Link to this document: https://tinyurl.com/seaicepractical

Sea Ice Thickness from Altimetry - Practical Exercise for the ESA Cryosphere Training Course 2018

The purpose of this practical is to get familiar with altimeter based sea ice thickness product (Climate Research Data Package or CRDP) from the ESA Climate Change Initiative project published in May 2018. The tools used are Python (version 2) language and its interactive shell (ipython). The important packages used are netCDF4, NumPy and Basemap.

This practical is best carried out in pairs. So team up with another student for the work. As this practical lies heavily on usage of Python, might be a good idea to team up with someone who knows the syntax, or at least with someone who has the data uncompressed on his/her VM. There are 6 tasks. Perfect performance is to complete all six (sixth being a bonus task for the really quick teams), but completing the three first will already be a good performance.

For all of the functions, google (or another search engine of your choice) is your friend! If you get a cryptic error message, try googling. The teachers, of course, will also help you out.

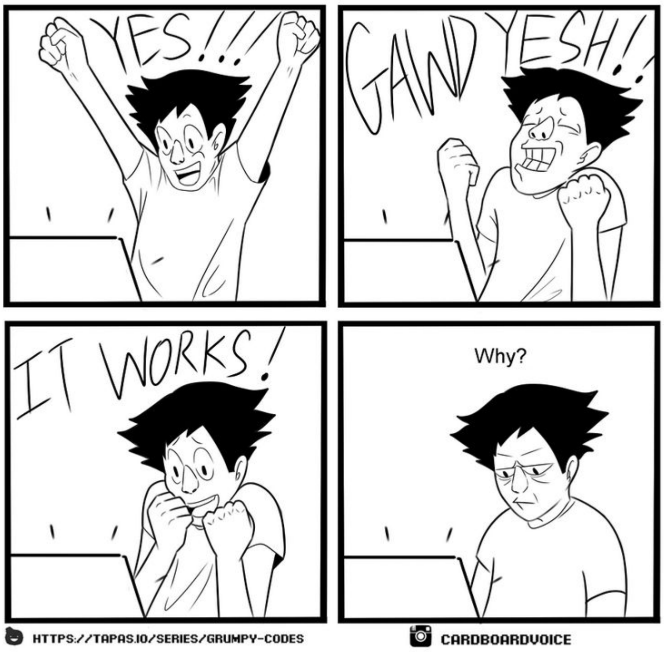
Most of the tasks are about plotting data. Make sure to save the plots and answers to the questions in this sheet to a document. You can either use for example Libreoffice writer on the VM or google docs for your document.

Note for Matlab users: If you are experienced in Matlab you may find the Matlab/Python cheatsheet at https://docs.scipy.org/doc/numpy/user/numpy-for-matlab-users.html useful.

COMPETITION - if you want to take part in the competition, send the best looking map you produce during this practical by email to: eero.rinne@fmi.fi - the winner will be announced at the pizza dinner later today!

Have fun,

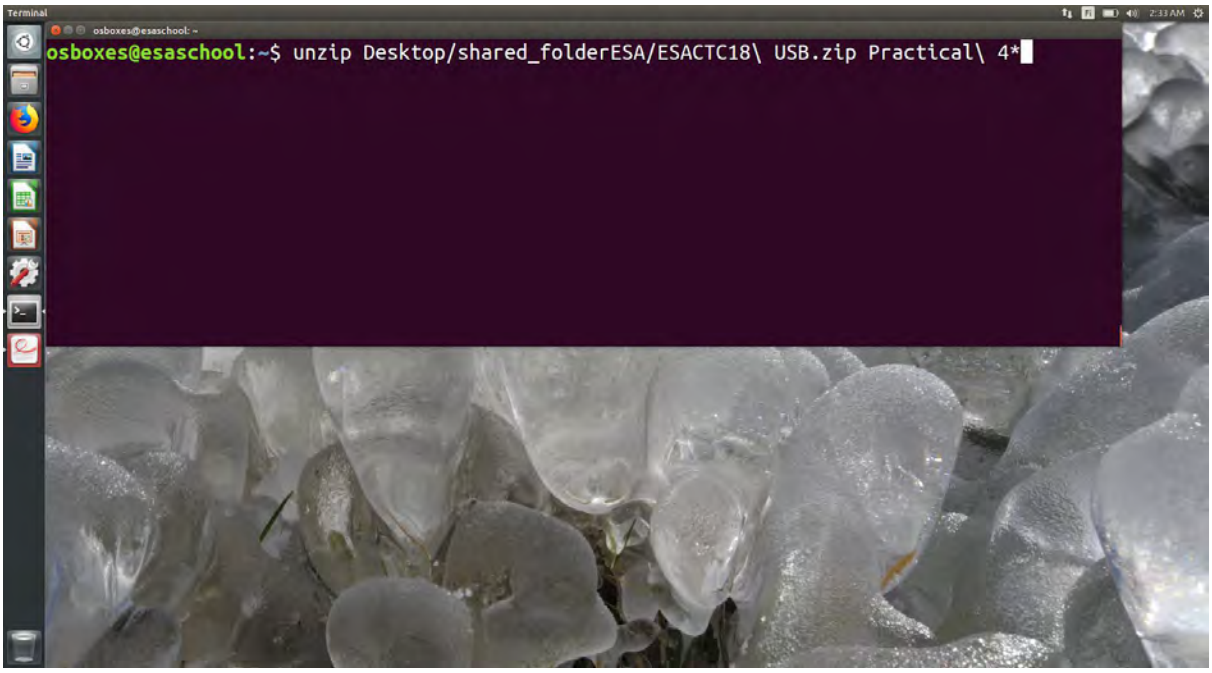
"IT WORKS. WHY?"



