

МОДЕЛЬ СОЗДАНИЯ АРХИВА РАСЧЁТНЫХ ПОГОДНЫХ ДАННЫХ для конкретной локации по данным нескольких референсных метеостанций

Цель: Воссоздать архив погодных явлений и в первую очередь высоты снегового покрова для локации Агробиостанции МГУ в пос. Чашниково Солнечногорского р-она Московской области

==== Тетрадь 1: EDA общего массива данных. ===

0. Подготовка данных

0.0. Импорт необходимых библиотек и настройки представления

```
In [1]: # Python interpreter version: 3.9.12
# Системная конфигурация
import sys

# Работа с файловой системой
from os import listdir
from os.path import isfile, join
from os import makedirs

# Вывод данных
from pprint import pprint
from io import StringIO

# Вычисления
from math import degrees, radians, cos, sin, asin, atan, sqrt, floor, log
import numpy as np # v. 1.22.3
import pandas as pd # v. 1.4.4
```

```
from scipy.stats import norm # v.1.9.1
from scipy.spatial.distance import pdist # v.1.9.1

# Работа с временем и датами
import datetime as dt
import time

# Работа со строками
import re
import pymorphy2 # v. 0.9.1

# Машинное обучение
#...#


# Построение визуализаций
import matplotlib as mpl # v. 3.5.2
import matplotlib.pyplot as plt # v. 3.5.2
import matplotlib.lines as mlines # v. 3.5.2
import seaborn as sns # v. 0.12.0
# import seaborn.objects as so
import missingno as msno # v. 0.4.2

# EDA tools
import sweetviz as sv
#from pandas_profiling import ProfileReport

# Настройки
import warnings
```

Настроим отображение вывода результатов кода в нескольких ячейках

```
In [2]: from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

Для удобства отображения данных изменим опции максимум отображаемых строк и столбцов.

```
In [3]: pd.set_option('display.max_rows', 400) # изменим максимум отображаемых строк
pd.set_option('display.max_columns', 50) # изменим максимум отображаемых столбцов
```

Установим для Seaborn настройки темы по умолчанию.

```
In [4]: sns.set_theme()
```

0.1. Импорт данных и создание датафреймов

```
In [5]: # Построение списка файлов
path = 'data/csv/' # путь к данным
list_csv_files = [file_name for file_name in.listdir(path) if.isfile(join(path, file_name))]
list_csv_files
```

```
Out[5]: ['26393.11.07.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '26499.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27316.11.07.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27402.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27417.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27419.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27502.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27509.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27511.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27515.26.09.2007.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27611.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 '27618.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv',
 'df_anomalies.csv',
 'df_stations.csv',
 'df_stations_lin_coords.csv',
 'df_station_bearings.csv',
 'df_station_dists.csv',
 'Meteo_stations.csv',
 'result_meteo_stations.csv']
```

0.1.1. Данные о метеостанциях

0.1.1.1. Справочная информация о метеостанциях и их архивах

```
In [6]: # Создадим словарь метеостанций по их номерам
dict_stations = {'26393': 'V_Volochev',
                  '26499': 'Staritsa',
                  '27316': 'Kashyn',
                  '27402': 'Tver',
                  '27417': 'Klin',
```

```
'27419': 'Dmitrov',
'27502': 'Volokolamsk',
'27509': 'Mozhaisk',
'27511': 'N_Jerusalem',
'27515': 'Nemchinovka',
'27611': 'Naro_Fominsk',
'27618': 'Serpukhov'
}
```

```
In [7]: # Загружаем файл с данными по метеостанциям для проекта
file_path = path + 'Meteo_stations.csv'
df_stations = pd.read_csv(file_path,
                           sep = ';',
                           header = 0,
                           index_col = None)
```

Добавим в полученный датафрейм информацию о дате начала и дате окончания архива, а также данные о соответствующем файле.

```
# Определим функцию для поиска значений в списке строк по начальным символам строки
def station_search_in_file_list(name_, list_files=list_csv_files):
    """ Функция поиска файла в списке файлов архивов метеостанций по номеру метеостанции
    на вводе - номер метеостанции, на выходе строка названия файла соответствующего архива"""
    result_ = '' # результирующая строка названия файла
    name_ = str(name_) # строковой эквивалент ID метеостанции
    # Ищем строку, начинающуюся с name_
    for string_ in list_files:
        if string_[0:len(name_)] == name_: # сравниваем первые символы по количеству, равные name_
            result_ = string_
            break
    return result_
```

```
# Отсортируем DF по идентификатору метеостанции
df_stations.sort_values(by = "station_ID", inplace = True, ignore_index = True)

### IMPORTANT! df["column"] = value is reported to be DEPRICATED soon ###

# Создадим колонку с названием файла метеостанции по её ID
df_stations = df_stations.assign(
    archive_file = df_stations.station_ID.apply(station_search_in_file_list)
)
# Создадим колонку с датой начала архива и сразу преобразуем её в формат datetime
```

```
df_stations = df_stations.assign(
    beg_date = pd.to_datetime(df_stations.archive_file.apply(lambda x: (x[6:16])),dayfirst=True)
)
# Создадим колонку с датой окончания архива
df_stations = df_stations.assign(
    end_date = pd.to_datetime(df_stations.archive_file.apply(lambda x: (x[17:27])),dayfirst=True)
)

df_stations
```

Out[9]:

	station_name	station_ID	height	latitude	longitude	degree_lo	minute_lo	lo	degree_la	minute_la	la	comment
0	Вышний Волочек	26393	161	N57.583333	E34.566667	57	35	N	34	34	E	validate 26393.11.07.2005.09.06.2022.
1	Старица	26499	185	N56.500000	E34.933333	56	30	N	34	56	E	validate 26499.01.02.2005.09.06.2022.
2	Кашин	27316	138	N57.350000	E37.583333	57	21	N	37	35	E	validate 27316.11.07.2005.09.06.2022.
3	Тверь	27402	132	N56.857300	E35.922000	56	51,438	N	35	55,32	E	validate 27402.01.02.2005.09.06.2022.
4	Клин	27417	167	N56.333333	E36.716667	56	20	N	36	43	E	working 27417.01.02.2005.09.06.2022.
5	Дмитров	27419	179	N56.366667	E37.533333	56	22	N	37	32	E	working 27419.01.02.2005.09.06.2022.
6	Волоколамск	27502	198	N56.016700	E35.933333	56	1,002	N	35	55,998	E	validate 27502.01.02.2005.09.06.2022.
7	Можайск	27509	185	N55.516700	E36.000000	55	31,002	N	36	0	E	learn 27509.01.02.2005.09.06.2022.
8	Ново-Иерусалим	27511	162	N55.900000	E36.816667	55	54	N	36	49	E	working 27511.01.02.2005.09.06.2022.
9	Немчиновка	27515	175	N55.716667	E37.350000	55	43	N	37	21	E	learn 27515.26.09.2007.09.06.2022.
10	Наро-Фоминск	27611	193	N55.383333	E36.700000	55	23	N	36	42	E	learn 27611.01.02.2005.09.06.2022.
11	Серпухов	27618	156	N54.916667	E37.416667	54	55	N	37	25	E	learn 27618.01.02.2005.09.06.2022.

Для удобства ассоциации данных каждой метеостанции с её собственными параметрами, добавим в df_stations информацию о соответствующем каждой станции DF.

In [10]:

```
# Создаём столбец name_of_df по 'df_' + значение по ключу в словаре dict_stations.
# Ключ равен строке из значений df_stations.station_ID
```

```

df_stations = df_stations.assign(
    name_of_df = df_stations.station_ID.astype('str').apply(
        lambda x: f"df_{dict_stations[str(x)]}"
    )
)
df_stations.sample(5)

```

Out[10]:

		station_name	station_ID	height	latitude	longitude	degree_lo	minute_lo	lo	degree_la	minute_la	la	comment
11		Серпухов	27618	156	N54.916667	E37.416667	54	55	N	37	25	E	learn 27618.01.02.2005.09.06.2022.1.
10		Наро-Фоминск	27611	193	N55.383333	E36.700000	55	23	N	36	42	E	learn 27611.01.02.2005.09.06.2022.1.
6		Волоколамск	27502	198	N56.016700	E35.933333	56	1,002	N	35	55,998	E	validate 27502.01.02.2005.09.06.2022.1.
4		Клин	27417	167	N56.333333	E36.716667	56	20	N	36	43	E	working 27417.01.02.2005.09.06.2022.1.
2		Кашин	27316	138	N57.350000	E37.583333	57	21	N	37	35	E	validate 27316.11.07.2005.09.06.2022.1.

Сохраним полученный датафрейм с данными о метеостанциями в csv файл

In [11]: `df_stations.to_csv(path_or_buf=path + '/df_stations.csv')`

0.1.1.2. Расстояния между метеостанциями

Для последующих вычислений создадим DF с расстояниями между станциями. С этой целью определим соответствующую функцию.

In [12]:

```

# Функция определения расстояний в км между точками по координатам
# from math import degrees, radians, cos, sin, asin, atan, sqrt, floor
def distance_km(La1_, La2_, Lo1_, Lo2_):
    """ Принимает координаты точек в градусах
    Возвращает расстояние между точками в км"""
    R_km_ = 6372.795 # радиус Земли
    # Конвертируем градусы в радианы:
    Lo1_ = radians(Lo1_)
    Lo2_ = radians(Lo2_)

```

```

La1_ = radians(La1_)
La2_ = radians(La2_)
# Применим формулу гаверсинуса:
D_Lo_ = Lo2_ - Lo1_ # Угловое расстояние по долготе
D_La_ = La2_ - La1_ # Угловое расстояние по широте
P_ = sin(D_La_ / 2)**2 + cos(La1_) * cos(La2_) * sin(D_Lo_ / 2)**2 # гаверсинус
dist_ = 2 * R_km_ * asin(sqrt(P_))
return(dist_)

```

Создаём датафрейм с расстояниями между станциями, а также Чашниково и условной точкой посередине:

```

In [13]: # Создадим отдельный DF с расстояниями между метеостанциями на основе df_stations
# Выберем столбцы:
df_station_dists = df_stations[["station_ID", "name_of_df", "latitude", "longitude", "height"]]
# Изменим наименование столбцов:
df_station_dists = df_station_dists.rename(columns = {"name_of_df": 'station', "latitude": 'LaN', "longitude": 'LoE'})
# Преобразуем значения:
df_station_dists.station = df_station_dists.station.apply(lambda x: x[3:]) # уберём 'df_'
df_station_dists.LaN = df_station_dists.LaN.apply(lambda x: float(x[1:])) # уберём признак широты 'N', сделаем 'float'
df_station_dists.LoE = df_station_dists.LoE.apply(lambda x: float(x[1:])) # уберём признак долготы 'E', сделаем 'float'
# Если бы мы имели точки в разных полусферах, то S и W надо было бы поставить с отрицательным знаком.

# Добавим сразу координаты Чашниково в df_station_dists
df_chashki = pd.DataFrame({'station_ID': 90000,
                           'station': 'Chashnikovo',
                           'LaN': 56.036999,
                           'LoE': 37.167467,
                           'height': 215
                           },
                           index =[12])
df_station_dists = pd.concat([df_station_dists, df_chashki])

# Добавим сразу координаты реперной центральной точки, находящейся в середине диапазона координат, в df_station_dists
# Её высоту возьмём из Google Earth
df_refer = pd.DataFrame({'station_ID': 99999,
                          'station': 'Rfrnce_point',
                          'LaN': df_station_dists.LaN.min() + (df_station_dists.LaN.max() - df_station_dists.LaN.min()) / 2,
                          'LoE': df_station_dists.LoE.min() + (df_station_dists.LoE.max() - df_station_dists.LoE.min()) / 2,
                          'height': 138
                          },
                          index =[13])
df_station_dists = pd.concat([df_station_dists, df_refer])

```

```

df_station_dists.sample(2)

# ВАЖНО!
# создадим копию df_station_dists - для последующего расчёта начальных азимутов
df_station_bearings = df_station_dists.copy(deep = True)

```

Out[13]:

	station_ID	station	LaN	LoE	height
0	26393	V_Volochev	57.583333	34.566667	161
10	27611	Naro_Fominsk	55.383333	36.700000	193

In [14]:

```

dict_distances_tmp = {} # Определим временный словарь, в котором будем собирать значения расстояний по столбцам
for name_st in df_station_dists.station: # по названиям станций в поле "stations" DF
    # Создаём ключ, равный названию текущей станции
    # и заполняем словарь по этому ключу списком расстояний от данной станции до других станций
    dict_distances_tmp[name_st] = df_station_dists.apply(
        (lambda x: distance_km(La1_ = df_station_dists[df_station_dists.station == name_st].LaN,
                                La2_ = x.LaN,
                                Lo1_ = df_station_dists[df_station_dists.station == name_st].LoE,
                                Lo2_ = x.LoE)),
        axis = 1).tolist() # Аргументы берутся из столбцов для каждой строки, результат записывается в список
df_tmp = pd.DataFrame(dict_distances_tmp) # Создаём временный DF из словаря

df_station_dists = pd.concat([df_station_dists, df_tmp], axis = 1) # Присоединяем df_tmp справа к df_station_dists
print("Расстояния между метеостанциями по большому кругу земной поверхности")
df_station_dists

del df_tmp, dict_distances_tmp # освободить память и предотвратить случайный вызов

```

Расстояния между метеостанциями по большому кругу земной поверхности

Out[14]:

	station_ID	station	LaN	LoE	height	V_Volocheok	Staritsa	Kashyn	Tver	Klin	Dmitrov	Volokolamsk	Moz
0	26393	V_Volocheok	57.583333	34.566667	161	0.000000	122.520236	182.287149	114.810932	190.589931	225.030989	193.100035	246.01
1	26499	Staritsa	56.500000	34.933333	185	122.520236	0.000000	186.562842	72.307124	111.270810	160.570705	81.891761	127.90
2	27316	Kashyn	57.350000	37.583333	138	182.287149	186.562842	0.000000	114.342862	124.765073	109.414691	179.302568	225.9!
3	27402	Tver	56.857300	35.922000	132	114.810932	72.307124	114.342862	0.000000	75.922631	112.714003	93.499387	149.18
4	27417	Klin	56.333333	36.716667	167	190.589931	111.270810	124.765073	75.922631	0.000000	50.469249	59.937380	101.2
5	27419	Dmitrov	56.366667	37.533333	179	225.030989	160.570705	109.414691	112.714003	50.469249	0.000000	106.393956	134.38
6	27502	Volokolamsk	56.016700	35.933333	198	193.100035	81.891761	179.302568	93.499387	59.937380	106.393956	0.000000	55.76
7	27509	Mozhaisk	55.516700	36.000000	185	246.074620	127.906874	225.951727	149.188028	101.215623	134.382110	55.769350	0.00
8	27511	N_Jerusalem	55.900000	36.816667	162	232.119134	134.280536	167.959040	119.885989	48.595243	68.317470	56.510342	66.60
9	27515	Nemchinovka	55.716667	37.350000	175	268.417700	173.356534	182.232070	154.482653	79.082620	73.188783	94.502129	87.66
10	27611	Naro_Fominsk	55.383333	36.700000	193	277.535596	165.931029	225.407227	170.890378	105.670064	121.101803	85.274593	46.58
11	27618	Serpukhov	54.916667	37.416667	156	344.879120	234.983905	270.847365	235.110978	163.585235	161.444192	153.997809	111.9!
12	90000	Chashnikovo	56.036999	37.167467	215	233.757462	147.277363	148.231732	119.104592	43.185421	43.092164	76.738211	93.11
13	99999	Rfrnce point	56.250000	36.075000	138	174.287843	75.614832	152.988800	68.195735	40.677898	90.907994	27.394472	81.69

Сохраним полученный датафрейм с расстояниями между метеостанциями в csv файл

In [15]: df_station_dists.to_csv(path_or_buf=path + '/df_station_dists.csv')

0.1.1.3. Начальные азимуты между метеостанциями

Для последующих вычислений создадим DF с начальными азимутами направлений между станциями. С этой целью определим соответствующую функцию.

In [16]:

```
# Функция определения начального азимута от точки к другой точке по координатам
# (угол между направлением на истинный Северный полюс и направлением на объект, отсчитанный по часовой стрелке)
# (для движения по большому кругу, по мере движения постоянно меняется)
# from math import degrees, radians, cos, sin, asin, atan, sqrt, floor
```

```

def init_bearing(La1_, La2_, Lo1_, Lo2_):
    """ Принимает координаты точек как широта/долгота в градусах
    Возвращает начальный азимут в десятичных градусах, если координаты точек равны, возвращает 0 """
    #pi - число pi, R_m_ - радиус сферы (Земли) в метрах
    R_m_ = 6372795
    # Конвертируем градусы в радианы:
    Lo1_ = radians(Lo1_)
    Lo2_ = radians(Lo2_)
    La1_ = radians(La1_)
    La2_ = radians(La2_)
    # Проверим не совпадают ли две точки и не являются ли они антиподами
    if Lo1_ != Lo2_ and La1_ != La2_:
        #косинусы и синусы широт:
        cl1_ = cos(La1_)
        cl2_ = cos(La2_)
        sl1_ = sin(La1_)
        sl2_ = sin(La2_)
        # разница долгот, её косинус и синус:
        delta_ = Lo2_ - Lo1_
        cdelta_ = cos(delta_)
        sdelta_ = sin(delta_)
        #вычисление начального азимута:
        x_ = (cl1_ * sl2_) - (sl1_ * cl2_ * cdelta_)
        y_ = sdelta_ * cl2_
        z_ = degrees(atan(-y_ / x_))

        if (x_ < 0): # Если полученный угол меньше 0 перевернём его на 180 градусов и сделаем положительным
            z_ = z_ + 180.

        z2_ = (z_ + 180.) % 360. - 180.
        z2_ = - radians(z2_)
        angle_radian_ = z2_ - ((2 * np.pi) * floor((z2_ / (2 * np.pi)))) # Искомый угол в радианах
    #     angle_degree = (angle_radian * 180.) / np.pi # Искомый угол в градусах
        angle_degree_ = degrees(angle_radian_) # Искомый угол в градусах
        return angle_degree_
    else:
        return 0

```

Создаём датафрейм с начальными азимутами между станциями, а также Чашниково и условной точкой посередине:

In [17]: # Создадим отдельный DF с начальными азимутами, продолжая df_station_bearings (определен выше)

```

# ВАЖНО! В отличие от расстояний, начальные азимуты не являются взаимообратными,
# если df расстояний был создан добавлением столбцов, то здесь нужно будет добавлять строки
# используем для этого построчную запись в массив питру

# Определим временный список, в котором будем собирать значения столбцов будущего DF:
list_bearings_tmp = []
# Создадим временный нулевой массив для данных размерностью 1 строка
# и количеством столбцов равном высоте df_station_bearings:
arr_bearings_tmp = np.zeros([1, df_station_bearings.shape[0]])

for name_st in df_station_bearings.station: # по названиям станций в поле "stations" DF
    list_bearings_tmp.append(name_st) # заполняем список для столбцов будущего DF
    # заполняем временный массив для каждой станции
    arr_tmp = np.array(df_station_bearings.apply(
        (lambda x: init_bearing(La1_ = x.LaN,
                                La2_ = df_station_bearings[df_station_bearings.station == name_st].LaN,
                                Lo1_ = x.LoE,
                                Lo2_ = df_station_bearings[df_station_bearings.station == name_st].LoE)),
        axis = 1).tolist())) # Аргументы берутся из столбцов для каждой строки, результат записывается в список

    arr_bearings_tmp = np.vstack((arr_bearings_tmp, arr_tmp)) # присоединяем временный массив построчно к arr_bearings_tmp

arr_bearings_tmp = np.delete(arr_bearings_tmp, 0, 0) # удаляем первую нулевую строку

# Создаём временный DF из массива и списка названий столбцов:
df_tmp = pd.DataFrame(arr_bearings_tmp, columns = list_bearings_tmp)

# Присоединяем df_tmp справа к df_station_bearings:
df_station_bearings = pd.concat([df_station_bearings, df_tmp], axis = 1)
print("Начальные азимуты между станциями по большому кругу земной поверхности (не являются взаимообратными)")
df_station_bearings

del df_tmp, arr_bearings_tmp, list_bearings_tmp # освободить память и предотвратить случайный вызов

```

Начальные азимуты между станциями по большому кругу земной поверхности (не являются взаимообратными)

Out[17]:

	station_ID	station	LaN	LoE	height	V_Volochev	Staritsa	Kashyn	Tver	Klin	Dmitrov	Volokolamsk	Moz
0	26393	V_Volochev	57.583333	34.566667	161	0.000000	349.720491	279.454332	315.263759	317.734920	308.198351	335.037630	339.61
1	26499	Staritsa	56.500000	34.933333	185	169.412820	0.000000	240.670335	237.073671	280.332008	276.381371	311.441730	329.20
2	27316	Kashyn	57.350000	37.583333	138	96.910870	58.449570	0.000000	60.667694	24.635481	1.571425	33.518222	24.86
3	27402	Tver	56.857300	35.922000	132	134.124213	56.247528	242.062682	0.000000	320.469870	299.627158	359.577667	358.11
4	27417	Klin	56.333333	36.716667	167	135.932460	98.846306	205.361054	139.806470	0.000000	266.127102	53.690151	25.88
5	27419	Dmitrov	56.366667	37.533333	179	125.710695	94.214824	181.613292	118.281716	85.447275	0.000000	67.876580	44.66
6	27502	Volokolamsk	56.016700	35.933333	198	153.893929	130.610165	214.897163	179.568223	234.340905	249.206059	0.000000	355.73
7	27509	Mozhaisk	55.516700	36.000000	185	158.478235	148.321106	206.187472	178.113085	206.480305	225.930747	175.682766	0.00
8	27511	N_Jerusalem	55.900000	36.816667	162	142.812105	119.014379	196.538695	152.266302	172.627367	220.854713	102.912734	49.86
9	27515	Nemchinovka	55.716667	37.350000	175	139.488364	119.161738	184.601878	144.610058	149.883997	189.029496	110.091084	74.74
10	27611	Naro_Fominsk	55.383333	36.700000	193	150.939601	137.723586	194.338862	163.280482	180.571050	205.773120	145.383840	108.28
11	27618	Serpukhov	54.916667	37.416667	156	148.106909	137.502971	182.255208	156.017214	164.122373	182.648169	141.992217	126.00
12	90000	Chashnikovo	56.036999	37.167467	215	136.269919	109.534182	190.040333	139.477086	139.561447	211.841192	87.802356	51.12
13	99999	Rfrnce point	56.250000	36.075000	138	147.670570	111.099939	217.533255	172.030849	257.096039	262.400729	18.636341	3.21

ВАЖНО:

- Начальные азимуты определены от строки с названием начальной метеостанции, находящейся в столбце `station`, к столцу с названием искомой метеостанции.
- Обратные начальные азимуты определены от столбца с названием начальной метеостанции к строке, находящейся в столбце `station`, с названием искомой метеостанции
- Начальные и обратные азимуты расчитаны по Большому кругу сфера земной поверхности и не являются взаимообратными, то есть отличаются друг от друга на число градусов неравное 180

Сохраним полученный датафрейм с данными о начальных азимутах между метеостанциями в csv файл

In [18]: `df_station_bearings.to_csv(path_or_buf=path + '/df_station_bearings.csv')`

0.1.1.4. Линейные координаты метеостанций (в км от минимальных долготы и широты)

```
In [19]: # Выберем из df_station_bearings первые 5 столбцов  
df_stations_lin_coords = df_station_bearings.iloc[:, :5]  
df_stations_lin_coords
```

Out[19]:

	station_ID	station	LaN	LoE	height
0	26393	V_Volochev	57.583333	34.566667	161
1	26499	Staritsa	56.500000	34.933333	185
2	27316	Kashyn	57.350000	37.583333	138
3	27402	Tver	56.857300	35.922000	132
4	27417	Klin	56.333333	36.716667	167
5	27419	Dmitrov	56.366667	37.533333	179
6	27502	Volokolamsk	56.016700	35.933333	198
7	27509	Mozhaisk	55.516700	36.000000	185
8	27511	N_Jerusalem	55.900000	36.816667	162
9	27515	Nemchinovka	55.716667	37.350000	175
10	27611	Naro_Fominsk	55.383333	36.700000	193
11	27618	Serpukhov	54.916667	37.416667	156
12	90000	Chashnikovo	56.036999	37.167467	215
13	99999	Rfrnce_point	56.250000	36.075000	138

```
In [20]: # Найдем минимальные координаты в градусах  
LaN_min = df_stations_lin_coords.LaN.min()  
LoE_min = df_stations_lin_coords.LoE.min()  
# Установим точку начала координат юго-западнее минимальных координат в поле метеостанций,  
# для этого уменьшим их на 0,1 градуса  
# LaN_min -= 0.1  
# LoE_min -= 0.1
```

```
In [21]: # По формуле нахождения расстояний между точками вычислим расстояния по осям Юг-Север и Запад-Восток
```

```
# Запишем их в DF
df_stations_lin_coords = \
df_stations_lin_coords.assign(
    lin_ver=df_stations_lin_coords.LaN.apply(
        lambda x: distance_km(LaN_min, x, LoE_min, LoE_min)
    ),
    lin_hor=df_stations_lin_coords.LoE.apply(
        lambda x: distance_km(LaN_min, LaN_min, LoE_min, x)
    )
)
df_stations_lin_coords
```

	station_ID	station	LaN	LoE	height	lin_ver	lin_hor
0	26393	V_Volochev	57.583333	34.566667	161	296.603273	0.000000
1	26499	Staritsa	56.500000	34.933333	185	176.108200	23.440640
2	27316	Kashyn	57.350000	37.583333	138	270.650517	192.838148
3	27402	Tver	56.857300	35.922000	132	215.849342	86.644011
4	27417	Klin	56.333333	36.716667	167	157.570454	137.442395
5	27419	Dmitrov	56.366667	37.533333	179	161.278070	189.642436
6	27502	Volokolamsk	56.016700	35.933333	198	122.352551	87.368487
7	27509	Mozhaisk	55.516700	36.000000	185	66.739424	91.630242
8	27511	N_Jerusalem	55.900000	36.816667	162	109.372447	143.834526
9	27515	Nemchinovka	55.716667	37.350000	175	88.981004	177.924555
10	27611	Naro_Fominsk	55.383333	36.700000	193	51.905512	136.377011
11	27618	Serpukhov	54.916667	37.416667	156	0.000000	182.185661
12	90000	Chashnikovo	56.036999	37.167467	215	124.610333	166.257522
13	99999	Rfrnce_point	56.250000	36.075000	138	148.301637	96.424668

```
In [22]: # Сохраним полученный датафрейм в файл
```

```
df_stations_lin_coords.to_csv(path_or_buf=path + '/df_stations_lin_coords.csv')
```

0.1.2. Данные архивов погоды

0.1.2.1. Чтение и преобразование данных из csv

При просмотре файлов архивов выявлена следующие особенности их структуры:

- каждый файл имеет заголовок размером в 6 строк;
- 7-я строка содержит название полей;
- данные начинаются с 8-й строки.

ВАЖНО

1. В директории `data/csv/` не должно быть посторонних файлов.
2. В директории `data/csv/` для каждой метеостанции должен быть только один файл архива, названный в формате `rp5.ru`.
3. В случае обновления архивов их файлы следует здесь перезаписывать, обязательно сохраняя ID метеостанции вначале и через точки дату начала и дату окончания архива.

```
In [23]: # Определим функцию чтения данных из csv файла формата rp5.ru
def csv_to_dfs(file_path):
    """ По заданной строке file_path, читаем файл csv и создаём df """
    df_ = pd.read_csv(file_path,
                      sep = ';',
                      header = 6,
                      parse_dates = True,
                      dayfirst = True,
                      index_col = None,
                      usecols = list(range(0, 29)), # Иначе некорректно "подцепляется" еще одна колонка
                      low_memory = False # избегаем Warning dtype mixed
                      )
    return df_
```

```
In [24]: # Создадим DF метеостанций из csv файлов в списке
dict_df_data = {} # Инициализируем словарь датафреймов
for file_name in list_csv_files: # По списку csv файлов
    file_name_beg = file_name[:5] # Вычисляем 5 первых символов названия файла
    if file_name_beg in dict_stations.keys(): # Если эти 5 символов совпадают с номером метеостанции
        file_path = path + file_name # Формируем путь к файлу
        print(file_path, ' - ', end='') # КОНТРОЛЬ: Обрабатываемый файл
```

```
# Создаём ключ (название DF) из названия станции соответствующего началу названия файла
# и записываем в словарь по этому ключу соответствующий DF - через вызов функции
dict_df_data[f'df_{dict_stations[file_name_beg]}'] = csv_to_dfs(file_path)
print('O.K.')
# Выводим полученные ключи словаря
dict_df_data.keys()

data/csv/26393.11.07.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/26499.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27316.11.07.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27402.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27417.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27419.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27502.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27509.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27511.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27515.26.09.2007.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27611.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.
data/csv/27618.01.02.2005.09.06.2022.1.0.0.ru.utf8.00000000.csv - O.K.

Out[24]: dict_keys(['df_V_Volochek', 'df_Staritsa', 'df_Kashyn', 'df_Tver', 'df_Klin', 'df_Dmitrov', 'df_Volokolamsk', 'df_Mozhaisk', 'df_N_Jerusalem', 'df_Nemchinovka', 'df_Naro_Fominsk', 'df_Serpukhov'])

В связи с некорректной обработкой данных о дате и времени наблюдений при импорте файлов csv, приведём соответствующие столбцы к формату datetime вручную.
```

```
In [25]: for df_name, df_data in dict_df_data.items():
    print(df_name, 'applying datetime - ', end='') # КОНТРОЛЬ: обрабатываемый DF
    df_data.iloc[:,0] = pd.to_datetime(df_data.iloc[:,0], dayfirst=True) # Преобразуем 1-ю колонку в datetime
    print('O.K.')
```

```
df_V_Volochek applying datetime - O.K.
df_Staritsa applying datetime - O.K.
df_Kashyn applying datetime - O.K.
df_Tver applying datetime - O.K.
df_Klin applying datetime - O.K.
df_Dmitrov applying datetime - O.K.
df_Volokolamsk applying datetime - O.K.
df_Mozhaisk applying datetime - O.K.
df_N_Jerusalem applying datetime - O.K.
df_Nemchinovka applying datetime - O.K.
df_Naro_Fominsk applying datetime - O.K.
df_Serpukhov applying datetime - O.K.
```

0.1.2.2. Вывод полученных DF архивов метеостанций

Для контроля выведем по нескольку строк из каждого полученного датафрейма

```
In [26]: for df_name, df_data in dict_df_data.items():
    df_name
    df_data.head(4)
```

```
Out[26]: 'df_V_Volochek'
```

	Местное время в Вышнем Волочке														Cl	Nh	H		
	T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx				
0	2022-06-09 21:00:00	19.8	743.9	758.5	-0.1	76.0	Ветер, дующий с западо-юго-запада	1.0	NaN	NaN	70 – 80%.	NaN	NaN	NaN	24.6	Слоисто-кучевых, слоистых, кучевых или кучево...	Облаков нет.	2500 или более, или облаков нет.	Высокок высокос. или с
1	2022-06-10 18:00:00	22.3	744.0	758.5	0.0	65.0	Ветер, дующий с запада	1.0	NaN	NaN	90 или более, но не 100%	NaN	NaN	NaN	NaN	Слоисто-кучевые, образовавшиеся не из кучевых.	90 или более, но не 100%	600-1000	Высокок высокос. или с
2	2022-06-09 15:00:00	22.6	744.0	758.5	-0.1	61.0	Ветер, дующий с западо-юго-запада	2.0	NaN	NaN	90 или более, но не 100%	NaN	NaN	NaN	NaN	Слоисто-кучевые, образовавшиеся не из кучевых.	90 или более, но не 100%	600-1000	Высокок высокос. или с
3	2022-06-09 12:00:00	23.3	744.1	758.5	0.4	59.0	Ветер, дующий с западо-северо-запада	3.0	NaN	NaN	90 или более, но не 100%	NaN	NaN	NaN	NaN	Слоисто-кучевые, образовавшиеся не из кучевых.	90 или более, но не 100%	600-1000	Высокок высокос. или с

Out[26]: 'df_Staritsa'

Out[26]:

	Местное время в Старице													CI	Nh	H		
	T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx			
0	2022-06-09 21:00:00	18.4	743.0	759.1	-0.1	92.0	Ветер, дующий с запада	1.0	NaN	11.0	90 или более, но не 100%	Состояние неба в общем не изменилось.	Гроза (грозы) с осадками или без них.	Ливень (ливни).	NaN	27.4	Слоисто-кучевые, образовавшиеся не из кучевых.	70 – 80%. 600-1000
1	2022-06-09 18:00:00	18.1	743.1	759.3	0.6	89.0	Ветер, дующий с северо-северо-востока	2.0	NaN	11.0	100%.	Слабый дождь в срок наблюдения. Гроза в течени...	Гроза (грозы) с осадками или без них.	Ливень (ливни).	NaN	NaN	Кучево-дождевые волокнистые (часто с наковальн...	100%. 300-600
2	2022-06-09 15:00:00	24.8	742.5	758.3	-0.4	45.0	Ветер, дующий с запада	1.0	NaN	NaN	90 или более, но не 100%		NaN	NaN	NaN	NaN	Кучево-дождевые волокнистые (часто с наковальн...	70 – 80%. 600-1000
3	2022-06-09 12:00:00	25.5	742.9	758.8	-0.3	41.0	Ветер, дующий с западо-юго-запада	3.0	NaN	NaN	20–30%.		NaN	NaN	NaN	NaN	Кучевые средние или мощные или вместе с кучевы...	20–30%. 600-1000

Out[26]: 'df_Kashyn'

Out[26]:

	Местное время в Кашине	T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	Cl	Nh	H
0	2022-06-09 21:00:00	17.8	746.5	758.6	-0.1	89.0	Ветер, дующий с юго-запада	1.0	NaN	NaN	70 – 80%.	NaN	NaN	NaN	27.0	Кучево-дождевые лысые с кучевыми, слоисто-куче...	20–30%.	600–1000	Высококучевые просвевающие расположены
1	2022-06-09 18:00:00	20.8	746.6	758.7	0.4	71.0	Ветер, дующий с юго-запада	1.0	NaN	NaN	90 или более, но не 100%	NaN	NaN	NaN	NaN	Кучево-дождевые лысые с кучевыми, слоисто-куче...	20–30%.	600–1000	Высококучевые просвевающие расположены
2	2022-06-09 15:00:00	26.8	746.2	758.1	-1.0	40.0	Ветер, дующий с западо-юго-запада	2.0	NaN	NaN	60%.	NaN	NaN	NaN	NaN	Кучевые средние или мощные или вместе с кучевы...	20–30%.	600–1000	Высококучевые высокослоистые или слоистые дож
3	2022-06-09 12:00:00	25.0	747.2	759.1	-0.6	36.0	Ветер, дующий с юга	2.0	NaN	NaN	20–30%.	NaN	NaN	NaN	NaN	Слоисто-кучевых, слоистых, кучевых или кучево...	Облаков нет.	2500 или более, или облаков нет.	Высококучевые высокослоистые или слоистые дож

Out[26]: 'df_Tver'

Out[26]:

	Местное время в Твери		T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	Cl	Nh
0	2022-06-09 21:00:00	18.8	746.3	759.0	-0.1	92.0	Штиль, безветрие	0.0	NaN	NaN	70 – 80%.	Облаца в целом рассеиваются или становятся мен...	Ливень (ливни).	Облаца покрывают более половины неба в течение...	NaN	28.2	Слоисто-кучевые, образовавшиеся из кучевых.	20–30%.	101
1	2022-06-09 18:00:00	21.8	746.4	758.9	0.6	63.0	Ветер, дующий с западо-северо-запада	1.0	NaN	NaN	90 или более, но не 100%	Состояние неба в общем не изменилось.	Ливень (ливни).	Облаца покрывают более половины неба в течение...	NaN	NaN	Слоисто-кучевые, образовавшиеся из кучевых.	50%.	101
2	2022-06-09 15:00:00	26.5	745.8	758.2	-0.6	34.0	Ветер, дующий с западо-юго-запада	2.0	NaN	NaN	90 или более, но не 100%				NaN	NaN	Кучево-дождевые волокнистые (часто с наковальнями)	60%.	101
3	2022-06-09 12:00:00	26.1	746.4	758.8	-0.4	38.0	Ветер, дующий с юга	1.0	NaN	NaN	60%.				NaN	NaN	Кучевые и слоисто-кучевые (но не слоисто-кучевые)	60%.	101

Out[26]: 'df_Klin'

Out[26]:

	Местное время в Клину		T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	Cl	Nh	H	Crr
0	2022-06-09 09:00:00	22.4	745.4	759.8	-0.2	41.0	Ветер, дующий с юга	1.0	NaN	NaN	20–30%.	NaN	NaN	7.0	NaN	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучевых высокослоистых или слоисто-дожд..		
1	2022-06-09 06:00:00	12.5	745.6	760.4	-0.3	72.0	Ветер, дующий с юго-юго-востока	1.0	NaN	NaN	NaN	NaN	NaN	7.0	NaN	NaN	NaN	NaN	NaN	NaN	
2	2022-06-09 03:00:00	7.0	745.9	761.0	0.0	97.0	Штиль, безветрие	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	2022-06-09 00:00:00	9.4	745.9	760.9	0.3	91.0	Штиль, безветрие	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Out[26]: 'df_Dmitrov'

Out[26]:

	Местное время в Дмитрове																			
	T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	Cl	Nh	H	I	
0	2022-06-09 03:00:00	9.6	744.7	761.0	NaN	77.0	Штиль, безветрие	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1	2022-06-09 00:00:00	12.5	744.6	760.7	NaN	68.0	Ветер, дующий с северо-запада	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2	2022-06-08 21:00:00	16.7	744.5	760.4	NaN	55.0	Ветер, дующий с востока	1.0	NaN	NaN	70 – 80%.	NaN	NaN	NaN	23.8	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучев высокослоист или слоис дож	
3	2022-06-08 18:00:00	23.2	744.4	760.0	NaN	27.0	Ветер, дующий с северо-северо-востока	2.0	NaN	NaN	40%.	NaN	NaN	NaN	NaN	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучев высокослоист или слоис дож	

Out[26]: 'df_Volokolamsk'

Out[26]:

	Местное время в Волоколамске		T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	CI	N
0	2022-06-09 21:00:00	18.7	742.0	759.1	NaN	80.0	Штиль, безветрие	0.0	NaN	NaN	90 или более, но не 100%	Состояние неба в общем не изменилось.	Ливень (ливни).	Облаха покрывали более половины неба в течение...	NaN	28.4	Слоисто-кучевые, образовавшиеся не из кучевых.	50%	
1	2022-06-09 18:00:00	26.7	741.5	758.2	NaN	34.0	Ветер, дующий с западо-северо-запада	1.0	NaN	NaN	70 – 80%.		NaN	NaN	NaN	NaN	Кучевые и слоисто-кучевые (но не слоисто-кучев...)	40%	
2	2022-06-09 15:00:00	28.2	741.8	758.4	NaN	32.0	Ветер, дующий с запада	1.0	NaN	NaN	60%.		NaN	NaN	NaN	NaN	Кучевые и слоисто-кучевые (но не слоисто-кучев...)	40%	
3	2022-06-09 12:00:00	26.7	742.3	759.1	NaN	36.0	Ветер, дующий с западо-северо-запада	1.0	NaN	NaN	70 – 80%.		NaN	NaN	NaN	NaN	Кучевые и слоисто-кучевые (но не слоисто-кучев...)	70 80%	

Out[26]: 'df_Mozhaisk'

Out[26]:

		Местное время в Можайске													T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	Cl	Nh	H
0	2022-06-09 03:00:00	8.5	744.2	761.1	NaN	97.0	Ветер, дующий с юго-юго- запада	1.0	NaN	NaN	Облаков нет.	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN			
1	2022-06-09 00:00:00	11.3	744.4	761.1	NaN	87.0	Штиль, безветрие	0.0	NaN	NaN	70 – 80%.	NaN	NaN	NaN	NaN	Слоисто- кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высокок высокос или с.	или	или	или	или	или	или	или	или	или	или	или	или	
2	2022-06-08 21:00:00	18.0	744.0	760.3	NaN	54.0	Ветер, дующий с севера	1.0	NaN	NaN	70 – 80%.	NaN	NaN	NaN	24.9	Слоисто- кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высокок высокос или с.	или	или	или	или	или	или	или	или	или	или	или	или	
3	2022-06-08 18:00:00	23.7	743.8	759.9	NaN	32.0	Ветер, дующий с северо- востока	1.0	NaN	NaN	50%.	NaN	NaN	NaN	NaN	Слоисто- кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высокок высокос или с.	или	или	или	или	или	или	или	или	или	или	или	или	

Out[26]: 'df_N_Jerusalem'

Out[26]:

	Местное время в Ново-Иерусалиме																CI	Nh	H
	T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx				
0	2022-06-09 09:00:00	22.5	746.0	759.7	-0.2	47.0	Штиль, безветрие	0.0	NaN	NaN	Облаков нет.	NaN	NaN	6.4	NaN	NaN	NaN	NaN	
1	2022-06-09 06:00:00	10.7	746.2	760.5	-0.2	86.0	Штиль, безветрие	0.0	NaN	NaN	Облаков нет.	NaN	NaN	6.4	NaN	NaN	NaN	NaN	
2	2022-06-09 03:00:00	7.2	746.4	760.9	0.1	95.0	Штиль, безветрие	0.0	NaN	NaN	20–30%.	NaN	NaN	NaN	NaN	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков	Высоко или высоко или нет.
3	2022-06-09 00:00:00	10.1	746.3	760.7	0.4	89.0	Штиль, безветрие	0.0	NaN	NaN	20–30%.	NaN	NaN	NaN	NaN	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков	Высоко или высоко или нет.

Out[26]: 'df_Nemchinovka'

Out[26]:

	Местное время в Немчиновке		T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	Cl	Nh	H
0	2022-06-09 03:00:00	10.9	745.0	761.0	NaN	81.0	Штиль, безветрие	0.0	NaN	NaN	Облаков нет.	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	2022-06-09 00:00:00	12.4	745.0	760.9	NaN	73.0	Штиль, безветрие	0.0	NaN	NaN	Облаков нет.	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	2022-06-08 21:00:00	17.5	744.7	760.3	NaN	41.0	Штиль, безветрие	0.0	NaN	NaN	70 – 80%.	NaN	NaN	NaN	25.9	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высоки высоки или нет.	
3	2022-06-08 18:00:00	24.5	744.5	759.8	NaN	22.0	Ветер, дующий с запада	1.0	NaN	NaN	Облаков нет.	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Out[26]: 'df_Naro_Fominsk'

Out[26]:

	Местное время в Наро-Фоминске															T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	Cl	Nh	H	
0	2022-06-09 09:00:00	22.8	743.5	760.0	0.0	47.0	Штиль, безветрие	0.0	NaN	NaN	Облаков нет.	NaN	NaN	7.4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN					
1	2022-06-09 06:00:00	10.3	743.5	760.7	0.0	97.0	Штиль, безветрие	0.0	NaN	NaN	Облаков нет.	NaN	NaN	7.4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN						
2	2022-06-09 03:00:00	7.9	743.5	760.9	0.0	97.0	Штиль, безветрие	0.0	NaN	NaN	20–30%.	NaN	NaN	NaN	NaN	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучесло или слс д	2500 или более, или облаков нет.	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучесло или слс д	2500 или более, или облаков нет.	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучесло или слс д	2500 или более, или облаков нет.	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучесло или слс д
3	2022-06-09 00:00:00	10.9	743.5	760.8	0.3	80.0	Штиль, безветрие	0.0	NaN	NaN	70 – 80%.	NaN	NaN	NaN	NaN	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучесло или слс д	2500 или более, или облаков нет.	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучесло или слс д	2500 или более, или облаков нет.	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучесло или слс д	2500 или более, или облаков нет.	Слоисто-кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высококучесло или слс д

Out[26]: 'df_Serpukhov'

Out[26]:

	Местное время в Серпухове		T	Po	P	Pa	U	DD	Ff	ff10	ff3	N	WW	W1	W2	Tn	Tx	Cl	Nh	H
0	2022-06-09 03:00:00	8.4	745.9	760.9	NaN	91.0	Ветер, дующий с востока	1.0	NaN	NaN	Облаков нет.	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2022-06-09 00:00:00	11.0	745.9	760.6	NaN	86.0	Штиль, безветрие	0.0	NaN	NaN	20-30%.	NaN	NaN	NaN	NaN	Слоисто- кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высокок высокос или с.	
2	2022-06-08 21:00:00	17.1	745.5	760.0	NaN	60.0	Штиль, безветрие	0.0	NaN	NaN	20-30%.	NaN	NaN	NaN	24.3	Слоисто- кучевых, слоистых, кучевых или кучево-...	Облаков нет.	2500 или более, или облаков нет.	Высокок высокос или с.	
3	2022-06-08 18:00:00	23.9	745.4	759.5	NaN	30.0	Ветер, дующий с северо- северо- запада	2.0	NaN	NaN	Облаков нет.	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

ВЫВОД

Импорт данных выполнен корректно

0.2. Code Book

0.2.1. Наименования колонок в полученных архивных данных

- Т Температура воздуха на высоте 2 м над поверхностью земли, градусов Цельсия
- Ро Атмосферное давление на уровне станции, мм рт. столба
- Р Атмосферное давление, приведённое к среднему уровню моря, мм рт. столба

- Pa Барическая тенденция: изменение атмосферного давления за последние 3 часа, мм рт. столба
- Uh Относительная влажность воздуха на высоте 2 м над поверхностью земли, %
- Dd Направление ветра на высоте 10-12 м над поверхностью земли, усреднённое за 10 минут непосредственно перед наблюдением,rumby
- Ff Скорость ветра на высоте 10-12 м над поверхностью земли, усреднённая за 10 минут непосредственно перед наблюдением, м/с
- ff10 Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за 10 минут непосредственно перед наблюдением, м/с
- ff3 Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за период между моментами наблюдения 3 часа, м/с
- N Общая облачность, %
- Ww Текущая погода, сообщаемая метеостанцией
- W1 Прошедшая погода между сроками наблюдения 1
- W2 Прошедшая погода между сроками наблюдения 2
- Tn Минимальная температура воздуха за прошедший период не более 12 часов, градусов Цельсия
- Tx Максимальная температура воздуха за прошедший период не более 12 часов, градусов Цельсия
- Cl Слоисто-кучевые, слоистые, кучевые и кучево-дождевые облака
- Nh Количество всех наблюдающихся облаков Cl, или при отсутствии облаков Cl, количество всех наблюдающихся облаков Cm, %
- H Высота основания самых низких облаков, м
- Cm Высоко-кучевые, высоко-слоистые и слоисто-дождевые облака
- Ch Перистые, перисто-кучевые и перисто-слоистые облака
- Vv Горизонтальная дальность видимости, км.
- Td Температура точки росы на высоте 2 м над поверхностью земли, градусов Цельсия
- Rrr Количество выпавших осадков, мм
- tR Период времени, за который выпало указанное количество осадков, ч.
- E Состояние поверхности почвы без снега или измеримого ледяного покрова
- Tg Минимальная температура поверхности почвы за ночь, градусов Цельсия
- E' Состояние поверхности почвы со снегом или измеримым ледяным покровом
- sss Высота снежного покрова, см

0.2.2. Предлагаемый CodeBook с понятными названиями полей.

Часть из этих полей будет создана в процессе преобразования данных

- T(T) Температура воздуха на высоте 2 м над поверхностью земли, градусов Цельсия
- P_station (Po) Атмосферное давление на уровне станции, мм рт. столба
- P_sea (P) Атмосферное давление, приведённое к среднему уровню моря, мм рт. столба
- P_drift (Pa) Барическая тенденция: изменение атмосферного давления за последние 3 часа, мм рт. столба
- Humid (U) Относительная влажность воздуха на высоте 2 м над поверхностью земли, %
- Wind_dir (DD) Направление ветра на высоте 10-12 м над поверхностью земли, усреднённое за 10 минут непосредственно перед наблюдением, румбы
- Wind_dir360 Направление ветра на высоте 10-12 м над поверхностью земли, усреднённое за 10 минут непосредственно перед наблюдением, градусы
- Wind_dir6k Направление ветра на высоте 10-12 м над поверхностью земли, усреднённое за 10 минут непосредственно перед наблюдением, тысячные
- Wind_speed, Ff Скорость ветра на высоте 10-12 м над поверхностью земли, усреднённая за 10 минут непосредственно перед наблюдением, м/с
- Gusts (ff10) Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за 10 минут непосредственно перед наблюдением, м/с
- Gusts_3h (ff3) Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за период между моментами наблюдения 3 часа, м/с
- Cloudness (N) Общая облачность, %
- Wthr_curr (WW) Текущая погода, сообщаемая метеостанцией
- Prcpttn_like - Первое предложение из Wthr_curr (WW) - информация об осадках и других атмосферных процессах, связанных с переносом жидких и твёрдых частиц
- Prcpttn_depo - вид отложения замерзающих водяных осадков
- Depo_diam_mm - диаметр отложения замерзающих водяных осадков
- Wthr_3h (W1) Прошедшая погода между сроками наблюдения 1
- Wthr_3h2 (W2) Прошедшая погода между сроками наблюдения 2
- T_min (Tn) Минимальная температура воздуха за прошёлый период не более 12 часов, градусов Цельсия
- T_max (Tx) Максимальная температура воздуха за прошёлый период не более 12 часов, градусов Цельсия
- Cl_Cumls (Cl) Слоисто-кучевые, слоистые, кучевые и кучево-дождевые облака

- Cl_viewed (Nh) Количество всех наблюдающихся облаков Cl, или при отсутствии облаков Cl, количество всех наблюдающихся облаков Cm, %
- Cl_bottom (H) Высота основания самых низких облаков, м
- Cl_cumls_hi (Cm) Высоко-кучевые, высоко-слоистые и слоисто-дождевые облака
- Cl_cirrus (Ch) Перистые, перисто-кучевые и перисто-слоистые облака
- Visibility (VV) Горизонтальная дальность видимости, км.
- Dew_point (Td) Температура точки росы на высоте 2 м над поверхностью земли, градусов Цельсия
- Prcptn (RRR) Количество выпавших осадков, мм
- Prcptn_tdelt (tR) Период времени, за который выпало указанное количество осадков, ч.
- Soil (E) Состояние поверхности почвы без снега или измеримого ледяного покрова
- Soil_T (Tg) Минимальная температура поверхности почвы за ночь, градусов Цельсия
- Soil_cover (E') Состояние поверхности почвы со снегом или измеримым ледяным покровом
- Snow_height (sss) Высота снежного покрова, см

0.2.3. Изменение названий полей

```
In [27]: # Определим словарь для замены названий столбцов в DF
dict_field_names = {'T': 'T', 'Po': 'P_station', 'P': 'P_sea', 'Pa': 'P_drift', 'U': 'Humid', 'DD': 'Wind_dir',
'Ff': 'Wind_speed', 'ff10': 'Gusts', 'ff3': 'Gusts_3h', 'N': 'Cloudness', 'WW': 'Wthr_curr',
'W1': 'Wthr_3h', 'W2': 'Wthr_3h2', 'Tn': 'T_min', 'Tx': 'T_max', 'Cl': 'Cl_Cumls',
'Nh': 'Cl_viewd', 'H': 'Cl_bottom', 'Cm': 'Cl_cumls_hi', 'Ch': 'Cl_cirrus', 'VV': 'Visibility',
'Td': 'Dew_point', 'RRR': 'Prcptn', 'tR': 'Prcptn_tdelt', 'E': 'Soil', 'Tg': 'Soil_T',
'E\''': 'Soil_cover', 'sss': 'Snow_height'}
```

```
In [28]: # В цикле по списку датафреймов произведём переименование полей.
for df in dict_df_data.values():
    df.rename(columns = dict_field_names, inplace = True)
```

```
In [29]: # Выведем полученные датафреймы
for df_name, df_data in dict_df_data.items():
    df_name
    df_data.sample(1)
```

```
Out[29]: 'df_V_Volochev'
```

Out[29]:

	Местное время в Вышнем Волочке										Wthr_curr	Wthr_3h	Wthr_3h2	T_mir	
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness					
577	2022-03-29 15:00:00	-3.1	733.7	749.5	0.7	57.0	Ветер, дующий с северо-запада	2.0	NaN	NaN	90 или более, но не 100%	Облака в целом образовывались или развились.	Ливень (ливни).	Облака покрывали более половины неба в течение...	NaN

Out[29]: 'df_Staritsa'

Out[29]:

	Местное время в Старице										T_min	T_max		
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	
50419	2005-02-10 18:00:00	-8.5	755.5	773.7	NaN	45.0	Ветер, дующий с юга	2.0	NaN	NaN	Облаков нет.	NaN	NaN	NaN

Out[29]: 'df_Kashyn'

Out[29]:

	Местное время в Кашине										T_min	T_max		
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	
36789	2009-05-04 09:00:00	14.9	749.8	762.2	NaN	45.0	Ветер, дующий с запада	2.0	NaN	NaN	20-30%.	NaN	NaN	4.0

Out[29]: 'df_Tver'

Out[29]:

Местное время в Твери																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_min	T_max	
4676	2020-11-02 03:00:00	2.1	754.2	767.9	-0.2	89.0	Ветер, дующий с востоко-северо-востока	2.0	NaN	NaN	100%	NaN	NaN	NaN	NaN	NaN

Out[29]: 'df_Klin'

Out[29]:

Местное время в Клину																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_min	T_max	
5427	2020-07-30 12:00:00	20.0	743.5	758.0	-0.3	63.0	Ветер, дующий с запада	3.0	NaN	NaN	90 или более, но не 100%	NaN	NaN	NaN	NaN	NaN

Out[29]: 'df_Dmitrov'

Out[29]:

Местное время в Дмитрове																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_min	T_max	
7417	2019-11-15 12:00:00	3.4	755.9	772.9	NaN	84.0	Ветер, дующий с юга	2.0	NaN	NaN	90 или более, но не 100%	NaN	NaN	NaN	NaN	NaN

Out[29]: 'df_Volokolamsk'

Out[29]:

Местное время в Волоколамске																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_m		
16804	2016-08-26 09:00:00	16.5	749.4	766.8	NaN	77.0	Ветер, дующий с юго-юго-запада	1.0	NaN	NaN	90 или более, но не 100%	NaN	NaN	NaN	7	

Out[29]: 'df_Mozhaisk'

Out[29]:

Местное время в Можайске																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_min	T_max	
6251	2020-04-10 12:00:00	6.8	742.2	759.1	NaN	54.0	Ветер, дующий с северо-запада	6.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	I

Out[29]: 'df_N_Jerusalem'

Out[29]:

Местное время в Ново-Иерусалиме																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_min	T_max	
13078	2017-12-17 06:00:00	7.5	736.0	750.4	-0.1	93.0	Ветер, дующий с юго-юго-востока	5.0	NaN	10.0	100%.	Ливневый(ые) дождь(и) слабый(ые) в срок наблюд...	Ливень (ливни).	Облаца покрывали более половины неба в течение...		

Out[29]: 'df_Nemchinovka'

Out[29]:

Местное время в Немчиновке																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_min	T_max	
18715	2014-11-21 12:00:00	-4.3	760.5	777.8	-0.1	77.0	Ветер, дующий с юго-юго-востока	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Out[29]: 'df_Naro_Fominsk'

Out[29]:

Местное время в Наро-Фоминске																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_min	T_max	
9886	2019-01-20 06:00:00	-6.5	743.0	761.5	-0.2	89.0	Ветер, дующий с юго-запада	3.0	NaN	NaN	90 или более, но не 100%	NaN	NaN	NaN	-8.2	

Out[29]: 'df_Serpukhov'

Out[29]:

Местное время в Серпухове																
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cloudness	Wthr_curr	Wthr_3h	Wthr_3h2	T_min	T_max	
2433	2021-08-05	12.8	748.4	763.1	NaN	91.0	Ветер, дующий с востоко-северо-востока	1.0	NaN	NaN	40.%	NaN	NaN	NaN	NaN	

0.3. Первичный обзор данных

0.3.1. Размеры датафреймов

```
In [30]: # Выведем размеры полученных DFs
for df_name, df_data in dict_df_data.items():
    print(df_name, ':', df_data.shape)
```

```
df_V_Volochev : (41712, 29)
df_Staritsa : (50496, 29)
df_Kashyn : (42289, 29)
df_Tver : (50487, 29)
df_Klin : (50490, 29)
df_Dmitrov : (46867, 29)
df_Volokolamsk : (46933, 29)
df_Mozhaisk : (46936, 29)
df_N_Jerusalem : (50487, 29)
df_Nemchinovka : (28687, 29)
df_Naro_Fominsk : (50445, 29)
df_Serpukhov : (46848, 29)
```

ВЫВОД

Полученные датафреймы одинаковы по количеству полей, но различаются по количеству записей. Известно, что только станция Немчиновка начала фиксировать данные в архиве позже других: с 26 ноября 2007 года. Некоторые станции, начали формировать архивы наблюдений с 11 июля 2005 года. Следовательно датафреймы содержат пропуски в фиксации наблюдений.

Необходимо:

- выявить, какие конкретно наблюдения являются пропущенными, а в каких наличие нулей оправдано;
- определить дату начала архива общую для всех метеостанций, используемых для обучения модели.

0.3.2. Типы и количество данных

```
In [31]: # Выведем информацию о данных в полученных DFs
for df_name, df_data in dict_df_data.items():
    df_name
    df_data.info()
```

```
Out[31]: 'df_V_Volochev'
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41712 entries, 0 to 41711
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Местное время в Вышнем Волочке  41712 non-null   datetime64[ns]
 1   T                             41701 non-null   float64
 2   P_station                     41660 non-null   float64
 3   P_sea                         41666 non-null   float64
 4   P_drift                       27296 non-null   float64
 5   Humid                         41671 non-null   float64
 6   Wind_dir                      41664 non-null   object  
 7   Wind_speed                    41664 non-null   float64
 8   Gusts                         724 non-null    float64
 9   Gusts_3h                      3207 non-null   float64
 10  Cloudness                     41638 non-null   object  
 11  Wthr_curr                     41712 non-null   object  
 12  Wthr_3h                       13832 non-null   object  
 13  Wthr_3h2                      13832 non-null   object  
 14  T_min                         9694 non-null   float64
 15  T_max                         6150 non-null   float64
 16  Cl_Cumls                      34793 non-null   object  
 17  Cl_viewd                      37156 non-null   object  
 18  Cl_bottom                     37184 non-null   object  
 19  Cl_cumls_hi                  29105 non-null   object  
 20  Cl_cirrus                     27909 non-null   object  
 21  Visibility                    41660 non-null   float64
 22  Dew_point                     41684 non-null   float64
 23  Prcptn                        16575 non-null   object  
 24  Prcptn_tdelt                 16575 non-null   float64
 25  Soil                          2213 non-null   object  
 26  Soil_T                        1850 non-null   float64
 27  Soil_cover                    2394 non-null   object  
 28  Snow_height                   2209 non-null   object  
dtypes: datetime64[ns](1), float64(14), object(14)
memory usage: 9.2+ MB
'df_Staritsa'
```

Out[31]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50496 entries, 0 to 50495
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   Местное время в Старице  50496 non-null   datetime64[ns]
 1   T                  50475 non-null   float64
 2   P_station          50407 non-null   float64
 3   P_sea              50441 non-null   float64
 4   P_drift             27284 non-null   float64
 5   Humid              50451 non-null   float64
 6   Wind_dir            50453 non-null   object 
 7   Wind_speed          50454 non-null   float64
 8   Gusts               1488 non-null   float64
 9   Gusts_3h            7174 non-null   float64
 10  Cloudness           50296 non-null   object 
 11  Wthr_curr           50496 non-null   object 
 12  Wthr_3h              19904 non-null   object 
 13  Wthr_3h2             19904 non-null   object 
 14  T_min                9858 non-null   float64
 15  T_max                6314 non-null   float64
 16  Cl_Cumls             37380 non-null   object 
 17  Cl_viewd             42848 non-null   object 
 18  Cl_bottom             42878 non-null   object 
 19  Cl_cumls_hi          32506 non-null   object 
 20  Cl_cirrus             29111 non-null   object 
 21  Visibility            50324 non-null   object 
 22  Dew_point             50473 non-null   float64
 23  Prcptn               16962 non-null   object 
 24  Prcptn_tdelt          16962 non-null   float64
 25  Soil                 2178 non-null   object 
 26  Soil_T                1736 non-null   float64
 27  Soil_cover             2434 non-null   object 
 28  Snow_height            2252 non-null   object 

dtypes: datetime64[ns](1), float64(13), object(15)
memory usage: 11.2+ MB
'df_Kashyn'
```

Out[31]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42289 entries, 0 to 42288
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Местное время в Кашине  42289 non-null   datetime64[ns]
 1   T                  42280 non-null   float64 
 2   P_station          42223 non-null   float64 
 3   P_sea              42246 non-null   float64 
 4   P_drift             27296 non-null   float64 
 5   Humid              42269 non-null   float64 
 6   Wind_dir            42242 non-null   object  
 7   Wind_speed          42242 non-null   float64 
 8   Gusts               703 non-null    float64 
 9   Gusts_3h            3158 non-null   float64 
 10  Cloudness           42211 non-null   object  
 11  Wthr_curr           42289 non-null   object  
 12  Wthr_3h              12291 non-null   object  
 13  Wthr_3h2             12291 non-null   object  
 14  T_min                9694 non-null   float64 
 15  T_max                6145 non-null   float64 
 16  Cl_Cumls             32529 non-null   object  
 17  Cl_viewd             35974 non-null   object  
 18  Cl_bottom             35983 non-null   object  
 19  Cl_cumls_hi          31054 non-null   object  
 20  Cl_cirrus             27361 non-null   object  
 21  Visibility            42234 non-null   float64 
 22  Dew_point             42279 non-null   float64 
 23  Prcptn               16284 non-null   object  
 24  Prcptn_tdelt          16284 non-null   float64 
 25  Soil                 2233 non-null   object  
 26  Soil_T                1844 non-null   float64 
 27  Soil_cover             2472 non-null   object  
 28  Snow_height            2283 non-null   object  
dtypes: datetime64[ns](1), float64(14), object(14)
memory usage: 9.4+ MB
```

Out[31]: 'df_Tver'

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50487 entries, 0 to 50486
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Местное время в Твери  50487 non-null   datetime64[ns]
 1   T                  50464 non-null   float64
 2   P_station          50404 non-null   float64
 3   P_sea              50444 non-null   float64
 4   P_drift             27287 non-null   float64
 5   Humid              50434 non-null   float64
 6   Wind_dir            50434 non-null   object  
 7   Wind_speed          50435 non-null   float64
 8   Gusts               348 non-null    float64
 9   Gusts_3h            2062 non-null   float64
 10  Cloudness           50416 non-null   object  
 11  Wthr_curr           50487 non-null   object  
 12  Wthr_3h              19072 non-null   object  
 13  Wthr_3h2             19072 non-null   object  
 14  T_min                9868 non-null   float64
 15  T_max                6316 non-null   float64
 16  Cl_Cumls             36811 non-null  object  
 17  Cl_viewd             43462 non-null  object  
 18  Cl_bottom             43538 non-null  object  
 19  Cl_cumls_hi          36736 non-null  object  
 20  Cl_cirrus             28471 non-null  object  
 21  Visibility            50441 non-null  object  
 22  Dew_point             50460 non-null   float64
 23  Prcptn               16851 non-null   object  
 24  Prcptn_tdelt          16851 non-null   float64
 25  Soil                 2227 non-null   object  
 26  Soil_T                1848 non-null   float64
 27  Soil_cover             2414 non-null   object  
 28  Snow_height            2246 non-null   object  
dtypes: datetime64[ns](1), float64(13), object(15)
memory usage: 11.2+ MB
'df_Klin'
```

Out[31]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50490 entries, 0 to 50489
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   Местное время в Клину  50490 non-null   datetime64[ns]
 1   T                  50476 non-null   float64
 2   P_station          50424 non-null   float64
 3   P_sea              50455 non-null   float64
 4   P_drift             27308 non-null   float64
 5   Humid              50444 non-null   float64
 6   Wind_dir            50437 non-null   object 
 7   Wind_speed          50437 non-null   float64
 8   Gusts               617 non-null    float64
 9   Gusts_3h            2856 non-null   float64
 10  Cloudness           49894 non-null   object 
 11  Wthr_curr           50490 non-null   object 
 12  Wthr_3h              16655 non-null   object 
 13  Wthr_3h2             16655 non-null   object 
 14  T_min                9855 non-null   float64
 15  T_max                6321 non-null   float64
 16  Cl_Cumls             38127 non-null   object 
 17  Cl_viewd             42724 non-null   object 
 18  Cl_bottom             42741 non-null   object 
 19  Cl_cumls_hi          29799 non-null   object 
 20  Cl_cirrus             24497 non-null   object 
 21  Visibility            49917 non-null   object 
 22  Dew_point             50465 non-null   float64
 23  Prcptn               17820 non-null   object 
 24  Prcptn_tdelt          17820 non-null   float64
 25  Soil                 2681 non-null   object 
 26  Soil_T                2010 non-null   float64
 27  Soil_cover             2407 non-null   object 
 28  Snow_height            2219 non-null   object 

dtypes: datetime64[ns](1), float64(13), object(15)
memory usage: 11.2+ MB
'df_Dmitrov'
```

Out[31]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46867 entries, 0 to 46866
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   Местное время в Дмитрове  46867 non-null   datetime64[ns]
 1   T                  46854 non-null   float64 
 2   P_station          46770 non-null   float64 
 3   P_sea              46827 non-null   float64 
 4   P_drift             7495 non-null   float64 
 5   Humid              46842 non-null   float64 
 6   Wind_dir            46859 non-null   object  
 7   Wind_speed          46861 non-null   float64 
 8   Gusts               1026 non-null   float64 
 9   Gusts_3h            4102 non-null   float64 
 10  Cloudness           45837 non-null   object  
 11  Wthr_curr           46867 non-null   object  
 12  Wthr_3h              15261 non-null   object  
 13  Wthr_3h2             15261 non-null   object  
 14  T_min                8890 non-null   float64 
 15  T_max                5860 non-null   float64 
 16  Cl_Cumls             38407 non-null   object  
 17  Cl_viewd             41561 non-null   object  
 18  Cl_bottom             39486 non-null   object  
 19  Cl_cumls_hi          26069 non-null   object  
 20  Cl_cirrus             26819 non-null   object  
 21  Visibility            43572 non-null   object  
 22  Dew_point             46850 non-null   float64 
 23  Prcptn               15226 non-null   object  
 24  Prcptn_tdelt          15226 non-null   float64 
 25  Soil                 2639 non-null   object  
 26  Soil_T                1454 non-null   float64 
 27  Soil_cover             2373 non-null   object  
 28  Snow_height            2134 non-null   object  
dtypes: datetime64[ns](1), float64(13), object(15)
memory usage: 10.4+ MB
'df_Volokolamsk'
```

Out[31]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46933 entries, 0 to 46932
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Местное время в Волоколамске  46933 non-null   datetime64[ns] 
 1   T                            46924 non-null   float64 
 2   P_station                    44831 non-null   float64 
 3   P_sea                        46905 non-null   float64 
 4   P_drift                      6662  non-null   float64 
 5   Humid                        46917 non-null   float64 
 6   Wind_dir                     46932 non-null   object  
 7   Wind_speed                   46932 non-null   float64 
 8   Gusts                        1236  non-null   float64 
 9   Gusts_3h                     3677  non-null   float64 
 10  Cloudness                    37945 non-null   object  
 11  Wthr_curr                    46933 non-null   object  
 12  Wthr_3h                      17374 non-null   object  
 13  Wthr_3h2                     15467 non-null   object  
 14  T_min                        6747  non-null   float64 
 15  T_max                        4765  non-null   float64 
 16  Cl_Cumls                     30169 non-null   object  
 17  Cl_viewd                     33925 non-null   object  
 18  Cl_bottom                    33923 non-null   object  
 19  Cl_cumls_hi                 23797 non-null   object  
 20  Cl_cirrus                    21739 non-null   object  
 21  Visibility                   46848 non-null   object  
 22  Dew_point                    38003 non-null   float64 
 23  Prcptn                       11541 non-null   object  
 24  Prcptn_tdelt                11541 non-null   float64 
 25  Soil                          2042  non-null   object  
 26  Soil_T                        878   non-null   float64 
 27  Soil_cover                   1914  non-null   object  
 28  Snow_height                  1728  non-null   object  
dtypes: datetime64[ns](1), float64(13), object(15)
memory usage: 10.4+ MB
'df_Mozhaisk'
```

Out[31]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46936 entries, 0 to 46935
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   Местное время в Можайске  46936 non-null   datetime64[ns]
 1   T                  46923 non-null   float64 
 2   P_station          44838 non-null   float64 
 3   P_sea              46904 non-null   float64 
 4   P_drift             6659 non-null   float64 
 5   Humid              46911 non-null   float64 
 6   Wind_dir            46913 non-null   object  
 7   Wind_speed          46914 non-null   float64 
 8   Gusts               515 non-null    float64 
 9   Gusts_3h            2200 non-null   float64 
 10  Cloudness           37950 non-null   object  
 11  Wthr_curr           46936 non-null   object  
 12  Wthr_3h              14080 non-null   object  
 13  Wthr_3h2             13202 non-null   object  
 14  T_min                6728 non-null   float64 
 15  T_max                4769 non-null   float64 
 16  Cl_Cumls             31178 non-null   object  
 17  Cl_viewd             32622 non-null   object  
 18  Cl_bottom             32618 non-null   object  
 19  Cl_cumls_hi          16739 non-null   object  
 20  Cl_cirrus             19778 non-null   object  
 21  Visibility            46845 non-null   float64 
 22  Dew_point             37996 non-null   float64 
 23  Prcptn               11284 non-null   object  
 24  Prcptn_tdelt          11284 non-null   float64 
 25  Soil                 2021 non-null   object  
 26  Soil_T                888 non-null    float64 
 27  Soil_cover             1839 non-null   object  
 28  Snow_height            1613 non-null   object  
dtypes: datetime64[ns](1), float64(14), object(14)
memory usage: 10.4+ MB
'df_N_Jerusalem'
```

Out[31]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50487 entries, 0 to 50486
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   Местное время в Ново-Иерусалиме 50487 non-null   datetime64[ns]
 1   T                           50478 non-null   float64
 2   P_station                   50389 non-null   float64
 3   P_sea                       50432 non-null   float64
 4   P_drift                     27306 non-null   float64
 5   Humid                       50447 non-null   float64
 6   Wind_dir                    50445 non-null   object 
 7   Wind_speed                  50445 non-null   float64
 8   Gusts                       884 non-null    float64
 9   Gusts_3h                   3565 non-null   float64
 10  Cloudness                   50359 non-null   object 
 11  Wthr_curr                   50487 non-null   object 
 12  Wthr_3h                     16104 non-null   object 
 13  Wthr_3h2                   16104 non-null   object 
 14  T_min                       9863 non-null   float64
 15  T_max                       6317 non-null   float64
 16  Cl_Cumls                    39988 non-null  object 
 17  Cl_viewd                   44842 non-null   object 
 18  Cl_bottom                   44860 non-null   object 
 19  Cl_cumls_hi                27023 non-null   object 
 20  Cl_cirrus                  28769 non-null   object 
 21  Visibility                  50386 non-null   object 
 22  Dew_point                   50458 non-null   float64
 23  Prcptn                      17734 non-null   object 
 24  Prcptn_tdelt                17734 non-null   float64
 25  Soil                         2497 non-null   object 
 26  Soil_T                      1852 non-null   float64
 27  Soil_cover                  2488 non-null   object 
 28  Snow_height                 2242 non-null   object 

dtypes: datetime64[ns](1), float64(13), object(15)
memory usage: 11.2+ MB
```

Out[31]: 'df_Nemchinovka'

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28687 entries, 0 to 28686
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Местное время в Немчиновке  28687 non-null   datetime64[ns]
 1   T                           28667 non-null   float64
 2   P_station                   28650 non-null   float64
 3   P_sea                       28647 non-null   float64
 4   P_drift                     5973 non-null   float64
 5   Humid                       28648 non-null   float64
 6   Wind_dir                    28645 non-null   object  
 7   Wind_speed                  28648 non-null   float64
 8   Gusts                       536  non-null    float64
 9   Gusts_3h                   2254 non-null   float64
 10  Cloudness                   25906 non-null   object  
 11  Wthr_curr                   28687 non-null   object  
 12  Wthr_3h                     9123 non-null   object  
 13  Wthr_3h2                   9123 non-null   object  
 14  T_min                       7538 non-null   float64
 15  T_max                       4892 non-null   float64
 16  Cl_Cumls                    20306 non-null   object  
 17  Cl_viewd                   21697 non-null   object  
 18  Cl_bottom                   19868 non-null   object  
 19  Cl_cumls_hi                13060 non-null   object  
 20  Cl_cirrus                  10249 non-null   object  
 21  Visibility                  23671 non-null   float64
 22  Dew_point                   28661 non-null   float64
 23  Prcptn                     12348 non-null   object  
 24  Prcptn_tdelt                12348 non-null   float64
 25  Soil                        2427 non-null   object  
 26  Soil_T                      1434 non-null   float64
 27  Soil_cover                  1735 non-null   object  
 28  Snow_height                 1710 non-null   object  
dtypes: datetime64[ns](1), float64(14), object(14)
memory usage: 6.3+ MB
'df_Naro_Fominsk'
```

Out[31]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50445 entries, 0 to 50444
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Местное время в Наро-Фоминске  50445 non-null   datetime64[ns]
 1   T                            50442 non-null   float64
 2   P_station                    50383 non-null   float64
 3   P_sea                        50425 non-null   float64
 4   P_drift                      27301 non-null   float64
 5   Humid                        50419 non-null   float64
 6   Wind_dir                     50408 non-null   object  
 7   Wind_speed                   50410 non-null   float64
 8   Gusts                        1076 non-null   float64
 9   Gusts_3h                     4148 non-null   float64
 10  Cloudness                    50394 non-null   object  
 11  Wthr_curr                    50445 non-null   object  
 12  Wthr_3h                      20803 non-null   object  
 13  Wthr_3h2                     20803 non-null   object  
 14  T_min                        9846 non-null   float64
 15  T_max                        6312 non-null   float64
 16  Cl_Cumls                     40775 non-null   object  
 17  Cl_viewd                     45561 non-null   object  
 18  Cl_bottom                    45566 non-null   object  
 19  Cl_cumls_hi                 30360 non-null   object  
 20  Cl_cirrus                    28411 non-null   object  
 21  Visibility                   50421 non-null   float64
 22  Dew_point                    50421 non-null   float64
 23  Prcptn                       18106 non-null   object  
 24  Prcptn_tdelt                18106 non-null   float64
 25  Soil                          2501 non-null   object  
 26  Soil_T                        1942 non-null   float64
 27  Soil_cover                   2460 non-null   object  
 28  Snow_height                  2232 non-null   object  
dtypes: datetime64[ns](1), float64(14), object(14)
memory usage: 11.2+ MB
'df_Serpukhov'
```

Out[31]:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46848 entries, 0 to 46847
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Местное время в Серпухове 46848 non-null   datetime64[ns]
 1   T                         46821 non-null   float64
 2   P_station                  46739 non-null   float64
 3   P_sea                      46734 non-null   float64
 4   P_drift                     7480 non-null   float64
 5   Humid                      46675 non-null   float64
 6   Wind_dir                   46826 non-null   object 
 7   Wind_speed                 46827 non-null   float64
 8   Gusts                      1245 non-null   float64
 9   Gusts_3h                   3517 non-null   float64
 10  Cloudness                  46571 non-null   object 
 11  Wthr_curr                  46848 non-null   object 
 12  Wthr_3h                    14143 non-null   object 
 13  Wthr_3h2                  14143 non-null   object 
 14  T_min                      8860 non-null   float64
 15  T_max                      5856 non-null   float64
 16  Cl_Cumls                   35847 non-null   object 
 17  Cl_viewd                   39752 non-null   object 
 18  Cl_bottom                  37802 non-null   object 
 19  Cl_cumls_hi                26695 non-null   object 
 20  Cl_cirrus                  23183 non-null   object 
 21  Visibility                  44333 non-null   object 
 22  Dew_point                  46704 non-null   float64
 23  Prcptn                     14274 non-null   object 
 24  Prcptn_tdelt                14274 non-null   float64
 25  Soil                        2706 non-null   object 
 26  Soil_T                      1487 non-null   float64
 27  Soil_cover                  2141 non-null   object 
 28  Snow_height                 1975 non-null   object 

dtypes: datetime64[ns](1), float64(13), object(15)
memory usage: 10.4+ MB

```

ВЫВОД

- Количество записей о времени наблюдений является максимальным среди других значений для каждого датафрейма, следовательно, даже элементарные параметры, такие как температура и давление, фиксировались не всегда. Следовательно, *все датафреймы содержат пропуски в значениях наблюдений.*

2. При сверке с CodeBook, можно понять, что тип данных у многих значений не соответствует смыслу записей.

- Местное время: object вместо DateTime,
- Cloudness: object вместо float64 (возможно наличие смешанных данных),
- Cl_viewd: object вместо float64 (возможно наличие смешанных данных),
- Prcptn: object вместо float64 (возможно наличие смешанных данных),
- Prcptn_tdelt: float64 - возможно, потребуется изменить на Timedelta,
- Snow_height: object вместо float64 (возможно наличие смешанных данных).

Необходимо :

- привести данные к единобразию внутри столбцов,
- по возможности, формализовать строковые данные, и перевести их в численные значения.

