3.1 Data Selection

August 14, 2020

Table of Contents

1 Ideas

- select parameters influence concentration of PM2.5 mostly in transport
- data should be indenpendent to PM2.5, and should be a physical entities rather a prepresentation of such data
- I explicitly select data from MERRA-2 groups (SLV: single level, and some from FLX: flux diagnosis)

2 Libraries

[1]:
```
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

```
import seaborn as sns
plt.rcParams['figure.figsize'] = (14,6)
```

[3]:
```
import datetime
```

[4]:
```
# plt.rcParamsDefault
```

- let visit some graph we hare in the previous exercise
- with observed ground data
- they are all basic parameter, let use all of them. For wind speeds, we will have more options with MERRA-2
- with SLV groups in MERRA-2 reanalysis
- selected columns T2MWET, T2MDEW, T2M, QV10M, PS, TQV, TQL, H1000, DISPH, wind (2-50, and 1400m)
- FLX group

select columns:
- QV10M 10-meter_specific_humidity
- QLML surface_specific_humidity
- TQL total_precipitable_liquid_water
- FRCAN areal_fraction_of_anvil_showers
- HLML surface_layer_height
- H1000 height_at_1000_mb
- DISPH zero_plane_displacement_height
- U2M 2-meter_eastward_wind
- V2M 2-meter_northward_wind
- U50M eastward_wind_at_50_meters
- V50M northward_wind_at_50_meters
- U850 eastward_wind_at_850_hPa
- V850 northward_wind_at_850_hPa
- SPEED surface_wind_speed, m s-1
- SPEEDMAX surface_wind_speed, m s-1

[5]:
```
# let build a function to look up the name and unit from netcdf4 file
import netCDF4 as nc
```

[6]:
```
def collect_meta_name():
    meta_name = dict()
    files = ['data/nc4/MERRA2_400.tavg1_2d_slv_Nx.20180722.nc4',
             'data/nc4/MERRA2_400.tavg1_2d_flx_Nx.20180102.nc4',
             'data/nc4/MERRA2_400.tavg1_2d_aer_Nx.20180101.nc4']
    for file in files:
        ds = nc.Dataset(file)
        for k, v in ds.variables.items():
            meta_name[k] = f'{v.long_name}, {v.units}'
```

```python
    return meta_name
```

[7]:
```python
name_ = collect_meta_name()
```

[8]:
```python
def look_kw(kw=None):
    name_ = collect_meta_name()

    for k,v in name_.items():
        if kw in v:
            print(k,v)
    return None
```

[9]:
```python
look_kw(kw='wind')
```

```
U2M 2-meter_eastward_wind, m s-1
V250 northward_wind_at_250_hPa, m s-1
U850 eastward_wind_at_850_hPa, m s-1
V850 northward_wind_at_850_hPa, m s-1
V50M northward_wind_at_50_meters, m s-1
U250 eastward_wind_at_250_hPa, m s-1
V2M 2-meter_northward_wind, m s-1
V10M 10-meter_northward_wind, m s-1
U50M eastward_wind_at_50_meters, m s-1
U10M 10-meter_eastward_wind, m s-1
V500 northward_wind_at_500_hPa, m s-1
U500 eastward_wind_at_500_hPa, m s-1
ULML surface_eastward_wind, m s-1
SPEED surface_wind_speed, m s-1
SPEEDMAX surface_wind_speed, m s-1
VLML surface_northward_wind, m s-1
BCFLUXU Black Carbon column u-wind mass flux, kg m-1 s-1
OCFLUXV Organic Carbon column v-wind mass flux __ENSEMBLE__, kg m-1 s-1
SUFLUXV SO4 column v-wind mass flux __ENSEMBLE__, kg m-1 s-1
SSFLUXU Sea Salt column u-wind mass flux, kg m-1 s-1
DUFLUXV Dust column v-wind mass flux, kg m-1 s-1
DUFLUXU Dust column u-wind mass flux, kg m-1 s-1
SSFLUXV Sea Salt column v-wind mass flux, kg m-1 s-1
SUFLUXU SO4 column u-wind mass flux __ENSEMBLE__, kg m-1 s-1
BCFLUXV Black Carbon column v-wind mass flux, kg m-1 s-1
OCFLUXU Organic Carbon column u-wind mass flux __ENSEMBLE__, kg m-1 s-1
```

[10]:
```python
look_kw(kw='plane')
```

```
DISPH zero_plane_displacement_height, m
TCZPBL transcom_planetary_boundary_layer_height, m
PBLH planetary_boundary_layer_height, m
```

[11]:
```
look_kw(kw='height')
```

```
H250 height_at_250_hPa, m
DISPH zero_plane_displacement_height, m
H1000 height_at_1000_mb, m
H850 height_at_850_hPa, m
H500 height_at_500_hPa, m
TCZPBL transcom_planetary_boundary_layer_height, m
HLML surface_layer_height, m
PBLH planetary_boundary_layer_height, m
```

3 Load data

3.1 PM2.5

[12]:
```
pm25 = pd.read_csv('data/cleaned_pm25_Hanoi_PM2.5_2018_YTD.csv',
                   parse_dates=['Date (LT)'],
                   index_col=['Date (LT)'])
pm25.head()
```

[12]:

Date (LT)	pm25
2018-01-01 01:00:00	69.2
2018-01-01 02:00:00	75.5
2018-01-01 03:00:00	90.2
2018-01-01 04:00:00	97.6
2018-01-01 05:00:00	89.1

3.2 Observed data from Noibai (ISD data)

[13]:
```
nb = pd.read_csv('data/cleaned_noibai_noaa_isd_2018.csv',
                 parse_dates=['DATE'],
                 index_col=['DATE'])
nb.head()
```

[13]:

DATE	CIG	VIS	TMP	DEW	WD	WS	CLDCR	CLDHT
2018-01-01 00:00:00	1067.0	8000	16.0	12.0	80	1.5	0.7	1067.0
2018-01-01 00:30:00	975.0	8000	16.0	12.0	60	1.5	0.7	975.0
2018-01-01 01:00:00	975.0	7000	16.0	12.0	80	1.5	0.7	975.0
2018-01-01 01:30:00	975.0	7000	17.0	12.0	60	2.1	0.7	975.0
2018-01-01 02:00:00	1006.0	7000	17.0	12.0	80	3.1	0.4	762.0

[14]:

```
# since it use UTC time, convert to the local time (GMT+7)
nb.index = nb.index + datetime.timedelta(hours=7)
```

3.3 MERRA-2 SLV group

[15]:

```
# T2MWET, T2MDEW, T2M, QV10M, PS, TQV, TQL, H1000, DISPH, winds
cols = ['T2MWET', 'T2MDEW', 'T2M', 'QV10M', 'PS', 'TQV', 'TQL', 'H1000',␣
↪'DISPH', 'U2M', 'V2M', 'U50M', 'V50M', 'U850', 'V850']
```

[16]:

```
# read and filter out by the column name
slv = pd.read_csv('data/merra2_slv_hanoi_2018.csv',
                  parse_dates=['time'],
                  index_col=['time'])
print(slv.columns)
slv = slv[cols]
slv.head()
```

```
Index(['U2M', 'V250', 'TROPT', 'TROPPB', 'T2M', 'TQL', 'T500', 'U850', 'PS',
       'V850', 'H250', 'Q250', 'T2MDEW', 'V50M', 'Q500', 'DISPH', 'H1000',
       'TS', 'T10M', 'TROPPT', 'SLP', 'U250', 'Q850', 'ZLCL', 'TQV', 'V2M',
       'T250', 'TROPQ', 'V10M', 'H850', 'T850', 'U50M', 'U10M', 'TROPPV',
       'H500', 'V500', 'T2MWET', 'U500', 'QV10M'],
      dtype='object')
```

[16]:

```
                         T2MWET     T2MDEW        T2M     QV10M         PS  \
time
2018-01-01 00:00:00  284.03730  284.03317  287.10890  0.007823  100905.08
2018-01-01 01:00:00  283.94345  283.94443  286.79376  0.007823  100865.09
2018-01-01 02:00:00  283.87656  283.87836  286.48932  0.007822  100819.56
2018-01-01 03:00:00  283.76090  283.75630  286.24753  0.007807  100793.71
2018-01-01 04:00:00  283.66614  283.65967  285.96360  0.007804  100791.80

                           TQV       TQL      H1000     DISPH       U2M  \
time
2018-01-01 00:00:00  34.673485  0.008423  164.23969  0.256348  0.023183
2018-01-01 01:00:00  34.909637  0.009235  160.25461  0.256226  0.189619
2018-01-01 02:00:00  35.195385  0.006260  156.44829  0.256226  0.243190
2018-01-01 03:00:00  35.590984  0.003489  154.54437  0.256104  0.195083
2018-01-01 04:00:00  35.827934  0.002314  154.16837  0.255981  0.132475

                          V2M      U50M      V50M      U850      V850
time
2018-01-01 00:00:00  0.507052  0.030755  1.167118 -0.678858  6.310610
2018-01-01 01:00:00  0.384886  0.424628  0.882620 -0.398818  6.162886
2018-01-01 02:00:00  0.296402  0.544786  0.681256 -0.217877  5.993750
2018-01-01 03:00:00  0.277474  0.383533  0.620761 -0.217092  5.911840
```

```
2018-01-01 04:00:00  0.275675  0.213028  0.631371 -0.108732  5.884082
```

3.4 MERRA-2 FLX group

[17]:
```
# - QV10M 10-meter_specific_humidity
# - QLML surface_specific_humidity
# - TQL total_precipitable_liquid_water
# - FRCAN areal_fraction_of_anvil_showers
# - HLML surface_layer_height
# - H1000 height_at_1000_mb
# - DISPH zero_plane_displacement_height
# - HLML surface_layer_height
# - U2M 2-meter_eastward_wind
# - V2M 2-meter_northward_wind
# - U50M eastward_wind_at_50_meters
# - V50M northward_wind_at_50_meters
# - U850 eastward_wind_at_850_hPa
# - V850 northward_wind_at_850_hPa
# - SPEED surface_wind_speed, m s-1
# - SPEEDMAX surface_wind_speed, m s-1
cols = ['QLML','FRCAN', 'HLML',  'SPEED', 'SPEEDMAX', 'RHOA']
```

[18]:
```
# similar to FLX group, select and filter out
flx = pd.read_csv('data/merra2_flx_hanoi_2018.csv',
                  parse_dates=['time'],
                  index_col=['time'])
flx.columns
flx = flx[cols]
flx.head()
```

[18]:
```
                               QLML     FRCAN       HLML     SPEED  SPEEDMAX  \
time
2018-01-01 00:00:00+07:00  0.007674  1.000000  63.991585  1.528854  1.593963
2018-01-01 01:00:00+07:00  0.007701  1.000000  63.907425  1.449315  1.488964
2018-01-01 02:00:00+07:00  0.007720  1.000000  63.832478  1.447317  1.493476
2018-01-01 03:00:00+07:00  0.007736  0.993164  63.766266  1.390683  1.483172
2018-01-01 04:00:00+07:00  0.007762  0.927490  63.718185  1.352254  1.409146

