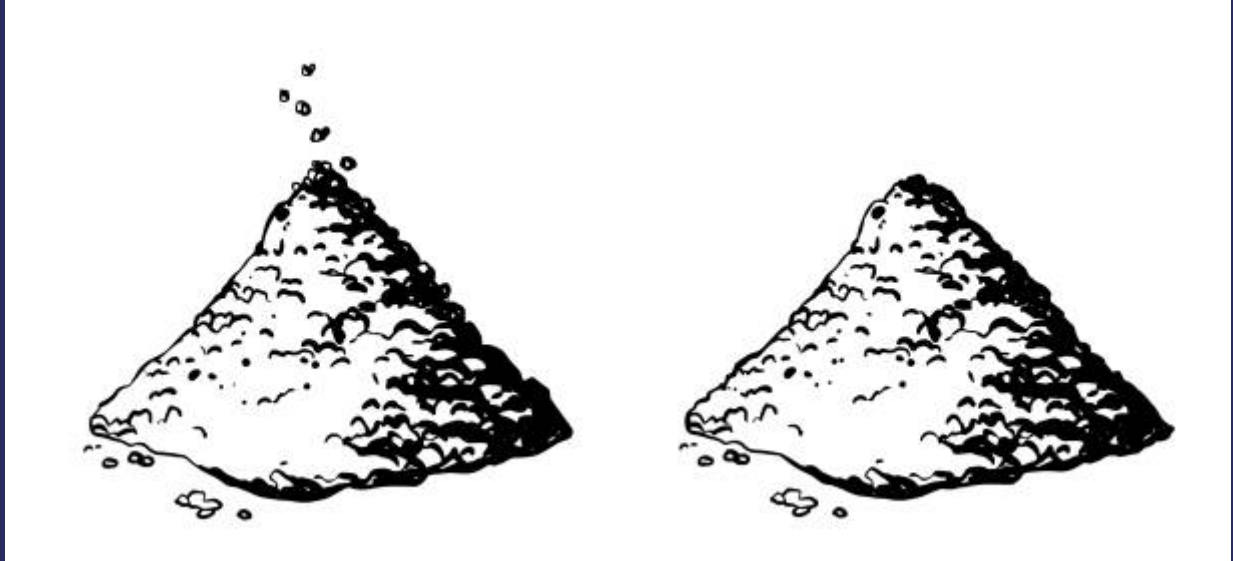
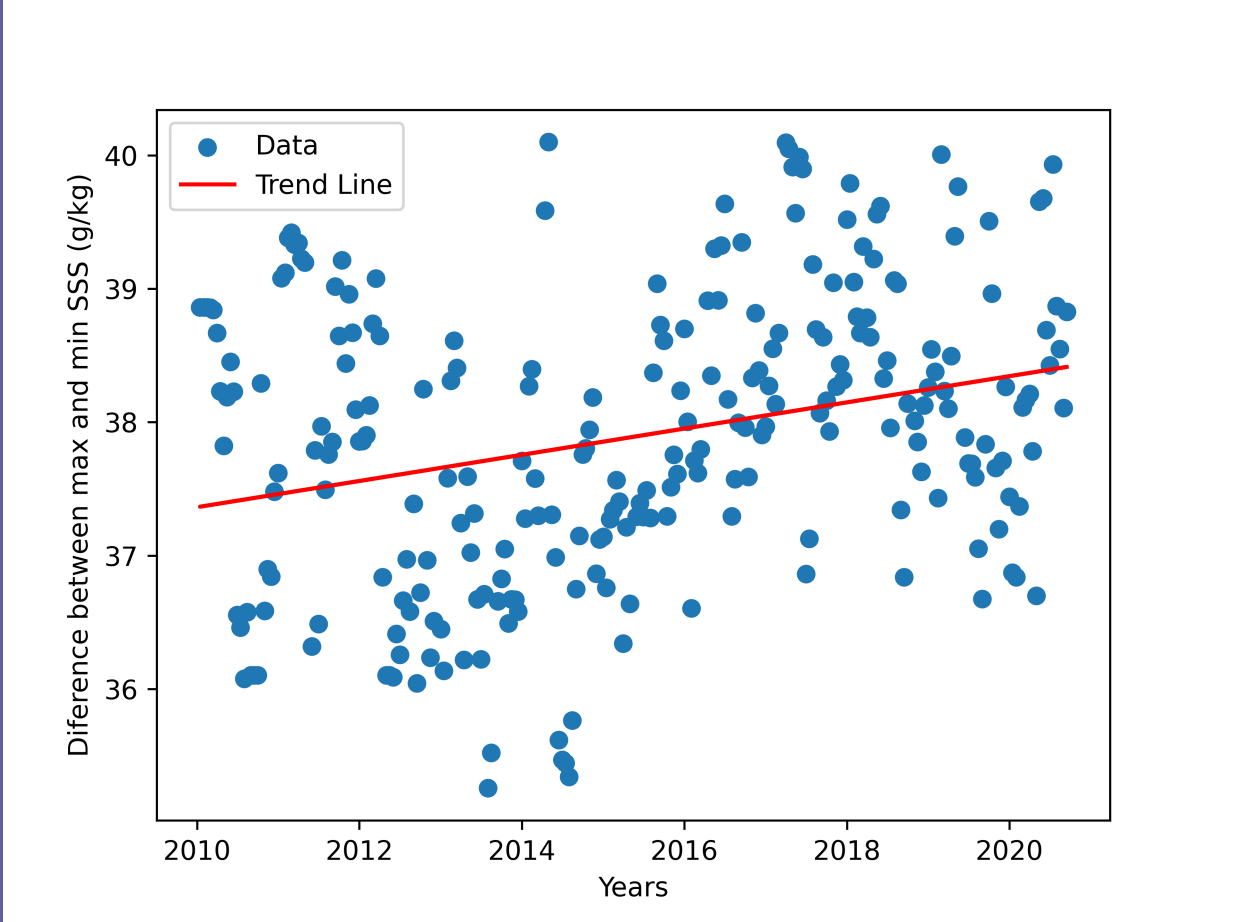
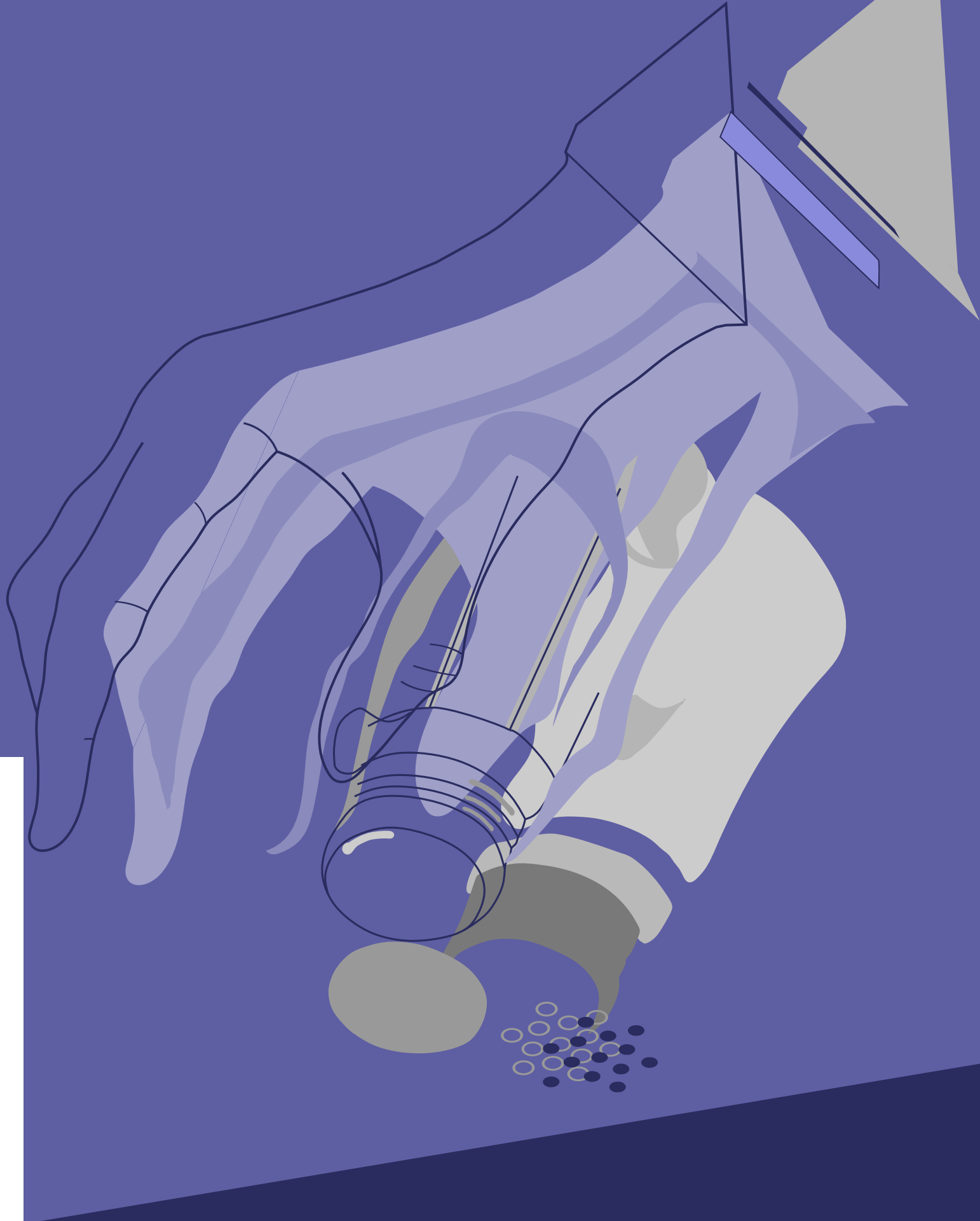
DATASET: ESA Sea Surface Salinity Climate Change Initiative (Sea\_Surface\_Salinity\_cci): Monthly sea surface salinity product, v03.21, for 2010 to 2020