0.3.3. Сводные статистические характеристики данных

```
In [32]: # Выведем основные статистические характеристики данных по каждому датафрейму
for name, data in dict_df_data.items():
    print(f'\nСводная статистика данных датафрейма {name}:')
    # Включить все типы данных, трактовать datetime как numeric, во избежании deprecation warning
    data.describe(include='all', datetime_is_numeric=True)
```

Сводная статистика данных датафрейма df_V_Volochev:

Out[32]:

	Местное время в Вышнем Волочке		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clc
count	41712	41701.000000	41660.000000	41666.000000	27296.000000	41671.000000		41664	41664.000000	724.000000	3207.000000	
unique	Nan	Nan	Nan	Nan	Nan	Nan	Nan	17	Nan	Nan	Nan	Nan
top	Nan	Nan	Nan	Nan	Nan	Nan	NaN	Штиль, безветрие	Nan	Nan	Nan	6с +
freq	Nan	Nan	Nan	Nan	Nan	Nan	NaN	5795	Nan	Nan	Nan	Nan
mean	2015-02-11 23:51:22.422324480	5.596283	745.555365	761.128287	-0.000059	79.723861	Nan	2.002064	11.030387	11.383224		
min	2005-07-11 21:00:00	-37.500000	77.900000	718.600000	-22.300000	13.000000	Nan	0.000000	5.000000	5.000000		
25%	2011-09-17 20:15:00	-1.100000	740.800000	756.200000	-0.500000	70.000000	Nan	1.000000	10.000000	10.000000		
50%	2015-04-19 07:30:00	5.100000	745.700000	761.100000	0.000000	86.000000	Nan	2.000000	10.500000	11.000000		
75%	2018-11-13 15:45:00	13.600000	750.400000	765.900000	0.500000	94.000000	Nan	3.000000	12.000000	12.000000		
max	2022-06-09 21:00:00	37.200000	777.600000	824.600000	21.600000	100.000000	Nan	50.000000	22.000000	23.000000		
std	Nan	10.563597	8.510866	8.122888	0.957556	18.546236	Nan	1.404727	1.522233	1.690126		

Сводная статистика данных датафрейма df_Staritsa:

Out[32]:

	Местное время в Старице		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Cls
count	50496	50475.000000	50407.000000	50441.000000	27284.000000	50451.000000	50453	50454.000000	1488.000000	7174.000000		
unique	Nan	Nan	Nan	Nan	Nan	Nan	Nan	17	Nan	Nan	Nan	Nan
top	Nan	Nan	Nan	Nan	Nan	Nan	Nan	Ветер, дующий с юга	Nan	Nan	Nan	Nan
freq	Nan	Nan	Nan	Nan	Nan	Nan	Nan	5890	Nan	Nan	Nan	Nan
mean	2013-10-13 17:07:29.786121728	5.482403	744.530537	761.645112	-0.001114	77.976988	Nan	2.842411	11.362903	12.218567		
min	2005-02-01 03:00:00	-37.200000	671.000000	684.500000	-13.000000	8.000000	Nan	0.000000	10.000000	5.000000		
25%	2009-06-15 14:15:00	-1.200000	739.900000	756.800000	-0.500000	68.000000	Nan	2.000000	10.000000	11.000000		
50%	2013-10-17 10:30:00	5.200000	744.600000	761.500000	0.000000	84.000000	Nan	3.000000	11.000000	12.000000		
75%	2018-02-12 15:45:00	13.700000	749.200000	766.300000	0.500000	92.000000	Nan	4.000000	12.000000	13.000000		
max	2022-06-09 21:00:00	36.600000	777.400000	795.500000	7.500000	100.000000	Nan	32.000000	22.000000	25.000000		
std	Nan	10.654885	7.618038	7.916523	0.915261	17.905807	Nan	1.741824	1.676218	2.151170		

Сводная статистика данных датафрейма df_Kashyn:

Out[32]:

	Местное время в Кашине		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clc
count	42289	42280.000000	42223.000000	42246.000000	27296.000000	42269.000000	42242	42242.000000	703.000000	3158.000000		
unique	Nan	Nan	Nan	Nan	Nan	Nan	Nan	17	Nan	Nan	Nan	Nan
top	Nan	Nan	Nan	Nan	Nan	Nan	NaN	Штиль, безветрие	Nan	Nan	Nan	6с +
freq	Nan	Nan	Nan	Nan	Nan	Nan	NaN	4739	Nan	Nan	Nan	Nan
mean	2014-12-28 21:59:37.951713024	5.301748	748.530130	761.356135	0.000150	78.836618	Nan	1.961034	10.933144	11.651045		
min	2005-07-11 15:00:00	-35.400000	712.600000	725.000000	-6.500000	12.000000	Nan	0.000000	5.000000	5.000000		
25%	2011-07-24 12:00:00	-1.700000	743.700000	756.400000	-0.500000	69.000000	Nan	1.000000	10.000000	10.000000		
50%	2015-03-14 15:00:00	4.900000	748.600000	761.200000	0.000000	85.000000	Nan	2.000000	11.000000	11.000000		
75%	2018-10-27 00:00:00	13.600000	753.200000	766.100000	0.500000	93.000000	Nan	3.000000	11.000000	12.000000		
max	2022-06-09 21:00:00	36.800000	781.100000	795.600000	6.300000	100.000000	Nan	10.000000	19.000000	25.000000		
std	Nan	10.945837	7.903261	8.127697	0.927874	17.829597	Nan	1.303243	1.342839	1.918307		

Сводная статистика данных датафрейма df_Tver:

Out[32]:

	Местное время в Твери		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clc
count	50487	50464.000000	50404.000000	50444.000000	27287.000000	50434.000000	50434	50435.000000	348.000000	2062.000000		
unique	Nan	Nan	Nan	Nan	Nan	Nan	Nan	17	Nan	Nan	Nan	Nan
top	Nan	Nan	Nan	Nan	Nan	Nan	Nan	Штиль, безветрие	Nan	Nan	Nan	Nan
freq	Nan	Nan	Nan	Nan	Nan	Nan	Nan	7672	Nan	Nan	Nan	Nan
mean	2013-10-13 18:26:23.266979584	5.869973	747.960402	761.383108	-0.001547	76.732760	Nan	1.725865	11.229885	11.296314		
min	2005-02-01 03:00:00	-36.400000	711.000000	723.800000	-10.800000	10.000000	Nan	0.000000	10.000000	5.000000		
25%	2009-06-15 10:30:00	-1.100000	743.200000	756.500000	-0.500000	66.000000	Nan	1.000000	10.000000	10.000000		
50%	2013-10-17 09:00:00	5.500000	748.000000	761.200000	0.000000	83.000000	Nan	2.000000	11.000000	11.000000		
75%	2018-02-12 19:30:00	14.300000	752.700000	766.100000	0.500000	92.000000	Nan	2.000000	12.000000	12.000000		
max	2022-06-09 21:00:00	38.600000	779.500000	794.800000	10.600000	100.000000	Nan	10.000000	16.000000	20.000000		
std	Nan	10.912238	7.770053	8.001672	0.933657	18.881898	Nan	1.313356	1.117853	1.446767		

Сводная статистика данных датафрейма df_Klin:

Out[32]:

	Местное время в Клину		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clc
count	50490	50476.000000	50424.000000	50455.000000	27308.000000	50444.000000	50437	50437.000000	617.000000	2856.000000		
unique	Nan	Nan	Nan	Nan	Nan	Nan	Nan	17	Nan	Nan	Nan	Nan
top	Nan	Nan	Nan	Nan	Nan	Nan	Nan	Штиль, безветрие	Nan	Nan	Nan	Nan
freq	Nan	Nan	Nan	Nan	Nan	Nan	Nan	10061	Nan	Nan	Nan	Nan
mean	2013-10-13 05:11:55.721925376	5.789629	746.206634	761.553134	-0.000652	77.960590	Nan	1.655392	11.011345	11.472339		
min	2005-02-01 03:00:00	-36.500000	710.800000	725.600000	-11.100000	13.000000	Nan	0.000000	5.000000	6.000000		
25%	2009-06-15 03:45:00	-1.200000	741.500000	756.700000	-0.500000	68.000000	Nan	1.000000	10.000000	10.000000		
50%	2013-10-17 01:30:00	5.500000	746.200000	761.300000	0.000000	84.000000	Nan	2.000000	11.000000	11.000000		
75%	2018-02-11 23:15:00	14.100000	750.900000	766.300000	0.500000	92.000000	Nan	2.000000	12.000000	12.000000		
max	2022-06-09 09:00:00	37.600000	777.100000	794.600000	8.200000	100.000000	Nan	9.000000	18.000000	25.000000		
std	Nan	10.988825	7.626360	7.902462	0.911849	18.263933	Nan	1.244286	1.370444	1.707695		

Сводная статистика данных датафрейма df_Dmitrov:

Out[32]:

	Местное время в Дмитрове		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clo
count	46867		46854.000000	46770.000000	46827.000000	7495.000000	46842.000000	46859	46861.000000	1026.000000	4102.000000	
unique	NaN		NaN	NaN	NaN	NaN	NaN	17	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Ветер, дующий с юга	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5007	NaN	NaN	NaN	
mean	2013-08-09 22:39:46.890562816	5.824711	745.068614	761.594949	-0.002588	76.895756	NaN	2.255372	11.384990	11.932960		
min	2005-02-01 03:00:00	-34.500000	710.200000	726.000000	-12.700000	11.000000	NaN	0.000000	10.000000	10.000000		
25%	2009-02-20 04:30:00	-1.500000	740.400000	756.700000	-0.500000	65.000000	NaN	1.000000	10.000000	10.000000		
50%	2013-03-03 15:00:00	5.600000	745.000000	761.400000	0.000000	82.000000	NaN	2.000000	11.000000	12.000000		
75%	2018-05-25 04:30:00	14.400000	749.700000	766.400000	0.500000	92.000000	NaN	3.000000	12.000000	13.000000		
max	2022-06-09 03:00:00	37.500000	776.000000	795.000000	12.800000	100.000000	NaN	15.000000	18.000000	24.000000		
std	NaN	11.003239	7.711554	8.008053	0.926052	18.604332	NaN	1.371241	1.509298	1.849927		

Сводная статистика данных датафрейма df_Volokolamsk:

Out[32]:

	Местное время в Волоколамске		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clouds
count	46933	46924.000000	44831.000000	46905.000000	6662.000000	46917.000000	46932	46932.000000	1236.000000	3677.000000	3	3
unique	Nan	Nan	Nan	Nan	Nan	Nan	Nan	17	Nan	Nan	Nan	Nan
top	Nan	Nan	Nan	Nan	Nan	Nan	Nan	Ветер, дующий с юга	Nan	Nan	Nan	Nan
freq	Nan	Nan	Nan	Nan	Nan	Nan	Nan	7683	Nan	Nan	Nan	Nan
mean	2013-08-13 19:20:19.304114176	5.630675	743.597194	761.73044	-0.001471	78.010764	Nan	2.263956	11.854369	12.334240		
min	2005-02-01 03:00:00	-35.600000	708.000000	725.40000	-9.200000	12.000000	Nan	0.000000	10.000000	10.000000		
25%	2009-02-23 21:00:00	-1.500000	739.000000	756.80000	-0.400000	68.000000	Nan	1.000000	11.000000	11.000000		
50%	2013-03-09 12:00:00	5.300000	743.600000	761.50000	0.000000	84.000000	Nan	2.000000	12.000000	12.000000		
75%	2018-05-25 00:00:00	14.000000	748.100000	766.40000	0.500000	92.000000	Nan	3.000000	12.000000	13.000000		
max	2022-06-09 21:00:00	37.300000	773.500000	794.20000	15.800000	100.000000	Nan	14.000000	20.000000	26.000000		
std	Nan	10.844014	7.516627	7.86646	0.880848	18.020030	Nan	1.477929	1.514782	2.037744		

Сводная статистика данных датафрейма df_Mozhaisk:

Out[32]:

	Местное время в Можайске		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clo
count	46936		46923.000000	44838.000000	46904.000000	6659.000000	46911.000000	46913	46914.000000	515.000000	2200.000000	
unique	NaN		NaN	NaN	NaN	NaN	NaN	17	NaN	NaN	NaN	
top	NaN		NaN	NaN	NaN	NaN	NaN	Штиль, безветрие	NaN	NaN	NaN	
freq	NaN		NaN	NaN	NaN	NaN	NaN	6501	NaN	NaN	NaN	
mean	2013-08-14 04:56:38.278506752	5.887051	745.064664	762.008347	-0.002583	77.206178	NaN	2.151000	11.048544	11.877727		
min	2005-02-01 03:00:00	-35.100000	709.900000	93.800000	-8.000000	12.000000	NaN	0.000000	10.000000	10.000000		
25%	2009-02-23 20:15:00	-1.300000	740.600000	757.300000	-0.400000	66.000000	NaN	1.000000	10.000000	10.000000		
50%	2013-03-10 04:30:00	5.600000	745.000000	761.800000	0.000000	83.000000	NaN	2.000000	11.000000	12.000000		
75%	2018-05-25 21:45:00	14.300000	749.500000	766.600000	0.500000	91.000000	NaN	3.000000	12.000000	13.000000		
max	2022-06-09 03:00:00	37.300000	775.100000	810.800000	7.700000	100.000000	NaN	16.000000	18.000000	24.000000		
std	NaN	10.955799	7.460357	8.390968	0.861024	18.093057	NaN	1.529273	1.373588	1.838909		

Сводная статистика данных датафрейма df_N_Jerusalem:

Out[32]:

	Местное время в Ново- Иерусалиме										
	T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clc	
count	50487	50478.000000	50389.000000	50432.000000	27306.000000	50447.000000	50445	50445.000000	884.000000	3565.000000	
unique	Nan	Nan	Nan	Nan	Nan	Nan	17	Nan	Nan	Nan	
top	Nan	Nan	Nan	Nan	Nan	Штиль, безветрие	Nan	Nan	Nan	Nan	
freq	Nan	Nan	Nan	Nan	Nan	7719	Nan	Nan	Nan	Nan	
mean	2013-10-13 10:19:08.303523584	5.832604	746.952234	761.707626	-0.002618	77.895494	Nan	2.119457	11.082579	11.685554	
min	2005-02-01 03:00:00	-35.400000	711.700000	725.600000	-13.900000	9.000000	Nan	0.000000	10.000000	5.000000	
25%	2009-06-15 10:30:00	-1.300000	742.300000	756.900000	-0.500000	68.000000	Nan	1.000000	10.000000	10.000000	
50%	2013-10-17 12:00:00	5.400000	746.800000	761.500000	0.000000	85.000000	Nan	2.000000	11.000000	11.000000	
75%	2018-02-12 07:30:00	14.200000	751.600000	766.300000	0.500000	92.000000	Nan	3.000000	12.000000	13.000000	
max	2022-06-09 09:00:00	37.400000	777.400000	794.200000	6.700000	100.000000	Nan	20.000000	19.000000	22.000000	
std	Nan	10.942460	7.575733	7.844906	0.916211	18.319623	Nan	1.504745	1.361571	1.796032	

Сводная статистика данных датафрейма df_Nemchinovka:

Out[32]:

	Местное время в Немчиновке		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clouds
count	28687	28667.000000	28650.000000	28647.000000	5973.000000	28648.000000	28645	28648.000000	536.000000	2254.000000	2	2
unique	Nan	Nan	Nan	Nan	Nan	Nan	NaN	17	Nan	Nan	Nan	Nan
top	Nan	Nan	Nan	Nan	Nan	Nan	NaN	Ветер, дующий с юга	Nan	Nan	Nan	Nan
freq	Nan	Nan	Nan	Nan	Nan	Nan	NaN	5057	Nan	Nan	Nan	Nan
mean	2016-08-12 02:36:04.492627456	6.585663	745.613065	762.010029	-0.005843	73.138997	NaN	2.038781	10.934701	11.439219		
min	2007-09-26 15:00:00	-31.000000	710.200000	725.800000	-12.900000	11.000000	NaN	0.000000	10.000000	10.000000		
25%	2013-12-01 01:30:00	-1.000000	740.900000	757.100000	-0.500000	60.000000	NaN	1.000000	10.000000	10.000000		
50%	2017-06-22 15:00:00	5.800000	745.600000	761.800000	0.000000	79.000000	NaN	2.000000	10.000000	11.000000		
75%	2019-12-18 10:30:00	15.200000	750.300000	766.700000	0.500000	89.000000	NaN	3.000000	11.000000	12.000000		
max	2022-06-09 03:00:00	37.900000	774.700000	803.800000	13.600000	100.000000	NaN	18.000000	17.000000	27.000000		
std	Nan	10.751080	7.641843	7.924870	0.947000	19.279221	NaN	1.303493	1.290305	1.777351		

Сводная статистика данных датафрейма df_Naro_Fominsk:

Out[32]:

	Местное время в Наро-Фоминске		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	C
count	50445		50442.000000	50383.000000	50425.000000	27301.000000	50419.000000	50408	50410.000000	1076.000000	4148.000000	
unique	NaN		NaN	NaN	NaN	NaN	NaN	17	NaN	NaN	NaN	
top	NaN		NaN	NaN	NaN	NaN	NaN	Штиль, безветрие	NaN	NaN	NaN	
freq	NaN		NaN	NaN	NaN	NaN	NaN	5273	NaN	NaN	NaN	
mean	2013-10-15 18:47:17.395182848		5.964014	744.172840	761.826878	-0.000370	77.349075	NaN	2.372863	11.192379	11.803520	
min	2005-02-01 03:00:00		-35.400000	709.000000	725.800000	-14.100000	9.000000	NaN	0.000000	6.000000	5.000000	
25%	2009-06-19 06:00:00		-1.300000	739.600000	757.000000	-0.500000	65.000000	NaN	1.000000	10.000000	10.000000	
50%	2013-10-19 21:00:00		5.600000	744.100000	761.500000	0.000000	83.000000	NaN	2.000000	11.000000	12.000000	
75%	2018-02-13 15:00:00		14.400000	748.700000	766.400000	0.500000	93.000000	NaN	3.000000	12.000000	13.000000	
max	2022-06-09 09:00:00		38.000000	773.300000	793.600000	14.700000	100.000000	NaN	12.000000	23.000000	23.000000	
std	NaN		10.869450	7.404233	7.725787	0.899545	19.287209	NaN	1.488899	1.462453	1.805524	

Сводная статистика данных датафрейма df_Serpukhov:

Out[32]:

	Местное время в Серпухове		T	P_station	P_sea	P_drift	Humid	Wind_dir	Wind_speed	Gusts	Gusts_3h	Clc
count	46848	46821.000000	46739.000000	46734.000000	7480.000000	46675.000000	46826	46827.000000	1245.000000	3517.000000		
unique	Nan	Nan	Nan	Nan	Nan	Nan	Nan	17	Nan	Nan	Nan	Nan
top	Nan	Nan	Nan	Nan	Nan	Nan	Nan	Штиль, безветрие	Nan	Nan	Nan	Nan
freq	Nan	Nan	Nan	Nan	Nan	Nan	Nan	5595	Nan	Nan	Nan	Nan
mean	2013-08-11 18:58:54.989753856	6.287476	746.812168	762.002850	-0.002019	76.037086	Nan	2.263779	11.406426	11.664202		
min	2005-02-01 03:00:00	-34.600000	713.200000	727.800000	-14.900000	13.000000	Nan	0.000000	7.000000	10.000000		
25%	2009-02-21 17:15:00	-1.200000	742.300000	757.300000	-0.500000	65.000000	Nan	1.000000	10.000000	10.000000		
50%	2013-03-05 01:30:00	6.100000	746.600000	761.600000	0.000000	82.000000	Nan	2.000000	11.000000	11.000000		
75%	2018-05-28 12:45:00	14.900000	751.300000	766.600000	0.500000	90.000000	Nan	3.000000	12.000000	12.000000		
max	2022-06-09 03:00:00	38.700000	776.300000	793.600000	16.000000	100.000000	Nan	42.000000	18.000000	22.000000		
std	Nan	11.068401	7.448341	7.735104	0.999746	18.141960	Nan	1.473725	1.546791	1.808715		

ВЫВОД В целом, на высоком уровне обобщения, принципиальных различий в распределении данных между метеостанциями не наблюдается. Следовательно, климатические различия между их местами расположения незначительны. Отдельные исключения:

1. Серпухов:

- более высокие (на приблизительно 0,7 градуса) показатели температуры.

1. Немчиновка и Можайск

- ниже, чем на других станциях, мода высоты снежного покрова (но здесь еще не произведена формализация данных).

1. EDA

1.1. Исследование полноты данных и преобразование типов данных

1.1.1. Информация о пропущенных данных

```
In [33]: # Определим функцию для печати сводных данных о количестве метеонаблюдений
def observation_quantities(df_):
    total_obsrvtns = df_.shape[0] # Всего записей о наблюдениях
    years_obsrvd = df_.iloc[:, 0].dt.year.nunique() # количество лет, в которых велись наблюдения
    dates_obsrvd = df_.iloc[:, 0].dt.date.nunique() # количество дней, в которые имеются записи о наблюдениях
    beg_obsrvtns = str(
        df_stations.beg_date[df_stations.name_of_df == name].values[0])[0:10] # дата начала архива, приведенная к строке
    # определена по данным df_stations
    snow_cover_months = df_.Snow_height.notna().sum() / dates_obsrvd * 12

    print(f'Архив метеостанции {name}\n'
          f'Дата начала наблюдений: {beg_obsrvtns}, количество записей: {total_obsrvtns}\n'
          f'Всего, дней наблюдений: {dates_obsrvd}, за {years_obsrvd} лет\n'
          f'Среднее количество наблюдений в день: {((total_obsrvtns / dates_obsrvd):.4f} (идеал = 8.0)\n'
          f'Среднегодовое число месяцев со снежным покровом: {snow_cover_months:.1f}'
    )
```

1.1.1.1 Анализ количества имеющихся данных против пропущенных и нулевых

```
In [34]: print('КОЛИЧЕСТВО ДАННЫХ ПО СТОЛБЦАМ БЕЗ УЧЁТА ПРОПУСКОВ\n');

for name, data in dict_df_data.items(): # по элементам словаря с данными DFs
    observation_quantities(data) # вызовем функцию распечатки сводных данных о количестве наблюдений

    msno.bar(data, color='navy', figsize=(15, 4), fontsize=10); # выводим график из библиотеки "missingno"
    plt.show()
```

КОЛИЧЕСТВО ДАННЫХ ПО СТОЛБЦАМ БЕЗ УЧЁТА ПРОПУСКОВ

Архив метеостанции df_V_Volochek

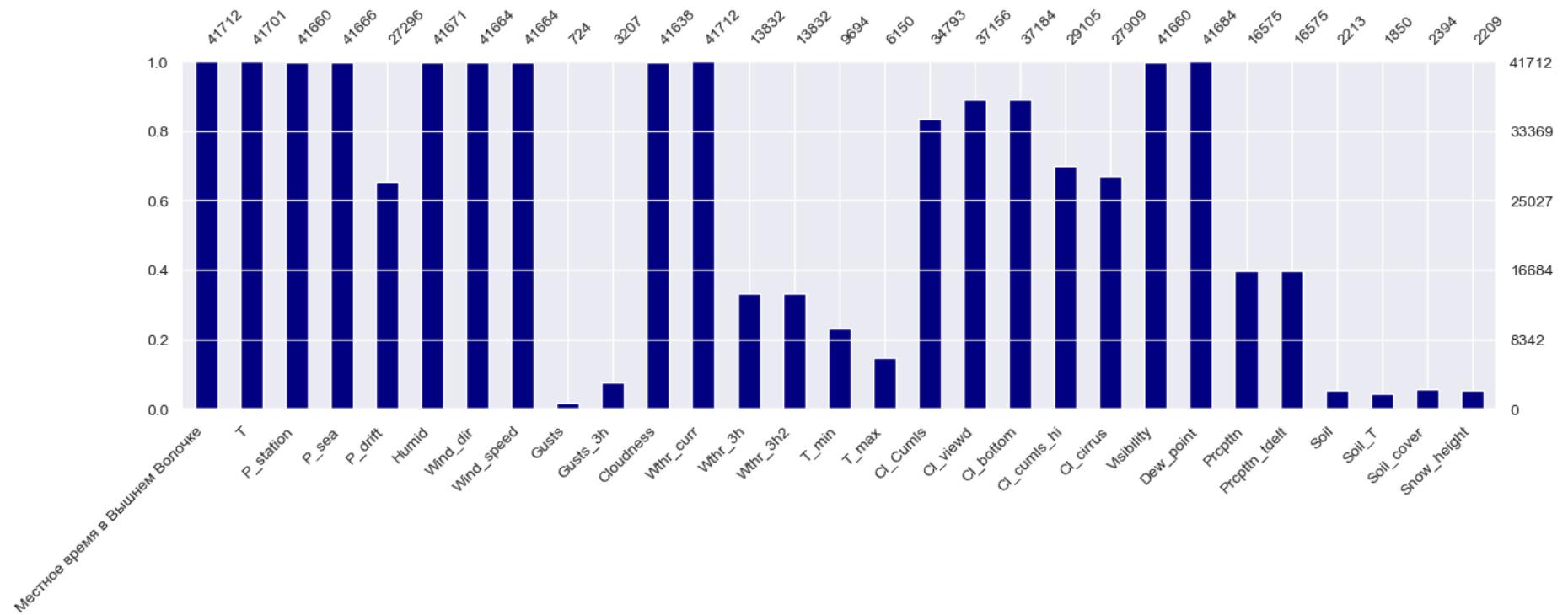
Дата начала наблюдений: 2005-07-11, количество записей: 41712

Всего, дней наблюдений: 6168, за 18 лет

Среднее количество наблюдений в день: 6.7626 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 4.3

Out[34]: <AxesSubplot:>



Архив метеостанции df_Staritsa

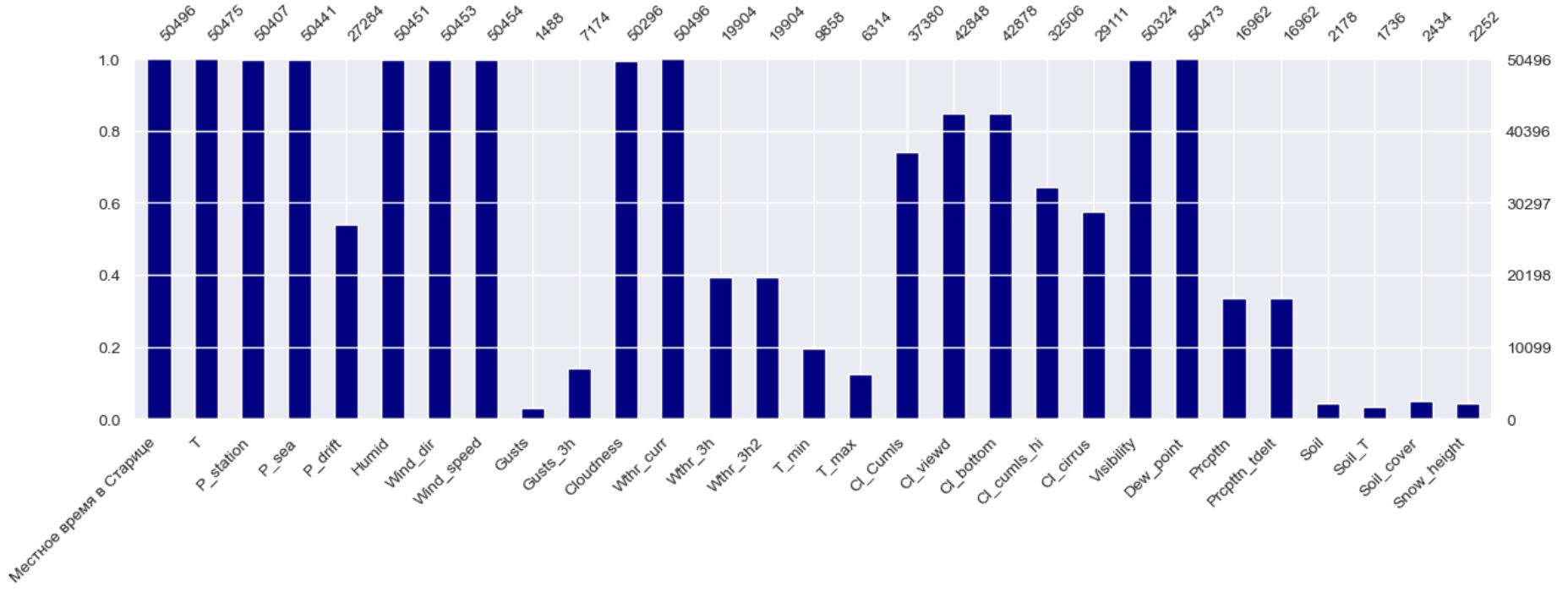
Дата начала наблюдений: 2005-02-01, количество записей: 50496

Всего, дней наблюдений: 6336, за 18 лет

Среднее количество наблюдений в день: 7.9697 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 4.3

Out[34]: <AxesSubplot:>



Архив метеостанции df_Kashyn

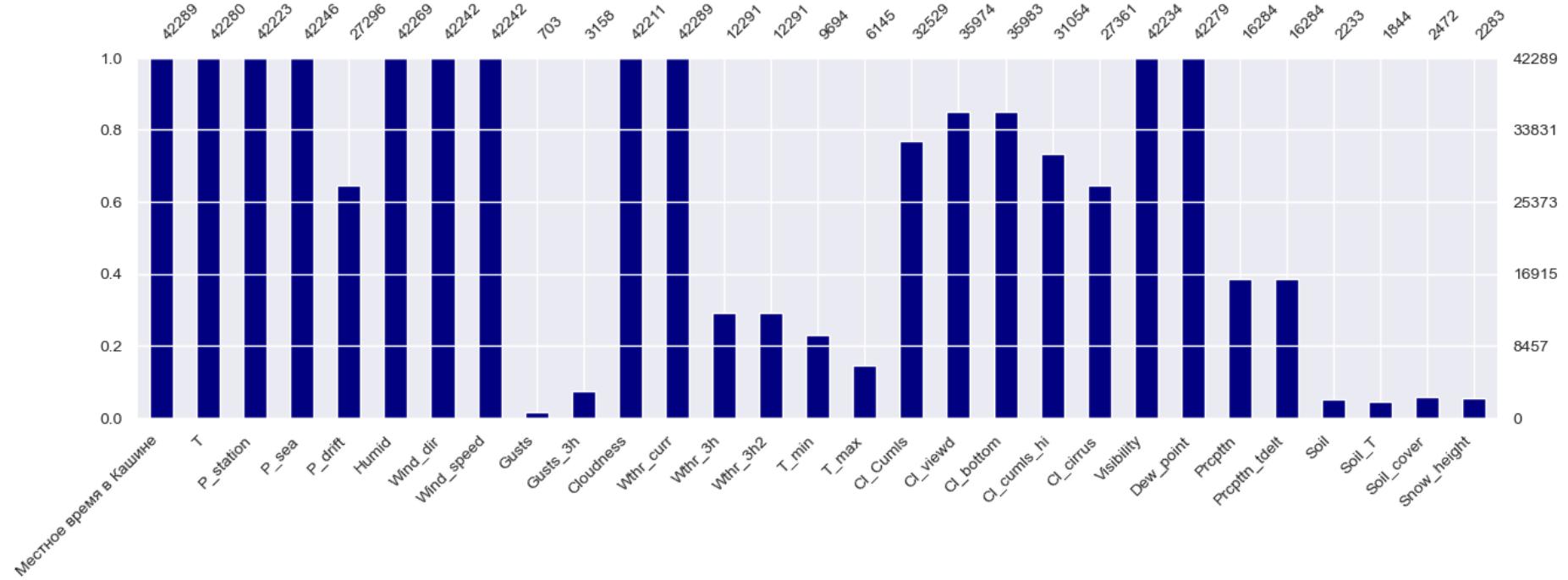
Дата начала наблюдений: 2005-07-11, количество записей: 42289

Всего, дней наблюдений: 6173, за 18 лет

Среднее количество наблюдений в день: 6.8506 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.4

Out[34]:
<AxesSubplot:>



Архив метеостанции df_Tver

Дата начала наблюдений: 2005-02-01, количество записей: 50487

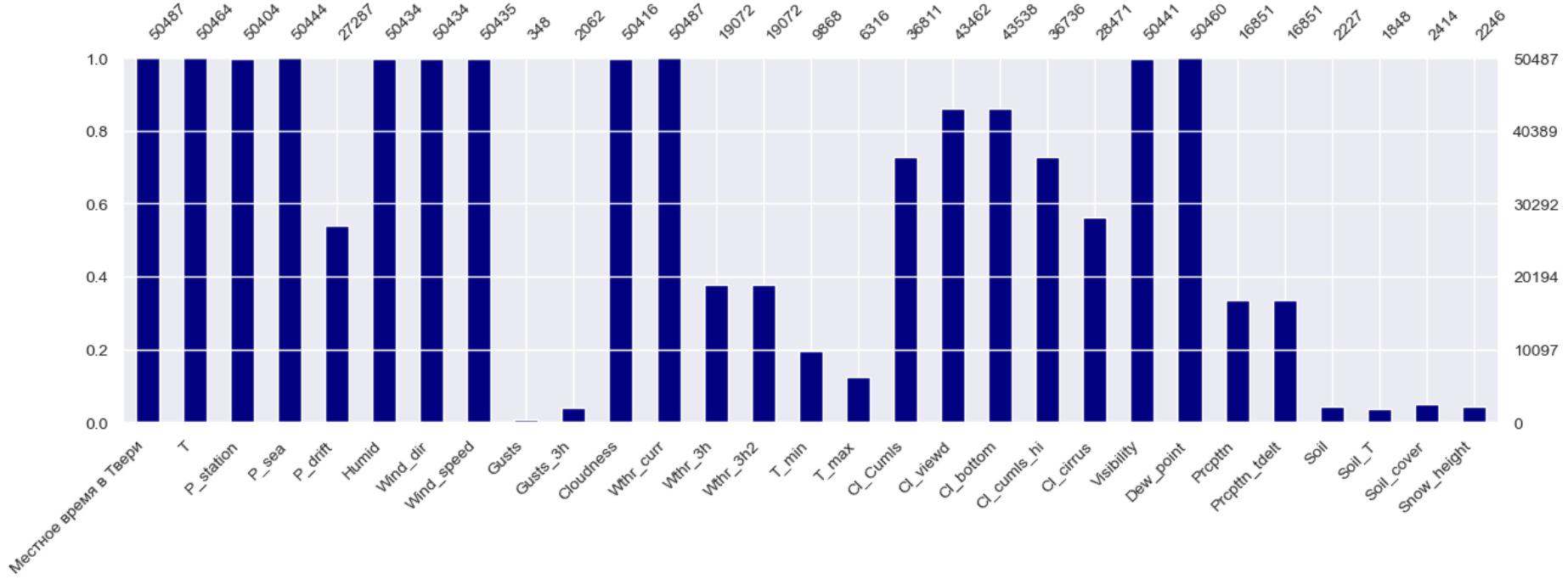
Всего, дней наблюдений: 6336, за 18 лет

Среднее количество наблюдений в день: 7.9683 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 4.3

Out[34]:

<AxesSubplot:>



Архив метеостанции df_Klin

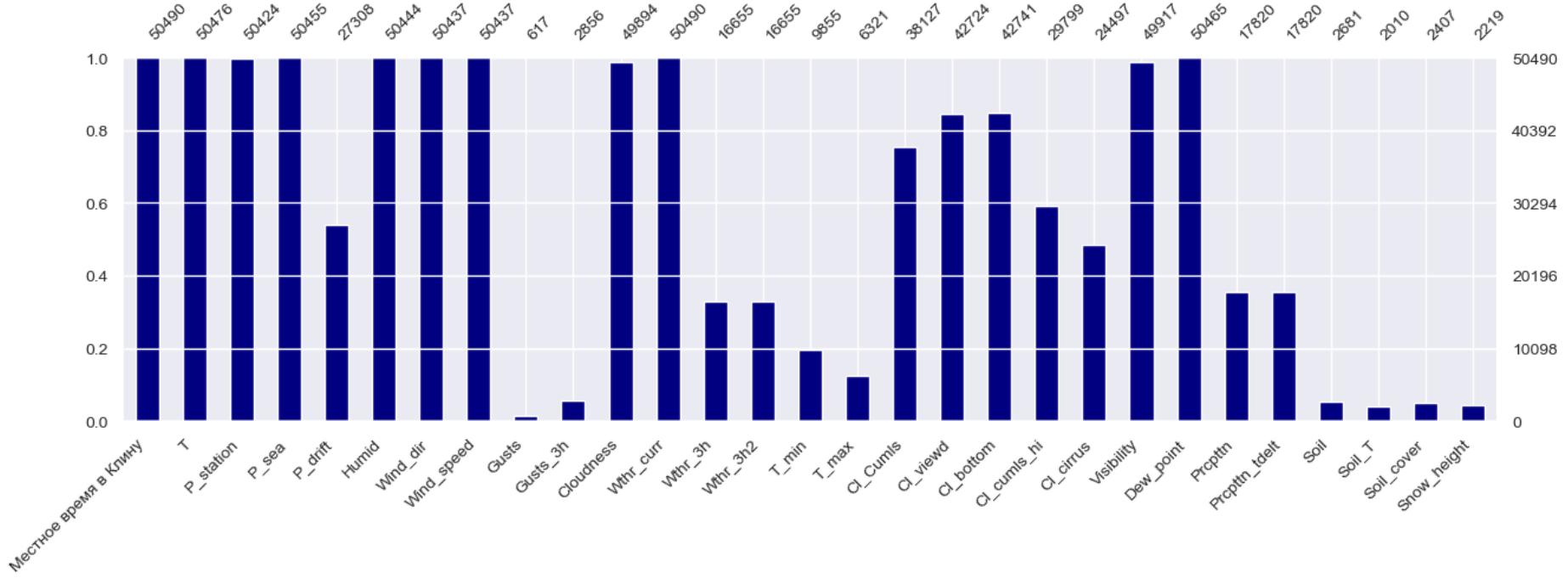
Дата начала наблюдений: 2005-02-01, количество записей: 50490

Всего, дней наблюдений: 6336, за 18 лет

Среднее количество наблюдений в день: 7.9688 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.2

Out[34]: <AxesSubplot:>



Архив метеостанции df_Dmitrov

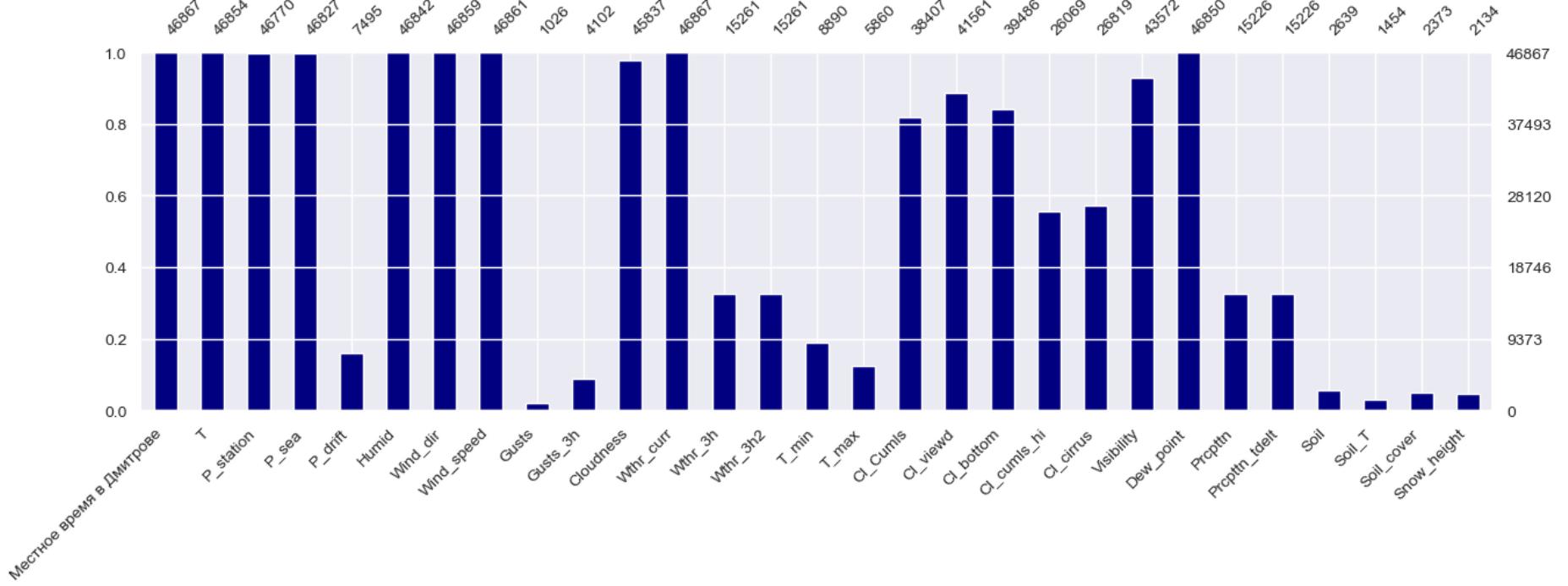
Дата начала наблюдений: 2005-02-01, количество записей: 46867

Всего, дней наблюдений: 5905, за 18 лет

Среднее количество наблюдений в день: 7.9368 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 4.3

Out[34]: <AxesSubplot:>



Архив метеостанции df_Volokolamsk

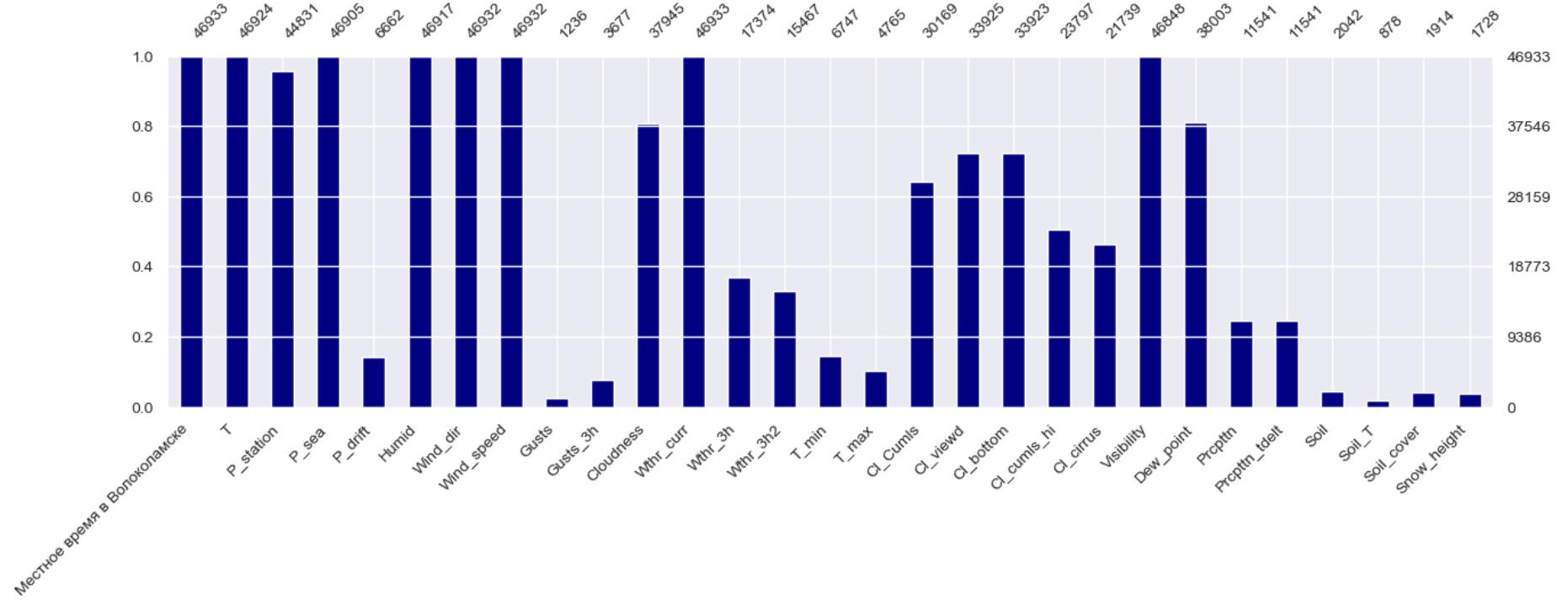
Дата начала наблюдений: 2005-02-01, количество записей: 46933

Всего, дней наблюдений: 5905, за 18 лет

Среднее количество наблюдений в день: 7.9480 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 3.5

Out[34]: <AxesSubplot:>



Архив метеостанции df_Mozhaisk

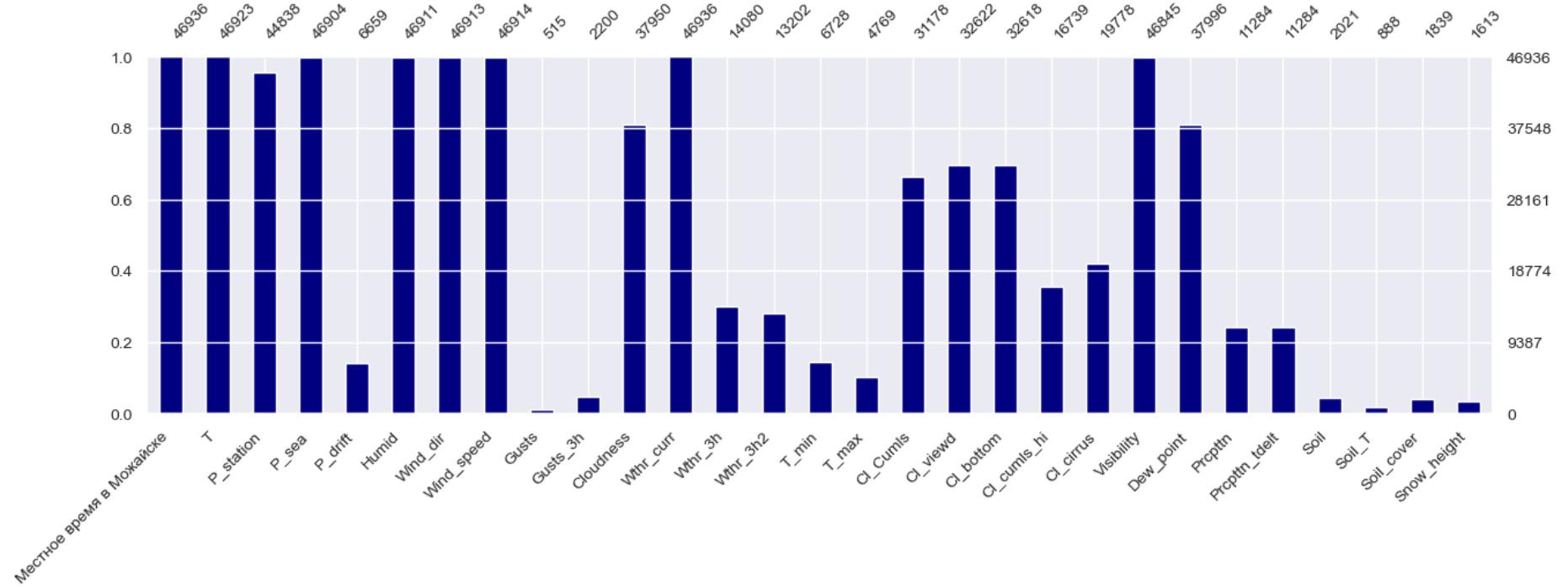
Дата начала наблюдений: 2005-02-01, количество записей: 46936

Всего, дней наблюдений: 5905, за 18 лет

Среднее количество наблюдений в день: 7.9485 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 3.3

Out[34]:
<AxesSubplot:>



Архив метеостанции df_N_Jerusalem

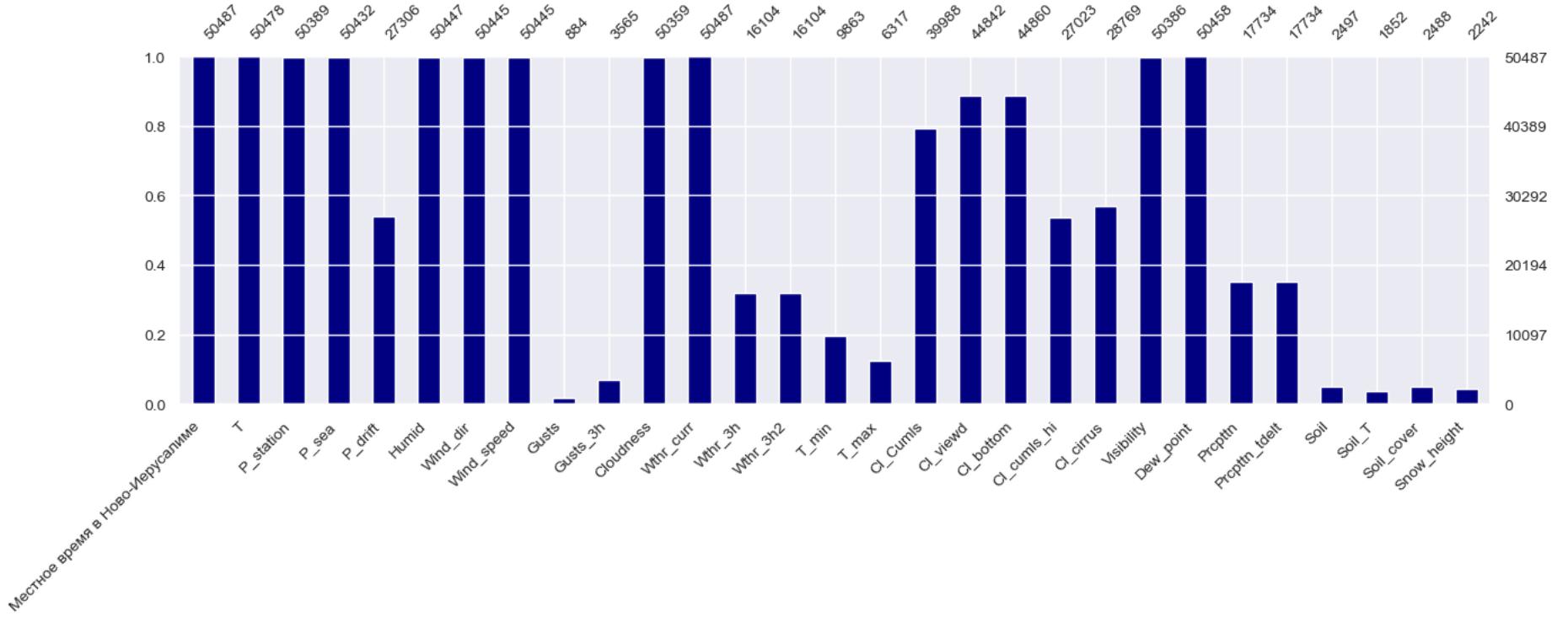
Дата начала наблюдений: 2005-02-01, количество записей: 50487

Всего, дней наблюдений: 6336, за 18 лет

Среднее количество наблюдений в день: 7.9683 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.2

Out[34]:



Архив метеостанции df_Nemchinovka

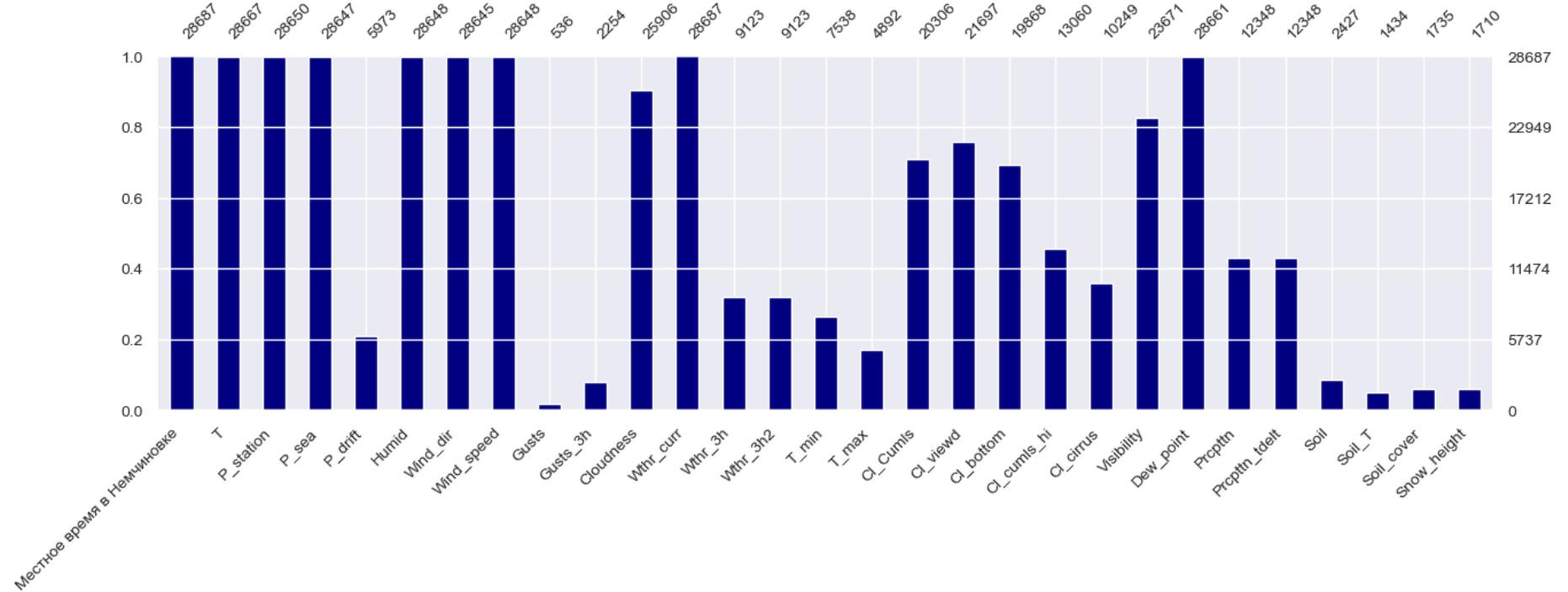
Дата начала наблюдений: 2007-09-26, количество записей: 28687

Всего, дней наблюдений: 4937, за 16 лет

Среднее количество наблюдений в день: 5.8106 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.2

Out[34]:
<AxesSubplot:>



Архив метеостанции df_Naro_Fominsk

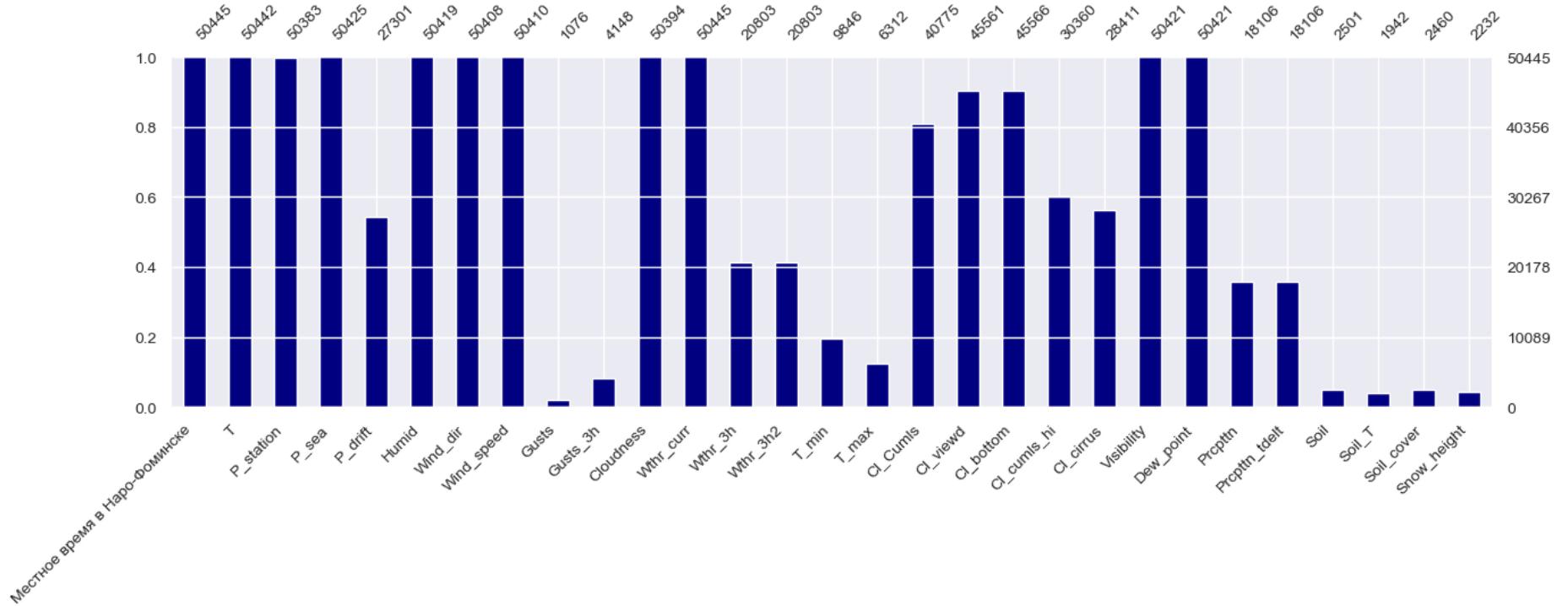
Дата начала наблюдений: 2005-02-01, количество записей: 50445

Всего, дней наблюдений: 6333, за 18 лет

Среднее количество наблюдений в день: 7.9654 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.2

Out[34]:



Архив метеостанции df_Serpukhov

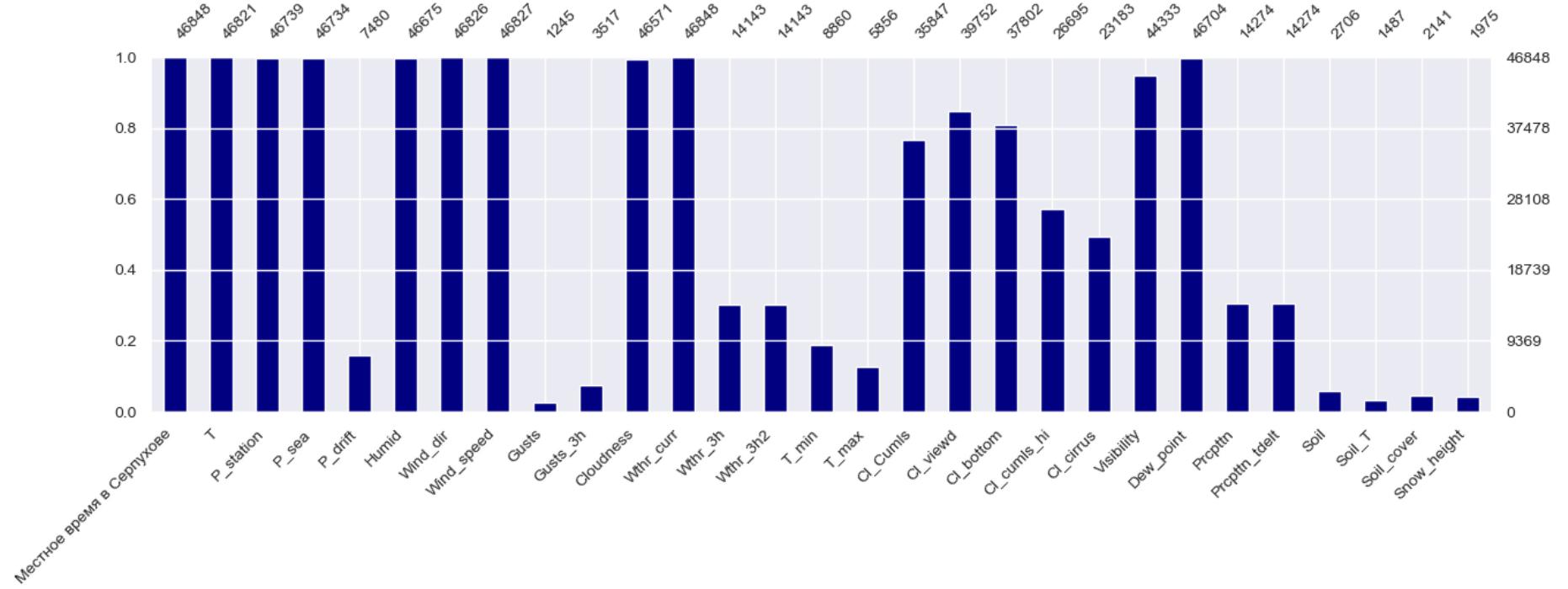
Дата начала наблюдений: 2005-02-01, количество записей: 46848

Всего, дней наблюдений: 5904, за 18 лет

Среднее количество наблюдений в день: 7.9350 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.0

Out[34]:
<AxesSubplot:>



ВЫВОД

Общие закономерности в количестве пропусков

- Количество наблюдений соответствует только словесному описанию текущей погоде, следовательно во всех параметрах, доступных для анализа, есть пропуски
- Незначительные пропуски (менее 1%) характерны для параметров: *температура, давление, влажность, направление ветра, скорость ветра, облачность, описание погоды, видимость, точка росы.
- Мало пропусков: все параметры облаков нижнего слоя (возможно отсутствие облачности),
- Среднее количество пропусков: точка росы, параметры высотных облаков (странны для точки росы, высотные облака образуются не всегда),
- Около половины пропусков: параметры осадков (они так же происходят не всегда,
- Много пропусков: описание погоды за период между наблюдениями, минимальная температура (для нас не очень критично, причины непонятны),

- Значительные пропуски: *параметры порывов ветра, максимальные температуры, состояние почвы и снегового покрова* (состояние почвы и параметры снегового покрова для нас могут оказаться критичными и требуют дополнительного анализа).

1. Вышний Волочек:

- Не во все дни происходило по 8 наблюдений
- Среднегодовое число месяцев с наблюдениями снегового покрова 4,3 месяца (относительно малая величина)
([количество данных о снеговом покрове]/[число дней наблюдения] * 12)

1. Старица:

- Близко к идеальному количество наблюдений в день,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,3 месяца

1. Кашин

- Не во все дни происходило по 8 наблюдений,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,4 месяца

1. Тверь

- Близко к идеальному количество наблюдений в день,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,3 месяца

1. Клин

- Близко к идеальному количество наблюдений в день,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,2 месяца

1. Дмитров

- Близко к идеальному количество наблюдений в день,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,3 месяца

1. Волоколамск

- Близко к идеальному количество наблюдений в день,

- Среднегодовое число месяцев с наблюдениями снегового покрова - 3,5 месяца (неправдоподобно мало!)

1. Можайск

- Близко к идеальному количество наблюдений в день,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 3,3 месяца (неправдоподобно мало!)

1. Ново-Иерусалим

- Близко к идеальному количество наблюдений в день,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,2 месяца

1. Немчиновка

- Наблюдения происходят существенно реже, чем 8 раз в день!
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,2 месяца

1. Наро-Фоминск

- Близко к идеальному количество наблюдений в день,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,2 месяца

1. Серпухов

- Близко к идеальному количество наблюдений в день,
- Среднегодовое число месяцев с наблюдениями снегового покрова - 4,0 месяца (относительно мало!)

1.1.1.2 Анализ распределения пропущенных и нулевых данных

```
In [35]: print('ПАСПРЕДЕЛЕНИЕ ДАННЫХ И ПРОПУСКОВ ПО ФРЕЙМУ\n')

for name, data in dict_df_data.items(): # по элементам словаря с данными DFs
    observation_quantities(data) # вызовем функцию распечатки сводных данных о количестве наблюдений

    msno.matrix(data, color=(0, 100/225, 0), figsize=(15, 7), fontsize=10); # выводим график из библиотеки "missingno"
    plt.show()
```

РАСПРЕДЕЛЕНИЕ ДАННЫХ И ПРОПУСКОВ ПО ФРЕЙМУ

Архив метеостанции df_V_Volochek

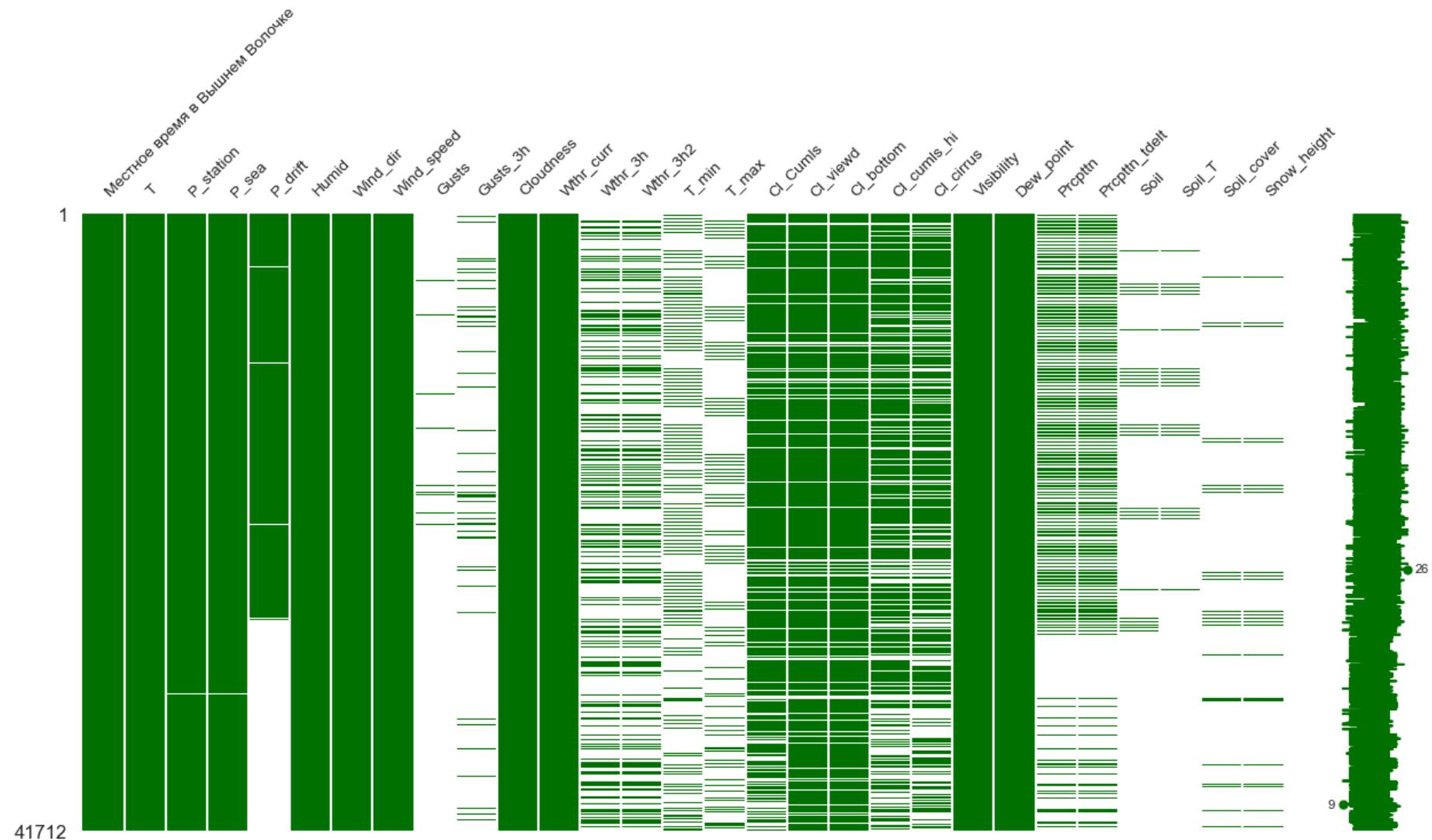
Дата начала наблюдений: 2005-07-11, количество записей: 41712

Всего, дней наблюдений: 6168, за 18 лет

Среднее количество наблюдений в день: 6.7626 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 4.3

Out[35]:



Архив метеостанции df_Staritsa

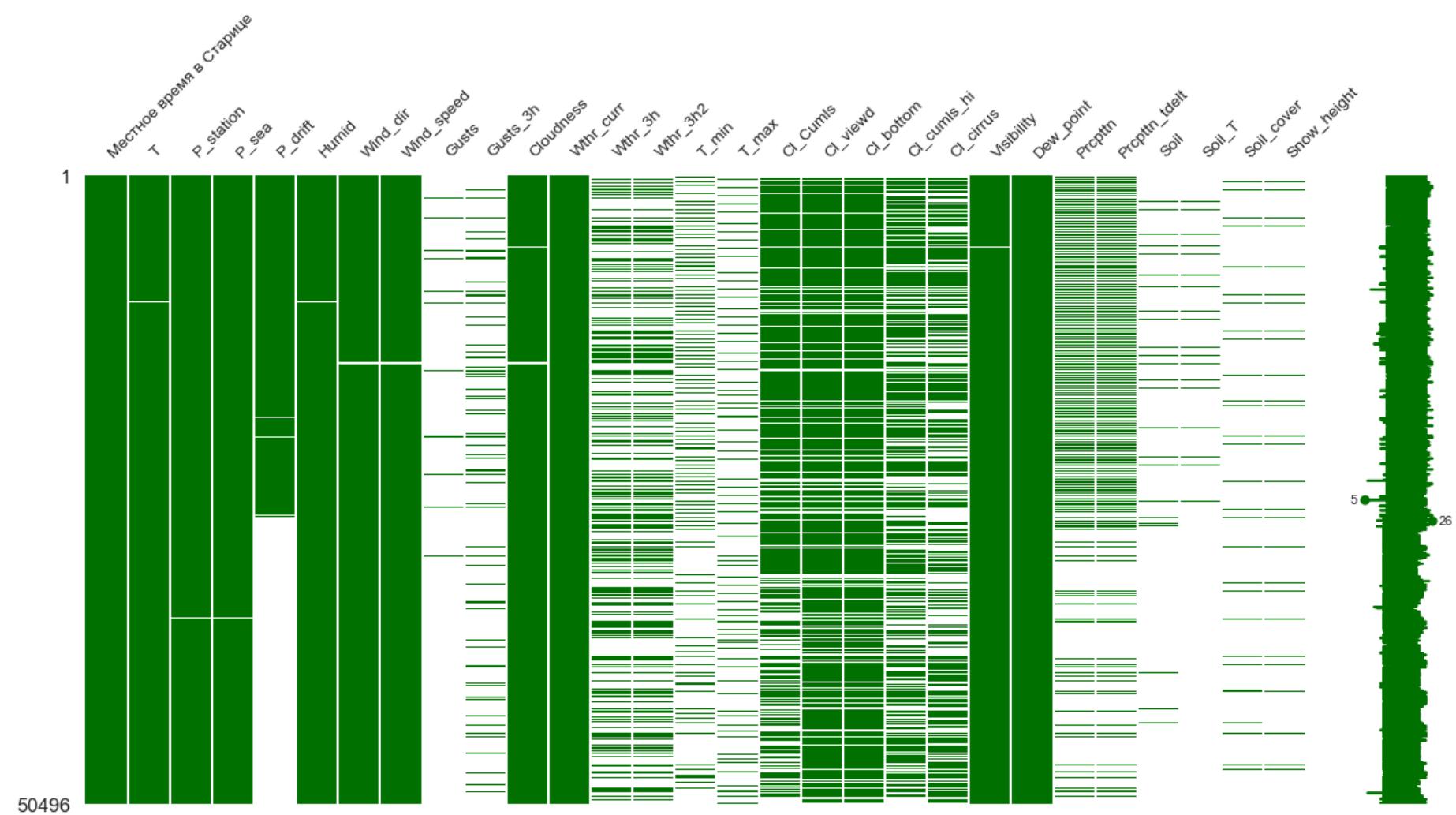
Дата начала наблюдений: 2005-02-01, количество записей: 50496

Всего, дней наблюдений: 6336, за 18 лет

Среднее количество наблюдений в день: 7.9697 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 4.3

Out[35]:



Архив метеостанции df_Kashyn

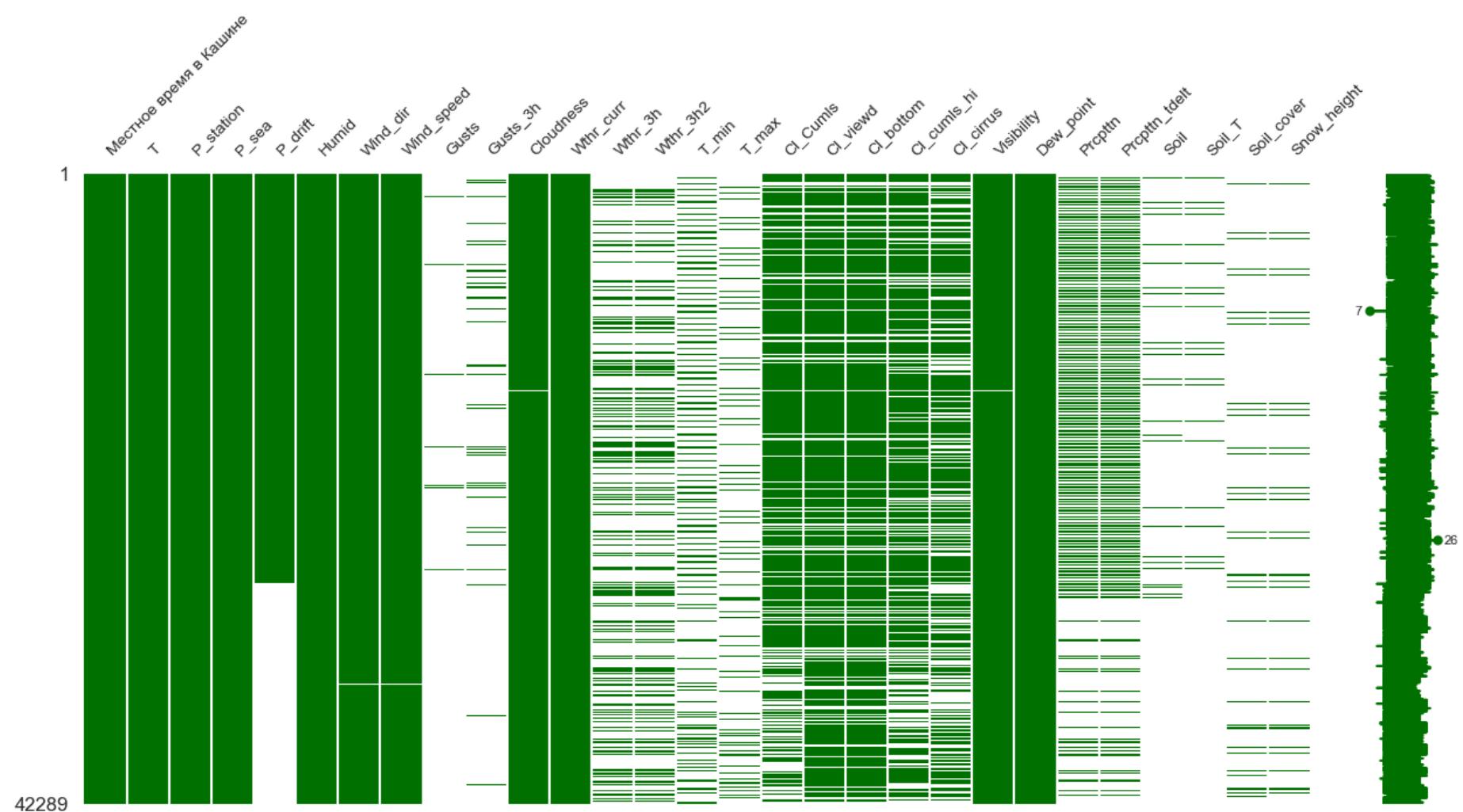
Дата начала наблюдений: 2005-07-11, количество записей: 42289

Всего, дней наблюдений: 6173, за 18 лет

Среднее количество наблюдений в день: 6.8506 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 4.4

Out[35]: <AxesSubplot:>



Архив метеостанции df_Tver

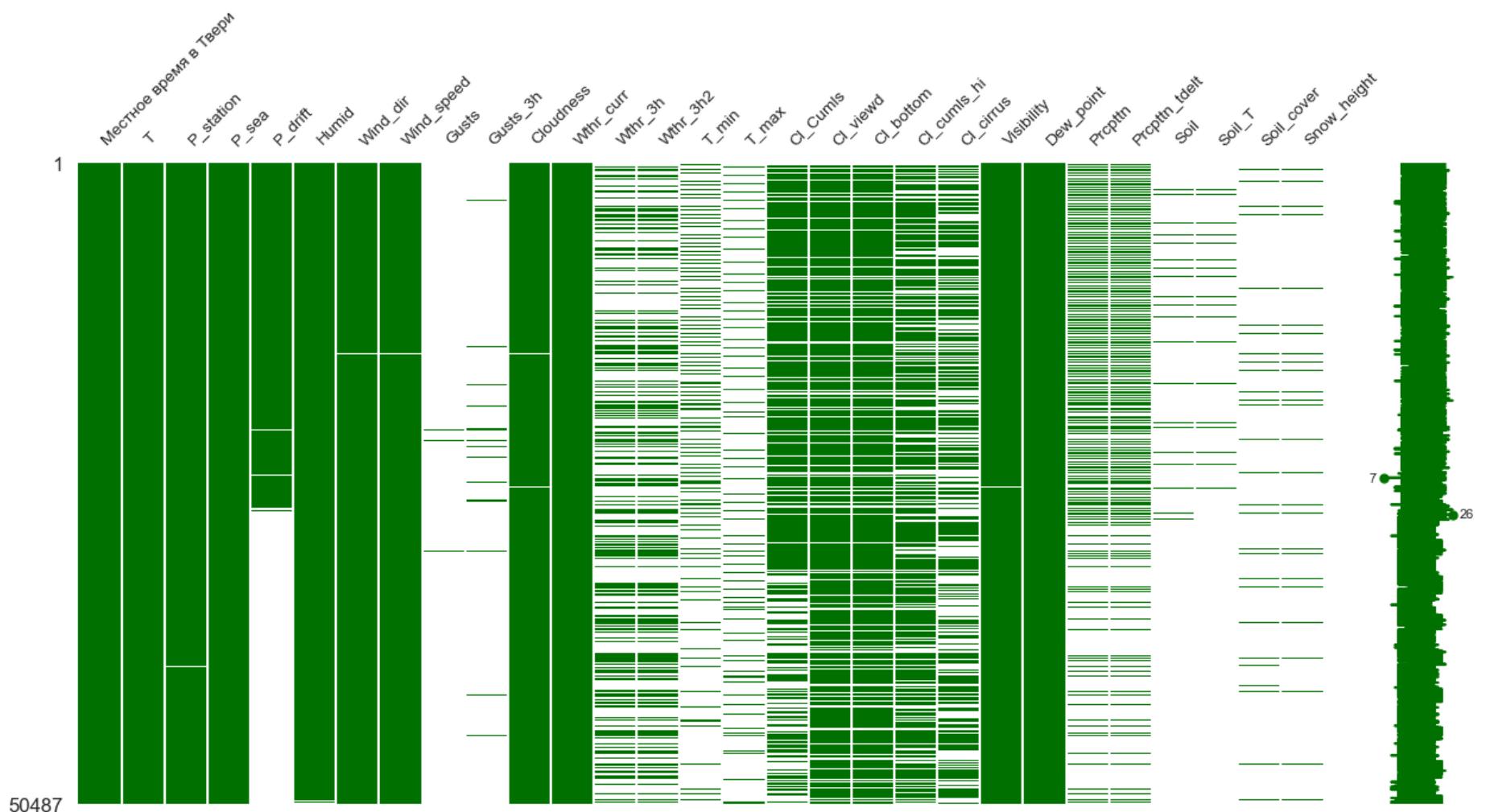
Дата начала наблюдений: 2005-02-01, количество записей: 50487

Всего, дней наблюдений: 6336, за 18 лет

Среднее количество наблюдений в день: 7.9683 (идеал = 8.0)

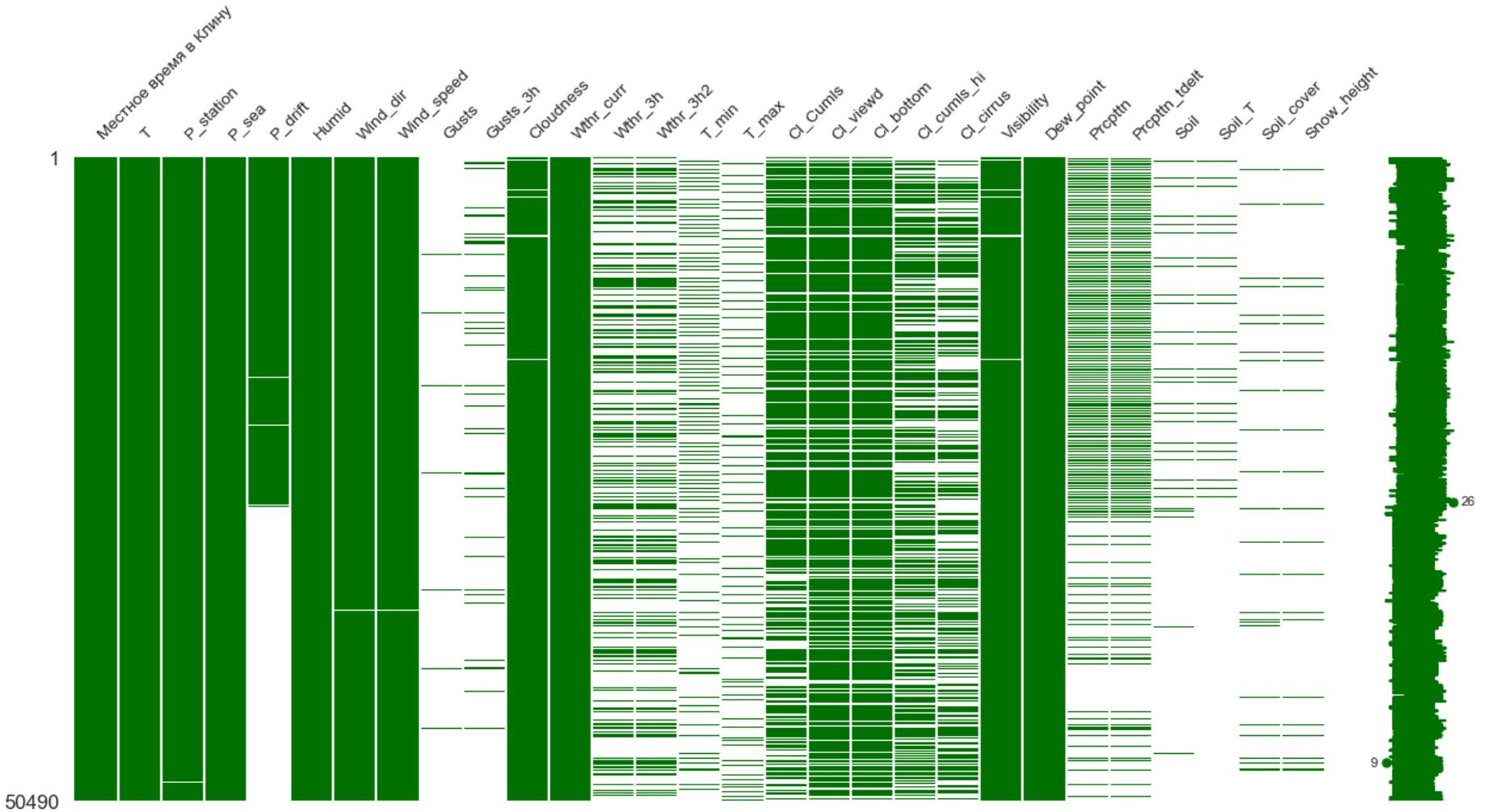
Среднегодовое число месяцев со снежевым покровом: 4.3

Out[35]: <AxesSubplot:>



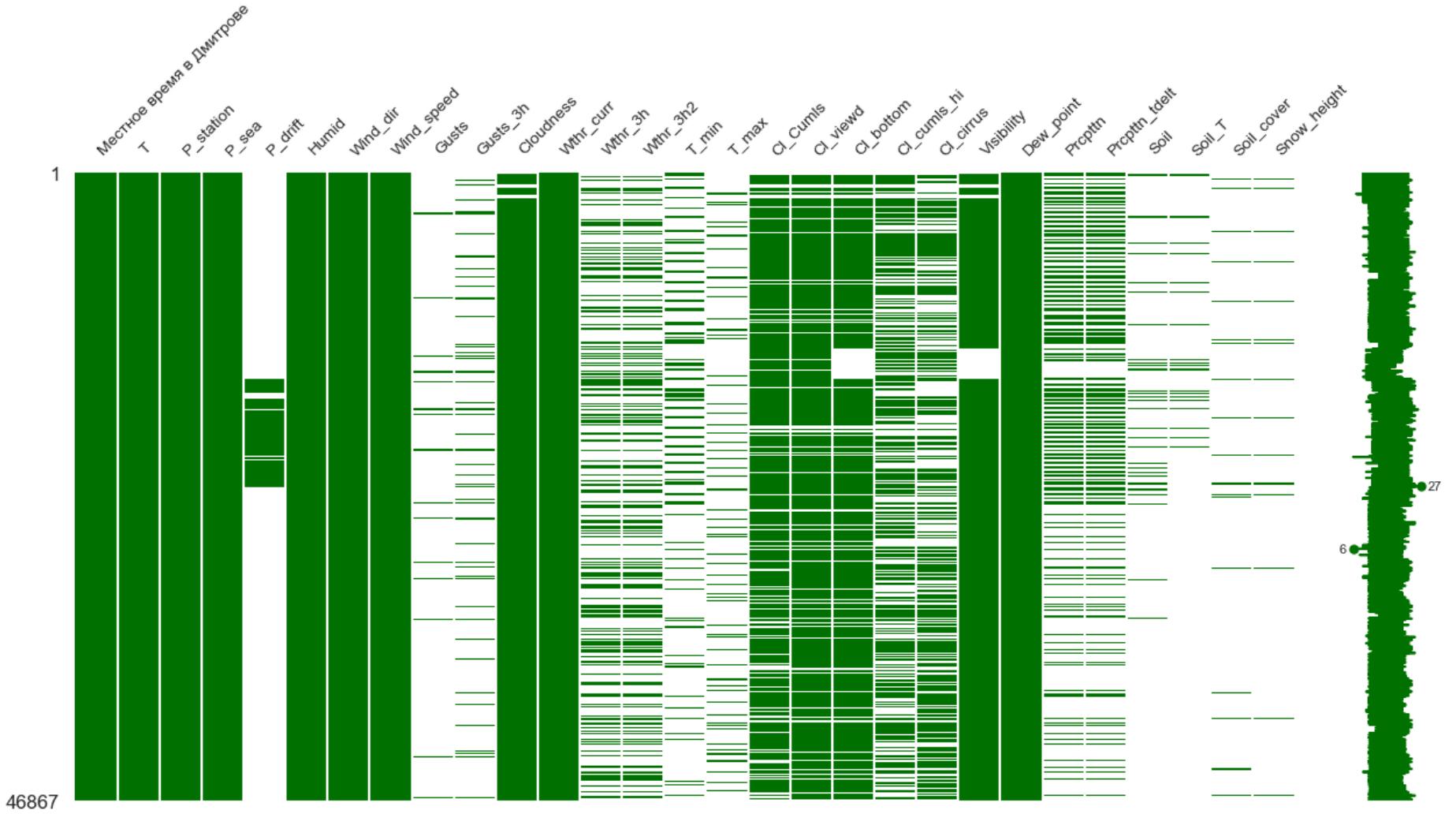
Архив метеостанции df_Klin
Дата начала наблюдений: 2005-02-01, количество записей: 50490
Всего, дней наблюдений: 6336, за 18 лет
Среднее количество наблюдений в день: 7.9688 (идеал = 8.0)
Среднегодовое число месяцев со снежным покровом: 4.2

