W12_KNMI_complete

January 12, 2021

1 KNMI import

Jefry el Bhwash 16095065

This notebook: - Reads KNMI data - Creates a dataframe that contains the correct parsed date

Bottom of the page:
One cell to copy to your own notebook is found at the bottom of the notebook

1.0.1 Imports

[69]:
```
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
from datetime import timedelta
import numpy as np
```

1.1 ## Headers of KNMI_20201016_hourly.txt for more information

BRON: KONINKLIJK NEDERLANDS METEOROLOGISCH INSTITUUT (KNMI)
Opmerking: door stationsverplaatsingen en veranderingen in waarneemmethodieken zijn deze tijdreeksen van uurwaarden mogelijk inhomogeen!
Dat betekent dat deze reeks van gemeten waarden niet geschikt is voor trendanalyse.
Voor studies naar klimaatverandering verwijzen we naar de gehomogeniseerde reeks maandtemperaturen van De Bilt http://www.knmi.nl/klimatologie/onderzoeksgegevens/homogeen_260/index.html of de Centraal Nederland Temperatuur http://www.knmi.nl/klimatologie/onderzoeksgegevens/CNT/.

```
STN      LON(east)   LAT(north)     ALT(m)  NAME 215:         4.437
52.141      -1.10  VOORSCHOTEN  YYYYMMDD = datum (YYYY=jaar,MM=maand,DD=dag);
HH       = tijd (HH=uur, UT.12 UT=13 MET, 14 MEZT. Uurvak 05 loopt
van 04.00 UT tot 5.00 UT;  DD       = Windrichting (in graden)
gemiddeld over de laatste 10 minuten van het afgelopen uur (360=noord,
90=oost, 180=zuid, 270=west, 0=windstil 990=veranderlijk. Zie
http://www.knmi.nl/kennis-en-datacentrum/achtergrond/klimatologische-brochures-en-boeken;
FH       = Uurgemiddelde windsnelheid (in 0.1 m/s). Zie http://www.knmi.nl/kennis-en-datacentru
```

```
FF       = Windsnelheid (in 0.1 m/s) gemiddeld over de laatste 10 minuten
van het afgelopen uur;  FX       = Hoogste windstoot (in 0.1 m/s) over het
afgelopen uurvak;  T        = Temperatuur (in 0.1 graden Celsius) op 1.50 m
hoogte tijdens de waarneming;  T10N     = Minimumtemperatuur (in 0.1 graden
Celsius) op 10 cm hoogte in de afgelopen 6 uur;  TD       = Dauwpuntstemperatuur
(in 0.1 graden Celsius) op 1.50 m hoogte tijdens de waarneming;  SQ       =
Duur van de zonneschijn (in 0.1 uren) per uurvak, berekend uit globale straling
(-1 for <0.05 uur);  Q        = Globale straling (in J/cm2) per uurvak;
DR       = Duur van de neerslag (in 0.1 uur) per uurvak;  RH       = Uursom
van de neerslag (in 0.1 mm) (-1 voor <0.05 mm);  P        = Luchtdruk (in 0.1
hPa) herleid naar zeeniveau, tijdens de waarneming;  VV       = Horizontaal
zicht tijdens de waarneming (0=minder dan 100m, 1=100-200m, 2=200-300m,...,
49=4900-5000m, 50=5-6km, 56=6-7km, 57=7-8km, ..., 79=29-30km, 80=30-35km,
81=35-40km,..., 89=meer dan 70km);  N        = Bewolking (bedekkingsgraad van
de bovenlucht in achtsten), tijdens de waarneming (9=bovenlucht onzichtbaar);
U        = Relatieve vochtigheid (in procenten) op 1.50 m hoogte tijdens de
waarneming;  WW       = Weercode (00-99), visueel(WW) of automatisch(WaWa)
waargenomen, voor het actuele weer of het weer in het afgelopen uur. Zie
http://bibliotheek.knmi.nl/scholierenpdf/weercodes_Nederland;  IX       =
Weercode indicator voor de wijze van waarnemen op een bemand of automatisch
station (1=bemand gebruikmakend van code uit visuele waarnemingen, 2,3=bemand
en weggelaten (geen belangrijk weersverschijnsel, geen gegevens), 4=automatisch
en opgenomen (gebruikmakend van code uit visuele waarnemingen), 5,6=automatisch
en weggelaten (geen belangrijk weersverschijnsel, geen gegevens), 7=automatisch
gebruikmakend van code uit automatische waarnemingen);  M        = Mist 0=niet
voorgekomen, 1=wel voorgekomen in het voorgaande uur en/of tijdens de waarneming;
R        = Regen 0=niet voorgekomen, 1=wel voorgekomen in het voorgaande uur
en/of tijdens de waarneming;  S        = Sneeuw 0=niet voorgekomen, 1=wel
voorgekomen in het voorgaande uur en/of tijdens de waarneming;  O        = Onweer
0=niet voorgekomen, 1=wel voorgekomen in het voorgaande uur en/of tijdens de
waarneming;  Y        = IJsvorming 0=niet voorgekomen, 1=wel voorgekomen in het
voorgaande uur en/of tijdens de waarneming;
```

2 Reading the file

Then dropping stuff
Then also typecasting the columns back to ints if they were changed by reading it.