HTTPS://TAPAS.IO/SERIES/GRUMPY-CODES CARDBOARDVOICE

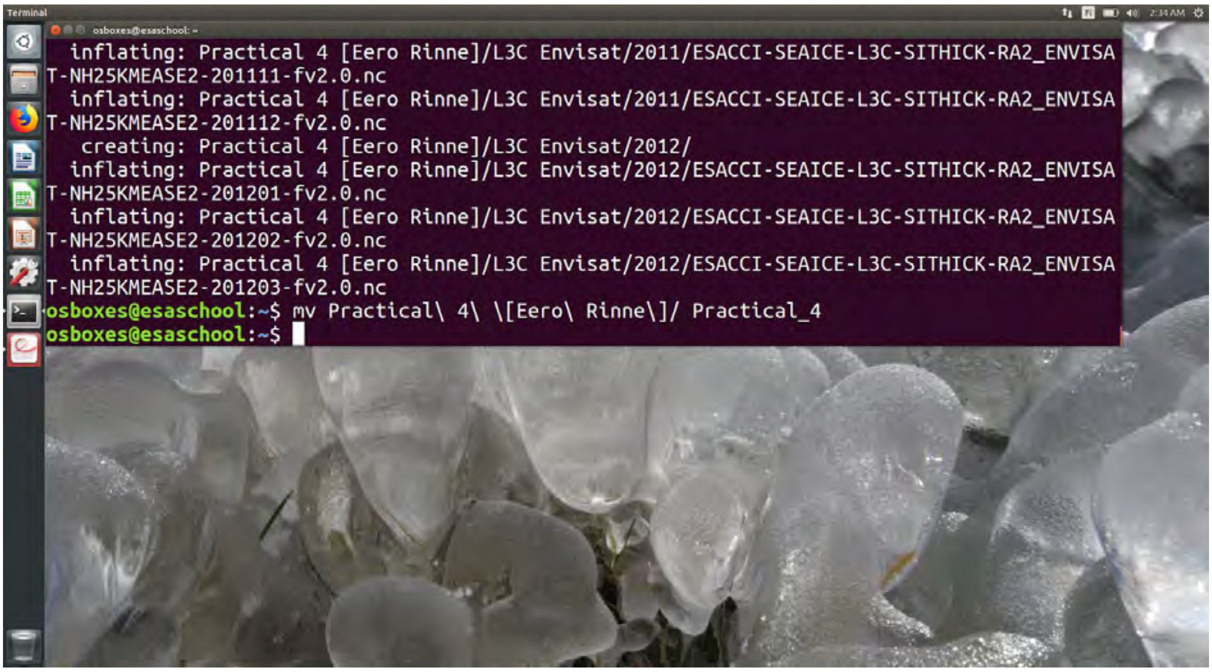
Getting the data from the USB to the VM:

You most likely have already managed to get the .zip archive to the shared folder (if you haven't, keep on trying!) If you've extracted the data, good, but if you haven't, do it (first open a terminal window by pressing ctrl-alt-t):



```
osboxes@esaschool:~$ unzip Desktop/shared_folderESA/ESACTC18\ USB.zip Practical\ 4*
```

Remember to rename the directory. Square brackets in file names are **not** a good idea in general, and they are pure poison to python:



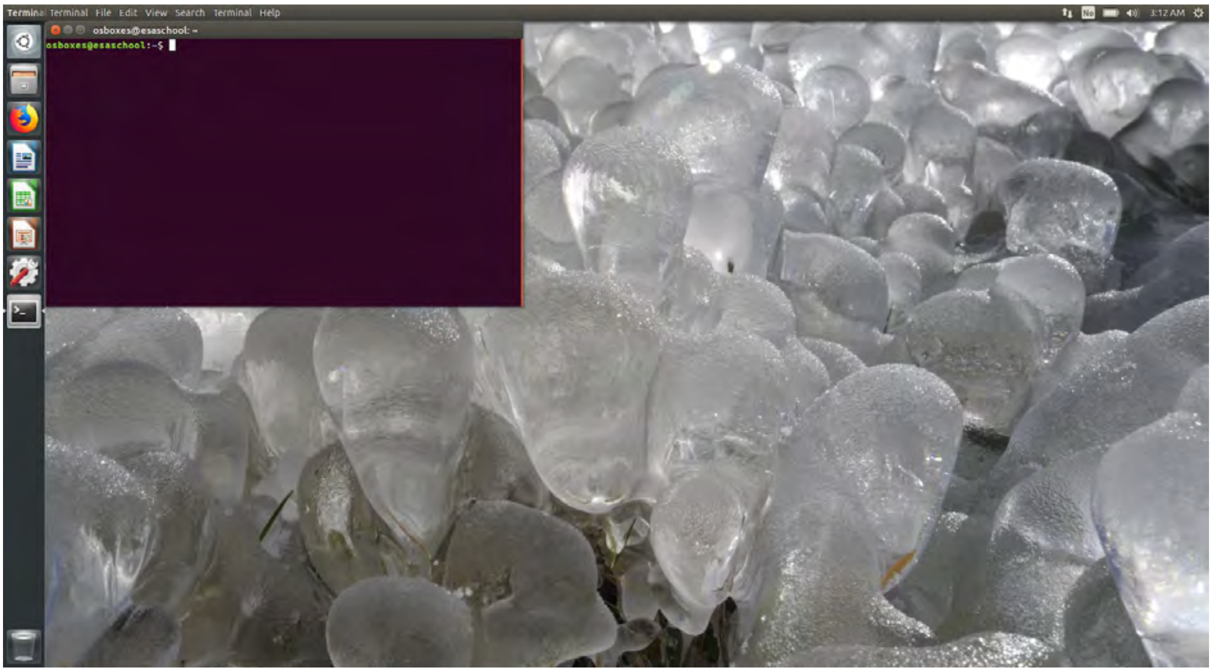
```
  inflating: Practical 4 [Eero Rinne]/L3C Envisat/2011/ESACCI-SEAICE-L3C-SITHICK-RA2_ENVISA
T-NH25KMEASE2-201111-fv2.0.nc
  inflating: Practical 4 [Eero Rinne]/L3C Envisat/2011/ESACCI-SEAICE-L3C-SITHICK-RA2_ENVISA
T-NH25KMEASE2-201112-fv2.0.nc
   creating: Practical 4 [Eero Rinne]/L3C Envisat/2012/
  inflating: Practical 4 [Eero Rinne]/L3C Envisat/2012/ESACCI-SEAICE-L3C-SITHICK-RA2_ENVISA
T-NH25KMEASE2-201201-fv2.0.nc
  inflating: Practical 4 [Eero Rinne]/L3C Envisat/2012/ESACCI-SEAICE-L3C-SITHICK-RA2_ENVISA
T-NH25KMEASE2-201202-fv2.0.nc
  inflating: Practical 4 [Eero Rinne]/L3C Envisat/2012/ESACCI-SEAICE-L3C-SITHICK-RA2_ENVISA
T-NH25KMEASE2-201203-fv2.0.nc
osboxes@esaschool:~$ mv Practical\ 4\ \[Eero\ Rinne\]/ Practical_4
osboxes@esaschool:~$ 
```

0. Introduction

In this practical, you shall be using iPython - an interactive shell of the Python programming language. This introduction task is meant to set you up so that you know which packages you need to use to plot maps, as well as get a first glimpse of the Python syntax. Remember that you will be using the map parameters you choose here for your maps later!

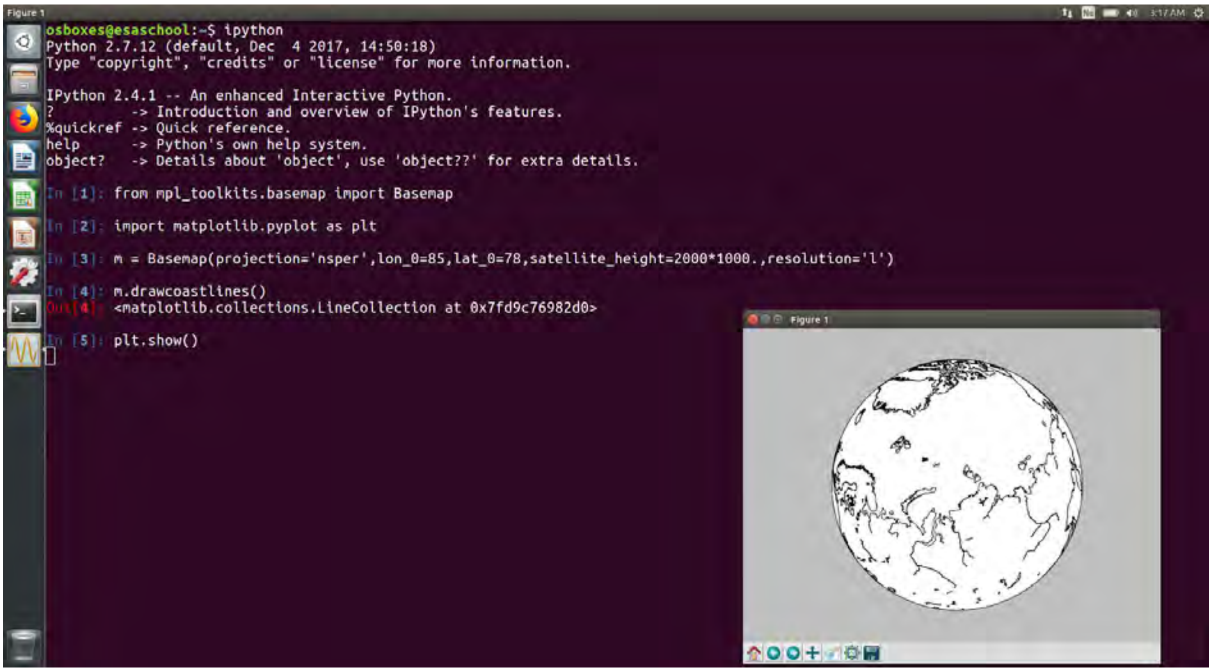
Launch your VM and open a terminal (quickest way to do this is to press ctrl-alt-t):

Launch ipython:



```
osboxes@esaschool:~$ 
```

Type in the commands as shown below:



```
osboxes@esaschool:~$ ipython
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
Type "copyright", "credits" or "license" for more information.

IPython 2.4.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: from mpl_toolkits.basemap import Basemap

In [2]: import matplotlib.pyplot as plt

In [3]: m = Basemap(projection='nsper',lon_0=85,lat_0=78,satellite_height=2000*1000.,resolution='l')

In [4]: m.drawcoastlines()
Out[4]: <matplotlib.collections.LineCollection at 0x7fd9c76982d0>

In [5]: plt.show()
```

Explain what each of the commands do. Try to change the map projection by using some of the examples from:

https://matplotlib.org/basemap/users/mapsetup.html

And try out some of the background functions from:

https://matplotlib.org/basemap/users/geography.html

for example m.fillcontinents() and m.bluemarble() will help you make the map look good.