                               RHOA
time
2018-01-01 00:00:00+07:00  1.215108
2018-01-01 01:00:00+07:00  1.216159
2018-01-01 02:00:00+07:00  1.217125
2018-01-01 03:00:00+07:00  1.218085
2018-01-01 04:00:00+07:00  1.218972
```

[19]:
```
flx.index = flx.index.tz_localize(None)
```

4 Concatinate data

[20]:
```
# now will combine each dataframe using merge
# if we have the same column name for the index, say 'time', pd.concat() can␣
↪combine a list of dataframe
flx.head()
```

[20]:
```
                         QLML     FRCAN       HLML     SPEED  SPEEDMAX  \
time
2018-01-01 00:00:00  0.007674  1.000000  63.991585  1.528854  1.593963
2018-01-01 01:00:00  0.007701  1.000000  63.907425  1.449315  1.488964
2018-01-01 02:00:00  0.007720  1.000000  63.832478  1.447317  1.493476
2018-01-01 03:00:00  0.007736  0.993164  63.766266  1.390683  1.483172
2018-01-01 04:00:00  0.007762  0.927490  63.718185  1.352254  1.409146

                         RHOA
time
2018-01-01 00:00:00  1.215108
2018-01-01 01:00:00  1.216159
2018-01-01 02:00:00  1.217125
2018-01-01 03:00:00  1.218085
2018-01-01 04:00:00  1.218972
```

[21]:
```
# PM2.5 and SLV group
df = pd.merge(pm25, slv, right_index=True, left_index=True)
```

[22]:
```
# then FLX group to the two previous dfs
df = pd.merge(df, flx, right_index=True, left_index=True, how='left')
```

[23]:
```
# finally data from Noibai Intl Airpot
df = pd.merge(df, nb, right_index=True, left_index=True, how='left')
```

[24]:
```
df.describe()
```

[24]:
```
              pm25       T2MWET       T2MDEW          T2M        QV10M  \
count  8116.000000  8116.000000  8116.000000  8116.000000  8116.000000
mean     40.758736   293.294133   293.293231   296.359950     0.015057
std      31.501209     5.838267     5.840456     6.123929     0.004624
min       0.000000   271.761080   271.669980   275.719970     0.002889
25%      19.000000   290.389635   290.387140   292.489165     0.011934
50%      32.000000   294.918210   294.917055   297.664350     0.015914
75%      52.000000   297.960532   297.960155   300.581087     0.019105
max     323.000000   301.152130   301.144470   309.323800     0.022890
```

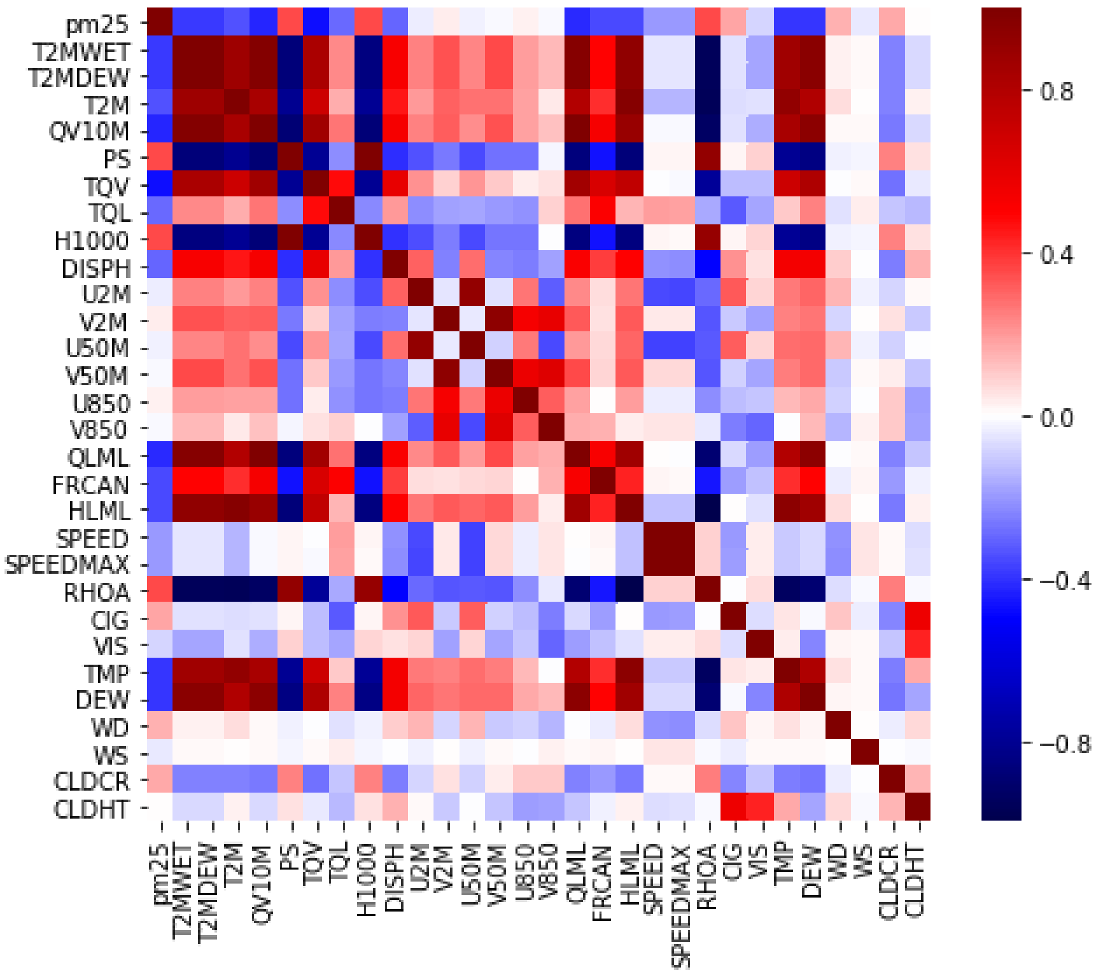
```
                  PS           TQV          TQL        H1000        DISPH  ...  \
count    8116.000000  8116.000000  8116.000000  8116.000000  8116.000000  ...
mean   100091.307728    46.593698     0.078603    94.100740     0.283273  ...
std       728.034422    15.289697     0.082412    63.071034     0.065183  ...
min     98338.780000    12.691939     0.000000   -62.420760     0.165833  ...
25%     99467.157500    34.854736     0.011534    39.747400     0.221359  ...
50%    100156.612500    46.939671     0.053711   101.305442     0.308167  ...
75%    100572.849000    60.093417     0.118324   137.084377     0.340698  ...
max    102189.990000    79.528755     0.451294   266.238680     0.353638  ...

          SPEEDMAX         RHOA           CIG            VIS          TMP  \
count  8092.000000  8092.000000   5088.000000    7811.000000  7811.000000
mean      4.429708     1.166763   4882.208137  178345.484317    24.536039
std       1.814446     0.033969   8266.805156  374942.058550     5.522560
min       0.353847     1.101529     91.000000     550.000000     9.000000
25%       3.111255     1.139971    610.000000    5000.000000    21.000000
50%       4.308590     1.161659    945.000000    9000.000000    25.000000
75%       5.540783     1.187278   1494.000000    9999.000000    28.000000
max      14.158017     1.276222  22000.000000  999999.000000    39.000000

               DEW           WD           WS        CLDCR        CLDHT
count  7811.000000  7811.000000  7811.000000  5502.000000  5502.000000
mean     20.224299   219.991806     3.107528     0.376863   609.800981
std       6.008754   272.325200    16.012484     0.190057   348.940325
min      -1.000000    10.000000     0.000000     0.200000    61.000000
25%      17.000000    70.000000     2.100000     0.200000   305.000000
50%      22.000000   120.000000     2.600000     0.400000   549.000000
75%      25.000000   270.000000     3.600000     0.400000   914.000000
max      30.000000   999.000000   999.900000     0.800000  1494.000000