Out[35]: <AxesSubplot:>



Архив метеостанции df_Dmitrov
Дата начала наблюдений: 2005-02-01, количество записей: 46867
Всего, дней наблюдений: 5905, за 18 лет
Среднее количество наблюдений в день: 7.9368 (идеал = 8.0)
Среднегодовое число месяцев со снежным покровом: 4.3

Out[35]:



Архив метеостанции df_Volokolamsk

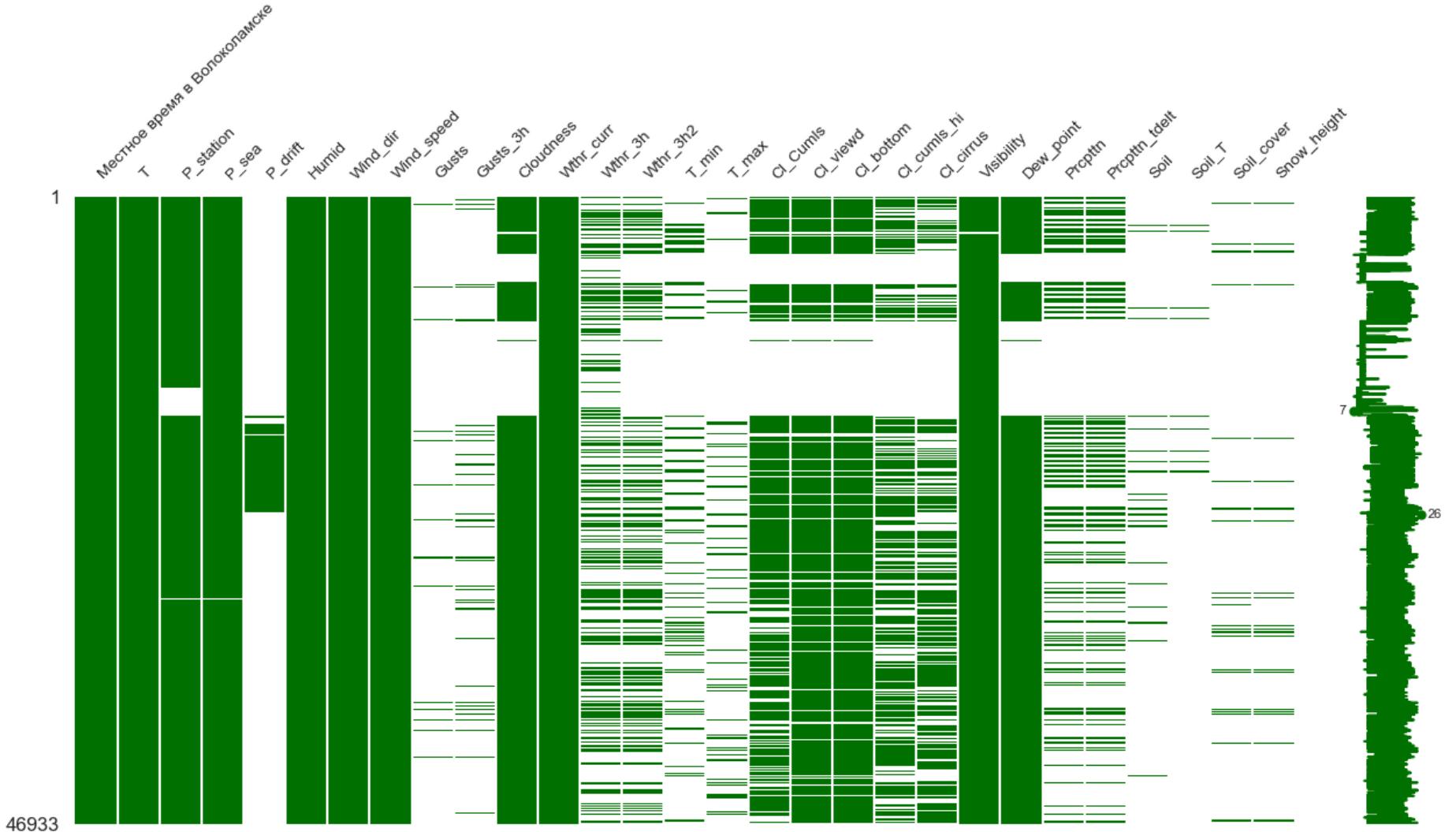
Дата начала наблюдений: 2005-02-01, количество записей: 46933

Всего, дней наблюдений: 5905, за 18 лет

Среднее количество наблюдений в день: 7.9480 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 3.5

Out[35]:



Архив метеостанции df_Mozhaisk

Дата начала наблюдений: 2005-02-01, количество записей: 46936

Всего, дней наблюдений: 5905, за 18 лет

Среднее количество наблюдений в день: 7.9485 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 3.3

Out[35]:
<AxesSubplot:>



Архив метеостанции df_N_Jerusalem

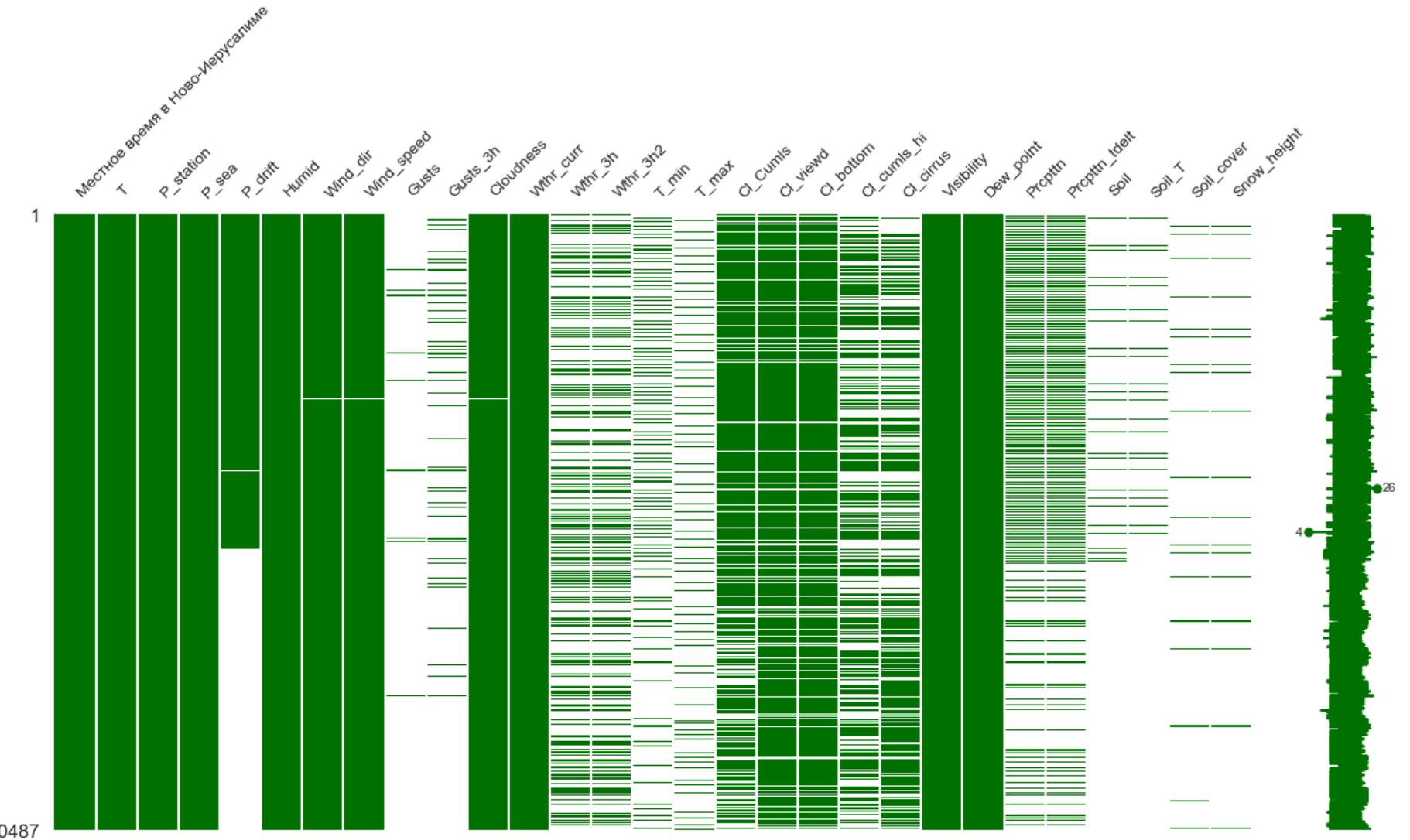
Дата начала наблюдений: 2005-02-01, количество записей: 50487

Всего, дней наблюдений: 6336, за 18 лет

Среднее количество наблюдений в день: 7.9683 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.2

Out[35]:



Архив метеостанции df_Nemchinovka

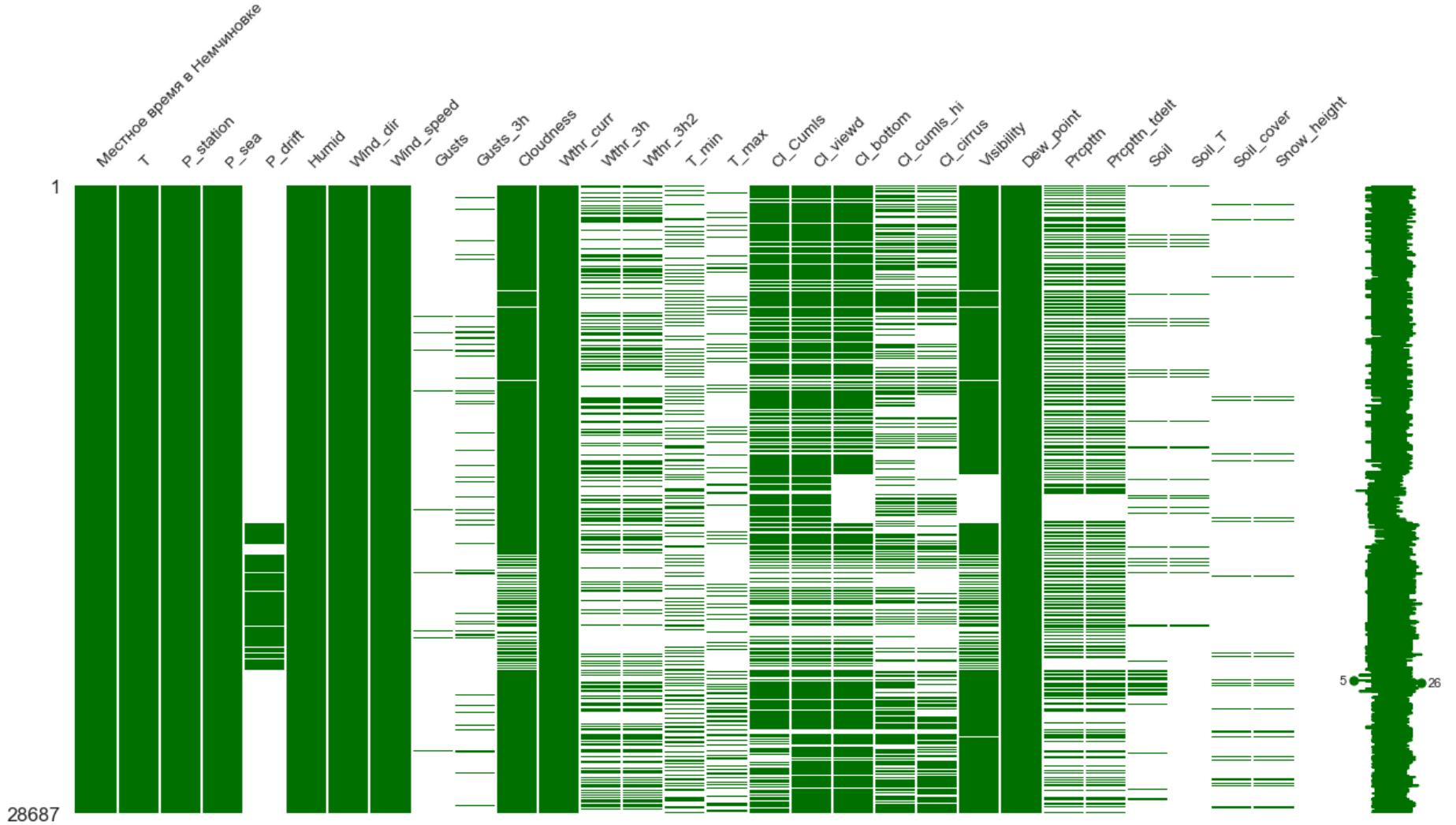
Дата начала наблюдений: 2007-09-26, количество записей: 28687

Всего, дней наблюдений: 4937, за 16 лет

Среднее количество наблюдений в день: 5.8106 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.2

Out[35]: <AxesSubplot:>



Архив метеостанции df_Naro_Fominsk

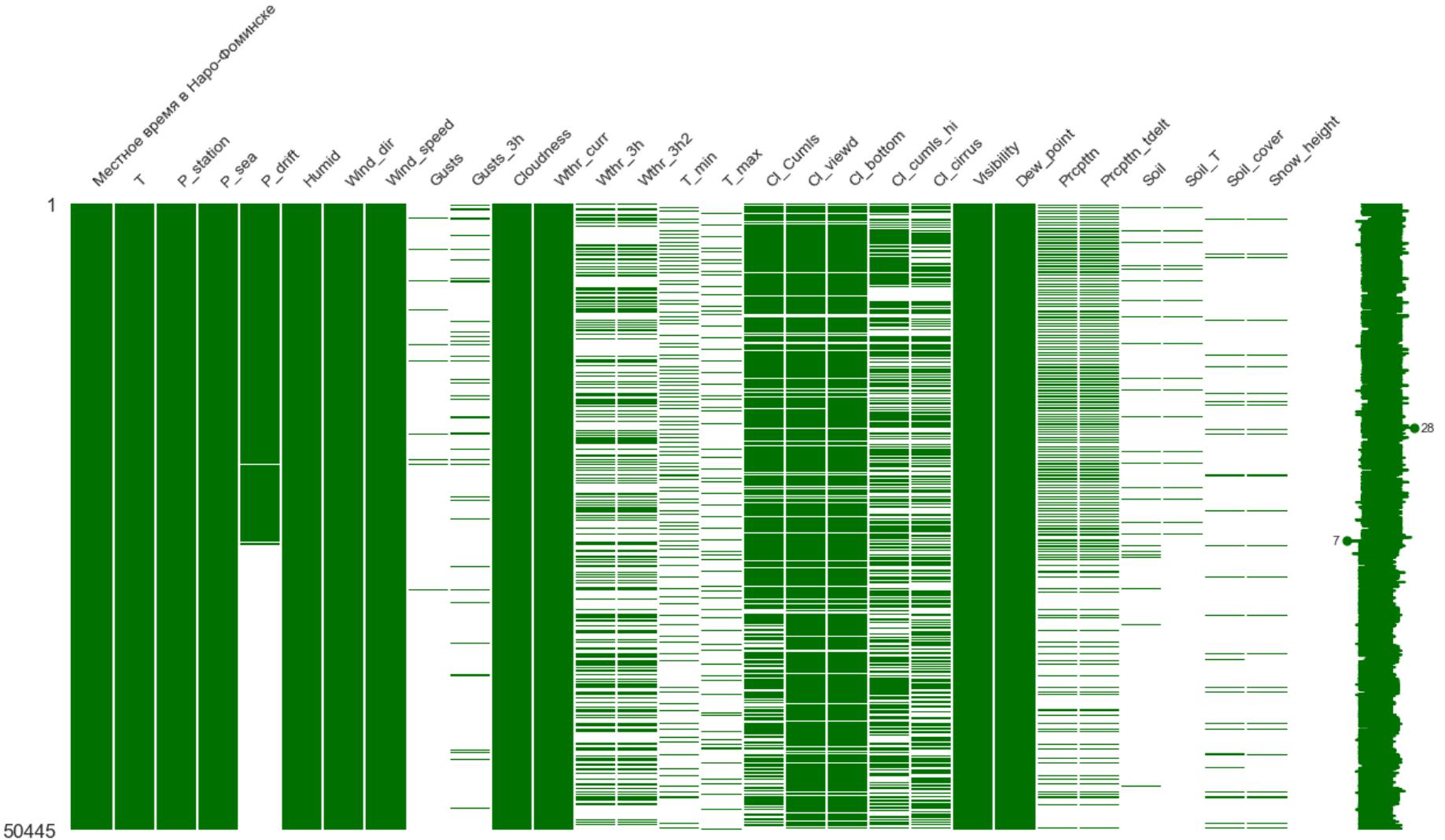
Дата начала наблюдений: 2005-02-01, количество записей: 50445

Всего, дней наблюдений: 6333, за 18 лет

Среднее количество наблюдений в день: 7.9654 (идеал = 8.0)

Среднегодовое число месяцев со снежевым покровом: 4.2

Out[35]:
<AxesSubplot:>



Архив метеостанции df_Serpukhov

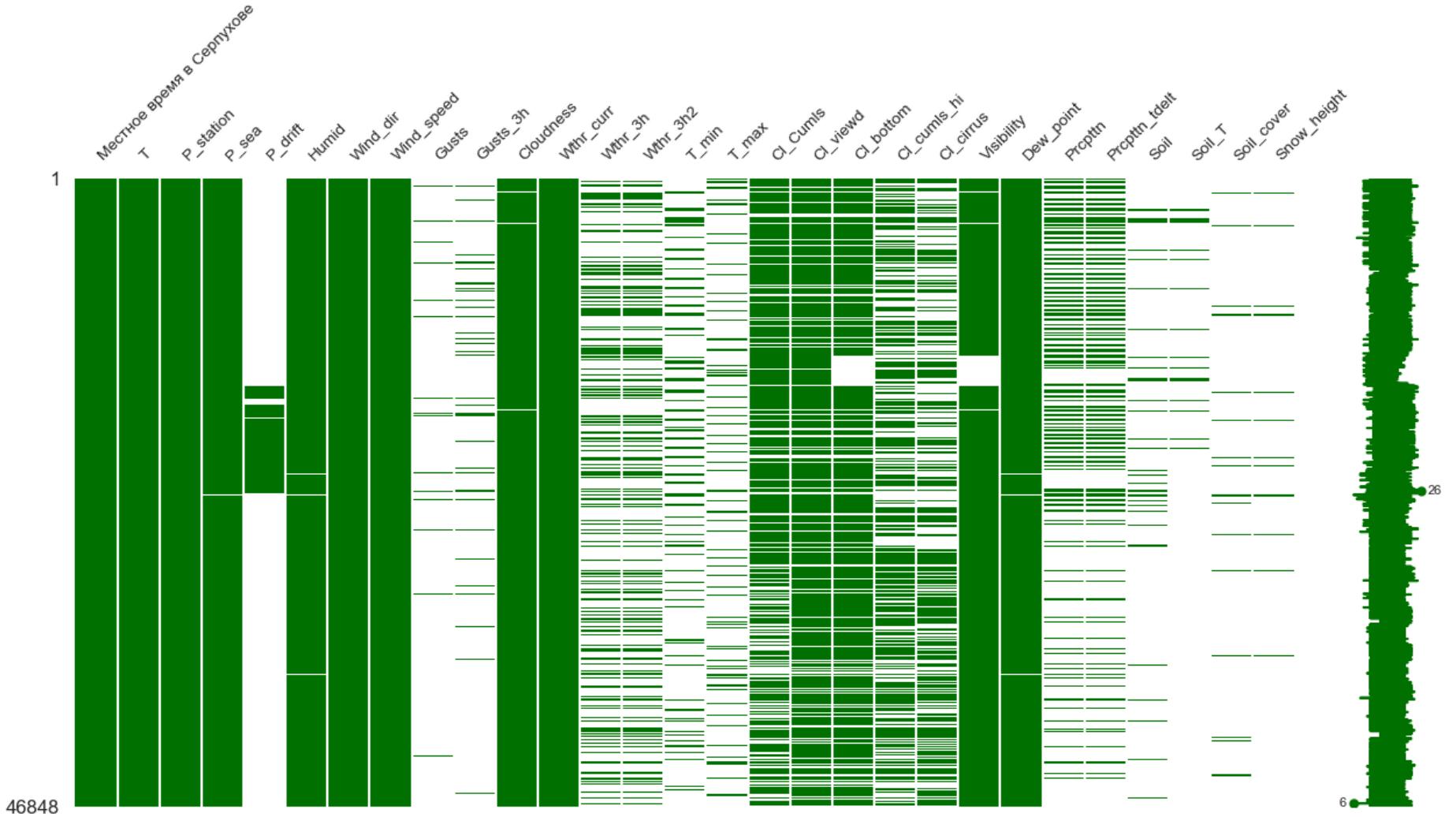
Дата начала наблюдений: 2005-02-01, количество записей: 46848

Всего, дней наблюдений: 5904, за 18 лет

Среднее количество наблюдений в день: 7.9350 (идеал = 8.0)

Среднегодовое число месяцев со снежным покровом: 4.0

Out[35]:
<AxesSubplot:>



ВЫВОД

В дополнения к предыдущим выводам

Общие закономерности в распределении пропусков.

- Барическая тенденция начала учитываться только с середины истории наблюдений, у части станций она в большинстве наблюдений отсутствует. (Это не столь важно, так как барическую тенденцию можно вычислить по интервалам наблюдений).

- Порывы ветра на ряде станций стали учитываться приблизительно с середины периода наблюдений. Пропуски равномерно распределены.
- Нерегулярно фиксируются минимальная и максимальная температура (это тоже вычисляемые значения)

1. Вышний Волочек:

- Возможно, в середине истории наблюдений есть большие пропуски в параметрах состоянии почвы и снегового покрова, или в этот период наблюдения фиксировались реже 8 раз в день.
- Большие пропуски в фиксации осадков в первой половине истории наблюдений.

1. Старица:

- Осадки стали регулярно фиксироваться во второй половине истории наблюдений.
- Состояние почвы (помимо снегового покрова) стали регулярно фиксироваться во второй половине истории наблюдений.

1. Кашин

- То же.

1. Тверь

- То же.

1. Клин

- То же.

1. Дмитров

- То же.

1. Волоколамск

- Помимо указанного в третьей четверти истории наблюдений существует "пробел", в течение которого многие параметры не фиксировались, включая важные для нас параметры осадков и снегового покрова.

1. Можайск

- Помимо указанного в третьей четверти истории наблюдений существует "пробел", в течение которого многие параметры не фиксировались, включая важные для нас параметры осадков и снегового покрова.

1. Ново-Иерусалим

- Осадки стали регулярно фиксироваться во второй половине истории наблюдений.
- Состояние почвы, а также снегового покрова стали регулярно фиксироваться во второй половине истории наблюдений.

1. Немчиновка

- Помимо указанного выше, есть пробелы в фиксации состояния почвы и снегового покрова.
- Есть пробелы в фиксации видимости.

1. Наро-Фоминск

- Есть пробелы в фиксации состояния почвы и снегового покрова.

1. Серпухов

- Осадки стали регулярно фиксироваться во второй половине истории наблюдений.
- Состояние почвы (помимо снегового покрова) стали регулярно фиксироваться во второй половине истории наблюдений.

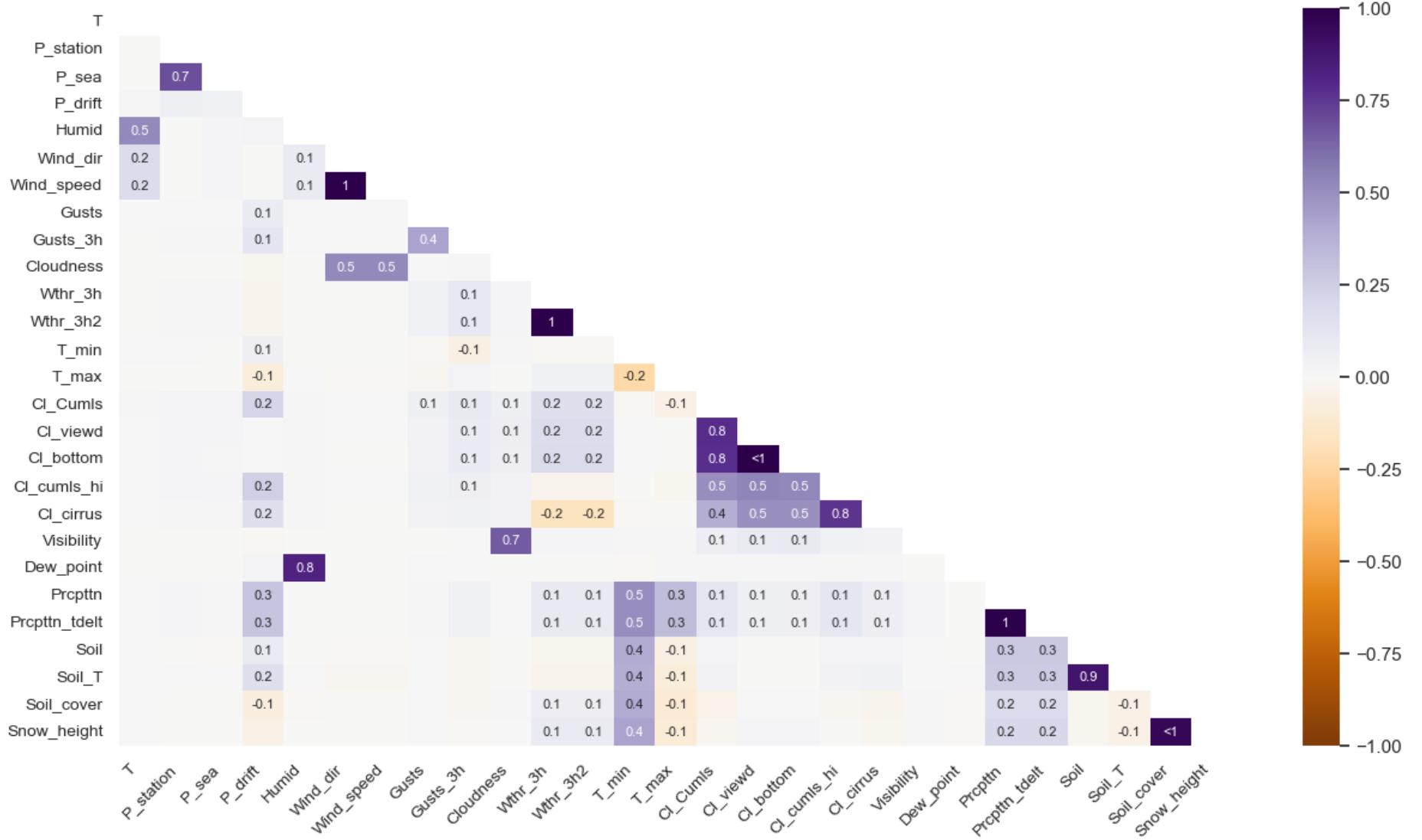
1.1.1.3 Анализ взаимосвязи между пропущенным (нулевыми) данными

```
In [36]: print('КОРРЕЛЯЦИЯ МЕЖДУ ПРОПУСКАМИ ДАННЫХ\n')
for name, data in dict_df_data.items(): # по элементам словаря с данными DFs
    print(f'Архив метеостанции {name}')
    msno.heatmap(data, cmap='PuOr', figsize=(15, 8), fontsize=10); # выводим график из библиотеки "missingno"
    plt.show()
```

КОРРЕЛЯЦИЯ МЕЖДУ ПРОПУСКАМИ ДАННЫХ

Архив метеостанции df_V_Volochek

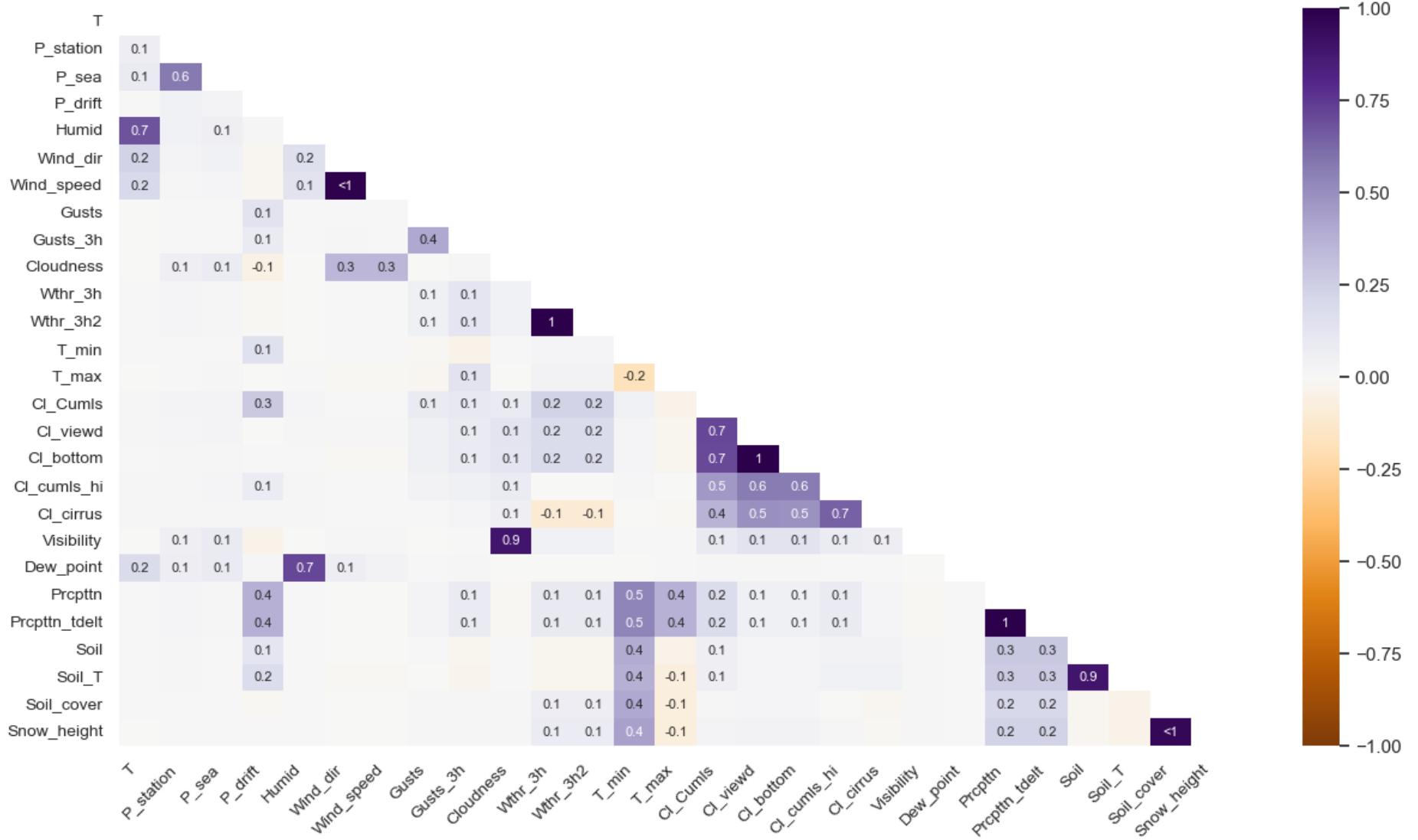
```
Out[36]: <AxesSubplot:>
```



Архив метеостанции df_Staritsa

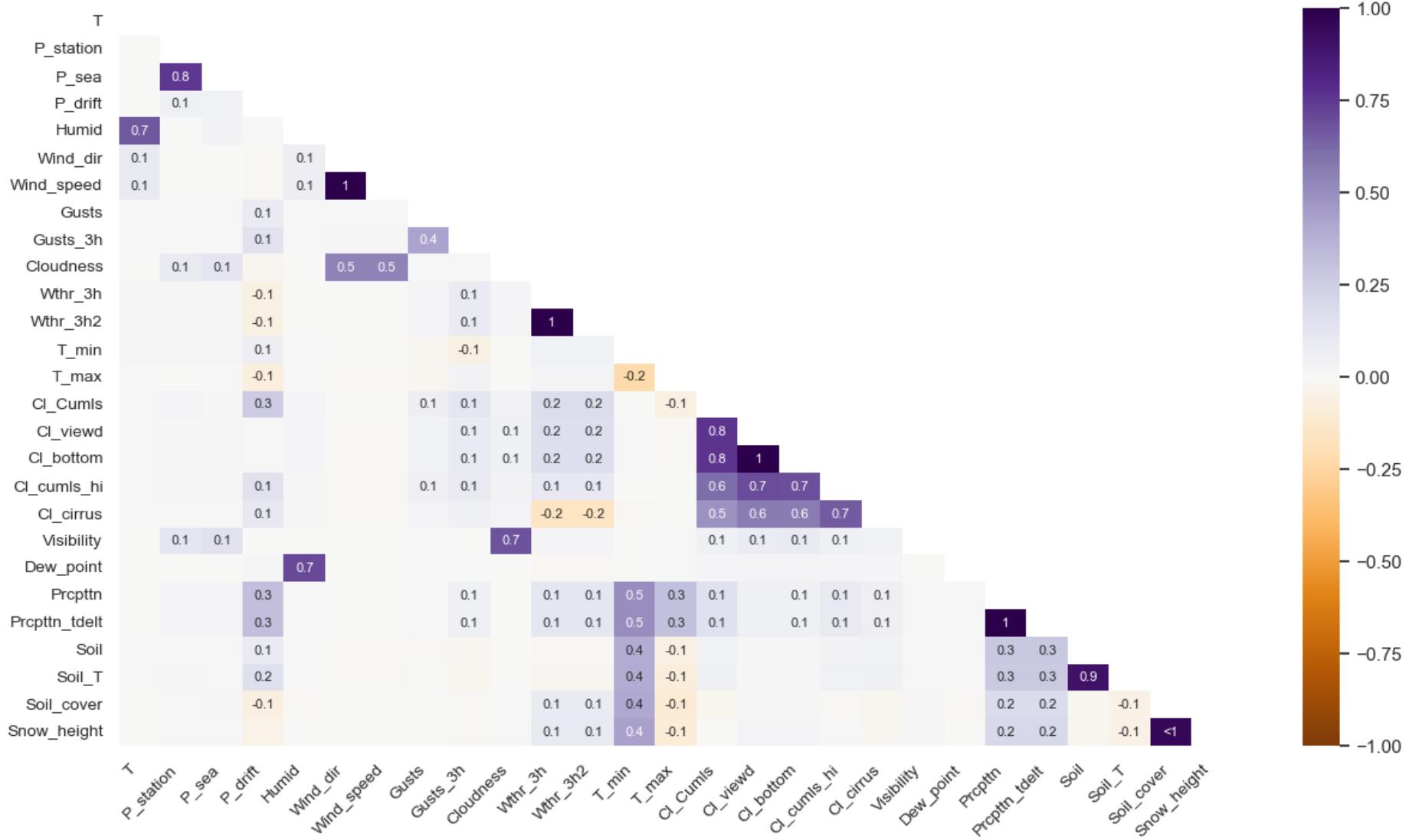
<AxesSubplot:>

Out[36]:



Архив метеостанции df_Kashyn

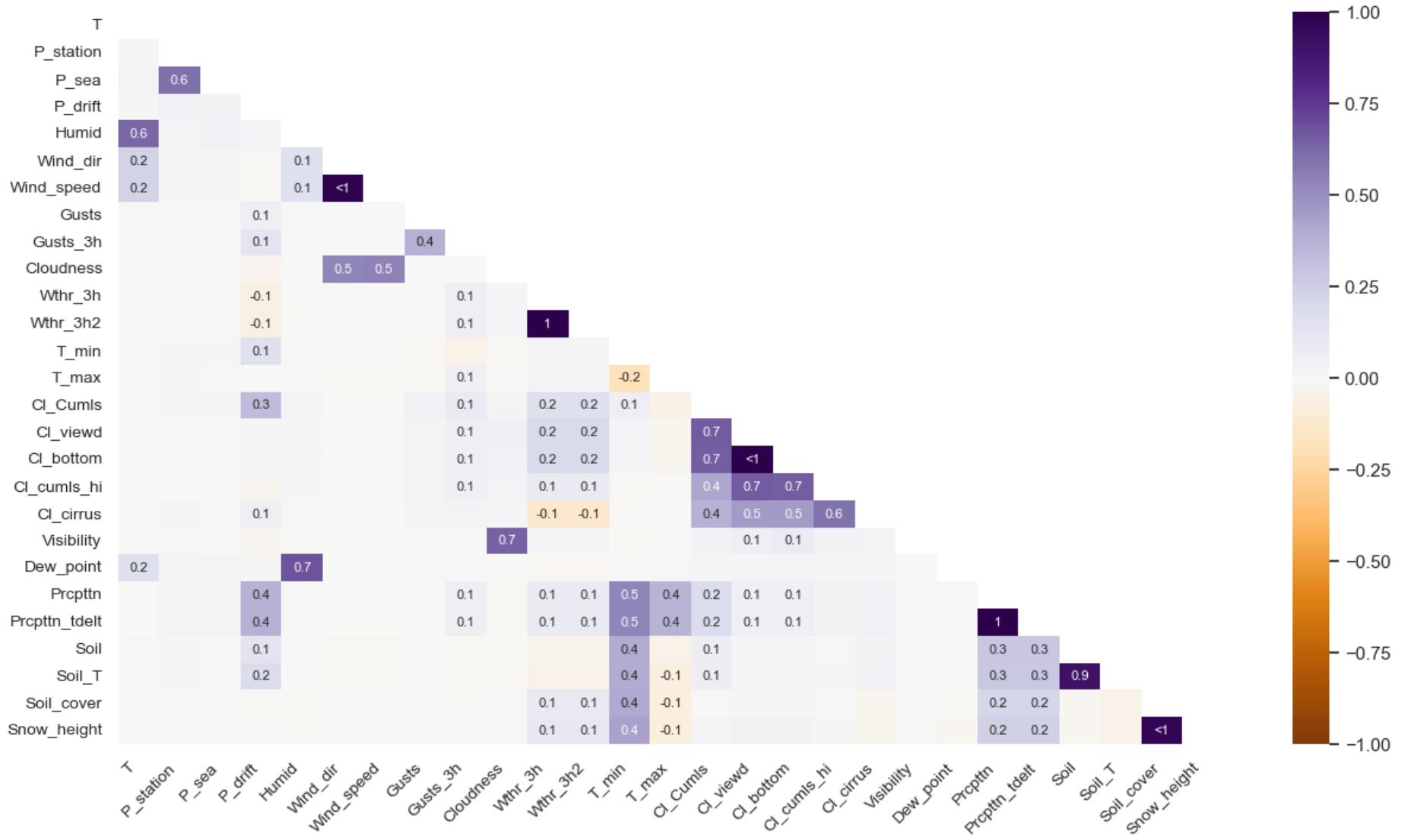
Out[36]: <AxesSubplot:>



Архив метеостанции df_Tver

<AxesSubplot:>

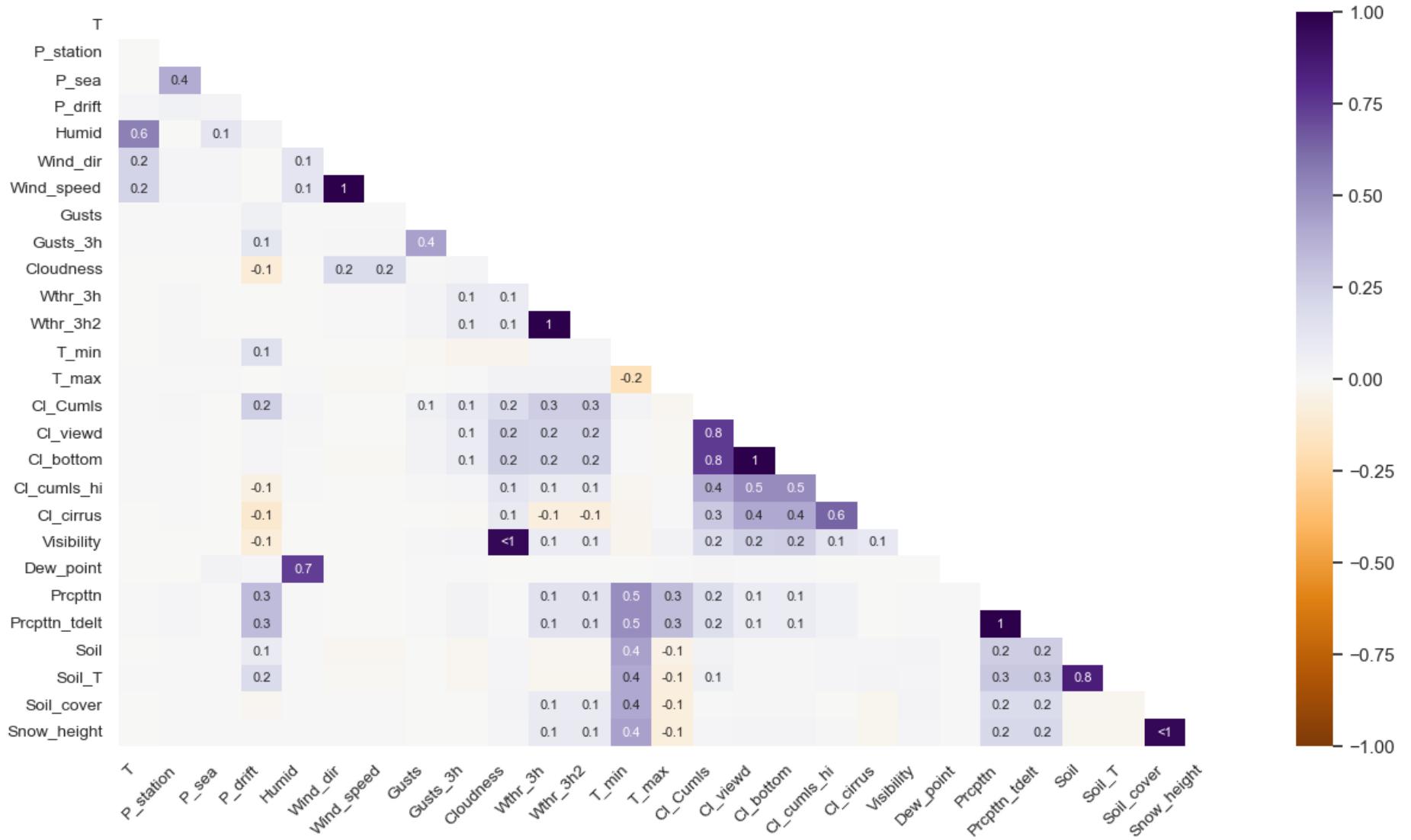
Out[36]:



Архив метеостанции df_Klin

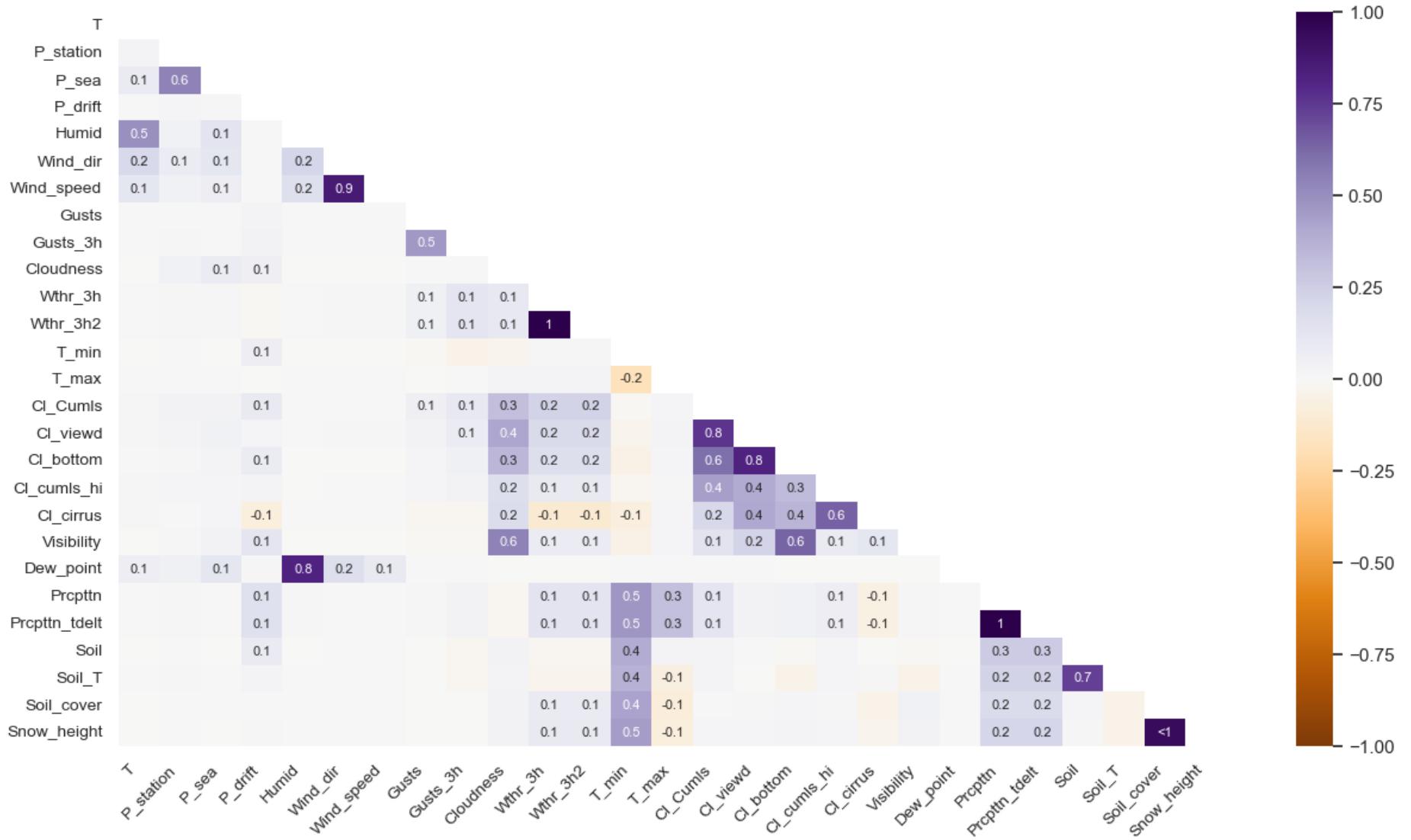
<AxesSubplot:>

Out[36]:



Архив метеостанции df_Dmitrov

Out[36]: <AxesSubplot:>



Архив метеостанции df_Volokolamsk

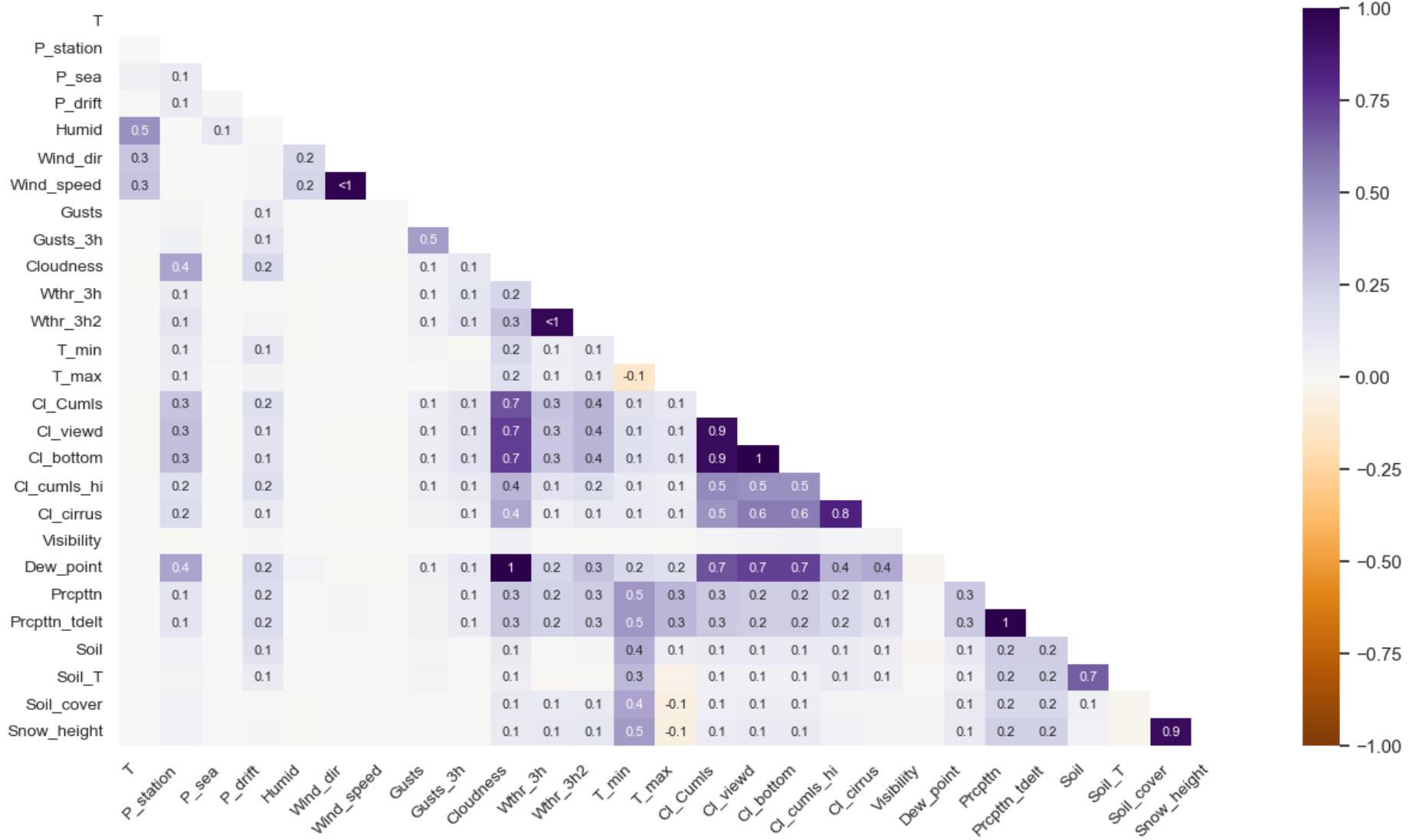
<AxesSubplot:>

Out[36]:



Архив метеостанции df_Mozhaisk

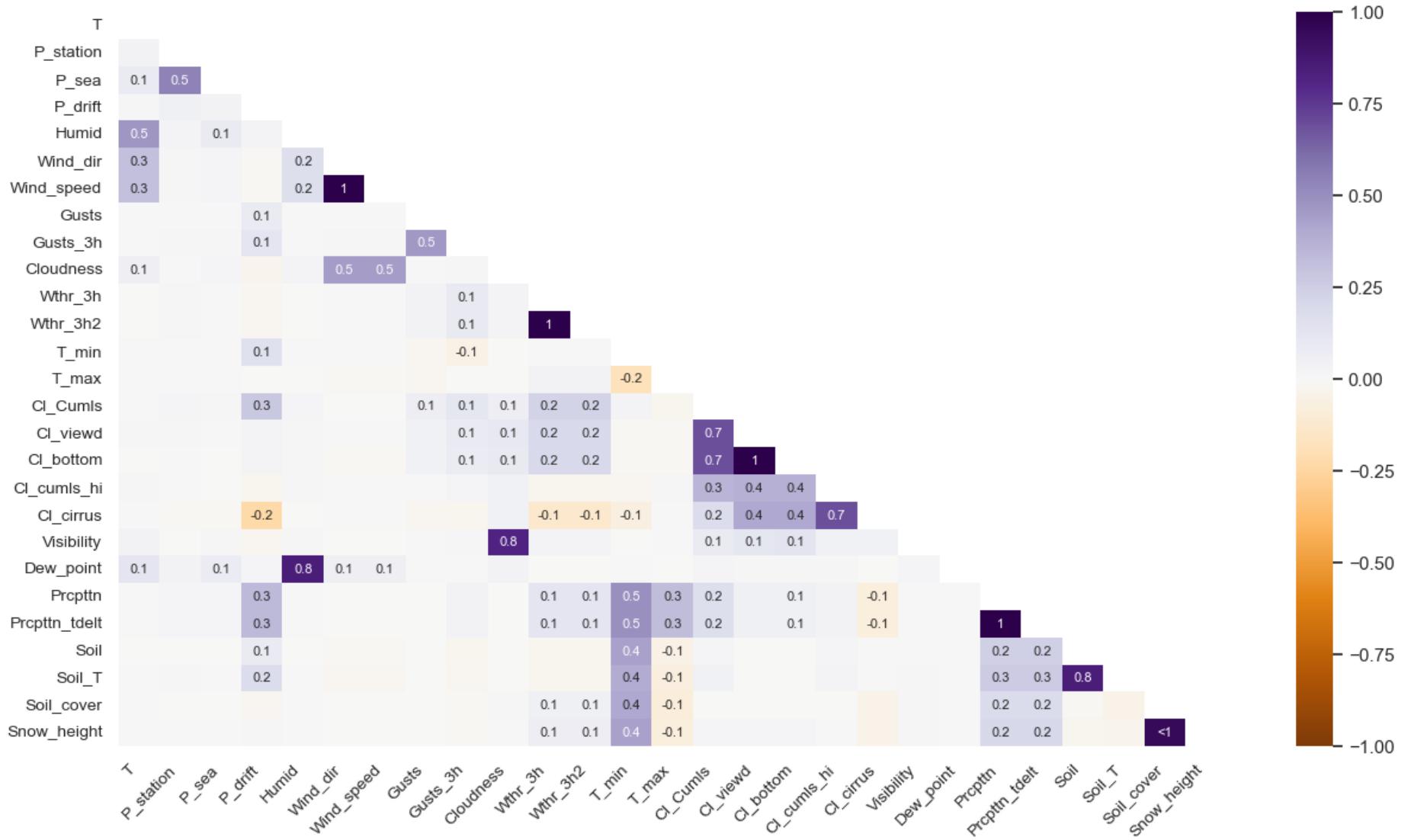
Out[36]: <AxesSubplot:>



Архив метеостанции df_N_Jerusalem

<AxesSubplot:>

Out[36]:



Архив метеостанции df_Nemchinovka

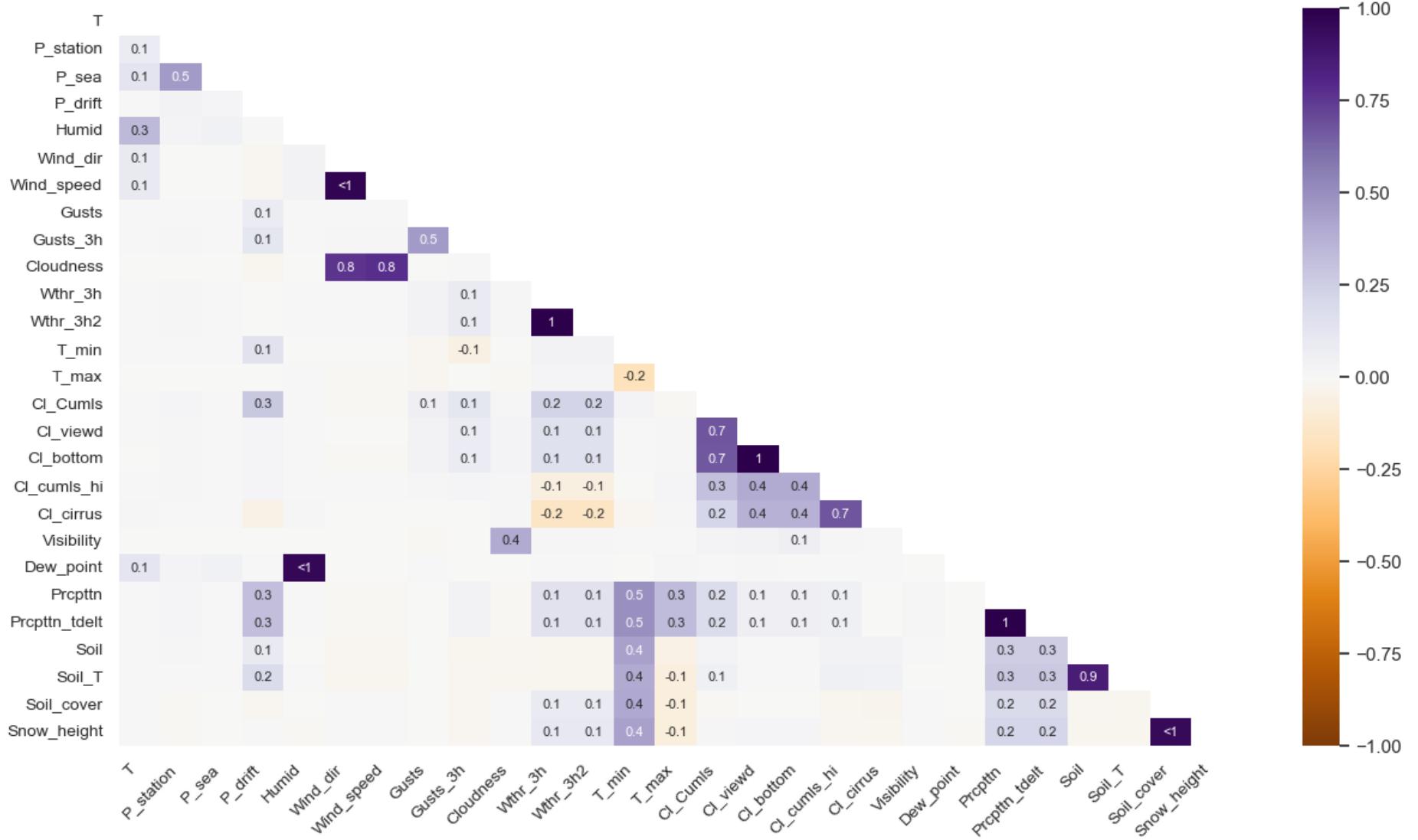
<AxesSubplot:>

Out[36]:



Архив метеостанции df_Naro_Fominsk

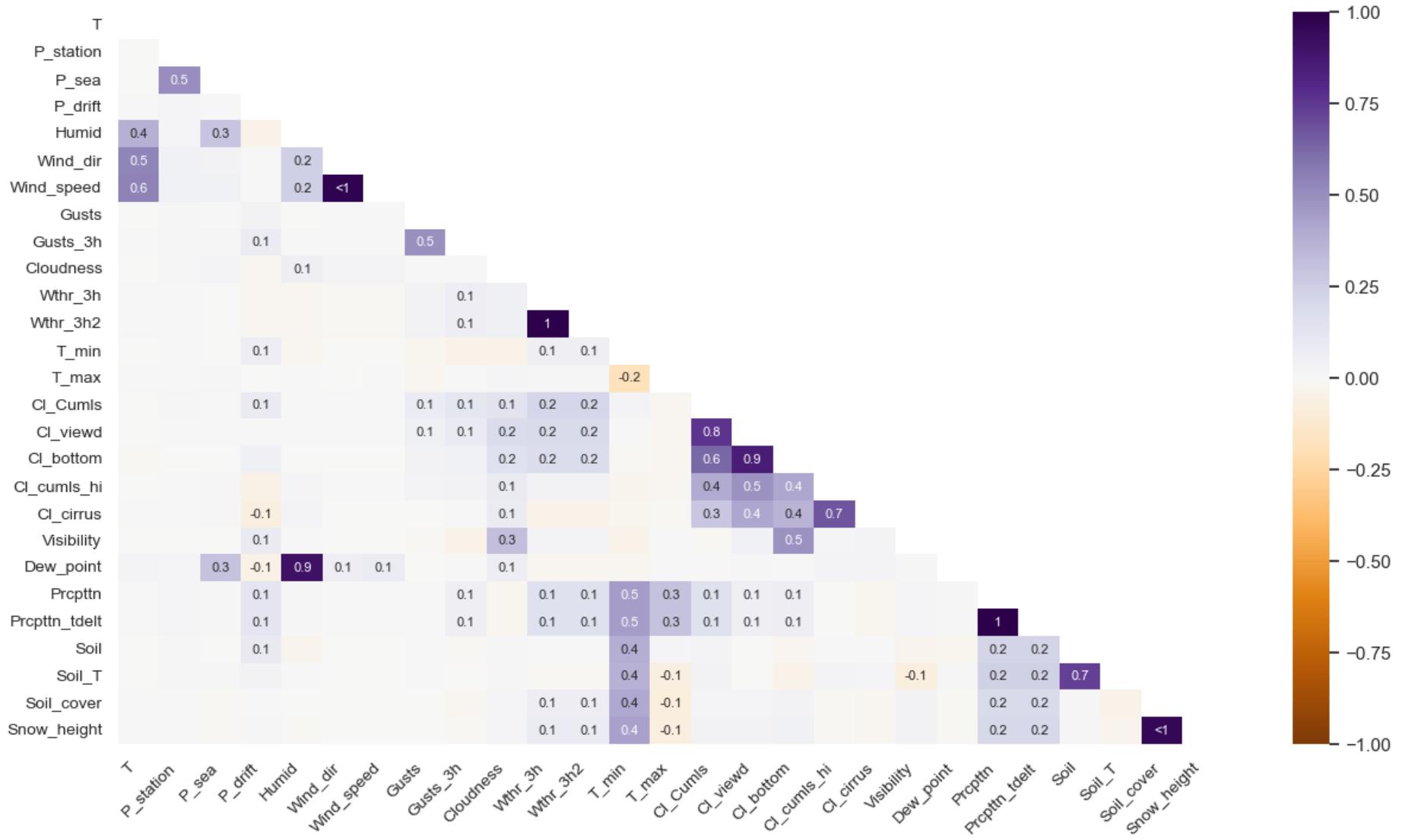
Out[36]: <AxesSubplot:>



Архив метеостанции df_Serpukhov

<AxesSubplot:>

Out[36]:



ВЫВОД

Все зависимости в отсутствующих параметрах являются логически объяснимыми:

- взаимозависимости точки росы, температуры и влажности,
- взаимозависимости характеристик облаков и наблюдаемой облачности,
- вычисляемые значения общего количества осадков в зависимости от количества осадков между наблюдениями

- взаимозависимости состояния поверхности почвы и данных о снеговом покрове, и т.д.

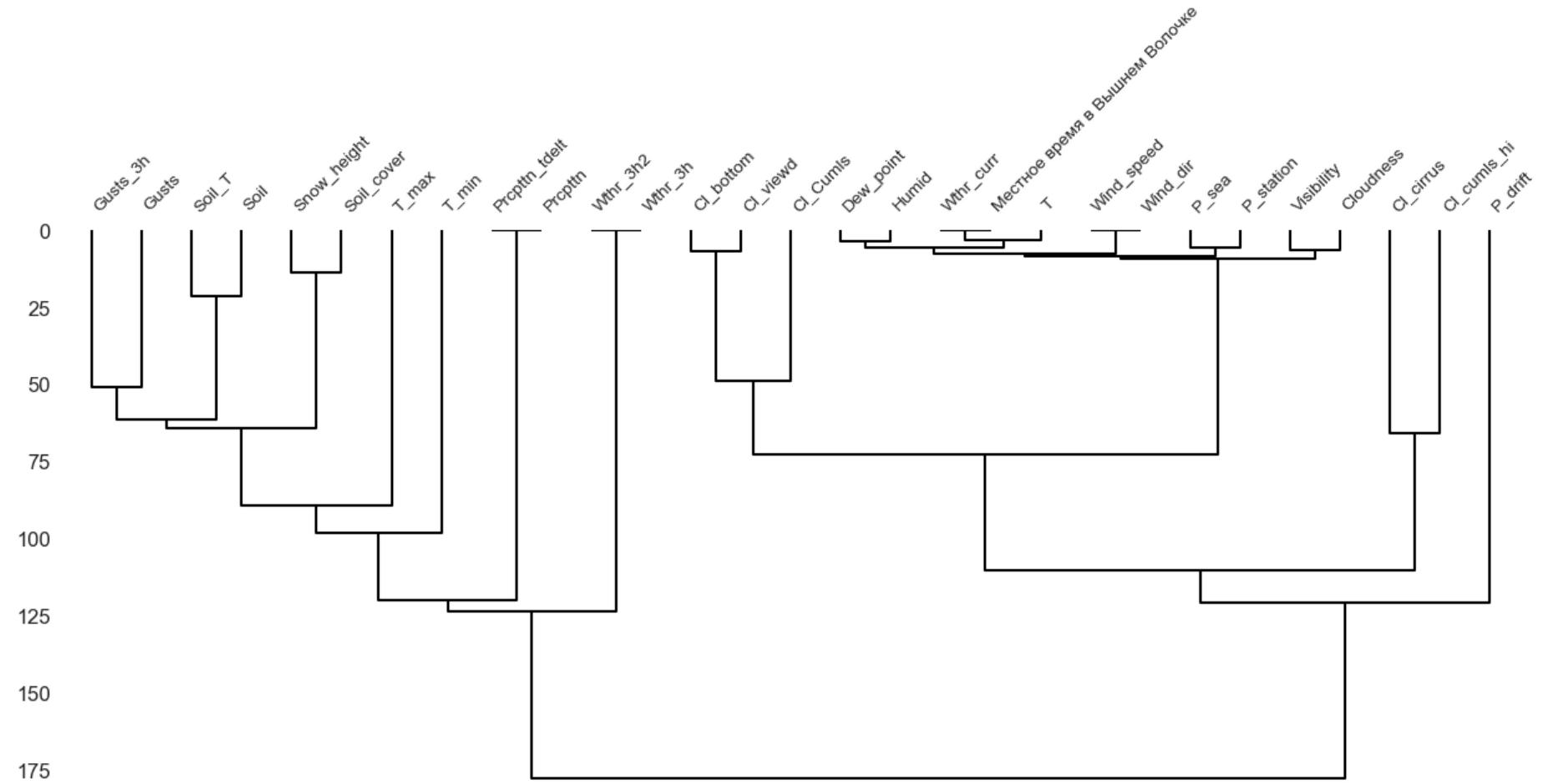
1.1.1.4. Анализ "близости" пропущенных (нулевых) данных

```
In [37]: print('ДЕНДРОГРАММА ВЗАИМОСВЯЗИ ПРОПУСКОВ ДАННЫХ')
for name, data in dict_df_data.items(): # по элементам словаря с данными DFs
    print(f'Архив метеостанции {name}')
    msno.dendrogram(data, figsize=(15, 6), fontsize=10); # выводим график из библиотеки "missingno"
    plt.show()
```

ДЕНДРОГРАММА ВЗАИМОСВЯЗИ ПРОПУСКОВ ДАННЫХ

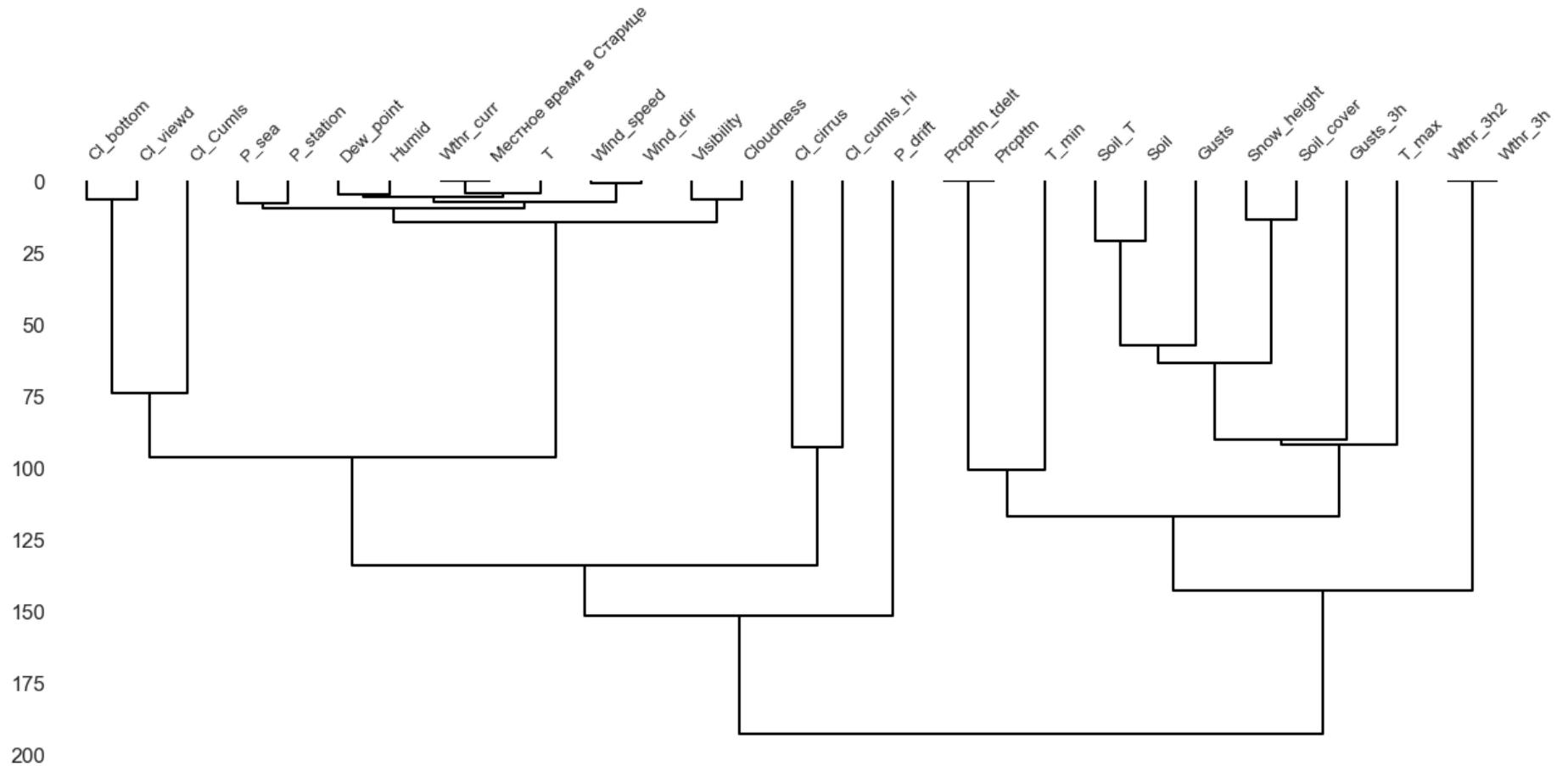
Архив метеостанции df_V_Volochek

Out[37]: <AxesSubplot:>



Архив метеостанции df_Staritsa

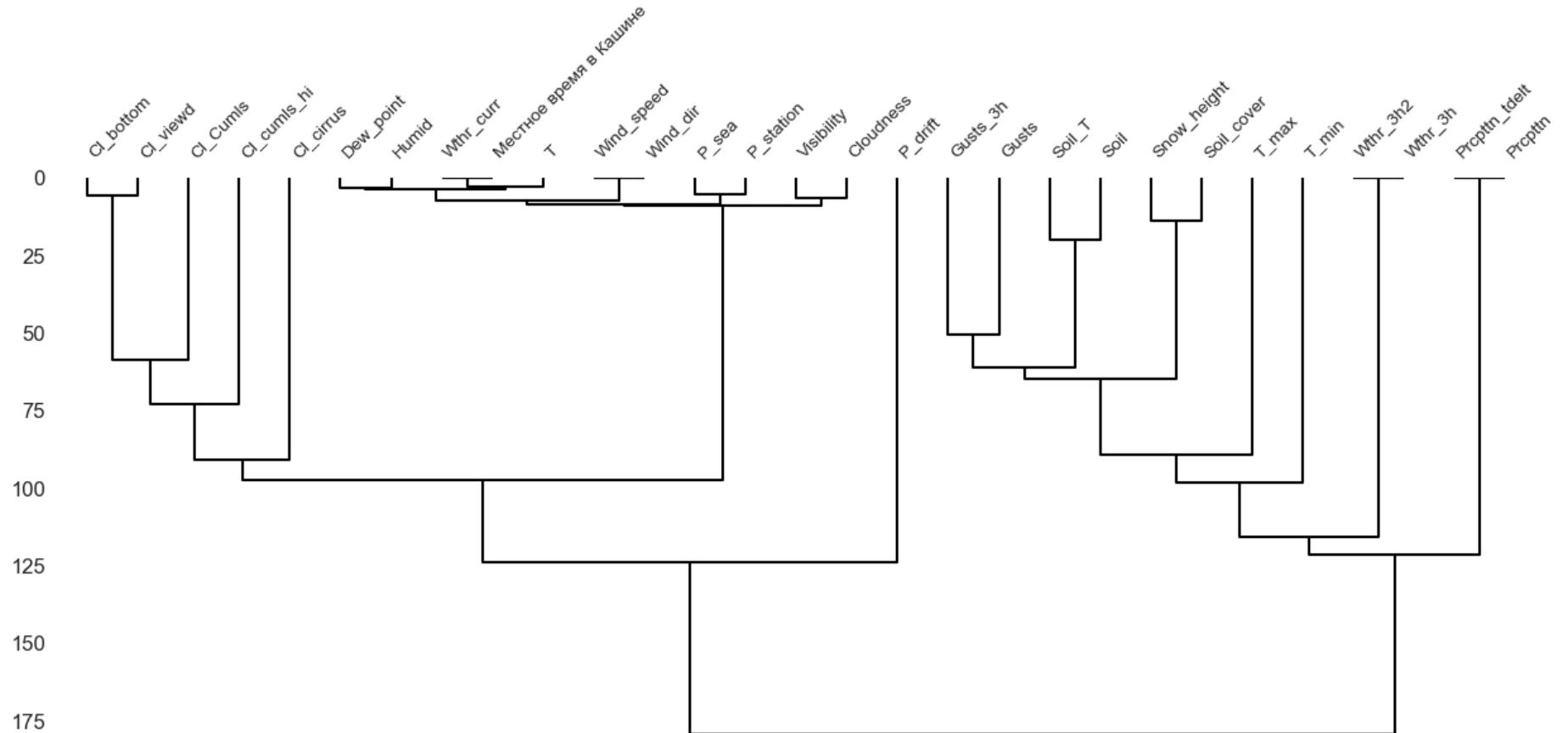
Out[37]:
<AxesSubplot:>



Архив метеостанции df_Kashyn

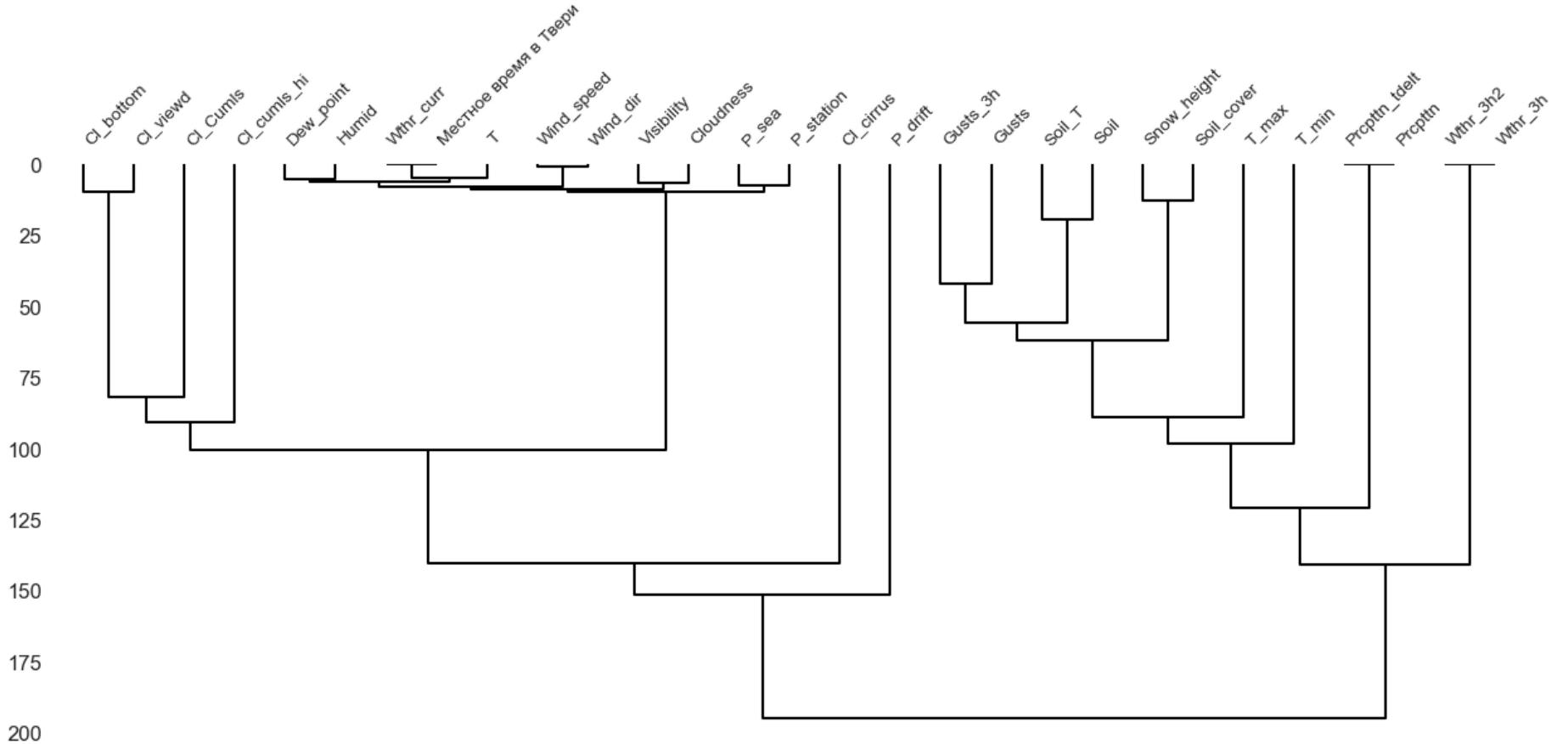
<AxesSubplot:>

Out[37]:



Архив метеостанции df_Tver

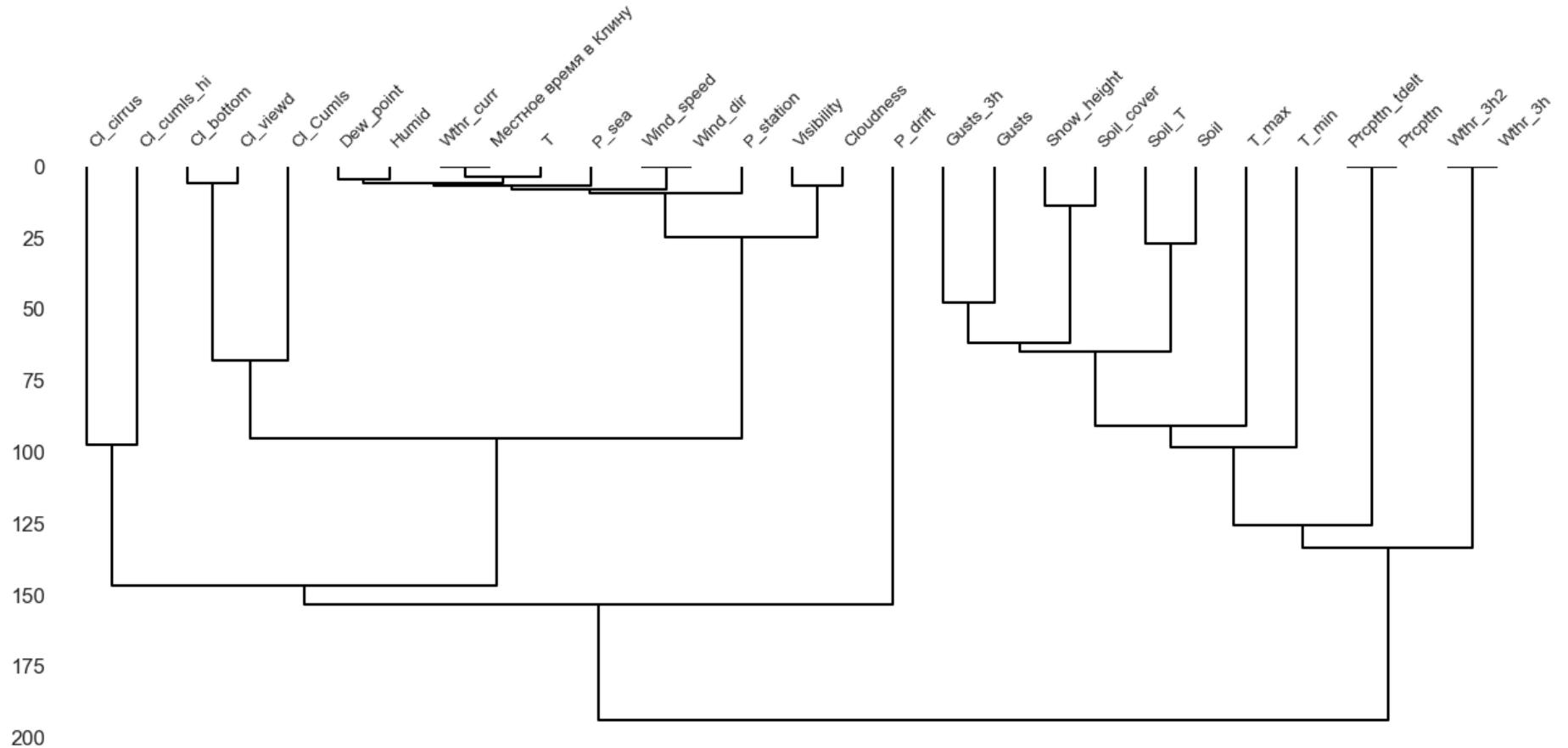
Out[37]:



Архив метеостанции df_Klin

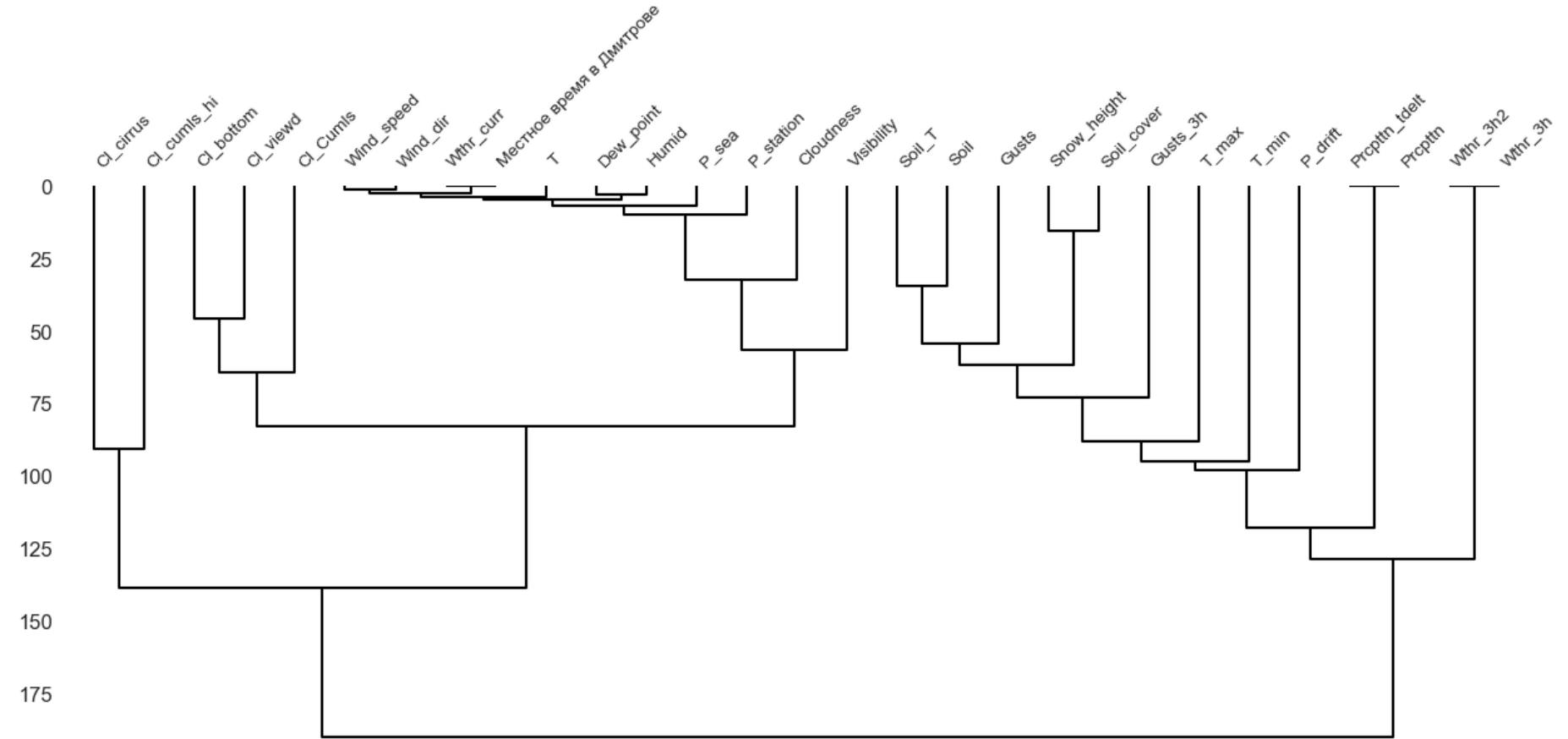
<AxesSubplot:>

Out[37]:



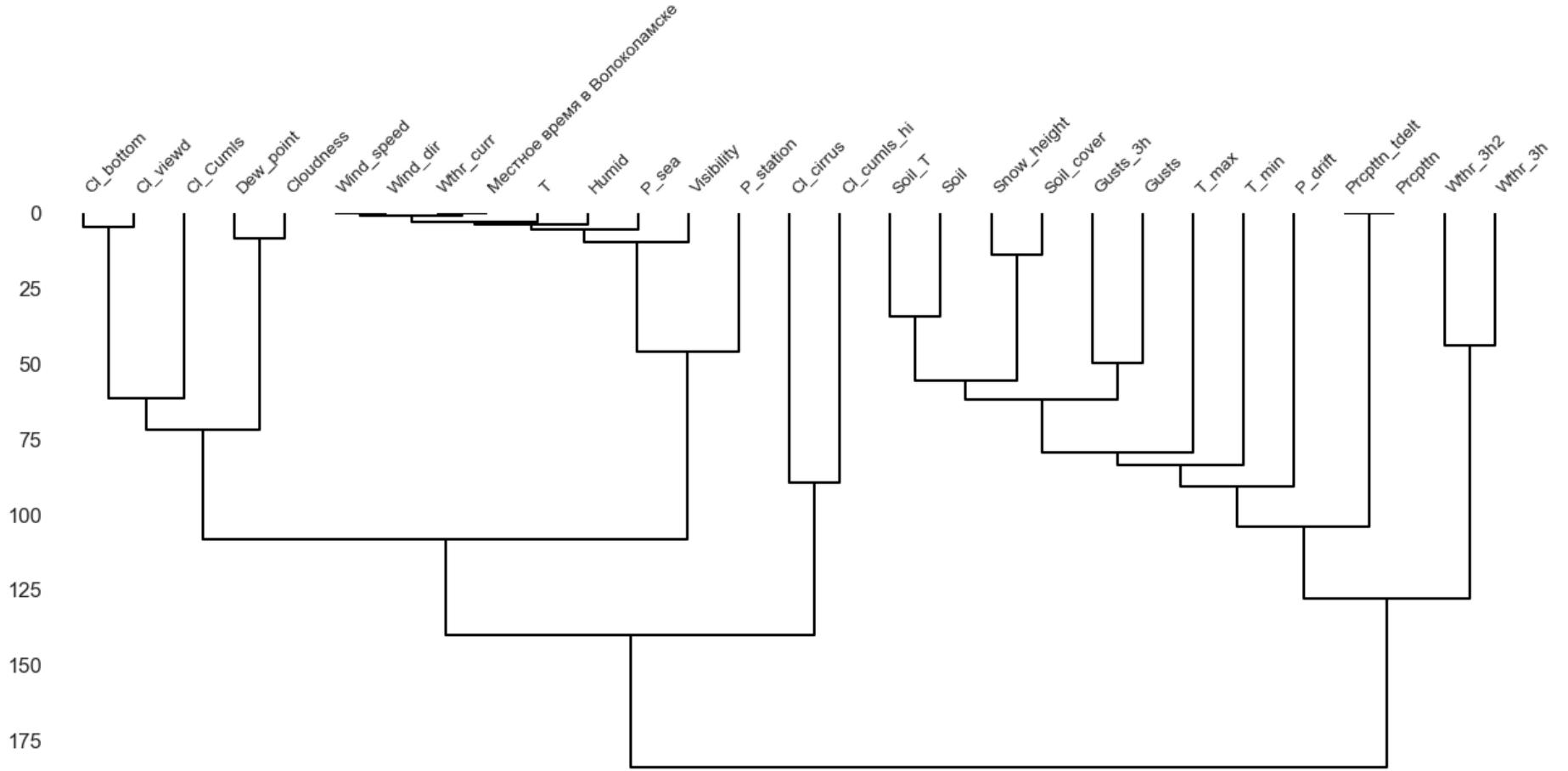
Архив метеостанции df_Dmitrov

Out[37]: <AxesSubplot:>



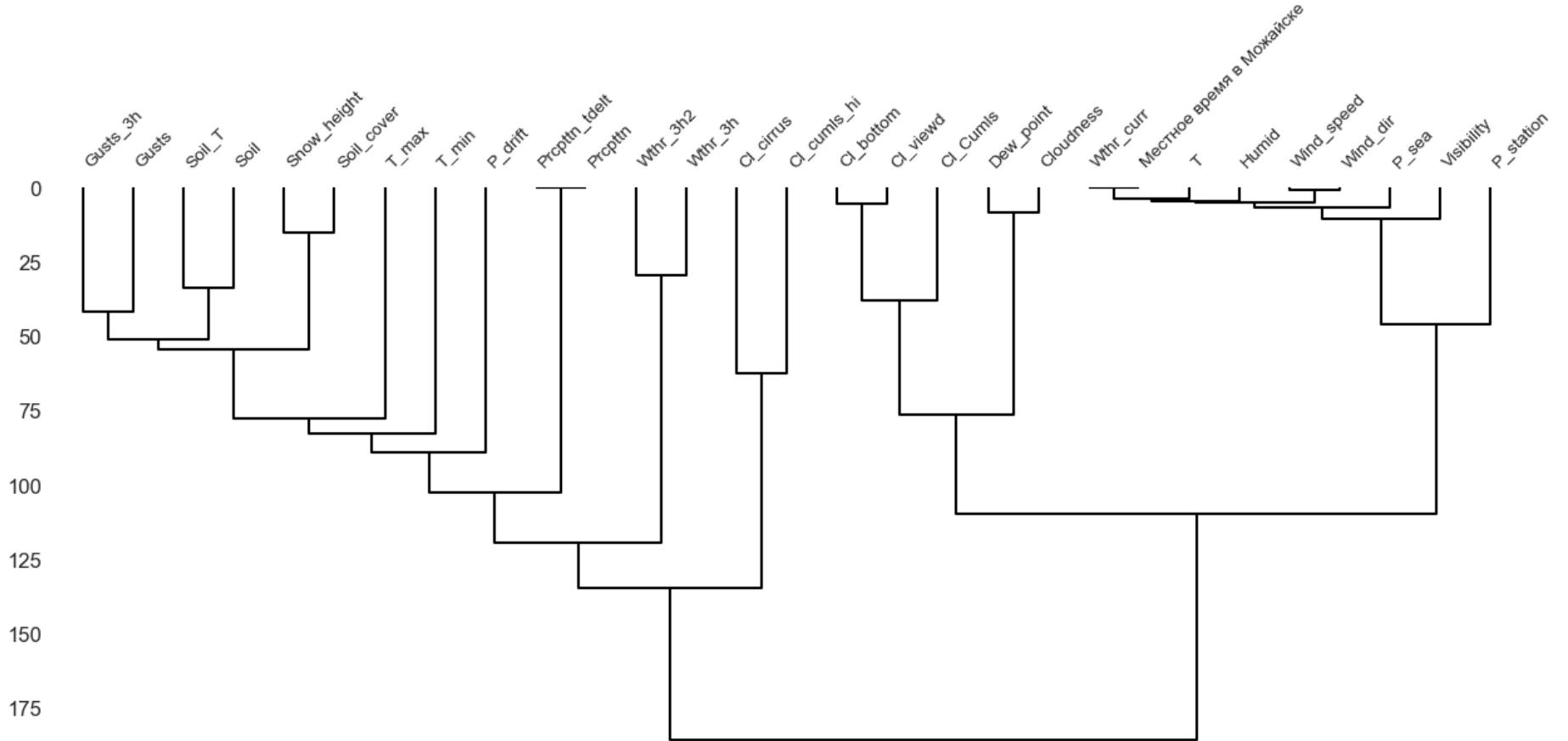
Архив метеостанции df_Volokolamsk

Out[37]:
<AxesSubplot:>



Архив метеостанции df_Mozhaisk

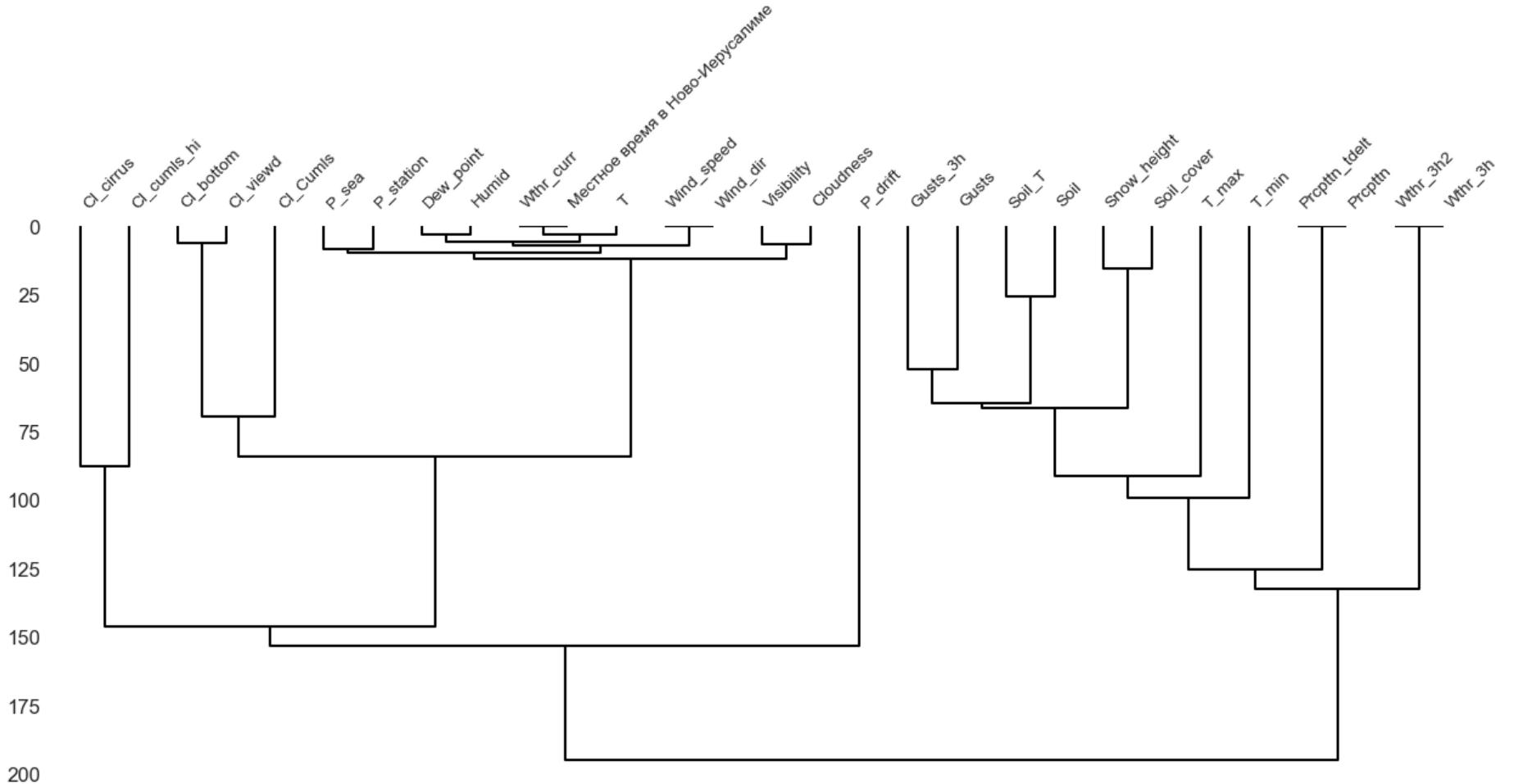
Out[37]:



Архив метеостанции df_N_Jerusalem

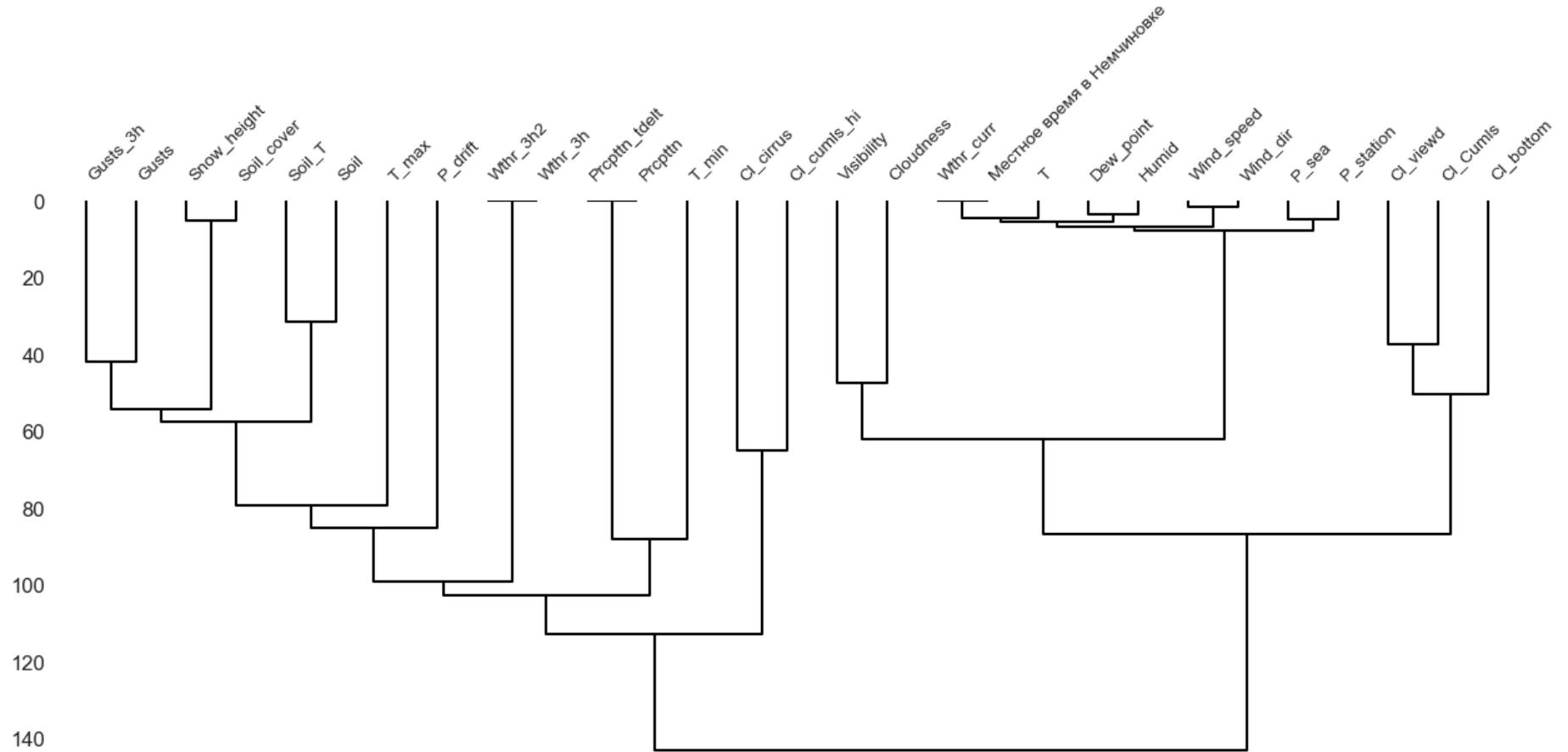
<AxesSubplot:>

Out[37]:



Архив метеостанции df_Nemchinovka

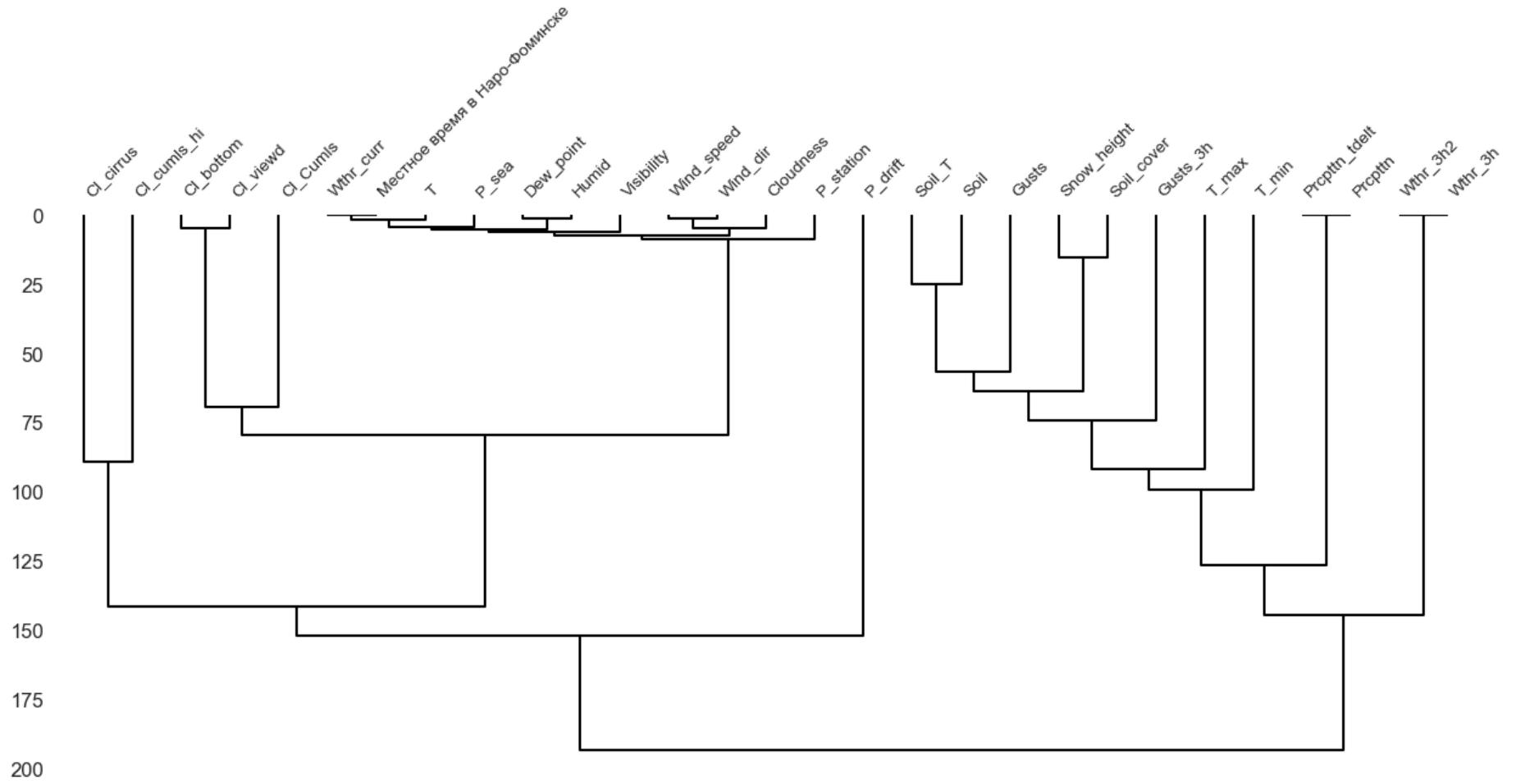
Out[37]:
<AxesSubplot:>



Архив метеостанции df_Naro_Fominsk

<AxesSubplot:>

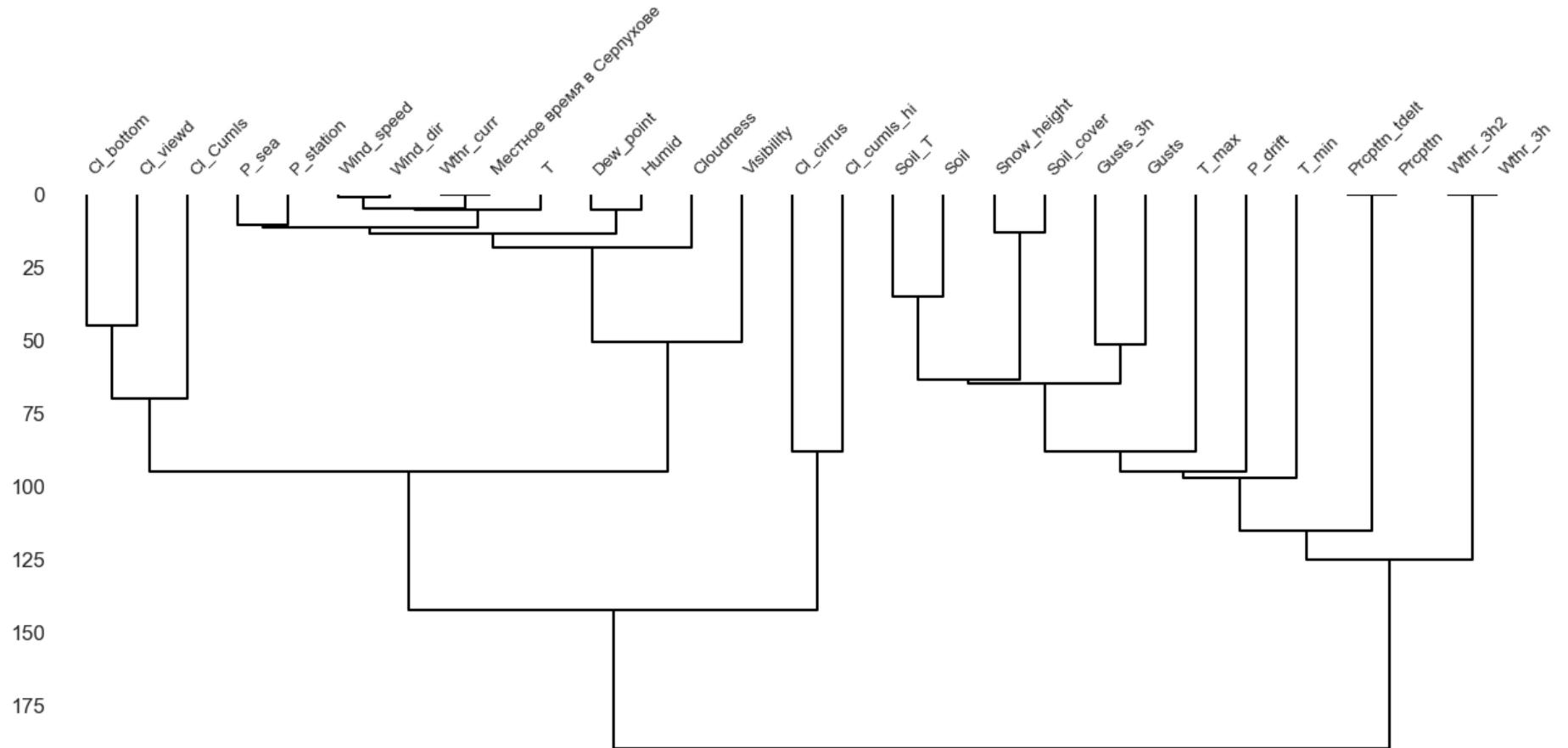
Out[37]:



Архив метеостанции df_Serpukhov

<AxesSubplot:>

Out[37]:



ВЫВОД

Дедограммы не выявляют иных зависимостей в пропущенных данных, кроме указанных выше.

ОБЩИЙ ВЫВОД ПО ПОДРАЗДЕЛУ "0.3.1.3. Информация о пропущенных данных\"

1. Большое количество пропусков в начале истории наблюдений для большинства станций: для построения модели начальные даты архивов придется увеличить.
2. Волоколамск, Можайск и Серпухов имеют относительно более низкое качество архивов по сравнению с другими станциями наблюдения: эти архивы для построения модели использовать менее желательно
3. Волоколамск, Можайск имеют длительные периоды пробелов в фиксации данных, данные по ним можно будет использовать только ограниченно, например, на этапе валидации.

1.1.2. Уникальные значения параметров

```
In [38]: for i in range(1, 29): # по количеству столбцов в DFs
    print(f'Уникальные значения всех датафреймов для столбца {data.columns[i]}')
    ser_column = pd.Series(data=None, dtype = 'object') # инициализируем пустую серию
    for name, data in dict_df_data.items(): # по элементам словаря с данными DFs
        # добавим к серии ser_column серию из уникальных значений текущего столбца текущего датафрейма
        ser_column = pd.concat([ser_column, pd.Series(pd.unique(data.iloc[:,i]))])
    # невозможно упорядочить столбцы, содержащие и строковые и числовые значения вперемешку
    # произведём упорядочивание только там, где возможно (данные однотипны и сравнимы между собой):
    try:
        ser_column = ser_column.sort_values()
    except:
        pass
    ser_column.unique() # выведем уникальные значения из всей полученной серии -> np.array
del ser_column # удалим временную серию, для освобождения памяти
```

Уникальные значения всех датафреймов для столбца T

```
Out[38]: array([-37.5, -37.2, -36.9, -36.7, -36.5, -36.4, -35.9, -35.8, -35.7,
 -35.6, -35.5, -35.4, -35.3, -35.2, -35.1, -35. , -34.9, -34.8,
 -34.7, -34.6, -34.5, -34.4, -34.3, -34.2, -34.1, -34. , -33.9,
 -33.8, -33.7, -33.6, -33.5, -33.4, -33.3, -33.2, -33.1, -33. ,
 -32.9, -32.8, -32.7, -32.6, -32.5, -32.4, -32.3, -32.2, -32.1,
 -32. , -31.9, -31.8, -31.7, -31.6, -31.5, -31.4, -31.3, -31.2,
 -31.1, -31. , -30.9, -30.8, -30.7, -30.6, -30.5, -30.4, -30.3,
 -30.2, -30.1, -30. , -29.9, -29.8, -29.7, -29.6, -29.5, -29.4,
 -29.3, -29.2, -29.1, -29. , -28.9, -28.8, -28.7, -28.6, -28.5,
 -28.4, -28.3, -28.2, -28.1, -28. , -27.9, -27.8, -27.7, -27.6,
 -27.5, -27.4, -27.3, -27.2, -27.1, -27. , -26.9, -26.8, -26.7,
 -26.6, -26.5, -26.4, -26.3, -26.2, -26.1, -26. , -25.9, -25.8,
 -25.7, -25.6, -25.5, -25.4, -25.3, -25.2, -25.1, -25. , -24.9,
 -24.8, -24.7, -24.6, -24.5, -24.4, -24.3, -24.2, -24.1, -24. ,
 -23.9, -23.8, -23.7, -23.6, -23.5, -23.4, -23.3, -23.2, -23.1,
 -23. , -22.9, -22.8, -22.7, -22.6, -22.5, -22.4, -22.3, -22.2,
 -22.1, -22. , -21.9, -21.8, -21.7, -21.6, -21.5, -21.4, -21.3,
 -21.2, -21.1, -21. , -20.9, -20.8, -20.7, -20.6, -20.5, -20.4,
 -20.3, -20.2, -20.1, -20. , -19.9, -19.8, -19.7, -19.6, -19.5,
 -19.4, -19.3, -19.2, -19.1, -19. , -18.9, -18.8, -18.7, -18.6,
 -18.5, -18.4, -18.3, -18.2, -18.1, -18. , -17.9, -17.8, -17.7,
 -17.6, -17.5, -17.4, -17.3, -17.2, -17.1, -17. , -16.9, -16.8,
 -16.7, -16.6, -16.5, -16.4, -16.3, -16.2, -16.1, -16. , -15.9,
 -15.8, -15.7, -15.6, -15.5, -15.4, -15.3, -15.2, -15.1, -15. ,
 -14.9, -14.8, -14.7, -14.6, -14.5, -14.4, -14.3, -14.2, -14.1,
 -14. , -13.9, -13.8, -13.7, -13.6, -13.5, -13.4, -13.3, -13.2,
 -13.1, -13. , -12.9, -12.8, -12.7, -12.6, -12.5, -12.4, -12.3,
 -12.2, -12.1, -12. , -11.9, -11.8, -11.7, -11.6, -11.5, -11.4,
 -11.3, -11.2, -11.1, -11. , -10.9, -10.8, -10.7, -10.6, -10.5,
 -10.4, -10.3, -10.2, -10.1, -10. , -9.9, -9.8, -9.7, -9.6,
 -9.5, -9.4, -9.3, -9.2, -9.1, -9. , -8.9, -8.8, -8.7,
 -8.6, -8.5, -8.4, -8.3, -8.2, -8.1, -8. , -7.9, -7.8,
 -7.7, -7.6, -7.5, -7.4, -7.3, -7.2, -7.1, -7. , -6.9,
 -6.8, -6.7, -6.6, -6.5, -6.4, -6.3, -6.2, -6.1, -6. ,
 -5.9, -5.8, -5.7, -5.6, -5.5, -5.4, -5.3, -5.2, -5.1,
 -5. , -4.9, -4.8, -4.7, -4.6, -4.5, -4.4, -4.3, -4.2,
 -4.1, -4. , -3.9, -3.8, -3.7, -3.6, -3.5, -3.4, -3.3,
 -3.2, -3.1, -3. , -2.9, -2.8, -2.7, -2.6, -2.5, -2.4,
 -2.3, -2.2, -2.1, -2. , -1.9, -1.8, -1.7, -1.6, -1.5,
 -1.4, -1.3, -1.2, -1.1, -1. , -0.9, -0.8, -0.7, -0.6,
 -0.5, -0.4, -0.3, -0.2, -0.1, 0. , 0.1, 0.2, 0.3,
 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,
 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1,
 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3. ,
```

```
3.1,    3.2,    3.3,    3.4,    3.5,    3.6,    3.7,    3.8,    3.9,  
4. ,     4.1,    4.2,    4.3,    4.4,    4.5,    4.6,    4.7,    4.8,  
4.9,    5. ,     5.1,    5.2,    5.3,    5.4,    5.5,    5.6,    5.7,  
5.8,    5.9,    6. ,     6.1,    6.2,    6.3,    6.4,    6.5,    6.6,  
6.7,    6.8,    6.9,    7. ,     7.1,    7.2,    7.3,    7.4,    7.5,  
7.6,    7.7,    7.8,    7.9,    8. ,     8.1,    8.2,    8.3,    8.4,  
8.5,    8.6,    8.7,    8.8,    8.9,    9. ,     9.1,    9.2,    9.3,  
9.4,    9.5,    9.6,    9.7,    9.8,    9.9,    10. ,    10.1,   10.2,  
10.3,   10.4,   10.5,   10.6,   10.7,   10.8,   10.9,   11. ,    11.1,  
11.2,   11.3,   11.4,   11.5,   11.6,   11.7,   11.8,   11.9,   12. ,  
12.1,   12.2,   12.3,   12.4,   12.5,   12.6,   12.7,   12.8,   12.9,  
13. ,    13.1,   13.2,   13.3,   13.4,   13.5,   13.6,   13.7,   13.8,  
13.9,   14. ,    14.1,   14.2,   14.3,   14.4,   14.5,   14.6,   14.7,  
14.8,   14.9,   15. ,    15.1,   15.2,   15.3,   15.4,   15.5,   15.6,  
15.7,   15.8,   15.9,   16. ,    16.1,   16.2,   16.3,   16.4,   16.5,  
16.6,   16.7,   16.8,   16.9,   17. ,    17.1,   17.2,   17.3,   17.4,  
17.5,   17.6,   17.7,   17.8,   17.9,   18. ,    18.1,   18.2,   18.3,  
18.4,   18.5,   18.6,   18.7,   18.8,   18.9,   19. ,    19.1,   19.2,  
19.3,   19.4,   19.5,   19.6,   19.7,   19.8,   19.9,   20. ,    20.1,  
20.2,   20.3,   20.4,   20.5,   20.6,   20.7,   20.8,   20.9,   21. ,  
21.1,   21.2,   21.3,   21.4,   21.5,   21.6,   21.7,   21.8,   21.9,  
22. ,    22.1,   22.2,   22.3,   22.4,   22.5,   22.6,   22.7,   22.8,  
22.9,   23. ,    23.1,   23.2,   23.3,   23.4,   23.5,   23.6,   23.7,  
23.8,   23.9,   24. ,    24.1,   24.2,   24.3,   24.4,   24.5,   24.6,  
24.7,   24.8,   24.9,   25. ,    25.1,   25.2,   25.3,   25.4,   25.5,  
25.6,   25.7,   25.8,   25.9,   26. ,    26.1,   26.2,   26.3,   26.4,  
26.5,   26.6,   26.7,   26.8,   26.9,   27. ,    27.1,   27.2,   27.3,  
27.4,   27.5,   27.6,   27.7,   27.8,   27.9,   28. ,    28.1,   28.2,  
28.3,   28.4,   28.5,   28.6,   28.7,   28.8,   28.9,   29. ,    29.1,  
29.2,   29.3,   29.4,   29.5,   29.6,   29.7,   29.8,   29.9,   30. ,  
30.1,   30.2,   30.3,   30.4,   30.5,   30.6,   30.7,   30.8,   30.9,  
31. ,    31.1,   31.2,   31.3,   31.4,   31.5,   31.6,   31.7,   31.8,  
31.9,   32. ,    32.1,   32.2,   32.3,   32.4,   32.5,   32.6,   32.7,  
32.8,   32.9,   33. ,    33.1,   33.2,   33.3,   33.4,   33.5,   33.6,  
33.7,   33.8,   33.9,   34. ,    34.1,   34.2,   34.3,   34.4,   34.5,  
34.6,   34.7,   34.8,   34.9,   35. ,    35.1,   35.2,   35.3,   35.4,  
35.5,   35.6,   35.7,   35.8,   35.9,   36. ,    36.1,   36.2,   36.3,  
36.4,   36.5,   36.6,   36.7,   36.8,   36.9,   37. ,    37.1,   37.2,  
37.3,   37.4,   37.5,   37.6,   37.7,   37.9,   38. ,    38.1,   38.6,  
38.7,   nan])
```

Уникальные значения всех датафреймов для столбца P_station

```
Out[38]: array([ 77.9, 668.4, 671. , 707.8, 708. , 708.1, 708.2, 708.4, 708.6,
 708.8, 708.9, 709. , 709.3, 709.4, 709.6, 709.7, 709.8, 709.9,
 710. , 710.1, 710.2, 710.3, 710.4, 710.5, 710.6, 710.7, 710.8,
 710.9, 711. , 711.1, 711.2, 711.4, 711.5, 711.6, 711.7, 711.8,
 711.9, 712. , 712.1, 712.2, 712.3, 712.4, 712.5, 712.6, 712.7,
 712.8, 712.9, 713. , 713.1, 713.2, 713.3, 713.4, 713.5, 713.6,
 713.7, 713.8, 713.9, 714. , 714.1, 714.2, 714.3, 714.4, 714.5,
 714.6, 714.7, 714.8, 714.9, 715. , 715.1, 715.2, 715.3, 715.4,
 715.5, 715.6, 715.7, 715.8, 715.9, 716. , 716.1, 716.2, 716.3,
 716.4, 716.5, 716.6, 716.7, 716.8, 716.9, 717. , 717.1, 717.2,
 717.3, 717.4, 717.5, 717.6, 717.7, 717.8, 717.9, 718. , 718.1,
 718.2, 718.3, 718.4, 718.5, 718.6, 718.7, 718.8, 718.9, 719. ,
 719.1, 719.2, 719.3, 719.4, 719.5, 719.6, 719.7, 719.8, 719.9,
 720. , 720.1, 720.2, 720.3, 720.4, 720.5, 720.6, 720.7, 720.8,
 720.9, 721. , 721.1, 721.2, 721.3, 721.4, 721.5, 721.6, 721.7,
 721.8, 721.9, 722. , 722.1, 722.2, 722.3, 722.4, 722.5, 722.6,
 722.7, 722.8, 722.9, 723. , 723.1, 723.2, 723.3, 723.4, 723.5,
 723.6, 723.7, 723.8, 723.9, 724. , 724.1, 724.2, 724.3, 724.4,
 724.5, 724.6, 724.7, 724.8, 724.9, 725. , 725.1, 725.2, 725.3,
 725.4, 725.5, 725.6, 725.7, 725.8, 725.9, 726. , 726.1, 726.2,
 726.3, 726.4, 726.5, 726.6, 726.7, 726.8, 726.9, 727. , 727.1,
 727.2, 727.3, 727.4, 727.5, 727.6, 727.7, 727.8, 727.9, 728. ,
 728.1, 728.2, 728.3, 728.4, 728.5, 728.6, 728.7, 728.8, 728.9,
 729. , 729.1, 729.2, 729.3, 729.4, 729.5, 729.6, 729.7, 729.8,
 729.9, 730. , 730.1, 730.2, 730.3, 730.4, 730.5, 730.6, 730.7,
 730.8, 730.9, 731. , 731.1, 731.2, 731.3, 731.4, 731.5, 731.6,
 731.7, 731.8, 731.9, 732. , 732.1, 732.2, 732.3, 732.4, 732.5,
 732.6, 732.7, 732.8, 732.9, 733. , 733.1, 733.2, 733.3, 733.4,
 733.5, 733.6, 733.7, 733.8, 733.9, 734. , 734.1, 734.2, 734.3,
 734.4, 734.5, 734.6, 734.7, 734.8, 734.9, 735. , 735.1, 735.2,
 735.3, 735.4, 735.5, 735.6, 735.7, 735.8, 735.9, 736. , 736.1,
 736.2, 736.3, 736.4, 736.5, 736.6, 736.7, 736.8, 736.9, 737. ,
 737.1, 737.2, 737.3, 737.4, 737.5, 737.6, 737.7, 737.8, 737.9,
 738. , 738.1, 738.2, 738.3, 738.4, 738.5, 738.6, 738.7, 738.8,
 738.9, 739. , 739.1, 739.2, 739.3, 739.4, 739.5, 739.6, 739.7,
 739.8, 739.9, 740. , 740.1, 740.2, 740.3, 740.4, 740.5, 740.6,
 740.7, 740.8, 740.9, 741. , 741.1, 741.2, 741.3, 741.4, 741.5,
 741.6, 741.7, 741.8, 741.9, 742. , 742.1, 742.2, 742.3, 742.4,
 742.5, 742.6, 742.7, 742.8, 742.9, 743. , 743.1, 743.2, 743.3,
 743.4, 743.5, 743.6, 743.7, 743.8, 743.9, 744. , 744.1, 744.2,
 744.3, 744.4, 744.5, 744.6, 744.7, 744.8, 744.9, 745. , 745.1,
 745.2, 745.3, 745.4, 745.5, 745.6, 745.7, 745.8, 745.9, 746. ,
 746.1, 746.2, 746.3, 746.4, 746.5, 746.6, 746.7, 746.8, 746.9,
 747. , 747.1, 747.2, 747.3, 747.4, 747.5, 747.6, 747.7, 747.8,
```

747.9, 748. , 748.1, 748.2, 748.3, 748.4, 748.5, 748.6, 748.7,
748.8, 748.9, 749. , 749.1, 749.2, 749.3, 749.4, 749.5, 749.6,
749.7, 749.8, 749.9, 750. , 750.1, 750.2, 750.3, 750.4, 750.5,
750.6, 750.7, 750.8, 750.9, 751. , 751.1, 751.2, 751.3, 751.4,
751.5, 751.6, 751.7, 751.8, 751.9, 752. , 752.1, 752.2, 752.3,
752.4, 752.5, 752.6, 752.7, 752.8, 752.9, 753. , 753.1, 753.2,
753.3, 753.4, 753.5, 753.6, 753.7, 753.8, 753.9, 754. , 754.1,
754.2, 754.3, 754.4, 754.5, 754.6, 754.7, 754.8, 754.9, 755. ,
755.1, 755.2, 755.3, 755.4, 755.5, 755.6, 755.7, 755.8, 755.9,
756. , 756.1, 756.2, 756.3, 756.4, 756.5, 756.6, 756.7, 756.8,
756.9, 757. , 757.1, 757.2, 757.3, 757.4, 757.5, 757.6, 757.7,
757.8, 757.9, 758. , 758.1, 758.2, 758.3, 758.4, 758.5, 758.6,
758.7, 758.8, 758.9, 759. , 759.1, 759.2, 759.3, 759.4, 759.5,
759.6, 759.7, 759.8, 759.9, 760. , 760.1, 760.2, 760.3, 760.4,
760.5, 760.6, 760.7, 760.8, 760.9, 761. , 761.1, 761.2, 761.3,
761.4, 761.5, 761.6, 761.7, 761.8, 761.9, 762. , 762.1, 762.2,
762.3, 762.4, 762.5, 762.6, 762.7, 762.8, 762.9, 763. , 763.1,
763.2, 763.3, 763.4, 763.5, 763.6, 763.7, 763.8, 763.9, 764. ,
764.1, 764.2, 764.3, 764.4, 764.5, 764.6, 764.7, 764.8, 764.9,
765. , 765.1, 765.2, 765.3, 765.4, 765.5, 765.6, 765.7, 765.8,
765.9, 766. , 766.1, 766.2, 766.3, 766.4, 766.5, 766.6, 766.7,
766.8, 766.9, 767. , 767.1, 767.2, 767.3, 767.4, 767.5, 767.6,
767.7, 767.8, 767.9, 768. , 768.1, 768.2, 768.3, 768.4, 768.5,
768.6, 768.7, 768.8, 768.9, 769. , 769.1, 769.2, 769.3, 769.4,
769.5, 769.6, 769.7, 769.8, 769.9, 770. , 770.1, 770.2, 770.3,
770.4, 770.5, 770.6, 770.7, 770.8, 770.9, 771. , 771.1, 771.2,
771.3, 771.4, 771.5, 771.6, 771.7, 771.8, 771.9, 772. , 772.1,
772.2, 772.3, 772.4, 772.5, 772.6, 772.7, 772.8, 772.9, 773. ,
773.1, 773.2, 773.3, 773.4, 773.5, 773.6, 773.7, 773.8, 773.9,
774. , 774.1, 774.2, 774.3, 774.4, 774.5, 774.6, 774.7, 774.8,
774.9, 775. , 775.1, 775.2, 775.3, 775.4, 775.5, 775.6, 775.7,
775.8, 775.9, 776. , 776.1, 776.2, 776.3, 776.4, 776.5, 776.6,
776.7, 776.8, 776.9, 777. , 777.1, 777.2, 777.3, 777.4, 777.6,
777.8, 777.9, 778.1, 778.4, 778.7, 779. , 779.2, 779.5, 779.6,
780.2, 780.4, 781. , 781.1, nan])

Уникальные значения всех датафреймов для столбца P_sea

```
Out[38]: array([ 93.8, 684.5, 718.6, 723. , 723.8, 723.9, 724. , 724.3, 724.4,
 724.5, 724.6, 724.9, 725. , 725.2, 725.3, 725.4, 725.5, 725.6,
 725.7, 725.8, 725.9, 726. , 726.1, 726.2, 726.3, 726.4, 726.5,
 726.6, 726.7, 726.8, 726.9, 727. , 727.1, 727.2, 727.3, 727.4,
 727.5, 727.6, 727.7, 727.8, 727.9, 728. , 728.1, 728.2, 728.3,
 728.4, 728.5, 728.6, 728.7, 728.8, 728.9, 729. , 729.1, 729.2,
 729.3, 729.4, 729.5, 729.6, 729.7, 729.8, 729.9, 730. , 730.1,
 730.2, 730.3, 730.4, 730.5, 730.6, 730.7, 730.8, 730.9, 731. ,
 731.1, 731.2, 731.3, 731.4, 731.5, 731.6, 731.7, 731.8, 731.9,
 732. , 732.1, 732.2, 732.3, 732.4, 732.5, 732.6, 732.7, 732.8,
 732.9, 733. , 733.1, 733.2, 733.3, 733.4, 733.5, 733.6, 733.7,
 733.8, 733.9, 734. , 734.1, 734.2, 734.3, 734.4, 734.5, 734.6,
 734.7, 734.8, 734.9, 735. , 735.1, 735.2, 735.3, 735.4, 735.5,
 735.6, 735.7, 735.8, 735.9, 736. , 736.1, 736.2, 736.3, 736.4,
 736.5, 736.6, 736.7, 736.8, 736.9, 737. , 737.1, 737.2, 737.3,
 737.4, 737.5, 737.6, 737.7, 737.8, 737.9, 738. , 738.1, 738.2,
 738.3, 738.4, 738.5, 738.6, 738.7, 738.8, 738.9, 739. , 739.1,
 739.2, 739.3, 739.4, 739.5, 739.6, 739.7, 739.8, 739.9, 740. ,
 740.1, 740.2, 740.3, 740.4, 740.5, 740.6, 740.7, 740.8, 740.9,
 741. , 741.1, 741.2, 741.3, 741.4, 741.5, 741.6, 741.7, 741.8,
 741.9, 742. , 742.1, 742.2, 742.3, 742.4, 742.5, 742.6, 742.7,
 742.8, 742.9, 743. , 743.1, 743.2, 743.3, 743.4, 743.5, 743.6,
 743.7, 743.8, 743.9, 744. , 744.1, 744.2, 744.3, 744.4, 744.5,
 744.6, 744.7, 744.8, 744.9, 745. , 745.1, 745.2, 745.3, 745.4,
 745.5, 745.6, 745.7, 745.8, 745.9, 746. , 746.1, 746.2, 746.3,
 746.4, 746.5, 746.6, 746.7, 746.8, 746.9, 747. , 747.1, 747.2,
 747.3, 747.4, 747.5, 747.6, 747.7, 747.8, 747.9, 748. , 748.1,
 748.2, 748.3, 748.4, 748.5, 748.6, 748.7, 748.8, 748.9, 749. ,
 749.1, 749.2, 749.3, 749.4, 749.5, 749.6, 749.7, 749.8, 749.9,
 750. , 750.1, 750.2, 750.3, 750.4, 750.5, 750.6, 750.7, 750.8,
 750.9, 751. , 751.1, 751.2, 751.3, 751.4, 751.5, 751.6, 751.7,
 751.8, 751.9, 752. , 752.1, 752.2, 752.3, 752.4, 752.5, 752.6,
 752.7, 752.8, 752.9, 753. , 753.1, 753.2, 753.3, 753.4, 753.5,
 753.6, 753.7, 753.8, 753.9, 754. , 754.1, 754.2, 754.3, 754.4,
 754.5, 754.6, 754.7, 754.8, 754.9, 755. , 755.1, 755.2, 755.3,
 755.4, 755.5, 755.6, 755.7, 755.8, 755.9, 756. , 756.1, 756.2,
 756.3, 756.4, 756.5, 756.6, 756.7, 756.8, 756.9, 757. , 757.1,
 757.2, 757.3, 757.4, 757.5, 757.6, 757.7, 757.8, 757.9, 758. ,
 758.1, 758.2, 758.3, 758.4, 758.5, 758.6, 758.7, 758.8, 758.9,
 759. , 759.1, 759.2, 759.3, 759.4, 759.5, 759.6, 759.7, 759.8,
 759.9, 760. , 760.1, 760.2, 760.3, 760.4, 760.5, 760.6, 760.7,
 760.8, 760.9, 761. , 761.1, 761.2, 761.3, 761.4, 761.5, 761.6,
 761.7, 761.8, 761.9, 762. , 762.1, 762.2, 762.3, 762.4, 762.5,
 762.6, 762.7, 762.8, 762.9, 763. , 763.1, 763.2, 763.3, 763.4,
```

763.5, 763.6, 763.7, 763.8, 763.9, 764. , 764.1, 764.2, 764.3,
764.4, 764.5, 764.6, 764.7, 764.8, 764.9, 765. , 765.1, 765.2,
765.3, 765.4, 765.5, 765.6, 765.7, 765.8, 765.9, 766. , 766.1,
766.2, 766.3, 766.4, 766.5, 766.6, 766.7, 766.8, 766.9, 767. ,
767.1, 767.2, 767.3, 767.4, 767.5, 767.6, 767.7, 767.8, 767.9,
768. , 768.1, 768.2, 768.3, 768.4, 768.5, 768.6, 768.7, 768.8,
768.9, 769. , 769.1, 769.2, 769.3, 769.4, 769.5, 769.6, 769.7,
769.8, 769.9, 770. , 770.1, 770.2, 770.3, 770.4, 770.5, 770.6,
770.7, 770.8, 770.9, 771. , 771.1, 771.2, 771.3, 771.4, 771.5,
771.6, 771.7, 771.8, 771.9, 772. , 772.1, 772.2, 772.3, 772.4,
772.5, 772.6, 772.7, 772.8, 772.9, 773. , 773.1, 773.2, 773.3,
773.4, 773.5, 773.6, 773.7, 773.8, 773.9, 774. , 774.1, 774.2,
774.3, 774.4, 774.5, 774.6, 774.7, 774.8, 774.9, 775. , 775.1,
775.2, 775.3, 775.4, 775.5, 775.6, 775.7, 775.8, 775.9, 776. ,
776.1, 776.2, 776.3, 776.4, 776.5, 776.6, 776.7, 776.8, 776.9,
777. , 777.1, 777.2, 777.3, 777.4, 777.5, 777.6, 777.7, 777.8,
777.9, 778. , 778.1, 778.2, 778.3, 778.4, 778.5, 778.6, 778.7,
778.8, 778.9, 779. , 779.1, 779.2, 779.3, 779.4, 779.5, 779.6,
779.7, 779.8, 779.9, 780. , 780.1, 780.2, 780.3, 780.4, 780.5,
780.6, 780.7, 780.8, 780.9, 781. , 781.1, 781.2, 781.3, 781.4,
781.5, 781.6, 781.7, 781.8, 781.9, 782. , 782.1, 782.2, 782.3,
782.4, 782.5, 782.6, 782.7, 782.8, 782.9, 783. , 783.1, 783.2,
783.3, 783.4, 783.5, 783.6, 783.7, 783.8, 783.9, 784. , 784.1,
784.2, 784.3, 784.4, 784.5, 784.6, 784.7, 784.8, 784.9, 785. ,
785.1, 785.2, 785.3, 785.4, 785.5, 785.6, 785.7, 785.8, 785.9,
786. , 786.1, 786.2, 786.3, 786.4, 786.5, 786.6, 786.7, 786.8,
786.9, 787. , 787.1, 787.2, 787.3, 787.4, 787.5, 787.6, 787.7,
787.8, 787.9, 788. , 788.1, 788.2, 788.3, 788.4, 788.5, 788.6,
788.7, 788.8, 788.9, 789. , 789.1, 789.2, 789.3, 789.4, 789.5,
789.6, 789.7, 789.8, 789.9, 790. , 790.1, 790.2, 790.3, 790.4,
790.5, 790.6, 790.7, 790.8, 790.9, 791. , 791.1, 791.2, 791.3,
791.4, 791.5, 791.6, 791.7, 791.8, 791.9, 792. , 792.1, 792.2,
792.3, 792.4, 792.5, 792.6, 792.7, 792.8, 792.9, 793. , 793.1,
793.2, 793.3, 793.4, 793.5, 793.6, 793.7, 793.8, 793.9, 794. ,
794.1, 794.2, 794.3, 794.4, 794.5, 794.6, 794.7, 794.8, 795. ,
795.1, 795.2, 795.3, 795.4, 795.5, 795.6, 803.8, 810.8, 816.7,
824.3, 824.6, nan])

Уникальные значения всех датафреймов для столбца P_drift

```
Out[38]: array([-22.3, -14.9, -14.1, -13.9, -13. , -12.9, -12.7, -11.1, -10.8,
-9.7, -9.6, -9.2, -9. , -8.7, -8.6, -8.5, -8.4, -8.3,
-8.2, -8.1, -8. , -7.9, -7.8, -7.5, -7.3, -7.2, -7.1,
-7. , -6.9, -6.8, -6.7, -6.6, -6.5, -6.4, -6.3, -6.1,
-6. , -5.9, -5.8, -5.7, -5.6, -5.5, -5.4, -5.3, -5.2,
-5.1, -5. , -4.9, -4.8, -4.7, -4.6, -4.5, -4.4, -4.3,
-4.2, -4.1, -4. , -3.9, -3.8, -3.7, -3.6, -3.5, -3.4,
-3.3, -3.2, -3.1, -3. , -2.9, -2.8, -2.7, -2.6, -2.5,
-2.4, -2.3, -2.2, -2.1, -2. , -1.9, -1.8, -1.7, -1.6,
-1.5, -1.4, -1.3, -1.2, -1.1, -1. , -0.9, -0.8, -0.7,
-0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0. , 0.1, 0.2,
0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1,
1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. ,
2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9,
3. , 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8,
3.9, 4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7,
4.8, 4.9, 5. , 5.1, 5.2, 5.4, 5.5, 5.6, 5.8,
5.9, 6. , 6.1, 6.3, 6.4, 6.7, 6.8, 6.9, 7. ,
7.1, 7.4, 7.5, 7.7, 8.1, 8.2, 9. , 10.6, 12.8,
13.6, 14.4, 14.7, 15.8, 16. , 21.6, nan])
```

Уникальные значения всех датафреймов для столбца Humid

```
Out[38]: array([ 8.,  9., 10., 11., 12., 13., 14., 15., 16., 17., 18.,
19., 20., 21., 22., 23., 24., 25., 26., 27., 28., 29.,
30., 31., 32., 33., 34., 35., 36., 37., 38., 39., 40.,
41., 42., 43., 44., 45., 46., 47., 48., 49., 50., 51.,
52., 53., 54., 55., 56., 57., 58., 59., 60., 61., 62.,
63., 64., 65., 66., 67., 68., 69., 70., 71., 72., 73.,
74., 75., 76., 77., 78., 79., 80., 81., 82., 83., 84.,
85., 86., 87., 88., 89., 90., 91., 92., 93., 94., 95.,
96., 97., 98., 99., 100., nan])
```

Уникальные значения всех датафреймов для столбца Wind_dir

```
Out[38]: array(['Ветер, дующий с востока',
'Ветер, дующий с востоко-северо-востока',
'Ветер, дующий с востоко-юго-востока', 'Ветер, дующий с запада',
'Ветер, дующий с западо-северо-запада',
'Ветер, дующий с западо-юго-запада', 'Ветер, дующий с севера',
'Ветер, дующий с северо-востока', 'Ветер, дующий с северо-запада',
'Ветер, дующий с северо-северо-востока',
'Ветер, дующий с северо-северо-запада', 'Ветер, дующий с юга',
'Ветер, дующий с юго-востока', 'Ветер, дующий с юго-запада',
'Ветер, дующий с юго-юго-востока',
'Ветер, дующий с юго-юго-запада', 'Штиль, безветрие', nan],
dtype=object)
```

Уникальные значения всех датафреймов для столбца Wind_speed

```
Out[38]: array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10., 11., 12.,
   13., 14., 15., 16., 18., 20., 32., 35., 42., 50., nan])
    Уникальные значения всех датафреймов для столбца Gusts
Out[38]: array([ 5.,  6.,  7.,  8.,  9., 10., 11., 12., 13., 14., 15., 16., 17.,
   18., 19., 20., 21., 22., 23., nan])
    Уникальные значения всех датафреймов для столбца Gusts_3h
Out[38]: array([ 5.,  6.,  7., 10., 11., 12., 13., 14., 15., 16., 17., 18., 19.,
   20., 21., 22., 23., 24., 25., 26., 27., nan])
    Уникальные значения всех датафреймов для столбца Cloudness
Out[38]: array(['10% или менее, но не 0', '100%.', '20-30%.', '40%.', '50%.',
   '60%.', '70 - 80%.', '90 или более, но не 100%',
   'Небо не видно из-за тумана и/или других метеорологических явлений.',
   'Облаков нет.', nan], dtype=object)
    Уникальные значения всех датафреймов для столбца Wthr_curr
```


' Диаметр отложения при гололеде составляет 1 мм.',
' Диаметр отложения при гололеде составляет 10 мм.',
' Диаметр отложения при гололеде составляет 11 мм.',
' Диаметр отложения при гололеде составляет 13 мм.',
' Диаметр отложения при гололеде составляет 14 мм.',
' Диаметр отложения при гололеде составляет 2 мм.',
' Диаметр отложения при гололеде составляет 3 мм.',
' Диаметр отложения при гололеде составляет 4 мм.',
' Диаметр отложения при гололеде составляет 9 мм.',
' Диаметр смешанного отложения составляет 3 мм.',
'Алмазная пыль (с туманом или без него). ',
'Алмазная пыль (с туманом или без него). Диаметр изморозевого отложения составляет 4 мм.',
'Алмазная пыль (с туманом или без него). Диаметр отложения при гололеде составляет 1 мм.',
'Более или менее сплошной слой приземного или ледяного тумана на станции, на море или на суше, высотой не более 2 м над сушей или не более 10 м над морем. ',
'Видимость ухудшена дымом, например: от степных или лесных пожаров, промышленных предприятий или вулканическим пеплом. ',
'Видна молния, грома не слышно. ',
'Гроза (с осадками или без них). ',
'Гроза (с осадками или без них). Максимальный диаметр градин составляет 12 мм.',
'Гроза (с осадками или без них). Максимальный диаметр градин составляет 13 мм.',
'Гроза (с осадками или без них). Максимальный диаметр градин составляет 2 мм.',
'Гроза (с осадками или без них). Максимальный диаметр градин составляет 5 мм.',
'Гроза (с осадками или без них). Максимальный диаметр градин составляет 9 мм.',
'Гроза вместе с пыльной или песчаной бурей в срок наблюдения. ',
'Гроза сильная без града, но с дождем и/или снегом в срок наблюдения. ',
'Гроза слабая или умеренная без града, но с дождем и/или снегом в срок наблюдения. ',
'Гроза слабая или умеренная без града, но с дождем и/или снегом в срок наблюдения. Максимальный диаметр градин составляет 1 мм.',
'Гроза слабая или умеренная без града, но с дождем и/или снегом в срок наблюдения. Максимальный диаметр градин составляет 2 мм.',
'Гроза слабая или умеренная без града, но с дождем и/или снегом в срок наблюдения. Максимальный диаметр градин составляет 5 мм.',
'Гроза слабая или умеренная без града, но с дождем и/или снегом в срок наблюдения. Максимальный диаметр градин составляет 6 мм.',
'Гроза слабая или умеренная с градом в срок наблюдения. ',
'Гроза слабая или умеренная с градом в срок наблюдения. Максимальный диаметр градин составляет 3 мм.',
'Гроза, но без осадков, в срок наблюдения. ',
'Гроза, но без осадков, в срок наблюдения. Максимальный диаметр градин составляет 1 мм.',
'Гроза, но без осадков, в срок наблюдения. Максимальный диаметр градин составляет 10 мм.',
'Гроза, но без осадков, в срок наблюдения. Максимальный диаметр градин составляет 2 мм.',
'Гроза, но без осадков, в срок наблюдения. Максимальный диаметр градин составляет 3 мм.',
'Гроза, но без осадков, в срок наблюдения. Максимальный диаметр градин составляет 4 мм.',
'Дождь (незамерзающий) неливневый. ', 'Дождь замерзающий слабый. ',
'Дождь замерзающий слабый. Высота свежевыпавшего снега составляет 0.3 см. Диаметр отложения при гололеде составляет 2 м

'Дымка. Диаметр изморозевого отложения составляет 1 мм.',
'Дымка. Диаметр изморозевого отложения составляет 12 мм.',
'Дымка. Диаметр изморозевого отложения составляет 14 мм.',
'Дымка. Диаметр изморозевого отложения составляет 2 мм.',
'Дымка. Диаметр изморозевого отложения составляет 3 мм.',
'Дымка. Диаметр изморозевого отложения составляет 4 мм.',
'Дымка. Диаметр изморозевого отложения составляет 7 мм.',
'Дымка. Диаметр изморозевого отложения составляет 9 мм.',
'Дымка. Диаметр отложения мокрого снега составляет 1 мм.',
'Дымка. Диаметр отложения мокрого снега составляет 2 мм.',
'Дымка. Диаметр отложения мокрого снега составляет 24 мм.',
'Дымка. Диаметр отложения мокрого снега составляет 3 мм.',
'Дымка. Диаметр отложения мокрого снега составляет 4 мм.',
'Дымка. Диаметр отложения мокрого снега составляет 9 мм.',
'Дымка. Диаметр отложения при гололеде составляет 1 мм.',
'Дымка. Диаметр отложения при гололеде составляет 13 мм.',
'Дымка. Диаметр отложения при гололеде составляет 14 мм.',
'Дымка. Диаметр отложения при гололеде составляет 2 мм.',
'Дымка. Диаметр отложения при гололеде составляет 3 мм.',
'Дымка. Диаметр отложения при гололеде составляет 4 мм.',
'Дымка. Диаметр отложения при гололеде составляет 5 мм.',
'Дымка. Диаметр отложения при гололеде составляет 9 мм.',
'Дымка. Диаметр смешанного отложения составляет 11 мм.',
'Дымка. Диаметр смешанного отложения составляет 3 мм.',
'Замерзающая морось или замерзающий дождь неливневые. ',
'Замерзающая морось или замерзающий дождь неливневые. Диаметр отложения при гололеде составляет 1 мм.',
'Замерзающая морось или замерзающий дождь неливневые. Диаметр отложения при гололеде составляет 2 мм.',
'Замерзающая морось или замерзающий дождь неливневые. Диаметр отложения при гололеде составляет 3 мм.',
'Замерзающая морось или замерзающий дождь неливневые. Диаметр отложения при гололеде составляет 4 мм.',
'Замерзающая морось или замерзающий дождь неливневые. Диаметр отложения при гололеде составляет 8 мм.',
'Замерзающая морось или замерзающий дождь. ',
'Клочья приземного или ледяного тумана на станции, на море или на суше, высотой не более 2 м над сушей или не более 10 м над морем. ',
'Клочья приземного или ледяного тумана на станции, на море или на суше, высотой не более 2 м над сушей или не более 10 м над морем. Диаметр отложения мокрого снега составляет 7 мм.',
'Ледяная крупа. ',
'Ледяная крупа. Высота свежевыпавшего снега составляет 1 см. Диаметр отложения при гололеде составляет 1 мм.',
'Ледяная крупа. Диаметр отложения при гололеде составляет 1 мм.',
'Ледяная крупа. Диаметр отложения при гололеде составляет 2 мм.',
'Ледяная крупа. Диаметр отложения при гололеде составляет 3 мм.',
'Ливневая снежная крупа или небольшой град с дождем или без него, или дождь со снегом слабые в срок наблюдения или за последний час. ',
'Ливневая снежная крупа или небольшой град с дождем или без него, или дождь со снегом слабые в срок наблюдения или за последний час. Диаметр отложения при гололеде составляет 1 мм.',

'Ливневый(ые) дождь(и) умеренный(ые) или сильный(ые) в срок наблюдения или за последний час. Максимальный диаметр градин составляет 5 мм.',

'Ливневый(ые) дождь(и). ',

'Ливневый(ые) дождь(и). Диаметр отложения при гололеде составляет 1 мм.',

'Ливневый(ые) дождь(и). Диаметр отложения при гололеде составляет 2 мм.',

'Ливневый(ые) дождь(и). Диаметр отложения при гололеде составляет 4 мм.',

'Ливневый(ые) дождь(и). Максимальный диаметр градин составляет 2 мм.',

'Мгла или дым, или пыль, взвешенная в воздухе, дальность видимости 1 км или более. ',

'Мгла или дым, или пыль, взвешенная в воздухе, дальность видимости менее 1 км. ',

'Мгла. ', 'Мгла. Диаметр изморозевого отложения составляет 4 мм.',

'Морось (незамерзающая) или снежные зерна неливневые. ',

'Морось (незамерзающая) или снежные зерна неливневые. Диаметр смешанного отложения составляет 2 мм.',

'Морось замерзающая слабая. ',

'Морось замерзающая слабая. Диаметр отложения при гололеде составляет 1 мм.',

'Морось замерзающая слабая. Диаметр отложения при гололеде составляет 2 мм.',

'Морось замерзающая слабая. Диаметр отложения при гололеде составляет 3 мм.',

'Морось замерзающая слабая. Диаметр отложения при гололеде составляет 4 мм.',

'Морось замерзающая слабая. Диаметр отложения при гололеде составляет 7 мм.',

'Морось замерзающая слабая. Диаметр смешанного отложения составляет 3 мм.',

'Морось замерзающая умеренная или сильная. ',

'Морось замерзающая умеренная или сильная. Диаметр отложения при гололеде составляет 2 мм.',

'Морось замерзающая умеренная или сильная. Диаметр смешанного отложения составляет 3 мм.',

'Морось и дождь слабые. ',

'Морось и дождь умеренные или сильные. ',

'Морось незамерзающая непрерывная сильная в срок наблюдения. ',

'Морось незамерзающая непрерывная слабая в срок наблюдения. ',

'Морось незамерзающая непрерывная слабая в срок наблюдения. Диаметр отложения при гололеде составляет 1 мм.',

'Морось незамерзающая непрерывная слабая в срок наблюдения. Диаметр отложения при гололеде составляет 4 мм.',

'Морось незамерзающая непрерывная умеренная в срок наблюдения. ',

'Морось незамерзающая с перерывами сильная в срок наблюдения. ',

'Морось незамерзающая с перерывами слабая в срок наблюдения. ',

'Морось незамерзающая с перерывами умеренная в срок наблюдения. ',

'Морось незамерзающая с перерывами умеренная в срок наблюдения. Диаметр отложения при гололеде составляет 1 мм.',

'Ниаких особых явлений погоды не наблюдалось. ',

'Облака в целом образовывались или развивались. ',

'Облака в целом образовывались или развивались. Высота свежевыпавшего снега составляет 0.4 см.',

'Облака в целом образовывались или развивались. Высота свежевыпавшего снега составляет 1 см.',

'Облака в целом образовывались или развивались. Диаметр изморозевого отложения составляет 1 мм.',

'Облака в целом образовывались или развивались. Диаметр изморозевого отложения составляет 2 мм.',

'Облака в целом образовывались или развивались. Диаметр изморозевого отложения составляет 8 мм.',

'Облака в целом образовывались или развивались. Диаметр отложения мокрого снега составляет 10 мм.',

'Облака в целом образовывались или развивались. Диаметр отложения мокрого снега составляет 2 мм.',

'Облака в целом образовывались или развивались. Диаметр отложения мокрого снега составляет 4 мм.',

'Облака в целом образовывались или развивались. Диаметр отложения мокрого снега составляет 5 мм.',

'Облака в целом образовывались или развивались. Диаметр отложения мокрого снега составляет 9 мм.',
'Облака в целом образовывались или развивались. Диаметр отложения при гололеде составляет 1 мм.',
'Облака в целом образовывались или развивались. Диаметр отложения при гололеде составляет 2 мм.',
'Облака в целом образовывались или развивались. Максимальный диаметр градин составляет 2 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. ',
'Облака в целом рассеиваются или становятся менее развитыми. Высота свежевыпавшего снега составляет 0.1 см.',
'Облака в целом рассеиваются или становятся менее развитыми. Высота свежевыпавшего снега составляет 0.2 см.',
'Облака в целом рассеиваются или становятся менее развитыми. Высота свежевыпавшего снега составляет 0.3 см.',
'Облака в целом рассеиваются или становятся менее развитыми. Высота свежевыпавшего снега составляет 1 см.',
'Облака в целом рассеиваются или становятся менее развитыми. Высота свежевыпавшего снега составляет 2 см.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр изморозевого отложения составляет 1 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр изморозевого отложения составляет 12 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр изморозевого отложения составляет 2 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр изморозевого отложения составляет 3 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр изморозевого отложения составляет 4 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр изморозевого отложения составляет 5 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр изморозевого отложения составляет 6 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр изморозевого отложения составляет 8 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения мокрого снега составляет 1 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения мокрого снега составляет 11 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения мокрого снега составляет 16 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения мокрого снега составляет 2 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения мокрого снега составляет 4 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения мокрого снега составляет 5 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения мокрого снега составляет 7 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения при гололеде составляет 1 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения при гололеде составляет 2 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр отложения при гололеде составляет 4 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Диаметр смешанного отложения составляет 4 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Максимальный диаметр градин составляет 10 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Максимальный диаметр градин составляет 2 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Максимальный диаметр градин составляет 3 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Максимальный диаметр градин составляет 5 мм.',
'Облака в целом рассеиваются или становятся менее развитыми. Максимальный диаметр градин составляет 6 мм.',
'Осадки ',
'Осадки в поле зрения, достигающие поверхности земли или моря на расстоянии более 5 км от станции. ',
'Осадки в поле зрения, достигающие поверхности земли или моря поблизости, но не на самой станции. ',
'Отдельные кристаллы снега в виде звездочек (с туманом или без него). ',
'Отдельные кристаллы снега в виде звездочек (с туманом или без него). Диаметр изморозевого отложения составляет 1 мм.',
'Пыль или песок, поднятые ветром на или вблизи станции в срок наблюдения, но нет хорошо развитого пыльного или песчаного вихря (вихрей), и никакой пыльной или песчаной бури не видно; или, в случае сообщения с судна или с береговой станции, брызги, переносимые ветром на станции. ',
'Пыль, взвешенная в воздухе на обширном пространстве, но не поднятая ветром на или вблизи станции в срок наблюдения. ',
'Развитие облачности не наблюдалось или не наблюдается. ',

'Развитие облачности не наблюдалось или не наблюдается. Диаметр изморозевого отложения составляет 2 мм.',
'Сильная пыльная или песчаная буря ослабела за последний час. ',
'Сильная снежная низовая метель в целом высокая (выше уровня глаз наблюдателя). ',
'Сильный поземок в целом низкий (ниже уровня глаз наблюдателя). ',
'Слабая или умеренная пыльная или песчаная буря началась или усилилась в течение последнего часа. ',
'Слабая или умеренная пыльная или песчаная буря ослабела за последний час. ',
'Слабая или умеренная снежная низовая метель в целом высокая (выше уровня глаз наблюдателя). ',
'Слабая или умеренная снежная низовая метель в целом высокая (выше уровня глаз наблюдателя). Диаметр отложения мокрого снега составляет 4 мм.',
'Слабая или умеренная снежная низовая метель в целом высокая (выше уровня глаз наблюдателя). Диаметр отложения при гололеде составляет 1 мм.',
'Слабый дождь в срок наблюдения. Гроза в течение последнего часа, но не в срок наблюдения. ',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). ',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). Высота свежевыпавшего снега составляет 0.2 см.',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). Высота свежевыпавшего снега составляет 0.3 см.',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). Высота свежевыпавшего снега составляет 0.4 см.',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). Высота свежевыпавшего снега составляет 1 см.',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). Высота свежевыпавшего снега составляет 4 см.',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). Диаметр изморозевого отложения составляет 5 мм.',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). Диаметр отложения мокрого снега составляет 4 мм.',
'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя). Диаметр отложения при гололеде составляет 1 см.',
'Слабый снег или дождь со снегом или град в срок наблюдения. Гроза в течение последнего часа, но не в срок наблюдения. ',
'Снег неливневый. ',
'Снег неливневый. Высота свежевыпавшего снега составляет 0.2 см.',
'Снег неливневый. Высота свежевыпавшего снега составляет 0.4 см.',
'Снег неливневый. Высота свежевыпавшего снега составляет 0.5 см.',
'Снег неливневый. Высота свежевыпавшего снега составляет 1 см. Диаметр отложения при гололеде составляет 1 мм.',
'Снег неливневый. Высота свежевыпавшего снега составляет 2 см.',
'Снег неливневый. Высота свежевыпавшего снега составляет 5 см.',
'Снег неливневый. Диаметр изморозевого отложения составляет 2 мм.',
'Снег неливневый. Диаметр изморозевого отложения составляет 3 мм.',
'Снег неливневый. Диаметр изморозевого отложения составляет 4 мм.',
'Снег неливневый. Диаметр изморозевого отложения составляет 5 мм.',
'Снег неливневый. Диаметр изморозевого отложения составляет 6 мм.',
'Снег неливневый. Диаметр изморозевого отложения составляет 7 мм.',
'Снег неливневый. Диаметр отложения мокрого снега составляет 1 мм.',

'Туман или ледяной туман на расстоянии в срок наблюдения, простирающийся выше уровня глаз наблюдателя. В течение последнего часа туман на станции не наблюдался.' ,

'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа.'

'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа. Высота свежевыпавшего снега составляет 0.2 см.';

'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа. Диаметр изморозевого отложения составляет 1 мм.',

'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа. Диаметр изморозевого отложения составляет 2 мм.'.

'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа. Диаметр изморозевого отложения составляет 3 мм.'.

'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа. Диаметр изморозевого отложения составляет 4 мм.'

'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа. Диаметр отложения при гололеде составляет 1 мм.'

'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа. Диаметр отложения при гололеде составляет 1 мкм.'