Choose the map projection and background that you want to use in this practical. Give a thought on what makes a projection good or bad for Arctic wide sea ice maps. ***There will be a competition on who makes the best looking map of sea ice thickness over the Arctic so pay attention here!***

Finally, save one of your plots by using the “save figure” icon below your map and add the image to your practical document.

1. First map

Story:

Someone from the communications department of your University gives you a call. He wants to have a map of sea ice thickness in the Arctic. Fortunately, you happen to have the recently published ESA Sea Ice CCI thickness product on your hard drive, so you tell him it will just be a moment and you'll plot one.

Steps:

Open the right NetCDF file using netCDF4.Dataset

D = netCDF4.Dataset(filename)

Read latitude, longitude and thickness matrixes into variables

variable = D.variables['variable_name'][:]

Plot the data using basemap function m.contourf()

Hints:

- ***Remember to import necessary packages***
- Have a look at examples in https://matplotlib.org/basemap/users/examples.html
- For syntax, you can always have a look at the manual by help(function)
- Note that latitude and longitude variables in the NetCDF have slightly different geometry than thickness, freeboard etc.
- You can list the variable names of a netCDF4 file with *D.variables.keys()* *(*where D is a netCDF4.Dataset object*)*

2 The average story

Story:

Your professor got a hint that you just made a sea ice thickness map for outreach. Being a professor, she immediately got worried that you may have used an outlier year and tells you to make an average map over _all of the_ months March during Cryosat period instead. Furthermore, make an anomaly map showing the difference of average sea ice thickness and April 2017 thickness. Do you see any anomalous areas in the map for April 2017?

Hints:

You need to open several files and read arrays into a data structure. In Python, list will be by far the most convenient way to do this. However, it may be a good idea to convert those lists into numpy.array later.

```
list_thickness = list()
```

Then for all of the data files:

```
D = Dataset(filename)
mtx_thickness = ...
list_thickness.append(mtx_thickness)
```

To average several arrays pixel-wise, numpy.nanmean() is the tool you are looking for.

```
array_thickness = numpy.array(list_thickness)
mtx_average_thickness = numpy.nanmean(something,something)
```

3 Level 2P vs. Level 3C

Whilst looking at the products on http://catalogue.ceda.ac.uk/?q=sea+ice+thickness you notice that in addition to the gridded products, there are level 2 (L2P) products "on the satellite swath" which looks interesting. You should find these on your USB as well. Explain the difference of L2 and L3 CCI products. What kind of users would prefer L2 over L3 and vice versa? Plot April 2017 L2 thicknesses (all 30 files) on a map. Zoom into the area close to Siberian coast and try to explain high thicknesses close to the coast.

Hints:

- Reading the L2 products is similar to reading L3 products. However, as now you have several files to read, you should build a for -loop to read and plot the data. First make the code that will read and plot the data. m.contourf will not work for L2 data, but you should use m.scatter() instead.

- There are several ways to build a list of filenames to loop over. One of them is glob.glob(): https://docs.python.org/2/library/glob.html

- For the syntax of for loops: http://www.pythonforbeginners.com/loops/for-while-and-nested-loops-in-python

4 Oh no, snow!

Your professor has been doing a bit of background research and asks you to give a thought on the snow. She has this feeling that the CCI snow on sea ice estimate may be wildly unrealistic and, as you know from previous lectures, sea ice thickness estimates rely heavily on snow estimate used! The professor wants to know how thick snow estimates have been used (that is, for you to plot a map of snow thickness).

Hint: You will find no ready snow thickness in the product. However, bear in mind that Freeboard = Radar freeboard + snow_thickness*0.22.

5 Climate is changing and we need to look at the change...

We've all heard the stories how the sea ice is melting. But is it? How much can we actually say about the overall volume based on satellite altimeter measurements.

Make a time series plot of overall ice volume in the Arctic (where Cryosat-2 data is available). Are the trends significant? What are the challenges of the Cryosat-2 time series? Can you estimate the uncertainty of total ice volume?

- Hints: Loop through all of the L3 files you have. For all files, read the variables you need and calculate ice volume from them. Store the time of the product file as well as the total volume (sum over the matrix) into lists.

- Everything you need to calculate volume of ice per grid cell is in the L3C file, including the grid size (which is also in the file name...) First calculate a volume matrix (like you did for snow depth in previous task). Then calculate the sum of ice volume and store that into a list.

- Parallel to the list of volume, make a list of timestamps. The datetime -package provides a perfect tool for this. You can make a datetime -object from from the timestamp in the CCI product (and add it to a list called *list_time*) by for example:

  list_time.append(datetime.datetime.fromtimestamp(D.variables['time'][0]))

- After you have the lists for volume and time, just use plt.plot to plot them.