[8 rows x 30 columns]
```

[25]:
```
df.columns
```

[25]:
```
Index(['pm25', 'T2MWET', 'T2MDEW', 'T2M', 'QV10M', 'PS', 'TQV', 'TQL', 'H1000',
       'DISPH', 'U2M', 'V2M', 'U50M', 'V50M', 'U850', 'V850', 'QLML', 'FRCAN',
       'HLML', 'SPEED', 'SPEEDMAX', 'RHOA', 'CIG', 'VIS', 'TMP', 'DEW', 'WD',
       'WS', 'CLDCR', 'CLDHT'],
      dtype='object')
```

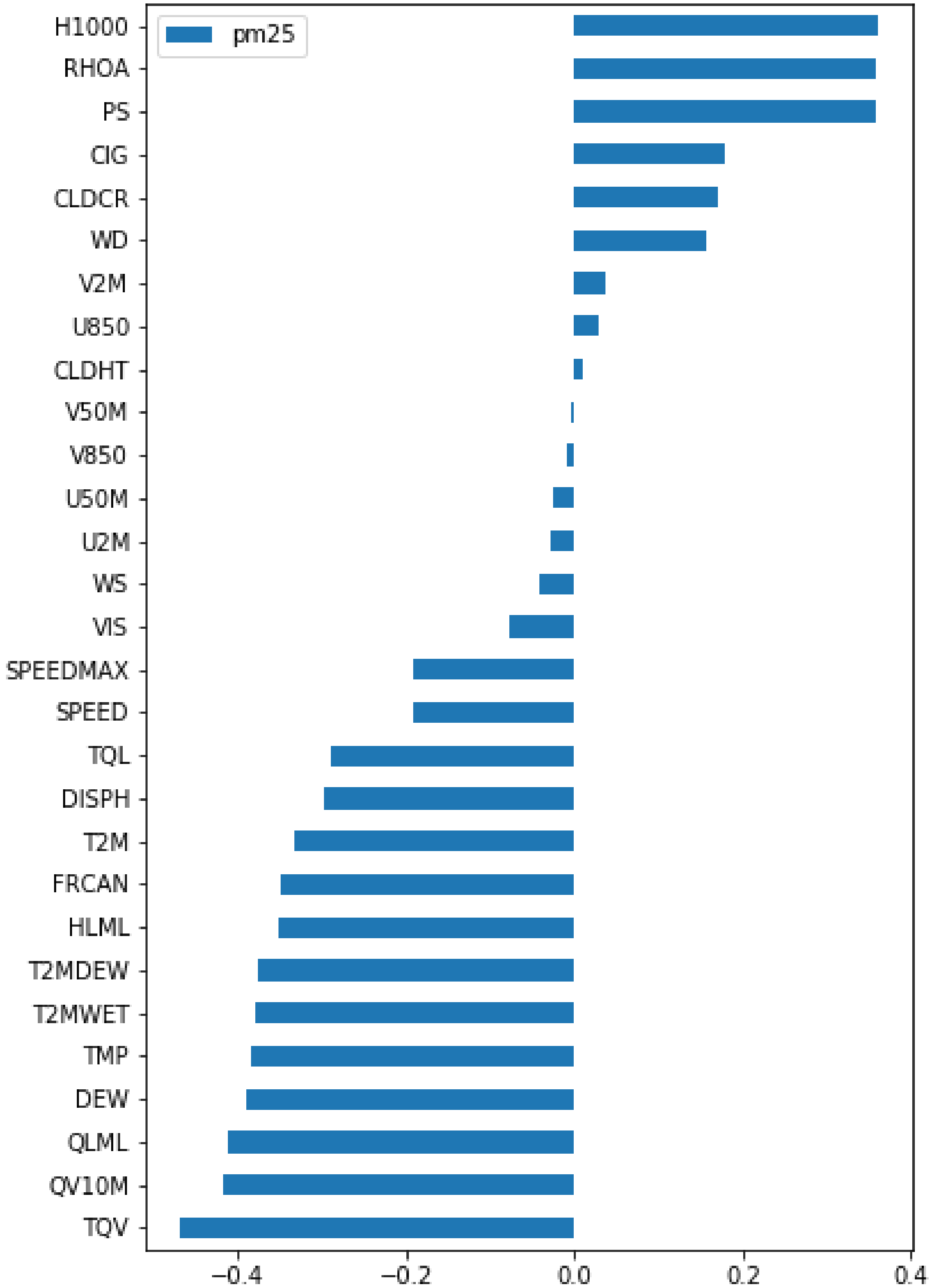
5 Correlation and filter out

[27]:
```
# we will do several steps to filter out dependent columns (well correlated␣
↪with existing columns)
fig, ax = plt.subplots(figsize=(7,6))
sns.heatmap(df.corr(), cmap='seismic')
fig.tight_layout()
fig.savefig('img/2020Aug-corr-heatmap.png', dpi=120, optimize=True)
```



[28]:
```
fig, ax = plt.subplots(figsize=(6,10))
df.corr()['pm25'].sort_values().to_frame().drop('pm25').plot.barh(ax=ax)
```

[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1c04686a20>



pm25
H1000
RHOA
PS
CIG
CLDCR
WD
V2M
U850
CLDHT
V50M
V850
U50M
U2M
WS
VIS
SPEEDMAX
SPEED
TQL
DISPH
T2M
FRCAN
HLML
T2MDEW
T2MWET
TMP
DEW
QLML
QV10M
TQV
−0.4
−0.2
0.0
0.2
0.4

5.1 let get wind speed data in different formats

[29]:
```
# let detour and get wind in speed and direction (degree from 2.4 exercise)
dfw = pd.read_csv('data/merra2_hanoi_2018_wind_converted.csv',
                  parse_dates=['Unnamed: 0'],
                  index_col=['Unnamed: 0'])
dfw.head()
```

[29]:
```
                          v_2m        d_2m      v_10m        d_10m     v_50m  \
2018-01-01 00:00:00  0.507582  182.617815  0.820326  182.212835  1.167523
2018-01-01 01:00:00  0.429060  206.227729  0.691324  206.109338  0.979452
2018-01-01 02:00:00  0.383400  219.367919  0.617108  218.824390  0.872296
2018-01-01 03:00:00  0.339189  215.109724  0.525337  212.637784  0.729687
2018-01-01 04:00:00  0.305853  205.666509  0.477588  200.748694  0.666341

                          d_50m     v_850       d_850      v_500       d_500  \
2018-01-01 00:00:00  181.509485  6.347019  173.860069  11.257534  276.654649
2018-01-01 01:00:00  205.692157  6.175777  176.297386  11.215821  280.471385
2018-01-01 02:00:00  218.648565  5.997708  177.918171  10.827758  282.437516
2018-01-01 03:00:00  211.709574  5.915825  177.896957  10.295540  282.633150
2018-01-01 04:00:00  198.644717  5.885087  178.941354   9.500886  280.860531

                         v_250       d_250
2018-01-01 00:00:00  23.196986  242.232266
2018-01-01 01:00:00  23.475563  241.081727
2018-01-01 02:00:00  23.842929  240.022727
2018-01-01 03:00:00  24.137831  239.375860
2018-01-01 04:00:00  24.453376  238.787131
```

[30]:
```
dfw.columns
```

[30]:
```
Index(['v_2m', 'd_2m', 'v_10m', 'd_10m', 'v_50m', 'd_50m', 'v_850', 'd_850',
       'v_500', 'd_500', 'v_250', 'd_250'],
      dtype='object')
```

[31]:
```
cols = ['v_2m', 'd_2m', 'v_50m', 'd_50m', 'v_850', 'd_850']
dfw = dfw[cols]
```

[32]:
```
# and combine all data in one place
df = pd.merge(df, dfw, right_index=True, left_index=True, how='left')
```

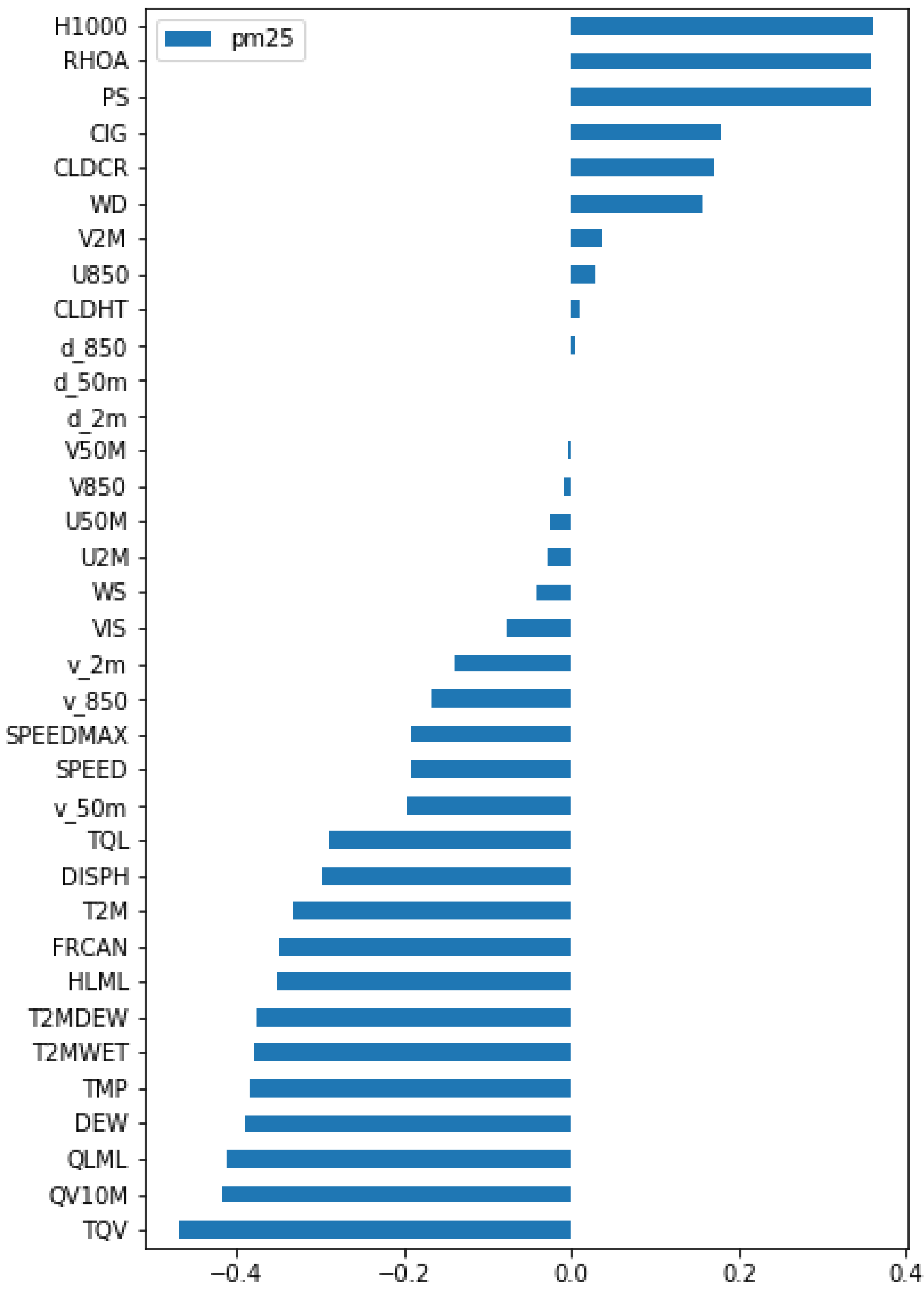
[33]:
```
df.columns
```

[33]:
```
Index(['pm25', 'T2MWET', 'T2MDEW', 'T2M', 'QV10M', 'PS', 'TQV', 'TQL', 'H1000',
       'DISPH', 'U2M', 'V2M', 'U50M', 'V50M', 'U850', 'V850', 'QLML', 'FRCAN',
       'HLML', 'SPEED', 'SPEEDMAX', 'RHOA', 'CIG', 'VIS', 'TMP', 'DEW', 'WD',
       'WS', 'CLDCR', 'CLDHT', 'v_2m', 'd_2m', 'v_50m', 'd_50m', 'v_850',
```

```
       'd_850'],
      dtype='object')
```

[34]:
```
fig, ax = plt.subplots(figsize=(6,10))
df.corr()['pm25'].sort_values().to_frame().drop('pm25').plot.barh(ax=ax)
```

[34]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1c03f89748>
```