[70]:
```
header = 22
weer = pd.read_csv('KNMI_20201124_hourly.txt', header = header+10)
weer = weer.drop(0).drop(['# STN'], axis =1)
#fix columns
weer['YYYYMMDD'] = weer['YYYYMMDD'].apply(lambda x: int(x))
weer['   HH'] = weer['   HH'].apply(lambda x: int(x))
```

[71]:
```
#see what columns are available
print(weer.columns)
```

```
Index(['YYYYMMDD', '   HH', '   DD', '   FH', '   FF', '   FX', '    T',
       ' T10N', '   TD', '   SQ', '    Q', '   DR', '   RH', '    P', '   VV',
       '    N', '    U', '   WW', '   IX', '    M', '    R', '    S', '    O',
       '    Y'],
      dtype='object')
```

[72]:
```
weer = weer.replace(r'^\s*$', np.nan, regex=True) #replacing empty values with␣
↪NaN's
print(weer.head(1))
df = pd.DataFrame()
```

```
   YYYYMMDD     HH     DD     FH     FF     FX      T  T10N     TD     SQ  \
1  20181201      1  190.0   40.0   40.0   50.0   59.0   NaN   53.0    0.0

   …     VV      N     U     WW    IX      M      R      S      O      Y
1  …     63      3  96.0    NaN   5.0      0      0      0      0      0

[1 rows x 24 columns]
```

---

3 Two options for the hour formatting

1. 04-05 = 05 [documentation says this]
2. 04-05 = 04

3.0.1 Option one

- 04-05 = 05 [documentation says this]

[73]:
```
weer['datetime'] = pd.to_datetime(weer['YYYYMMDD'], format='%Y%m%d')
df['date'] = weer.apply(lambda x: x['datetime']+timedelta(days=1) if x['  ␣
↪HH']==24 else x['datetime'], axis=1)
df['date'] = df.apply(lambda x: x['date'].strftime('%Y%m%d'), axis=1)
df['hour'] = weer.apply(lambda x: 0 if x['   HH']==24 else x['   HH'], axis=1)
df_04_to_05_is_05 = df
df_04_to_05_is_05.head()
```

[73]:

	date	hour
1	20181201	1
2	20181201	2
3	20181201	3
4	20181201	4

```
5  20181201      5
```

4 Creating a dataframe called knmi

This will contain the neccessary columns (which you have to add yourself) with the correct datetime as the index

[74]:
```
data = df_04_to_05_is_05
knmi = pd.DataFrame()
knmi['Date'] = data['date']
knmi['Hour'] = data['hour'].apply('{:0>2}'.format)

#Add the info columns
knmi['wind_direction_last10'] =weer['   DD']
knmi['wind_speed_mean'] =weer['   FH']
knmi['wind_speed_last10'] =weer['   FF']
knmi['wind_speed_maximum'] =weer['   FX']
knmi['temperature'] =weer['    T']
knmi['temperature_minimum'] =weer[' T10N']
knmi['temperature_dewpoint'] =weer['   TD']
knmi['sun_duration'] =weer['   SQ']
knmi['sun_irradiance'] =weer['    Q']
knmi['rain_duration'] =weer['   DR']
knmi['rain_hoursum'] =weer['   RH']
knmi['air_pressure'] =weer['    P']
knmi['view_distance'] =weer['   VV']
knmi['cloudiness'] =weer['    N']
knmi['humidity_relative'] =weer['    U']
knmi['weather_code_nr'] =weer['   WW']
knmi['weather_code_indication'] =weer['   IX']
knmi['mist_bool'] =weer['    M']
knmi['rain_bool'] =weer['    R']
knmi['snow_bool'] =weer['    S']
knmi['thunder_bool'] =weer['    O']
knmi['ice_formation_bool'] =weer['    Y']

#Create datetime index
knmi['DateTime'] = knmi['Date'].astype(str) + knmi['Hour'].astype(str)
knmi['DateTime'] = pd.to_datetime(knmi['DateTime'], format='%Y%m%d%H')
#drop unneccesary columns
knmi = knmi.set_index(knmi['DateTime']).drop(['Date', 'Hour', 'DateTime'],␣
 ↪axis=1)

print()
knmi = knmi.astype('float64')
```

```
print(knmi.dtypes)
knmi.head(2)
```

```
wind_direction_last10      float64
wind_speed_mean            float64
wind_speed_last10          float64
wind_speed_maximum         float64
temperature                float64
temperature_minimum        float64
temperature_dewpoint       float64
sun_duration               float64
sun_irradiance             float64
rain_duration              float64
rain_hoursum               float64
air_pressure               float64
view_distance              float64
cloudiness                 float64
humidity_relative          float64
weather_code_nr            float64
weather_code_indication    float64
mist_bool                  float64
rain_bool                  float64
snow_bool                  float64
thunder_bool               float64
ice_formation_bool         float64
dtype: object
```