6 Bonus - Helping out the ship insurance agent

By this time, you've gained a reputation of being rather savvy with thickness products. You meet an insurance agent at a sea ice conference, who asks if you could produce a maps of the Antarctic which will show the areas of very thick ice during winter - just in case a ship gets in trouble, it would be good to have the background information of heavy ice areas in the past. Can you? Would it make sense to use freeboard instead of thickness for the Antarctic?

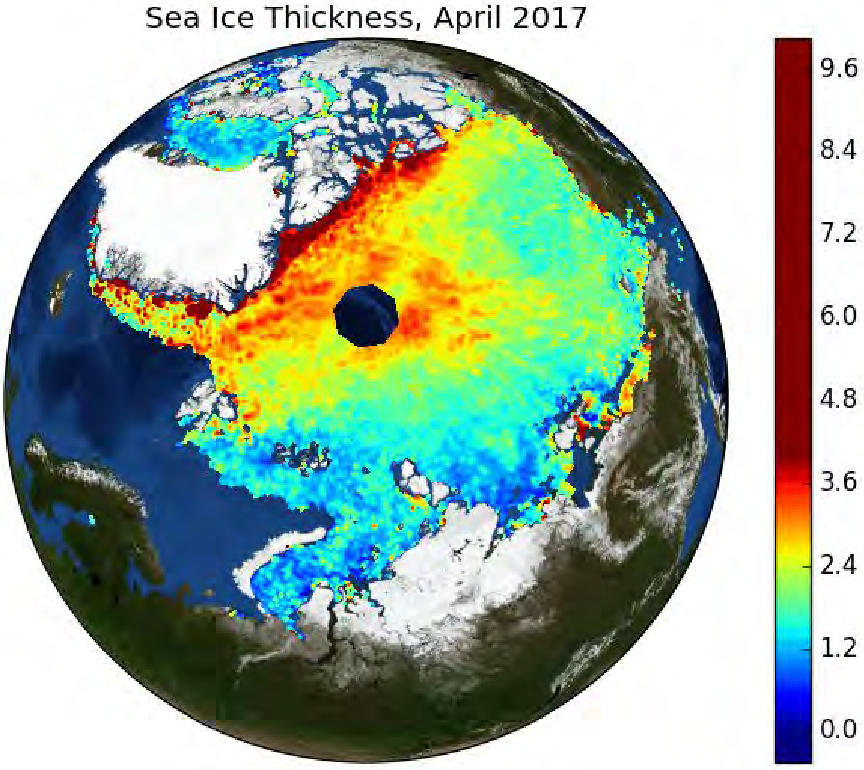
Hints: - None.

1. First map

Story:

Someone from the communications department of your University gives you a call. He wants to have a map of sea ice thickness in the Arctic. Fortunately, you happen to have the recently published ESA Sea Ice CCI thickness product on your hard drive, so you tell him it will just be a moment and you'll plot one.

```
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from netCDF4 import Dataset


D = Dataset('../CCICDR/L3C/cryosat2/v2.0/NH/2017/ESACCI-SEAICE-L3C-SITHICK-SIRAL_CRYOSAT2-NH25KMEASE2-201704-fv2.0.nc')

lat = D.variables['lat'][:]
lon = D.variables['lon'][:]
thi = D.variables['sea_ice_thickness']


m = Basemap(projection='nsper',lon_0=80,lat_0=85,satellite_height=2000*1000.,resolution='l')

x,y = m(lon, lat)

m.contourf(x[:],y[:],thi[0], 100, vmin=0, vmax=4, cmap='jet')
m.bluemarble()
cbar = plt.colorbar()
plt.title('Sea Ice Thickness, April 2017')

plt.show()
```


Sea Ice Thickness, April 2017
9.6
8.4
7.2
6.0
4.8
3.6
2.4
1.2
0.0

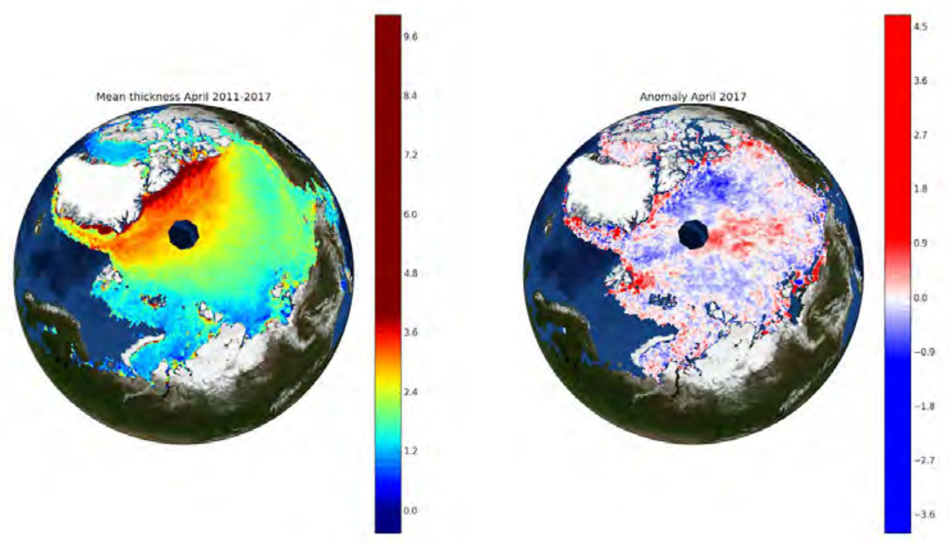
2 The average story

Story:

Your professor got a hint that you just made a sea ice thickness map for outreach. Being a professor, she immediately got worried that you may have used an outlier year and tells you to make an average map over _all of the_ months March during Cryosat period instead. Furthermore, make an anomaly map showing the difference of average sea ice thickness and April 2017 thickness. Are there anomalous areas?

```
from glob import glob
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from netCDF4 import Dataset
import numpy as np


filelist = glob('../CCICDR/L3C/cryosat2/v2.0/NH/20??/ESACCI-SEAICE-L3C-SITHICK-SIRAL_CRYOSAT2-NH25KMEASE2-20??04-fv2.0.nc')

filelist.sort()

list_thi = list()

for str_filename in filelist:

        print "Now processing: " + str_filename

        D = Dataset(str_filename)

        lat = D.variables['lat'][:]
        lon = D.variables['lon'][:]


        mtx_thi = D.variables['sea_ice_thickness'][0]
        list_thi.append(mtx_thi)



mtx_mean_thi = np.nanmean(np.array(list_thi), 0)


m = Basemap(projection='nsper',lon_0=80,lat_0=85,satellite_height=2000*1000.,resolution='l')

x,y = m(lon, lat)

plt.subplot(1,2,1)


m.contourf(x,y,mtx_mean_thi, 100, vmin=0, vmax=4, cmap='jet')
m.bluemarble()
cbar = plt.colorbar()

plt.title('Mean thickness April 2011-2017')


plt.subplot(1,2,2)
```

```
m.contourf(x,y,list_thi[-1]-mtx_mean_thi, 100, vmin=-1, vmax=1, cmap='bwr')
m.bluemarble()
cbar = plt.colorbar()
plt.title('Anomaly April 2017')

plt.show()
```