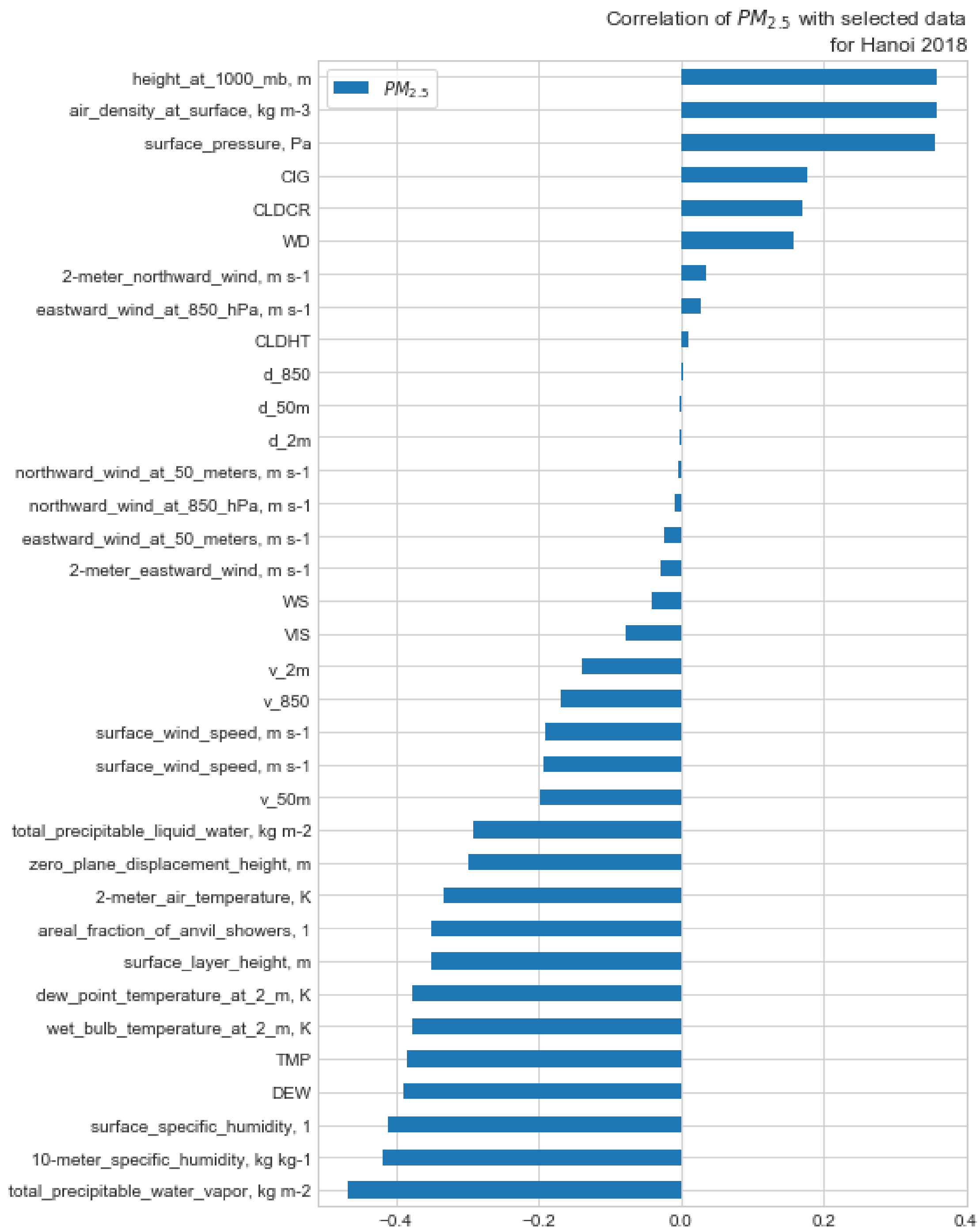
[35]:
```
df.index.rename('DATE', inplace=True)
```

[36]:
```python
df.rename(columns={'pm25': 'PM2.5'}, inplace=True)
```

[37]:
```python
# save it, and we can use it later one (if needed)
df.to_csv('data/comb_PM25_Hanoi_2018.csv')
```

5.2 Chart

[38]:
```python
# and recreate the figure above with the standard name
plt.style.use('seaborn-whitegrid')
fig, ax = plt.subplots(figsize=(8,10))
df.corr()['PM2.5'].sort_values().to_frame().dropna().drop('PM2.5').plot.
 ↪barh(ax=ax)
ax.legend(['$PM_{2.5}$'], frameon=True)
labels = [item.get_text() for item in ax.get_yticklabels()]

# looking for a standard name for each abbreviation
new_label = dict()
for label in labels:
    if label in list(name_.keys()):
        new_label[label] = name_[label]
    else:
        new_label[label] = label
ax.set_yticklabels(new_label.values())
plt.title('Correlation of $PM_{2.5}$ with selected data\nfor Hanoi 2018',␣
 ↪loc='right')
plt.tight_layout()
```



- too many, and some duplicated data, let define a list of columns to be dropped

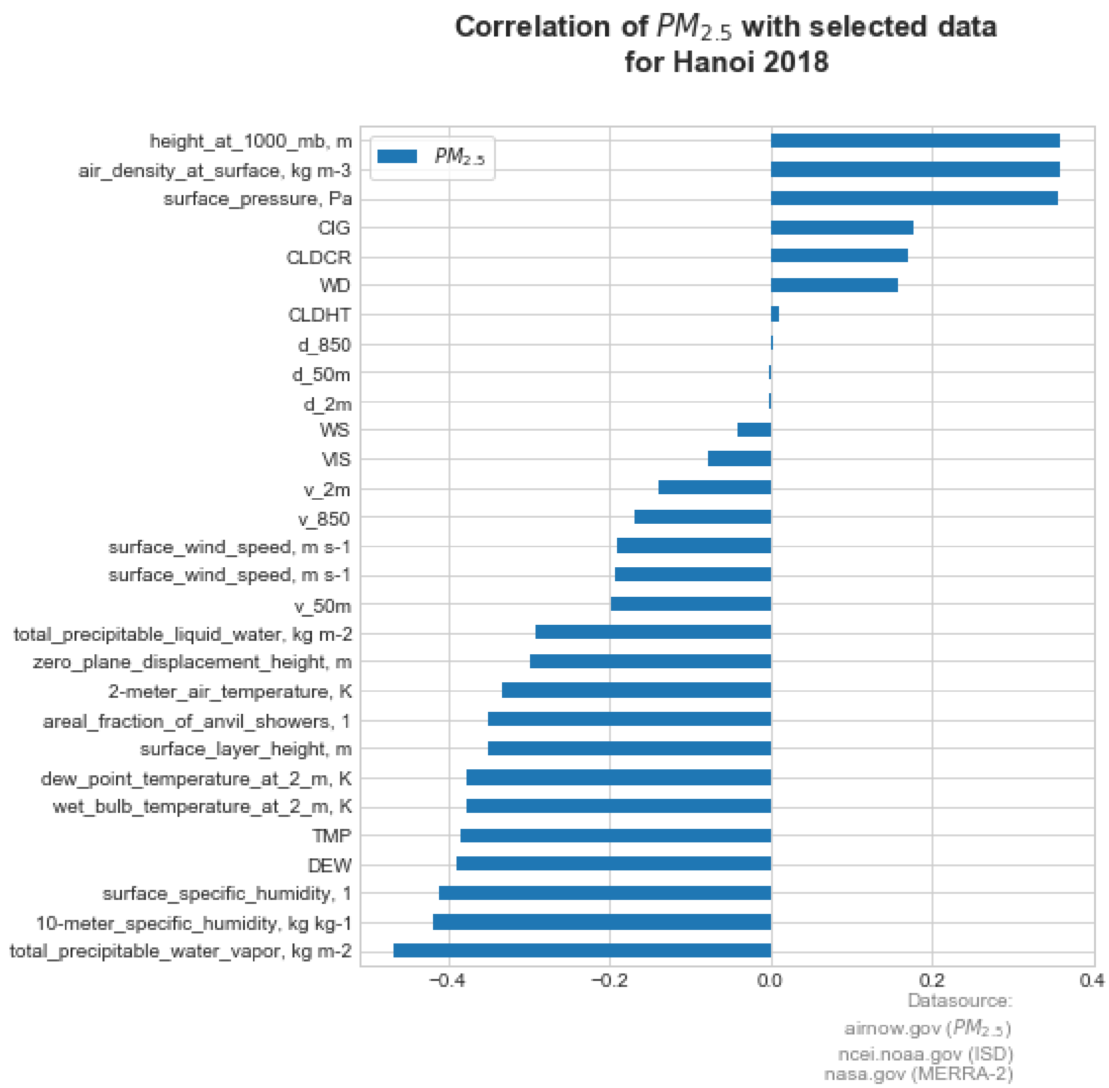
[39]:
```
df.drop(columns=['U2M', 'V2M', 'U50M', 'V50M', 'U850', 'V850'], inplace=True)
```

[40]:
```
# save data as always
df.to_csv('data/comb_PM25_wind_Hanoi_2018.csv')
```