'Умеренный или сильный дождь в срок наблюдения. Гроза в течение последнего часа, но не в срок наблюдения. ',
'Умеренный или сильный снег или дождь со снегом, или град в срок наблюдения. Гроза в течение последнего часа, но не в срок наблюдения. ',
'Хорошо развитый пыльный или песчаный вихрь (вихри) на или вблизи станции в течение последнего часа или в срок наблюдения, но пыльной или песчаной бури нет. ',
'Шквалы на станции или в поле зрения за последний час или в срок наблюдения. '],
dtype=object)

Уникальные значения всех датафреймов для столбца Wthr_3h

```
Out[38]: array(['Буря', 'Гроза (грозы) с осадками или без них.',  
               'Дождь со снегом или другими видами твердых осадков', 'Дождь.',  
               'Ливень (ливни).', 'Ливни или перемещающиеся осадки.', 'Метель',  
               'Морось.', 'Никаких особых явлений погоды не наблюдалось.',  
               'Облака покрывали более половины неба в течение всего соответствующего периода.',  
               'Облака покрывали более половины неба в течение одной части соответствующего периода и половину или менее в течение другой части периода.',  
               'Облака покрывали половину неба или менее в течение всего соответствующего периода.',  
               'Пониженная видимость.', 'Снег и/или другие виды твердых осадков',  
               'Снежная или ледяная крупа.',  
               'Туман или ледяной туман или сильная мгла.', nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца Wthr_3h2

```
Out[38]: array(['Буря', 'Гроза (грозы) с осадками или без них.',  
               'Дождь со снегом или другими видами твердых осадков', 'Дождь.',  
               'Ливень (ливни).', 'Метель', 'Морось.',  
               'Никаких особых явлений погоды не наблюдалось.',  
               'Облака покрывали более половины неба в течение всего соответствующего периода.',  
               'Облака покрывали более половины неба в течение одной части соответствующего периода и половину или менее в течение другой части периода.',  
               'Облака покрывали половину неба или менее в течение всего соответствующего периода.',  
               'Пониженная видимость.', 'Снег и/или другие виды твердых осадков',  
               'Туман', 'Туман или ледяной туман или сильная мгла.',  
               'Явление, связанное с переносом ветром твердых частиц, видимость пониженная.',  
               nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца T_min

```
Out[38]: array([-38.4, -37.6, -37.5, -37.2, -37. , -36.8, -36.7, -36.2, -36. ,  
-35.9, -35.8, -35.7, -35.6, -35.5, -35.4, -35.2, -35.1, -35. ,  
-34.9, -34.8, -34.6, -34.5, -34.4, -34.2, -34.1, -34. , -33.9,  
-33.8, -33.7, -33.6, -33.5, -33.4, -33.3, -33.2, -33.1, -33. ,  
-32.9, -32.8, -32.7, -32.6, -32.5, -32.4, -32.3, -32.2, -32.1,  
-32. , -31.9, -31.8, -31.7, -31.6, -31.5, -31.4, -31.3, -31.2,  
-31.1, -31. , -30.9, -30.8, -30.7, -30.6, -30.5, -30.4, -30.3,  
-30.2, -30.1, -30. , -29.9, -29.8, -29.7, -29.6, -29.5, -29.4,  
-29.3, -29.2, -29.1, -29. , -28.9, -28.8, -28.7, -28.6, -28.5,  
-28.4, -28.3, -28.2, -28.1, -28. , -27.9, -27.8, -27.7, -27.6,  
-27.5, -27.4, -27.3, -27.2, -27.1, -27. , -26.9, -26.8, -26.7,  
-26.6, -26.5, -26.4, -26.3, -26.2, -26.1, -26. , -25.9, -25.8,  
-25.7, -25.6, -25.5, -25.4, -25.3, -25.2, -25.1, -25. , -24.9,  
-24.8, -24.7, -24.6, -24.5, -24.4, -24.3, -24.2, -24.1, -24. ,  
-23.9, -23.8, -23.7, -23.6, -23.5, -23.4, -23.3, -23.2, -23.1,  
-23. , -22.9, -22.8, -22.7, -22.6, -22.5, -22.4, -22.3, -22.2,  
-22.1, -22. , -21.9, -21.8, -21.7, -21.6, -21.5, -21.4, -21.3,  
-21.2, -21.1, -21. , -20.9, -20.8, -20.7, -20.6, -20.5, -20.4,  
-20.3, -20.2, -20.1, -20. , -19.9, -19.8, -19.7, -19.6, -19.5,  
-19.4, -19.3, -19.2, -19.1, -19. , -18.9, -18.8, -18.7, -18.6,  
-18.5, -18.4, -18.3, -18.2, -18.1, -18. , -17.9, -17.8, -17.7,  
-17.6, -17.5, -17.4, -17.3, -17.2, -17.1, -17. , -16.9, -16.8,  
-16.7, -16.6, -16.5, -16.4, -16.3, -16.2, -16.1, -16. , -15.9,  
-15.8, -15.7, -15.6, -15.5, -15.4, -15.3, -15.2, -15.1, -15. ,  
-14.9, -14.8, -14.7, -14.6, -14.5, -14.4, -14.3, -14.2, -14.1,  
-14. , -13.9, -13.8, -13.7, -13.6, -13.5, -13.4, -13.3, -13.2,  
-13.1, -13. , -12.9, -12.8, -12.7, -12.6, -12.5, -12.4, -12.3,  
-12.2, -12.1, -12. , -11.9, -11.8, -11.7, -11.6, -11.5, -11.4,  
-11.3, -11.2, -11.1, -11. , -10.9, -10.8, -10.7, -10.6, -10.5,  
-10.4, -10.3, -10.2, -10.1, -10. , -9.9, -9.8, -9.7, -9.6,  
-9.5, -9.4, -9.3, -9.2, -9.1, -9. , -8.9, -8.8, -8.7,  
-8.6, -8.5, -8.4, -8.3, -8.2, -8.1, -8. , -7.9, -7.8,  
-7.7, -7.6, -7.5, -7.4, -7.3, -7.2, -7.1, -7. , -6.9,  
-6.8, -6.7, -6.6, -6.5, -6.4, -6.3, -6.2, -6.1, -6. ,  
-5.9, -5.8, -5.7, -5.6, -5.5, -5.4, -5.3, -5.2, -5.1,  
-5. , -4.9, -4.8, -4.7, -4.6, -4.5, -4.4, -4.3, -4.2,  
-4.1, -4. , -3.9, -3.8, -3.7, -3.6, -3.5, -3.4, -3.3,  
-3.2, -3.1, -3. , -2.9, -2.8, -2.7, -2.6, -2.5, -2.4,  
-2.3, -2.2, -2.1, -2. , -1.9, -1.8, -1.7, -1.6, -1.5,  
-1.4, -1.3, -1.2, -1.1, -1. , -0.9, -0.8, -0.7, -0.6,  
-0.5, -0.4, -0.3, -0.2, -0.1, 0. , 0.1, 0.2, 0.3,  
0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,  
1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1,  
2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3. ,
```

3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9,
4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8,
4.9, 5. , 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7,
5.8, 5.9, 6. , 6.1, 6.2, 6.3, 6.4, 6.5, 6.6,
6.7, 6.8, 6.9, 7. , 7.1, 7.2, 7.3, 7.4, 7.5,
7.6, 7.7, 7.8, 7.9, 8. , 8.1, 8.2, 8.3, 8.4,
8.5, 8.6, 8.7, 8.8, 8.9, 9. , 9.1, 9.2, 9.3,
9.4, 9.5, 9.6, 9.7, 9.8, 9.9, 10. , 10.1, 10.2,
10.3, 10.4, 10.5, 10.6, 10.7, 10.8, 10.9, 11. , 11.1,
11.2, 11.3, 11.4, 11.5, 11.6, 11.7, 11.8, 11.9, 12. ,
12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, 12.9,
13. , 13.1, 13.2, 13.3, 13.4, 13.5, 13.6, 13.7, 13.8,
13.9, 14. , 14.1, 14.2, 14.3, 14.4, 14.5, 14.6, 14.7,
14.8, 14.9, 15. , 15.1, 15.2, 15.3, 15.4, 15.5, 15.6,
15.7, 15.8, 15.9, 16. , 16.1, 16.2, 16.3, 16.4, 16.5,
16.6, 16.7, 16.8, 16.9, 17. , 17.1, 17.2, 17.3, 17.4,
17.5, 17.6, 17.7, 17.8, 17.9, 18. , 18.1, 18.2, 18.3,
18.4, 18.5, 18.6, 18.7, 18.8, 18.9, 19. , 19.1, 19.2,
19.3, 19.4, 19.5, 19.6, 19.7, 19.8, 19.9, 20. , 20.1,
20.2, 20.3, 20.4, 20.5, 20.6, 20.7, 20.8, 20.9, 21. ,
21.1, 21.2, 21.3, 21.4, 21.5, 21.6, 21.7, 21.8, 21.9,
22. , 22.2, 22.3, 22.4, 22.5, 23.1, 23.2, 23.3, 23.4,
23.5, 23.6, 23.8, 23.9, 24. , 24.7, 24.8, 25. , 25.3,
25.4, 25.6, 26.2, 26.4, 26.7, 28.2, 28.6, 29. , 31.4,
31.6, nan])

Уникальные значения всех датафреймов для столбца T_max

```
Out[38]: array([-9.999e+02, -2.960e+01, -2.930e+01, -2.890e+01, -2.880e+01,
 -2.850e+01, -2.840e+01, -2.830e+01, -2.820e+01, -2.800e+01,
 -2.780e+01, -2.760e+01, -2.750e+01, -2.740e+01, -2.710e+01,
 -2.700e+01, -2.690e+01, -2.680e+01, -2.670e+01, -2.630e+01,
 -2.620e+01, -2.600e+01, -2.570e+01, -2.560e+01, -2.540e+01,
 -2.520e+01, -2.510e+01, -2.500e+01, -2.490e+01, -2.470e+01,
 -2.460e+01, -2.450e+01, -2.440e+01, -2.430e+01, -2.420e+01,
 -2.410e+01, -2.400e+01, -2.390e+01, -2.380e+01, -2.360e+01,
 -2.350e+01, -2.340e+01, -2.330e+01, -2.320e+01, -2.310e+01,
 -2.300e+01, -2.290e+01, -2.280e+01, -2.270e+01, -2.260e+01,
 -2.250e+01, -2.240e+01, -2.230e+01, -2.220e+01, -2.210e+01,
 -2.200e+01, -2.190e+01, -2.180e+01, -2.170e+01, -2.160e+01,
 -2.150e+01, -2.140e+01, -2.130e+01, -2.120e+01, -2.110e+01,
 -2.100e+01, -2.090e+01, -2.080e+01, -2.070e+01, -2.060e+01,
 -2.050e+01, -2.040e+01, -2.030e+01, -2.020e+01, -2.010e+01,
 -2.000e+01, -1.990e+01, -1.980e+01, -1.970e+01, -1.960e+01,
 -1.950e+01, -1.940e+01, -1.930e+01, -1.920e+01, -1.910e+01,
 -1.900e+01, -1.890e+01, -1.880e+01, -1.870e+01, -1.860e+01,
 -1.850e+01, -1.840e+01, -1.830e+01, -1.820e+01, -1.810e+01,
 -1.800e+01, -1.790e+01, -1.780e+01, -1.770e+01, -1.760e+01,
 -1.750e+01, -1.740e+01, -1.730e+01, -1.720e+01, -1.710e+01,
 -1.700e+01, -1.690e+01, -1.680e+01, -1.670e+01, -1.660e+01,
 -1.650e+01, -1.640e+01, -1.630e+01, -1.620e+01, -1.610e+01,
 -1.600e+01, -1.590e+01, -1.580e+01, -1.570e+01, -1.560e+01,
 -1.550e+01, -1.540e+01, -1.530e+01, -1.520e+01, -1.510e+01,
 -1.500e+01, -1.490e+01, -1.480e+01, -1.470e+01, -1.460e+01,
 -1.450e+01, -1.440e+01, -1.430e+01, -1.420e+01, -1.410e+01,
 -1.400e+01, -1.390e+01, -1.380e+01, -1.370e+01, -1.360e+01,
 -1.350e+01, -1.340e+01, -1.330e+01, -1.320e+01, -1.310e+01,
 -1.300e+01, -1.290e+01, -1.280e+01, -1.270e+01, -1.260e+01,
 -1.250e+01, -1.240e+01, -1.230e+01, -1.220e+01, -1.210e+01,
 -1.200e+01, -1.190e+01, -1.180e+01, -1.170e+01, -1.160e+01,
 -1.150e+01, -1.140e+01, -1.130e+01, -1.120e+01, -1.110e+01,
 -1.100e+01, -1.090e+01, -1.080e+01, -1.070e+01, -1.060e+01,
 -1.050e+01, -1.040e+01, -1.030e+01, -1.020e+01, -1.010e+01,
 -1.000e+01, -9.900e+00, -9.800e+00, -9.700e+00, -9.600e+00,
 -9.500e+00, -9.400e+00, -9.300e+00, -9.200e+00, -9.100e+00,
 -9.000e+00, -8.900e+00, -8.800e+00, -8.700e+00, -8.600e+00,
 -8.500e+00, -8.400e+00, -8.300e+00, -8.200e+00, -8.100e+00,
 -8.000e+00, -7.900e+00, -7.800e+00, -7.700e+00, -7.600e+00,
 -7.500e+00, -7.400e+00, -7.300e+00, -7.200e+00, -7.100e+00,
 -7.000e+00, -6.900e+00, -6.800e+00, -6.700e+00, -6.600e+00,
 -6.500e+00, -6.400e+00, -6.300e+00, -6.200e+00, -6.100e+00,
 -6.000e+00, -5.900e+00, -5.800e+00, -5.700e+00, -5.600e+00,
```

-5.500e+00, -5.400e+00, -5.300e+00, -5.200e+00, -5.100e+00,
-5.000e+00, -4.900e+00, -4.800e+00, -4.700e+00, -4.600e+00,
-4.500e+00, -4.400e+00, -4.300e+00, -4.200e+00, -4.100e+00,
-4.000e+00, -3.900e+00, -3.800e+00, -3.700e+00, -3.600e+00,
-3.500e+00, -3.400e+00, -3.300e+00, -3.200e+00, -3.100e+00,
-3.000e+00, -2.900e+00, -2.800e+00, -2.700e+00, -2.600e+00,
-2.500e+00, -2.400e+00, -2.300e+00, -2.200e+00, -2.100e+00,
-2.000e+00, -1.900e+00, -1.800e+00, -1.700e+00, -1.600e+00,
-1.500e+00, -1.400e+00, -1.300e+00, -1.200e+00, -1.100e+00,
-1.000e+00, -9.000e-01, -8.000e-01, -7.000e-01, -6.000e-01,
-5.000e-01, -4.000e-01, -3.000e-01, -2.000e-01, -1.000e-01,
0.000e+00, 1.000e-01, 2.000e-01, 3.000e-01, 4.000e-01,
5.000e-01, 6.000e-01, 7.000e-01, 8.000e-01, 9.000e-01,
1.000e+00, 1.100e+00, 1.200e+00, 1.300e+00, 1.400e+00,
1.500e+00, 1.600e+00, 1.700e+00, 1.800e+00, 1.900e+00,
2.000e+00, 2.100e+00, 2.200e+00, 2.300e+00, 2.400e+00,
2.500e+00, 2.600e+00, 2.700e+00, 2.800e+00, 2.900e+00,
3.000e+00, 3.100e+00, 3.200e+00, 3.300e+00, 3.400e+00,
3.500e+00, 3.600e+00, 3.700e+00, 3.800e+00, 3.900e+00,
4.000e+00, 4.100e+00, 4.200e+00, 4.300e+00, 4.400e+00,
4.500e+00, 4.600e+00, 4.700e+00, 4.800e+00, 4.900e+00,
5.000e+00, 5.100e+00, 5.200e+00, 5.300e+00, 5.400e+00,
5.500e+00, 5.600e+00, 5.700e+00, 5.800e+00, 5.900e+00,
6.000e+00, 6.100e+00, 6.200e+00, 6.300e+00, 6.400e+00,
6.500e+00, 6.600e+00, 6.700e+00, 6.800e+00, 6.900e+00,
7.000e+00, 7.100e+00, 7.200e+00, 7.300e+00, 7.400e+00,
7.500e+00, 7.600e+00, 7.700e+00, 7.800e+00, 7.900e+00,
8.000e+00, 8.100e+00, 8.200e+00, 8.300e+00, 8.400e+00,
8.500e+00, 8.600e+00, 8.700e+00, 8.800e+00, 8.900e+00,
9.000e+00, 9.100e+00, 9.200e+00, 9.300e+00, 9.400e+00,
9.500e+00, 9.600e+00, 9.700e+00, 9.800e+00, 9.900e+00,
1.000e+01, 1.010e+01, 1.020e+01, 1.030e+01, 1.040e+01,
1.050e+01, 1.060e+01, 1.070e+01, 1.080e+01, 1.090e+01,
1.100e+01, 1.110e+01, 1.120e+01, 1.130e+01, 1.140e+01,
1.150e+01, 1.160e+01, 1.170e+01, 1.180e+01, 1.190e+01,
1.200e+01, 1.210e+01, 1.220e+01, 1.230e+01, 1.240e+01,
1.250e+01, 1.260e+01, 1.270e+01, 1.280e+01, 1.290e+01,
1.300e+01, 1.310e+01, 1.320e+01, 1.330e+01, 1.340e+01,
1.350e+01, 1.360e+01, 1.370e+01, 1.380e+01, 1.390e+01,
1.400e+01, 1.410e+01, 1.420e+01, 1.430e+01, 1.440e+01,
1.450e+01, 1.460e+01, 1.470e+01, 1.480e+01, 1.490e+01,
1.500e+01, 1.510e+01, 1.520e+01, 1.530e+01, 1.540e+01,
1.550e+01, 1.560e+01, 1.570e+01, 1.580e+01, 1.590e+01,
1.600e+01, 1.610e+01, 1.620e+01, 1.630e+01, 1.640e+01,

1.650e+01,	1.660e+01,	1.670e+01,	1.680e+01,	1.690e+01,
1.700e+01,	1.710e+01,	1.720e+01,	1.730e+01,	1.740e+01,
1.750e+01,	1.760e+01,	1.770e+01,	1.780e+01,	1.790e+01,
1.800e+01,	1.810e+01,	1.820e+01,	1.830e+01,	1.840e+01,
1.850e+01,	1.860e+01,	1.870e+01,	1.880e+01,	1.890e+01,
1.900e+01,	1.910e+01,	1.920e+01,	1.930e+01,	1.940e+01,
1.950e+01,	1.960e+01,	1.970e+01,	1.980e+01,	1.990e+01,
2.000e+01,	2.010e+01,	2.020e+01,	2.030e+01,	2.040e+01,
2.050e+01,	2.060e+01,	2.070e+01,	2.080e+01,	2.090e+01,
2.100e+01,	2.110e+01,	2.120e+01,	2.130e+01,	2.140e+01,
2.150e+01,	2.160e+01,	2.170e+01,	2.180e+01,	2.190e+01,
2.200e+01,	2.210e+01,	2.220e+01,	2.230e+01,	2.240e+01,
2.250e+01,	2.260e+01,	2.270e+01,	2.280e+01,	2.290e+01,
2.300e+01,	2.310e+01,	2.320e+01,	2.330e+01,	2.340e+01,
2.350e+01,	2.360e+01,	2.370e+01,	2.380e+01,	2.390e+01,
2.400e+01,	2.410e+01,	2.420e+01,	2.430e+01,	2.440e+01,
2.450e+01,	2.460e+01,	2.470e+01,	2.480e+01,	2.490e+01,
2.500e+01,	2.510e+01,	2.520e+01,	2.530e+01,	2.540e+01,
2.550e+01,	2.560e+01,	2.570e+01,	2.580e+01,	2.590e+01,
2.600e+01,	2.610e+01,	2.620e+01,	2.630e+01,	2.640e+01,
2.650e+01,	2.660e+01,	2.670e+01,	2.680e+01,	2.690e+01,
2.700e+01,	2.710e+01,	2.720e+01,	2.730e+01,	2.740e+01,
2.750e+01,	2.760e+01,	2.770e+01,	2.780e+01,	2.790e+01,
2.800e+01,	2.810e+01,	2.820e+01,	2.830e+01,	2.840e+01,
2.850e+01,	2.860e+01,	2.870e+01,	2.880e+01,	2.890e+01,
2.900e+01,	2.910e+01,	2.920e+01,	2.930e+01,	2.940e+01,
2.950e+01,	2.960e+01,	2.970e+01,	2.980e+01,	2.990e+01,
3.000e+01,	3.010e+01,	3.020e+01,	3.030e+01,	3.040e+01,
3.050e+01,	3.060e+01,	3.070e+01,	3.080e+01,	3.090e+01,
3.100e+01,	3.110e+01,	3.120e+01,	3.130e+01,	3.140e+01,
3.150e+01,	3.160e+01,	3.170e+01,	3.180e+01,	3.190e+01,
3.200e+01,	3.210e+01,	3.220e+01,	3.230e+01,	3.240e+01,
3.250e+01,	3.260e+01,	3.270e+01,	3.280e+01,	3.290e+01,
3.300e+01,	3.310e+01,	3.320e+01,	3.330e+01,	3.340e+01,
3.350e+01,	3.360e+01,	3.370e+01,	3.380e+01,	3.390e+01,
3.400e+01,	3.410e+01,	3.420e+01,	3.430e+01,	3.440e+01,
3.450e+01,	3.460e+01,	3.470e+01,	3.480e+01,	3.490e+01,
3.500e+01,	3.510e+01,	3.520e+01,	3.530e+01,	3.540e+01,
3.550e+01,	3.560e+01,	3.570e+01,	3.580e+01,	3.590e+01,
3.600e+01,	3.610e+01,	3.620e+01,	3.630e+01,	3.640e+01,
3.650e+01,	3.660e+01,	3.670e+01,	3.680e+01,	3.690e+01,
3.700e+01,	3.710e+01,	3.720e+01,	3.730e+01,	3.740e+01,
3.750e+01,	3.760e+01,	3.770e+01,	3.780e+01,	3.790e+01,
3.800e+01,	3.810e+01,	3.830e+01,	3.840e+01,	3.850e+01,

```
3.860e+01, 3.870e+01, 3.880e+01, 3.900e+01, 3.940e+01,  
nan])
```

Уникальные значения всех датафреймов для столбца Cl_Cumls

```
Out[38]: array(['Кучево-дождевые волокнистые (часто с наковальней), либо с кучево-дождевыми лысыми, кучевыми, слоистыми, разорванно-дожде-  
выми, либо без них.',  
    'Кучево-дождевые лысые с кучевыми, слоисто-кучевыми или слоистыми, либо без них.',  
    'Кучевые и слоисто-кучевые (но не слоисто-кучевые, образовавшиеся из кучевых), основания расположены на разных уровнях.',  
    'Кучевые плоские или кучевые разорванные, или те и другие вместе, не относящиеся к облакам плохой погоды.',  
    'Кучевые средние или мощные или вместе с кучевыми разорванными, или с кучевыми плоскими, или со слоисто-кучевыми, либо бе-  
з них; основания всех этих облаков расположены на одном уровне.',  
    'Слоисто-кучевые, образовавшиеся из кучевых.',  
    'Слоисто-кучевые, образовавшиеся не из кучевых.',  
    'Слоисто-кучевых, слоистых, кучевых или кучево-дождевых облаков нет.',  
    'Слоистые разорванные или кучевые разорванные облака плохой погоды, либо те и другие вместе (разорванно-дождевые); обычно  
расположены под слоистыми или слоисто-дождевыми облаками.',  
    'Слоистые туманообразные или слоистые разорванные, либо те и другие, но не относящиеся к облакам плохой погоды.',  
    nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца Cl_viewd

```
Out[38]: array(['10% или менее, но не 0', '100%.', '20-30%.', '40%.', '50%.',  
    '60%.', '70 - 80%.', '90 или более, но не 100%',  
    'Небо не видно из-за тумана и/или других метеорологических явлений.',  
    'Облаков нет.', nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца Cl_bottom

```
Out[38]: array(['100-200', '1000-1500', '1500-2000', '200-300', '2000-2500',  
    '2500 или более, или облаков нет.', '300-600', '50-100',  
    '600-1000', 'Менее 50', nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца Cl_cumls_hi

```
Out[38]: array(['Высококучевые башенообразные или хлопьевидные.',  
    'Высококучевые при хаотическом виде неба, обычно расположенные на нескольких уровнях.',  
    'Высококучевые просвечивающие, полосами, либо один или несколько слоев высококучевых просвечивающих, распространяющихся п-  
о небу; эти высококучевые в целом уплотняются.',  
    'Высококучевые просвечивающие, расположенные на одном уровне.',  
    'Высококучевые, образовавшиеся из кучевых (или кучево-дождевых).',  
    'Высококучевые, просвечивающие или плотные в двух или более слоях или высококучевые плотные в одном слое, не распространя-  
ющиеся по небу, либо высококучевые с высокослоистыми или слоисто-дождевыми.',  
    'Высококучевых, высокослоистых или слоисто-дождевых облаков нет.',  
    'Высокослоистые непросвечивающие или слоисто-дождевые.',  
    'Высокослоистые просвечивающие.',  
    'Ключья (часто чечевицеобразные) высококучевых просвечивающих, непрерывно изменяющихся; расположены на одном или более у-  
ровнях.',  
    nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца Cl_cirrus

```
Out[38]: array(['Перисто-кучевые одни или перисто-кучевые, сопровождаемые перистыми или перисто-слоистыми, либо те и другие, но перисто-кучевые преобладают среди них.',  
   'Перисто-слоистые, не распространяющиеся по небу и не покрывающие его полностью.',  
   'Перисто-слоистые, покрывающие все небо.',  
   'Перистые (часто в виде полос) и перисто-слоистые, распространяющиеся по небу и в целом обычно уплотняющиеся, но сплошная пелена поднимается над горизонтом менее чем на 45°.',  
   'Перистые (часто в виде полос) и перисто-слоистые, распространяющиеся по небу и в целом обычно уплотняющиеся; сплошная пелена, поднимающаяся над горизонтом выше 45°, не покрывает всего неба.',  
   'Перистые когтевидные или нитевидные или первые и вторые, распространяющиеся по небу и в целом обычно уплотняющиеся.',  
   'Перистые нитевидные, иногда когтевидные, не распространяющиеся по небу.',  
   'Перистые плотные в виде кучевых или скрученных склонов, количество которых обычно не увеличивается, иногда могут казаться я остатками верхней части кучево-дождевых; или перистые башенообразные, или перистые хлопьевидные.',  
   'Перистые плотные, образовавшиеся от кучево-дождевых.',  
   'Перистых, перисто-кучевых или перисто-слоистых нет.', nan],  
  dtype=object)
```

Уникальные значения всех датафреймов для столбца Visibility

```
Out[38]: array([0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2,  
   1.3, 1.4, 1.5, 1.6, 1.8, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.8, 3.0,  
   3.1, 3.5, 3.6, 4.0, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.8, 5.0, 6.0,  
   7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0,  
   18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0,  
   29.0, 30.0, 35.0, 40.0, 50.0, 65.0, nan, '10.0', '20.0', '4.0',  
   '0.5', '0.2', '1.0', '2.0', '50.0', '0.7', '0.05', 'менее 0.1',  
   '0.4', '0.6', '0.3', '0.1', 45.0, 2.7, 4.7, 3.9, 3.8, 3.2, 3.7,  
   2.6, 3.4, 3.3, '2.7', '19.0', '2.9', 'менее 0.05', '15.0', '70.0',  
   '1.7', '65.0', '29.0', '60.0', '71.0', '40.0', '12.0', '17.0',  
   '11.0', '8.0', '3.0', '5.0', '13.0', '7.0', '9.0', '14.0', '18.0',  
   '16.0', '6.0', '2.5', '1.5', '1.3', '1.1', '3.5', '3.2', '0.8',  
   '1.8', '21.0', '4.5', '2.2', '1.2', '0.9', '1.9', '2.3', '3.8',  
   '35.0', '3.4', '23.0', '30.0', '1.6', '26.0', '27.0', '4.8',  
   '25.0', '28.0', '24.0', '22.0', '3.7', '3.3', '2.4', '4.2', '1.4',  
   '4.1', '3.9', '4.9', '3.1', '2.6', '3.6', '4.6', '2.1', '45.0',  
   '2.8', 1.9, 4.9, 2.9, 1.7, 60.0, 71.0, '4.7', '4.4', '55.0', '4.3'],  
  dtype=object)
```

Уникальные значения всех датафреймов для столбца Dew_point

```
Out[38]: array([-72.7, -40.5, -40.4, -39.6, -39.5, -39.4, -39.2, -39.1, -39. ,  
-38.8, -38.7, -38.6, -38.4, -38.3, -38.2, -38.1, -38. , -37.8,  
-37.7, -37.6, -37.4, -37.3, -37.2, -37.1, -37. , -36.9, -36.8,  
-36.7, -36.6, -36.5, -36.4, -36.3, -36.2, -36.1, -36. , -35.9,  
-35.8, -35.7, -35.6, -35.5, -35.4, -35.3, -35.2, -35.1, -35. ,  
-34.9, -34.8, -34.7, -34.6, -34.5, -34.4, -34.3, -34.2, -34.1,  
-34. , -33.9, -33.8, -33.7, -33.6, -33.5, -33.4, -33.3, -33.2,  
-33.1, -33. , -32.9, -32.8, -32.7, -32.6, -32.5, -32.4, -32.3,  
-32.2, -32.1, -32. , -31.9, -31.8, -31.7, -31.6, -31.5, -31.4,  
-31.3, -31.2, -31.1, -31. , -30.9, -30.8, -30.7, -30.6, -30.5,  
-30.4, -30.3, -30.2, -30.1, -30. , -29.9, -29.8, -29.7, -29.6,  
-29.5, -29.4, -29.3, -29.2, -29.1, -29. , -28.9, -28.8, -28.7,  
-28.6, -28.5, -28.4, -28.3, -28.2, -28.1, -28. , -27.9, -27.8,  
-27.7, -27.6, -27.5, -27.4, -27.3, -27.2, -27.1, -27. , -26.9,  
-26.8, -26.7, -26.6, -26.5, -26.4, -26.3, -26.2, -26.1, -26. ,  
-25.9, -25.8, -25.7, -25.6, -25.5, -25.4, -25.3, -25.2, -25.1,  
-25. , -24.9, -24.8, -24.7, -24.6, -24.5, -24.4, -24.3, -24.2,  
-24.1, -24. , -23.9, -23.8, -23.7, -23.6, -23.5, -23.4, -23.3,  
-23.2, -23.1, -23. , -22.9, -22.8, -22.7, -22.6, -22.5, -22.4,  
-22.3, -22.2, -22.1, -22. , -21.9, -21.8, -21.7, -21.6, -21.5,  
-21.4, -21.3, -21.2, -21.1, -21. , -20.9, -20.8, -20.7, -20.6,  
-20.5, -20.4, -20.3, -20.2, -20.1, -20. , -19.9, -19.8, -19.7,  
-19.6, -19.5, -19.4, -19.3, -19.2, -19.1, -19. , -18.9, -18.8,  
-18.7, -18.6, -18.5, -18.4, -18.3, -18.2, -18.1, -18. , -17.9,  
-17.8, -17.7, -17.6, -17.5, -17.4, -17.3, -17.2, -17.1, -17. ,  
-16.9, -16.8, -16.7, -16.6, -16.5, -16.4, -16.3, -16.2, -16.1,  
-16. , -15.9, -15.8, -15.7, -15.6, -15.5, -15.4, -15.3, -15.2,  
-15.1, -15. , -14.9, -14.8, -14.7, -14.6, -14.5, -14.4, -14.3,  
-14.2, -14.1, -14. , -13.9, -13.8, -13.7, -13.6, -13.5, -13.4,  
-13.3, -13.2, -13.1, -13. , -12.9, -12.8, -12.7, -12.6, -12.5,  
-12.4, -12.3, -12.2, -12.1, -12. , -11.9, -11.8, -11.7, -11.6,  
-11.5, -11.4, -11.3, -11.2, -11.1, -11. , -10.9, -10.8, -10.7,  
-10.6, -10.5, -10.4, -10.3, -10.2, -10.1, -10. , -9.9, -9.8,  
-9.7, -9.6, -9.5, -9.4, -9.3, -9.2, -9.1, -9. , -8.9,  
-8.8, -8.7, -8.6, -8.5, -8.4, -8.3, -8.2, -8.1, -8. ,  
-7.9, -7.8, -7.7, -7.6, -7.5, -7.4, -7.3, -7.2, -7.1,  
-7. , -6.9, -6.8, -6.7, -6.6, -6.5, -6.4, -6.3, -6.2,  
-6.1, -6. , -5.9, -5.8, -5.7, -5.6, -5.5, -5.4, -5.3,  
-5.2, -5.1, -5. , -4.9, -4.8, -4.7, -4.6, -4.5, -4.4,  
-4.3, -4.2, -4.1, -4. , -3.9, -3.8, -3.7, -3.6, -3.5,  
-3.4, -3.3, -3.2, -3.1, -3. , -2.9, -2.8, -2.7, -2.6,  
-2.5, -2.4, -2.3, -2.2, -2.1, -2. , -1.9, -1.8, -1.7,  
-1.6, -1.5, -1.4, -1.3, -1.2, -1.1, -1. , -0.9, -0.8,  
-0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0. , 0.1,
```

```
0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ,
1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9,
2. , 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8,
2.9, 3. , 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7,
3.8, 3.9, 4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6,
4.7, 4.8, 4.9, 5. , 5.1, 5.2, 5.3, 5.4, 5.5,
5.6, 5.7, 5.8, 5.9, 6. , 6.1, 6.2, 6.3, 6.4,
6.5, 6.6, 6.7, 6.8, 6.9, 7. , 7.1, 7.2, 7.3,
7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8. , 8.1, 8.2,
8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 9. , 9.1,
9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9, 10. ,
10.1, 10.2, 10.3, 10.4, 10.5, 10.6, 10.7, 10.8, 10.9,
11. , 11.1, 11.2, 11.3, 11.4, 11.5, 11.6, 11.7, 11.8,
11.9, 12. , 12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7,
12.8, 12.9, 13. , 13.1, 13.2, 13.3, 13.4, 13.5, 13.6,
13.7, 13.8, 13.9, 14. , 14.1, 14.2, 14.3, 14.4, 14.5,
14.6, 14.7, 14.8, 14.9, 15. , 15.1, 15.2, 15.3, 15.4,
15.5, 15.6, 15.7, 15.8, 15.9, 16. , 16.1, 16.2, 16.3,
16.4, 16.5, 16.6, 16.7, 16.8, 16.9, 17. , 17.1, 17.2,
17.3, 17.4, 17.5, 17.6, 17.7, 17.8, 17.9, 18. , 18.1,
18.2, 18.3, 18.4, 18.5, 18.6, 18.7, 18.8, 18.9, 19. ,
19.1, 19.2, 19.3, 19.4, 19.5, 19.6, 19.7, 19.8, 19.9,
20. , 20.1, 20.2, 20.3, 20.4, 20.5, 20.6, 20.7, 20.8,
20.9, 21. , 21.1, 21.2, 21.3, 21.4, 21.5, 21.6, 21.7,
21.8, 21.9, 22. , 22.1, 22.2, 22.3, 22.4, 22.5, 22.7,
22.8, 22.9, 23.1, 23.2, 23.4, 23.5, 23.6, 24.3, 24.9,
27.3, nan])
```

Уникальные значения всех датафреймов для столбца Prcpttn

```
Out[38]: array(['0.1', '0.2', '0.3', '0.4', '0.5', '0.6', '0.7', '0.8', '0.9',
'1.0', '10.0', '11.0', '12.0', '120.0', '13.0', '14.0', '15.0',
'16.0', '17.0', '18.0', '19.0', '2.0', '20.0', '21.0', '22.0',
'23.0', '24.0', '25.0', '26.0', '27.0', '28.0', '29.0', '3.0',
'30.0', '31.0', '32.0', '33.0', '34.0', '35.0', '36.0', '37.0',
'38.0', '39.0', '4.0', '40.0', '41.0', '42.0', '43.0', '44.0',
'45.0', '46.0', '47.0', '48.0', '49.0', '5.0', '50.0', '51.0',
'52.0', '53.0', '54.0', '55.0', '59.0', '6.0', '606.0', '63.0',
'66.0', '67.0', '7.0', '8.0', '9.0', '960.0', '989.0', '99.0',
'Осадков нет', 'Следы осадков', nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца Prcpttn_tdelt

```
Out[38]: array([ 1.,  2.,  3.,  6.,  9., 12., 15., 18., 24., nan])
```

Уникальные значения всех датафреймов для столбца Soil

```
Out[38]: array(['Сухая',
   'Сухая (без трещин, заметного количества пыли или сыпучего песка)',
   'Затопленная водой.', 'Ледяной покров на поверхности почвы.',
   'Несвязанная сухая пыль или песок не покрывают поверхность почвы полностью.',
   'Поверхность почвы влажная.', 'Поверхность почвы замерзшая.',
   'Поверхность почвы сырья (вода застаивается на поверхности и образует малые или большие лужи).',
   'Тонкий слой несвязанной сухой пыли или песка покрывает почву полностью.',
   'Чрезвычайно сухая, с трещинами.', nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца Soil_T

```
Out[38]: array([-29., -27., -24., -22., -19., -18., -16., -15., -14., -13., -12.,
   -11., -10., -9., -8., -7., -6., -5., -4., -3., -2., -1.,
   0., 1., 2., 3., 4., 5., 6., 7., 8., 9., 10.,
   11., 12., 13., 14., 15., 16., 17., 18., 19., 20., 21.,
   22., 23., 24., 26., nan])
```

Уникальные значения всех датафреймов для столбца Soil_cover

```
Out[38]: array(['Неровный слой слежавшегося или мокрого снега покрывает почву полностью.',
   'Неровный слой сухого рассыпчатого снега покрывает поверхность почвы полностью.',
   'Поверхность почвы преимущественно покрыта льдом.',
   'Ровный слой слежавшегося или мокрого снега покрывает поверхность почвы полностью.',
   'Ровный слой сухого рассыпчатого снега покрывает поверхность почвы полностью.',
   'Слежавшийся или мокрый снег (со льдом или без него), покрывающий менее половины поверхности почвы.',
   'Слежавшийся или мокрый снег (со льдом или без него), покрывающий по крайней мере половину поверхности почвы, но почва не
покрыта полностью.',
   'Снег покрывает поверхность почвы полностью; глубокие сугробы.',
   'Сухой рассыпчатый снег покрывает меньше половины поверхности почвы.',
   'Сухой рассыпчатый снег покрывает по крайней мере половину поверхности почвы (но не полностью).',
   nan], dtype=object)
```

Уникальные значения всех датафреймов для столбца Snow_height

```
Out[38]: array(['0.0', '1.0', '10.0', '101.0', '103.0', '11.0', '12.0', '123.0',
   '13.0', '14.0', '15.0', '16.0', '17.0', '18.0', '19.0', '2.0',
   '20.0', '21.0', '22.0', '23.0', '24.0', '25.0', '26.0', '27.0',
   '28.0', '29.0', '3.0', '30.0', '31.0', '32.0', '33.0', '34.0',
   '35.0', '36.0', '37.0', '38.0', '39.0', '4.0', '40.0', '41.0',
   '42.0', '43.0', '44.0', '45.0', '46.0', '47.0', '48.0', '49.0',
   '5.0', '50.0', '51.0', '52.0', '53.0', '54.0', '55.0', '56.0',
   '57.0', '58.0', '59.0', '597.0', '6.0', '60.0', '61.0', '62.0',
   '63.0', '64.0', '65.0', '66.0', '67.0', '68.0', '69.0', '7.0',
   '70.0', '71.0', '72.0', '73.0', '74.0', '75.0', '76.0', '77.0',
   '78.0', '79.0', '8.0', '81.0', '82.0', '9.0',
   'Измерение невозможно или неточно.', 'Менее 0.5',
   'Снежный покров не постоянный.', nan], dtype=object)
```

ВЫВОД

1. Все поля содержат значения NaN. Скорее всего это пропуски, но они также могут означать отсутствие метеорологического явления в момент наблюдения.

2. Отдельные параметры:

- T(T) Температура воздуха на высоте 2 м над поверхностью земли, градусов Цельсия: *числовые значения, однотипны, есть NaN;*
- P_station (Po) Атмосферное давление на уровне станции, мм рт. столба: *числовые значения, однотипны, есть NaN;*
- P_sea (P) Атмосферное давление, приведённое к среднему уровню моря, мм рт. столба: *числовые значения, однотипны, есть NaN;*
- P_drift (Pa) Барическая тенденция: изменение атмосферного давления за последние 3 часа, мм рт. столба: *числовые значения, однотипны, есть NaN;*
- Humid (U) Относительная влажность воздуха на высоте 2 м над поверхностью земли, %: *числовые значения, однотипны, есть NaN;*
- Wind_dir (DD) Направление ветра на высоте 10-12 м над поверхностью земли, усреднённое за 10 минут непосредственно перед наблюдением, румбы: *строковые значения, однотипны, есть NaN, можно формализовать и представить в числовом виде - размеры углов, - но необходимо также выразить "Штиль, безветрие";*
- Wind_speed, Ff Скорость ветра на высоте 10-12 м над поверхностью земли, усреднённая за 10 минут непосредственно перед наблюдением, м/с: *числовые значения, однотипны, есть NaN;*
- Gusts (ff10) Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за 10 минут непосредственно перед наблюдением, м/с: *числовые значения, однотипны, есть NaN;*
- Gusts_3h (ff3) Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за период между моментами наблюдения 3 часа, м/с: *числовые значения, однотипны, есть NaN;*
- Cloudness (N) Общая облачность, %: *строковые значения, однотипны, есть NaN, данные представлены преимущественно в виде диапазонов, можно формализовать и представить в числовом виде, но необходимо также выразить "Небо не видно из-за тумана и/или других метеорологических явлений";*
- Wthr_curr (WW) Текущая погода, сообщаемая метеостанцией: *строковые значения, однотипны, есть NaN, требуют обработки для вычленения данных, имеющих значение для темы модели, например, "Высота свежевыпавшего снега составляет [x,x] см." и подобных, после чего потребуется формализация и приведение в числовой вид;*
- Wthr_3h (W1) Прошедшая погода между сроками наблюдения 1: *строковые значения, однотипны, есть NaN, могут содержать данные о форме осадков;*
- Wthr_3h2 (W2) Прошедшая погода между сроками наблюдения 2: *строковые значения, однотипны, есть NaN, могут содержать данные о форме осадков;*
- T_min (Tn) Минимальная температура воздуха за прошёлый период не более 12 часов, градусов Цельсия: *числовые значения, однотипны, есть NaN;*

- T_max (Tx) Максимальная температура воздуха за прошедший период не более 12 часов, градусов Цельсия: числовые значения, однотипны, есть NaN;
 - Cl_Cumls (Cl) Слоисто-кучевые, слоистые, кучевые и кучево-дождевые облака: строковые значения, однотипны, есть NaN;
 - Cl_viewed (Nh) Количество всех наблюдающихся облаков Cl, или при отсутствии облаков Cl, количество всех наблюдающихся облаков Cm, % - строковые значения, однотипны, есть NaN;
 - Cl_bottom (H) Высота основания самых низких облаков, м: строковые значения, однотипны, есть NaN - **данные представлены преимущественно в виде диапазонов, можно формализовать и представить в числовом виде, но необходимо также выразить "2500 или более, или облаков нет.", "Менее 50";**
 - Cl_cumls_hi (Cm) Высоко-кучевые, высоко-слоистые и слоисто-дождевые облака: строковые значения, однотипны, есть NaN
 - Cl_cirrus (Ch) Перистые, перисто-кучевые и перисто-слоистые облака: строковые значения, однотипны, есть NaN
 - Visibility (VV) Горизонтальная дальность видимости, км.: значения разнородны (числовые и строковые), **необходимо формализовать численно выражить строковые значения;**
 - Dew_poin (Td) Температура точки росы на высоте 2 м над поверхностью земли, градусов Цельсия: числовые значения, однотипны, есть NaN;
 - Prcptn (RRR) Количество выпавших осадков, мм: строковые(!) значения, однотипны, есть NaN, **можно формализовать и представить в числовом виде, но необходимо также выразить "Осадков нет" и "Следы осадков";**
 - Prcptn_tdelt (tR) Период времени, за который выпало указанное количество осадков, ч.: числовые значения, однотипны, есть NaN;
 - Soil (E) Состояние поверхности почвы без снега или измеримого ледяного покрова: строковые значения, однотипны, есть NaN;
 - Soil_T (Tg) Минимальная температура поверхности почвы за ночь, градусов Цельсия: числовые значения, однотипны, есть NaN;
 - Soil_cover (E') Состояние поверхности почвы со снегом или измеримым ледяным покровом: строковые значения, однотипны, есть NaN - **можно формализовать и представить в числовом виде;**
 - Snow_height (sss) Высота снежного покрова, см: строковые (!) значения, однотипны, есть NaN , **можно формализовать и представить в числовом виде, но необходимо также выразить словесные описания".**
1. Часть данных, например, количество осадков, содержат аномальные значения, правдоподобность которых сложно восстановить логически. Поэтому данные об описании погоды (не имеющие отношения к цели модели) нужно сохранить до момента восстановления пропущенных и ошибочных значений.

Предлагается

1. Проверить корреляцию и, возможно, исключить из расчётов параметры:

- P_station (Po) Атмосферное давление на уровне станции, мм рт. столба - вычисляемый из P_sea (P) давления, приведённого к уровню моря.
 - P_drift (Pa) Барическая тенденция: изменение атмосферного давления за последние 3 часа, мм рт. столба - вычисляемый из данных на начало и конец периода между наблюдениями P_sea (P) давления, приведённого к уровню моря.
 - Gusts (ff10) Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за 10 минут непосредственно перед наблюдением, м/с - возможно, не связан с целью модели
 - Gusts_3h (ff3) Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за период между моментами наблюдения 3 часа, м/с - возможно, не связан с целью модели
 - Dew_point (Td) Температура точки росы на высоте 2 м над поверхностью земли, градусов Цельсия - возможно, не связан с целью модели, можно вычислить
1. Привести к численному виду данные, выраженные формализованными строками.
2. Данные столбцов ниже могут содержать важную информацию о характере осадков и конденсации тумана. Необходимо определиться, насколько эти данные нужны для модели, в таком случае - формализовать через регулярные выражения.
- Wthr_curr (WW) Текущая погода, сообщаемая метеостанцией - словесное описание
 - Wthr_3h (W1) Прошедшая погода между сроками наблюдения 1 - словесное описание
 - Wthr_3h2 (W2) Прошедшая погода между сроками наблюдения 2 - словесное описание
 - Cl_Cumls (Cl) Слоисто-кучевые, слоистые, кучевые и кучево-дождевые облака - возможно, не связан с целью модели
 - Cl_viewed (Nh) Количество всех наблюдающихся облаков Cl, или при отсутствии облаков Cl, количество всех наблюдающихся облаков Cm, % - возможно, не связан с целью модели
 - Cl_bottom (H) Высота основания самых низких облаков, м - возможно, не связан с целью модели
 - Cl_cumls_hi (Cm) Высоко-кучевые, высоко-слоистые и слоисто-дождевые облака - возможно, не связан с целью модели
 - Cl_cirrus (Ch) Перистые, перисто-кучевые и перисто-слоистые облака - возможно, не связан с целью модели

1.1.3. Необходимое преобразование типов данных, разделение категориальных и числовых данных, упрощение наименований категорий (классов).

Как видно из информации о данных в датафреймах и об их уникальных значениях, часть данных, которая, теоретически, должна быть представлена в числовом формате, представлена в формате объекта или строки. Здесь *не идёт речь о категориальных данных*. Но у нас есть данные, в которых смешаны строки и цифры, либо те, которые трактуются как строки из-за наличия в них нескольких строковых значений. У нас также имеются данные, выраженные словами, но для которых существуют другие, числовые измерения.

Все преобразования будем делать в копии исходного словаря датафреймов

```
In [39]: dict_df_locations = {}
for name, data in dict_df_data.items():
    dict_df_locations[name] = data.copy(deep=True) # Копирование датафрейма, включая все его данные
```

Для удобства преобразований, сделаем столбец с локальным временем на метеостанции индексом, добавим индекс все отсутствующие моменты наблюдений от начала и до конца архивов.

```
In [40]: # Массив с датами и временем за всю историю наблюдений
# минимальная и максимальная дата наблюдений
start_date = df_stations.beg_date.min()
finish_date = df_stations.end_date.max() + pd.Timedelta(1, 'day') # Добавим еще один день к концу периода
# искомый массив
arr_dates = np.arange(start=start_date, stop=finish_date, step = np.timedelta64(3, 'h'), dtype='datetime64[ns]')

for name, data in dict_df_locations.items(): # по элементам словаря с данными DFs
    data.set_index(data.columns[0], inplace=True) # сделаем 1й столбец индексом (дата и время наблюдения)
    # Создадим временный пустой датафрейм с индексом по всему диапазону моментов наблюдения
    df_time_tmp = pd.DataFrame(index=arr_dates).sort_index(ascending = False)
    # Объединим текущий датафрейм с временным, использую индекс с полным временным рядом.
    dict_df_locations[name] = data.merge(df_time_tmp, how='right', left_index=True, right_index=True)
    idx_name = f'{name[3:]}_Local_time' # Создадим строку для названия индекса
    dict_df_locations[name].rename_axis(idx_name, inplace=True) # Установим название индекса в каждом DF
del df_time_tmp # удаляем df_tmp, чтобы не засорять память (и самим не запутаться)
## Проверка
# for name in dict_df_locations.keys():
#     dict_df_locations[name].tail(10)
```

1.1.3.1. Преобразование параметра Wind_dir в более короткие строки, а также в цифровые значения

Помимо **сокращения строки категорий направления ветра** (для минимизации) преобразуем направления ветра в **единицу измерения "Градусы"**: $2\pi = 360$, а также **в тысячиные**: $2\pi = 6000$.

ВАЖНО: мы всегда измеряем направление ветра по углу, откуда дует ветер. Угол отсчитывается по часовой стрелке от направления на Север.

Количество значений небольшое, поэтому для преобразования просто создадим словарь.

```
In [41]: # Словарь соответствия румбов списку
# более кратких значений, направлений ветра в градусах и в тысячных - для будущих преобразований.

dict_wind_dir = {'Ветер, дующий с севера': ['N', 0.0, 0.0],
                 'Ветер, дующий с северо-северо-востока': ['NNE', 22.5, 375.0],
                 'Ветер, дующий с северо-востока': ['NE', 45.0, 750.0],
                 'Ветер, дующий с востоко-северо-востока': ['ENE', 67.5, 1125.0],
                 'Ветер, дующий с востока': ['E', 90.0, 1500],
                 'Ветер, дующий с востоко-юго-востока': ['ESE', 112.5, 1875.0],
                 'Ветер, дующий с юго-востока': ['SE', 135.0, 2250.0],
                 'Ветер, дующий с юго-юго-востока': ['SSE', 157.5, 2625.0],
                 'Ветер, дующий с юга': ['S', 180.0, 3000.0],
                 'Ветер, дующий с юго-юго-запада': ['SSW', 202.5, 3375.0],
                 'Ветер, дующий с юго-запада': ['SW', 225.0, 3750.0],
                 'Ветер, дующий с западо-юго-запада': ['WSW', 247.5, 4125.0],
                 'Ветер, дующий с запада': ['W', 270.0, 4500.0],
                 'Ветер, дующий с западо-северо-запада': ['WNW', 292.5, 4875.0],
                 'Ветер, дующий с северо-запада': ['NW', 315.0, 5250.0],
                 'Ветер, дующий с северо-северо-запада': ['NNW', 337.5, 5625.0],
                 'Штиль, безветрие': ['штиль', 360.0, 6000] # Специальное значение, скоростью ветра = 0
                }
```

```
In [42]: # Функция преобразования направления ветра в румбах
def wind_dir_to_other(direction_r = 'Штиль, безветрие'):
    """
    Принимает: направление ветра в румбах в исходной формулировке,
    Возвращает: кортеж из
    - строки с короткой формулировкой [0],
    - направления ветра в градусах [1],
    - направления ветра в тысячных[2].
    """

    direction_d = (dict_wind_dir[direction_r][0], dict_wind_dir[direction_r][1], dict_wind_dir[direction_r][2])
    return direction_d
```

```
In [43]: for name in dict_df_locations.keys(): # по ключам словаря с данными DFs
    # присваиваем значения колонке Wind_dir в DF dict_df_locations
    # выбираем значения кроме nan

    # Вставляем столбец со значением направления ветра в градусах
    dict_df_locations[name].insert(dict_df_locations[name].columns.get_loc("Wind_dir")+1, # индекс колонки после Win_dir
                                    "Wind_dir360", # название вставляемого столбца
                                    dict_df_locations[name].Wind_dir[(dict_df_locations[name].Wind_dir.notna())]\
```

```

    .apply(lambda x: wind_dir_to_other(x)[1])) # значение в градусах, кроме NaN

    # Вставляем столбец со значением направления ветра в тысячных
    dict_df_locations[name].insert(dict_df_locations[name].columns.get_loc("Wind_dir") + 2, # индекс колонки после Wind_dir
                                    "Wind_dir6k", # название вставляемого столбца
                                    dict_df_locations[name].Wind_dir[(dict_df_locations[name].Wind_dir.notna())]\n
                                    .apply(lambda x: wind_dir_to_other(x)[2])) # значение в тысячных, кроме NaN

    # сокращённая строка направления ветра (должна выполняться последней, так как иначе не сможем преобразовать в числа)
    dict_df_locations[name].Wind_dir \
    = dict_df_locations[name].Wind_dir[(dict_df_locations[name].Wind_dir.notna())]\n
    .apply(lambda x: wind_dir_to_other(x)[0])

#     dict_df_locations[name].sample(2)

```

1.1.3.2. Преобразование параметра Cloudness

замена строк на более удобные строковые значения

```
In [44]: # Создадим словарь для замены значений
dict_replacement = {'10% или менее, но не 0': 'больше 0 до 10%',
                    '90 или более, но не 100%': 'от 90 менее 100%',
                    'Небо не видно из-за тумана и/или других метеорологических явлений.': 'Небо вне видимости',
                    'Облаков нет.': '0%'}

for name in dict_df_locations.keys():
    dict_df_locations[name] = dict_df_locations[name].replace({"Cloudness": dict_replacement})
```

1.1.3.3. Преобразование параметра Wthr_curr в столбцы с названием категорий и числовыми измерениями

Выше, при анализе уникальных значений параметров (п. 1.1.2), мы увидели, что строковые значения столбца Wthr_curr описывают не только состояние погоды на момент наблюдения и между наблюдениями, но также и указывают на наличие осадков или конденсата, их формы, интенсивности, отложения и выпадения твёрдых осадков и виды твёрдых осадков. Эти строки включают в себя от одного до трёх формализованных предложений.

```
In [45]: reg_string = r'\.\s+' # Регулярное выражение для разбиения текста на предложения (по разделяющим точкам)
comp_regex = re.compile(reg_string) # Компилируем регулярное выражение

# создаём список из списков предложений для каждой исходной строки
list_sentences = []
# Проходим по всем архивам и вычисляем отдельные предложения в столбце Wthr_curr
```

```
for df_name, df_data in dict_df_locations.items():
    # Заполняем список предложений через extend, чтобы не добавлять а расширять список списком
    list_sentences.extend(list(map(comp_regex.split, df_data.Wthr_curr.dropna())))

# преображаем список в датафрейм
df_text = pd.DataFrame(list_sentences)
# Переименуем столбцы
df_text = df_text.rename(columns = {0:'txt0', 1:'txt1', 2:'txt2'})
# Заполним NaN
df_text = df_text.fillna('n/a')
```

In [46]: # Выводим уникальные значения 1, 2 и 3го предложения.

```
for col_name in df_text.columns:
    df_text[col_name].unique()
```

```
Out[46]: array(['',
   'Более или менее сплошной слой приземного или ледяного тумана на станции, на море или на суше, высотой не более 2 м над сушей или не более 10 м над морем',
   "'Облака в целом рассеиваются или становятся менее развитыми",
   "'Ливневый(ые) дождь(и)', 'Состояние неба в общем не изменилось',
   "'Ливневый(ые) дождь(и) слабый(ые) в срок наблюдения или за последний час',
   "'Дымка', 'Дождь (незамерзающий) неливневый',
   "'Дождь незамерзающий непрерывный слабый в срок наблюдения',
   "'Ливневый снег или ливневый дождь и снег',
   "'Ливневый снег слабый в срок наблюдения или за последний час',
   "'Ливневый снег умеренный или сильный в срок наблюдения или за последний час',
   "'Облака в целом образовывались или развивались',
   "'Снег непрерывный слабый в срок наблюдения',
   "'Слабый или умеренный поземок в целом низкий (ниже уровня глаз наблюдателя)',
   "'Морось незамерзающая непрерывная слабая в срок наблюдения',
   "'Снег неливневый', 'Снег с перерывами слабый в срок наблюдения',
   "' Диаметр изморозевого отложения составляет 5 мм.',
   "' Диаметр изморозевого отложения составляет 4 мм.',
   "' Диаметр изморозевого отложения составляет 1 мм.',
   "' Диаметр изморозевого отложения составляет 3 мм.',
   "'Снег непрерывный умеренный в срок наблюдения',
   "' Диаметр отложения мокрого снега составляет 8 мм.',
   "' Диаметр отложения мокрого снега составляет 7 мм.',
   "'Дождь незамерзающий с перерывами слабый в срок наблюдения',
   "'Ключья приземного или ледяного тумана на станции, на море или на суше, высотой не более 2 м над сушей или не более 10 м над морем',
   "'Гроза слабая или умеренная без града, но с дождем и/или снегом в срок наблюдения',
   "'Гроза (с осадками или без них)',
   "'Гроза, но без осадков, в срок наблюдения',
   "'Морось (незамерзающая) или снежные зерна неливневые',
   "' Диаметр отложения при гололеде составляет 1 мм.',
   "'Замерзающая морось или замерзающий дождь неливневые',
   "' Диаметр изморозевого отложения составляет 2 мм.',
   "' Диаметр отложения мокрого снега составляет 4 мм.',
   "'Морось незамерзающая с перерывами слабая в срок наблюдения',
   "'Дождь замерзающий слабый',
   "' Диаметр отложения мокрого снега составляет 9 мм.',
   "'Ливневый град, или дождь и град',
   "'Ливневый(ые) дождь(и) со снегом слабый(ые) в срок наблюдения или за последний час',
   "'Туман или ледяной туман, неба не видно, без заметного изменения интенсивности в течение последнего часа',
   "'Туман или ледяной туман, небо видно, начался или усилился в течение последнего часа',
   "' Диаметр отложения мокрого снега составляет 2 мм.',
   "'Дождь незамерзающий непрерывный умеренный в срок наблюдения',
   "'Ливневый(ые) дождь(и) умеренный(ые) или сильный(ые) в срок наблюдения или за последний час',
```

'Морось замерзающая слабая',
'Морось незамерзающая с перерывами умеренная в срок наблюдения',
' Диаметр отложения при гололеде составляет 2 мм.',
'Туман или ледяной туман, неба не видно, начался или усилился в течение последнего часа',
' Диаметр отложения мокрого снега составляет 1 мм.', 'Осадки ',
'Морось незамерзающая непрерывная умеренная в срок наблюдения',
'Снег непрерывный сильный в срок наблюдения',
'Ливневая снежная крупа или небольшой град с дождем или без него, или дождь со снегом слабые в срок наблюдения или за последний час',
'Туман или ледяной туман, небо видно, ослабел за последний час',
'Слабый дождь в срок наблюдения', 'Туман или ледяной туман',
'Слабая или умеренная снежная низовая метель в целом высокая (выше уровня глаз наблюдателя)',
'Туман или ледяной туман, небо видно, без заметного изменения интенсивности в течение последнего часа',
'Умеренный или сильный дождь в срок наблюдения',
' Диаметр изморозевого отложения составляет 7 мм.',
' Диаметр изморозевого отложения составляет 6 мм.',
' Диаметр изморозевого отложения составляет 8 мм.',
'Развитие облачности не наблюдалось или не наблюдается',
'Ливневый град с дождем или без него, или дождь со снегом без грома слабый в срок наблюдения или за последний час',
'Туман или ледяной туман, неба не видно, ослабел за последний час',
'Снег с перерывами умеренный в срок наблюдения',
'Видимость ухудшена дымом, например: от степных или лесных пожаров, промышленных предприятий или вулканическим пеплом',
'Мгла', 'Морось и дождь слабые',
'Гроза вместе с пыльной или песчаной бурей в срок наблюдения',
'Ливневый(ые) дождь(и) очень сильный(ые) в срок наблюдения или за последний час',
'Дождь или морось со снегом слабые',
'Ливневая снежная крупа или небольшой град с дождем или без него, или дождь со снегом умеренные или сильные в срок наблюдения или за последний час',
'Ливневый(ые) дождь(и) со снегом умеренный(ые) или сильный(ые) в срок наблюдения или за последний час',
' Диаметр изморозевого отложения составляет 12 мм.',
'Видна молния, грома не слышино',
'Туман с отложением изморози, неба не видно',
' Диаметр отложения мокрого снега составляет 24 мм.',
' Диаметр отложения мокрого снега составляет 3 мм.',
' Диаметр изморозевого отложения составляет 11 мм.',
'Снежные зерна (с туманом или без него)', 'Ледяная крупа',
'Туман с отложением изморози, небо видно',
' Диаметр отложения при гололеде составляет 3 мм.',
'Морось и дождь умеренные или сильные',
' Диаметр изморозевого отложения составляет 17 мм.',
' Диаметр изморозевого отложения составляет 10 мм.',
' Диаметр изморозевого отложения составляет 13 мм.',
' Диаметр изморозевого отложения составляет 9 мм.',
'Алмазная пыль (с туманом или без него)',

' Диаметр изморозевого отложения составляет 14 мм.',
'Морось замерзающая умеренная или сильная',
'Морось незамерзающая непрерывная сильная в срок наблюдения',
'Дождь или морось со снегом умеренные или сильные',
'Туман или обрывки ледяного тумана', 'Туман ',
'Мгла или дым, или пыль, взвешенная в воздухе, дальность видимости 1 км или более',
'Дождь незамерзающий слабый',
' Диаметр отложения при гололеде составляет 9 мм.',
'Замерзающая морось или замерзающий дождь',
'Дождь незамерзающий с перерывами умеренный в срок наблюдения',
' Диаметр отложения мокрого снега составляет 33 мм.',
' Диаметр отложения при гололеде составляет 4 мм.',
'Дождь замерзающий умеренный или сильный',
'Сильная снежная низовая метель в целом высокая (выше уровня глаз наблюдателя)',
'Дождь со снегом или ледяная крупа неливневые',
' Диаметр отложения мокрого снега составляет 18 мм.',
'Снег с перерывами сильный в срок наблюдения',
'Дождь незамерзающий непрерывный сильный в срок наблюдения',
'Снег слабый',
' Диаметр отложения мокрого снега составляет 11 мм.',
'Ниаких особых явлений погоды не наблюдалось',
'Сильная пыльная или песчаная буря ослабела за последний час',
'Сильный поземок в целом низкий (ниже уровня глаз наблюдателя)',
'Отдельные кристаллы снега в виде звездочек (с туманом или без него)',
' Диаметр изморозевого отложения составляет 15 мм.',
' Диаметр отложения мокрого снега составляет 13 мм.',
' Диаметр отложения мокрого снега составляет 25 мм.',
'Гроза сильная без града, но с дождем и/или снегом в срок наблюдения',
'Шквалы на станции или в поле зрения за последний час или в срок наблюдения',
'Дождь незамерзающий с перерывами сильный в срок наблюдения',
'Ливневый град с дождем или без него, или дождь со снегом без грома умеренный или сильный в срок наблюдения или за последний час',
' Высота свежевыпавшего снега составляет 0.1 см.',
' Высота свежевыпавшего снега составляет 0.2 см.',
' Высота свежевыпавшего снега составляет 2 см.',
' Высота свежевыпавшего снега составляет 0.3 см.',
' Высота свежевыпавшего снега составляет 3 см.',
' Высота свежевыпавшего снега составляет 4 см.',
' Высота свежевыпавшего снега составляет 0.4 см.',
' Высота свежевыпавшего снега составляет 1 см.',
' Высота свежевыпавшего снега составляет 0.6 см.',
' Высота свежевыпавшего снега составляет 0.5 см.',
' Диаметр отложения при гололеде составляет 0.4 мм.',
' Диаметр отложения при гололеде составляет 14 мм.'

' Высота свежевыпавшего снега составляет 0.4 см',
' Высота свежевыпавшего снега составляет 8 см.',
' Диаметр смешанного отложения составляет 3 мм.',
'Гроза слабая или умеренная с градом в срок наблюдения',
'Слабый снег или дождь со снегом или град в срок наблюдения',
'Слабая или умеренная пыльная или песчаная буря ослабела за последний час',
'Морось незамерзающая с перерывами сильная в срок наблюдения',
' Диаметр отложения при гололеде составляет 10 мм.',
' Диаметр отложения при гололеде составляет 11 мм.',
' Диаметр отложения мокрого снега составляет 10 мм.',
' Диаметр отложения мокрого снега составляет 5 мм.',
'Туман или ледяной туман на расстоянии в срок наблюдения, простирающийся выше уровня глаз наблюдателя',
'Осадки в поле зрения, достигающие поверхности земли или моря на расстоянии более 5 км от станции',
'Умеренный или сильный снег или дождь со снегом, или град в срок наблюдения',
'Осадки в поле зрения, достигающие поверхности земли или моря поблизости, но не на самой станции',
'Пыль или песок, поднятые ветром на или вблизи станции в срок наблюдения, но нет хорошо развитого пыльного или песчаного вихря (вихрей), и никакой пыльной или песчаной бури не видно; или, в случае сообщения с судна или с береговой станции, брызги, переносимые ветром на станции',
'Пыль, взвешенная в воздухе на обширном пространстве, но не поднятая ветром на или вблизи станции в срок наблюдения',
'Хорошо развитый пыльный или песчаный вихрь (вихри) на или вблизи станции в течение последнего часа или в срок наблюдения, но пыльной или песчаной бури нет',
' Диаметр отложения при гололеде составляет 13 мм.',
'Мгла или дым, или пыль, взвешенная в воздухе, дальность видимости менее 1 км',
'Слабая или умеренная пыльная или песчаная буря началась или усилилась в течение последнего часа'],
dtype=object)

```
Out[46]: array(['н/а', '', 'Диаметр изморозевого отложения составляет 5 мм.',  
   'Диаметр изморозевого отложения составляет 1 мм.',  
   'Диаметр отложения мокрого снега составляет 8 мм.',  
   'Диаметр отложения мокрого снега составляет 2 мм.',  
   'Диаметр отложения мокрого снега составляет 3 мм.',  
   'Диаметр отложения мокрого снега составляет 7 мм.',  
   'Диаметр отложения мокрого снега составляет 4 мм.',  
   'Диаметр отложения мокрого снега составляет 1 мм.',  
   'Диаметр отложения при гололеде составляет 4 мм.',  
   'Диаметр отложения при гололеде составляет 1 мм.',  
   'Диаметр изморозевого отложения составляет 4 мм.',  
   'Диаметр отложения мокрого снега составляет 9 мм.',  
   'Диаметр отложения при гололеде составляет 2 мм.',  
   'Диаметр изморозевого отложения составляет 3 мм.',  
   'Диаметр изморозевого отложения составляет 2 мм.',  
   'Диаметр отложения мокрого снега составляет 10 мм.',  
   'Диаметр отложения мокрого снега составляет 11 мм.',  
   'Гроза в течение последнего часа, но не в срок наблюдения',  
   'Диаметр отложения при гололеде составляет 3 мм.',  
   'Диаметр отложения мокрого снега составляет 6 мм.',  
   'Диаметр изморозевого отложения составляет 8 мм.',  
   'Диаметр изморозевого отложения составляет 6 мм.',  
   'Диаметр отложения мокрого снега составляет 16 мм.',  
   'Диаметр отложения мокрого снега составляет 24 мм.',  
   'Диаметр отложения мокрого снега составляет 18 мм.',  
   'Диаметр отложения мокрого снега составляет 21 мм.',  
   'Максимальный диаметр градин составляет 10 мм.',  
   'Диаметр изморозевого отложения составляет 11 мм.',  
   'Диаметр смешанного отложения составляет 7 мм.',  
   'Диаметр отложения мокрого снега составляет 15 мм.',  
   'Максимальный диаметр градин составляет 3 мм.',  
   'Диаметр отложения мокрого снега составляет 5 мм.',  
   'Максимальный диаметр градин составляет 2 мм.',  
   'Максимальный диаметр градин составляет 13 мм.',  
   'Диаметр изморозевого отложения составляет 12 мм.',  
   'Диаметр изморозевого отложения составляет 7 мм.',  
   'Диаметр изморозевого отложения составляет 17 мм.',  
   'Максимальный диаметр градин составляет 1 мм.',  
   'Высота свежевыпавшего снега составляет 10 см.',  
   'Диаметр изморозевого отложения составляет 13 мм.',  
   'Диаметр изморозевого отложения составляет 9 мм.',  
   'Максимальный диаметр градин составляет 6 мм.',  
   'Диаметр изморозевого отложения составляет 14 мм.',  
   'Диаметр отложения при гололеде составляет 9 мм.',
```

'Диаметр отложения при гололеде составляет 7 мм.',
'Диаметр отложения мокрого снега составляет 33 мм.',
'Диаметр отложения мокрого снега составляет 29 мм.',
'Максимальный диаметр градин составляет 5 мм.',
'Диаметр отложения мокрого снега составляет 20 мм.',
'Диаметр отложения мокрого снега составляет 13 мм.',
'Диаметр отложения мокрого снега составляет 32 мм.',
'Диаметр отложения мокрого снега составляет 17 мм.',
'Диаметр отложения при гололеде составляет 13 мм.',
'Диаметр отложения при гололеде составляет 11 мм.',
'Диаметр отложения при гололеде составляет 6 мм.',
'Максимальный диаметр градин составляет 4 мм.',
'Диаметр отложения мокрого снега составляет 12 мм.',
'Диаметр отложения мокрого снега составляет 25 мм.',
'Максимальный диаметр градин составляет 9 мм.',
'Диаметр смешанного отложения составляет 4 мм.',
'Высота свежевыпавшего снега составляет 4 см.',
'Высота свежевыпавшего снега составляет 0.5 см.',
'Высота свежевыпавшего снега составляет 2 см.',
'Высота свежевыпавшего снега составляет 0.4 см.',
'Высота свежевыпавшего снега составляет 1 см.',
'Высота свежевыпавшего снега составляет 0.3 см.',
'Высота свежевыпавшего снега составляет 3 см.',
'Высота свежевыпавшего снега составляет 8 см.',
'Высота свежевыпавшего снега составляет 0.1 см.',
'Диаметр отложения при гололеде составляет 2 мм',
'Высота свежевыпавшего снега составляет 5 см.',
'Максимальный диаметр градин составляет 7 мм.',
'Высота свежевыпавшего снега составляет 7 см.',
'Высота свежевыпавшего снега составляет 6 см.',
'Высота свежевыпавшего снега составляет 4 см',
'Высота свежевыпавшего снега составляет 0.6 см.',
'Высота свежевыпавшего снега составляет 0.2 см.',
'Высота свежевыпавшего снега составляет 3 см',
'Высота свежевыпавшего снега составляет 0.5 см',
'Высота свежевыпавшего снега составляет 1 см',
'Диаметр отложения при гололеде составляет 14 мм.',
'Высота свежевыпавшего снега составляет 0.1 см',
'Высота свежевыпавшего снега составляет 0.3 см',
'Высота свежевыпавшего снега составляет 6 см',
'Высота свежевыпавшего снега составляет 2 см',
'Диаметр смешанного отложения составляет 1 мм.',
'Диаметр смешанного отложения составляет 2 мм.',
'Диаметр смешанного отложения составляет 3 мм.',

```
'Максимальный диаметр градин составляет 12 мм.',  
'Диаметр отложения при гололеде составляет 10 мм.',  
'Диаметр отложения при гололеде составляет 8 мм.',  
'Диаметр отложения при гололеде составляет 5 мм.',  
'В течение последнего часа туман на станции не наблюдался',  
'Диаметр смешанного отложения составляет 11 мм.',  
'Диаметр отложения мокрого снега составляет 30 мм.'], dtype=object)  
Out[46]: array(['n/a', '', 'Высота свежевыпавшего снега составляет 1 см.',  
   'Диаметр отложения при гололеде составляет 2 мм.',  
   'Диаметр отложения при гололеде составляет 3 мм.',  
   'Диаметр отложения при гололеде составляет 1 мм.',  
   'Диаметр отложения при гололеде составляет 14 мм.'], dtype=object)
```

Как видно из выведенных значений, хотя предложения встречаются вперемешку, у них есть чёткая структура. В первую очередь нас интересует выпадающие осадки. Выпадение осадков и конденсация влаги на поверхности характеризуется двумя параметрами: диаметр (мм) и высота (см).

Проверим, можем ли мы использовать значения Высоты свежевыпавшего снега, упомянутые выше.

```
In [47]: reg_expr1=r'Высота свежевыпавшего снега составляет' # Регулярное выражение  
regex1 = re.compile(reg_expr1) # Скомпилированное регулярное выражение  
  
for name in dict_df_locations.keys(): # По архивам метеостанций  
    # Выедем количество записей 'Высота свежевыпавшего снега составляет' и имя датафрейма  
    dict_df_locations[name].Wthr_curr.str.contains(regex1).sum(), name
```

```
Out[47]: (0, 'df_V_Volochek')  
Out[47]: (1, 'df_Staritsa')  
Out[47]: (0, 'df_Kashyn')  
Out[47]: (0, 'df_Tver')  
Out[47]: (0, 'df_Klin')  
Out[47]: (235, 'df_Dmitrov')  
Out[47]: (0, 'df_Volokolamsk')  
Out[47]: (0, 'df_Mozhaisk')
```

```
Out[47]: (0, 'df_N_Jerusalem')
Out[47]: (0, 'df_Nemchinovka')
Out[47]: (0, 'df_Naro_Fominsk')
Out[47]: (0, 'df_Serpukhov')
```

ВЫВОД

Значения строки *Высота свежевыпавшего снега составляет* не может быть нами использовано, так как систематически упоминается только в одном архиве: Дмитров.

Попытаемся вывести данные об отложениях осадков в отдельные столбцы: вид и диаметр

```
In [48]: reg_expr1=r'Диаметр\D+отложения' # Регулярное выражение (Ищем варианты 'Диаметр...отложения...составляет')
regex1 = re.compile(reg_expr1) # Скомпилированное регулярное выражение

for name in dict_df_locations.keys(): # По архивам метеостанций
    # Выедем количество записей 'Диаметр...отложения' и имя датафрейма
    dict_df_locations[name].Wthr_curr.str.contains(regex1).sum(), name
```

```
Out[48]: (525, 'df_V_Volochek')
Out[48]: (464, 'df_Staritsa')
Out[48]: (166, 'df_Kashyn')
Out[48]: (536, 'df_Tver')
Out[48]: (265, 'df_Klin')
Out[48]: (169, 'df_Dmitrov')
Out[48]: (66, 'df_Volokolamsk')
Out[48]: (52, 'df_Mozhaisk')
Out[48]: (190, 'df_N_Jerusalem')
Out[48]: (22, 'df_Nemchinovka')
Out[48]: (153, 'df_Naro_Fominsk')
```

Out[48]: (112, 'df_Serpukhov')

ВЫВОД

Существует достаточно малое количество наблюдений об отложениях осадков в строках *Wthr_curr*. Тем не менее, можно попробовать вывести эту информацию.

Выведем в отдельную колонку вид отложения осадков

```
In [49]: # Регулярные выражения
reg_expr1 = r'((?<=диаметр\s).+(?=составляет))' # ищет строку между 'диаметр' и 'составляет'
reg_expr2 = r'диаметр\d+([\d,]+)' # все цифры где-то после слова диаметр; искл. град - case sensitive

# Компилируем регулярные выражения
regex1 = re.compile(reg_expr1)
regex2 = re.compile(reg_expr2)

for name in dict_df_locations.keys(): # по ключам словаря с данными DFs
    # присваиваем значения колонке prcpttn_depo в DF dict_df_locations
    # выбираем значения кроме nan

    # Вставляем столбец с описанием отложений
    dict_df_locations[name].insert(dict_df_locations[name].columns.get_loc("Wthr_curr")+1, # столбец после Wthr_curr
                                    "Prcpttn_depo", # название вставляемого столбца
                                    dict_df_locations[name].Wthr_curr[(dict_df_locations[name].Wthr_curr.notna())]\n
                                    .str.extract(regex1, expand=False)) # из Wthr_curr берём соответствие regex1

    # Вставляем столбец с диаметром отложений
    dict_df_locations[name].insert(dict_df_locations[name].columns.get_loc("Prcpttn_depo")+1, #столбец после Prcpttn_depo
                                    "Depo_diam_mm", # название вставляемого столбца
                                    dict_df_locations[name].Wthr_curr[(dict_df_locations[name].Wthr_curr.notna())]\n
                                    .str.extract(regex2, expand=False)) # из Wthr_curr берём соответствие regex2

    # Преобразуем значение Depo_diam_mm в float
    dict_df_locations[name].Depo_diam_mm = dict_df_locations[name].Depo_diam_mm.astype('float')

    # Изменим форму и согласование слов в столбце Prcpttn_depo (без pymorphy2)
    dict_df_locations[name].Prcpttn_depo = dict_df_locations[name].Prcpttn_depo.str.replace('отложения', 'отложение')
    dict_df_locations[name].Prcpttn_depo = dict_df_locations[name].Prcpttn_depo.str.replace('смешанного', 'смешанное')
    dict_df_locations[name].Prcpttn_depo = dict_df_locations[name].Prcpttn_depo.str.replace('изморозевого', 'изморозевое')
```

Выведем в отдельную колонку информацию об осадках

```
In [50]: # Регулярные выражения
reg_expr1 = r'(.\+\.\. {1})' # Все символы до точки только 1 раз

# Компилируем регулярные выражения
regex1 = re.compile(reg_expr1)

for name in dict_df_locations.keys(): # по ключам словаря с данными DFs
    # присваиваем значения колонке prcptn_depo в DF dict_df_Locations
    # выбираем значения кроме nan

    # Вставляем столбец с описанием осадков и схожих явлений
    dict_df_locations[name].insert(dict_df_locations[name].columns.get_loc("Wthr_curr") + 1, # столбец после Wthr_curr
                                    "Prcptn_like", # название вставляемого столбца
                                    dict_df_locations[name].Wthr_curr[(dict_df_locations[name].Wthr_curr.notna())]\.
                                    str.extract(regex1, expand=False)) # разбиваем строку
```

Заменим NaN на 'n/a' и 0 (по смыслу) во вновь созданных столбцах, там где из столба "Wthr_curr" не было вычислено значений.

```
In [51]: for name in dict_df_locations.keys():
    # Заменяем в значении Wthr_curr пробел на 'n/a'
    dict_df_locations[name].Wthr_curr.replace({' ': 'n/a'}, inplace = True)
    # Там, где Wthr_curr не равно NaN заменяем значения NaN в колонках согласно словарю:
    dict_df_locations[name][(dict_df_locations[name].Wthr_curr.notna())] = \
        dict_df_locations[name][(dict_df_locations[name].Wthr_curr.notna())].fillna({'Prcptn_like': 'n/a',
                                                                                   'Prcptn_depo': 'n/a',
                                                                                   'Depo_diam_mm': 0.})
```

1.1.3.4. Преобразование параметра Visibility в dtype = float

замена строк на условные числовые значения

```
In [52]: # Преобразуем значения в столбце Visibility
for name in dict_df_locations.keys(): # по ключам словаря с данными DFs
    # заменим значения 'менее 0.1' на 0,05
    dict_df_locations[name].loc[(dict_df_locations[name].Visibility == 'менее 0.1'), "Visibility"] = 0.05
    # заменим значения 'менее 0.05' на 0,01
    dict_df_locations[name].loc[(dict_df_locations[name].Visibility == 'менее 0.05'), "Visibility"] = 0.025
    # изменим тип данный в столбце "Visibility" на 'float'
```

```
dict_df_locations[name].Visibility = dict_df_locations[name].Visibility.astype('float')
# dict_df_locations[name].info()
```

1.1.3.5. Преобразование параметра Prcpttn в dtype = float

замена строк на условные числовые значения

```
In [53]: # Преобразуем значения в столбце Visibility
for name in dict_df_locations.keys(): # по ключам словаря с данными DFs
    # заменим значения 'Осадков нет' на 0,0
    dict_df_locations[name].loc[(dict_df_locations[name].Prcpttn == 'Осадков нет'), "Prcpttn"] = 0.0
    # заменим значения 'Следы осадков' на 0,5
    dict_df_locations[name].loc[(dict_df_locations[name].Prcpttn == 'Следы осадков'), "Prcpttn"] = 0.5
    # изменим тип данный в столбце "Prcpttn" на 'float'
    dict_df_locations[name].Prcpttn = dict_df_locations[name].Prcpttn.astype('float')
# dict_df_locations[name].info()
```

1.1.3.6. Преобразование параметра Snow_height в dtype = float,¶

замена строк на условные числовые значения

```
In [54]: # Преобразуем значения в столбце Snow_height
for name in dict_df_locations.keys(): # по ключам словаря с данными DFs
    # заменим значения 'Измерение невозможно или неточно.' на 0,01
    dict_df_locations[name].loc[(dict_df_locations[name].Snow_height == 'Измерение невозможно или неточно.'),
                                  "Snow_height"] = 0.01 # Условная величина
    # заменим значения 'Снежный покров не постоянный.' на 0.1
    dict_df_locations[name].loc[(dict_df_locations[name].Snow_height == 'Снежный покров не постоянный.'),
                                  "Snow_height"] = 0.1 # Условная величина
    # заменим значения 'Менее 0.5' на 0.25
    dict_df_locations[name].loc[(dict_df_locations[name].Snow_height == 'Менее 0.5'),
                                  "Snow_height"] = 0.25
    # изменим тип данный в столбце "Snow_height" на 'float'
    dict_df_locations[name].Snow_height = dict_df_locations[name].Snow_height.astype('float')
# dict_df_locations[name].info()
```

ВАЖНО

В параметре высоты снегового покрова было использовано 2 условные величины:

- 'Измерение невозможно или неточно.' = 0,09
- 'Снежный покров не постоянный.' = 0,1

Это необходимо учитывать при выявлении аномалий, ошибок и пропусков, а также при трактовке результатов.