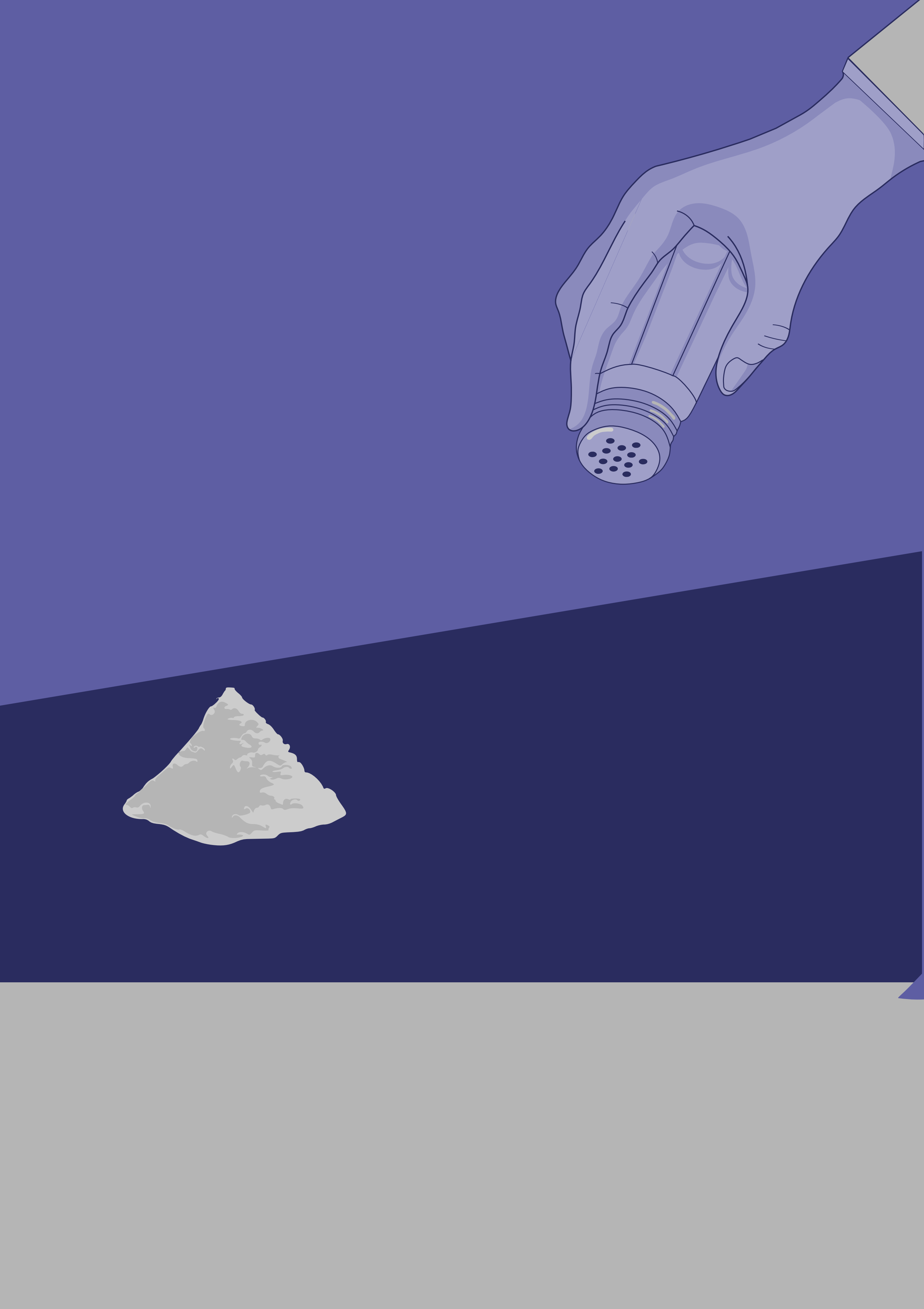


## sea surface salinity contrast

2010-2020

This illustration depicts the increase tendency in salinity contrast from 2010 to 2020. Projecting a 7.4% rise by 2050. This parameter represents the difference between the highest and lowest salinity values, playing a role in global warming, intensifying the global water cycle, and raising the risk of extreme rainfall events.











# sea surface salinity contrast

2010-2020



# increment in salinity contrast

**2010-2020**

This illustration represents the increment of salinity contrast, showing a tendency with the background line of more salty oceans.





# sea surface salinity contrast

2010-2020

This illustration depicts a 1.5% increase in salinity contrast from 2010 to 2020. Sea surface salinity contrast represents the difference between the highest and lowest salinity values, contributing to global warming, intensifying the global water cycle, and raising the risk of extreme rainfall events.

