main

April 18, 2024

[1]:
```
import pandas as pd
import os
import requests

from datetime import date
from dateutil.relativedelta import relativedelta
```

[2]:
```
# Suppress DtypeWarning
import warnings
warnings.filterwarnings("ignore")
```

[3]:
```
# Root of project
root_dir = os.path.join("analysis_data")
os.makedirs(root_dir, exist_ok=True)

# Raw data from api
data_dir = os.path.join(root_dir, "data")
os.makedirs(data_dir, exist_ok=True)

# Formatted CSV data
csv_dir = os.path.join(root_dir, "csv")
os.makedirs(csv_dir, exist_ok=True)

# Data split by location
location_dir = os.path.join(root_dir, "location")
os.makedirs(location_dir, exist_ok=True)
```

[4]:
```
def get_data(start: str, end: str):
    data = requests.get(
        f"https://ilm2.site.dustmonitoring.nl/download?
 ↪from={start}&to={end}&interval=600&align=1&type=csv-semicolon&p=531&p=521&p=542&p=543&p=553
    )
    return data.text

_date = date(2020, 11, 1)

while True:
    start_date = _date
    end_date = _date + relativedelta(months=2)
    data = get_data(start_date.strftime("%Y-%m-%d"), end_date.
↪strftime("%Y-%m-%d"))

    file_path = os.path.join(data_dir, f"data_{start_date.strftime('%Y-%m-%d')}.
↪csv")

    with open(file_path, "+w") as file:
        file.write(data)

    _date = end_date
    if _date > date(2024, 3, 1):
        break
```

[5]:
```
dfs = []
for file in os.listdir(data_dir):

    file_path = os.path.join(data_dir, file)

    if not os.path.isfile(file_path):
        continue
    df = pd.read_csv(file_path, index_col=False, sep=";")

    # Get the values of the first 2 rows
    header_string = df.iloc[:2].values
    row_1 = [row.split(".")[0] for row in df.columns.tolist()]
    row_2 = header_string[0]

    # Merge these into the new column names
    new_columns = []
    for row1, row2 in zip(row_1, row_2):
        row1 = row1.replace("Unnamed: ", "")
        new_columns.append(f"{row1}-{row2}")
    # Remove the used rows
    df = df.iloc[2:]
    # Set new column names
    df.columns = new_columns

    csv_file_path = os.path.join(csv_dir, file)
    df.to_csv(csv_file_path, index=False, index_label=False)
```

[6]:
```
# Get all monthly datasets
dfs = []
for file in os.listdir(csv_dir):
    csv_file_path = os.path.join(csv_dir, file)
    df = pd.read_csv(csv_file_path, index_col=False)
    dfs.append(df)
```

```
# Join datasets togather into one
df = pd.concat(dfs, ignore_index=True)
df.shape
```

[6]: (184964, 287)

[7]:
```
for location in range(1, 60):
    location = str(location)
    if len(location) < 2:
        location = '0' + location

    location_columns = ['0-Tijd', '1-Tijd']
    for column in df.columns:
        if location in column:
            location_columns.append(column)

    if len(location_columns) > 2:
        df_temp = df[location_columns]
        location_csv_path = os.path.join(location_dir, f"I{location}.csv")
        df_temp.to_csv(location_csv_path, index=False, index_label=False)
```

