

ORP 数据整理

Guangyao Zhao

2023-02-07

Contents

```
1 import os
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 dir_path = os.path.dirname(__file__) # directory path
7 file_path = os.path.join(dir_path, "Raw data") # dataset file path
8 file_list = os.listdir(file_path)
9
10 # toc: 合并 csv 文件
11 # 删除非 csv 文件
12 for file in file_list:
13     if ".CSV" not in file:
14         file_list.remove(file)
15
16 res = pd.DataFrame([]) # 合并后的结果
17 for file in file_list:
18     tmp_file_path = os.path.join(file_path, file)
19     # 数据分隔
20     dataset = (
21         pd.read_csv(tmp_file_path, header=None, index_col=None)
22         .iloc[:, 0]
23         .str.split(";", expand=True)
```

```

24 )
25 dataset.columns = dataset.iloc[0, :] # 保留表头名称
26 dataset = dataset.loc[1:, ["Date/Time", "Value"]] # 提取出值
27 dataset.columns = ["Date", "Value"] # 更改表头名称
28 dataset.set_index("Date", inplace=True) # 设置 index 为 Date 列
29
30 res = pd.concat([res, dataset], axis=0) # 合并当前 csv 到结果文件中
31 res.index = pd.to_datetime(res.index).strftime("%Y-%m-%d %H:%M")
32
33 # 处理重复值
34 res = res.sort_index(ascending=True)
35 res = res[~res.index.duplicated(keep="first")] # 删除重复的 index
36
37 # toc: 保存含有缺失值的结果
38 res_path = os.path.join(dir_path, "res_isna.csv")
39 res.to_csv(res_path) # file: 输出结果
40
41 # toc: 处理缺失值
42 dir_path = os.path.dirname(__file__) # directory path
43 file_path = os.path.join(dir_path, "res_isna.csv") # dataset file path
44 dataset = pd.read_csv(file_path, index_col=[0], header=[0], parse_dates=True) # 读取文件
45
46 date_range = pd.date_range(dataset.index.min(), dataset.index.max(), freq="min")
47 res = pd.DataFrame(range(date_range.size), index=date_range)
48 res = pd.concat([res, dataset], axis=1, join="outer")["Value"]
49 print(" 缺失值有 {num} 个:\n".format(num=res[res.isna()].shape[0]), res[res.isna()])
50 res = res.fillna(method="ffill") # 去除缺失值
51
52 # toc: 保存无缺失值的结果
53 res.index.name = "Date"
54 res_path = os.path.join(dir_path, "res_nona.csv")
55 res.to_csv(res_path) # file: 输出结果
56
57 # toc: 做图
58 dir_path = os.path.dirname(__file__) # directory path
59 file_path = os.path.join(dir_path, "res_nona.csv") # dataset file path

```

```

60 dataset = pd.read_csv(file_path, index_col=[0], header=[0], parse_dates=True) # 读取文件
61 year_li = set(list(map(lambda x: x.year, dataset.index))) # 有多少个年份有数据
62 print(" 当前年份有: ", year_li)
63
64 # 对图做标注
65 line_li = []
66 for i in range(4):
67     line_li.append(1 + 6 * i) # 本循环结束后
68     line_li.append(2.5 + 1 + 6 * i) # 厌气后
69     line_li.append(5.0 + 1 + 6 * i) # 好气后
70
71 # 分别对每个月进行做图
72 for year in year_li:
73     month_li = set(list(map(lambda x: x.month, dataset.index))) # 有多少个月有数据
74     print(" 当前月份有: ", month_li)
75     for month in month_li:
76         month_dataset = dataset[dataset.index.month == month] # 提取出当前月数据
77         day_li = list(set(list(map(lambda x: x.day, month_dataset.index)))) # 本月有多少天有数据
78         print("{month}月有数据的日期为:".format(month=month), day_li)
79
80     # ORP curve
81     fig_1 = plt.figure(figsize=(8 * 2.5, 6 * len(day_li)))
82     axes = fig_1.subplots(len(day_li))
83
84     # ORP extremum
85     fig_2 = plt.figure(figsize=(8 * 2, 6)) # 每日最大和最小 ORP
86     ax_extremum = fig_2.add_subplot(111)
87     max_ORP, min_ORP = [], []
88
89     # 分别对每一天进行做图
90     for j, day in enumerate(day_li):
91         day_dataset = month_dataset[month_dataset.index.day == day]
92         day_dataset[
93             day_dataset.index.hour.isin([0, 6, 12, 18])
94             ] = pd.NA #! 将闲置的时间值设置为空置
95         max_ORP.append(day_dataset["Value"].max()) # 本日最大 ORP

