Bijlage 1 - data cleaning

In de van Rijkswaterstaat ontvangen dataset staat veel informatie die voor de doeleinden van dit onderzoek niet relevant zijn. Onderstaande stappen voeren werkzaamheden uit aan de dataset, die tot doel hebben irrelevante data te verwijderen en enkele andere velden aan te passen.

In [ ]:

```
#Libraries
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline
```

In [ ]:

```
#Read initial file
df = pd.read_csv("../data/data_tot.csv", encoding='latin1')
```

In [ ]:

```
def proper_datetime():
    '''Get a less useless datetime field, the one the DF comes with isn't i
deal'''
    df['DATETIME'] = pd.to_datetime(df.DATUM+df.TIJD, format='%Y-%m-%d\'%H%
M')
def fix_order(df):
    '''We want the datetime field up front'''
    cols = df.columns.tolist()
    cols = cols[-1:] + cols[:-1]
    df = df[cols]
    return df
```

In [ ]:

```
def drop_bad_columns(df):
    '''Get rid of columns that contain bad or irrelevant data'''
    df.drop(columns=['Unnamed: 0',
                     'knmi_STN',
                    'DATUM',
                    'TIJD',
                    'DOM',
                    'BEW',
                    'SGK',
                    'ORG',
                    'IVS',
                    'BTNOMS',
                    'BTXCOD',
                     'BTXOMS',
                     'GBDOMS',
                     'OGIOMS',
                     'ANIOMS',
                     'BHIOMS',
```

```
                       'BMIOMS',
                       'VATOMS',
                      'LOC:TYPE',
                      'SYS',
                      'SYSOMS',
                      'TYP',
                      'TYPOMS',
                      'TYD:BEGINDAT',
                      'TYD:BEGINTYD',
                      'TYD:EINDDAT',
                      'TYD:EINDTYD',
                      'STA:BEGINDAT',
                      'STA:BEGINTYD',
                      'STA:EINDDAT',
                      'STA:EINDTYD',
                      'STA:RKSSTATUS',
                      'EXTCODE',
                      'BRON',
                      'ORGOMS',
                      'IVSOMS',
                      'is_PAK', 'ID'], inplace=True)
    return df
```

In [ ]:

```
def filter_messy_observations(df):
    '''Filter out rows with bad observations, and rewrite tsome of the colu
mns'''
    df = df[df['KWC'].isin([0,6])]
    df.loc[:,'BTX'] = df.loc[:,'BTXOMS']
    df.loc[:,'GBD'] = df.loc[:,'GBDOMS']
    df.loc[:,'OGI'] = df.loc[:,'OGIOMS']
    df.loc[:,'ANI'] = df.loc[:,'ANIOMS']
    df.loc[:,'BHI'] = df.loc[:,'BHIOMS']
    df.loc[:,'BMI'] = df.loc[:,'BMIOMS']
    df.loc[:,'VAT'] = df.loc[:,'VATOMS']
    return df
```

In [ ]:

```
def clean_my_df(df):
    """Combination of several steps to clean the DF"""
    proper_datetime()
    df = filter_messy_observations(df)
    df= drop_bad_columns(df)
    df = fix_order(df)
    df.drop(columns=df.columns[-40:], inplace=True)
    return df
```

In [ ]:

```
#Perform all necessary cleaning steps.
df = clean_my_df(df)
```

In [ ]:

```
#All done - write to file
df.to_csv("../data/data_clean.csv")
```