University of Amsterdam

Anton Pannekoek instituut

Basic Linux and Coding for AA (BLAC)
Exercise 3 (week 2)

Author:
Timo Halbesma, 6126561

Supervisor:
Dr. T. Coenen

Listing 1: TLRH's solution for the BLAC homework 3 (week 2)

```
# knmi-1-6126561.py <-- Assignment 2

# Python script for Basic Linux and Coding for AA homework 3 (week 2).
# Usage: python knmi-1-6126561.py
# TLR Halbesma, 6126561, september 9, 2014. Version 1.0; implemented

# NB All functions in this program require the entire dataset as input.
# This behavior could be altered such that main() subsets the dataset and feeds
# it to the functions. I might change this later on for aesthetic reasons.

# Not yet required. However, dr. Coenen did mention we will plot this data.
# import math
# import matplotlib.pyplot as plt

INPUTFILE = './KNMI_20000101.txt'

def read_data(datasetKNMI): # <-- Assignment 5
    """
    Function to read KNMI dataset obtained from
    http://www.knmi.nl/climatology/daily_data/selection.cgi

    datasetKNMI : list containing the entire dataset including header

    returns a list containing a list of all datapoints per station per date.
    """

    lines = []

    # Assignment 3
    # Skip first 85 lines because that is the header. Very ugly solution :-(
    # NB this breaks down if the header size changed. Be cautious!
    for i in range(85,500): # Header ends after line 85 && Only use first 500.
            myLine = datasetKNMI[i].strip().split(',') # strip to remove '\n'
            cleanLine = []
            for entry in myLine:
                # entry.strip() removes the whitespace around the datapoint.
                # entry.strip() returns False if len(x.strip()) == 0 (missing..)
                if entry.strip():
                    cleanLine.append(entry.strip())
                else:
                    # Assignment 4. Use None for missing data entries.
                    cleanLine.append(None)
            lines.append(cleanLine)

    return lines

def read_StationID(datasetKNMI): # <-- Assignment 8
    """
    Function to read header from KNMI dataset, in particular the station info.

    datasetKNMI : list containing the entire dataset including header.

    returns a list containing one list for each station.
    """

    allStations = datasetKNMI[3:41]
    allStationsCleaned = list()

    for station in allStations[1:]: # First line contains column info, remove.
        # Remove leading '#', split and unpack first four columns.
        stationID, lon, lat, alt = \
                station.replace(':','').strip('#').split()[:4]
        # The name may contain spaces. Take sublist until last element.
        name = ' '.join(station.strip('#').split()[4:])

        # Create list containing one list for each station.
        # That list contains for each station 5 entries
        # Explicit typecasts to sensible datatypes. Trivial datatype choices.
        # StationID is a natural number, thus, an integer.
        # longitude, latitude and altitude are rational numbers, thus, floats.
        # The name consists of multiple characters, thus, is saved to string.
        allStationsCleaned.append([int(stationID), float(lon), \
                float(lat), float(alt), str(name)])

    return allStationsCleaned

def read_ColumnDescription(datasetKNMI): # <-- Assignment 9
    """
    Function to read header from KNMI dataset, in particular column descriptions
    datasetKNMI : list containing the entire dataset including header.

    returns a dictionary mapping the column name to its description
    NB dictionaries may be printed in random order.
    """

    columnDescription = datasetKNMI[42:82]
    columnDescriptionCleaned = dict()

    for entry in columnDescription:
        print entry
        abbreviation = ''.join(entry.strip('#').split('=')[:1]).strip()
        description = ' '.join(entry.strip('#').split('=')[1:])
        columnDescriptionCleaned[abbreviation] = description

    return columnDescriptionCleaned

def read_ColumnHeader(datasetKNMI): # <-- Assignment 10
    """
    Function to read header from KNMI dataset, in particular column header.

    datasetKNMI : list containing the entire dataset including header.

    returns list of column names.
    """

    columnHeader = datasetKNMI[83:84]

    columnHeadersCleaned = ''.join(columnHeader).strip('#').strip().split(',')

    return columnHeadersCleaned

def main():
    # Assignment 1
    f = open(INPUTFILE, 'r')
    datasetKNMI = f.readlines()
    f.close()

    # Assignment 12
    print read_data(datasetKNMI)[0], '\n\n'
    print read_StationID(datasetKNMI), '\n\n'
    print read_ColumnDescription(datasetKNMI), '\n\n'
    print read_ColumnHeader(datasetKNMI)
    # NB there is one entry more in the list returned by read_ColumnHeader
    # STN is in the line with column headers but it has no description.

# This codeblock is executed from CLI, but not upon import.
if __name__ == '__main__': # <-- Assignment 6; was already in my file though.
    main()
```