[8]:
```
results = []
for location in range(1, 60):
    location = str(location)
    if len(location) < 2:
        location = '0' + location
    location_csv_path = os.path.join(location_dir, f"I{location}.csv")
    if not os.path.isfile(location_csv_path):
        continue
    df = pd.read_csv(location_csv_path)
    df = df.iloc[df[df.columns[2]].first_valid_index():]
    df = df.iloc[:df[df.columns[2]].last_valid_index()]

    total = df.shape[0]
    missing = df[df.columns[2]].isnull().sum()
    perc = round(missing/total*100, 2)

    obj = {
        "location": location,
        "start": df["0-Tijd"][df["0-Tijd"].first_valid_index()],
        "end": df["0-Tijd"][df["0-Tijd"].last_valid_index()],
        "total rows": total,
        "missing rows": missing,
        "percentage missing": f"{round(perc, 2)} %"
    }
    results.append(obj)
```

```
[9]: df_res = pd.DataFrame.from_records(results)
     df_res
```

```
[9]:    location                start                  end  total rows  \
     0        01  2022-06-08 00:10:00  2024-04-18 08:10:00       99564
     1        02  2020-11-01 00:00:00  2024-04-18 08:00:00      184963
     2        03  2021-03-10 18:10:00  2024-04-18 08:10:00      165989
     3        04  2021-01-07 16:20:00  2024-04-18 08:10:00      175073
     4        05  2021-03-25 12:00:00  2024-04-18 08:10:00      163868
     5        06  2022-11-05 00:00:00  2024-04-18 08:10:00       77530
     6        07  2021-01-11 16:10:00  2024-04-18 08:10:00      174498
     7        08  2021-01-25 16:50:00  2024-04-18 08:10:00      172478
     8        09  2020-12-11 16:00:00  2024-04-18 08:10:00      179108
     9        10  2020-11-01 00:00:00  2024-04-18 08:00:00      184963
     10       11  2020-12-21 16:00:00  2024-04-18 08:10:00      177668
     11       12  2020-12-11 16:00:00  2024-04-18 08:10:00      179108
     12       13  2021-06-07 20:50:00  2024-04-18 08:10:00      153014
     13       14  2021-01-20 16:40:00  2024-04-18 08:10:00      173199
     14       16  2021-03-22 07:00:00  2024-04-18 08:10:00      164330
     15       17  2021-01-11 16:10:00  2024-04-18 08:10:00      174498
     16       18  2021-05-31 08:00:00  2024-04-18 08:10:00      154099
     17       19  2021-01-07 09:00:00  2024-04-18 08:10:00      175117
     18       22  2021-01-11 16:30:00  2024-04-18 08:10:00      174496
     19       23  2021-01-25 12:00:00  2024-04-18 08:10:00      172507
     20       24  2020-12-21 16:00:00  2024-04-18 08:10:00      177668
     21       25  2020-11-01 00:00:00  2024-04-18 07:50:00      184962
     22       28  2021-02-05 12:10:00  2024-04-18 08:10:00      170922
     23       29  2020-12-16 12:30:00  2024-04-18 08:10:00      178409
     24       30  2021-01-07 16:20:00  2024-04-18 08:10:00      175073
     25       32  2020-12-16 16:00:00  2024-04-18 08:10:00      178388
     26       33  2021-03-15 18:20:00  2024-04-18 08:10:00      165270
     27       36  2020-11-01 00:00:00  2024-04-18 08:00:00      184963
     28       37  2020-11-01 00:00:00  2024-04-18 08:00:00      184963
     29       39  2020-11-01 00:10:00  2024-04-18 08:00:00      184962
     30       40  2020-11-01 00:10:00  2024-04-18 08:10:00      184963
     31       41  2021-03-16 18:20:00  2024-04-18 08:10:00      165126
     32       42  2021-06-07 20:30:00  2024-04-18 08:10:00      153016
     33       43  2021-03-16 18:20:00  2024-04-18 08:10:00      165126
     34       44  2021-06-08 03:40:00  2024-04-18 08:10:00      152973
     35       45  2021-05-31 08:00:00  2024-04-18 08:10:00      154099
     36       46  2021-08-06 20:00:00  2024-04-18 08:10:00      144234
     37       47  2021-08-06 19:50:00  2024-04-18 08:10:00      144235
     38       48  2021-06-09 09:00:00  2024-04-18 08:10:00      152797
     39       49  2021-06-09 09:00:00  2024-04-18 08:10:00      152797
     40       50  2022-01-24 09:00:00  2024-04-18 08:10:00      119241
     41       51  2021-06-09 09:00:00  2024-04-18 08:10:00      152797
     42       52  2021-06-09 09:00:00  2024-04-18 08:10:00      152797
```

```
43       55  2021-11-25 16:20:00  2024-04-18 08:10:00      127982
44       56  2021-08-05 20:00:00  2024-04-18 08:10:00      144378
45       58  2021-08-06 20:00:00  2024-04-18 08:10:00      144234
46       59  2024-02-01 00:00:00  2024-04-18 08:10:00       11283

    missing rows percentage missing
0          17992            18.07 %
1          35646            19.27 %
2          21267            12.81 %
3          35770            20.43 %
4          33199            20.26 %
5          29508            38.06 %
6          24548            14.07 %
7          78749            45.66 %
8          33697            18.81 %
9          35030            18.94 %
10         24838            13.98 %
11         35026            19.56 %
12         30317            19.81 %
13         32034             18.5 %
14         30558             18.6 %
15         40219            23.05 %
16         21594            14.01 %
17         32094            18.33 %
18         45906            26.31 %
19         32818            19.02 %
20         31007            17.45 %
21         35876             19.4 %
22         47469            27.77 %
23         28033            15.71 %
24         36556            20.88 %
25         19337            10.84 %
26         27120            16.41 %
27         40487            21.89 %
28         33832            18.29 %
29         42586            23.02 %
30         34309            18.55 %
31         31578            19.12 %
32         22961            15.01 %
33         29955            18.14 %
34         21426            14.01 %
35         30884            20.04 %
36         34444            23.88 %
37         26428            18.32 %
38         33083            21.65 %
39         25002            16.36 %
40         27776            23.29 %
```

```
41         24404             15.97 %
42         29584             19.36 %
43         28342             22.15 %
44         17895             12.39 %
45         13055              9.05 %
46          2432             21.55 %
```