[41]:
```
df.columns
```

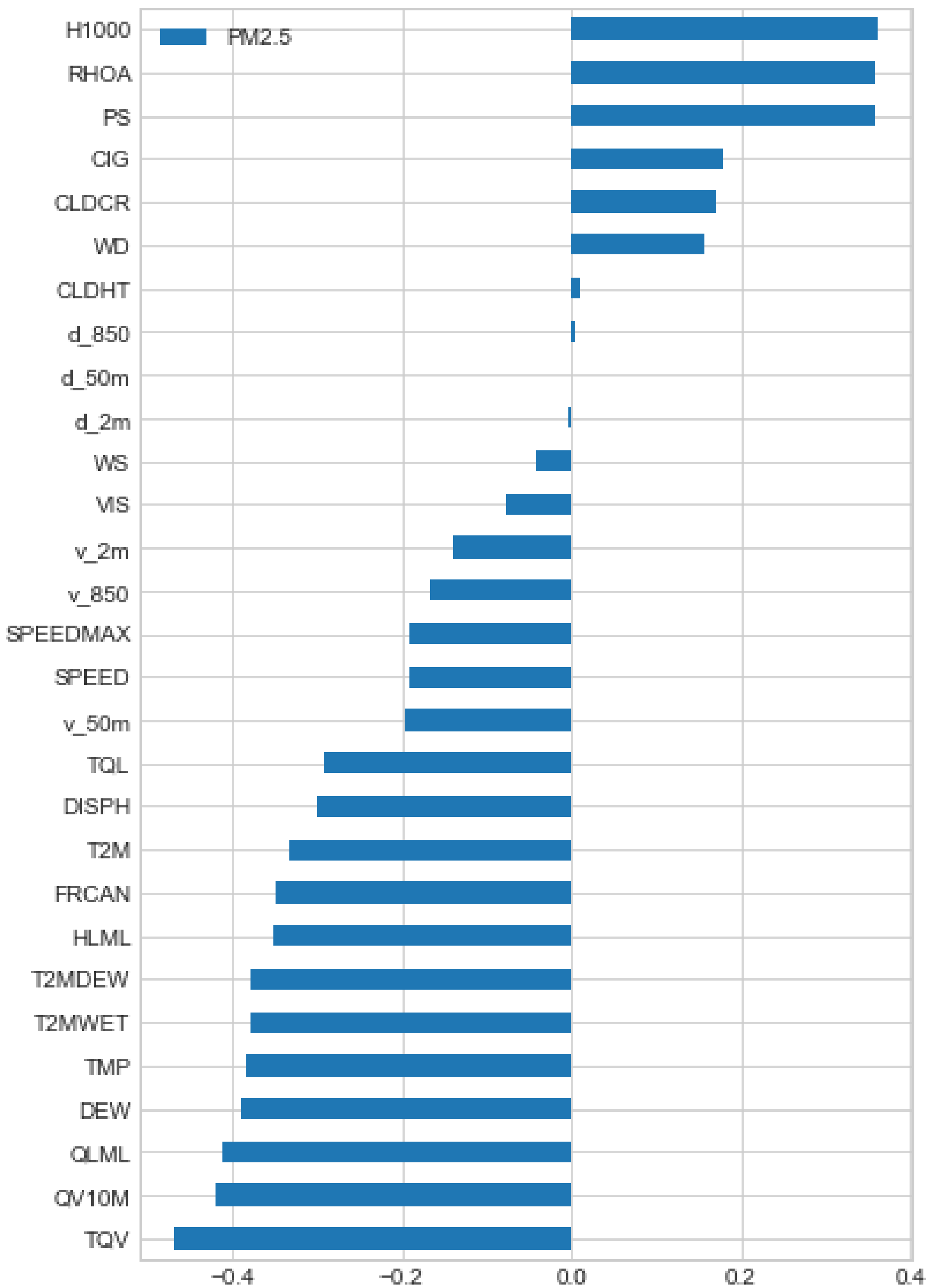
[41]:
```
Index(['PM2.5', 'T2MWET', 'T2MDEW', 'T2M', 'QV10M', 'PS', 'TQV', 'TQL',
       'H1000', 'DISPH', 'QLML', 'FRCAN', 'HLML', 'SPEED', 'SPEEDMAX', 'RHOA',
       'CIG', 'VIS', 'TMP', 'DEW', 'WD', 'WS', 'CLDCR', 'CLDHT', 'v_2m',
       'd_2m', 'v_50m', 'd_50m', 'v_850', 'd_850'],
      dtype='object')
```

[42]:
```
# and recreate the figure above with the standard name
fig, ax = plt.subplots(figsize=(8,8))
df.corr()['PM2.5'].sort_values().to_frame().dropna().drop('PM2.5').plot.
 ↪barh(ax=ax)
ax.legend(['$PM_{2.5}$'], frameon=True)
labels = [item.get_text() for item in ax.get_yticklabels()]

# looking for a standard name for each abbreviation
new_label = dict()
for label in labels:
    if label in list(name_.keys()):
        new_label[label] = name_[label]
    else:
        new_label[label] = label
ax.set_yticklabels(new_label.values())
ax.set_title('Correlation of $PM_{2.5}$ with selected data\nfor Hanoi 2018',
          y=1.05, fontsize=15, weight='bold')
fig.subplots_adjust(bottom=0.1)
fig.text(0.9,.04,s='Datasource:\n airnow.gov ($PM_{2.5}$)\nncei.noaa.gov␣
 ↪(ISD)\nnasa.gov (MERRA-2)',
         ha='right', color='gray')
plt.tight_layout(rect=(0, 0.1, 1,1))
# plt.tight_layout()
plt.savefig('img/2020Aug-PM25-allin.png', dpi=120, optimize=True)
```




```
[43]: # less inputs
fig, ax = plt.subplots(figsize=(6,10))
df.corr()['PM2.5'].sort_values().to_frame().drop('PM2.5').plot.barh(ax=ax)
```

```
[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1c045a5a90>
```



[44]:
```
# some test to make sure we know they these are highly-dependent,
df.corr()['T2M']['TMP']
```

[44]: 0.9262485548028426

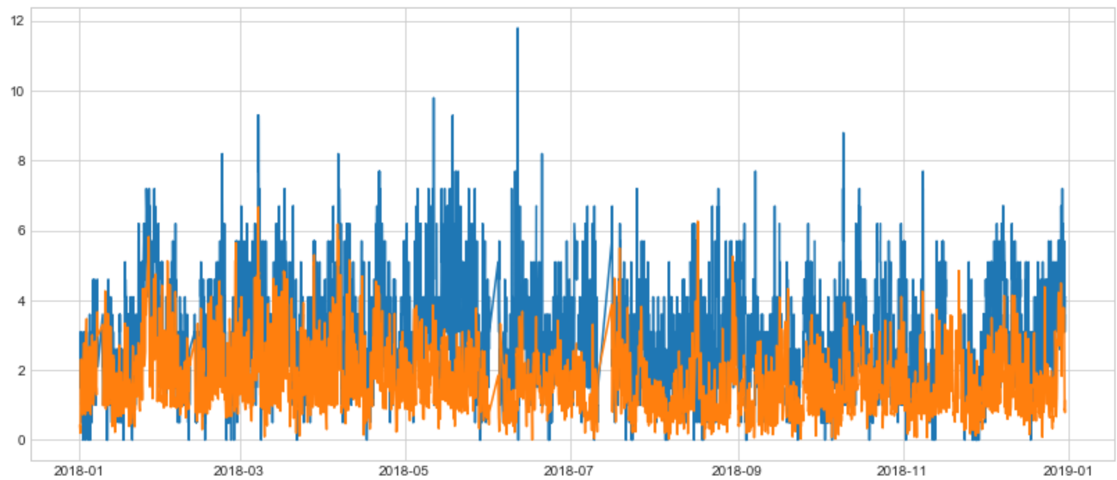
[45]:
```
df.corr()['WS']['v_2m']
```

[45]: 0.05110859128542097

[46]:
```
# some cleaning
df.loc[df.WS>20, 'WS'] = None
```

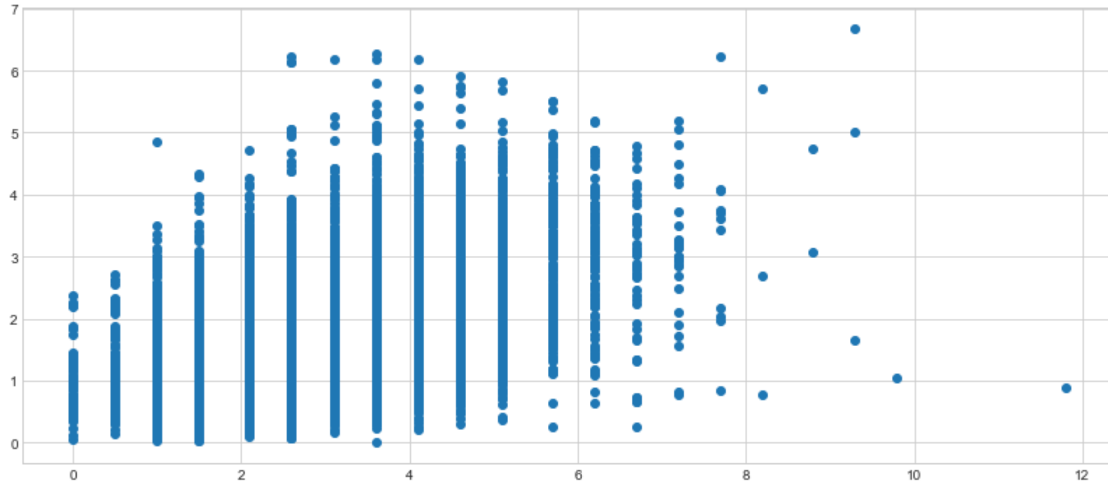
[47]:
```
# not always as expected, one from observed station, another from a ground␣
↪observed stations
plt.plot(df.index, df.WS)
plt.plot(df.index, df.v_2m)
```

[47]: [<matplotlib.lines.Line2D at 0x7f1c045166a0>]



[48]:
```
# ground station reported with coarse resolution, 0.5m/s increment
plt.scatter(df.WS, df.v_2m)
```

[48]: <matplotlib.collections.PathCollection at 0x7f1c0460e1d0>

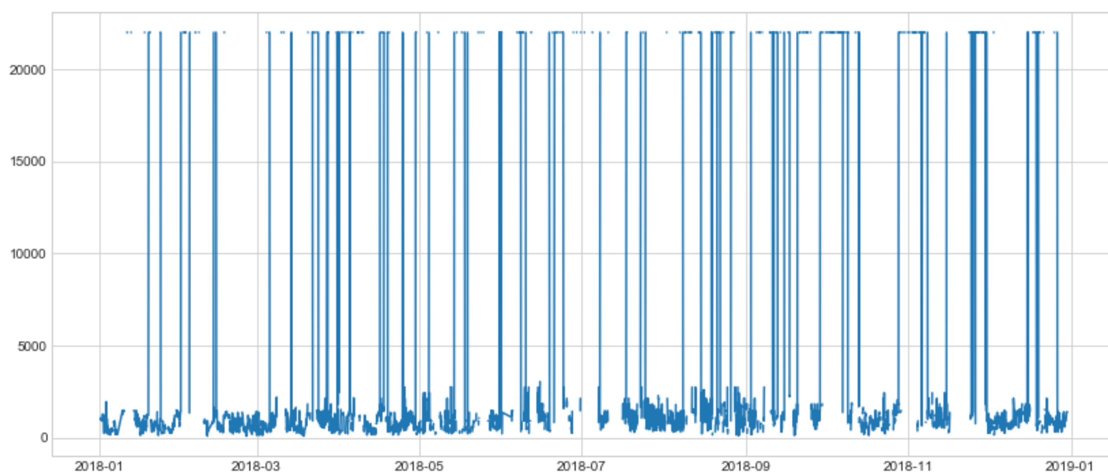


[49]:
```
df.corr()['T2MDEW']['T2MWET']
```

[49]: 0.9999992520668848

[50]:
```
plt.plot(df.index, df.CIG)
```

[50]: [<matplotlib.lines.Line2D at 0x7f1c04008c50>]