1.1.4. Создание словаря из DFs архивов метеонаблюдений в разрезе параметров (другое измерение архива метеонаблюдений в разрезе метеостанций)

Изначально архив метеонаблюдений был создан в разрезе метеостанций. Для удобства дальнейшей работы создадим архив метеонаблюдений в разрезе параметров.

```
In [55]: # DFs для учёта параметров

# Массив с датами и временем за всю историю наблюдений
# минимальная и максимальная дата наблюдений
start_date = df_stations.beg_date.min()
finish_date = df_stations.end_date.max() + pd.Timedelta(1, 'day') # Добавим еще один день к концу периода
# Искомый массив
arr_dates = np.arange(start=start_date, stop=finish_date, step = np.timedelta64(3, 'h'), dtype='datetime64[ns]')

# Список параметров (Создаётся из названий столбцов любого архивного датафрейма - названия столбцов везде одноковые)
list_parameters = dict_df_locations['df_Klin'].columns.tolist()

# Словарь для df_ из значений параметров (на основе dict_df_locations)
dict_df_parameters={}

# Заполним словарь пустыми DF и зададим индекс равный списку дат arr_dates по убыванию
for param in list_parameters:
    dict_df_parameters[f"df_{param}"] = pd.DataFrame(index=arr_dates).sort_index(ascending = False)

for name, data in dict_df_locations.items(): # по элементам словаря с данными DFs
    for param in dict_df_parameters.keys(): # по ключам словаря со значениями параметров
        col_name = f'{param[3:]#{name[3:]}}' # зададим названия столбца (параметр#станция) в отдельную переменную
        # Объединим текущие данные по параметру со значениями по этому параметру с каждой следующей метеостанцией
        dict_df_parameters[param] = dict_df_parameters[param].merge(data[param[3:]],
                                                                    how='left',
                                                                    left_index=True,
                                                                    right_index=True,
                                                                    )
# переименуем вновь присоединённый столбец по значению col_name
```

```
dict_df_parameters[param] = dict_df_parameters[param].rename(columns={param[3:]:col_name})
```

```
In [56]: for data in dict_df_parameters.values():
    data.sample(3)
```

```
Out[56]:
```

	T#V_Volochek	T#Staritsa	T#Kashyn	T#Tver	T#Klin	T#Dmitrov	T#Volokolamsk	T#Mozhaisk	T#N_Jerusalem	T#Nemchinovka	T#Naro_Fomin
2017-09-13 15:00:00	23.9	23.9	24.0	25.5	23.7	21.7	23.6	22.5	21.9	22.6	22
2010-01-11 09:00:00	-23.1	-22.3	-21.5	-20.8	-20.6	-19.9	-25.3	-25.8	-22.2	-19.6	-25
2009-05-01 06:00:00	NaN	1.2	NaN	1.7	0.7	0.6	2.2	2.8	2.5	NaN	2

```
Out[56]:
```

	P_station#V_Volochek	P_station#Staritsa	P_station#Kashyn	P_station#Tver	P_station#Klin	P_station#Dmitrov	P_station#Volokolamsk	P_station#Mozhaisk	P_station#N_Jerusalem	P_station#Nemchinovka	P_station#Naro_Fomin
2005-09-19 03:00:00	NaN	752.2	755.7	755.2	754.3	753.4	751.6	751.6	751.6	751.6	751.6
2010-03-25 09:00:00	753.4	752.4	755.0	755.3	753.3	751.0	750.8	750.8	750.8	750.8	750.8
2008-04-08 06:00:00	NaN	740.1	NaN	743.8	743.0	742.6	739.6	739.6	739.6	739.6	739.6

Out[56]:	P_sea#V_Volochek	P_sea#Staritsa	P_sea#Kashyn	P_sea#Tver	P_sea#Klin	P_sea#Dmitrov	P_sea#Volokolamsk	P_sea#Mozhaisk	P_sea#N_Jerusalem
2009-09-17 15:00:00	755.6	756.3	755.8	756.4	756.5	756.4	756.8	758.2	751.
2020-06-17 18:00:00	761.0	761.2	761.8	761.0	761.1	761.1	761.3	762.1	761.
2006-01-18 15:00:00	NaN	774.3	775.3	774.4	774.8	774.8	774.6	773.9	773.
Out[56]:	P_drift#V_Volochek	P_drift#Staritsa	P_drift#Kashyn	P_drift#Tver	P_drift#Klin	P_drift#Dmitrov	P_drift#Volokolamsk	P_drift#Mozhaisk	P_drift#Jerusalem
2014-09-08 18:00:00	-0.3	-0.5	-0.2	-0.6	-0.3	-0.3	-0.4	-0.5	
2014-11-17 15:00:00	0.1	-0.3	-0.1	-0.3	-0.3	-0.3	-0.4	-0.7	
2007-04-24 15:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Out[56]:	Humid#V_Volochek	Humid#Staritsa	Humid#Kashyn	Humid#Tver	Humid#Klin	Humid#Dmitrov	Humid#Volokolamsk	Humid#Mozhaisk	Humid#Jerusalem
2014-07-19 06:00:00	97.0	89.0	88.0	87.0	87.0	74.0	90.0	94.0	
2014-04-15 15:00:00	63.0	63.0	56.0	54.0	49.0	54.0	66.0	61.0	
2007-01-21 15:00:00	93.0	95.0	84.0	95.0	92.0	93.0	95.0	92.0	

Out[56]:	Wind_dir#V_Volochek	Wind_dir#Staritsa	Wind_dir#Kashyn	Wind_dir#Tver	Wind_dir#Klin	Wind_dir#Dmitrov	Wind_dir#Volokolamsk	Wind_dir#
2020-01-17 21:00:00	WSW	W	WSW	WSW	W	WSW	WSW	W
2022-04-09 12:00:00	S	S	SSW	SSW	S	SW		S
2015-04-24 06:00:00	WNW	WNW	NNW	W	W	W	WNW	
Out[56]:	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#	
2011-02-22 15:00:00	135.0	135.0	157.5	135.0	157.5	135.0		
2009-04-05 09:00:00	225.0	225.0	315.0	225.0	157.5	180.0		
2011-10-28 18:00:00	180.0	180.0	247.5	180.0	270.0	270.0		
Out[56]:	Wind_dir6k#V_Volochek	Wind_dir6k#Staritsa	Wind_dir6k#Kashyn	Wind_dir6k#Tver	Wind_dir6k#Klin	Wind_dir6k#Dmitrov	Wind_dir6k#Volokolamsk	Wind_dir6k#
2013-11-22 15:00:00	2625.0	2625.0	3000.0	2625.0	3000.0	2250.0	3000.0	3000.0
2015-07-22 12:00:00	3750.0	3750.0	4500.0	3375.0	4125.0	Nan		
2021-11-08 03:00:00	3000.0	3375.0	3375.0	3375.0	3000.0	3375.0	3375.0	3375.0

Out[56]:	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2019-07-05 00:00:00	3.0	1.0	3.0	2.0	1.0	2.0	
2008-02-14 00:00:00	Nan	6.0	Nan	4.0	3.0	5.0	
2021-05-04 12:00:00	1.0	4.0	3.0	2.0	3.0	5.0	

Out[56]:	Cloudness#V_Volochev	Cloudness#Staritsa	Cloudness#Kashyn	Cloudness#Tver	Cloudness#Klin	Cloudness#Dmitrov	Cloudness#Volokolamsk	Cloudness#Moscow
2015-10-18 18:00:00	от 90 менее 100%	100%.	от 90 менее 100%	от 90 менее 100%	100%.	Nan	NaN	NaN
2020-03-12 12:00:00	от 90 менее 100%	от 90 менее 100%	от 90 менее 100%	от 90 менее 100%	от 90 менее 100%	100%.	от 90 менее 100%	от 90 менее 100%
2006-09-16 09:00:00	20–30%.	40%.	20–30%.	20–30%.	60%.	100%.	50%.	50%.
Out[56]:	Wthr_curr#V_Volochev	Wthr_curr#Staritsa	Wthr_curr#Kashyn	Wthr_curr#Tver	Wthr_curr#Klin	Wthr_curr#Dmitrov	Wthr_curr#Volokolamsk	Wthr_curr#Moscow
2017-08-20 03:00:00	n/a	n/a	Дымка.	n/a	n/a	Облака в целом рассеиваются или становятся мен...	n/a	n/a
2018-03-08 03:00:00	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
2010-11-29 21:00:00	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Out[56]:	Prcptn_like#V_Volochev	Prcptn_like#Staritsa	Prcptn_like#Kashyn	Prcptn_like#Tver	Prcptn_like#Klin	Prcptn_like#Dmitrov	Prcptn_like#Volokolamsk	Prcptn_like#Moscow
2014-08-01 00:00:00	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
2017-10-26 06:00:00	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
2011-01-11 09:00:00	n/a	Состояние неба в общем не изменилось.	Состояние неба в общем не изменилось.	Снег неливневый.	Снег неливневый.	Снег непрерывный слабый в срок наблюдения.	Облака в целом рассеиваются или становятся мен...	Облака в целом рассеиваются или становятся мен...

Out[56]:	Prcptn_depo#V_Volochev	Prcptn_depo#Staritsa	Prcptn_depo#Kashyn	Prcptn_depo#Tver	Prcptn_depo#Klin	Prcptn_depo#Dmitrov	Prcptn_depo#Moscow	
2016-05-16 00:00:00	n/a	n/a	n/a	n/a	n/a	n/a	NaN	
2009-10-02 18:00:00	NaN	n/a	NaN	n/a	n/a	n/a	n/a	
2019-04-16 18:00:00	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
Out[56]:	Depo_diam_mm#V_Volochev	Depo_diam_mm#Staritsa	Depo_diam_mm#Kashyn	Depo_diam_mm#Tver	Depo_diam_mm#Klin	Depo_diam_mm#Dmitrov	Depo_diam_mm#Moscow	
2015-03-27 18:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2011-11-01 12:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2011-06-29 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Out[56]:	Wthr_3h#V_Volochev	Wthr_3h#Staritsa	Wthr_3h#Kashyn	Wthr_3h#Tver	Wthr_3h#Klin	Wthr_3h#Dmitrov	Wthr_3h#Volokolamsk	Wthr_3h#Moscow
2009-10-29 21:00:00	Ливень (ливни).	Ливень (ливни).	Ливень (ливни).	Ливень (ливни).	Дождь.	Дождь.	Ливень (ливни).	Ливень (ливни).
2011-05-23 18:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-01-09 06:00:00	Снег и/или другие виды твердых осадков	NaN	NaN	Снег и/или другие виды твердых осадков	NaN	Снег и/или другие виды твердых осадков	NaN	NaN

Out[56]:	Wthr_3h2#V_Volochev	Wthr_3h2#Staritsa	Wthr_3h2#Kashyn	Wthr_3h2#Tver	Wthr_3h2#Klin	Wthr_3h2#Dmitrov	Wthr_3h2#Volokolamsk	Wthr_3h2#Leningrad
2008-09-04 18:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2013-10-10 18:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-02-25 12:00:00	Облака покрывали более половины неба в течение...	NaN	Облака покрывали более половины неба в течение...	Облака покрывали более половины неба в течение...	NaN	NaN	NaN	NaN

Out[56]:

		Cl_Cumls#V_Volocheok	Cl_Cumls#Staritsa	Cl_Cumls#Kashyn	Cl_Cumls#Tver	Cl_Cumls#Klin	Cl_Cumls#Dmitrov	Cl_Cumls#Volokolamsk	Cl_Cumls#I
2012-03-16 15:00:00	Слоисто-кучевых, слоистых, кучевых или кучево-...	NaN	Слоисто-кучевых, слоистых, кучевых или кучево-...	Слоисто-кучевых, слоистых, кучевых или кучево-...	Слоисто-кучевых, слоистых, кучевых или кучево-...				
2020-10-19 18:00:00	Кучево-дождевые волокнистые (часто с наковальн...	Кучево-дождевые волокнистые (часто с наковальн...	Слоисто-кучевые, образовавшиеся не из кучевых.	Кучево-дождевые лысые с кучевыми, слоисто-куче...	Кучево-дождевые лысые с кучевыми, слоисто-куче...	Слоисто-кучевые, образовавшиеся не из кучевых.	NaN		
2012-05-17 12:00:00	Nan	Nan	Nan	Nan	Nan	Nan	Kучевые плоские или кучевые разорванные, или т...	Kучевые или мон... вместе с	

Out[56]:

	Cl_viewd#V_Volocheok	Cl_viewd#Staritsa	Cl_viewd#Kashyn	Cl_viewd#Tver	Cl_viewd#Klin	Cl_viewd#Dmitrov	Cl_viewd#Volokolamsk	Cl_viewd#M	
2013-08-27 21:00:00	60%.	10% или менее, но не 0		20–30%.	50%.	NaN	10% или менее, но не 0	20–30%.	90 или более
2018-01-16 18:00:00	10% или менее, но не 0	90 или более, но не 100%	90 или более, но не 100%		60%.	70 – 80%.	20–30%.	NaN	
2010-11-26 03:00:00	100%.	100%.	90 или более, но не 100%		100%.	100%.	90 или более, но не 100%	100%.	

	Cl_bottom#V_Volochev	Cl_bottom#Staritsa	Cl_bottom#Kashyn	Cl_bottom#Tver	Cl_bottom#Klin	Cl_bottom#Dmitrov	Cl_bottom#Volokolamsk	C
2016-10-22 06:00:00	600-1000	600-1000	2500 или более, или облаков нет.	1000-1500	600-1000	300-600	600-1000	
2005-08-02 15:00:00	Nan	600-1000	600-1000	600-1000	600-1000	600-1000	600-1000	600-1000
2018-07-01 21:00:00	600-1000	300-600	600-1000	1000-1500	600-1000	300-600		Nan

	Cl_cumls_hi#V_Volochev	Cl_cumls_hi#Staritsa	Cl_cumls_hi#Kashyn	Cl_cumls_hi#Tver	Cl_cumls_hi#Klin	Cl_cumls_hi#Dmitrov	Cl_cumls_hi#Vo	
2011-04-22 18:00:00	Высококучевых, высокослоистых или слоисто-дожд...	Высококучевые, просвечающие или плотные в дв...	Высококучевых, высокослоистых или слоисто-дожд...	Высококучевые просвечающие, расположенные на...	Высококучевых, высокослоистых или слоисто-дожд...	Высококучевых, высокослоистых или слоисто-дожд...	Высококучевых, высокослоистых или слоисто-дожд...	Высо просве располож
2006-07-17 18:00:00	Nan	Nan	Nan	Nan	Nan	Высококучевые просвечающие, расположенные на...	Nan	Высо просве располож
2017-12-06 12:00:00	Nan	Высокослоистые непросвечающие или слоисто-до...	Высококучевые просвечающие, расположенные на...	Высокослоистые непросвечающие или слоисто-до...	Высокослоистые непросвечающие или слоисто-до...	Высокослоистые непросвечающие или слоисто-до...		

Out[56]:

	Cl_cirrus#V_Volochev	Cl_cirrus#Staritsa	Cl_cirrus#Kashyn	Cl_cirrus#Tver	Cl_cirrus#Klin	Cl_cirrus#Dmitrov	Cl_cirrus#Volokolamsk	Cl_cirri
--	----------------------	--------------------	------------------	----------------	----------------	-------------------	-----------------------	----------

2006-02-19 12:00:00	NaN	Перистые нитевидные, иногда когтевидные, не ра...	NaN	NaN	Перисто-слоистые, не распространяющиеся по неб...	Перистые нитевидные, иногда когтевидные, не ра...	Перистые (часто в виде полос) и перисто-слоист...	Периодиче...
2017-10-06 06:00:00	Перистых, перисто-кучевых или перисто-слоистых...	Перистых, перисто-кучевых или перисто-слоистых...	NaN	NaN	Перистых, перисто-кучевых или перисто-слоистых...	NaN	NaN	NaN
2011-08-12 00:00:00	Перистых, перисто-кучевых или перисто-слоистых...	Перистые нитевидные, иногда когтевидные, не ра...	Перистых, перисто-кучевых или перисто-слоистых...	Перистые нитевидные, иногда когтевидные, не ра...	Перистых, перисто-кучевых или перисто-слоистых...	Перистые нитевидные, иногда когтевидные, не ра...	Перистые когтевидные или нитевидные или первые...	ког...

Out[56]:

	Visibility#V_Volochev	Visibility#Staritsa	Visibility#Kashyn	Visibility#Tver	Visibility#Klin	Visibility#Dmitrov	Visibility#Volokolamsk	Visibility#M
--	-----------------------	---------------------	-------------------	-----------------	-----------------	--------------------	------------------------	--------------

2010-05-11 09:00:00	16.0	10.0	50.0	10.0	20.0	18.0	40.0	
2014-07-03 09:00:00	20.0	20.0	50.0	10.0	10.0	10.0	10.0	
2011-05-14 12:00:00	25.0	10.0	50.0	10.0	20.0	12.0	21.0	

Out[56]:

	Snow_height#V_Volochek	Snow_height#Staritsa	Snow_height#Kashyn	Snow_height#Tver	Snow_height#Klin	Snow_height#Dmitrov	Snow_height#Leningrad
--	------------------------	----------------------	--------------------	------------------	------------------	---------------------	-----------------------

2016-02-02 15:00:00	NaN						
2009-09-08 03:00:00	NaN						
2020-08-28 21:00:00	NaN						

Сохраним полученные словари датафреймов в csv файлы

In [57]:

```
# Создадим директории для этих файлов
raw_path1 = path + 'raw/locations/'
raw_path2 = path + 'raw/'

makedirs(raw_path1, exist_ok=True)
makedirs(raw_path2, exist_ok=True)

# Запишем данные в файлы
for name in dict_df_locations.keys():
    print(name + '.csv ->', end=' ')
    dict_df_locations[name].to_csv(path_or_buf=f'{raw_path1}{name}.csv')
    print('DONE!')

print()
for name in dict_df_parameters.keys():
    print(name + '.csv ->', end=' ')
    dict_df_parameters[name].to_csv(path_or_buf=f'{raw_path2}{name}.csv')
    print('DONE!')
```

df_V_Volochev.csv -> DONE!
df_Staritsa.csv -> DONE!
df_Kashyn.csv -> DONE!
df_Tver.csv -> DONE!
df_Klin.csv -> DONE!
df_Dmitrov.csv -> DONE!
df_Volokolamsk.csv -> DONE!
df_Mozhaisk.csv -> DONE!
df_N_Jerusalem.csv -> DONE!
df_Nemchinovka.csv -> DONE!
df_Naro_Fominsk.csv -> DONE!
df_Serpukhov.csv -> DONE!

df_T.csv -> DONE!
df_P_station.csv -> DONE!
df_P_sea.csv -> DONE!
df_P_drift.csv -> DONE!
df_Humid.csv -> DONE!
df_Wind_dir.csv -> DONE!
df_Wind_dir360.csv -> DONE!
df_Wind_dir6k.csv -> DONE!
df_Wind_speed.csv -> DONE!
df_Gusts.csv -> DONE!
df_Gusts_3h.csv -> DONE!
df_Cloudness.csv -> DONE!
df_Wthr_curr.csv -> DONE!
df_Prcptn_like.csv -> DONE!
df_Prcptn_depo.csv -> DONE!
df_Depo_diam_mm.csv -> DONE!
df_Wthr_3h.csv -> DONE!
df_Wthr_3h2.csv -> DONE!
df_T_min.csv -> DONE!
df_T_max.csv -> DONE!
df_C1_Cumls.csv -> DONE!
df_C1_viewd.csv -> DONE!
df_C1_bottom.csv -> DONE!
df_C1_cumls_hi.csv -> DONE!
df_C1_cirrus.csv -> DONE!
df_Visibility.csv -> DONE!
df_Dew_point.csv -> DONE!
df_Prcptn.csv -> DONE!
df_Prcptn_tdelt.csv -> DONE!
df_Soil.csv -> DONE!
df_Soil_T.csv -> DONE!

```
df_Soil_cover.csv -> DONE!
df_Snow_height.csv -> DONE!
```

1.2. Аномалии и выбросы

1.2.1. Графическое представление распределения и аномалий в "сырых" данных.

Представим аномалии и выбросы графически, это поможет нам выявить, в том числе, ошибочные значения

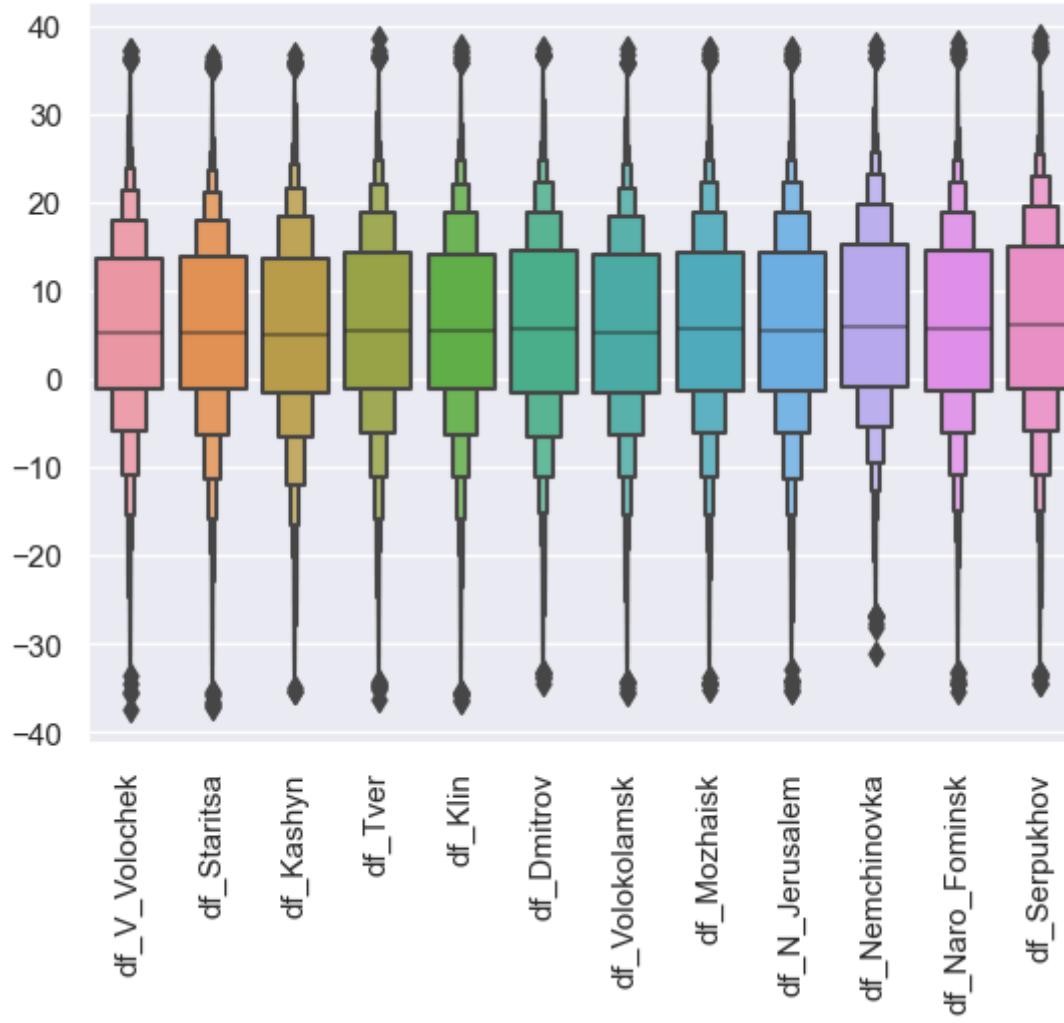
```
In [58]: # графики распределения данных по каждому числовому параметру, в разрезе метеостанций

# Определим список числовых параметров
list_parameters = ["T", "P_station", "P_sea", "P_drift", "Humid", "Wind_speed", "Wind_dir360",
                    "Gusts", "Gusts_3h", "T_min", "T_max", "Visibility", "Dew_point",
                    "Prcpttn", "Depo_diam_mm", "Prcpttn_tdelt", "Soil_T", "Snow_height"
                   ]

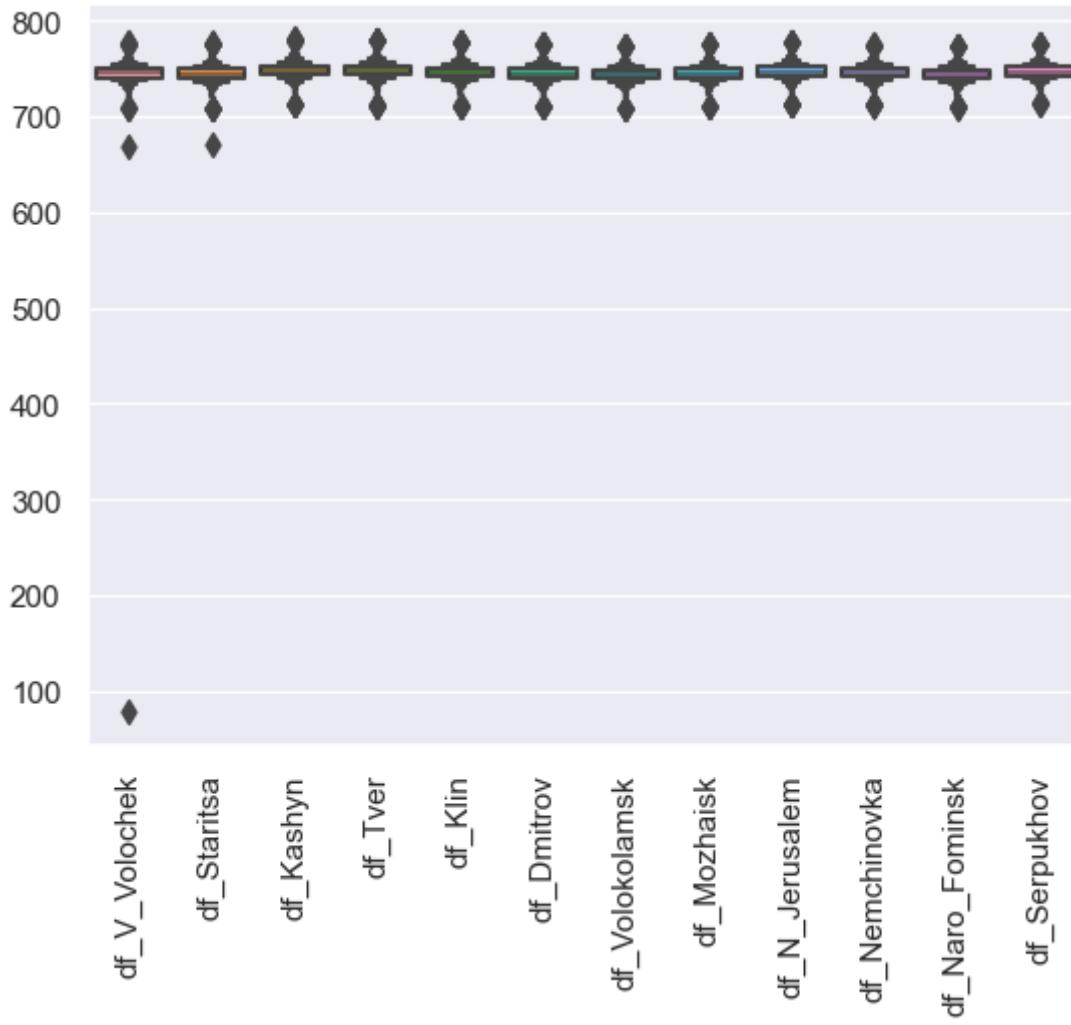
for column in list_parameters: # столбец по списку параметров
    df_tmp = pd.DataFrame(None) # создаём временный DF (изначально пустой)
    for name, data in dict_df_locations.items(): # по элементам словаря с данными DFs
        df_tmp[name] = data[column] # добавляем в df_tmp данные соответствующего столбца

    fig, ax = plt.subplots() # определяем поле графика
    # так как у нас сразу выводятся все результаты кода, чтобы избежать лишних выводов, введём переменную dummy
    dummy = sns.boxenplot(data = df_tmp, # строим график - у нас формат df - wide-data
                           ax = ax)
    dummy = ax.set_title(f'Распределение значений {column} в разрезе метеостанций')
    dummy = plt.xticks(rotation = 90) # поворачиваем подписи делений шкалы X
    plt.show() # выводим график
    del df_tmp # удаляем df_tmp, чтобы не засорять память (и самим не запутаться)
```

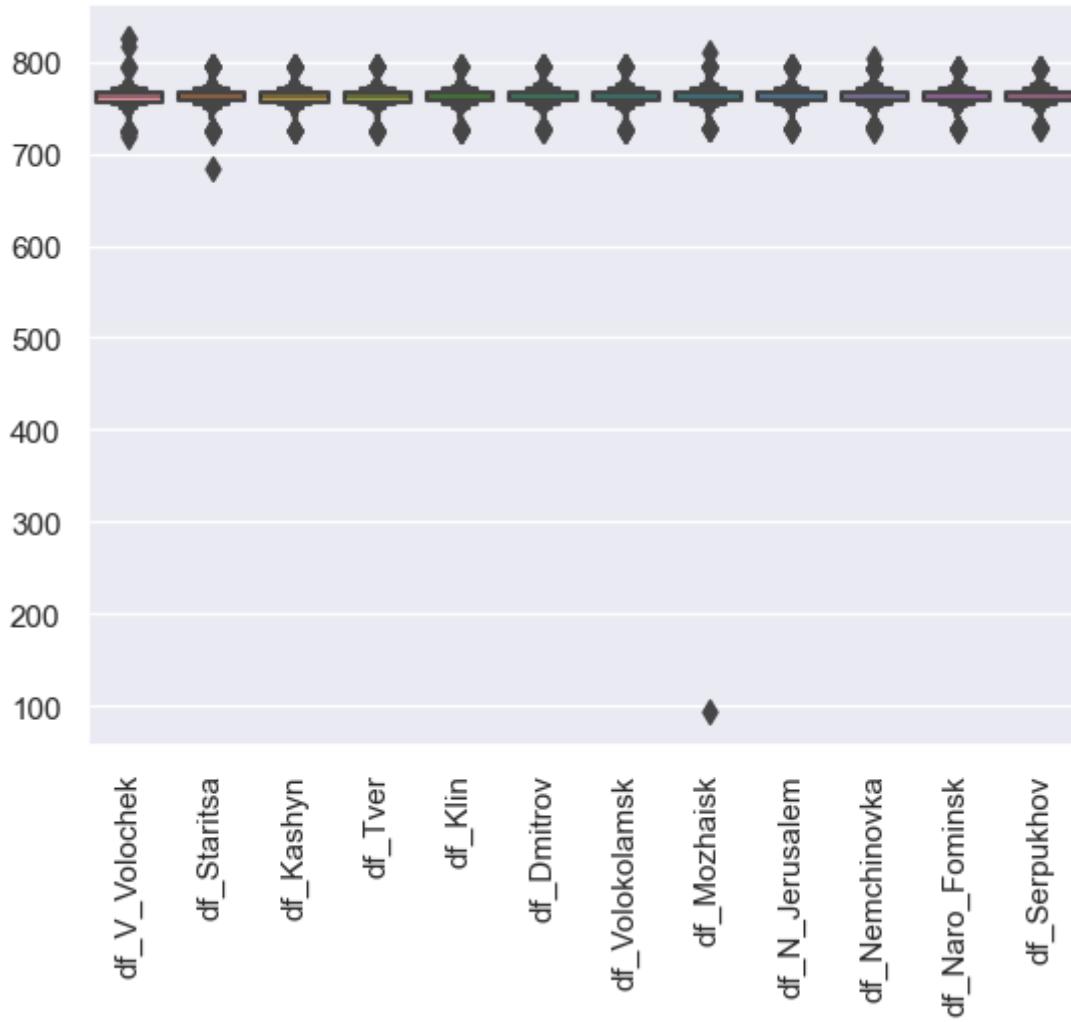
Распределение значений Т в разрезе метеостанций



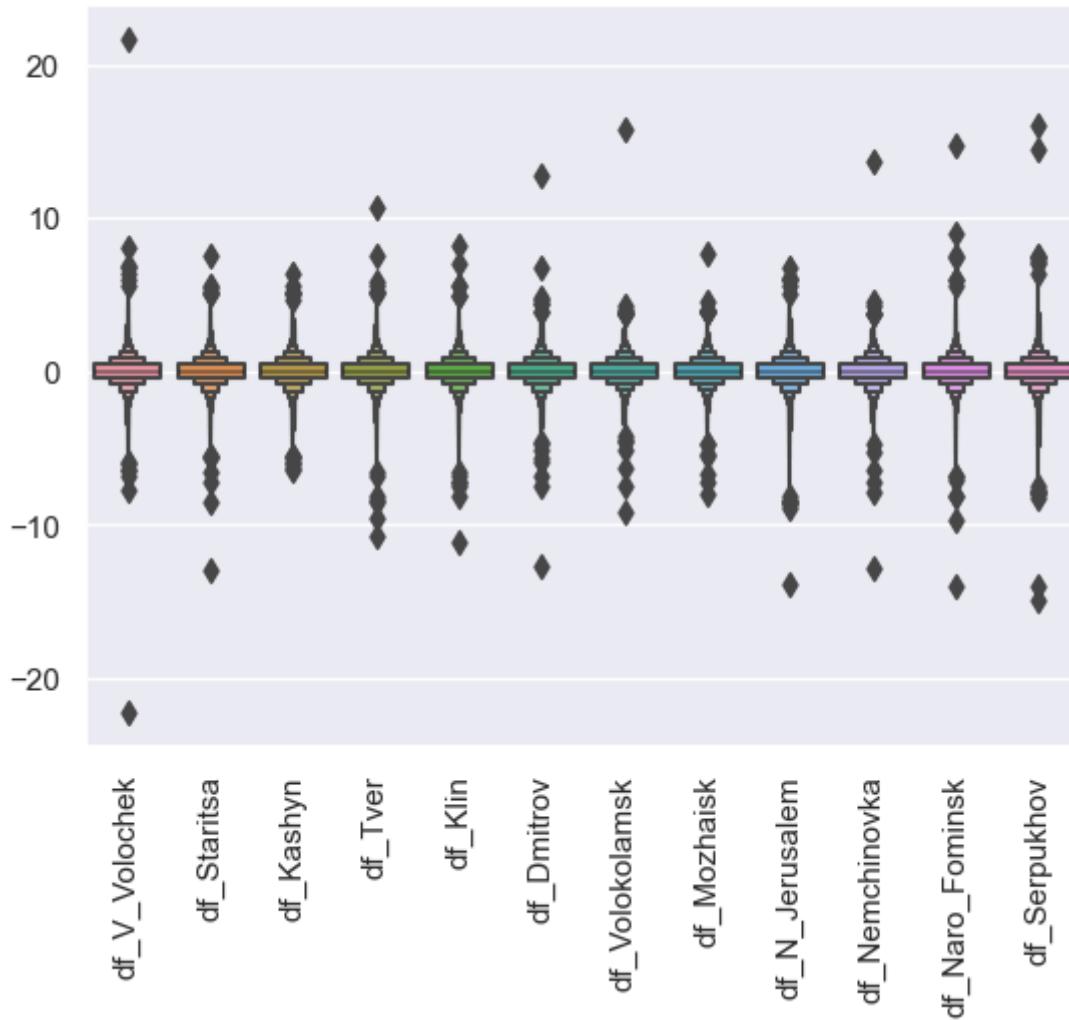
Распределение значений P_station в разрезе метеостанций



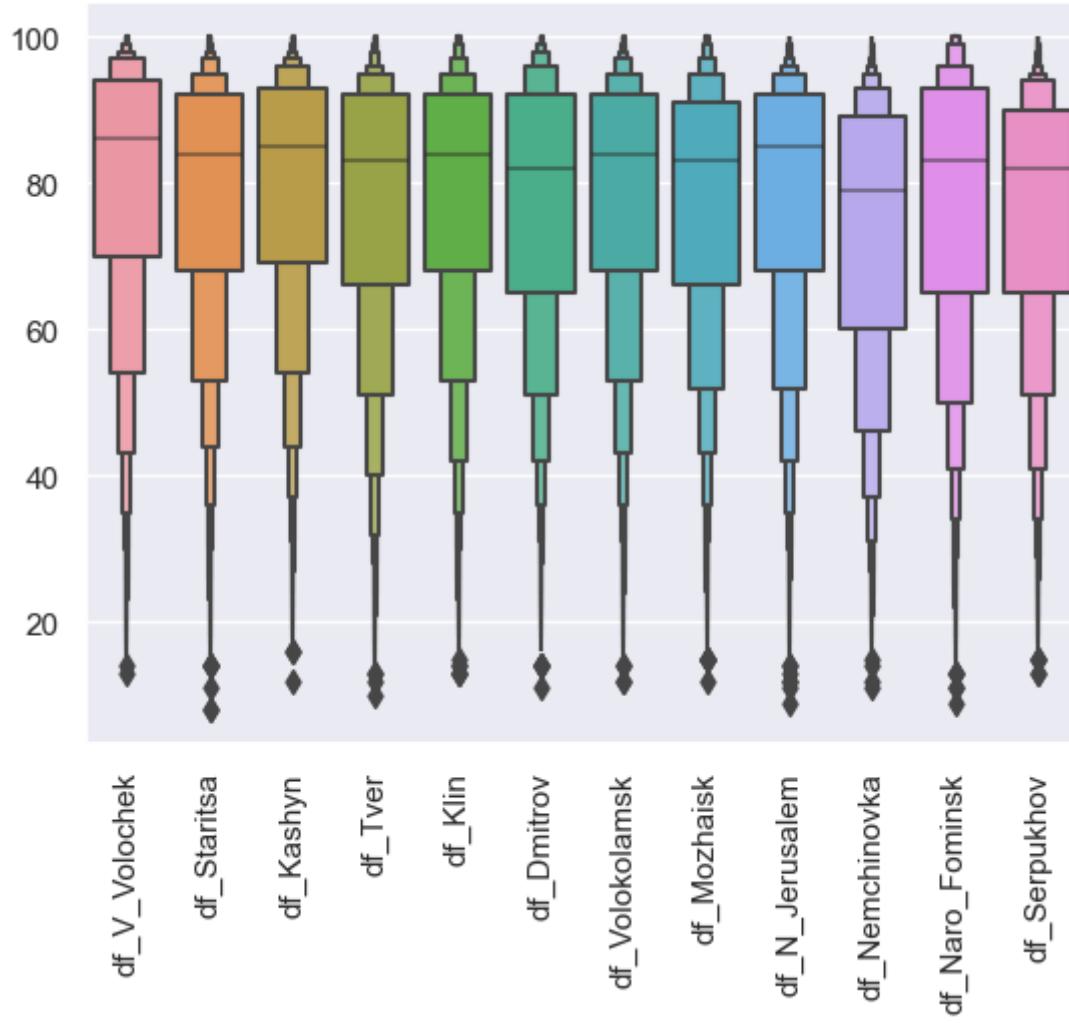
Распределение значений P_sea в разрезе метеостанций



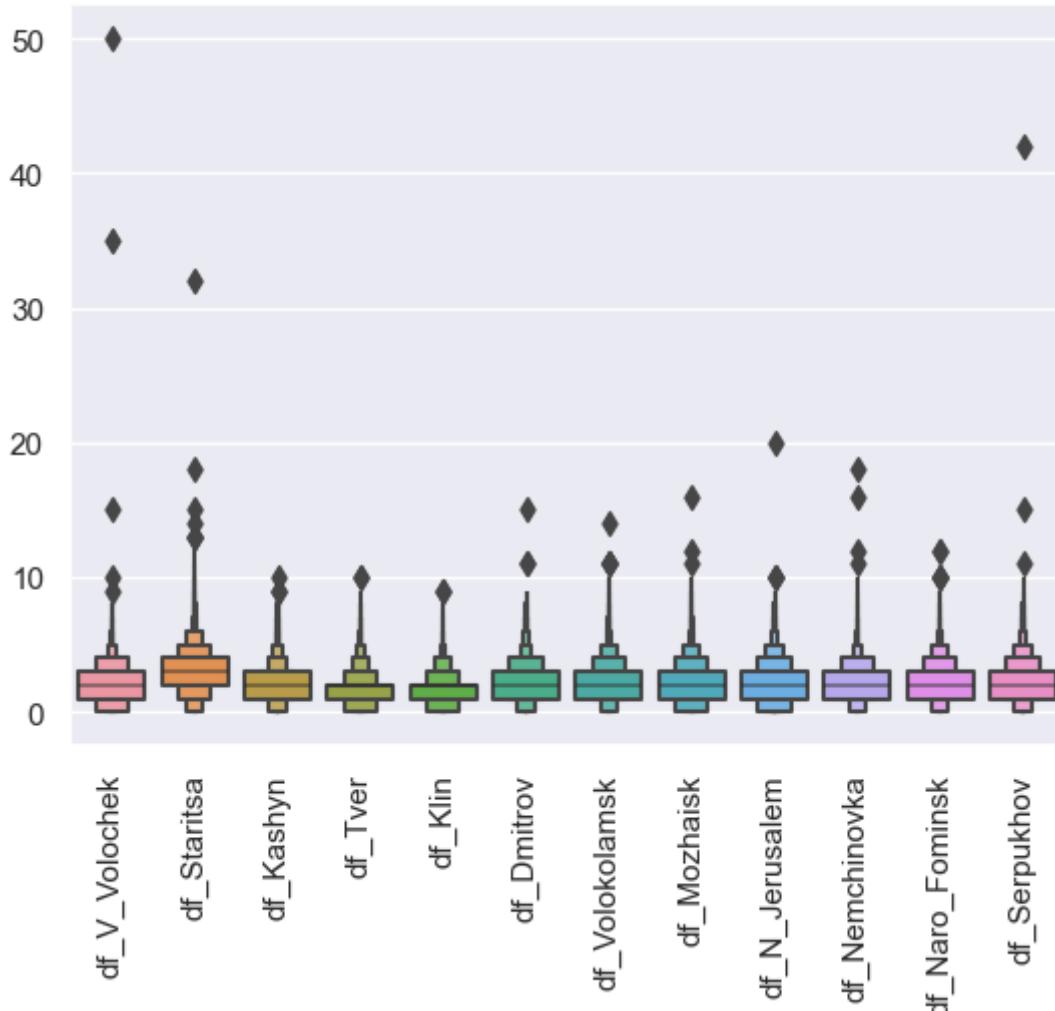
Распределение значений P_drift в разрезе метеостанций



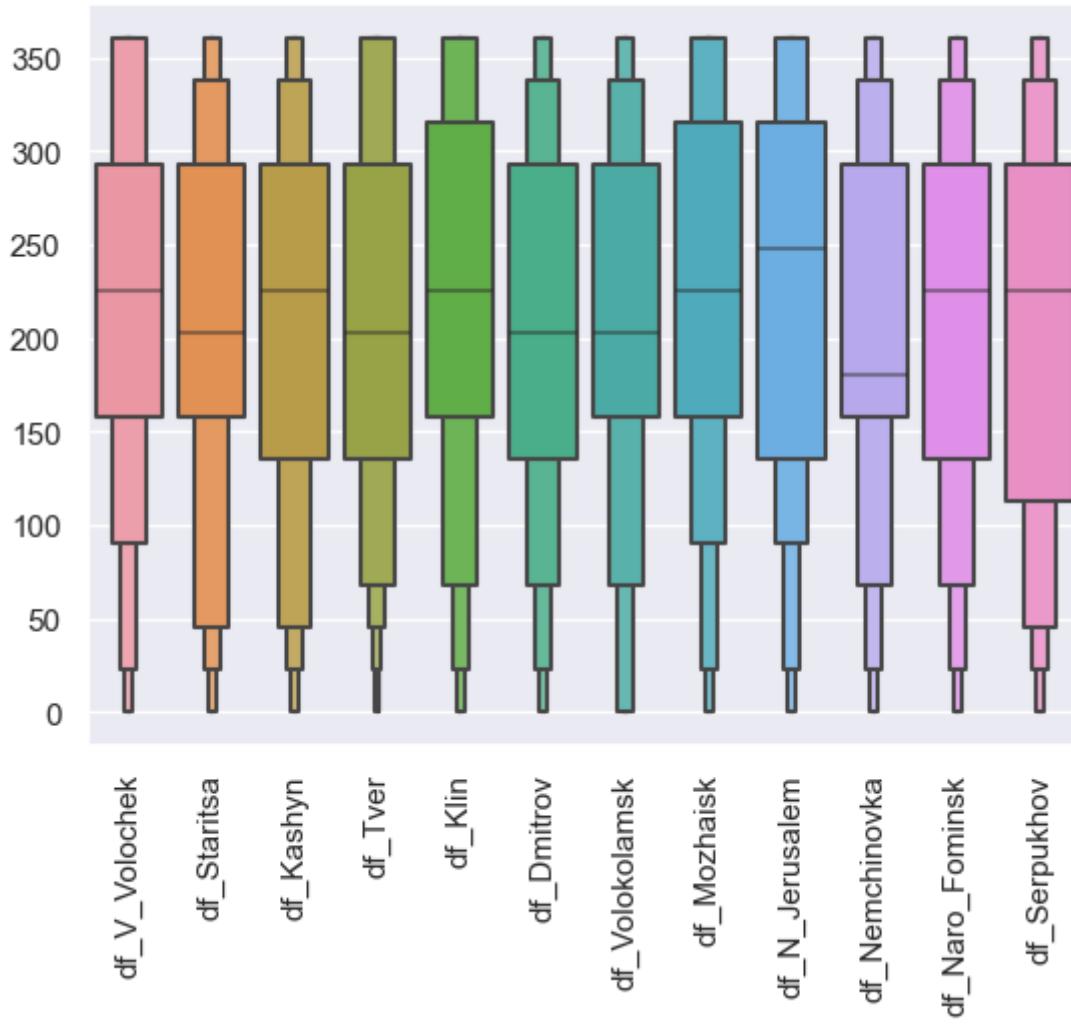
Распределение значений Humid в разрезе метеостанций



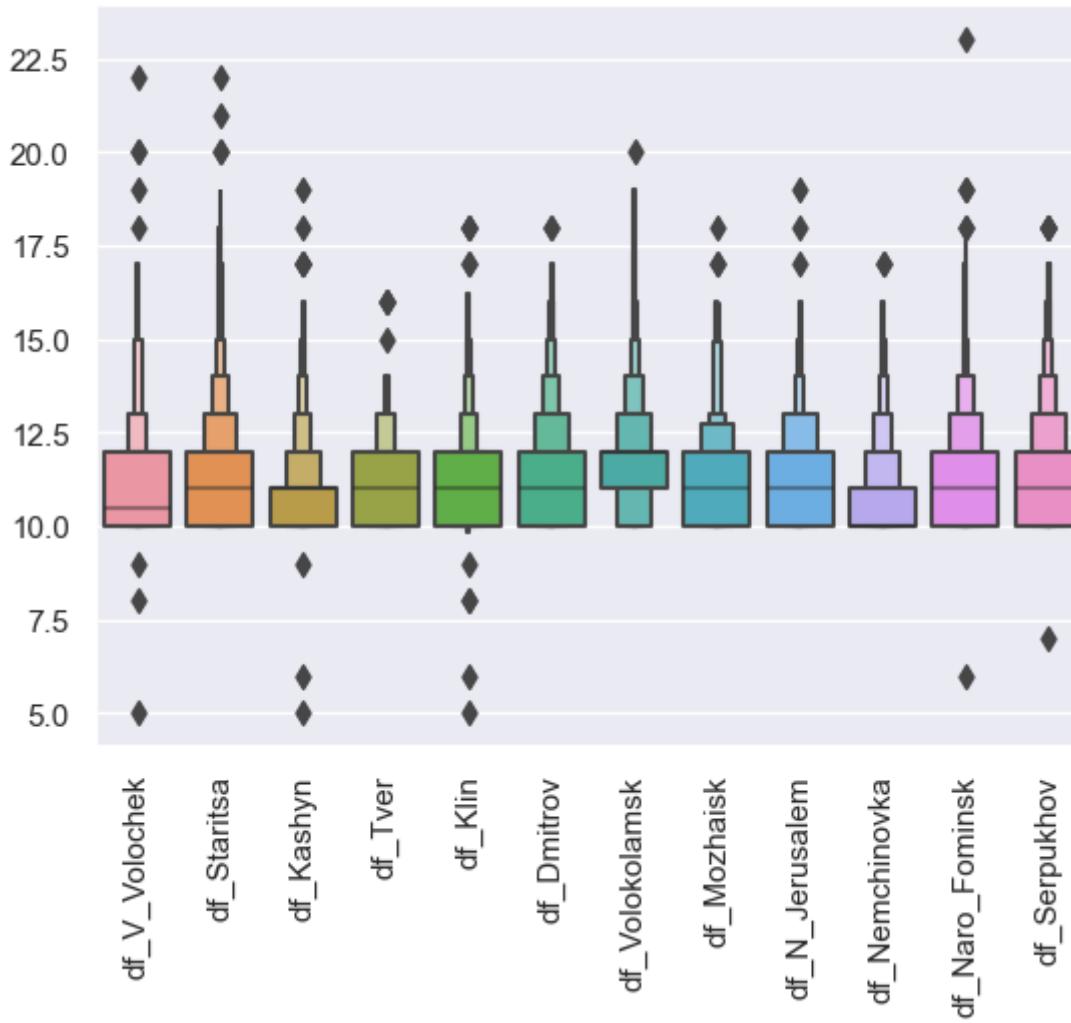
Распределение значений Wind_speed в разрезе метеостанций



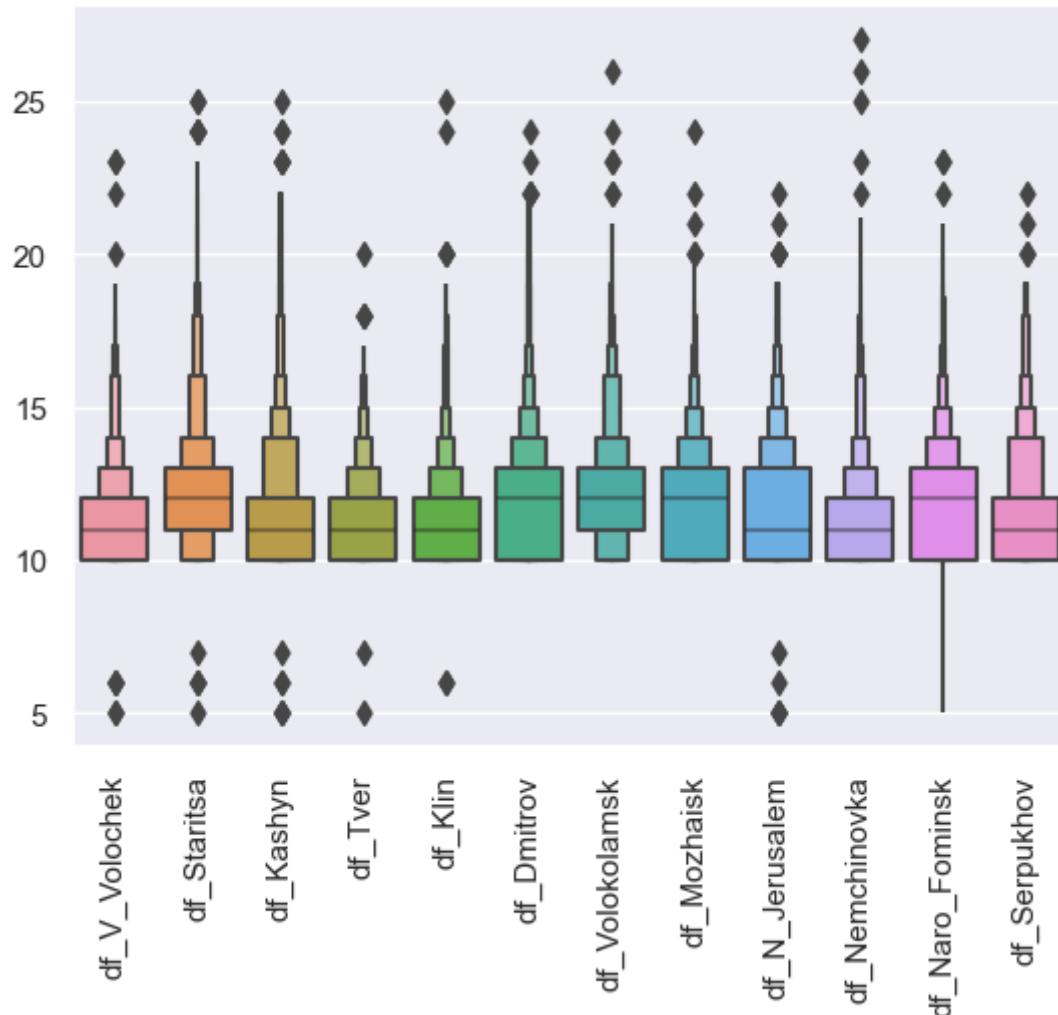
Распределение значений Wind_dir360 в разрезе метеостанций



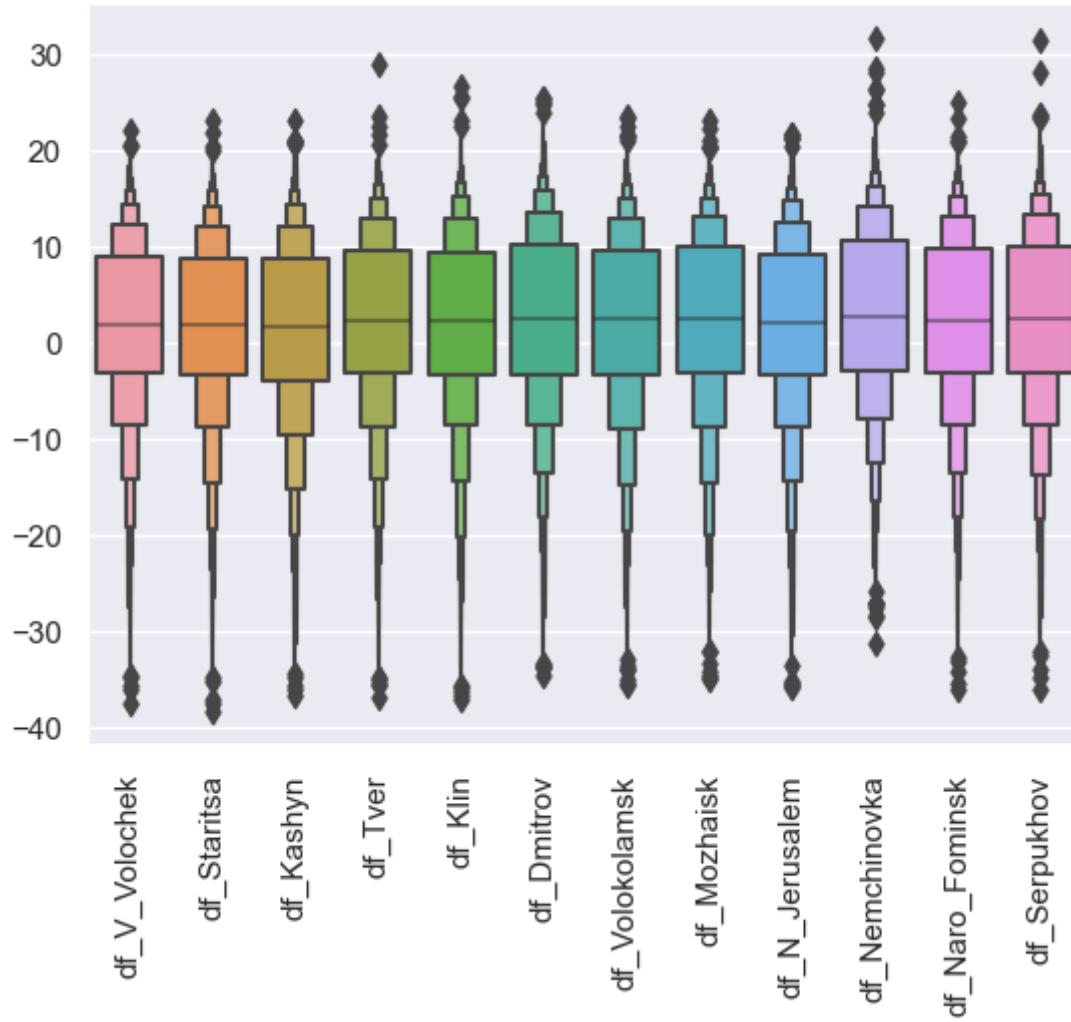
Распределение значений Gysts в разрезе метеостанций



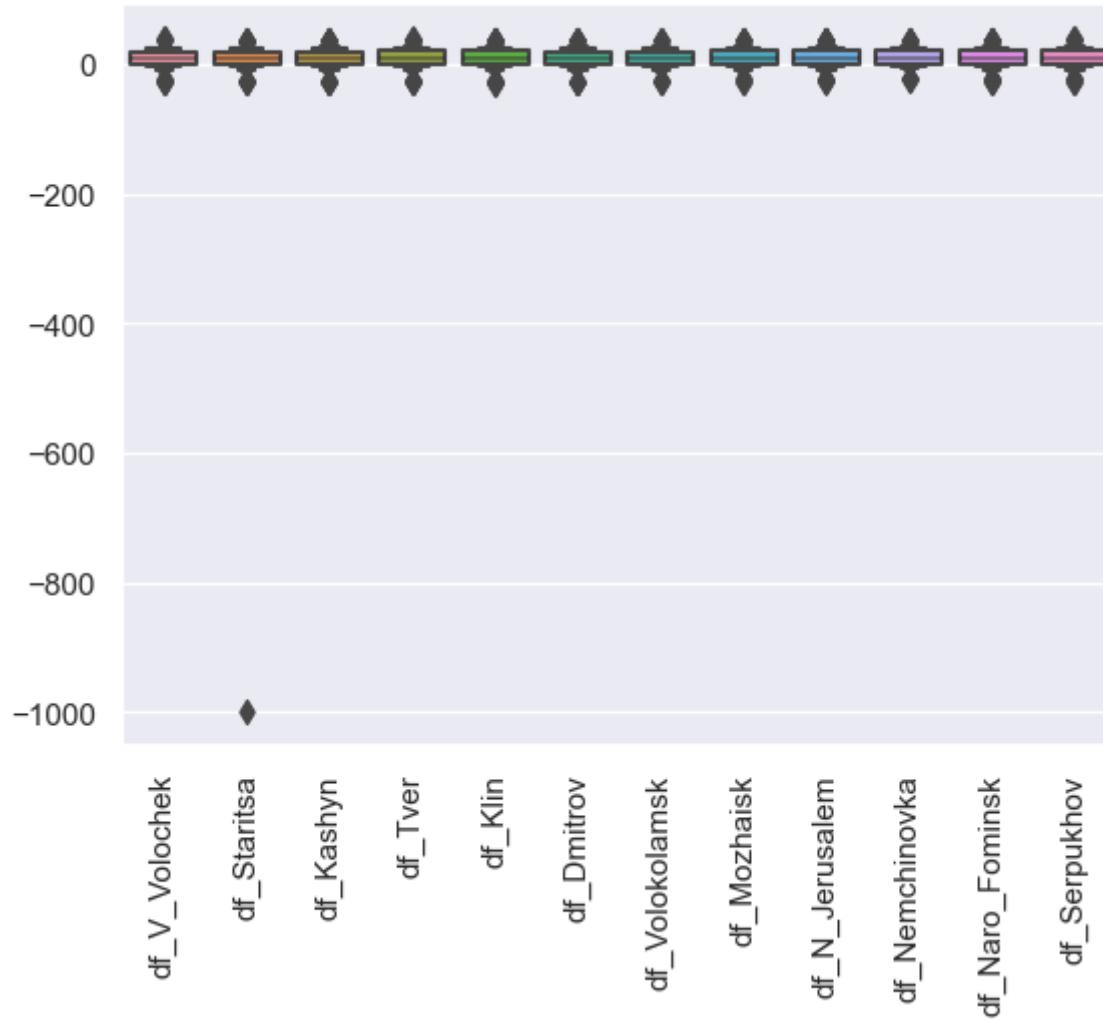
Распределение значений Gists_3h в разрезе метеостанций



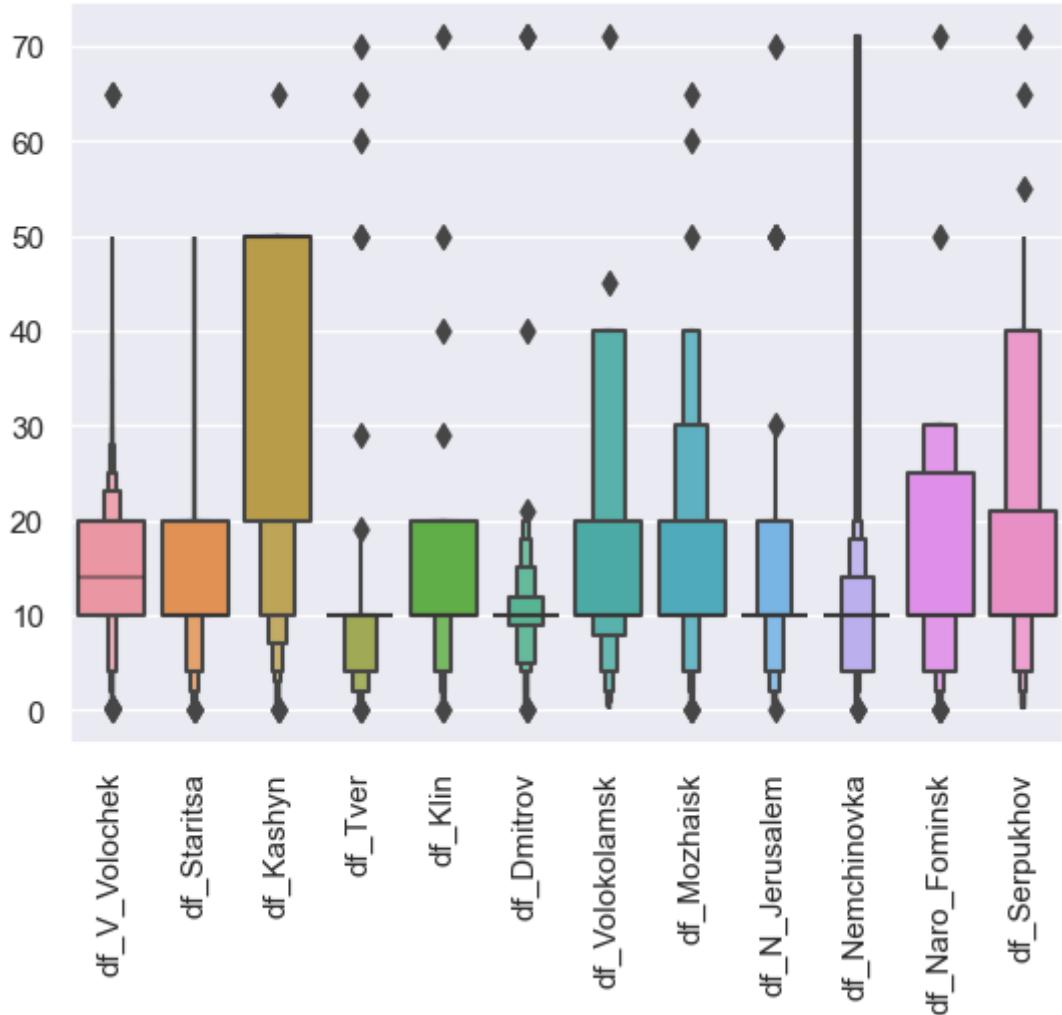
Распределение значений T_min в разрезе метеостанций



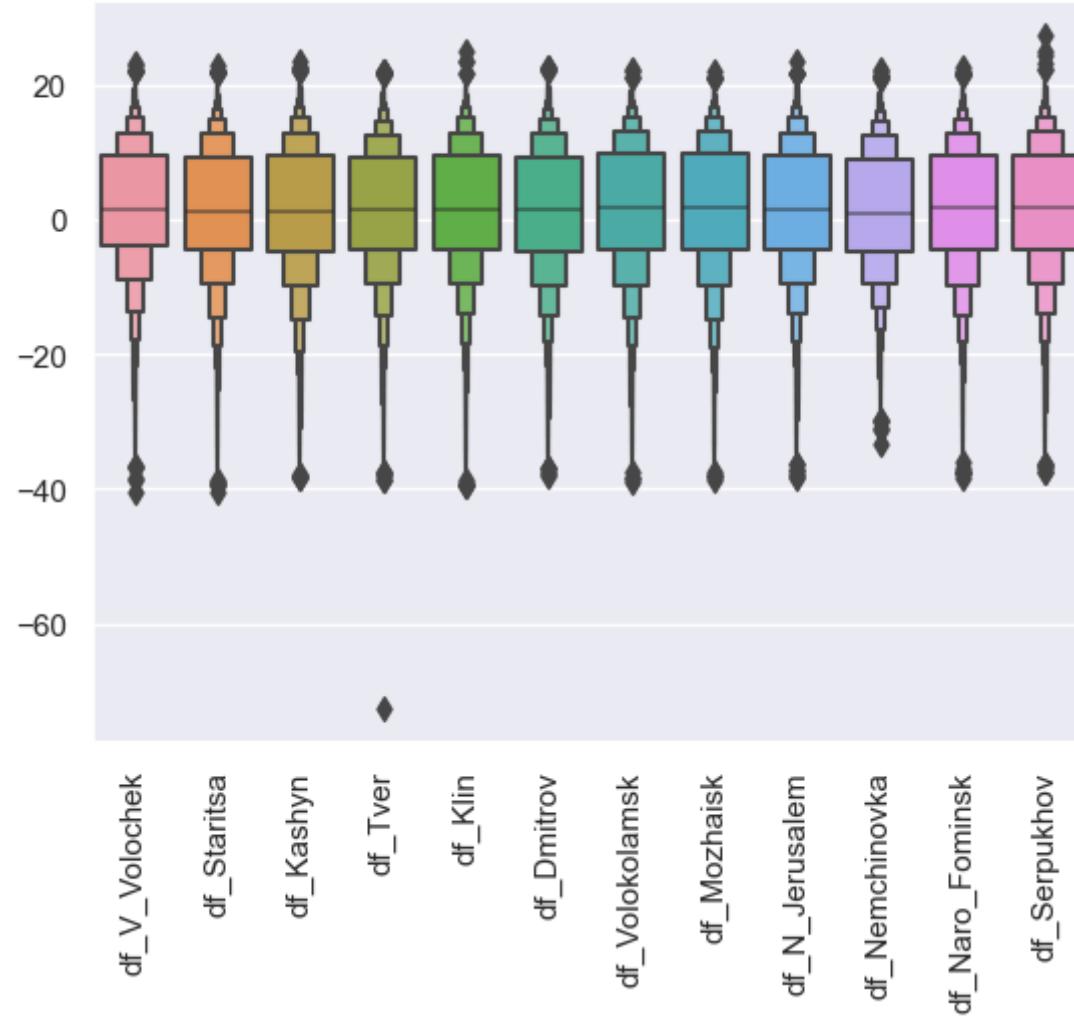
Распределение значений T_max в разрезе метеостанций



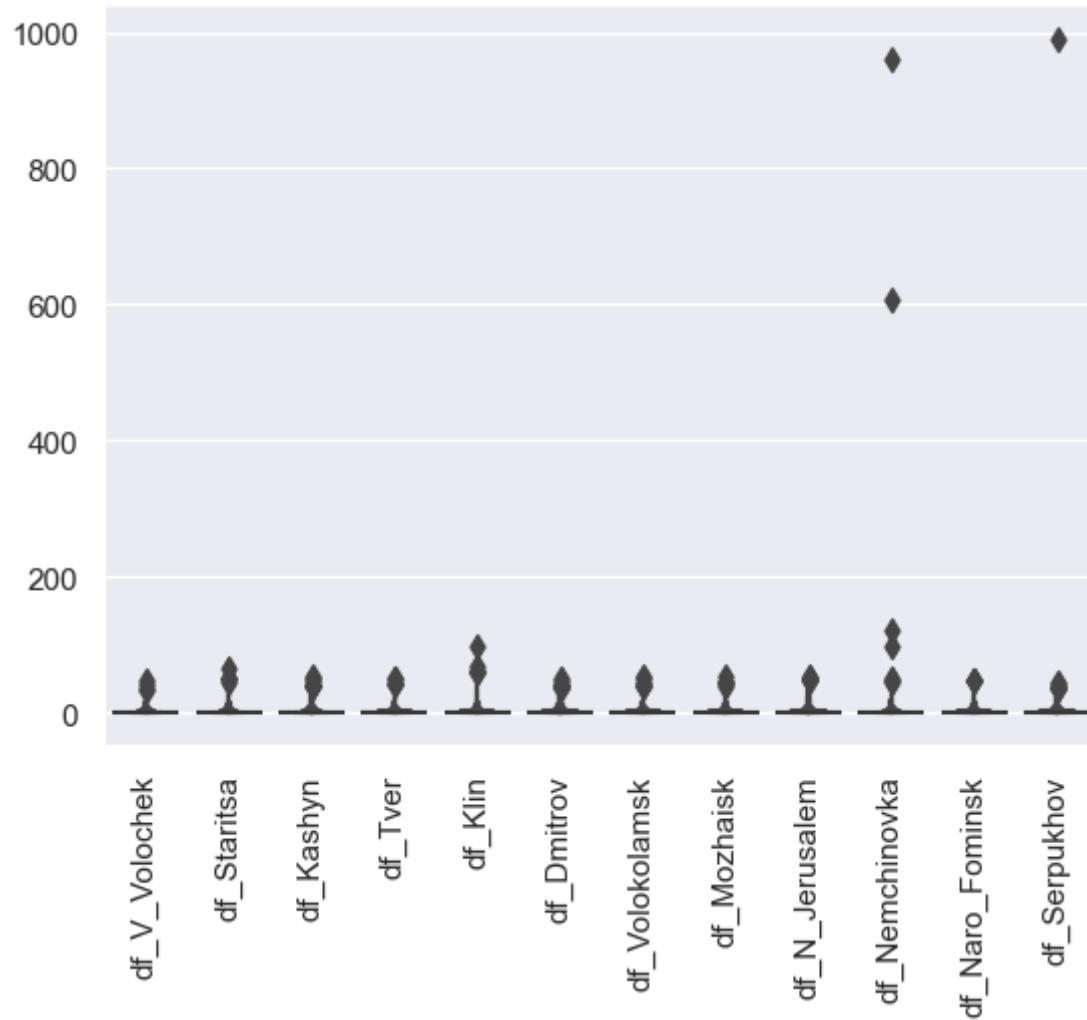
Распределение значений Visibility в разрезе метеостанций



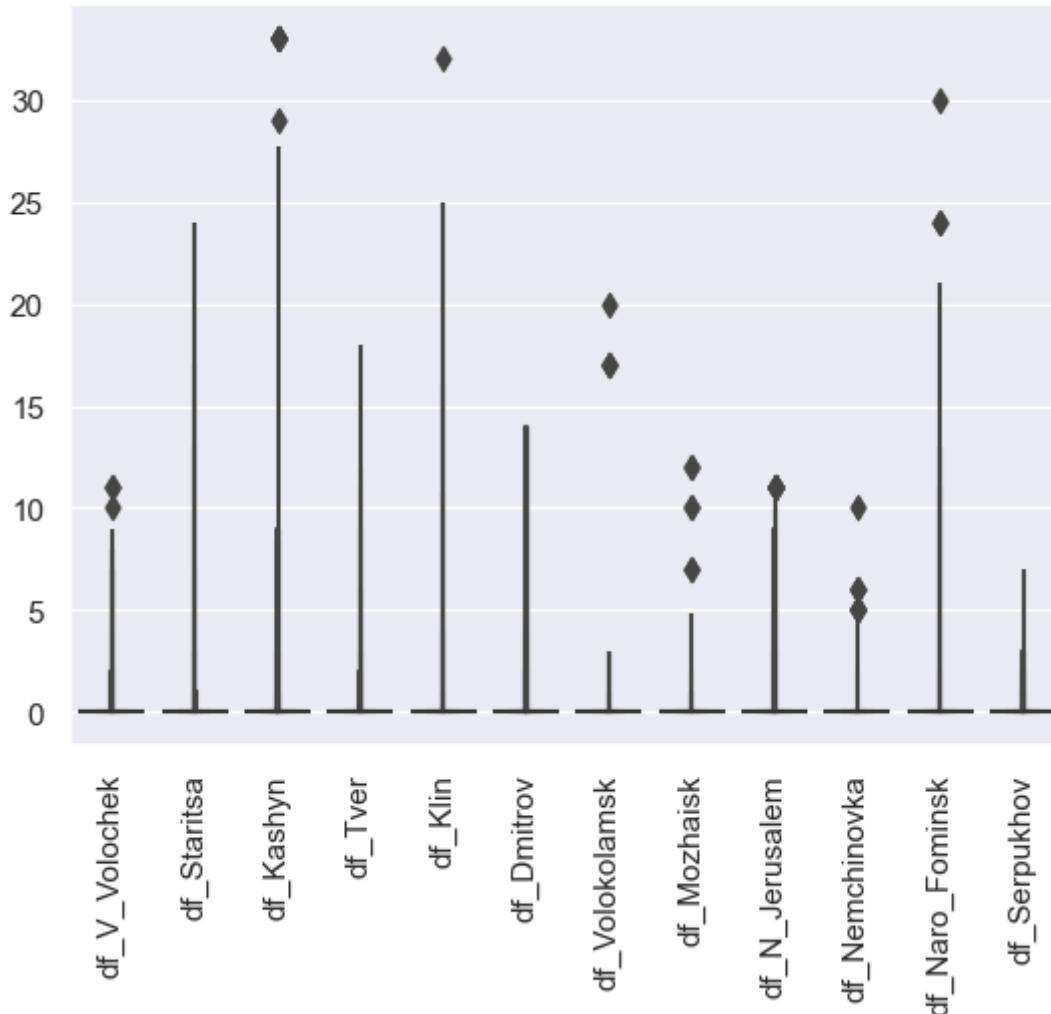
Распределение значений Dew_point в разрезе метеостанций



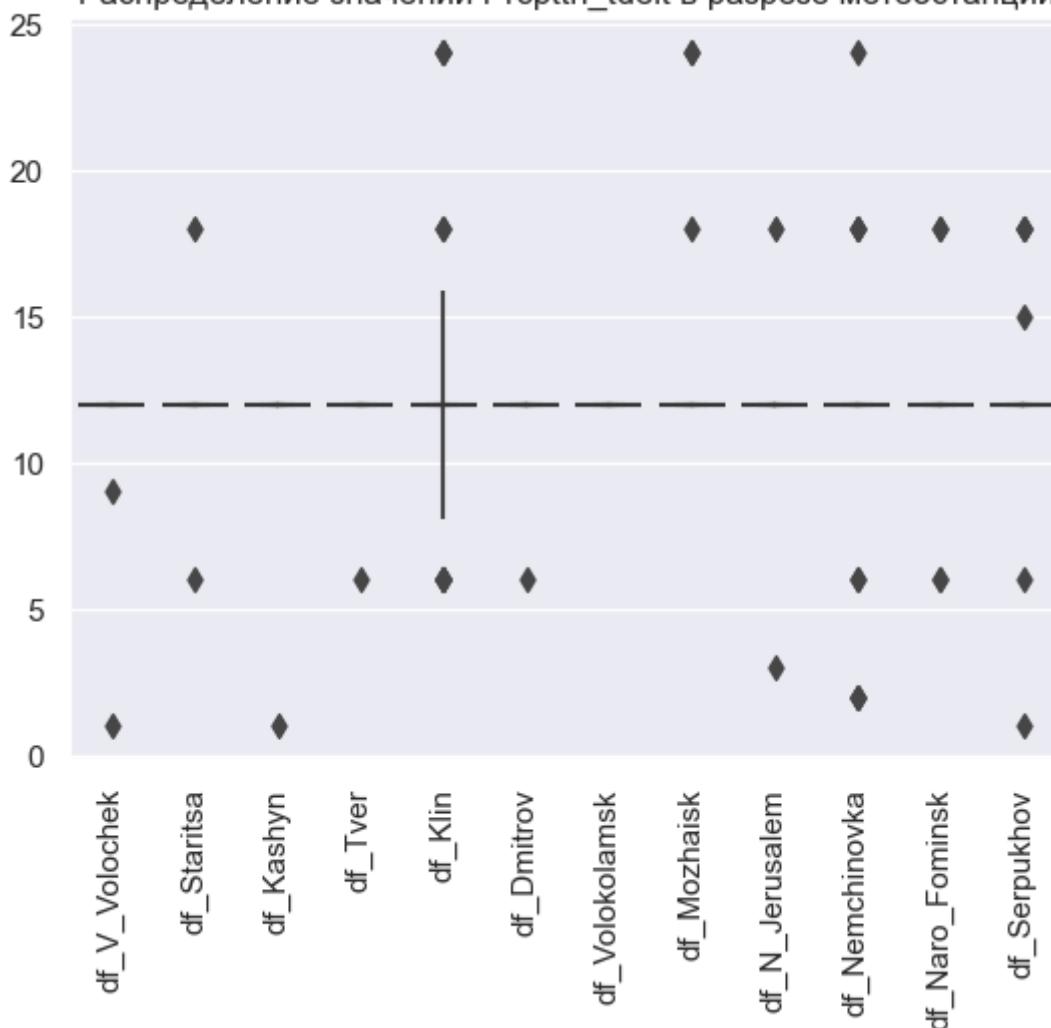
Распределение значений Prcptn в разрезе метеостанций



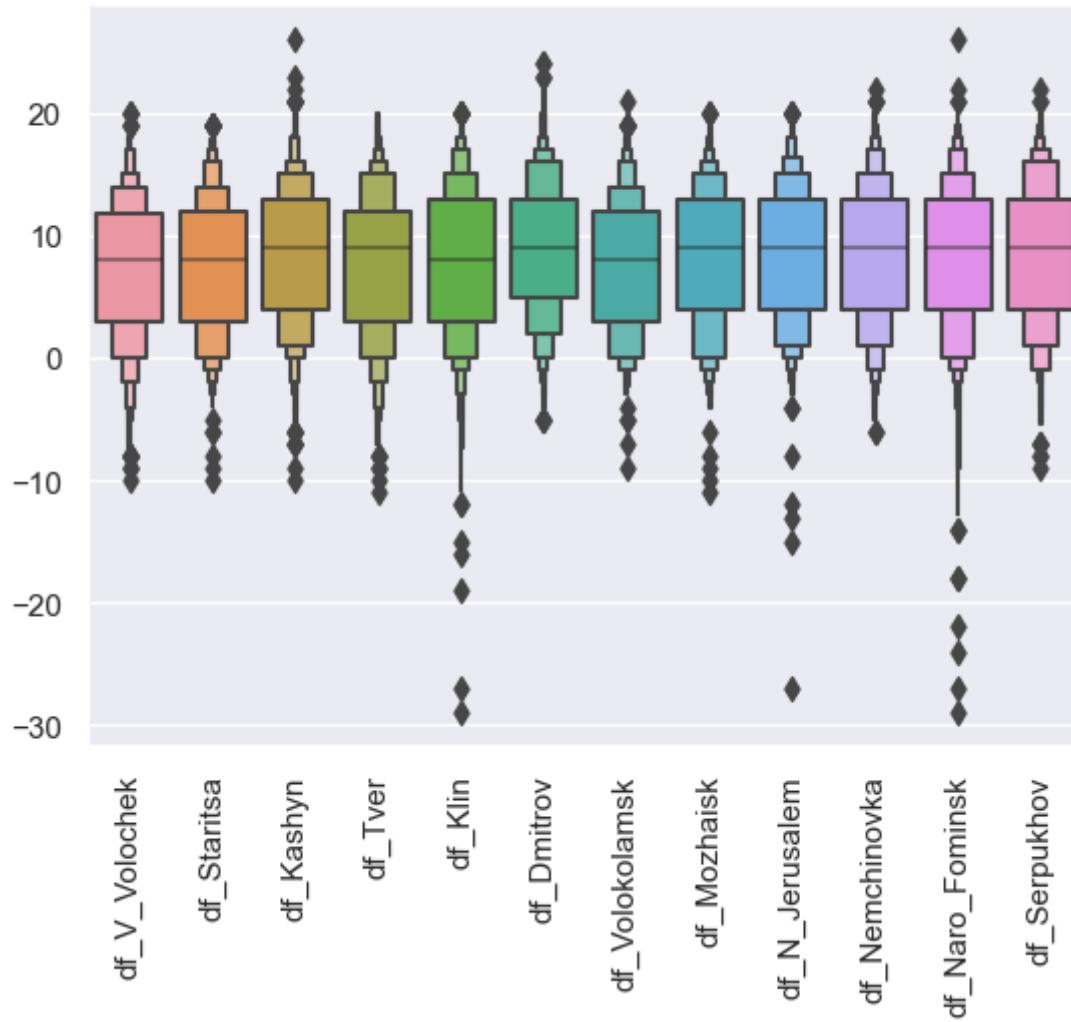
Распределение значений Depo_diam_mm в разрезе метеостанций



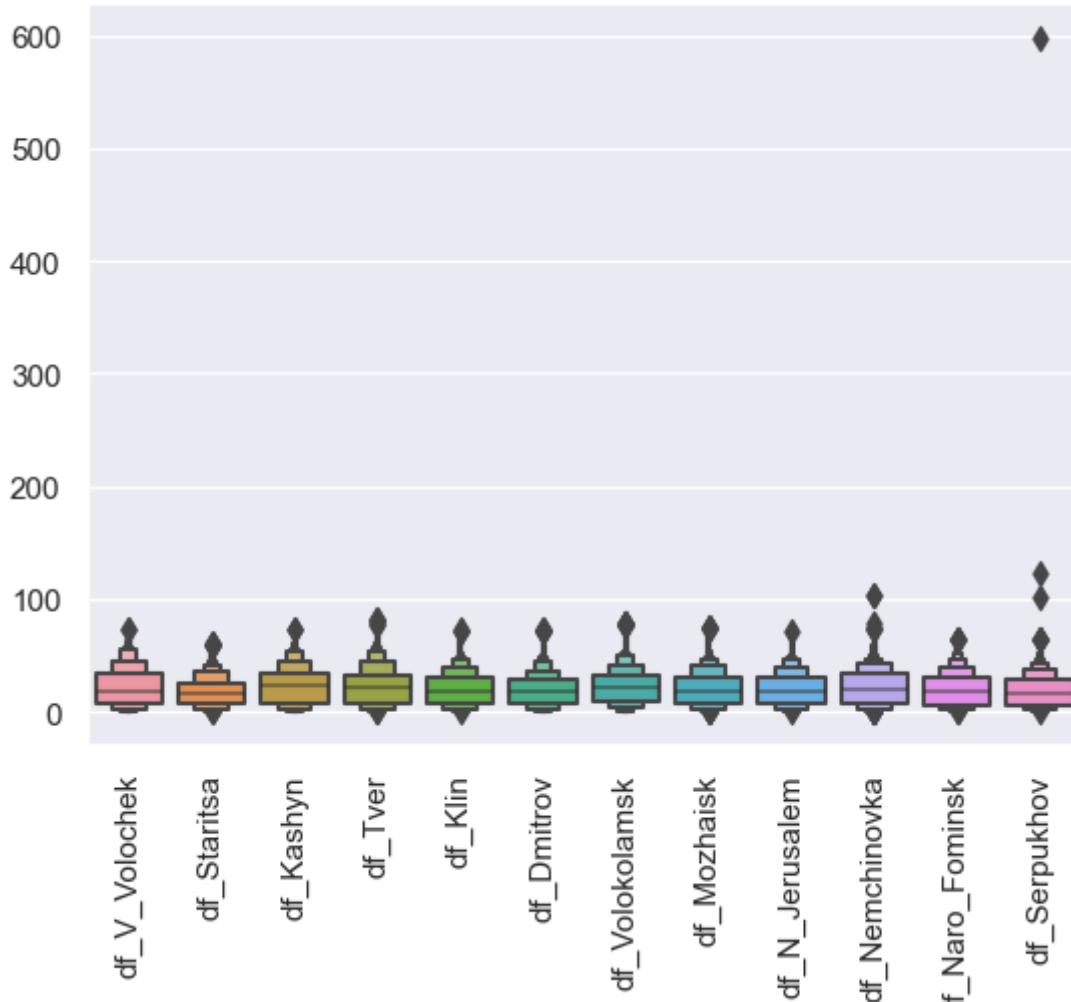
Распределение значений Prcptn_tdelt в разрезе метеостанций



Распределение значений Soil_T в разрезе метеостанций



Распределение значений Snow_height в разрезе метеостанций



ВЫВОД

Как видно из построенных графиков, существуют выбросы, которые невозможно объяснить сложившимися метеорологическими условиями на данной метеостанции, например, слишком низкие показатели давления, слишком высокая скорость ветра и т.п. Скорее всего, это - ошибки.