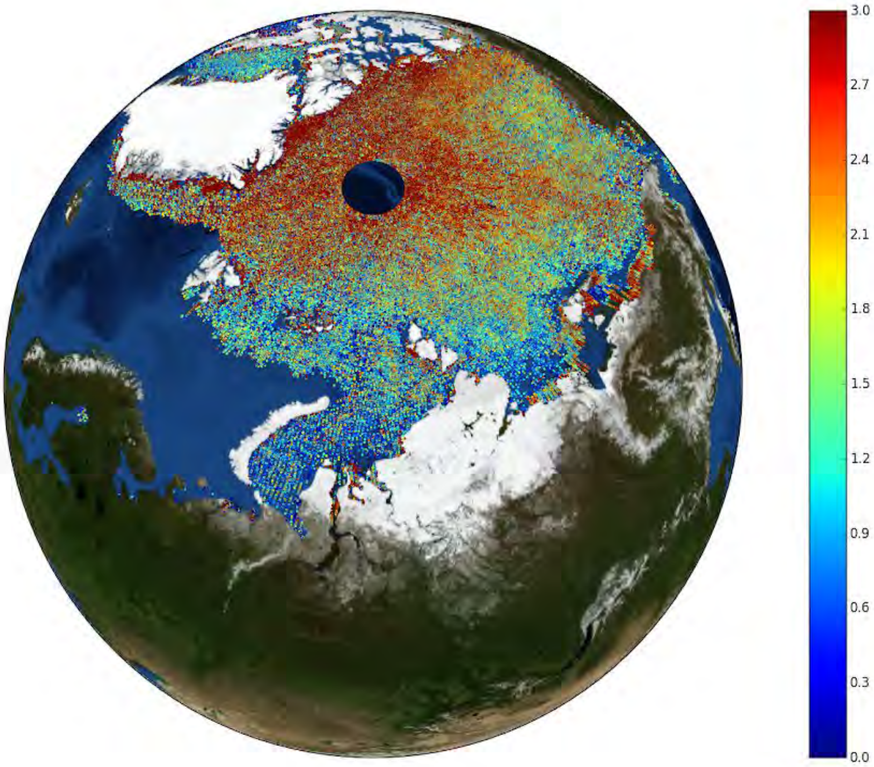


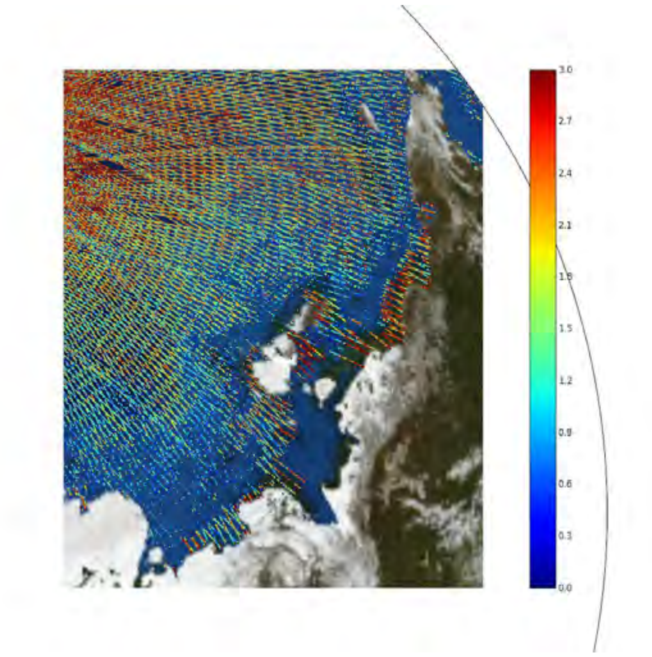
3 Level 2P vs. Level 3C

Whilst looking at the products on http://catalogue.ceda.ac.uk/?q=sea+ice+thickness you notice that in addition to the gridded products, there are level 2 (L2P) products "on the satellite swath" which looks interesting. Explain the difference of L2 and L3 CCI products. What kind of users would prefer L2 over L3 and vice versa? Plot April 2017 L2 thicknesses (all 30 files) on a map. Zoom into the area close to Siberian coast and try to explain high thicknesses close to the coast.

```
from glob import glob
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from netCDF4 import Dataset


filelist = glob('../CCICDR/L2P/cryosat2/v2.0/NH/2017/04/*201704*')

filelist.sort()

m =
Basemap(projection='nsper',lon_0=80,lat_0=78,satellite_height=2000*1000.,resolution='l')
```

```
for str_filename in filelist:

    print "Now processing: " + str_filename
    D = Dataset(str_filename)

    lat = D.variables['lat'][:]
    lon = D.variables['lon'][:]
    thi = D.variables['sea_ice_thickness']


    x,y = m(lon, lat)
    m.scatter(x[::1],y[::1],c=thi[::1],s=3,vmin=0,vmax=3,edgecolor='none',cmap='jet')



m.bluemarble()
cbar = plt.colorbar()
plt.show()
```



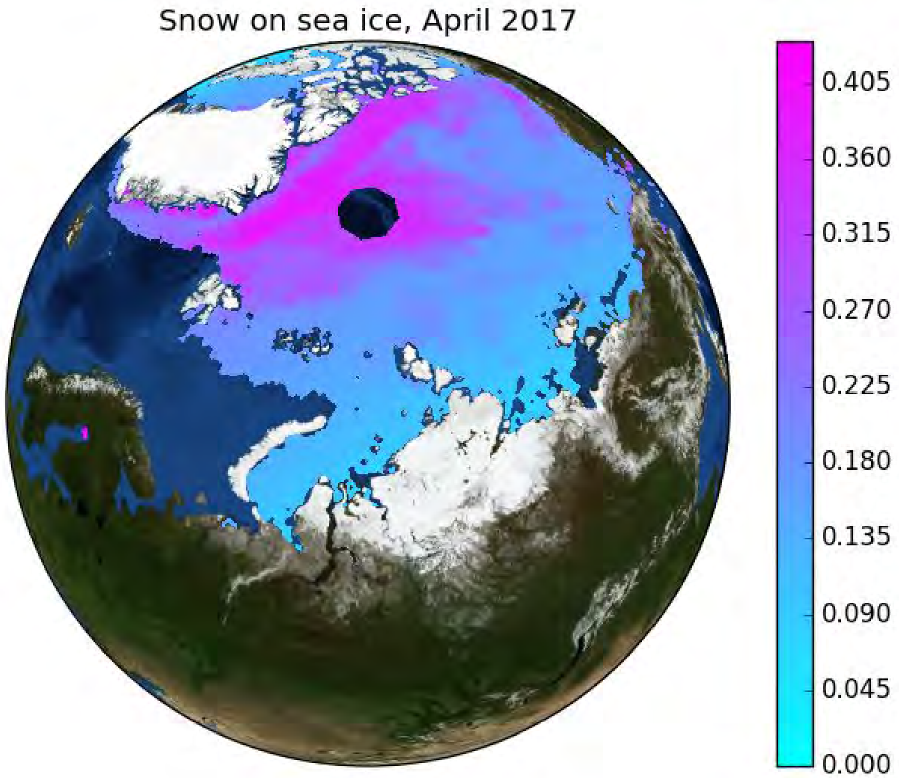


4 Oh no, snow!

Your professor has been doing a bit of background research and asks you to give a thought on the snow. As you know from previous lectures, sea ice thickness estimates rely heavily on snow estimate used! The professor wants to know how thick snow estimates have been used (that is, for you to plot a map of snow thickness).

```
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from netCDF4 import Dataset


D = Dataset('../CCICDR/L3C/cryosat2/v2.0/NH/2017/ESACCI-SEAICE-L3C-SITHICK-SIRAL_CRYOSAT2-NH25KMEASE2-201704-fv2.0.nc')


lat = D.variables['lat'][:]
lon = D.variables['lon'][:]
thi = D.variables['sea_ice_thickness'][0]
radar_fb = D.variables['radar_freeboard'][0]
fb = D.variables['freeboard'][0]

snow = (fb - radar_fb)/0.22

m = Basemap(projection='nsper',lon_0=80,lat_0=78,satellite_height=2000*1000.,resolution='l')

x,y = m(lon, lat)
```

```
m.contourf(x,y,snow,100,cmap='cool')
m.bluemarble()
cbar = plt.colorbar()
plt.title('Snow on sea ice, April 2017')
plt.show()
```



5 Climate is changing and we need to look at the change...

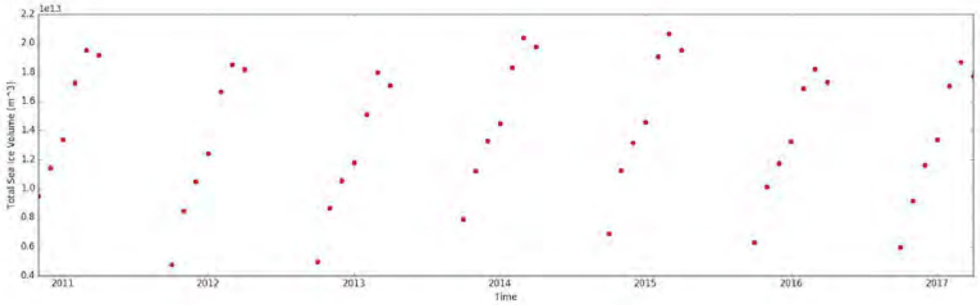
We’ve all heard the stories how the sea ice is melting. But is it? How much can we actually say about the overall volume based on satellite altimeter measurements.