[51]:
```
df.corr()['SPEED'].sort_values()
```

[51]:
```
DISPH      -0.211756
WD         -0.211602
PM2.5      -0.192395
CIG        -0.192119
```

```
T2M         -0.140154
HLML        -0.115054
TMP         -0.102803
d_2m        -0.097074
d_50m       -0.094817
DEW         -0.069703
d_850       -0.064683
CLDHT       -0.060311
T2MDEW      -0.047923
T2MWET      -0.047921
QV10M       -0.006900
TQV         -0.000806
QLML         0.004860
CLDCR        0.015283
FRCAN        0.019606
H1000        0.020673
PS           0.024060
VIS          0.038930
RHOA         0.094989
TQL          0.192514
v_850        0.448324
WS           0.499856
v_2m         0.747837
v_50m        0.992749
SPEEDMAX     0.996841
SPEED        1.000000
Name: SPEED, dtype: float64
```

[52]:
```
df.corr()['DEW']['T2MDEW']
```

[52]:
```
0.9566090636828841
```

[53]:
```
df.columns
```

[53]:
```
Index(['PM2.5', 'T2MWET', 'T2MDEW', 'T2M', 'QV10M', 'PS', 'TQV', 'TQL',
       'H1000', 'DISPH', 'QLML', 'FRCAN', 'HLML', 'SPEED', 'SPEEDMAX', 'RHOA',
       'CIG', 'VIS', 'TMP', 'DEW', 'WD', 'WS', 'CLDCR', 'CLDHT', 'v_2m',
       'd_2m', 'v_50m', 'd_50m', 'v_850', 'd_850'],
      dtype='object')
```

[54]:
```
cols_drop = ['T2MWET', 'SPEED', 'SPEEDMAX', 'DEW', 'TMP']
```

[55]:
```
df.columns
```

[55]:
```
Index(['PM2.5', 'T2MWET', 'T2MDEW', 'T2M', 'QV10M', 'PS', 'TQV', 'TQL',
       'H1000', 'DISPH', 'QLML', 'FRCAN', 'HLML', 'SPEED', 'SPEEDMAX', 'RHOA',
       'CIG', 'VIS', 'TMP', 'DEW', 'WD', 'WS', 'CLDCR', 'CLDHT', 'v_2m',
```

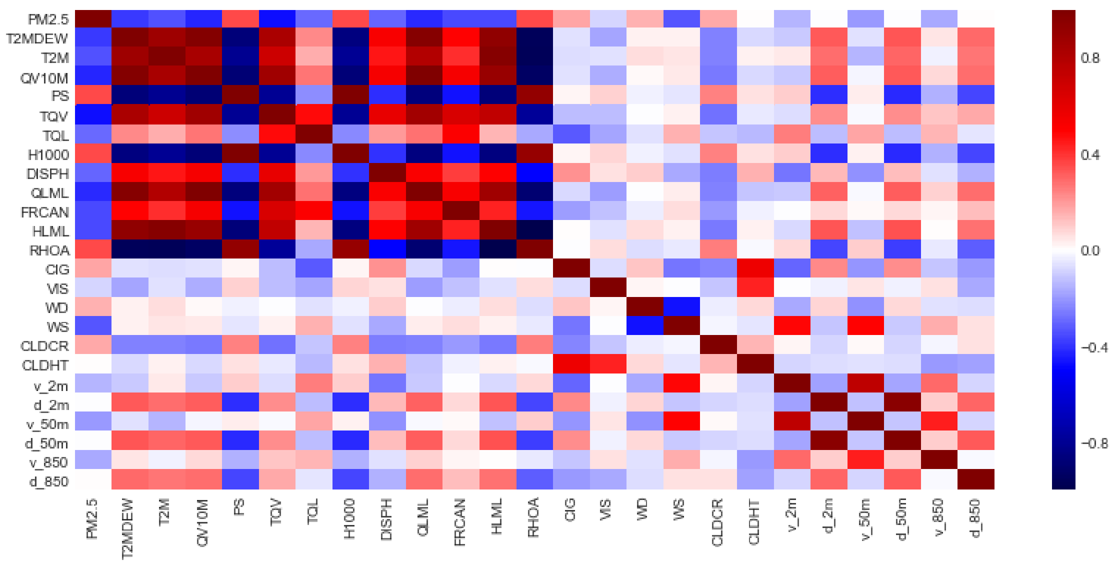
```
        'd_2m', 'v_50m', 'd_50m', 'v_850', 'd_850'],
       dtype='object')
```

[56]:
```
df.drop(columns=cols_drop, inplace=True)
```

[57]:
```
df.to_csv('data/comb_PM25_wind_Hanoi_2018_v1.csv')
```

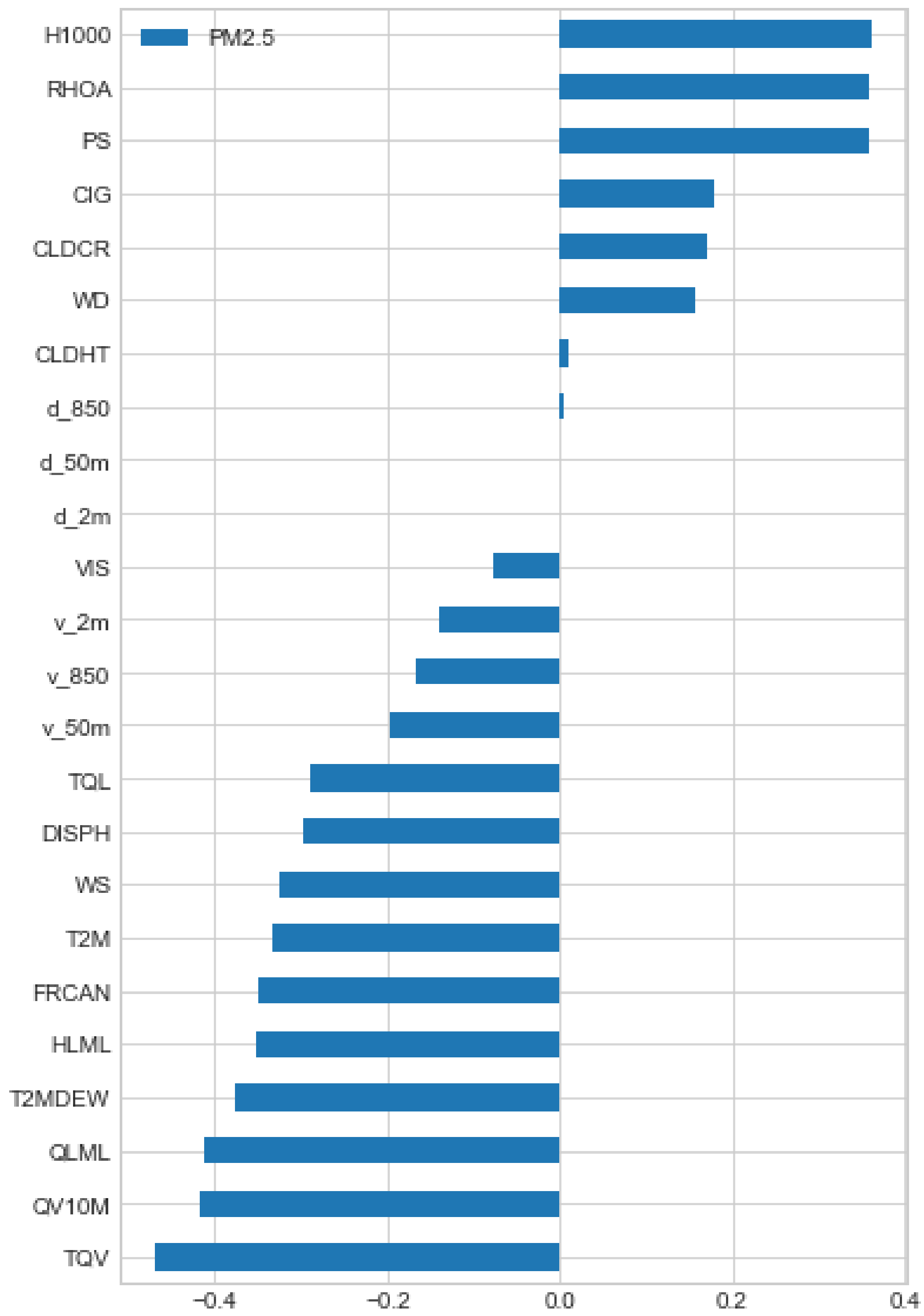
[58]:
```
# let see how the heatmap again
sns.heatmap(df.corr(), cmap='seismic')
```

[58]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f1c04568358>`



[59]:
```
# and correlation only with PM2.5
fig, ax = plt.subplots(figsize=(6,10))
df.corr()['PM2.5'].sort_values().to_frame().drop('PM2.5').plot.barh(ax=ax)
```

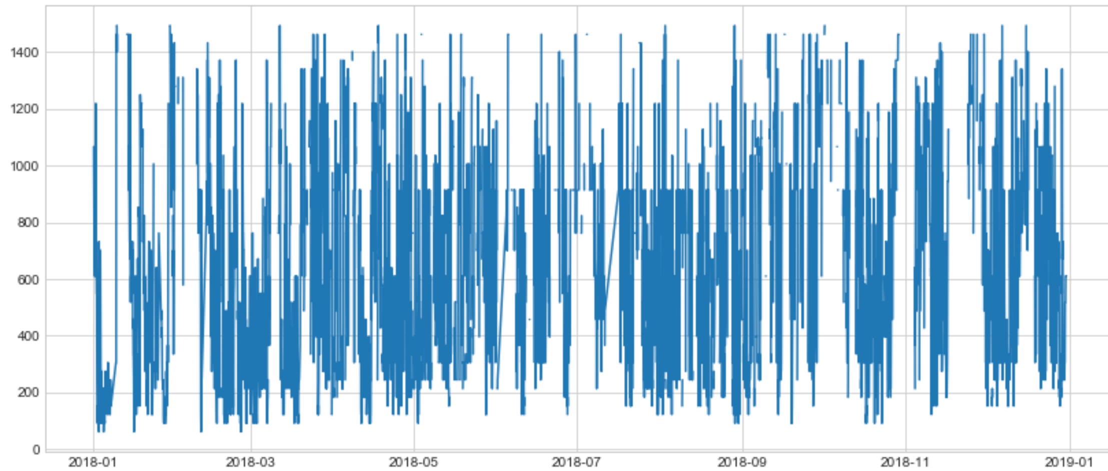
[59]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f1c042fb278>`




```
[60]: colsd = ['QV10M', 'CLDHT', 'QLML']
```