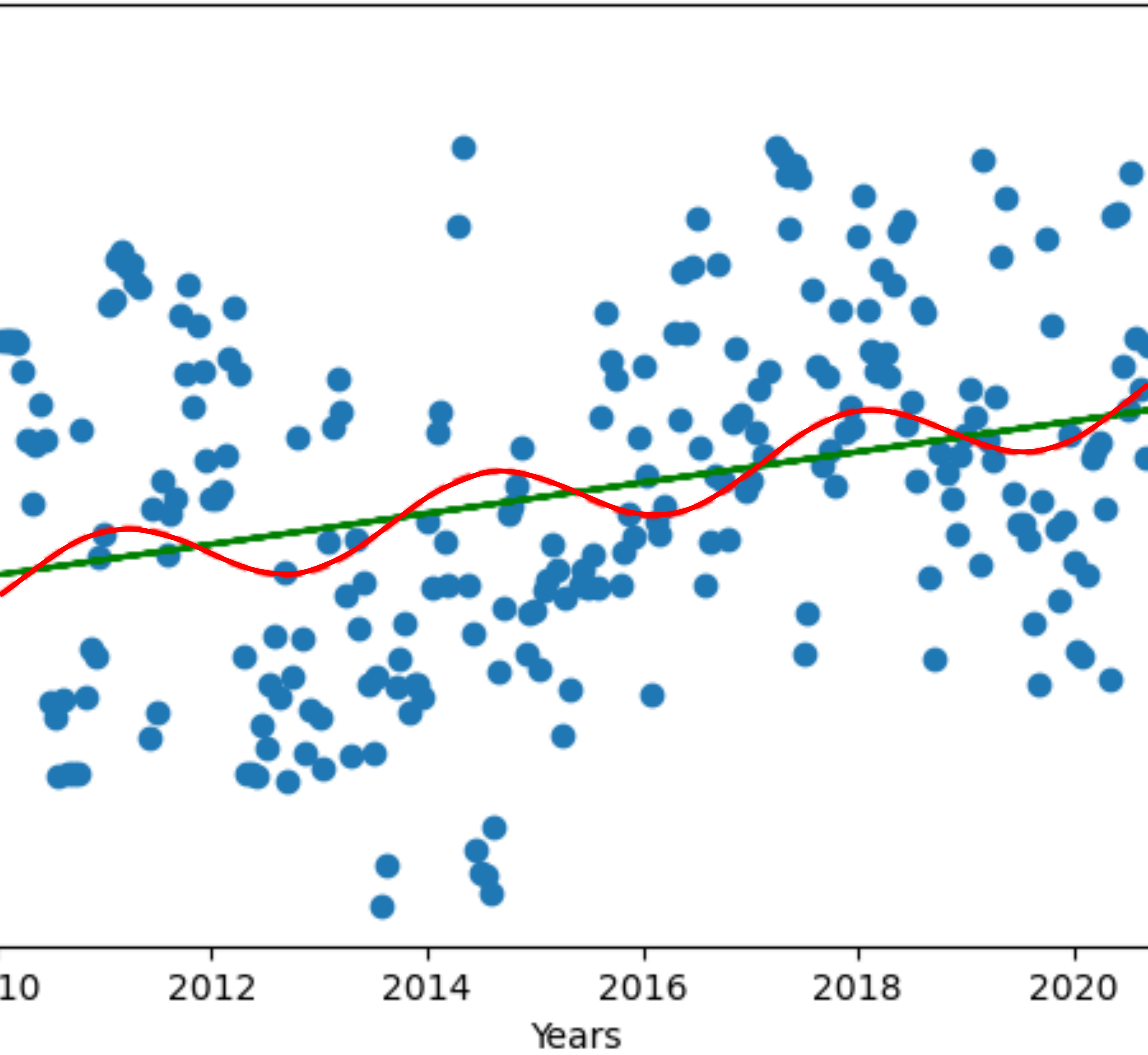


```

1 import netCDF4
2 import numpy as np
3 import re
4 from datetime import datetime
5 import matplotlib.pyplot as plt
6 import os
7 from matplotlib.dates import date2num, num2date
8 from scipy.optimize import curve_fit
9 import csv
10
11 dates_values = []
12 maximals = []
13 minimal = []
14 file_names = os.listdir('/home/home/Desktop/DATA/30days/2010')
15
16 for i in range(23):
17     file_name = '/home/home/Desktop/DATA/30days/2010'+ '/' +file_names[i]
18     date = re.search("([0-9]{4}[0-9]{2}[0-9]{2})", file_name)
19     date_value = datetime.strptime(date.group(), '%Y%m%d').date()
20     dates_values.append(date_value)
21     date_string = str(datetime.strptime(date.group(), '%Y%m%d').date())
22     date_string = date_string.replace("-", "")
23     f = netCDF4.Dataset(file_name)
24     sss = f.variables['sss']
25     values_max = []
26     values_min = []
27
28     for ii in range(584):
29         a = sss[0,ii,:]
30         a = list(a)
31         values_max.append(max(a))
32         values_min.append(min(a))
33
34
35     max_value = [v for v in values_max if v != '--']
36     max_value = max(max_value)
37     maximals.append(max_value)
38
39     min_value = [v for v in values_min if v != '--']
40     min_value = min(min_value)
41     minimal.append(min_value)
42
43 print(2010)
44
45 years = ['2011','2012','2013','2014','2015','2016','2017',
46         '2018','2019']
47
48 for j in range(9):
49
50     file_names = os.listdir('/home/home/Desktop/DATA/30days/'+years[j])
51
52     for k in range(23):
53         file_name = '/home/home/Desktop/DATA/30days/'+years[j]+'/' +file_names[k]
54         date = re.search("([0-9]{4}[0-9]{2}[0-9]{2})", file_name)
55         date_value = datetime.strptime(date.group(), '%Y%m%d').date()
56         dates_values.append(date_value)
57         date_string = str(datetime.strptime(date.group(), '%Y%m%d').date())
58         date_string = date_string.replace("-", "")
59         f = netCDF4.Dataset(file_name)
60         sss = f.variables['sss']
61         values_max = []
62         values_min = []
63
64         for kk in range(584):
65             a = sss[0,kk,:]