Более того, данный график не учитывает сезонные колебания показателей, аномальные значения для одного сезона здесь могут попасть в нормальные значения для другого сезона, и таким образом остаться невыявленными

1.2.2. Выборка значений вне пределов 3х сигм среди наблюдений всех метеостанций на данный момент времени

1.2.2.А. Вариант А

Построим отдельный датафрейм, в который запишем момент наблюдения, название датафрейма метеостанции, параметр, аномальное значение, описание погоды (оно поможет определить, является ли аномальное значение ошибкой или нет)

```
In [59]: """# DF для учёта аномальных значений

# Создадим перечень моментов наблюдения:
# массив из моментов времени от минимальной до максимальной даты наблюдения с интервалом 3 часа

# минимальная и максимальная дата наблюдений
start_date = df_stations.beg_date.min()
finish_date = df_stations.end_date.max() + pd.Timedelta(1, 'day') # Добавим еще один день к концу периода

# искомый массив
arr_dates = np.arange(start=start_date, stop=finish_date, step = np.timedelta64(3, 'h'), dtype='datetime64[ns]')

# Список столбцов для датафрейма
list_columns = ['obsrvtn', 'station', 'parameter', 'a_value',
                 'mean_', 'std_', '3sigma_down', '3sigma_up',
                 'description', 'description3h1', 'description3h2']

# Создание пустого датафрейма с заданой структурой столбцов
df_anomalies = pd.DataFrame(columns=list_columns, data=None)

for observation in arr_dates: # по моментам наблюдения

    for parameter in list_parameters: # столбец по списку параметров (определён выше)
        # создадим временные списки df станции и значение параметра
        list_stations, list_values = [], []

        for name, data in dict_df_locations.items(): # по элементам словаря с данными DFs
            # значения parameter на дату (observation) в текущем DF
            if (data.iloc[:,0] == observation).any(): # если момент наблюдения входит в соответствующий архив
                # значение параметра в момент observation в текущем DF (это серия, приведём к списку и выведем):
                param_value = data[(data.iloc[:,0] == observation)][parameter].tolist()[0]
```

```

        if pd.isna(param_value): #если pd.isna() от param_value вернёт True, то есть param_value = nan
            continue
        else:
            # добавляем в list_tmp значение параметра и станцию
            list_stations.append(name)
            list_values.append(param_value)
    else:
        continue
# Список значений parameter по всем станциям на момент наблюдения observation создан

# Если сумма True - обратных False от isna - <= 1, то есть если список без учета NaN содержит 1 и менее элемент
if (sum(~pd.isna(list_values)) <= 1):
    continue # то следующая итерация цикла
else:
    ###
    print('go further', list_values)
    # Анализируем вариацию значений данного параметра на момент наблюдения:
    mean_value = np.nanmean(list_values) # средняя по серии, ингорируя nan
    std_value = np.nanstd(list_values) # сигма по серии, ингорируя nan
    upper_level = mean_value + 3*std_value # верний порог 3x сигм
    lower_level = mean_value - 3*std_value # нижний порог 3x сигм

    for param_val, name in zip(list_values, list_stations):
        # если значение параметра выходит за порог 3x сигм
        if not(lower_level <= param_val <= upper_level):
            ###
            print('control 3sigma', param_val)
            # считываем значения описаний погоды из соответствующего DF на соответствующий момент наблюдения:
            description = dict_df_locations[name][(dict_df_locations[name].iloc[:,0] == observation)]\ 
            ["Wthr_curr"].tolist()[0]
            description3h1 = dict_df_locations[name][(dict_df_locations[name].iloc[:,0] == observation)]\ 
            ["Wthr_3h"].tolist()[0]
            description3h2 = dict_df_locations[name][(dict_df_locations[name].iloc[:,0] == observation)]\ 
            ["Wthr_3h2"].tolist()[0]
            # составляем список параметров, сопутствующих аномальному значению
            list_anomaly = [observation, name[3:], parameter, param_val,
                            mean_value, std_value, lower_level, upper_level,
                            description, description3h1, description3h2]

            # Присоединяем к df_anomalies df, созданный из списка list_anomaly
            ###
            list_values, list_anomaly
            df_anomalies = pd.concat([df_anomalies, pd.DataFrame([list_anomaly], columns = list_columns)], 
            axis=0, ignore_index=True)
        else:
            continue
print(observation, '\r', end='')

```

```
df_anomalies  
'''';
```

```
In [60]: # df_anomalies.to_csv(path + 'df_anomalies.csv')
```

```
In [61]: df_anomalies = pd.read_csv(path + 'df_anomalies.csv', index_col=0)  
df_anomalies.head(3)
```

```
Out[61]:
```

	obsvtn	station	parameter	a_value	mean_	std_	3sigma_down	3sigma_up	description	description3h1	description3h2
0	2005-08-28 21:00:00	Staritsa	Wind_speed	1.0	0.090909	0.287480	-0.771530	0.953348		NaN	NaN
1	2005-09-03 09:00:00	Tver	Humid	48.0	90.000000	13.611492	49.165523	130.834477		NaN	NaN
2	2005-09-03 09:00:00	Tver	Dew_point	1.0	9.445455	2.676960	1.414574	17.476335		NaN	NaN

1.2.2.Б. Вариант Б

Воспользуемся уже созданным архивом значений параметров

```
In [62]: for data in dict_df_parameters.values():  
    data.sample(3)
```

```
Out[62]:
```

	T#V_Volochev	T#Staritsa	T#Kashyn	T#Tver	T#Klin	T#Dmitrov	T#Volokolamsk	T#Mozhaisk	T#N_Jerusalem	T#Nemchinovka	T#Naro_Fomin
2008-08-17 03:00:00	19.7	18.5	18.7	19.4	20.4	20.8	19.4	16.6	15.9	NaN	18
2021-12-21 03:00:00	-18.3	-18.0	-19.0	-17.3	-17.1	-17.6	-17.2	-15.8	-15.9	-15.9	-15
2011-12-15 09:00:00	2.1	1.6	1.3	1.6	1.5	1.3	1.3	1.4	1.3	1.2	1

Out[62]:	P_station#V_Volochev	P_station#Staritsa	P_station#Kashyn	P_station#Tver	P_station#Klin	P_station#Dmitrov	P_station#Volokolamsk	P_station#
2016-12-26 12:00:00	736.6	738.3	742.0	741.8	741.3	740.2	NaN	
2014-03-13 15:00:00	743.3	741.7	744.1	745.2	742.7	741.0	740.6	

Out[62]:	P_sea#V_Volochev	P_sea#Staritsa	P_sea#Kashyn	P_sea#Tver	P_sea#Klin	P_sea#Dmitrov	P_sea#Volokolamsk	P_sea#Mozhaisk	P_sea#N_Jerusalem
2010-07-03 09:00:00	761.3	760.6	761.1	760.8	759.9	759.6	759.9	760.8	760.8
2019-09-28 06:00:00	763.6	763.9	764.5	764.3	764.0	764.0	763.9	764.1	764.1
2011-05-18 18:00:00	NaN	759.8	NaN	758.8	758.2	758.2	759.2	759.8	759.8

Out[62]:	Humid#V_Volochev	Humid#Staritsa	Humid#Kashyn	Humid#Tver	Humid#Klin	Humid#Dmitrov	Humid#Volokolamsk	Humid#Mozhaisk	Humid#
2011-06-22 03:00:00	97.0	84.0	95.0	92.0	96.0	91.0	97.0	96.0	
2018-05-31 06:00:00	64.0	70.0	67.0	76.0	72.0	85.0	74.0	73.0	
2005-02-16 15:00:00	Nan	95.0	Nan	88.0	96.0	92.0	86.0	92.0	
Out[62]:	Wind_dir#V_Volochev	Wind_dir#Staritsa	Wind_dir#Kashyn	Wind_dir#Tver	Wind_dir#Klin	Wind_dir#Dmitrov	Wind_dir#Volokolamsk	Wind_dir#	
2022-02-13 09:00:00	WSW	W	W	WSW	WSW	W	W		
2012-11-24 18:00:00	SSE	SSW	SW	SSE	S	SE	SSW		
2011-03-30 21:00:00	WNW	WNW	NW	NW	WNW	WNW		NW	
Out[62]:	Wind_dir360#V_Volochev	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#		
2006-04-22 06:00:00	Nan	247.5	Nan	247.5	270.0	270.0			
2006-04-28 06:00:00	Nan	90.0	Nan	360.0	360.0	360.0			
2018-10-24 03:00:00	112.5	157.5	157.5	157.5	135.0	157.5			

Out[62]:	Wind_dir6k#V_Volochek	Wind_dir6k#Staritsa	Wind_dir6k#Kashyn	Wind_dir6k#Tver	Wind_dir6k#Klin	Wind_dir6k#Dmitrov	Wind_dir6k#Volokola
2014-03-12 12:00:00	4500.0	5250.0	5625.0	4500.0	5250.0	5625.0	5250.0
2018-04-17 00:00:00	6000.0	2250.0	4500.0	6000.0	2250.0	6000.0	7250.0
2017-12-20 12:00:00	6000.0	750.0	6000.0	1875.0	750.0	6000.0	2250.0

Out[62]:	Wind_speed#V_Volochev	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2011-01-10 21:00:00	2.0	4.0	3.0	2.0	3.0	3.0	
2021-11-09 18:00:00	3.0	3.0	2.0	2.0	2.0	3.0	
2008-07-02 06:00:00	NaN	2.0	NaN	1.0	0.0	0.0	

Out[62]:	Gusts_3h#V_Volocheok	Gusts_3h#Staritsa	Gusts_3h#Kashyn	Gusts_3h#Tver	Gusts_3h#Klin	Gusts_3h#Dmitrov	Gusts_3h#Volokolamsk	Gusts_3h#I
2007-02-03 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13.0
2008-08-04 21:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012-08-18 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Out[62]:	Cloudness#V_Volocheok	Cloudness#Staritsa	Cloudness#Kashyn	Cloudness#Tver	Cloudness#Klin	Cloudness#Dmitrov	Cloudness#Volokolamsk	Cl
2018-10-24 12:00:00	от 90 менее 100%	100%.	от 90 менее 100%	100%.	100%.	100%.	100%.	NaN
2019-01-09 06:00:00	100%.	0%	от 90 менее 100%	100%.	100%.	100%.	100%.	NaN
2007-12-08 18:00:00	NaN	100%.	NaN	100%.	100%.	100%.	100%.	100%.

Out[62]:	Wthr_curr#V_Volocheok	Wthr_curr#Staritsa	Wthr_curr#Kashyn	Wthr_curr#Tver	Wthr_curr#Klin	Wthr_curr#Dmitrov	Wthr_curr#Volokolamsk	Wt
2013-04-16 12:00:00	n/a	Состояние неба в общем не изменилось.	n/a	Дымка.	n/a	n/a	n/a	n/a
2014-03-28 00:00:00	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
2008-12-27 15:00:00	Снег непрерывный слабый в срок наблюдения.	n/a	Снег непрерывный слабый в срок наблюдения.	Снег непрерывный слабый в срок наблюдения.	n/a	n/a	n/a	Состояние неба в общем не изменилось.

Out[62]:	Prcptn_like#V_Volochev	Prcptn_like#Staritsa	Prcptn_like#Kashyn	Prcptn_like#Tver	Prcptn_like#Klin	Prcptn_like#Dmitrov	Prcptn_like#Volok
2019-12-16 00:00:00	Снег непрерывный слабый в срок наблюдения.	Состояние неба в общем не изменилось.	Снег непрерывный слабый в срок наблюдения.	Ливневый с ливневый дождь			
2013-08-07 00:00:00	Более или менее сплошной слой приземного или л...	n/a	Дымка.	Дымка.	Дымка.	n/a	
2014-05-11 06:00:00	Ливневый(ые) дождь(и) слабый(ые) в срок наблюд...	n/a	Состояние неба в общем не изменилось.	n/a	n/a	n/a	
Out[62]:	Prcptn_depo#V_Volochev	Prcptn_depo#Staritsa	Prcptn_depo#Kashyn	Prcptn_depo#Tver	Prcptn_depo#Klin	Prcptn_depo#Dmitrov	Prcptn_depo#Volok
2011-04-26 18:00:00	n/a	n/a	n/a	n/a	n/a	n/a	n/a
2005-11-23 00:00:00	NaN	n/a	NaN	n/a	n/a	n/a	n/a
2006-12-15 09:00:00	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Out[62]:	Depo_diam_mm#V_Volochev	Depo_diam_mm#Staritsa	Depo_diam_mm#Kashyn	Depo_diam_mm#Tver	Depo_diam_mm#Klin	Depo_diam_mm#Dmitrov	Depo_diam_mm#Volok
2012-11-14 15:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2021-08-10 15:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-12-05 06:00:00	0.0	1.0	0.0	1.0	0.0	0.0	0.0

Wthr_3h#V_Volochek Wthr_3h#Staritsa Wthr_3h#Kashlyn Wthr_3h#Tver Wthr_3h#Klin Wthr_3h#Dmitrov Wthr_3h#Volokolamsk Wthr_3h#Moze

NaN	Ливень (ливни).	Дождь со снегом или другими видами твердых осадков	NaN	Дождь со снегом или другими видами твердых осадков	NaN	Дождь.
2022-02-20 00:00:00	NaN	Дождь со снегом или другими видами твердых осадков	NaN	Дождь со снегом или другими видами твердых осадков	NaN	Дождь.
2018-12-28 09:00:00	NaN	Снег и/или другие виды твердых осадков	Снег и/или другие виды твердых осадков	Снег и/или другие виды твердых осадков	Снег и/или другие виды твердых осадков	Снег и/или другие виды твердых осадков
2015-02-17 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN

Out[62]: Wthr_3h2#V_Volochek Wthr_3h2#Staritsa Wthr_3h2#Kashyn Wthr_3h2#Tver Wthr_3h2#Klin Wthr_3h2#Dmitrov Wthr_3h2#Volokolamsk Wthr

2010-05-06 18:00:00	NaN	NaN	NaN	NaN	NaN	NaN	Облаха покрываали более половины неба в течение...
2016-11-12 03:00:00	Метель	Метель	Дождь.	Облаха покрываали более половины неба в течение...	Облаха покрываали более половины неба в течение...	Метель	NaN
2020-04-22 21:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Out[62]:	T_min#V_Volochek	T_min#Staritsa	T_min#Kashyn	T_min#Tver	T_min#Klin	T_min#Dmitrov	T_min#Volokolamsk	T_min#Mozhaisk	T_min#N_Jer
2021-06-21 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2010-11-23 03:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012-03-01 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Out[62]:	T_max#V_Volochek	T_max#Staritsa	T_max#Kashyn	T_max#Tver	T_max#Klin	T_max#Dmitrov	T_max#Volokolamsk	T_max#Mozhaisk	T_max#N
2007-06-24 06:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2008-04-16 21:00:00	7.8	5.9	8.4	7.5	6.9	6.5	6.6	8.5	
2018-12-18 21:00:00	-13.8	-13.4	-12.2	-11.9	-12.3	-11.9	NaN	NaN	

Out[62]:

	Cl_Cumls#V_Volocheok	Cl_Cumls#Staritsa	Cl_Cumls#Kashyn	Cl_Cumls#Tver	Cl_Cumls#Klin	Cl_Cumls#Dmitrov	Cl_Cumls#Volokolamsk	Cl_Cun
2011-12-22 12:00:00	Слоисто-кучевых, слоистых, кучевых или кучево-...	Слоистые разорванные или кучевые разорванные о...	Слоистые разорванные или кучевые разорванные о...	Слоистые разорванные или кучевые разорванные о...	Слоисто-кучевых, слоистых, кучевых или кучево-...	Слоистые разорванные или кучевые разорванные о...	Слоистые разорванные или кучевые разорванные о...	Кучевые волоки
2009-11-12 21:00:00	Слоисто-кучевые, образовавшиеся не из кучевых.	Слоисто-кучевые, образовавшиеся из кучевых.	Слоисто-кучевые, образовавшиеся из кучевых.	Слоистые разорванные или кучевые разорванные о...	Слои образо			
2020-04-08 12:00:00	Слоисто-кучевые, образовавшиеся не из кучевых.	Слоисто-кучевые, образовавшиеся не из кучевых.	Кучево-дождевые лысые с кучевыми, слоисто-куче...	Слоисто-кучевые, образовавшиеся не из кучевых.	Кучево-дождевые лысые с кучевыми, слоисто-куче...	Кучево-дождевые лысые с кучевыми,	NaN	

Out[62]:

	Cl_viewd#V_Volocheok	Cl_viewd#Staritsa	Cl_viewd#Kashyn	Cl_viewd#Tver	Cl_viewd#Klin	Cl_viewd#Dmitrov	Cl_viewd#Volokolamsk	Cl_viewd#M
2017-11-29 15:00:00	70 – 80%.	90 или более, но не 100%	90 или более, но не 100%	100%.	100%.	100%.	100%.	NaN
2013-07-22 03:00:00	100%.	100%.	90 или более, но не 100%	100%.	100%.	100%.	100%.	100%.
2007-09-23 06:00:00	NAN	70 – 80%.	NAN	70 – 80%.	Облаков нет.	Облаков нет.	20–30%.	Обла

Out[62]:

	Cl_bottom#V_Volochev	Cl_bottom#Staritsa	Cl_bottom#Kashyn	Cl_bottom#Tver	Cl_bottom#Klin	Cl_bottom#Dmitrov	Cl_bottom#Volokolamsk	C
2021-02-20 06:00:00	2500 или более, или облаков нет.	NaN	2500 или более, или облаков нет.	2500 или более, или облаков нет.	1000-1500	2500 или более, или облаков нет.	2500 или более, или облаков нет.	
2020-01-06 06:00:00	600-1000	300-600	600-1000	600-1000	600-1000	300-600	600-1000	
2013-04-01 03:00:00	600-1000	600-1000	2500 или более, или облаков нет.					

Out[62]:

	Cl_cumls_hi#V_Volochev	Cl_cumls_hi#Staritsa	Cl_cumls_hi#Kashyn	Cl_cumls_hi#Tver	Cl_cumls_hi#Klin	Cl_cumls_hi#Dmitrov	Cl_cumls_hi#Vol
2021-05-25 09:00:00	Высококучевых, высокослоистых или слоисто-дожд...	Высококучевых, высокослоистых или слоисто-дожд...	NaN	Высококучевых, высокослоистых или слоисто-дожд...			
2012-02-23 03:00:00	Высококучевых, высокослоистых или слоисто-дожд...	NaN	Высококучевых, высокослоистых или слоисто-дожд...	Высококучевые просвечивающие, расположенные на...	Высокослоистые непросвечивающие или слоисто-до...	Высококучевых, высокослоистых или слоисто-дожд...	
2021-04-07 00:00:00	Высококучевых, высокослоистых или слоисто-дожд...	Высококучевые, просвечивающие или плотные в дв...	Высококучевых, высокослоистых или слоисто-дожд...	Высококучевые просвечивающие, расположенные на...	NaN	Высокослоистые непросвечивающие или слоисто-до...	Высокослоистые непросвечиваю...

Out[62]:

	Cl_cirrus#V_Volochev	Cl_cirrus#Staritsa	Cl_cirrus#Kashyn	Cl_cirrus#Tver	Cl_cirrus#Klin	Cl_cirrus#Dmitrov	Cl_cirrus#Volokolamsk	Cl_cirrus#Moz
--	----------------------	--------------------	------------------	----------------	----------------	-------------------	-----------------------	---------------

2009-07-04 00:00:00	NaN	Перистые нитевидные, иногда когтевидные, не ра...	NaN	NaN	NaN	Перисто-кучевые одни или перисто-кучевые, сопр...	NaN
2008-11-09 15:00:00	NaN	NaN	Перистые нитевидные, иногда когтевидные, не ра...	Перистые нитевидные, иногда когтевидные, не ра...	Перисто-кучевые одни или перисто-кучевые, сопр...	Перистые нитевидные, иногда когтевидные, не ра...	Перисто-кучевые одни или перисто-кучевые, сопр...
2007-05-12 18:00:00	NaN	Перистые нитевидные, иногда когтевидные, не ра...	NaN	NaN	Перисто-кучевые одни или перисто-кучевые, сопр...	Перистые плотные в виде клаочьев или скрученных...	NaN

Out[62]:

	Visibility#V_Volochev	Visibility#Staritsa	Visibility#Kashyn	Visibility#Tver	Visibility#Klin	Visibility#Dmitrov	Visibility#Volokolamsk	Visibility#Moz
--	-----------------------	---------------------	-------------------	-----------------	-----------------	--------------------	------------------------	----------------

2021-01-15 15:00:00	10.0	20.0	10.0	10.0	10.0	10.0	4.0
2018-11-17 03:00:00	10.0	10.0	50.0	10.0	10.0	10.0	10.0
2009-09-14 03:00:00	17.0	0.2	50.0	10.0	1.0	10.0	10.0

Out[62]:

	Snow_height#V_Volochek	Snow_height#Staritsa	Snow_height#Kashyn	Snow_height#Tver	Snow_height#Klin	Snow_height#Dmitrov	Snow_height#Leningrad
--	------------------------	----------------------	--------------------	------------------	------------------	---------------------	-----------------------

2014-07-22 06:00:00	NaN						
2013-08-19 03:00:00	NaN						
2015-10-20 03:00:00	NaN						

In [63]:

```
# функция вычисления пределов 3х сигм для ряда значений в DF
def sigma3_limits(row_):
    """ Принимает значения из ряда DF, возвращает кортеж: нижний и верхний порог 3х сигм """
    # Анализируем вариацию значений данного параметра на момент наблюдения:
    mean_value_ = (row_).mean() # средняя по серии, ингорируя nan
    std_value_ = (row_).std() # сигма по серии, ингорируя nan
    upper_level_ = mean_value_ + 3*std_value_ # верхний порог 3х сигм
    lower_level_ = mean_value_ - 3*std_value_ # нижний порог 3х сигм

    return (lower_level_, upper_level_)

# функция для вычисления выбросов вне пределов 3х сигм для параметра x_ исходя из результатов sigma3_limits
def outliers(row_):
    """
    Принимает значения из ряда DF,
    Возвращает Если выбросы обнаружены - список из кортежей (на каждый выброс свой кортеж):
    значение выброса, его индекс в серии и границы 3х сигм, ему соответствующие
    Если выбросов нет - возвращает NaN
    """

    sigma3_levels = sigma3_limits(row_) # Определим границы интервала 3х сигм (через функцию sigma3_limits())
    list_outliers =[] # Список для значений и параметров выбросов
    for val_ in row_: # по значению в полученной строке DF
        if val_ < sigma3_levels[0] or val_ > sigma3_levels[1]: #Если значение вне границ 3х сигм
            idx_ = row_.tolist().index(val_) # порядковый номер значения в row_, приведённом к списку
            # добавить выброс и его параметры в список:
            list_outliers.append((val_, idx_, sigma3_levels[0], sigma3_levels[1]))
    if len(list_outliers) > 0: # Если выбросы есть (длина списка больше 0)
        return list_outliers # Вывести список
```

```

else:
    return np.nan # Иначе вывести NaN

```

```

In [64]: # Определим столбец в DFs параметров с указанием значения выброса, номера столбца с выбросом, границ 3х сигм
for param in dict_df_parameters.keys():
    if param[3:] in list_parameters: # в списке параметров указаны только параметры, выраженные числовым значением

        # Применяем функцию outliers() - см. выше
        dict_df_parameters[param] = dict_df_parameters[param].assign(
            sigma3=dict_df_parameters[param].apply(lambda x: outliers(x[:-1]), axis=1))
dict_df_parameters[param].head(5)

```

Out[64]:

	T#V_Volochev	T#Staritsa	T#Kashyn	T#Tver	T#Klin	T#Dmitrov	T#Volokolamsk	T#Mozhaisk	T#N_Jerusalem	T#Nemchinovka	T#Naro_Fomin
2022-06-09 21:00:00	19.8	18.4	17.8	18.8	NaN	NaN	18.7	NaN	NaN	NaN	NaN
2022-06-09 18:00:00	22.3	18.1	20.8	21.8	NaN	NaN	26.7	NaN	NaN	NaN	NaN
2022-06-09 15:00:00	22.6	24.8	26.8	26.5	NaN	NaN	28.2	NaN	NaN	NaN	NaN
2022-06-09 12:00:00	23.3	25.5	25.0	26.1	NaN	NaN	26.7	NaN	NaN	NaN	NaN
2022-06-09 09:00:00	22.4	22.1	20.4	21.9	22.4	NaN	23.0	NaN	22.5	NaN	22

Out[64]:	P_station#V_Volochek	P_station#Staritsa	P_station#Kashyn	P_station#Tver	P_station#Klin	P_station#Dmitrov	P_station#Volokolamsk	P_station#Mozhaisk
2022-06-09 21:00:00	743.9	743.0	746.5	746.3	NaN	NaN	742.0	741.8
2022-06-09 18:00:00	744.0	743.1	746.6	746.4	NaN	NaN	741.5	741.8
2022-06-09 15:00:00	744.0	742.5	746.2	745.8	NaN	NaN	742.3	741.8
2022-06-09 12:00:00	744.1	742.9	747.2	746.4	NaN	NaN	742.3	741.8
2022-06-09 09:00:00	743.7	743.2	747.8	746.8	745.4	NaN	742.6	741.8

Out[64]:	P_sea#V_Volochek	P_sea#Staritsa	P_sea#Kashyn	P_sea#Tver	P_sea#Klin	P_sea#Dmitrov	P_sea#Volokolamsk	P_sea#Mozhaisk	P_sea#N_Jerusalem
2022-06-09 21:00:00	758.5	759.1	758.6	759.0	NaN	NaN	759.1	NaN	NaN
2022-06-09 18:00:00	758.5	759.3	758.7	758.9	NaN	NaN	758.2	NaN	NaN
2022-06-09 15:00:00	758.5	758.3	758.1	758.2	NaN	NaN	758.4	NaN	NaN
2022-06-09 12:00:00	758.5	758.8	759.1	758.8	NaN	NaN	759.1	NaN	NaN
2022-06-09 09:00:00	758.2	759.1	760.0	759.4	759.8	NaN	759.5	NaN	759.8

Out[64]:	P_drift#V_Volochek	P_drift#Staritsa	P_drift#Kashyn	P_drift#Tver	P_drift#Klin	P_drift#Dmitrov	P_drift#Volokolamsk	P_drift#Mozhaisk	P_drift#Ostankino
2022-06-09 21:00:00	-0.1	-0.1	-0.1	-0.1	NaN	NaN	NaN	NaN	NaN
2022-06-09 18:00:00	0.0	0.6	0.4	0.6	NaN	NaN	NaN	NaN	NaN
2022-06-09 15:00:00	-0.1	-0.4	-1.0	-0.6	NaN	NaN	NaN	NaN	NaN
2022-06-09 12:00:00	0.4	-0.3	-0.6	-0.4	NaN	NaN	NaN	NaN	NaN
2022-06-09 09:00:00	-0.3	-0.3	-0.3	-0.3	-0.2	NaN	NaN	NaN	NaN

Out[64]:	Humid#V_Volochek	Humid#Staritsa	Humid#Kashyn	Humid#Tver	Humid#Klin	Humid#Dmitrov	Humid#Volokolamsk	Humid#Mozhaisk	Humid#Ostankino
2022-06-09 21:00:00	76.0	92.0	89.0	92.0	NaN	NaN	80.0	NaN	NaN
2022-06-09 18:00:00	65.0	89.0	71.0	63.0	NaN	NaN	34.0	NaN	NaN
2022-06-09 15:00:00	61.0	45.0	40.0	34.0	NaN	NaN	32.0	NaN	NaN
2022-06-09 12:00:00	59.0	41.0	36.0	38.0	NaN	NaN	36.0	NaN	NaN
2022-06-09 09:00:00	56.0	56.0	52.0	48.0	41.0	NaN	41.0	NaN	NaN

Out[64]:	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#Vol
2022-06-09 21:00:00	247.5	270.0	225.0	360.0	NaN	NaN	NaN
2022-06-09 18:00:00	270.0	22.5	225.0	292.5	NaN	NaN	NaN
2022-06-09 15:00:00	247.5	270.0	247.5	247.5	NaN	NaN	NaN
2022-06-09 12:00:00	292.5	247.5	180.0	180.0	NaN	NaN	NaN
2022-06-09 09:00:00	225.0	202.5	202.5	202.5	180.0	NaN	NaN

Out[64]:	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2022-06-09 21:00:00	1.0	1.0	1.0	0.0	NaN	NaN	NaN
2022-06-09 18:00:00	1.0	2.0	1.0	1.0	NaN	NaN	NaN
2022-06-09 15:00:00	2.0	1.0	2.0	2.0	NaN	NaN	NaN
2022-06-09 12:00:00	3.0	3.0	2.0	1.0	NaN	NaN	NaN
2022-06-09 09:00:00	1.0	2.0	1.0	2.0	1.0	NaN	NaN

Out[64]:	Depo_diam_mm#V_Volochek	Depo_diam_mm#Staritsa	Depo_diam_mm#Kashyn	Depo_diam_mm#Tver	Depo_diam_mm#Klin	Depo_diam_mm#Dm
2022-06-09 21:00:00	0.0	0.0	0.0	0.0	0.0	NaN
2022-06-09 18:00:00	0.0	0.0	0.0	0.0	0.0	NaN
2022-06-09 15:00:00	0.0	0.0	0.0	0.0	0.0	NaN
2022-06-09 12:00:00	0.0	0.0	0.0	0.0	0.0	NaN
2022-06-09 09:00:00	0.0	0.0	0.0	0.0	0.0	0.0

Out[64]:	T_min#V_Volochek	T_min#Staritsa	T_min#Kashyn	T_min#Tver	T_min#Klin	T_min#Dmitrov	T_min#Volokolamsk	T_min#Mozhaisk	T_min#N_Jer
2022-06-09 21:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 18:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 15:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 09:00:00	9.2	7.9	5.8	7.9	7.0	NaN	9.2	NaN	NaN

Out[64]:	T_max#V_Volochek	T_max#Staritsa	T_max#Kashyn	T_max#Tver	T_max#Klin	T_max#Dmitrov	T_max#Volokolamsk	T_max#Mozhaisk	T_max#N
2022-06-09 21:00:00	24.6	27.4	27.0	28.2	NaN	NaN	28.4	NaN	NaN
2022-06-09 18:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 15:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 09:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Out[64]:	Visibility#V_Volochek	Visibility#Staritsa	Visibility#Kashyn	Visibility#Tver	Visibility#Klin	Visibility#Dmitrov	Visibility#Volokolamsk	Visibility#Mozhaisk	Visibility#N
2022-06-09 21:00:00	20.0	10.0	20.0	10.0	NaN	NaN	10.0	NaN	10.0
2022-06-09 18:00:00	20.0	10.0	20.0	10.0	NaN	NaN	10.0	NaN	10.0
2022-06-09 15:00:00	20.0	10.0	20.0	10.0	NaN	NaN	10.0	NaN	10.0
2022-06-09 12:00:00	20.0	20.0	20.0	10.0	NaN	NaN	10.0	NaN	10.0
2022-06-09 09:00:00	20.0	20.0	20.0	10.0	10.0	NaN	10.0	NaN	10.0

Out[64]:	Dew_point#V_Volochev	Dew_point#Staritsa	Dew_point#Kashyn	Dew_point#Tver	Dew_point#Klin	Dew_point#Dmitrov	Dew_point#Volokolamsk	F
2022-06-09 21:00:00	15.5	17.1	15.9	17.5	NaN	NaN	15.2	
2022-06-09 18:00:00	15.4	16.3	15.4	14.3	NaN	NaN	9.6	
2022-06-09 15:00:00	14.7	12.0	12.1	9.3	NaN	NaN	9.9	
2022-06-09 12:00:00	14.7	11.2	8.8	10.5	NaN	NaN	10.5	
2022-06-09 09:00:00	13.2	13.0	10.2	10.2	8.6	NaN	8.9	
Out[64]:	Prcptn#V_Volochev	Prcptn#Staritsa	Prcptn#Kashyn	Prcptn#Tver	Prcptn#Klin	Prcptn#Dmitrov	Prcptn#Volokolamsk	Prcptn#Mozhaisk F
2022-06-09 21:00:00	0.0	2.0	0.0	0.7	NaN	NaN	0.1	NaN
2022-06-09 18:00:00	0.0	2.0	0.0	0.1	NaN	NaN	0.0	NaN
2022-06-09 15:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 09:00:00	0.0	0.0	0.0	0.0	0.0	NaN	0.0	NaN

Out[64]:	Prcptn_tdelt#V_Volochek	Prcptn_tdelt#Staritsa	Prcptn_tdelt#Kashyn	Prcptn_tdelt#Tver	Prcptn_tdelt#Klin	Prcptn_tdelt#Dmitrov	Prcptn_tde
2022-06-09 21:00:00	12.0	12.0	12.0	12.0	NaN	NaN	NaN
2022-06-09 18:00:00	12.0	12.0	12.0	12.0	NaN	NaN	NaN
2022-06-09 15:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 09:00:00	12.0	12.0	12.0	12.0	12.0	12.0	NaN

Out[64]:	Soil_T#V_Volochek	Soil_T#Staritsa	Soil_T#Kashyn	Soil_T#Tver	Soil_T#Klin	Soil_T#Dmitrov	Soil_T#Volokolamsk	Soil_T#Mozhaisk	Soil_T#N_Jeru
2022-06-09 21:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 18:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 15:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 09:00:00	9.0	9.0	8.0	9.0	11.0	NaN	10.0	NaN	NaN

Out[64]:

	Snow_height#V_Volochek	Snow_height#Staritsa	Snow_height#Kashyn	Snow_height#Tver	Snow_height#Klin	Snow_height#Dmitrov	Snow_height#Moscow
2022-06-09 21:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 18:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 15:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2022-06-09 09:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [65]:

```
# Выведем только значения выбросов
list_max_outliers = [] # Список с максимумами значений выбросов на момент наблюдения по всем станциям
for param, param_data in dict_df_parameters.items():
    if param[3:] in list_parameters: # в списке параметров указаны только параметры, выраженные числовым значением
        print(f"Выбросы по параметру {param}", end = '')
        param_data[(~pd.isna(param_data.sigmas))] # sigmas не является NaN
        # Максимум значений выбросов на момент наблюдения по всем станциям
        max_outliers = param_data[(~pd.isna(param_data.sigmas))].sigmas.apply(lambda x: len(x)).max() # длина списка в sigmas
        list_max_outliers.append((param, max_outliers)) # добавляем в список кортеж: параметр, максимум выбросов на момент
```

Выбросы по параметру df_T

Out[65]:	T#V_Volochek	T#Staritsa	T#Kashyn	T#Tver	T#Klin	T#Dmitrov	T#Volokolamsk	T#Mozhaisk	T#N_Jerusalem	T#Nemchinovka	T#Naro_Fomin
2014-09-30 06:00:00	2.2	10.9	10.7	11.1	11.2	10.8	11.1	11.4	11.0	11.0	11

2011-08-22 15:00:00	21.1	19.8	21.0	21.0	20.1	20.3	19.9	19.4	20.4	-20.1	19
----------------------------	------	------	------	------	------	------	------	------	------	-------	----

2011-08-22 09:00:00	15.9	14.2	15.3	15.2	14.8	15.0	13.6	14.4	15.6	-15.3	14
----------------------------	------	------	------	------	------	------	------	------	------	-------	----

Выбросы по параметру df_P_station

Out[65]:	P_station#V_Volochek	P_station#Staritsa	P_station#Kashyn	P_station#Tver	P_station#Klin	P_station#Dmitrov	P_station#Volokolamsk	P_station#Mozhaisk	P_station#Jerusalem	P_station#Nemchinovka	P_station#Naro_Fomin
2008-12-12 09:00:00	77.9	759.8	765.1	762.9	761.4	760.8	758.7	758.7	758.7	758.7	758.7
2008-04-18 09:00:00	739.8	671.0	743.9	741.9	740.7	739.8	737.5	737.5	737.5	737.5	737.5

Выбросы по параметру df_P_sea

Out[65]:

	P_sea#V_Volochek	P_sea#Staritsa	P_sea#Kashyn	P_sea#Tver	P_sea#Klin	P_sea#Dmitrov	P_sea#Volokolamsk	P_sea#Mozhaisk	P_sea#N_Jerusalem
2019-09-25 09:00:00	763.9	764.7	763.7	764.1	764.5	764.2	764.8	765.5	764.8
2019-06-04 06:00:00	768.4	769.2	769.0	758.2	769.2	769.2	769.0	769.5	769.5
2019-06-04 03:00:00	768.1	768.7	768.5	758.8	768.5	768.5	768.4	768.7	768.7
2019-06-04 00:00:00	767.5	767.9	767.7	759.6	767.8	767.6	767.7	767.7	767.7
2019-06-03 21:00:00	766.2	766.3	766.3	760.0	766.3	766.3	766.2	766.5	766.5
2017-06-15 00:00:00	751.5	751.2	747.6	749.9	749.4	748.6	750.1	93.8	749.4
2016-12-14 03:00:00	760.9	762.4	761.7	762.3	763.2	763.4	763.6	810.8	763.6
2015-06-18 12:00:00	761.6	762.0	761.9	761.9	762.1	762.1	762.1	762.1	753.1
2014-09-11 12:00:00	769.9	770.0	770.9	770.5	770.3	757.6	769.9	770.4	770.4
2014-06-05 18:00:00	764.7	764.5	765.1	764.8	764.5	764.7	764.9	764.6	751.1
2012-07-08 09:00:00	762.4	763.0	763.6	763.4	762.7	763.4	762.9	762.7	762.7

P_sea#V_Volochek	P_sea#Staritsa	P_sea#Kashyn	P_sea#Tver	P_sea#Klin	P_sea#Dmitrov	P_sea#Volokolamsk	P_sea#Mozhaisk	P_sea#N_Jerusalem
2012-03-30 09:00:00	816.7	740.9	739.6	740.1	738.8	738.4	739.0	739.7
2010-01-18 21:00:00	786.2	786.8	785.7	785.8	786.4	785.8	786.2	786.9
2009-03-10 15:00:00	824.6	750.2	750.2	750.1	749.9	749.9	749.8	749.6
2009-02-20 21:00:00	774.1	774.5	774.5	774.0	773.9	774.0	773.7	773.6

Выбросы по параметру df_P_drift

Out[65]:

	P_drift#V_Volochek	P_drift#Staritsa	P_drift#Kashyn	P_drift#Tver	P_drift#Klin	P_drift#Dmitrov	P_drift#Volokolamsk	P_drift#Mozhaisk	P_drift#Ostankino
2016-10-06 15:00:00	-0.3	-0.1	-0.4	-0.2	-0.3	-0.1	-0.4	-0.4	-0.2
2016-10-06 12:00:00	-0.5	-0.6	-0.5	-0.6	-0.6	-0.7	-0.3	-0.4	
2015-06-18 12:00:00	-0.1	0.0	0.2	-0.1	0.1	0.0	0.0	0.1	
2015-04-17 06:00:00	0.5	0.6	1.4	1.0	0.9	1.2	15.8	0.6	
2014-11-18 03:00:00	0.4	7.5	0.1	0.1	0.0	0.1	-0.1	0.2	
2014-11-06 06:00:00	0.4	0.2	0.9	0.2	0.4	0.4	0.3	0.3	
2014-09-11 15:00:00	-0.3	-0.3	-0.5	-0.4	-0.4	12.8	-0.3	-0.4	
2014-09-11 12:00:00	0.6	0.6	0.4	0.5	0.5	-12.7	0.6	0.7	
2014-07-31 18:00:00	-4.5	-0.5	-0.5	-0.6	-0.6	-0.5	-0.8	-0.6	
2014-06-18 09:00:00	-7.8	0.3	0.5	0.7	0.7	1.0	0.3	0.9	
2014-06-05 18:00:00	-0.4	-0.6	-0.7	-0.7	-0.6	-0.8	-0.2	-0.4	

	P_drift#V_Volochek	P_drift#Staritsa	P_drift#Kashyn	P_drift#Tver	P_drift#Klin	P_drift#Dmitrov	P_drift#Volokolamsk	P_drift#Mozhaisk	P_drift#
--	--------------------	------------------	----------------	--------------	--------------	-----------------	---------------------	------------------	----------

2014-04-30 00:00:00	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6
2014-04-10 18:00:00	-0.1	0.0	-0.1	0.0	0.1	0.2	0.0	0.0	0.3
2014-04-10 15:00:00	-0.2	0.0	0.2	0.3	0.2	0.2	0.0	0.0	0.3
2014-02-24 06:00:00	0.9	1.0	1.1	1.0	1.0	1.2	0.9	0.9	1.1
2013-09-22 18:00:00	-1.2	-0.6	-0.7	-0.9	-0.9	-0.5	-0.7	-0.7	-0.7

Выбросы по параметру df_Humid

Out[65]:	Humid#V_Volochek	Humid#Staritsa	Humid#Kashyn	Humid#Tver	Humid#Klin	Humid#Dmitrov	Humid#Volokolamsk	Humid#Mozhaisk	Humid#
	2021-01-24 12:00:00	100.0	100.0	100.0	97.0	100.0	100.0	100.0	100.0
	2021-01-24 06:00:00	100.0	100.0	100.0	100.0	100.0	100.0	99.0	100.0
	2017-05-15 12:00:00	63.0	25.0	26.0	25.0	24.0	27.0	27.0	26.0
	2014-12-27 03:00:00	17.0	89.0	89.0	87.0	88.0	92.0	93.0	90.0
	2014-04-10 15:00:00	28.0	29.0	31.0	28.0	29.0	31.0	31.0	25.0

Выбросы по параметру df_Wind_dir360

Out[65]:

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2022-05-10 18:00:00	22.5	0.0	22.5	0.0	337.5	0.0	
2022-04-13 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	
2022-04-03 03:00:00	0.0	0.0	22.5	0.0	0.0	0.0	
2022-03-24 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2022-03-08 06:00:00	0.0	0.0	22.5	0.0	0.0	0.0	
2022-03-04 18:00:00	0.0	0.0	0.0	0.0	337.5	0.0	
2022-03-04 15:00:00	315.0	0.0	0.0	22.5	0.0	0.0	
2022-02-25 15:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2022-02-19 12:00:00	157.5	180.0	180.0	180.0	180.0	180.0	
2022-01-20 18:00:00	180.0	202.5	180.0	180.0	180.0	180.0	
2022-01-11 12:00:00	0.0	0.0	22.5	0.0	0.0	0.0	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2021-07-11 06:00:00	360.0	360.0	360.0	45.0	360.0	360.0	
2021-07-07 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2021-07-06 03:00:00	360.0	360.0	360.0	360.0	360.0	90.0	
2021-06-27 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2021-06-27 00:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2021-06-17 06:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2021-06-07 06:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2021-06-01 06:00:00	360.0	337.5	360.0	360.0	360.0	360.0	
2021-06-01 00:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2021-05-30 03:00:00	0.0	0.0	360.0	0.0	0.0	0.0	
2021-05-23 12:00:00	157.5	180.0	180.0	180.0	180.0	180.0	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2021-05-11 06:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2021-05-09 09:00:00	0.0	337.5	337.5	337.5	337.5	337.5	337.5
2021-05-06 12:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2021-04-25 09:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2021-04-10 09:00:00	157.5	180.0	180.0	180.0	180.0	180.0	180.0
2021-04-05 21:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2021-03-28 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2021-03-26 06:00:00	360.0	247.5	360.0	360.0	360.0	360.0	360.0
2021-03-22 00:00:00	225.0	225.0	225.0	225.0	225.0	225.0	225.0
2021-03-15 09:00:00	157.5	180.0	180.0	180.0	180.0	180.0	180.0
2021-03-15 06:00:00	180.0	180.0	180.0	360.0	180.0	180.0	180.0

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2021-03-14 03:00:00	157.5	180.0	180.0	180.0	180.0	180.0	180.0
2021-03-14 00:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2021-03-13 21:00:00	157.5	180.0	180.0	180.0	180.0	180.0	180.0
2021-03-13 18:00:00	180.0	180.0	180.0	157.5	180.0	180.0	180.0
2021-03-13 15:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2021-03-13 12:00:00	180.0	202.5	180.0	180.0	180.0	180.0	180.0
2021-03-13 09:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2021-03-03 18:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2021-02-26 15:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2021-02-14 12:00:00	337.5	337.5	0.0	337.5	337.5	337.5	337.5
2021-01-25 12:00:00	157.5	157.5	157.5	157.5	180.0	157.5	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2020-04-13 09:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2020-03-31 12:00:00	0.0	0.0	22.5	0.0	22.5	0.0	
2020-03-31 03:00:00	0.0	0.0	0.0	0.0	22.5	0.0	
2020-03-30 03:00:00	337.5	0.0	0.0	0.0	0.0	0.0	
2020-03-30 00:00:00	0.0	0.0	22.5	0.0	22.5	0.0	
2020-03-23 00:00:00	360.0	22.5	22.5	0.0	22.5	22.5	
2020-03-01 18:00:00	180.0	202.5	180.0	180.0	180.0	180.0	
2020-03-01 15:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2020-03-01 09:00:00	157.5	180.0	180.0	180.0	180.0	180.0	
2020-02-23 03:00:00	180.0	202.5	180.0	180.0	180.0	180.0	
2020-02-11 12:00:00	180.0	202.5	180.0	180.0	180.0	180.0	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2020-02-11 09:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2020-02-11 03:00:00	180.0	202.5	180.0	180.0	180.0	180.0	180.0
2020-02-11 00:00:00	180.0	202.5	180.0	180.0	180.0	180.0	180.0
2020-02-07 09:00:00	337.5	337.5	337.5	337.5	337.5	337.5	337.5
2020-02-04 09:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2020-02-02 09:00:00	180.0	180.0	180.0	180.0	180.0	180.0	157.5
2020-01-22 12:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2020-01-20 21:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2020-01-20 09:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2019-12-17 03:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2019-12-16 09:00:00	180.0	202.5	180.0	180.0	180.0	180.0	180.0

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2019-06-08 00:00:00	360.0	360.0	360.0	360.0	360.0	360.0	135.0
2019-06-07 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	67.5
2019-06-04 00:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2019-05-17 06:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2019-05-16 00:00:00	0.0	0.0	22.5	22.5	0.0	0.0	0.0
2019-04-13 03:00:00	360.0	0.0	0.0	0.0	22.5	22.5	0.0
2019-04-13 00:00:00	360.0	0.0	22.5	22.5	22.5	22.5	22.5
2019-04-04 03:00:00	360.0	360.0	0.0	360.0	360.0	360.0	360.0
2019-03-30 15:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2019-03-23 00:00:00	315.0	337.5	0.0	315.0	315.0	315.0	315.0
2019-03-22 15:00:00	292.5	315.0	315.0	315.0	315.0	315.0	315.0

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#Moscow
2018-11-20 00:00:00	0.0	22.5	0.0	0.0	0.0	0.0	0.0
2018-11-01 15:00:00	180.0	180.0	180.0	180.0	180.0	202.5	180.0
2018-10-30 03:00:00	90.0	90.0	90.0	90.0	90.0	90.0	90.0
2018-10-21 03:00:00	360.0	337.5	360.0	360.0	360.0	360.0	360.0
2018-10-03 06:00:00	157.5	180.0	180.0	180.0	180.0	180.0	180.0
2018-09-27 15:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2018-09-27 12:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2018-09-24 12:00:00	157.5	157.5	180.0	157.5	157.5	157.5	157.5
2018-09-22 18:00:00	157.5	180.0	180.0	180.0	180.0	180.0	180.0
2018-09-14 21:00:00	360.0	360.0	360.0	360.0	360.0	157.5	180.0
2018-09-08 03:00:00	360.0	360.0	360.0	360.0	360.0	90.0	180.0

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2018-03-14 06:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2018-03-13 21:00:00	180.0	180.0	202.5	180.0	180.0	180.0	180.0
2018-03-13 18:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2018-03-05 00:00:00	0.0	0.0	22.5	0.0	0.0	0.0	0.0
2018-02-27 00:00:00	360.0	360.0	67.5	360.0	360.0	360.0	360.0
2018-02-24 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2018-02-24 00:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2018-02-10 21:00:00	135.0	135.0	135.0	135.0	135.0	135.0	135.0
2018-02-07 21:00:00	180.0	180.0	180.0	180.0	180.0	180.0	180.0
2018-02-07 12:00:00	180.0	202.5	180.0	180.0	180.0	180.0	180.0
2018-02-04 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2017-10-03 12:00:00	157.5	180.0	180.0	180.0	180.0	180.0	180.0
2017-09-30 12:00:00	0.0	0.0	0.0	337.5	0.0	0.0	0.0
2017-09-24 00:00:00	360.0	360.0	360.0	360.0	360.0	112.5	
2017-09-22 06:00:00	90.0	90.0	90.0	90.0	67.5	90.0	
2017-09-21 21:00:00	90.0	90.0	90.0	90.0	90.0	360.0	
2017-09-21 12:00:00	90.0	90.0	90.0	90.0	45.0	90.0	
2017-09-12 15:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2017-09-06 18:00:00	90.0	90.0	90.0	90.0	90.0	90.0	
2017-09-02 18:00:00	360.0	0.0	22.5	0.0	22.5	22.5	
2017-08-29 09:00:00	337.5	0.0	0.0	0.0	0.0	22.5	
2017-08-29 06:00:00	337.5	0.0	0.0	0.0	0.0	0.0	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2017-03-27 09:00:00	315.0	337.5	315.0	315.0	315.0	315.0	315.0
2017-03-22 15:00:00	180.0	202.5	180.0	180.0	180.0	180.0	180.0
2017-03-15 09:00:00	180.0	180.0	180.0	202.5	180.0	180.0	180.0
2017-03-14 09:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2017-03-14 03:00:00	360.0	360.0	22.5	360.0	360.0	360.0	360.0
2017-03-13 21:00:00	360.0	202.5	360.0	360.0	360.0	360.0	360.0
2017-03-12 21:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2017-03-07 15:00:00	90.0	90.0	90.0	90.0	90.0	90.0	90.0
2017-03-07 12:00:00	90.0	90.0	90.0	90.0	90.0	90.0	90.0
2017-03-07 09:00:00	90.0	90.0	90.0	90.0	90.0	90.0	90.0
2017-03-05 00:00:00	180.0	180.0	202.5	180.0	180.0	180.0	180.0

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2016-10-04 15:00:00	0.0	22.5	0.0	0.0	0.0	0.0	0.0
2016-09-30 06:00:00	157.5	180.0	180.0	180.0	180.0	180.0	180.0
2016-09-27 06:00:00	22.5	0.0	0.0	0.0	0.0	0.0	0.0
2016-09-26 15:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2016-09-25 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	67.5
2016-09-17 06:00:00	337.5	0.0	337.5	360.0	337.5	337.5	337.5
2016-09-16 21:00:00	0.0	22.5	0.0	0.0	0.0	0.0	22.5
2016-09-15 03:00:00	315.0	315.0	315.0	315.0	315.0	315.0	315.0
2016-09-08 00:00:00	270.0	270.0	270.0	225.0	270.0	270.0	270.0
2015-07-04 15:00:00	270.0	270.0	270.0	270.0	270.0	270.0	270.0
2015-07-02 18:00:00	315.0	337.5	337.5	337.5	0.0	337.5	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2015-02-16 03:00:00	0.0	0.0	0.0	337.5	0.0	0.0	
2015-02-16 00:00:00	0.0	0.0	22.5	0.0	0.0	0.0	
2015-02-15 21:00:00	0.0	0.0	22.5	0.0	0.0	0.0	
2015-02-01 18:00:00	180.0	180.0	180.0	180.0	180.0	180.0	
2015-01-28 15:00:00	180.0	180.0	225.0	180.0	180.0	180.0	
2015-01-28 03:00:00	180.0	180.0	180.0	180.0	180.0	180.0	
2015-01-16 18:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2015-01-09 21:00:00	157.5	180.0	180.0	180.0	180.0	180.0	
2015-01-09 12:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2014-12-19 09:00:00	180.0	180.0	180.0	157.5	180.0	180.0	
2014-12-16 21:00:00	180.0	180.0	202.5	180.0	180.0	180.0	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2014-12-16 12:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2014-11-30 03:00:00	360.0	337.5	360.0	360.0	360.0	360.0	
2014-11-04 12:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2014-11-04 09:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2014-10-19 18:00:00	180.0	180.0	180.0	180.0	180.0	180.0	
2014-10-09 18:00:00	180.0	180.0	180.0	180.0	180.0	180.0	
2014-10-04 09:00:00	337.5	337.5	22.5	337.5	315.0	337.5	
2014-09-16 18:00:00	0.0	337.5	0.0	0.0	0.0	0.0	
2014-09-07 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2014-07-07 06:00:00	360.0	337.5	360.0	360.0	360.0	360.0	
2014-07-01 18:00:00	180.0	180.0	180.0	157.5	180.0	180.0	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2014-06-30 21:00:00	180.0	180.0	180.0	180.0	180.0	180.0	202.5
2014-06-11 18:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2014-05-24 03:00:00	360.0	360.0	0.0	360.0	360.0	360.0	360.0
2014-04-20 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2014-04-14 12:00:00	180.0	180.0	180.0	157.5	180.0	180.0	180.0
2014-04-12 03:00:00	180.0	180.0	202.5	180.0	180.0	180.0	180.0
2014-04-08 06:00:00	360.0	360.0	360.0	360.0	360.0	360.0	360.0
2014-03-29 06:00:00	22.5	0.0	0.0	0.0	0.0	0.0	0.0
2014-03-28 18:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2014-03-28 15:00:00	0.0	22.5	0.0	0.0	0.0	0.0	0.0
2014-03-21 09:00:00	180.0	180.0	180.0	180.0	157.5	180.0	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2014-02-02 12:00:00	180.0	180.0	202.5	180.0	180.0	180.0	180.0
2013-12-29 00:00:00	180.0	180.0	180.0	180.0	157.5	180.0	
2013-12-28 18:00:00	180.0	180.0	180.0	180.0	157.5	180.0	
2013-12-26 15:00:00	180.0	180.0	202.5	180.0	180.0	180.0	
2013-12-10 06:00:00	337.5	337.5	0.0	337.5	337.5	315.0	
2013-11-26 15:00:00	0.0	0.0	22.5	0.0	0.0	0.0	
2013-11-25 00:00:00	180.0	180.0	180.0	180.0	157.5	180.0	
2013-09-24 15:00:00	22.5	0.0	0.0	0.0	0.0	0.0	
2013-08-28 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2013-08-18 03:00:00	360.0	360.0	360.0	360.0	360.0	360.0	
2013-03-04 09:00:00	45.0	22.5	45.0	45.0	45.0	45.0	

	Wind_dir360#V_Volochek	Wind_dir360#Staritsa	Wind_dir360#Kashyn	Wind_dir360#Tver	Wind_dir360#Klin	Wind_dir360#Dmitrov	Wind_dir360#
2008-12-23 21:00:00	157.5	135.0	135.0	135.0	135.0	135.0	135.0
2008-09-26 21:00:00	360.0	360.0	360.0	360.0	360.0	360.0	0.0
2008-09-12 21:00:00	45.0	45.0	45.0	67.5	45.0	45.0	
2008-07-09 21:00:00	360.0	180.0	360.0	360.0	360.0	360.0	360.0
2008-06-11 09:00:00	225.0	225.0	225.0	225.0	225.0	225.0	202.5
2008-05-22 09:00:00	45.0	45.0	45.0	45.0	45.0	45.0	45.0
2008-01-28 15:00:00	337.5	337.5	0.0	337.5	337.5	337.5	337.5
2007-10-03 21:00:00	135.0	135.0	135.0	135.0	360.0	135.0	

Выбросы по параметру df_Wind_speed

Out[65]:

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2022-04-28 09:00:00	1.0	2.0	1.0	1.0	1.0	1.0	1.0
2022-04-22 09:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2022-03-24 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2022-02-09 15:00:00	2.0	3.0	2.0	2.0	2.0	2.0	2.0
2022-02-05 03:00:00	1.0	1.0	0.0	1.0	1.0	1.0	1.0
2022-01-04 09:00:00	1.0	1.0	1.0	1.0	0.0	0.0	1.0
2021-10-09 15:00:00	2.0	2.0	2.0	0.0	2.0	2.0	2.0
2021-10-01 15:00:00	2.0	1.0	1.0	1.0	1.0	1.0	1.0
2021-09-17 09:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2021-08-11 03:00:00	1.0	1.0	1.0	1.0	1.0	1.0	2.0
2021-08-09 06:00:00	1.0	0.0	0.0	0.0	0.0	0.0	0.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2021-08-05 06:00:00	0.0	0.0	0.0	0.0	1.0	0.0	
2021-07-27 15:00:00	1.0	1.0	1.0	1.0	1.0	1.0	
2021-07-27 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	
2021-07-27 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	
2021-07-26 09:00:00	1.0	1.0	1.0	1.0	1.0	1.0	
2021-07-19 06:00:00	1.0	1.0	0.0	1.0	1.0	1.0	
2021-07-18 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	
2021-07-11 06:00:00	0.0	0.0	0.0	1.0	0.0	0.0	
2021-07-07 09:00:00	1.0	1.0	0.0	1.0	1.0	1.0	
2021-07-07 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	
2021-07-06 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2021-06-27 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2021-06-27 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2021-06-01 06:00:00	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2021-06-01 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2021-05-11 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2021-03-28 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2021-03-26 06:00:00	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2021-01-03 15:00:00	1.0	1.0	1.0	1.0	0.0	1.0	
2020-11-25 18:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2020-10-10 12:00:00	1.0	3.0	1.0	1.0	1.0	1.0	1.0
2020-10-10 06:00:00	1.0	1.0	1.0	1.0	1.0	0.0	1.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2020-09-23 15:00:00	2.0	2.0	2.0	1.0	2.0	2.0	2.0
2020-09-23 12:00:00	2.0	3.0	2.0	2.0	2.0	2.0	2.0
2020-09-16 09:00:00	0.0	1.0	1.0	1.0	1.0	1.0	1.0
2020-08-25 12:00:00	2.0	2.0	2.0	2.0	2.0	2.0	3.0
2020-08-22 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-08-21 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-08-01 09:00:00	1.0	3.0	1.0	1.0	1.0	1.0	1.0
2020-07-23 21:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2020-06-27 09:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2020-06-23 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	2.0
2020-06-08 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	1.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2020-05-26 06:00:00	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2020-02-08 15:00:00	1.0	2.0	1.0	1.0	1.0	1.0	1.0
2020-01-07 09:00:00	2.0	2.0	2.0	1.0	2.0	2.0	2.0
2019-12-26 06:00:00	1.0	1.0	1.0	1.0	0.0	0.0	1.0
2019-11-22 21:00:00	2.0	2.0	2.0	3.0	2.0	2.0	2.0
2019-10-19 03:00:00	1.0	1.0	1.0	0.0	1.0	1.0	1.0
2019-09-12 21:00:00	2.0	2.0	2.0	1.0	2.0	2.0	2.0
2019-09-10 18:00:00	1.0	1.0	1.0	1.0	0.0	0.0	1.0
2019-09-09 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-09-08 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2019-09-08 15:00:00	1.0	2.0	1.0	1.0	1.0	1.0	1.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2019-08-31 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2019-08-31 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-08-30 12:00:00	1.0	2.0	1.0	1.0	1.0	1.0	1.0
2019-08-28 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-07-16 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2019-07-16 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-06-09 09:00:00	1.0	2.0	1.0	1.0	1.0	1.0	1.0
2019-06-08 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	3.0
2019-06-07 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2019-06-06 09:00:00	1.0	1.0	1.0	0.0	1.0	1.0	1.0
2019-06-05 15:00:00	2.0	2.0	2.0	2.0	2.0	2.0	3.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2019-06-04 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-05-17 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-01-09 09:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2018-12-27 21:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2018-11-03 15:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2018-10-21 18:00:00	1.0	1.0	1.0	0.0	1.0	1.0	1.0
2018-10-20 00:00:00	2.0	1.0	1.0	1.0	1.0	1.0	1.0
2018-09-14 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2018-09-08 15:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2018-09-08 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2018-09-02 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2018-08-18 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-08-17 06:00:00	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2018-08-02 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-08-01 06:00:00	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2018-08-01 03:00:00	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2018-07-20 00:00:00	2.0	4.0	2.0	2.0	2.0	2.0	2.0
2018-07-14 09:00:00	1.0	1.0	1.0	2.0	1.0	1.0	1.0
2018-07-14 06:00:00	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2018-07-13 00:00:00	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2018-06-27 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-06-26 12:00:00	1.0	1.0	1.0	1.0	1.0	1.0	2.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2018-06-26 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-06-18 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-05-15 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2018-05-14 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-04-24 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-04-21 03:00:00	3.0	2.0	2.0	2.0	2.0	2.0	2.0
2018-03-21 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	2.0
2018-02-24 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-02-24 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-01-29 21:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2018-01-14 12:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2018-01-13 03:00:00	1.0	1.0	1.0	1.0	0.0	1.0	
2017-12-18 06:00:00	1.0	1.0	1.0	0.0	1.0	1.0	
2017-12-15 09:00:00	3.0	4.0	3.0	3.0	3.0	3.0	
2017-11-07 18:00:00	1.0	1.0	1.0	1.0	1.0	1.0	
2017-10-12 09:00:00	1.0	1.0	1.0	0.0	1.0	1.0	
2017-10-11 09:00:00	3.0	4.0	3.0	3.0	3.0	3.0	
2017-09-24 00:00:00	0.0	0.0	0.0	0.0	0.0	3.0	
2017-08-09 09:00:00	1.0	1.0	1.0	0.0	1.0	1.0	
2017-07-22 15:00:00	1.0	1.0	1.0	1.0	0.0	1.0	
2017-07-22 06:00:00	0.0	0.0	1.0	0.0	0.0	0.0	
2017-07-20 09:00:00	1.0	1.0	1.0	1.0	1.0	1.0	

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2017-06-24 09:00:00	1.0	1.0	1.0	1.0	1.0	1.0	2.0
2017-06-17 09:00:00	1.0	1.0	2.0	1.0	1.0	1.0	1.0
2017-06-07 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-03-14 09:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-03-14 03:00:00	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2017-03-13 21:00:00	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2017-03-12 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-02-03 21:00:00	2.0	2.0	2.0	2.0	2.0	2.0	3.0
2016-12-26 00:00:00	2.0	2.0	2.0	1.0	2.0	2.0	2.0
2016-10-19 15:00:00	1.0	1.0	1.0	0.0	1.0	1.0	1.0
2016-10-16 03:00:00	1.0	3.0	1.0	1.0	1.0	1.0	1.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2016-09-30 00:00:00	2.0	2.0	2.0	1.0	2.0	2.0	2.0
2016-09-23 18:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2016-08-13 03:00:00	0.0	1.0	1.0	1.0	1.0	1.0	1.0
2016-08-06 03:00:00	0.0	1.0	1.0	1.0	1.0	1.0	1.0
2015-06-24 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2015-05-25 09:00:00	1.0	2.0	1.0	1.0	1.0	1.0	1.0
2015-05-20 15:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2015-05-08 09:00:00	2.0	2.0	1.0	2.0	2.0	2.0	2.0
2015-03-19 12:00:00	1.0	1.0	2.0	1.0	1.0	1.0	1.0
2015-03-19 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2015-03-18 21:00:00	0.0	0.0	1.0	0.0	0.0	0.0	0.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2015-03-11 21:00:00	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2015-03-11 18:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2015-01-27 21:00:00	2.0	2.0	2.0	0.0	2.0	2.0	
2014-12-31 00:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-12-30 21:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-12-10 06:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-11-21 12:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-10-27 06:00:00	2.0	2.0	2.0	2.0	2.0	2.0	4.0
2014-10-25 12:00:00	1.0	2.0	1.0	1.0	1.0	1.0	1.0
2014-09-07 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2014-09-06 15:00:00	2.0	2.0	2.0	2.0	2.0	2.0	3.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2014-08-16 03:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-07-31 09:00:00	1.0	2.0	1.0	1.0	1.0	1.0	1.0
2014-07-26 09:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2014-07-25 21:00:00	1.0	1.0	1.0	0.0	1.0	1.0	1.0
2014-07-14 09:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2014-07-07 06:00:00	0.0	2.0	0.0	0.0	0.0	0.0	0.0
2014-05-24 09:00:00	1.0	1.0	1.0	1.0	1.0	1.0	0.0
2014-05-24 03:00:00	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2014-05-11 21:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-04-20 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2014-04-08 06:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2014-03-05 03:00:00	1.0	3.0	1.0	1.0	1.0	1.0	1.0
2014-03-03 06:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-02-21 03:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-02-12 06:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2014-02-07 18:00:00	1.0	2.0	2.0	2.0	2.0	2.0	2.0
2014-02-05 21:00:00	2.0	2.0	2.0	1.0	2.0	2.0	2.0
2014-01-26 00:00:00	1.0	0.0	1.0	1.0	1.0	1.0	1.0
2014-01-19 18:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-01-15 06:00:00	2.0	1.0	1.0	1.0	1.0	1.0	1.0
2014-01-07 03:00:00	1.0	1.0	1.0	2.0	1.0	1.0	1.0
2013-10-08 09:00:00	2.0	2.0	2.0	1.0	2.0	2.0	2.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2013-08-28 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2013-08-18 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2013-08-17 18:00:00	2.0	1.0	1.0	1.0	1.0	1.0	1.0
2013-07-05 15:00:00	1.0	2.0	2.0	2.0	2.0	2.0	2.0
2013-06-29 09:00:00	1.0	1.0	1.0	1.0	0.0	0.0	1.0
2013-06-27 09:00:00	1.0	0.0	1.0	1.0	1.0	1.0	1.0
2013-06-17 03:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2013-06-09 09:00:00	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2013-04-29 15:00:00	2.0	3.0	2.0	2.0	2.0	2.0	2.0
2013-04-28 09:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2013-01-06 09:00:00	2.0	4.0	2.0	2.0	2.0	2.0	2.0

	Wind_speed#V_Volochek	Wind_speed#Staritsa	Wind_speed#Kashyn	Wind_speed#Tver	Wind_speed#Klin	Wind_speed#Dmitrov	Wind_speed#Vol
2012-11-23 21:00:00	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2012-08-26 15:00:00	2.0	2.0	2.0	2.0	1.0	2.0	
2012-04-22 21:00:00	1.0	1.0	1.0	1.0	1.0	3.0	
2011-07-11 09:00:00	1.0	1.0	1.0	1.0	1.0	1.0	
2011-06-11 09:00:00	2.0	3.0	2.0	2.0	2.0	2.0	
2010-12-02 15:00:00	1.0	1.0	1.0	0.0	1.0	1.0	
2010-10-05 09:00:00	1.0	0.0	0.0	0.0	0.0	0.0	
2010-09-13 21:00:00	1.0	0.0	0.0	0.0	0.0	0.0	
2010-08-07 09:00:00	1.0	0.0	0.0	0.0	0.0	0.0	
2010-04-05 21:00:00	0.0	0.0	0.0	0.0	0.0	1.0	
2010-02-16 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0	

Wind_speed#V_Volochek Wind_speed#Staritsa Wind_speed#Kashyn Wind_speed#Tver Wind_speed#Klin Wind_speed#Dmitrov Wind_speed#Vol

2010-01-19 21:00:00	0.0	0.0	2.0	0.0	0.0	0.0
2009-09-01 09:00:00	1.0	1.0	1.0	1.0	0.0	1.0
2008-07-09 21:00:00	0.0	2.0	0.0	0.0	0.0	0.0
2007-09-28 15:00:00	2.0	2.0	2.0	2.0	2.0	2.0

Выбросы по параметру df_Gusts

Out[65]: Gусты#V_Volochek Gусты#Staritsa Gусты#Kashyn Gусты#Tver Gусты#Klin Gусты#Dmitrov Gусты#Volokolamsk Gусты#Mozhaisk Gусты#N_Jerusalem Gусты#Vol

Выбросы по параметру df_Gusts_3h

Out[65]: Gусты_3h#V_Volochek Gусты_3h#Staritsa Gусты_3h#Kashyn Gусты_3h#Tver Gусты_3h#Klin Gусты_3h#Dmitrov Gусты_3h#Volokolamsk Gусты_3h#Mozhaisk

Выбросы по параметру df_Depo_diam_mm

Out[65]:

	Depo_diam_mm#V_Volochek	Depo_diam_mm#Staritsa	Depo_diam_mm#Kashyn	Depo_diam_mm#Tver	Depo_diam_mm#Klin	Depo_diam_mm#Dm
2022-02-23 12:00:00	0.0	0.0	0.0	0.0	0.0	0.0
2022-02-23 09:00:00	0.0	0.0	0.0	0.0	0.0	0.0
2022-01-25 21:00:00	0.0	0.0	0.0	0.0	0.0	1.0
2022-01-14 03:00:00	0.0	0.0	0.0	0.0	0.0	0.0
2022-01-13 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0
...
2013-02-05 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0
2013-02-05 15:00:00	0.0	0.0	0.0	0.0	0.0	0.0
2013-02-05 09:00:00	0.0	0.0	0.0	0.0	0.0	0.0
2013-02-04 21:00:00	0.0	0.0	0.0	0.0	0.0	0.0
2013-02-03 09:00:00	0.0	2.0	0.0	0.0	0.0	0.0

1030 rows × 13 columns

Выбросы по параметру df_T_min

Out[65]: T_min#V_Volochek T_min#Staritsa T_min#Kashyn T_min#Tver T_min#Klin T_min#Dmitrov T_min#Volokolamsk T_min#Mozhaisk T_min#N_Jerusalem

Выбросы по параметру df_T_max

Out[65]: T_max#V_Volochek T_max#Staritsa T_max#Kashyn T_max#Tver T_max#Klin T_max#Dmitrov T_max#Volokolamsk T_max#Mozhaisk T_max#N

2013- 04-17 21:00:00	17.4	16.7	15.4	17.5	17.9	17.3	18.2	18.4
-------------------------------------	------	------	------	------	------	------	------	------

2010- 06-25 21:00:00	31.5	30.5	31.8	33.6	3.3	33.3	31.9	32.4
-------------------------------------	------	------	------	------	-----	------	------	------

2009- 12-15 21:00:00	-18.5	-19.2	-21.5	-22.4	-21.8	-22.6	-21.7	-20.8
-------------------------------------	-------	-------	-------	-------	-------	-------	-------	-------

Выбросы по параметру df_Visibility

Out[65]:

	Visibility#V_Volochev	Visibility#Staritsa	Visibility#Kashyn	Visibility#Tver	Visibility#Klin	Visibility#Dmitrov	Visibility#Volokolamsk	Visibility#Moscow
2022-05-13 21:00:00	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
2022-05-10 21:00:00	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
2022-05-07 21:00:00	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
2022-05-03 21:00:00	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
2022-04-30 21:00:00	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
...
2013-05-10 03:00:00	10.0	10.0	50.0	10.0	10.0	10.0	10.0	10.0
2013-03-14 03:00:00	10.0	10.0	50.0	10.0	10.0	10.0	10.0	10.0
2013-02-22 09:00:00	10.0	10.0	50.0	10.0	10.0	10.0	10.0	10.0
2012-01-18 21:00:00	10.0	10.0	30.0	10.0	10.0	10.0	10.0	10.0
2008-03-25 09:00:00	20.0	0.5	0.5	0.5	0.5	0.8	0.2	

2985 rows × 13 columns

Выбросы по параметру df_Dew_point

Out[65]: Dew_point#V_Volochek Dew_point#Staritsa Dew_point#Kashyn Dew_point#Tver Dew_point#Klin Dew_point#Dmitrov Dew_point#Volokolamsk Dew_point#Luzhki Dew_point#Bogorodsk Dew_point#Kolomna Dew_point#Pereslavl

Выбросы по параметру df_Prcpttn

Out[65]:

	Prcpttn#V_Volocheok	Prcpttn#Staritsa	Prcpttn#Kashyn	Prcpttn#Tver	Prcpttn#Klin	Prcpttn#Dmitrov	Prcpttn#Volokolamsk	Prcpttn#Mozhaisk	F
2022-05-30 09:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2022-05-26 09:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2022-05-24 18:00:00	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0
2022-05-24 09:00:00	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0
2022-05-19 06:00:00	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0
...
2009-10-30 21:00:00	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
2009-04-20 21:00:00	0.5	0.5	1.0	0.5	0.5	0.5	0.5	0.5	0.5
2009-04-02 21:00:00	0.5	0.5	0.5	0.5	0.5	0.3	0.5	0.5	0.5
2009-02-17 09:00:00	5.0	3.0	1.0	2.0	63.0	2.0	2.0	0.4	
2008-04-19 09:00:00	0.3	9.0	4.0	2.0	9.0	7.0	5.0	2.0	

562 rows × 13 columns

Выбросы по параметру df_Prcptn_tdelt


```
Prcptn_tdelt#V_Volocheok Prcptn_tdelt#Staritsa Prcptn_tdelt#Kashyn Prcptn_tdelt#Tver Prcptn_tdelt#Klin Prcptn_tdelt#Dmitrov Prcptn_tde
```

2008-08-25 21:00:00	12.0	12.0	12.0	12.0	12.0	12.0	12.0
------------------------	------	------	------	------	------	------	------

2008-01-20 09:00:00	12.0	12.0	12.0	12.0	24.0	12.0
------------------------	------	------	------	------	------	------

Выбросы по параметру df_Soil_T

```
Out[65]: Soil_T#V_Volocheok Soil_T#Staritsa Soil_T#Kashyn Soil_T#Tver Soil_T#Klin Soil_T#Dmitrov Soil_T#Volokolamsk Soil_T#Mozhaisk Soil_T#N_Jeru
```

2019-08-15 09:00:00	13.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
------------------------	------	------	------	------	------	------	------	------

Выбросы по параметру df_Snow_height

```
Out[65]: Snow height#V_Volocheok Snow height#Staritsa Snow height#Kashyn Snow height#Tver Snow height#Klin Snow height#Dmitrov Snow height#Voloko
```

```
In [66]: # Выведем максимум выбросов по параметрам на момент наблюдения
list_max_outliers
```

```
Out[66]: [('df_T', 1),
('df_P_station', 1),
('df_P_sea', 1),
('df_P_drift', 1),
('df_Humid', 1),
('df_Wind_dir360', 1),
('df_Wind_speed', 1),
('df_Gusts', nan),
('df_Gusts_3h', nan),
('df_Depo_diam_mm', 1),
('df_T_min', nan),
('df_T_max', 1),
('df_Visibility', 1),
('df_Dew_point', nan),
('df_Prcptn', 1),
('df_Prcptn_tdelt', 1),
('df_Soil_T', 1),
('df_Snow_height', nan)]
```

ВАЖНО

Количество выбросов по одному параметру на момент наблюдения по всем станциям нигде не превышает единицы

1.2.3. Поиск максимальных отклонений от средней

Определим функции, которые могут понадобиться

```
In [67]: # Определим функцию для оценки вхождения значения в диапазон множитель*сигма
def sigma_assess(value_, mean_, std_, mult_):
    """Принимает значение, среднюю, сигму и множитель для сигмы
    оценивает, находится ли значение вне сигмы*множитель
    выводит boolean"""
    lower_ = mean_ - mult_*std_
    upper_ = mean_ + mult_*std_
    if (value_ < lower_) or (upper_ < value_):
        return True
    else:
        return False

# Определим функцию для оценки индивидуального отклонения признака в отношении к средней.
def individ_variance(value_, mean_):
    """Принимает значение и среднюю,
    выводит частное от индивидуального абсолютного отклонения и средней"""
    return (abs(value_-mean_)) / mean_ if mean_ != 0 else (abs(value_+1-mean_)) / (0+1*len(dict_stations.keys()))
    # Если знаменатель - mean_ окажется равным нулю, сдвинем значение на 1 и изменим среднюю
    # (если к каждому значению прибавить k, то средняя увеличится на k*n)
```

Определим пределы, в количествах сигм, в которых находятся реальные значения

```
In [68]: # определим множитель для сигмы (очевидно, более 3, так у нас уже значения вне 3х сигм)
sigma_mult = list(np.arange(3., 3.4, 0.02)) # список множителей для сигмы

for multplr in sigma_mult:
    # применим функцию sigma_assess к соответствующим столбцам df_anomalies и выведем результат в отдельный столбец
    out_count = df_anomalies.assign(is_err = df_anomalies.apply(lambda x:
                                                                sigma_assess(x.a_value, x.mean_, x.std_, multplr),
                                                                axis = 1)).is_err.sum()

    print(f"При диапазоне {multplr:.2f}*сигма, количество выбросов равно {out_count}")
    if out_count == 0:
```

break

При диапазоне 3.00*сигма, количество выбросов равно 14380
При диапазоне 3.02*сигма, количество выбросов равно 13916
При диапазоне 3.04*сигма, количество выбросов равно 13483
При диапазоне 3.06*сигма, количество выбросов равно 13101
При диапазоне 3.08*сигма, количество выбросов равно 12159
При диапазоне 3.10*сигма, количество выбросов равно 11648
При диапазоне 3.12*сигма, количество выбросов равно 11287
При диапазоне 3.14*сигма, количество выбросов равно 8583
При диапазоне 3.16*сигма, количество выбросов равно 8373
При диапазоне 3.18*сигма, количество выбросов равно 6792
При диапазоне 3.20*сигма, количество выбросов равно 6577
При диапазоне 3.22*сигма, количество выбросов равно 4834
При диапазоне 3.24*сигма, количество выбросов равно 4447
При диапазоне 3.26*сигма, количество выбросов равно 4097
При диапазоне 3.28*сигма, количество выбросов равно 3979
При диапазоне 3.30*сигма, количество выбросов равно 3505
При диапазоне 3.32*сигма, количество выбросов равно 0

Вычислим индивидуальные относительные отклонения от средней для значений-выбросов ("Индивидуальный коэффициент вариации")

```
In [69]: # применим функцию individ_variance к соответствующим столбцам df_anomalies и выведем результат в отдельный столбец
df_anomalies = df_anomalies.assign(ind_var \
= df_anomalies.apply(lambda x: individ_variance(x.a_value, x.mean_), \
axis = 1))
df_anomalies.sample(3)
```

Out[69]:

	obsrvtn	station	parameter	a_value	mean_	std_	3sigma_down	3sigma_up	description	description3h1	description3h2	ind_var
1441	2013-05-28 12:00:00	Kashyn	Visibility	50.0	14.909091	11.484809	-19.545335	49.363516		NaN	NaN	2.353659
11577	2020-02-22 03:00:00	Kashyn	Visibility	50.0	13.333333	11.055416	-19.832915	46.499581		NaN	NaN	2.750000
304	2009-03-20 03:00:00	Staritsa	Wind_speed	3.0	0.272727	0.862439	-2.314591	2.860045	Снег непрерывный слабый в срок наблюдения.	Ливень (ливни).	Снег и/или другие виды твердых осадков	10.000000

Определим количество выбросов и максимальный индивидуальный коэффициент вариации для каждого параметра

In [70]:

```
# отсортируем df_anomalies по индивидуальному коэффициенту вариации количеству выбросов по каждому параметру и
# подсчитаем среднюю из индивидуальных коэффициентов вариации.

df_anomalies.groupby("parameter").agg({'a_value':'count', 'ind_var': 'max'})\
.rename(columns={"a_value": "values_count", "ind_var": "max_ind_relative_var"})\
.sort_values("values_count", ascending = False)
```

Out[70]:

values_count max_ind_relative_var

parameter	values_count	max_ind_relative_var
Visibility	11232	8.557945
Prcptn	1201	11.000000
Humid	423	2.471698
T	414	265.000000
Wind_speed	263	11.000000
Dew_point	252	325.000000
P_sea	220	0.865051
P_drift	150	164.000000
T_min	95	49.000000
T_max	68	9.745763
P_station	31	0.889407
Prcptn_tdelt	18	0.908257
Soil_T	6	0.558442
Snow_height	5	6.020537
Gusts_3h	2	0.726027

Посмотрим на максимальные значения вариации

In [71]: `df_anomalies[((df_anomalies.parameter == 'Prcptn') & (df_anomalies.ind_var >= 10.9))]`

Out[71]:

	obsrvtn	station	parameter	a_value	mean_	std_	3sigma_down	3sigma_up	description	description3h1	description3h2
655	2011-04-07 21:00:00	Nemchinovka	Prcpttn	960.0	80.483333	265.184394	-715.069850	876.036517	Морось незамерзающая непрерывная умеренная в с...	Морось.	Туман или ледяной туман или сильная мгла.
926	2012-09-05 21:00:00	Tver	Prcpttn	0.5	0.041667	0.138193	-0.372911	0.456245	Состояние неба в общем не изменилось.	Ливень (ливни).	Облака покрывали более половины неба в течение...
935	2012-09-20 09:00:00	V_Volochev	Prcpttn	0.8	0.066667	0.221108	-0.596658	0.729992	Облака в целом рассеиваются или становятся мен...	Ливень (ливни).	Облака покрывали более половины неба в течение...
940	2012-09-20 21:00:00	V_Volochev	Prcpttn	9.0	0.750000	2.487469	-6.712406	8.212406	Ливневый(ые) дождь(и) слабый(ые) в срок наблюд...	Ливень (ливни).	Облака покрывали более половины неба в течение...
951	2012-09-26 21:00:00	Kashyn	Prcpttn	0.4	0.033333	0.110554	-0.298329	0.364996	Состояние неба в общем не изменилось.	Дождь.	Морось.
...
14522	2022-05-19 06:00:00	Tver	Prcpttn	0.3	0.025000	0.082916	-0.223747	0.273747	Ливневый(ые) дождь(и) слабый(ые) в срок наблюд...	Ливень (ливни).	Облака покрывали более половины неба в течение...
14525	2022-05-24 09:00:00	Kashyn	Prcpttn	0.3	0.025000	0.082916	-0.223747	0.273747	Снег непрерывный слабый в срок наблюдения.	Дождь со снегом или другими видами твердых осад...	Облака покрывали более половины

obsrvtn	station	parameter	a_value	mean_	std_	3sigma_down	3sigma_up	description	description3h1	description3h2
неба в течение...										
14526	2022-05-24 18:00:00	Kashyn	Prcpttn	0.3	0.025000	0.082916	-0.223747	0.273747	NaN	NaN 1°
14527	2022-05-26 09:00:00	Naro_Fominsk	Prcpttn	0.5	0.041667	0.138193	-0.372911	0.456245	NaN	NaN 1°
14530	2022-05-30 09:00:00	Naro_Fominsk	Prcpttn	0.5	0.041667	0.138193	-0.372911	0.456245	Состояние неба в общем не изменилось.	Туман или ледяной туман или сильная мгла. Облаца покрывали половину неба или менее в теч...
100	100	100	100	100	100	100	100	100	100	100

1.3. Графическое представление корреляции значений для метеостанций по каждому параметру на основе "сырых" данных.