[61]:
```
# Height to the lowest cloud is very weak, let see data before we drop it
plt.plot(df.index, df.CLDHT)
```

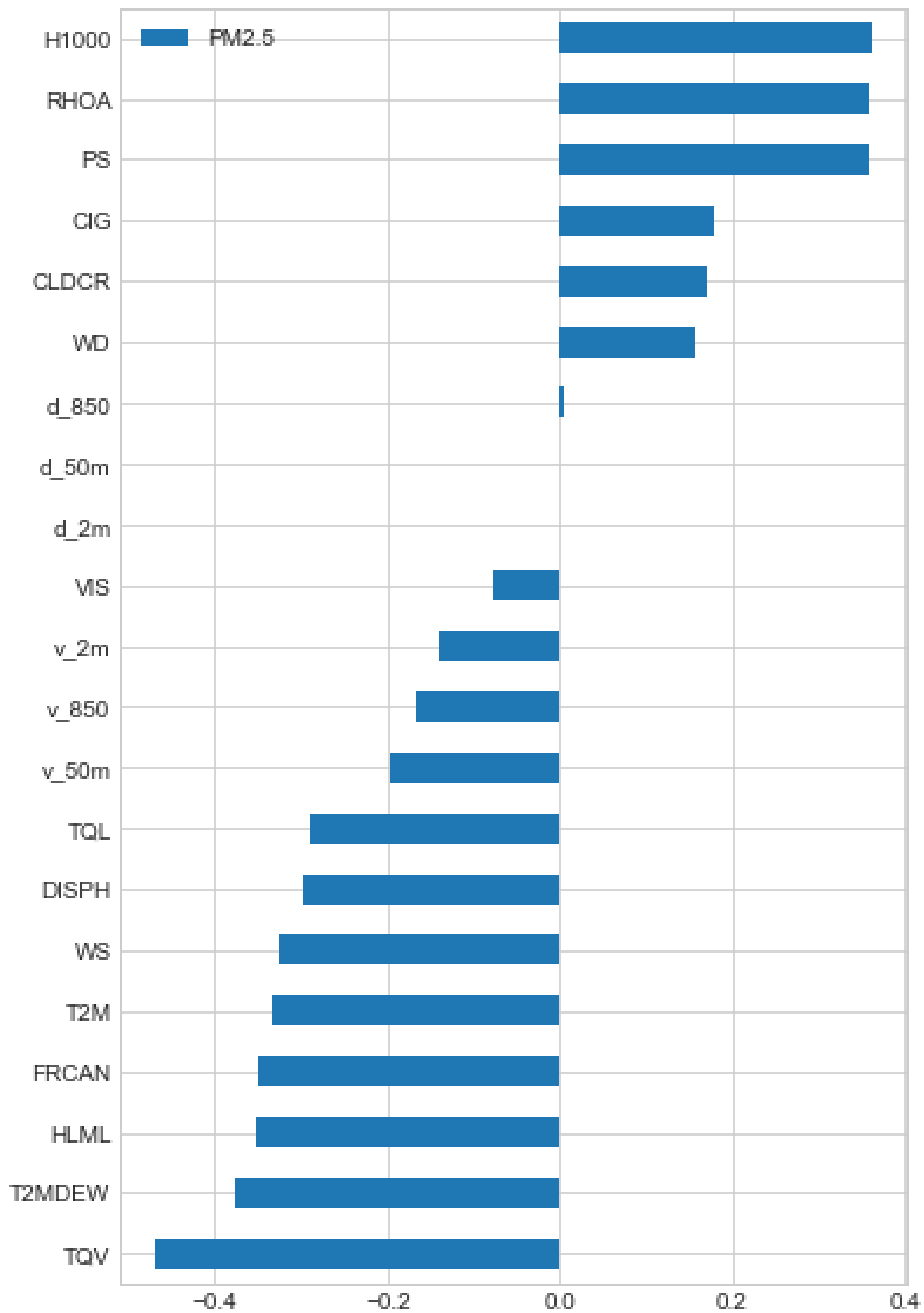
[61]: [<matplotlib.lines.Line2D at 0x7f1c02143940>]



[62]:
```
df.drop(columns=colsd, inplace=True)
```

[63]:
```
fig, ax = plt.subplots(figsize=(6,10))
df.corr()['PM2.5'].sort_values().to_frame().drop('PM2.5').plot.barh(ax=ax)
```

[63]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1c020657f0>



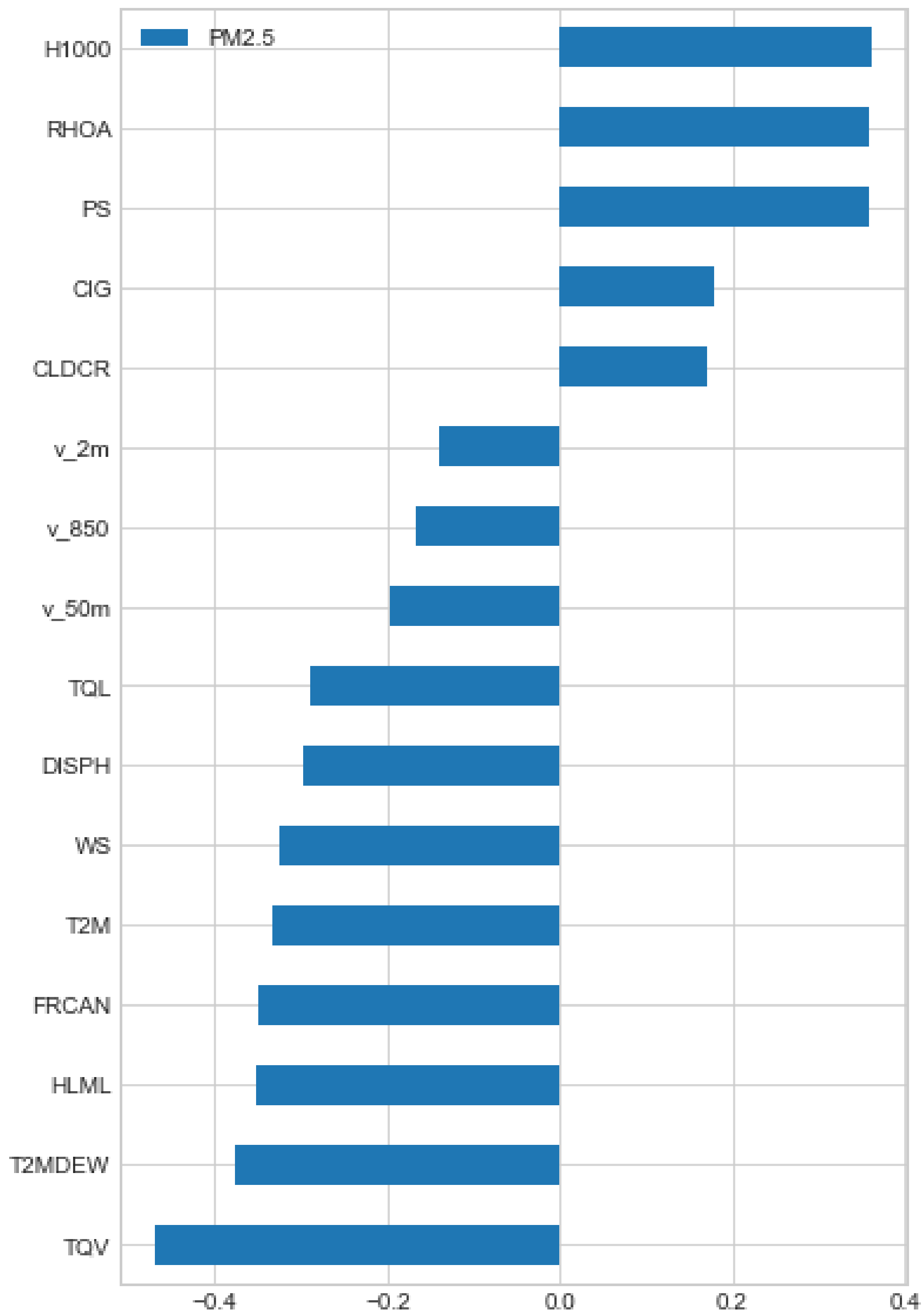
[64]:
```
df.to_csv('data/comb_PM25_wind_Hanoi_2018_v2.csv')
```

[65]:
```
more_cols = ['d_850', 'd_50m', 'd_2m', 'VIS', 'WD']
```

[66]:
```
df.drop(columns=more_cols, inplace=True)
```

[67]:
```
fig, ax = plt.subplots(figsize=(6,10))
df.corr()['PM2.5'].sort_values().to_frame().drop('PM2.5').plot.barh(ax=ax)
```

[67]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1c041c1710>



[68]:
```
df['T2MDEW'] = df['T2MDEW'] - 273.15
df['T2M'] = df['T2M'] - 273.15
```

[69]:
```python
df.to_csv('data/comb_PM25_wind_Hanoi_2018_v3.csv')
```

[70]:
```python
df.describe()
```

[70]:

	PM2.5	T2MDEW	T2M	PS	TQV
count	8116.000000	8116.000000	8116.000000	8116.000000	8116.000000
mean	40.758736	20.143231	23.209950	100091.307728	46.593698
std	31.501209	5.840456	6.123929	728.034422	15.289697
min	0.000000	-1.480020	2.569970	98338.780000	12.691939
25%	19.000000	17.237140	19.339165	99467.157500	34.854736
50%	32.000000	21.767055	24.514350	100156.612500	46.939671
75%	52.000000	24.810155	27.431088	100572.849000	60.093417
max	323.000000	27.994470	36.173800	102189.990000	79.528755