66             a = list(a)
67             values_max.append(max(a))
68             values_min.append(min(a))
69
70
71         max_value = [v for v in values_max if v != '--']
72         max_value = max(max_value)
73         maximals.append(max_value)
74
75         min_value = [v for v in values_min if v != '--']
76         min_value = min(min_value)
77         minimal.append(min_value)
78
79     print(years[j])
80
81 file_names = os.listdir('/home/home/Desktop/DATA/30days/2020')
82
83 for l in range(18):
84     file_name = '/home/home/Desktop/DATA/30days/2020'+ '/' +file_names[l]
85     date = re.search("([0-9]{4}[0-9]{2}[0-9]{2})", file_name)
86     date_value = datetime.strptime(date.group(), '%Y%m%d').date()
87     dates_values.append(date_value)
88     date_string = str(datetime.strptime(date.group(), '%Y%m%d').date())
89     date_string = date_string.replace("-", "")
90     f = netCDF4.Dataset(file_name)
91     sss = f.variables['sss']
92     values_max = []
93     values_min = []
94
95     for ll in range(584):
96         a = sss[0,ll,:]
97         a = list(a)
98         values_max.append(max(a))
99         values_min.append(min(a))
100
101
102     max_value = [v for v in values_max if v != '--']
103     max_value = max(max_value)
104     maximals.append(max_value)
105
106     min_value = [v for v in values_min if v != '--']
107     min_value = min(min_value)
108     minimal.append(min_value)
109
110
111 difference = np.array(maximals) - np.array(minimal)
112 plt.scatter(dates_values,difference)
113 plt.show()
114
115 # Convert datetime objects to numeric values
116 numeric_dates = date2num(dates_values)
117
118 difference = np.array(maximals) - np.array(minimal)
119
120 # Scatter plot
121 plt.scatter(num2date(numeric_dates), difference, label='Data')
122
123 # Add a trend line
124 coefficients = np.polyfit(numeric_dates, difference, 1)
125 trend_line = np.poly1d(coefficients)
126 plt.plot(num2date(numeric_dates), trend_line(numeric_dates),
127         color='red', label='Trend Line')
128
129 # Show the legend
130 plt.legend()
131
132 plt.ylabel('Diference between max and min SSS (g/kg)')
133 plt.xlabel('Years')
134
135 # Display the plot