```

```

96 min_ORP.append(day_dataset["Value"].min()) # 本日最小 ORP
97
98 # fig_1
99 # 判断起始时间
100 start_hour, end_hour = day_dataset.index.hour[0], day_dataset.index.hour[-1]
101 start_minute, end_minute = (
102     day_dataset.index.minute[0],
103     day_dataset.index.minute[-1],
104 )
105 x = np.linspace(
106     start_hour + start_minute / 60,
107     end_hour + end_minute / 60,
108     day_dataset.shape[0],
109 )
110 # 开始绘图
111 axes[j].plot(x, day_dataset.values, color="green", linewidth=2)
112
113 axes[j].set_xlim(-0.5, 24.5)
114 axes[j].set_ylim(-300, 600)
115 locator_x_major = plt.MultipleLocator(1)
116 axes[j].xaxis.set_major_locator(locator_x_major)
117 axes[j].set_xlabel("Elapsed time (hour)")
118 axes[j].set_ylabel("ORP (mV)")
119 axes[j].set_title(
120     "{date}".format(date=day_dataset.index[0].strftime("%Y-%m-%d"))
121 )
122 axes[j].tick_params(axis="both", which="major", direction="inout")
123
124 # 标注线
125 for line in line_li:
126     if line % 6 == 1:
127         axes[j].axvline(x=line, linestyle="--", color="red", linewidth=2)
128     else:
129         axes[j].axvline(
130             x=line, linestyle="--", color="black", linewidth=0.3
131         )

```

```

132
133     # axes[j].axhline(
134     #     y=day_dataset.values.min(), linestyle="--", linewidth=0.3
135     # ) # 画出当日最小值
136
137     # axes[j].axhline(
138     #     y=day_dataset.values.max(), linestyle="--", linewidth=0.3
139     # ) # 画出当日最大值
140
141     # fig_2
142     markersize = 10
143     linewidth = 0.5
144     alpha = 1
145     color_li = [(85 / 255, 187 / 255, 194 / 255), (232 / 255, 125 / 255, 115 / 255)]
146     data_li = [max_ORP, min_ORP]
147     for k, ORP in enumerate(data_li):
148         ax_extremum.plot(
149             day_li,
150             ORP,
151             marker="o",
152             color="black",
153             markersize=markersize,
154             markerfacecolor=color_li[k],
155             markeredgecolor="black",
156             markeredgewidth=1,
157             linewidth=linewidth,
158             linestyle="--",
159             alpha=alpha,
160         )
161
162     ax_extremum.set_xlim(-0.3, month_dataset.index[0].days_in_month + 0.3)
163     ax_extremum.set_ylim(-300, 700)
164     locator_x_major = plt.MultipleLocator(2) # 设置主刻度
165     ax_extremum.xaxis.set_major_locator(locator_x_major)
166     locator_y_major = plt.MultipleLocator(100) # 设置主刻度
167     ax_extremum.yaxis.set_major_locator(locator_y_major)

```

```

168 ax_extremum.set_xlabel("Elapsed time (day)")
169 ax_extremum.set_ylabel("ORP (mV)")
170 ax_extremum.set_title(
171     "{date}".format(date=month_dataset.index[0].strftime("%Y-%m"))
172 )
173 ax_extremum.tick_params(axis="both", which="major", direction="inout")
174
175 tmp_x = 0.7
176 tmp_y = 20 # 标注的偏移量
177 for i, value in enumerate(max_ORP):
178     ax_extremum.text(day_li[i] - tmp_x, value + tmp_y, s=value)
179
180 for i, value in enumerate(min_ORP):
181     ax_extremum.text(day_li[i] - tmp_x, value + tmp_y, s=value)
182
183 # toc: 保存图片
184 fig_1_path = os.path.join(
185     dir_path, "Curve_{year}_{month}.pdf".format(year=year, month=month)
186 )
187 fig_1.savefig(fig_1_path)
188
189 fig_2_path = os.path.join(
190     dir_path, "Min_max_{year}_{month}.pdf".format(year=year, month=month)
191 )
192 fig_2.savefig(fig_2_path)

```