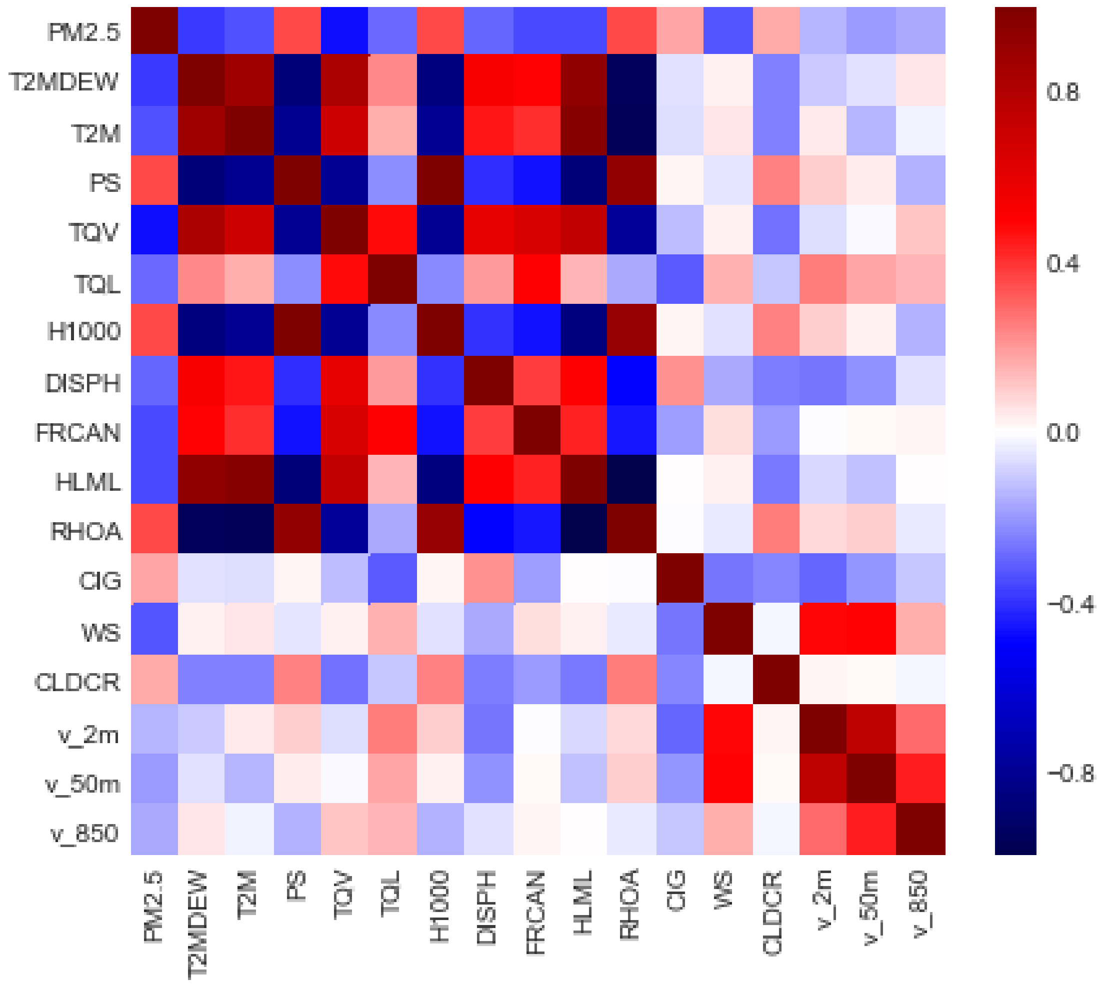
	TQL	H1000	DISPH	FRCAN	HLML
count	8116.000000	8116.000000	8116.000000	8092.000000	8092.000000
mean	0.078603	94.100740	0.283273	0.521452	66.146428
std	0.082412	63.071034	0.065183	0.374627	1.463402
min	0.000000	-62.420760	0.165833	0.000000	61.526400
25%	0.011534	39.747400	0.221359	0.148132	65.226177
50%	0.053711	101.305442	0.308167	0.511841	66.474025
75%	0.118324	137.084377	0.340698	0.931946	67.216239
max	0.451294	266.238680	0.353638	1.000000	68.976160

	RHOA	CIG	WS	CLDCR	v_2m
count	8092.000000	5088.000000	7809.000000	5502.000000	8116.000000
mean	1.166763	4882.208137	2.852235	0.376863	1.725790
std	0.033969	8266.805156	1.375927	0.190057	1.000010
min	1.101529	91.000000	0.000000	0.200000	0.005149
25%	1.139971	610.000000	2.100000	0.200000	0.948639
50%	1.161659	945.000000	2.600000	0.400000	1.486214
75%	1.187278	1494.000000	3.600000	0.400000	2.322390
max	1.276222	22000.000000	11.800000	0.800000	6.670765

	v_50m	v_850
count	8116.000000	8116.000000
mean	3.989913	5.674146
std	1.756854	3.112177
min	0.022971	0.059713
25%	2.751847	3.458131
50%	3.924952	5.236085
75%	5.082285	7.178591
max	12.574891	26.583689

[71]:
```python
# much smaller dataset now
fig, ax = plt.subplots(figsize=(7,6))
sns.heatmap(df.corr(), cmap='seismic')
```

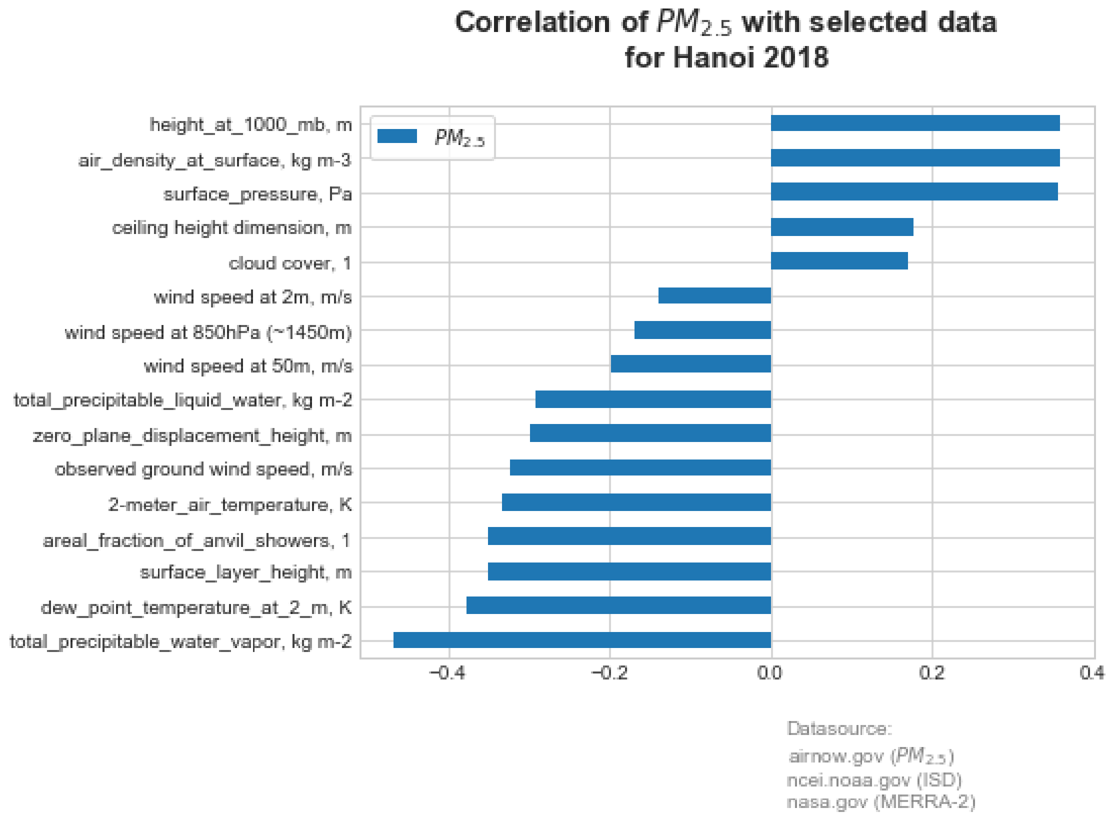
[71]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1c03df4dd8>
```



[72]:
```
# let add name to nameplace holder,
name_['CIG'] = 'ceiling height dimension, m'
name_['CLDCR'] = 'cloud cover, 1'
name_['v_2m'] = 'wind speed at 2m, m/s'
name_['v_50m'] = 'wind speed at 50m, m/s'
name_['v_850'] = 'wind speed at 850hPa (~1450m)'
name_['WS'] = 'observed ground wind speed, m/s'
```

6 Nudgets are here!

[73]:
```
# and recreate the figure above with the standard name
fig, ax = plt.subplots(figsize=(8,6))
df.corr()['PM2.5'].sort_values().to_frame().dropna().drop('PM2.5').plot.
↪barh(ax=ax)
ax.legend(['$PM_{2.5}$'], frameon=True)
labels = [item.get_text() for item in ax.get_yticklabels()]

# looking for a standard name for each abbreviation
new_label = dict()
for label in labels:
    if label in list(name_.keys()):
        new_label[label] = name_[label]
    else:
        new_label[label] = label
ax.set_yticklabels(new_label.values())
ax.set_title('Correlation of $PM_{2.5}$ with selected data\nfor Hanoi 2018',
          y=1.05, fontsize=15, weight='bold')
fig.subplots_adjust(bottom=0.1)
fig.text(0.7,.03,s='Datasource:\nairnow.gov ($PM_{2.5}$)\nncei.noaa.gov␣
↪(ISD)\nnasa.gov (MERRA-2)',
         ha='left', color='gray')
plt.tight_layout(rect=(0, 0.15, 1,1))
# plt.tight_layout()
plt.savefig('img/2020Aug-PM25-selected.png', dpi=120, optimize=True)
```



- OK, now we are ready to rock some prediction

[74]:
```
df.shape
```

[74]:
```
(8116, 17)
```

[75]:
```
new_label
```

[75]:
```
{'TQV': 'total_precipitable_water_vapor, kg m-2',
 'T2MDEW': 'dew_point_temperature_at_2_m, K',
 'HLML': 'surface_layer_height, m',
 'FRCAN': 'areal_fraction_of_anvil_showers, 1',
 'T2M': '2-meter_air_temperature, K',
 'WS': 'observed ground wind speed, m/s',
 'DISPH': 'zero_plane_displacement_height, m',
 'TQL': 'total_precipitable_liquid_water, kg m-2',
 'v_50m': 'wind speed at 50m, m/s',
 'v_850': 'wind speed at 850hPa (~1450m)',
 'v_2m': 'wind speed at 2m, m/s',
 'CLDCR': 'cloud cover, 1',
 'CIG': 'ceiling height dimension, m',
```

```
 'PS': 'surface_pressure, Pa',
 'RHOA': 'air_density_at_surface, kg m-3',
 'H1000': 'height_at_1000_mb, m'}
```

[ ]:

```

```