136 plt.savefig('filename2.png', dpi=600)
137 plt.show()
138
139 with open('data.csv', 'w', newline='') as file:
140     # Step 4: Using csv.writer to write the list to the CSV file
141     writer = csv.writer(file)
142     writer.writerow(dates_values) # Use writerow for single list
143     writer.writerow(minimal) # Use writerow for single list
144     writer.writerow(maximals) # Use writerow for single list
145     writer.writerow(difference) # Use writerow for single list
146
147 numeric_dates = date2num(dates_values)
148 date_plus_agenda = np.append(numeric_dates, date2num(datetime(2050, 1, 1)))
149 date_plus_agenda = list(date_plus_agenda)
150 del date_plus_agenda[248]
151 difference = np.array(maximals) - np.array(minimal)
152 difference_plus_agenda = np.append(difference, 39.320752)
153
154 # Scatter plot
155 plt.scatter(date_plus_agenda, difference_plus_agenda, label='Data')
156
157 # Add a trend line
158 coefficients = np.polyfit(numeric_dates, difference_plus_agenda, 1)
159 trend_line = np.poly1d(coefficients)
160 plt.plot(date_plus_agenda, trend_line(date_plus_agenda),
161         color='red', label='Trend Line')
162
163 # Show the legend
164 plt.legend()
165
166 plt.ylabel('Diference between max and min SSS (g/kg)')
167 plt.xlabel('Years')
168
169 def objective(x, a, b, c, d):
170     return a * np.sin(b - x/200) + c * x**2 + d
171
172
173 # choose the input and output variables
174 x, y = date2num(dates_values), difference
175 # curve fit
176 popt, _ = curve_fit(objective, x, y)
177 # summarize the parameter values
178 a, b, c, d = popt
179 print(popt)
180 # plot input vs output
181 plt.scatter(x, y)
182 # define a sequence of inputs between the smallest and largest known inputs
183 x_line = np.arange(min(x), max(x), 1)
184 # calculate the output for the range
185 y_line = objective(x_line, a, b, c, d)
186 # create a line plot for the mapping function
187 plt.plot(x_line, y_line, '--', color='red')
188 plt.plot(num2date(numeric_dates), trend_line(numeric_dates),
189         color='green', label='Trend Line')
190 plt.ylabel('Diference between max and min SSS (g/kg)')
191 plt.xlabel('Years')
192 plt.ylim([35, 41])
193 plt.show()