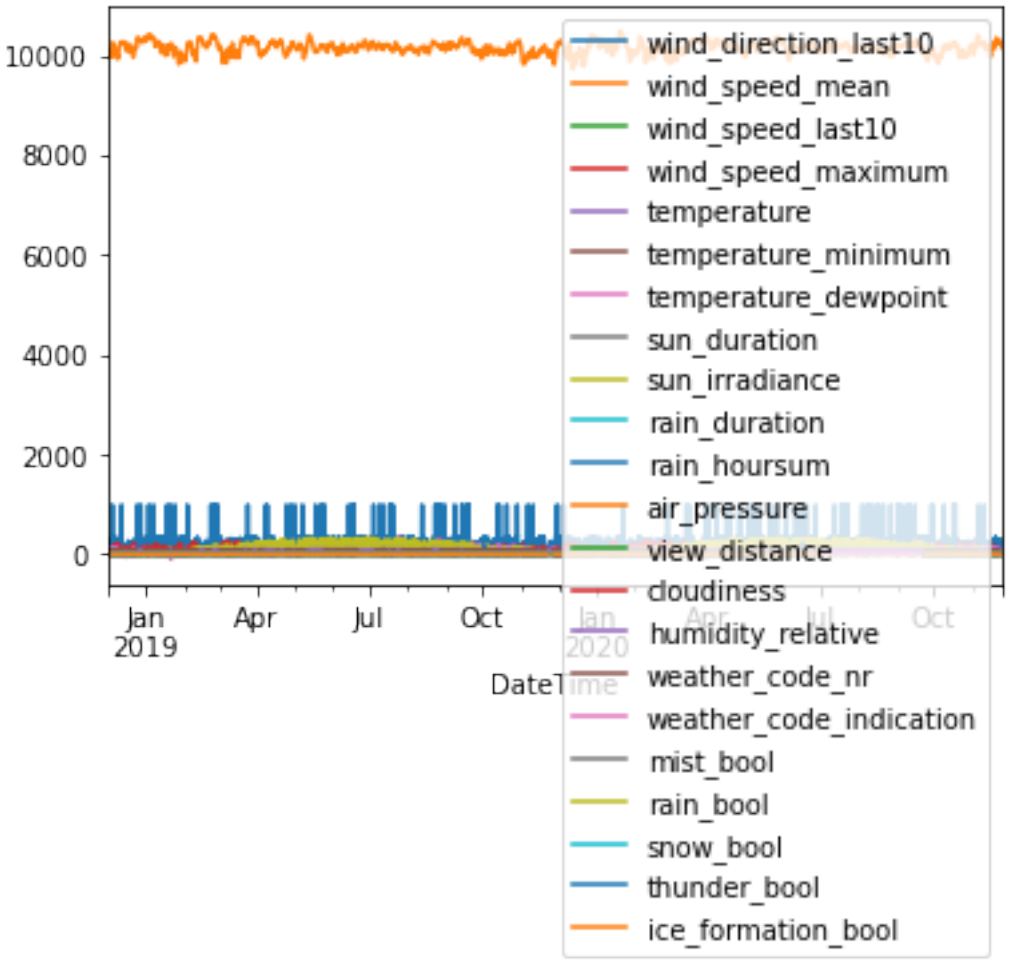
[74]:

DateTime	wind_direction_last10	wind_speed_mean	wind_speed_last10	wind_speed_maximum	temperature	temperature_minimum	temperature_dewpoint	sun_duration	sun_irradiance	rain_duration	…	view_distance
2018-12-01 01:00:00	190.0	40.0	40.0	50.0	59.0	NaN	53.0	0.0	0.0	0.0	…	63.0
2018-12-01 02:00:00	200.0	40.0	30.0	60.0	60.0	NaN	54.0	0.0	0.0	0.0	…	65.0

```
                     cloudiness  humidity_relative  weather_code_nr  \
DateTime
2018-12-01 01:00:00         3.0               96.0              NaN
2018-12-01 02:00:00         2.0               95.0              NaN

                     weather_code_indication  mist_bool  rain_bool  snow_bool  \
DateTime
2018-12-01 01:00:00                      5.0        0.0        0.0        0.0
2018-12-01 02:00:00                      5.0        0.0        0.0        0.0

                     thunder_bool  ice_formation_bool
DateTime
2018-12-01 01:00:00           0.0                 0.0
2018-12-01 02:00:00           0.0                 0.0

[2 rows x 22 columns]
```

4.0.1 Plotting the new dataframe

[75]:
```
plt.close()
knmi.plot()
plt.show()
```



---

5 Saving this dataframe

[76]:
```
knmi.to_pickle('W12_KNMI_complete')
```