

Supplementary Table S1. Example in Python retrieving daily data, exporting results to a text file

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
# enable debugging
import cgitb
cgitb.enable()
import csv
import StringIO
import urllib2
import urllib
import time
# Make this piece of text working as a program
"""
Open notepad or a similar simple text editor
Copy-paste this boxed text to the opened document
Save it as a file named myfilename.py
If you have not installed python, download it and install from www.python.org
After installation, browse your myfilename.py and double click it (or run it from the command line)
It should open a "black screen" for less than 10 seconds and then it closes.
A file should be created called ClipickExportedData.csv in the same directory
play around with the section "CONTROL VARIABLES OF THIS ALGORITHM" to retrieve different date ranges (expect more or
less than ten seconds as more daily data will increase the retrieval time).
"""
# Needed daily DATA:
# Column 0: Year
# Column 1: Month
# Column 2: Day
# Column 3: Average temperature (degrees C)
# Column 3: Radiation (MJ / sq meter)
# Column 3: Precipitation (mm)

#CONTROL VARIABLES OF THIS ALGORITHM:
"""
you can explore other options here. you can try to create a loop if you need several locations or different scenarios
"""
#
Longitude = -8.335343513
Latitude = 39.2816836
StartYear = 2000 # Min=1951, Max=2100
StartMonth = 1
StartDay = 1
EndYear = 2003
EndMonth = 12
EndDay = 31
IPCCAssessmentReport = 4 # either 4 or 5
Dataset = 'METO-HC_HadRM3Q0_A1B_HadCM3Q0_DM_25km'
# if IPCCAssessmentReport =4 use METO-HC_HadRM3Q0_A1B_HadCM3Q0_DM_25km.
# If IPCCAssessmentReport =5 use either knmihistorical, knmievaluation, knmircp45, knmircp85
# If Dataset= 'knmihistorical', 1951 < StartYear < 2005, 1951 < EndYear < 2005
# If Dataset= 'knmievaluation', 1981 < StartYear < 2010, 1981 < EndYear < 2010
# If Dataset= 'knmircp45' or 'knmircp85', 2006 < StartYear < 2100, 2006 < EndYear < 2100

#the file to export the output:
outFileName = 'ClipickExportedData.csv' # tip you can build the name of the file to be according to the dates extracted
outFileHandle = open(outFileName, 'w')

start_time = time.time() # this is facultative, just to calculate timing of retrieval

# Build the HTTP REQUEST
pars = {}
pars['lat'] = Latitude
pars['lon'] = Longitude
pars['fmt'] = 'csv' # either csv, htmltable
pars['tspan'] = 'd' # d=daily; m=monthly
pars['sd'] = StartDay #'01'
pars['sm'] = StartMonth #'01'
pars['sy'] = StartYear
pars['ed'] = EndDay
pars['em'] = EndMonth
pars['ey'] = EndYear
pars['dts'] = Dataset # Beware of dates for extraction
pars['ar'] = IPCCAssessmentReport # either 4 or 5
pars['mod'] = "hisafe" # either yieldsafe or hisafe
url = 'http://www.isa.ulisboa.pt/proj/clipick/climaterequest_fast.php'
```

Supplementary table to the Resource Communication CliPick – Climate change web picker. A tool bridging daily climate needs in process based modelling in forestry and agriculture, by Joao HN Palma. Forest Systems Vol. 26 No. 1, April 2017 (<http://dx.doi.org/10.5424/fs/2017261-10251>)

```

url_pars      = urllib.urlencode(pars)
full_url      = url + '?' + url_pars
print "Request made to " + full_url
response      = urllib2.urlopen(full_url)
the_page      = response.read()

f             = StringIO.StringIO(the_page)
reader = csv.reader(f, delimiter=',')

# CEATE AN ARRAY FROM THE REQUESTED CSV OUTPUT
result=[]
for row in reader:
    result.append(row)

# WRITE IT DOWN IN THE OUTPUT FILE
"""
the daily output comes as
yieldsafe : Day, Month, Year, tas, rss, pr
hisafe    : Day, Month, Year, tasmax,tasmin,hursmax,hursmin,rss,pr,wss
in AR5 datasets, there are no min and max relative humidity (at the time of this development).
therefore, at the moment, hisafe format for AR5 is as follows:
hisafe : Day, Month, Year, tasmax,tasmin,hurs,evspsbl,rlds,pr,sfcWind

Currently Valid variables are (nomenclature as ENSEMBLES, and CORDEX project):
"pr"       : Precipitation
"tas"      : Average Temperature
"tasmin"   : Minimum Temperature
"tasmax"   : Maximum temperature
"rss"      : Radiation (ensembles)
"rlds"     : Radiation (cordex)
"evspsbl"  : Evaporation
"hurs"     : Relative Humidity
"hursmax": Maximum Relative humidity
"hursmin"  : Minimum Relative humidity
"wss"      : Wind Speed (ensembles)
"sfcWind"  : Wind Speed (cordex)
"""

# WRITE THE RESULTS IN THE FILE AND CLOSE IT
print "Output is being written in " + outFileName
for i in result:
    outFileHandle.write(",".join(i) + "\n")
outFileHandle.flush()
outFileHandle.close()

#Facultative...
end_time = time.time()
print "Processed in " + str(round(end_time - start_time,0)) + " seconds"
print "Output stored in " + outFileName
print "done"

```