```





2010-2020



# sea surface salinity contrast

2010-2020



# sea surface salinity contrast

2010-2020



low-salt areas  
with 88% less salinity,  
lets adjust the sizes:

$164\text{px} \times 0.12 = 20\text{px}$

size of the figure: 164px

# sea surface salinity contrast

2010-2020



# sea surface salinity contrast

**2010-2020**

This illustration depicts the increase tendency in salinity contrast from 2010 to 2020. Projecting a 7.4% rise by 2050. This parameter represents the difference between the highest and lowest salinity values, playing a role in global warming, intensifying the global water cycle, and raising the risk of extreme rainfall events.



# sea surface salinity contrast

**2010-2020**

This illustration depicts the increase tendency in salinity contrast from 2010 to 2020. Projecting a 7.4% rise by 2050. This parameter represents the difference between the highest and lowest salinity values, playing a role in global warming, intensifying the global water cycle, and raising the risk of extreme rainfall events.





# sea surface salinity contrast

2010-2020



# sea surface salinity contrast

**2010-2020**

This illustration depicts the increase tendency in salinity contrast from 2010 to 2020. Projecting a 7.4% rise by 2050. This parameter represents the difference between the highest and lowest salinity values, playing a role in global warming, intensifying the global water cycle, and raising the risk of extreme rainfall events.