Для дальнейшего поиска и исправления ошибок, выбросов и NaN необходимо проверить корреляцию данных между метеостанциями по каждому параметру.

```
In [72]: # Восстановим данные о параметрах метеонаблюдений

# Создадим список файлов с архивами параметров
list_df_parameters_files = [file_name for file_name in listdir(raw_path2)
                            if isfile(join(raw_path2, file_name))]

# Заново запишем данные архивов параметров в словарь
for file_name in list_df_parameters_files: # По списку csv файлов
    name_df = file_name[:-4] # Вычиляем название датафрейма из названия файла
    file_path = raw_path2 + file_name # Формируем путь к файлу
    print(file_path, ' - ', end='') # КОНТРОЛЬ: Обрабатываемый файл

    # Создаём ключ (название DF) из названия файла
    # и записываем в словарь по этому ключу соответствующий DF
```

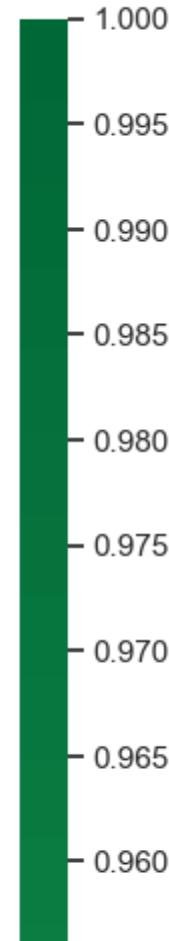
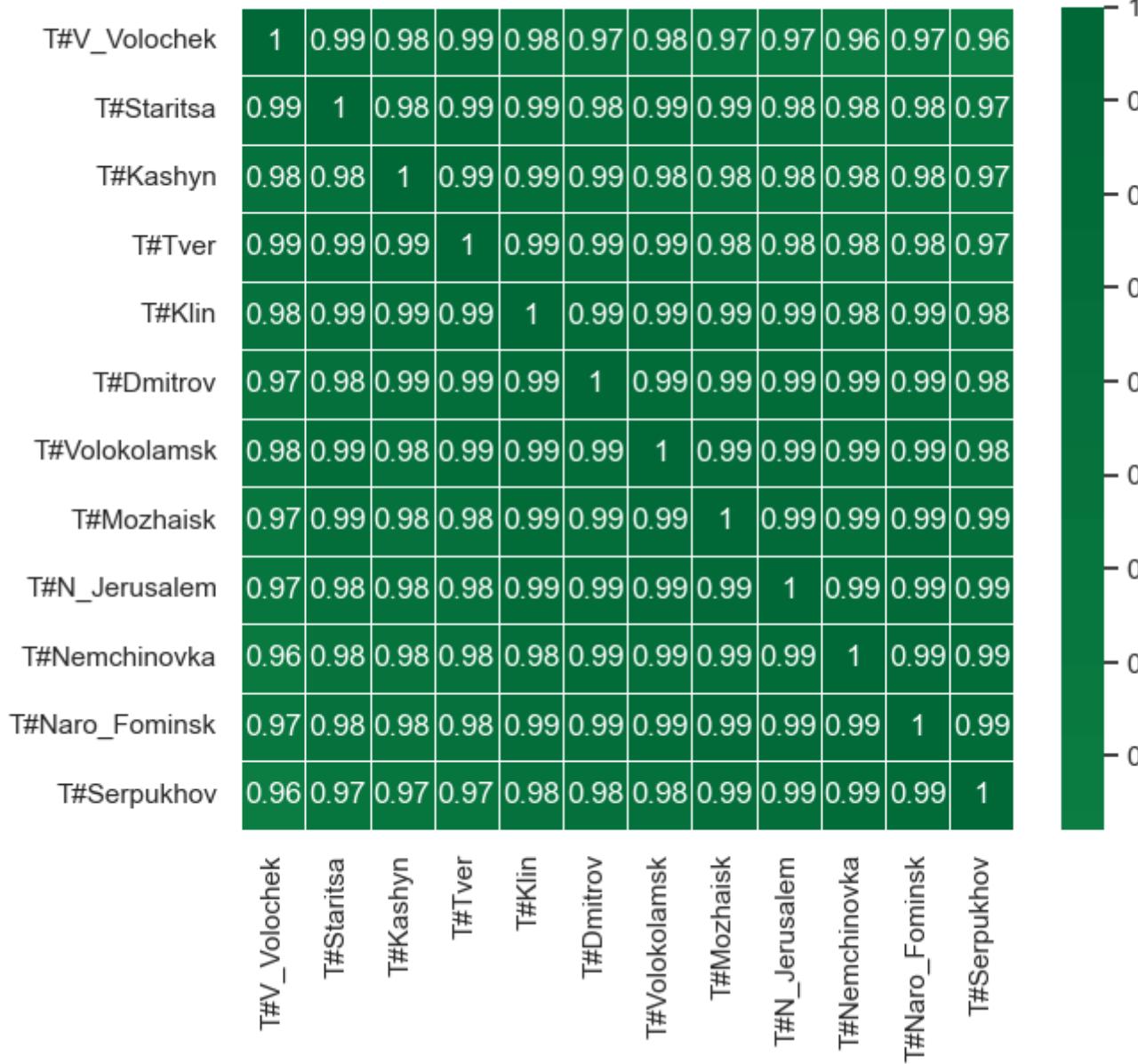
```
dict_df_parameters[f'{name_df}'] = pd.read_csv(
    file_path, index_col=0, parse_dates=True, infer_datetime_format = True, low_memory=False)
print('O.K.')
# Выводим полученные ключи словаря
dict_df_parameters.keys()

data/csv/raw/df_Cloudness.csv - O.K.
data/csv/raw/df_Cl_bottom.csv - O.K.
data/csv/raw/df_Cl_cirrus.csv - O.K.
data/csv/raw/df_Cl_Cumls.csv - O.K.
data/csv/raw/df_Cl_cumls_hi.csv - O.K.
data/csv/raw/df_Cl_viewd.csv - O.K.
data/csv/raw/df_Depo_diam_mm.csv - O.K.
data/csv/raw/df_Dew_point.csv - O.K.
data/csv/raw/df_Gusts.csv - O.K.
data/csv/raw/df_Gusts_3h.csv - O.K.
data/csv/raw/df_Humid.csv - O.K.
data/csv/raw/df_Prcpttn.csv - O.K.
data/csv/raw/df_Prcpttn_depo.csv - O.K.
data/csv/raw/df_Prcpttn_like.csv - O.K.
data/csv/raw/df_Prcpttn_tdelt.csv - O.K.
data/csv/raw/df_P_drift.csv - O.K.
data/csv/raw/df_P_sea.csv - O.K.
data/csv/raw/df_P_station.csv - O.K.
data/csv/raw/df_Snow_height.csv - O.K.
data/csv/raw/df_Soil.csv - O.K.
data/csv/raw/df_Soil_cover.csv - O.K.
data/csv/raw/df_Soil_T.csv - O.K.
data/csv/raw/df_T.csv - O.K.
data/csv/raw/df_T_max.csv - O.K.
data/csv/raw/df_T_min.csv - O.K.
data/csv/raw/df_Visibility.csv - O.K.
data/csv/raw/df_Wind_dir.csv - O.K.
data/csv/raw/df_Wind_dir360.csv - O.K.
data/csv/raw/df_Wind_dir6k.csv - O.K.
data/csv/raw/df_Wind_speed.csv - O.K.
data/csv/raw/df_Wthr_3h.csv - O.K.
data/csv/raw/df_Wthr_3h2.csv - O.K.
data/csv/raw/df_Wthr_curr.csv - O.K.

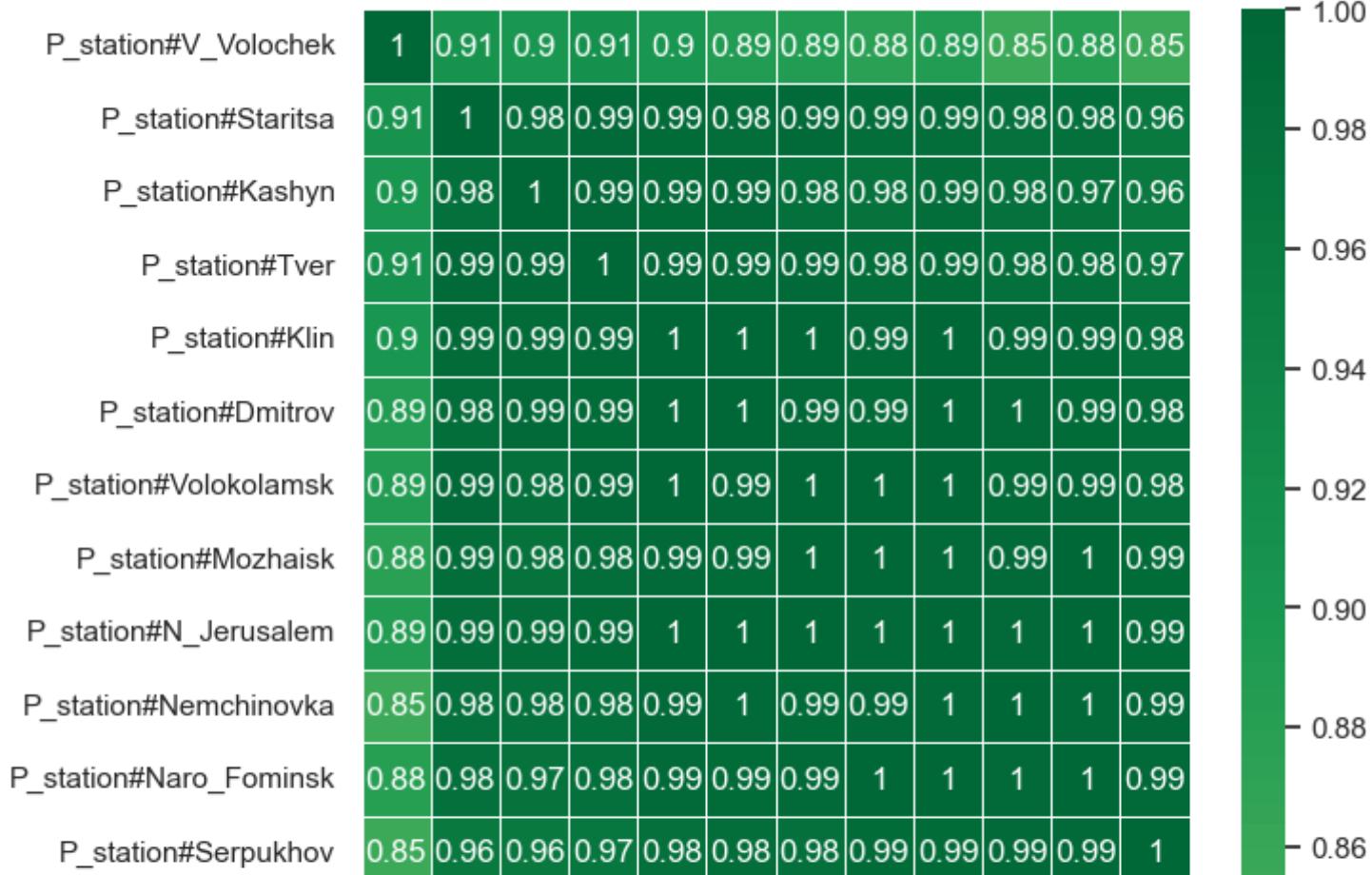
Out[72]: dict_keys(['df_T', 'df_P_station', 'df_P_sea', 'df_P_drift', 'df_Humid', 'df_Wind_dir', 'df_Wind_dir360', 'df_Wind_dir6k', 'df_Wind_speed', 'df_Gusts', 'df_Gusts_3h', 'df_Cloudness', 'df_Wthr_curr', 'df_Prcpttn_like', 'df_Prcpttn_depo', 'df_Depo_diam_mm', 'df_Wthr_3h', 'df_Wthr_3h2', 'df_T_min', 'df_T_max', 'df_Cl_Cumls', 'df_Cl_viewd', 'df_Cl_bottom', 'df_Cl_cumls_hi', 'df_Cl_cirrus', 'df_Visibility', 'df_Dew_point', 'df_Prcpttn', 'df_Prcpttn_tdelt', 'df_Soil', 'df_Soil_T', 'df_Soil_cover', 'df_Snow_height'])
```

```
In [73]: for param_name in list_parameters: # Проходим по списку параметров (не затрагивая категориальные данные)
    # Определяем матрицу корреляции из архива данного параметра
    df_corr_matrix = dict_df_parameters[f'df_{param_name}'].corr()
    fig, ax = plt.subplots(figsize=(7, 6))
    ax = sns.heatmap(data=df_corr_matrix, # Выводим график heatmap
                      cmap='RdYlGn', # matplotlib цветовая палитра (желательно расходящаяся)
                      center=0.5, # центральное значение (для расходящейся цветовой карты):
                                  # заметная связь по шкале Чеддока
                      annot=True, # определим подписи значений
                      linewidths=0.5 # толщина разделяющих линий между ячейками
                     )
    dummy = plt.suptitle(f'Корреляция параметра {param_name} между метеостанциями ')
    plt.show()
```

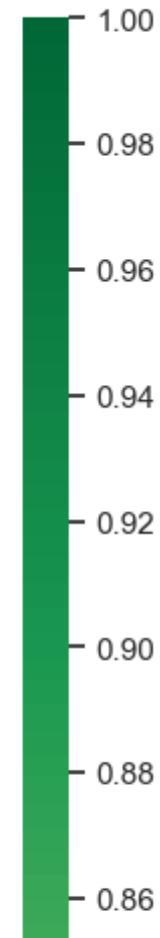
Корреляция параметра Т между метеостанциями



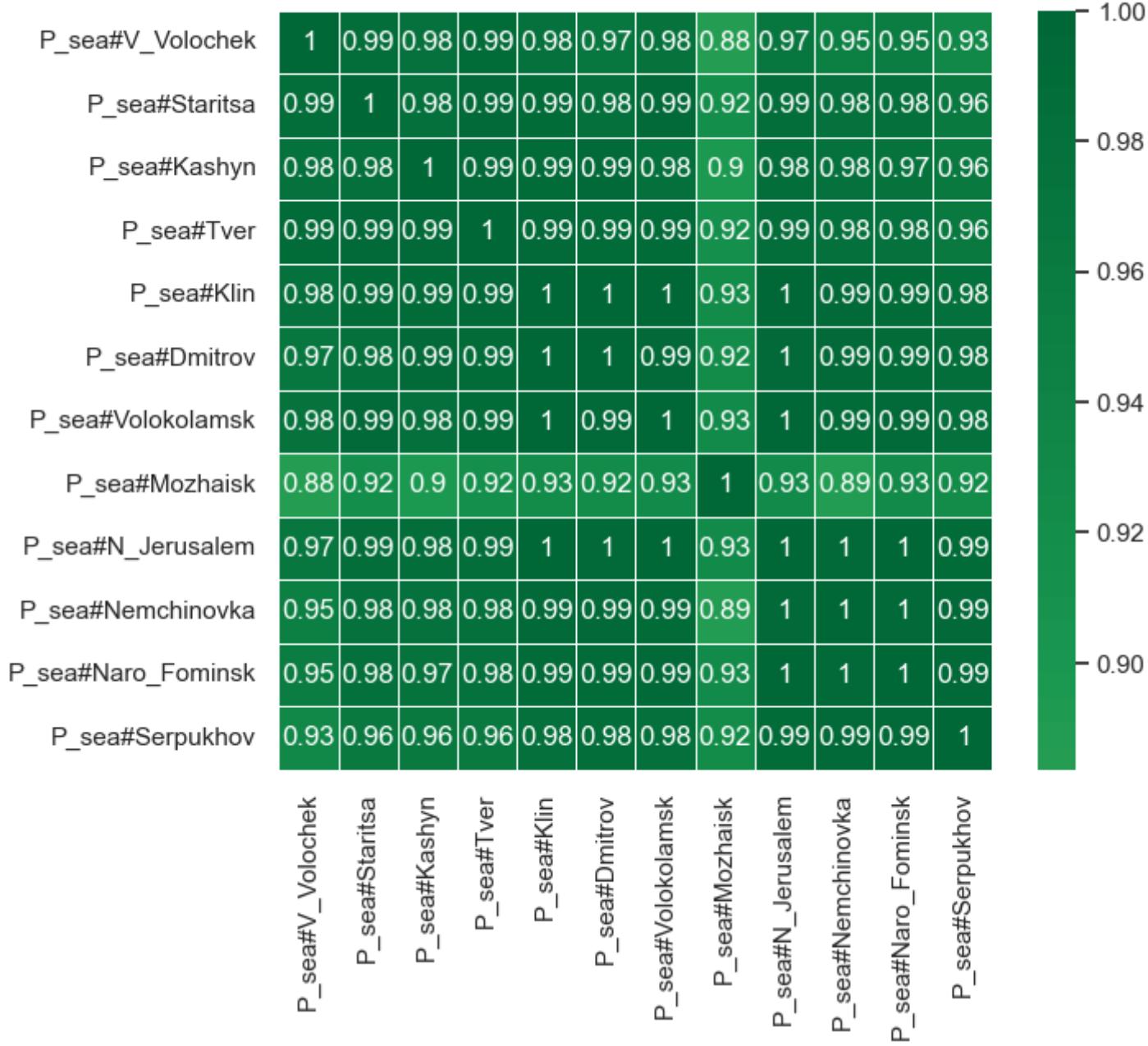
Корреляция параметра P_station между метеостанциями



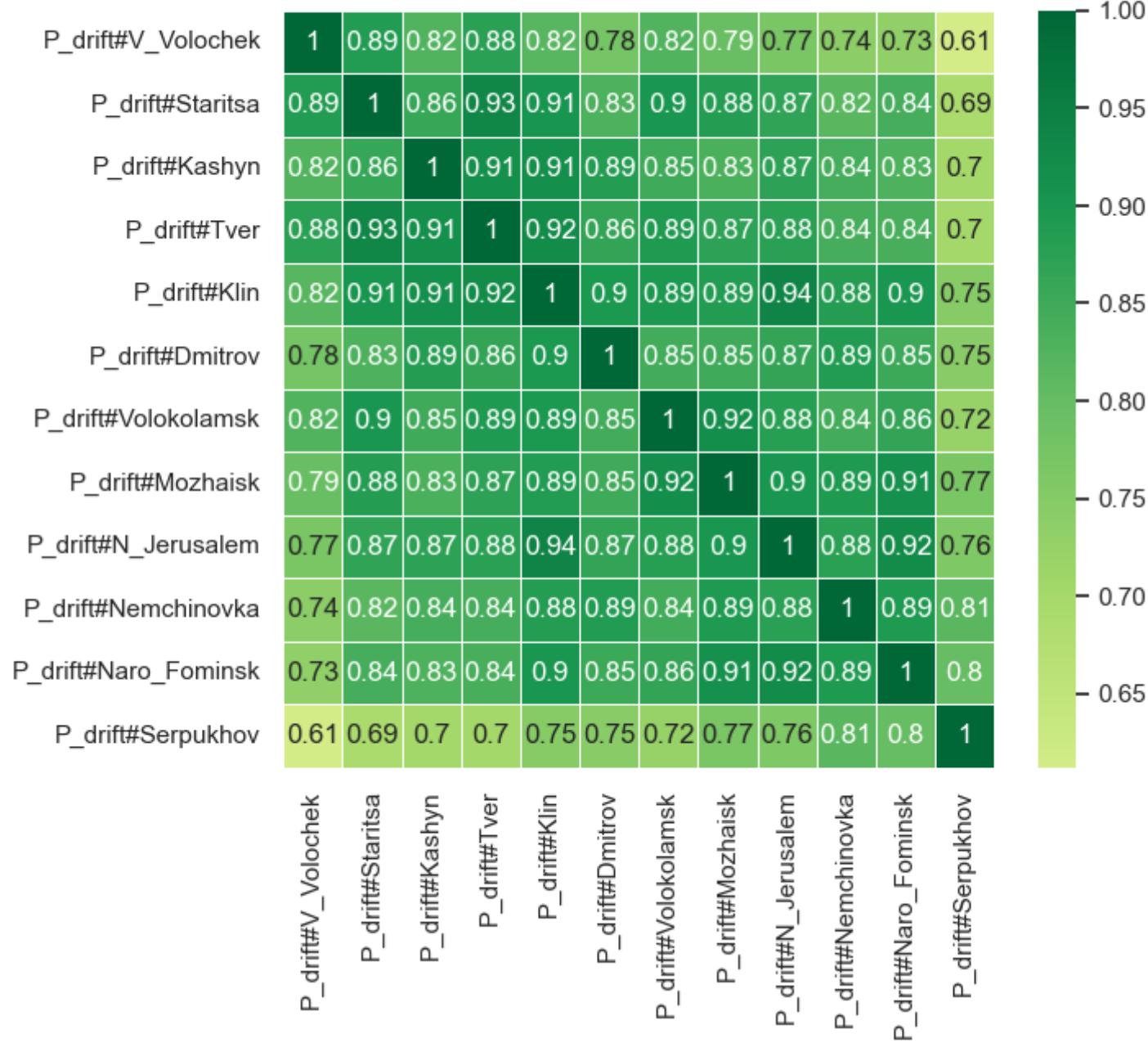
P_station#V_Volochek
 P_station#Staritsa
 P_station#Kashyn
 P_station#Tver
 P_station#Klin
 P_station#Dmitrov
 P_station#Volokolamsk
 P_station#Mozhaisk
 P_station#N_Jerusalem
 P_station#Nemchinovka
 P_station#Naro_Fominsk
 P_station#Serpukhov



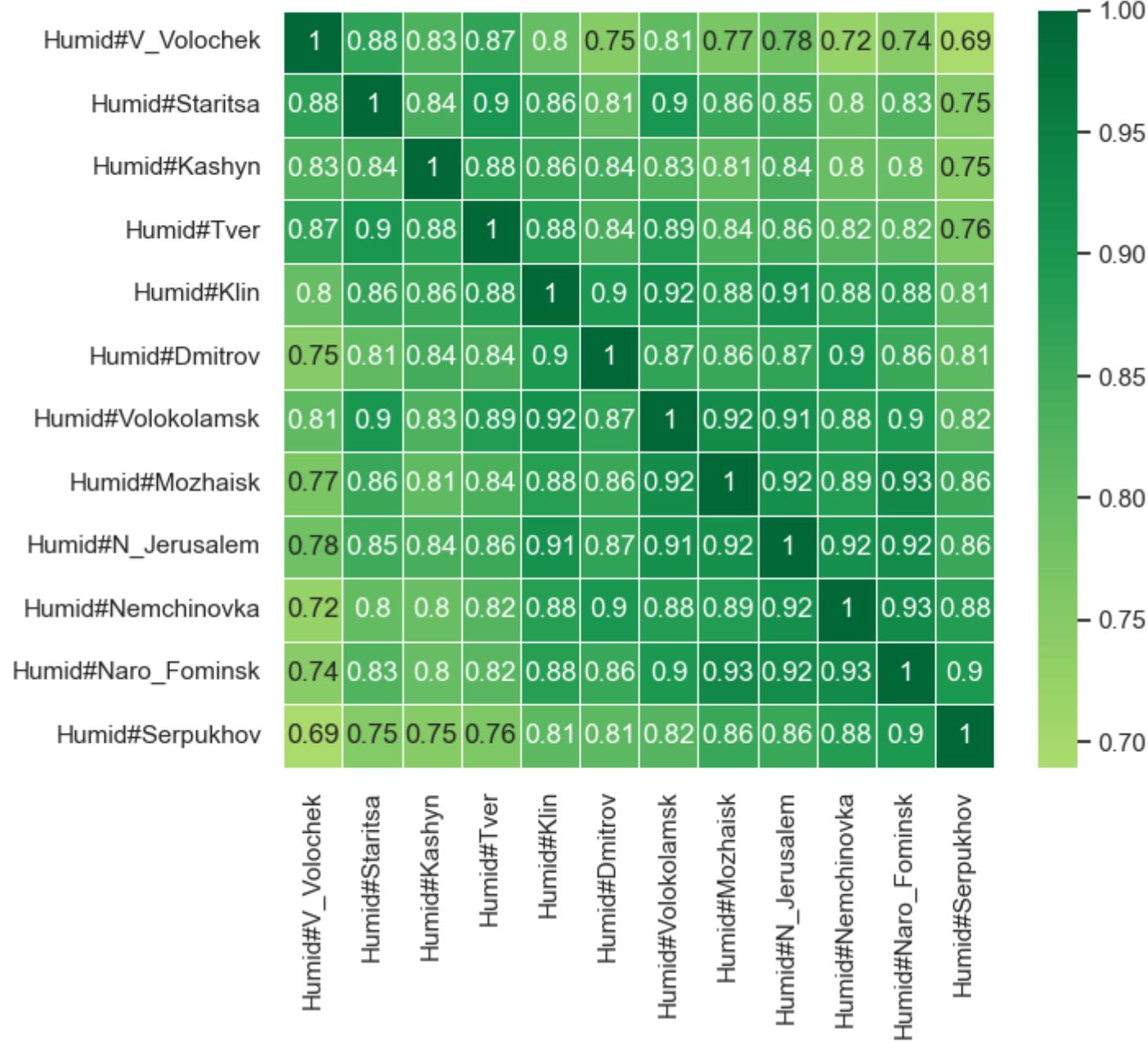
Корреляция параметра P_sea между метеостанциями



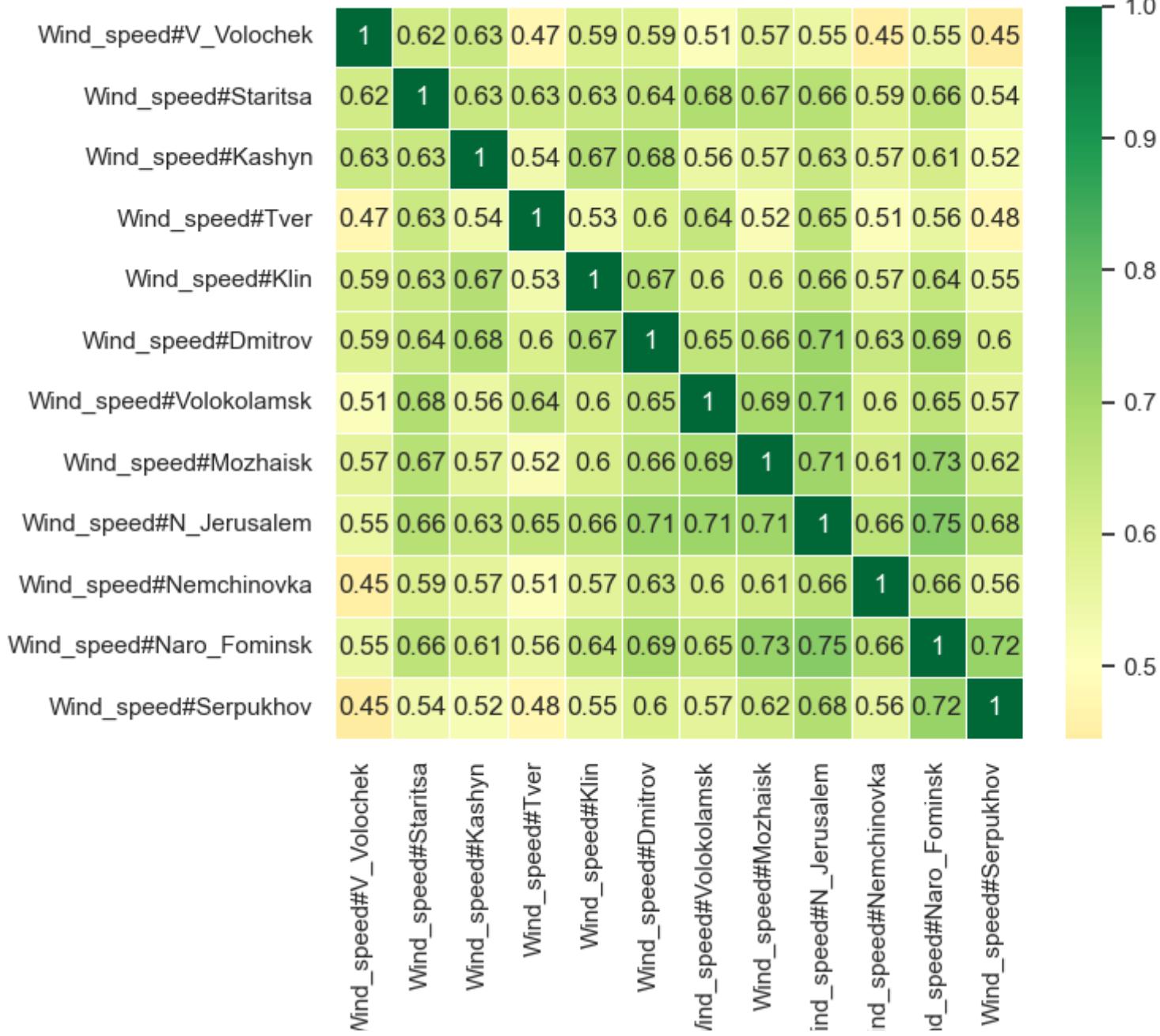
Корреляция параметра P_drift между метеостанциями



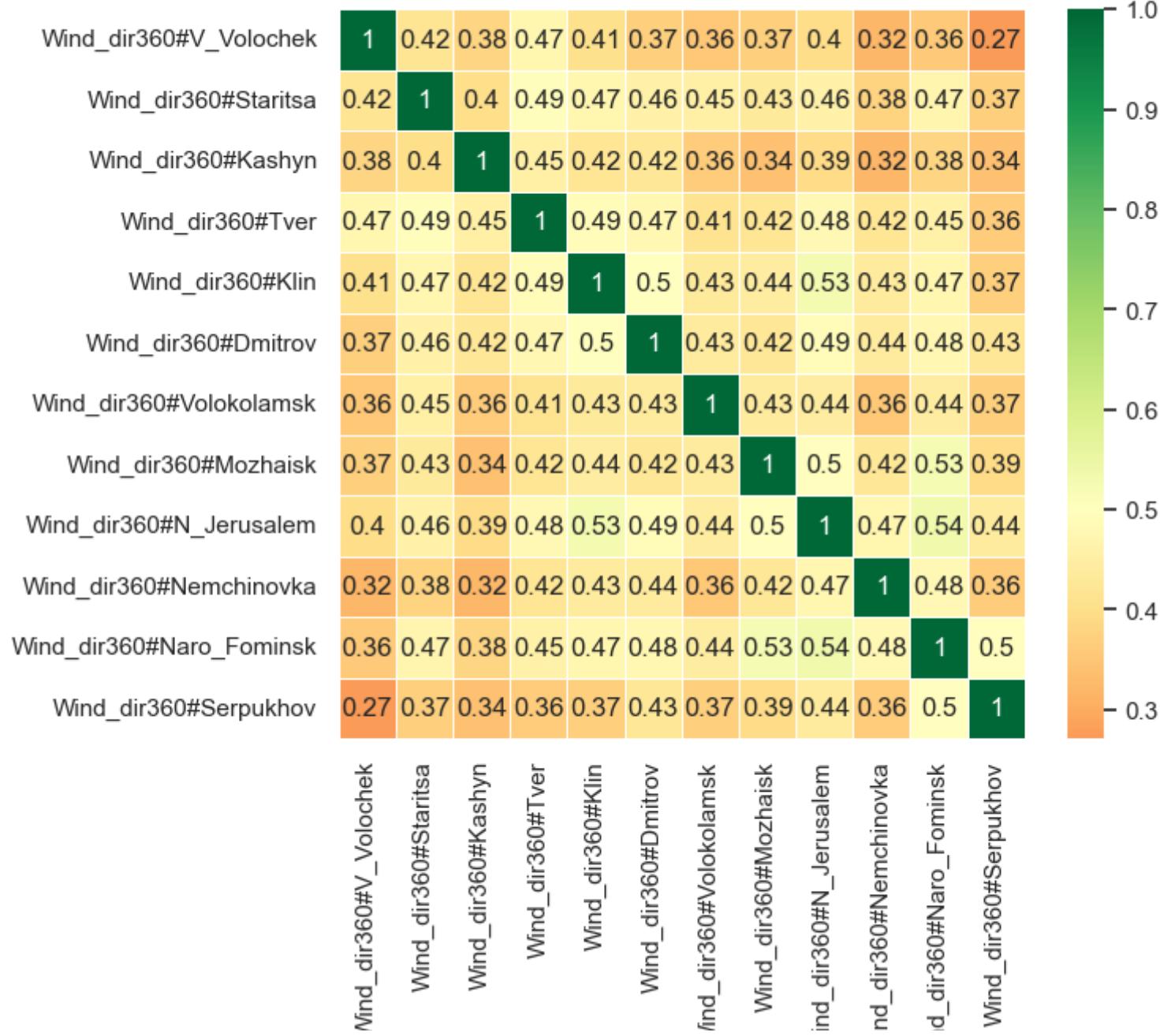
Корреляция параметра Humid между метеостанциями



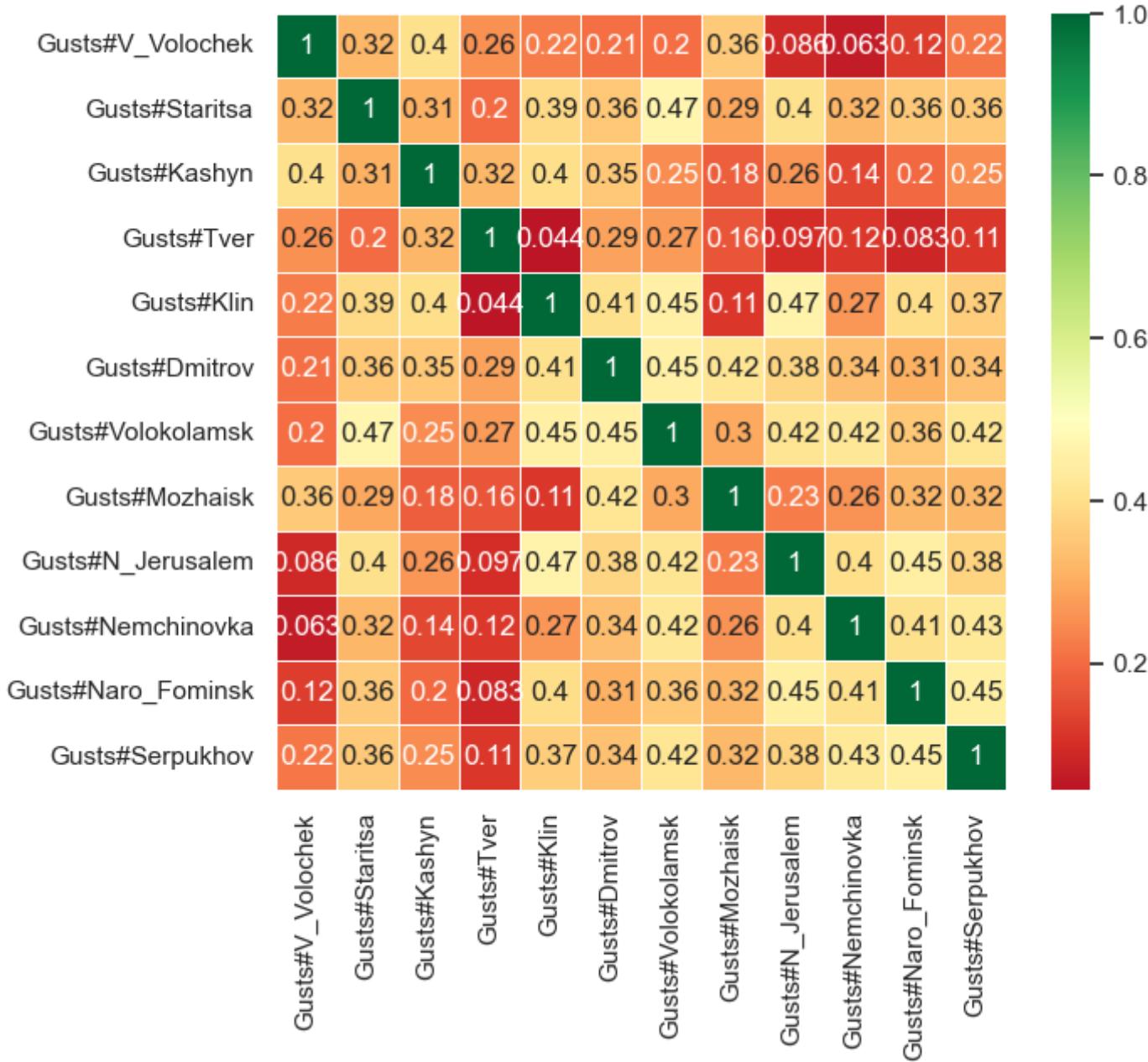
Корреляция параметра Wind_speed между метеостанциями



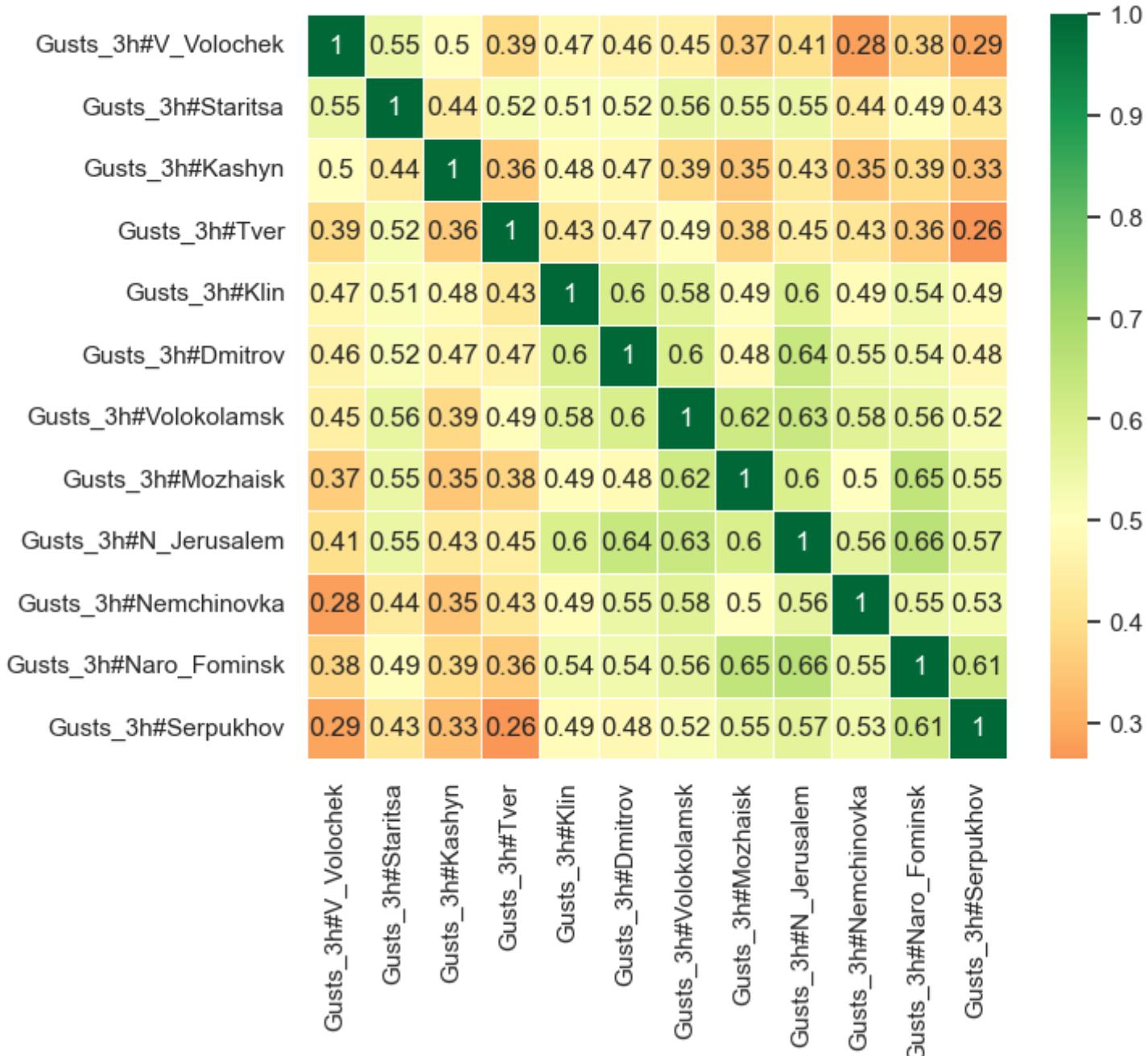
Корреляция параметра Wind_dir360 между метеостанциями



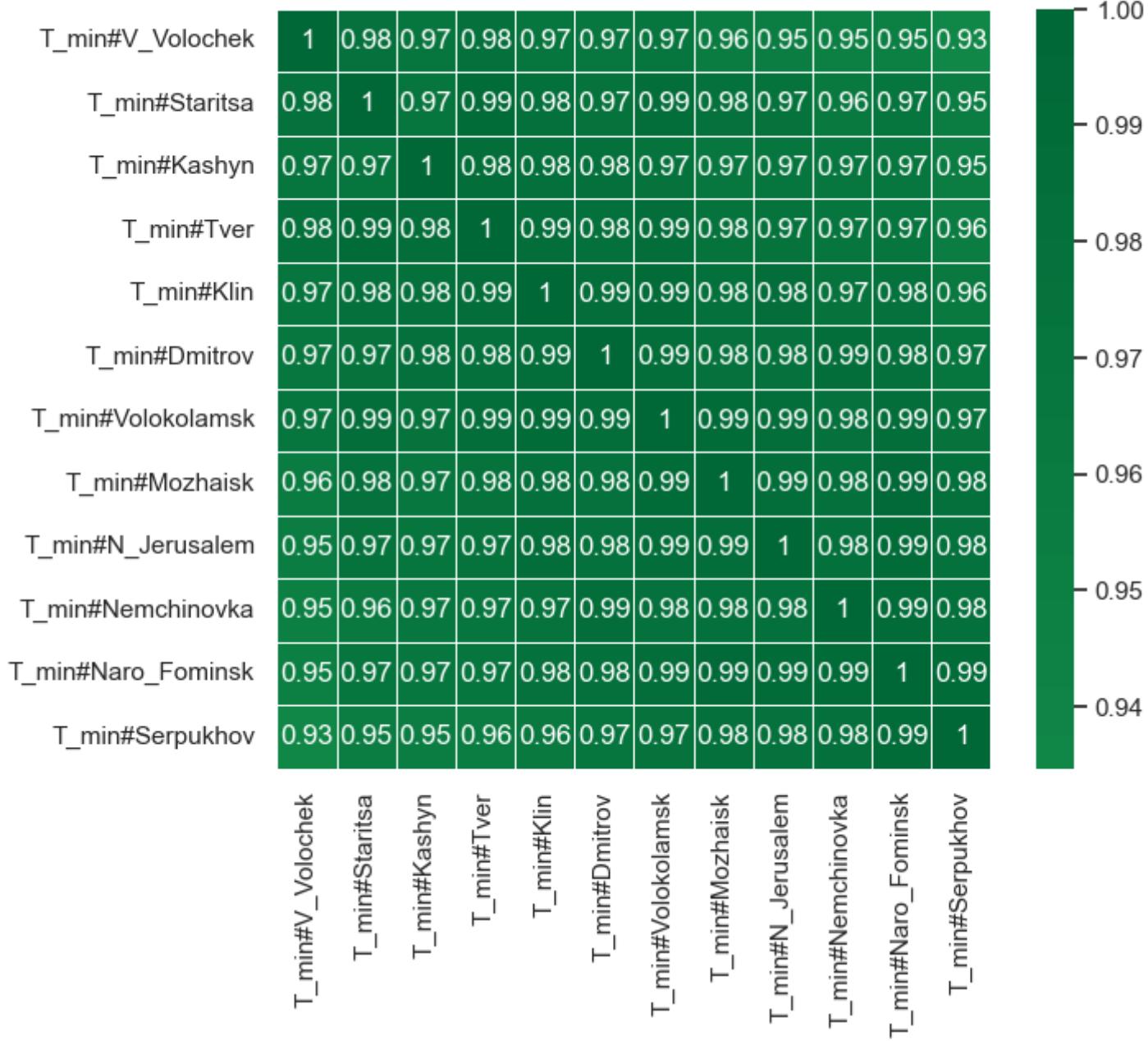
Корреляция параметра Gists между метеостанциями



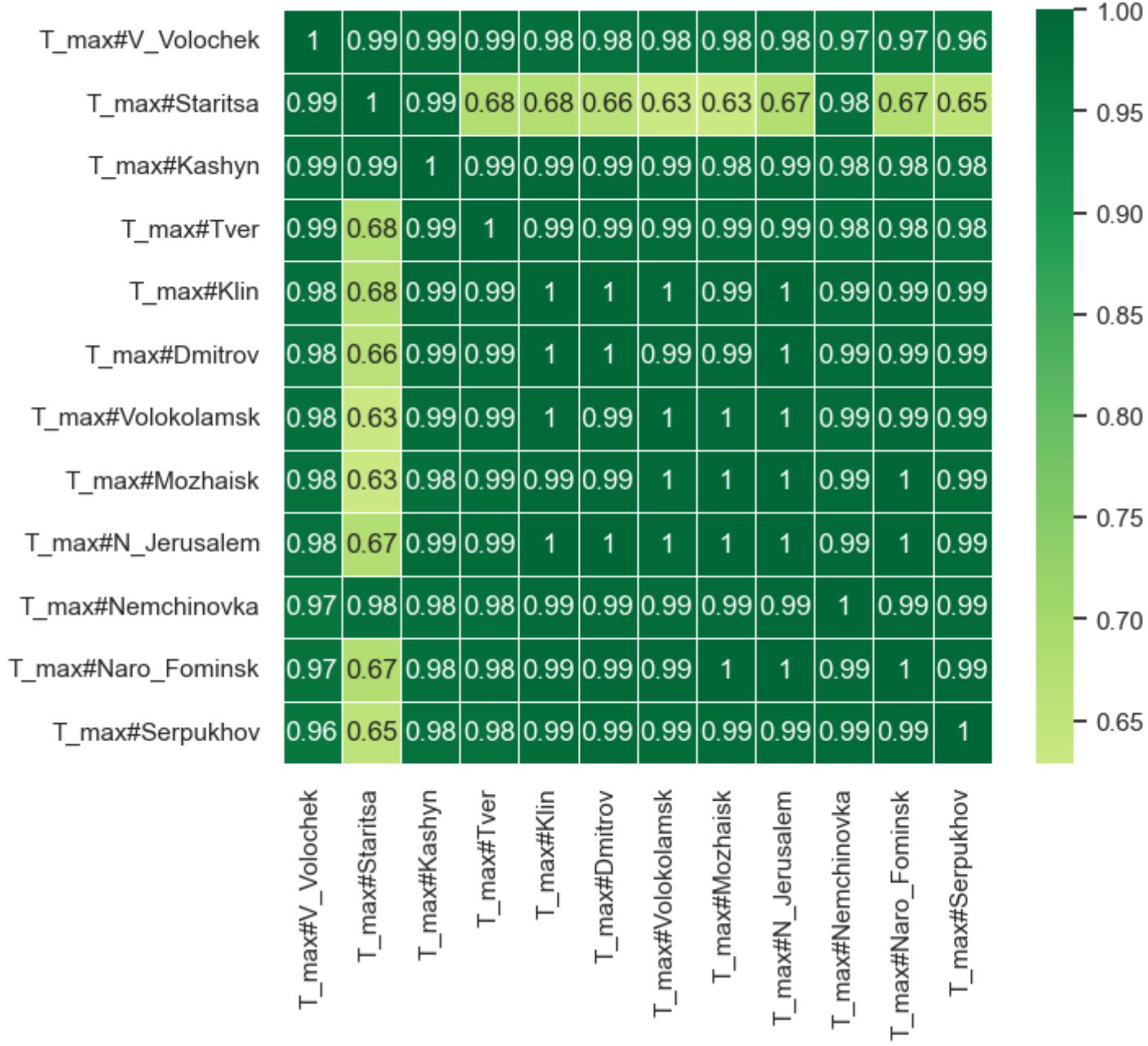
Корреляция параметра Gists_3h между метеостанциями



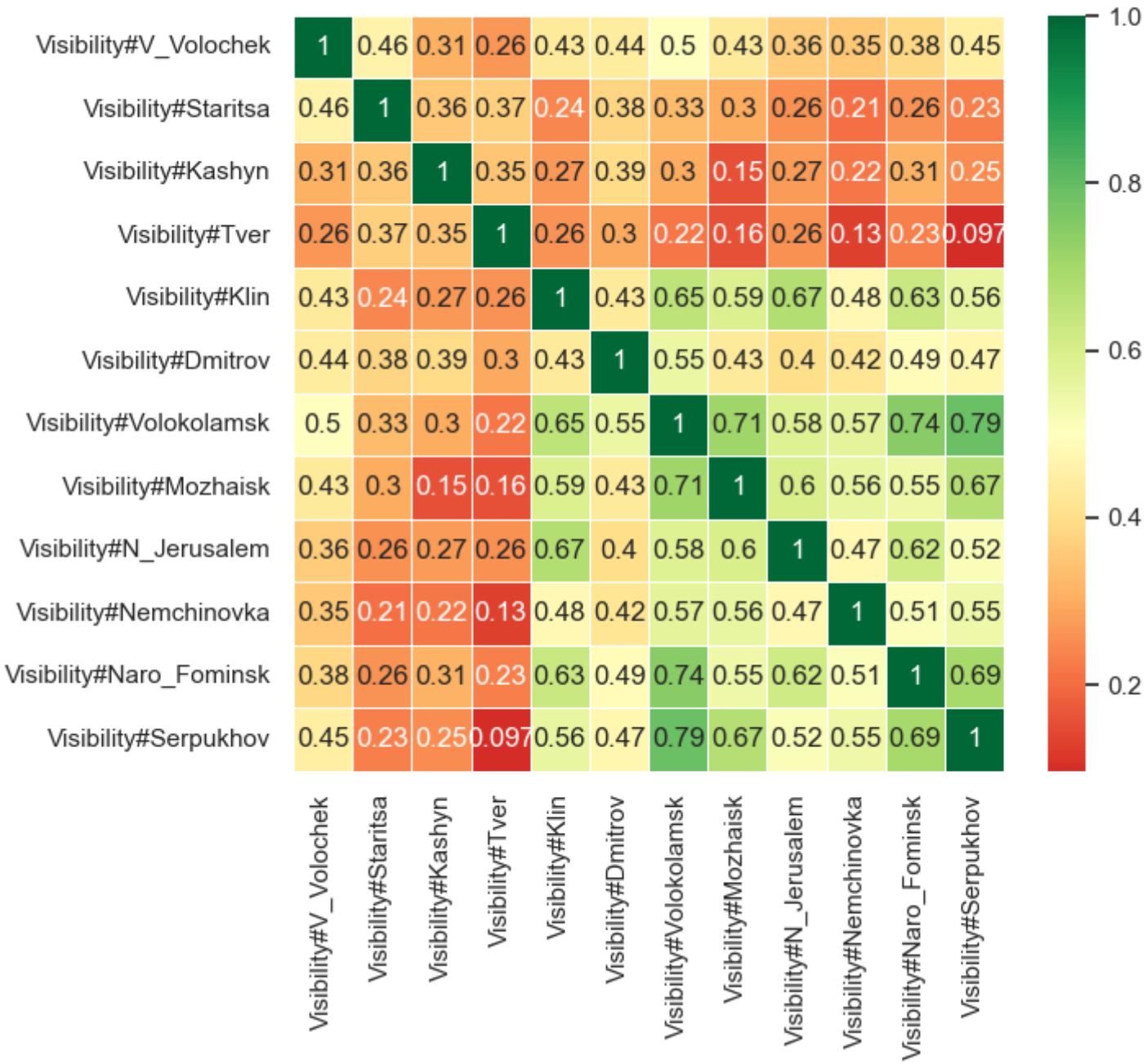
Корреляция параметра T_min между метеостанциями



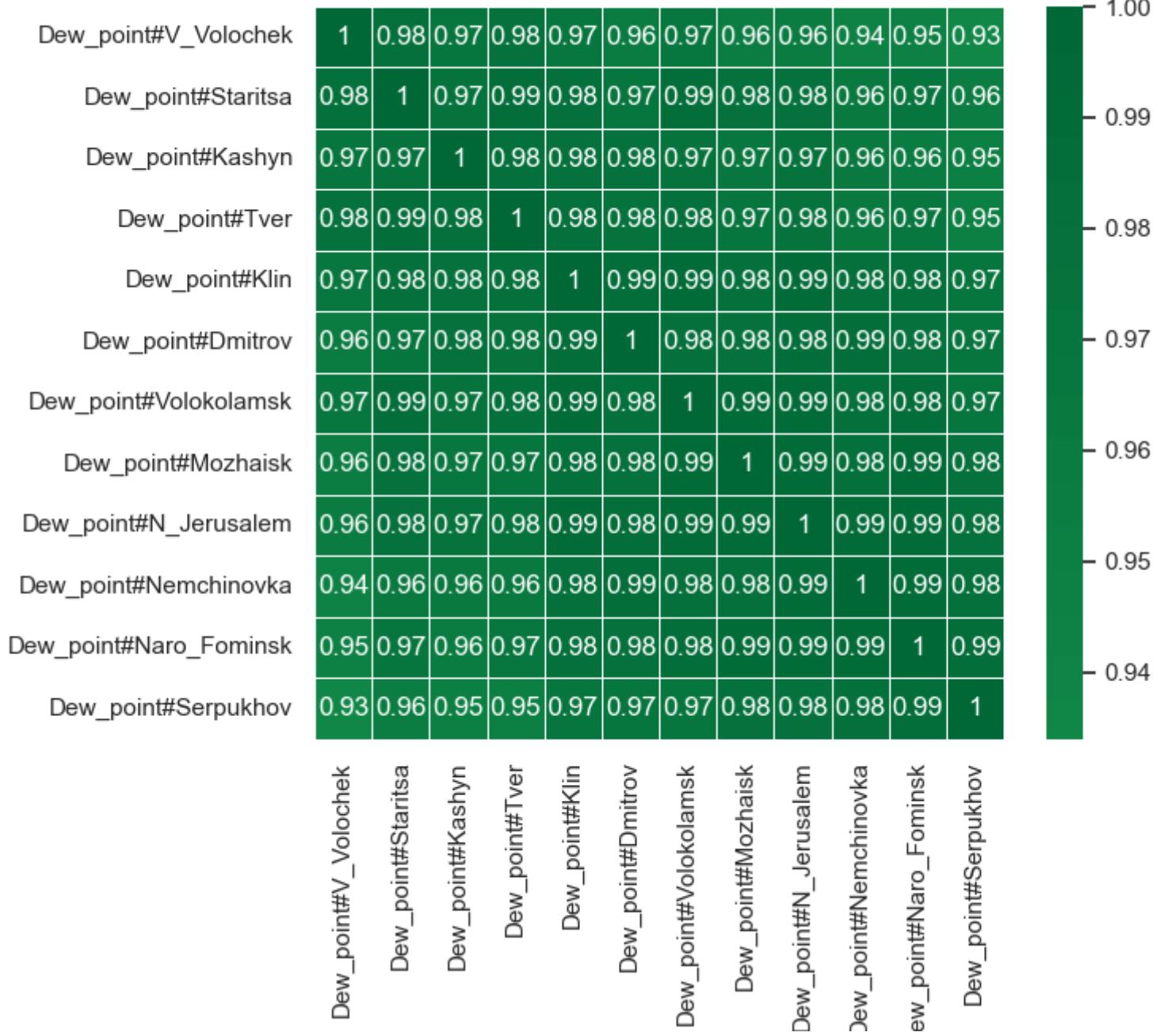
Корреляция параметра T_max между метеостанциями



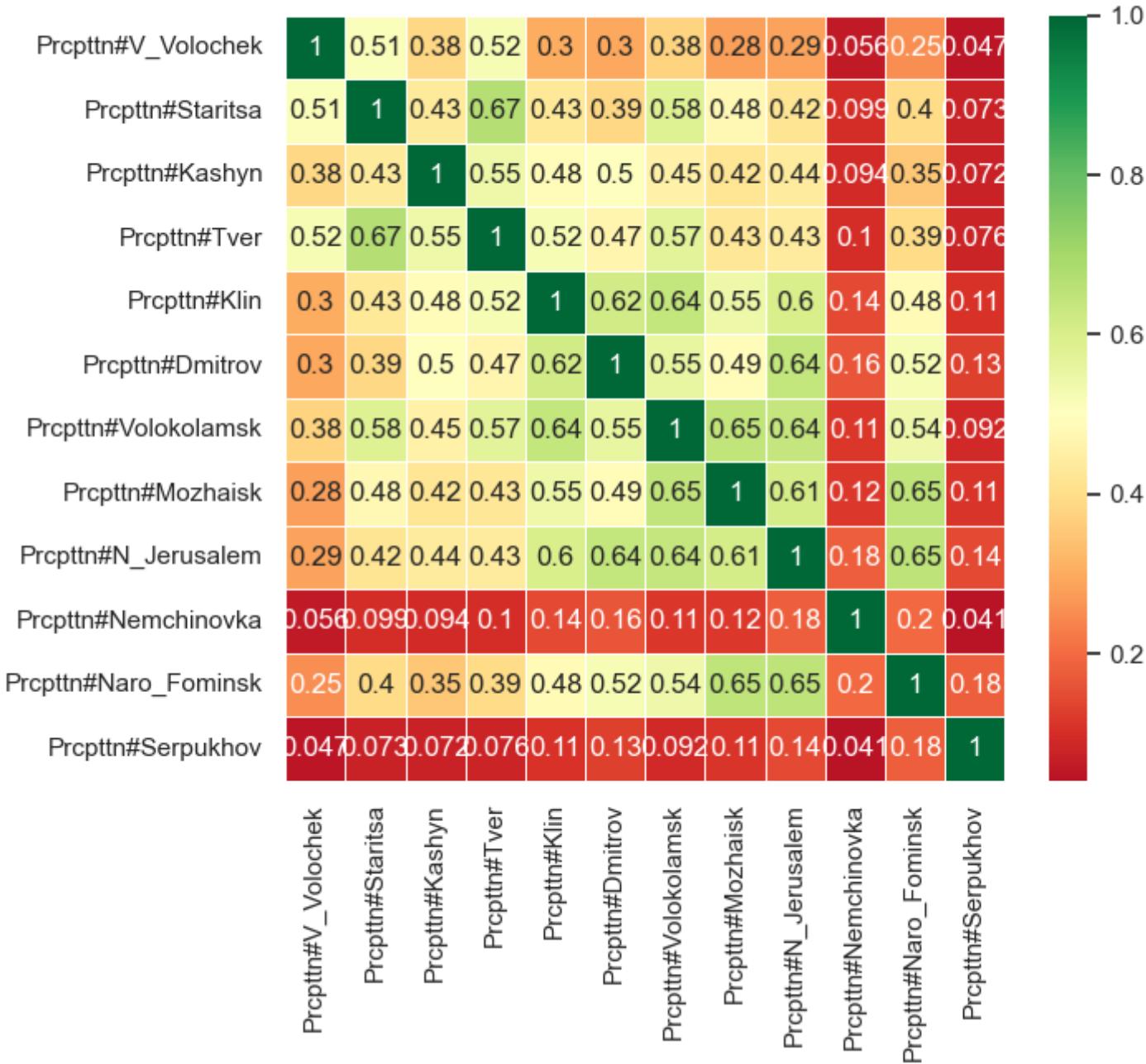
Корреляция параметра Visibility между метеостанциями



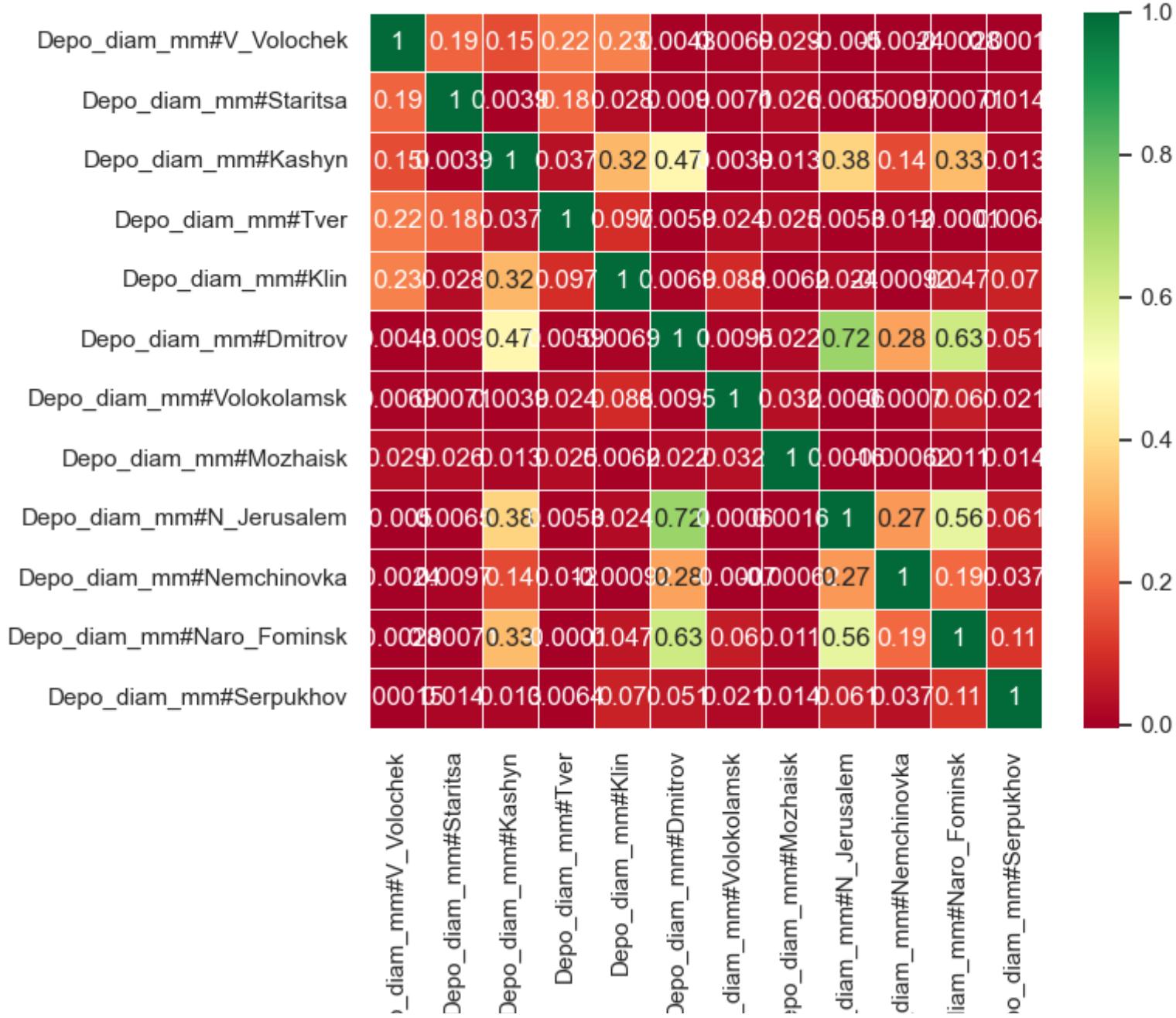
Корреляция параметра Dew_point между метеостанциями



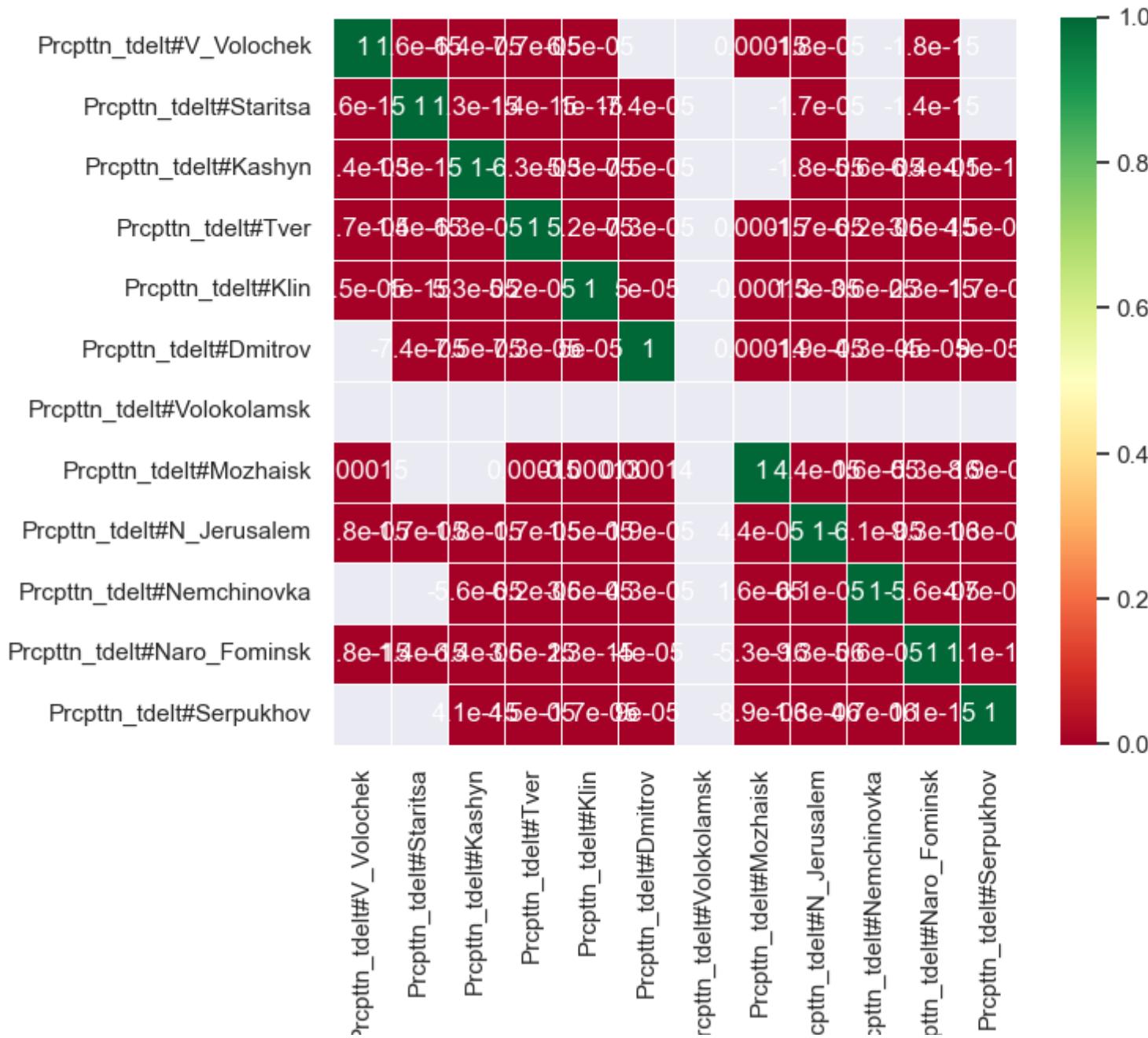
Корреляция параметра Prcpttn между метеостанциями



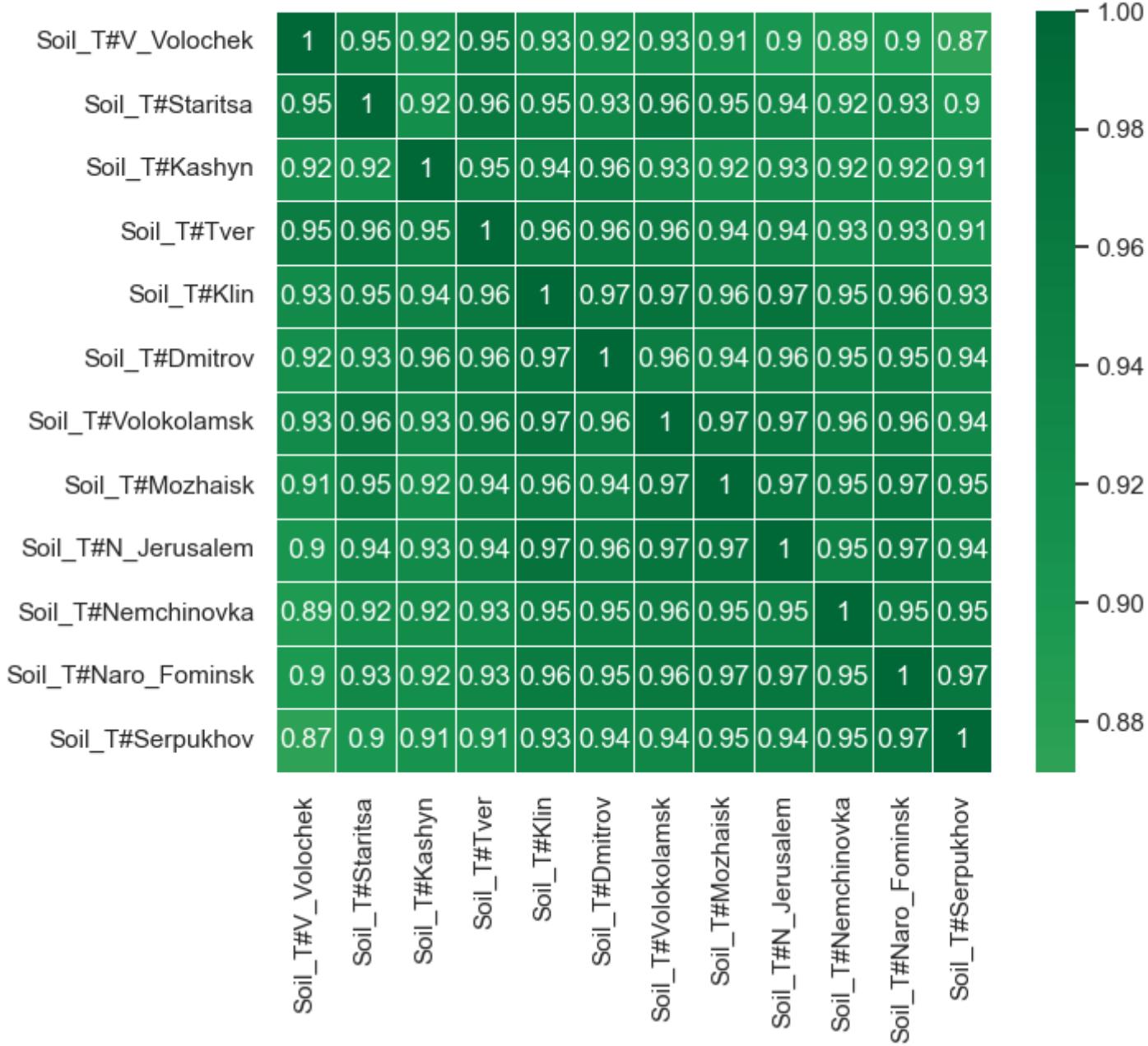
Корреляция параметра Depo_diam_mm между метеостанциями



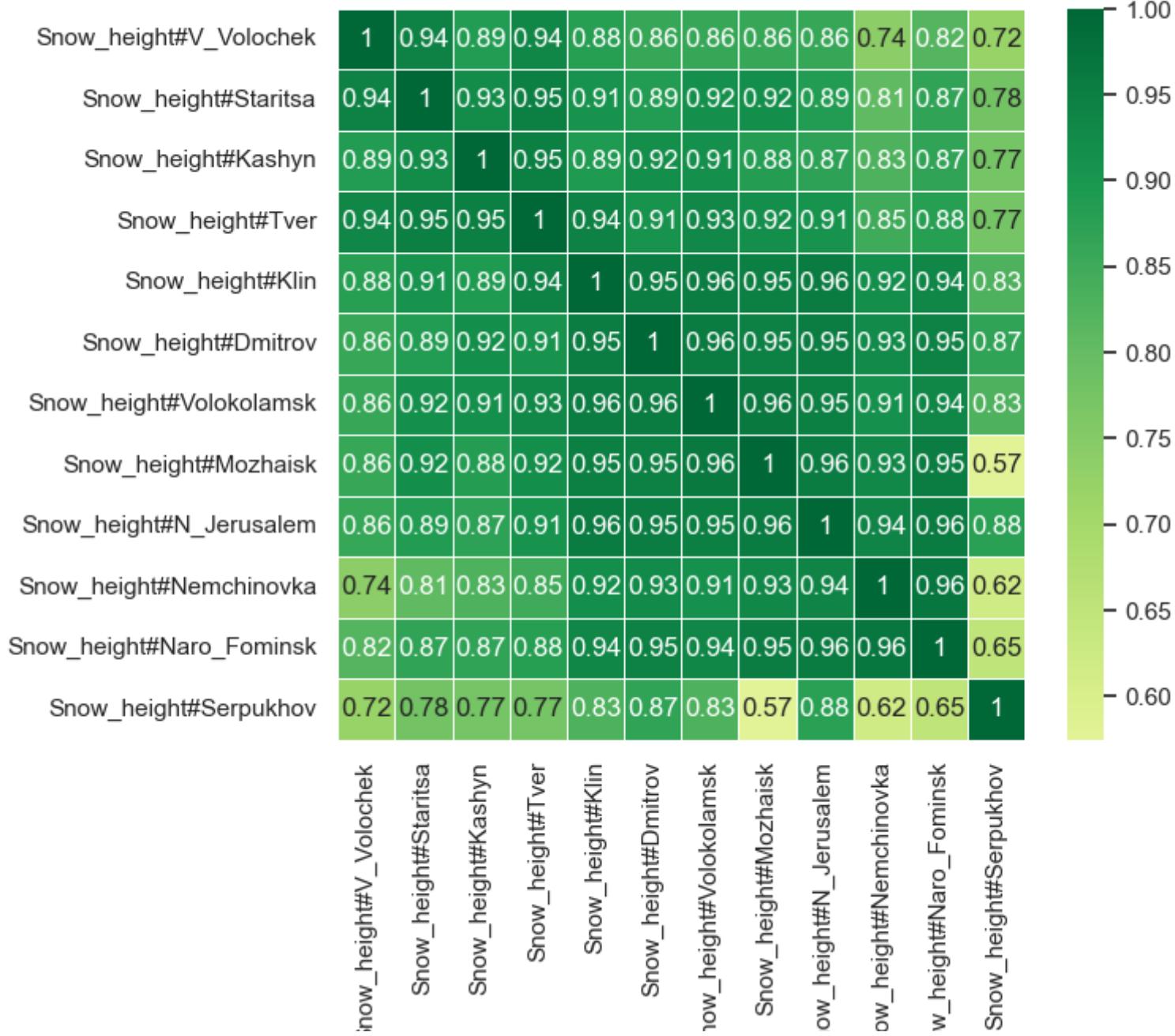
Корреляция параметра Precipitation_tdelta между метеостанциями



Корреляция параметра Soil_T между метеостанциями



Корреляция параметра Snow_height между метеостанциями



ВЫВОД

Используем шкалу оценки корреляции Чеддока : слабая — от 0,1 до 0,3; умеренная — от 0,3 до 0,5; заметная — от 0,5 до 0,7; высокая — от 0,7 до 0,9; весьма высокая (сильная) — от 0,9 до 1,0.

1. T(T) Температура воздуха на высоте 2 м над поверхностью земли, градусов Цельсия - *Сильная корреляция между всеми метеостанциями.*
2. P_station (Po) Атмосферное давление на уровне станции, мм рт. столба - *Сильная корреляция между всеми метеостанциями, кроме Вышнего Волочка (высокая).*
3. P_sea (P) Атмосферное давление, приведённое к среднему уровню моря, мм рт. столба - *Сильная корреляция между всеми метеостанциями, несколько ниже для Можайска.*
4. P_drift (Pa) Барическая тенденция: изменение атмосферного давления за последние 3 часа, мм рт. столба - *В основном высокая корреляция между метеостанциями.*
5. Humid (U) Относительная влажность воздуха на высоте 2 м над поверхностью земли, % - *Сильная корреляция между всеми метеостанциями, в ряде случаев - высокая.*
6. Wind_dir (DD)(Wind_dir360) Направление ветра на высоте 10-12 м над поверхностью земли, усреднённое за 10 минут непосредственно перед наблюдением, румбы (градусы) - *Умеренная корреляция между метеостанциями.*
7. Wind_speed, Ff Скорость ветра на высоте 10-12 м над поверхностью земли, усреднённая за 10 минут непосредственно перед наблюдением, м/с - *Заметная корреляция между всеми метеостанциями, в ряде случаев - умеренная, в ряде случаев - высокая.*
8. Gusts (ff10) Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за 10 минут непосредственно перед наблюдением, м/с- *Слабая корреляция между всеми метеостанциями, в ряде случаев умеренная.*
9. Gusts_3h (ff3) Максимальная скорость порыва ветра на высоте 10-12 м над поверхностью земли, за период между моментами наблюдения 3 часа, м/с *Умеренная корреляция между всеми метеостанциями, в ряде случаев - заметная.*
10. Cloudness (N) Общая облачность, % - *(Пока не анализировалась)*
11. Wthr_curr (WW) Текущая погода, сообщаемая метеостанцией - *(Пока не анализировалась)*
12. Wthr_3h (W1) Прошедшая погода между сроками наблюдения 1 - *(Пока не анализировалась)*
13. Wthr_3h2 (W2) Прошедшая погода между сроками наблюдения 2 - *(Пока не анализировалась)*
14. T_min (Tn) Минимальная температура воздуха за прошёдший период не более 12 часов, градусов Цельсия - *Сильная корреляция между всеми метеостанциями.*
15. T_max (Tx) Максимальная температура воздуха за прошёдший период не более 12 часов, градусов Цельсия - *Сильная корреляция между всеми метеостанциями, кроме Старицы - заметная.*

16. Cl_Cumls (Cl) Слоисто-кучевые, слоистые, кучевые и кучево-дождевые облака - *(Пока не анализировалась)*
17. Cl_viewed (Nh) Количество всех наблюдающихся облаков Cl, или при отсутствии облаков Cl, количество всех наблюдающихся облаков Cm, % - *(Пока не анализировалась)*
18. Cl_bottom (H) Высота основания самых низких облаков, м - *(Пока не анализировалась)*
19. Cl_cumls_hi (Cm) Высоко-кучевые, высоко-слоистые и слоисто-дождевые облака - *(Пока не анализировалась)*
20. Cl_cirrus (Ch) Перистые, перисто-кучевые и перисто-слоистые облака - *(Пока не анализировалась)*
21. Visibility (VV) Горизонтальная дальность видимости, км. - *Слабая и умеренная корреляция для метеостанций Тверской области, умеренная и заметная - для метеостанций Московской области.*
22. Dew_point (Td) Температура точки росы на высоте 2 м над поверхностью земли, градусов Цельсия - *Сильная корреляция между всеми метеостанциями.*
23. Prcpttn (RRR) Количество выпавших осадков, мм - *Слабая корреляция между далеко расположенными метеостанциями, умеренная и заметная приблизительно для метеостанций Северо-Запада Московской области.*
24. Prcpttn_tdelt (tR) Период времени, за который выпало указанное количество осадков, ч. - *Корреляция отсутствует*
25. Soil (E) Состояние поверхности почвы без снега или измеримого ледяного покрова - *(Пока не анализировалась).*
26. Soil_T (Tg) Минимальная температура поверхности почвы за ночь, градусов Цельсия - *Сильная корреляция между всеми метеостанциями.*
27. Soil_cover (E') Состояние поверхности почвы со снегом или измеримым ледяным покровом - *(Пока не анализировалась).*
28. Snow_height (sss) Высота снежного покрова, см - *В основном, сильная корреляция между всеми метеостанциями, для Тверской области - чаще высокая, для Серпухова в ряде случаев - заметная.*

1.4. Сводный EDA предварительных данных с помощью библиотеки SWEETVIZ - ad libitum

```
In [74]: # def function_that_warns():
#     warnings.warn("future deprecated", FutureWarning)

# with warnings.catch_warnings():
#     warnings.simplefilter("ignore")
#     for name, data in dict_df_locations.items(): # по элементам словаря с данными DFs
#         my_report = sv.analyze(data)
#         print(f"Название датафрейма: {name}\n")
#         my_report.show_notebook()
```

```
In [75]: # dict_df_parameters.keys()
```

```
In [76]: # def function_that_warns():
#     warnings.warn("future deprecated", FutureWarning)

# with warnings.catch_warnings():
#     warnings.simplefilter("ignore")
#     for name, data in dict_df_parameters.items(): # по элементам словаря с данными DFs
#         my_report = sv.analyze(data.iloc[:, :-1])
#         print(f"Название датафрейма: {name}\n")
#         my_report.show_notebook()
```

ПРОДОЛЖЕНИЕ В ТЕТЕРАДИ 2

Разное

Вернём значения отображения к настройкам по умолчанию

```
In [77]: pd.set_option('display.max_rows', 60) # восстановим значение по умолчанию максимума отображаемых строк
pd.set_option('display.max_columns', 20) # восстановим значение по умолчанию максимума отображаемых столбцов
```

```
In [78]: InteractiveShell.ast_node_interactivity = "last_expr" # Отображение результата только последней строки кода
```

```
In [ ]:
```