Make a time series plot of overall ice volume in the Arctic (where Cryosat-2 data is available). Are the trends significant? What are the challenges of the Cryosat-2 time series? Can you estimate the uncertainty of total ice volume?

```
from glob import glob
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from netCDF4 import Dataset
import numpy as np
import datetime


filelist =
glob('../CCICDR/L3C/cryosat2/v2.0/NH/20??/ESACCI-SEAICE-L3C-SITHICK-SIRAL_CRYOSAT2-NH25KMEASE2-20????-fv2.0.nc')

filelist.sort()

list_thi = list()
list_conc = list()
list_volume = list()
list_time = list()

grid_cell_area = 25000*25000 #one grid cell area in m^2

for str_filename in filelist:

        print "Now processing: " + str_filename

        D = Dataset(str_filename)

        lat = D.variables['lat'][:]
        lon = D.variables['lon'][:]

        mtx_thi = D.variables['sea_ice_thickness'][0]
        mtx_conc = D.variables['sea_ice_concentration'][0]

        mtx_volume = mtx_thi * mtx_conc * grid_cell_area * 0.01


        list_thi.append(mtx_thi)
```

```
    list_conc.append(mtx_thi)
    list_volume.append(mtx_volume)
    list_time.append(datetime.datetime.fromtimestamp(D.variables['time'][0]))
```

```
list_total_volume = list()

for mtx_volume in list_volume:
    list_total_volume.append(np.nansum(mtx_volume))

plt.plot(list_time, list_total_volume, 'ro')
plt.xlabel('Time')
plt.ylabel('Total Sea Ice Volume [m^3]')
plt.show()
```

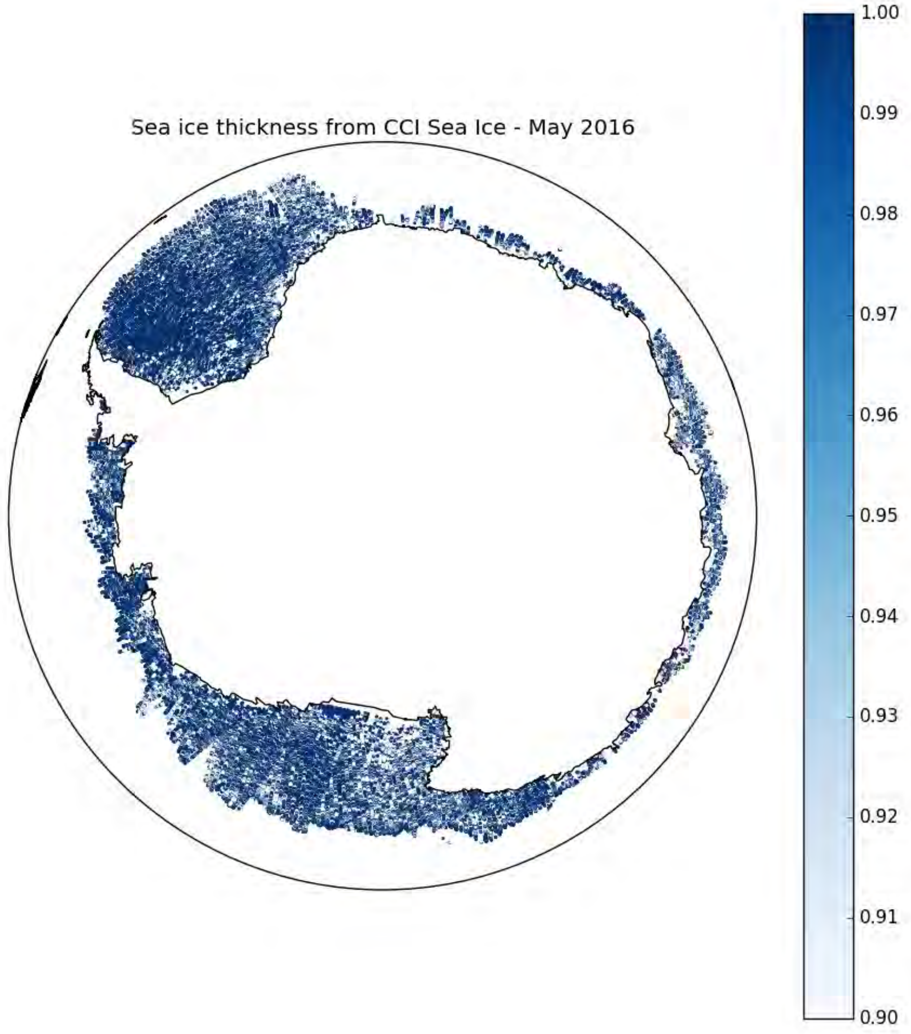


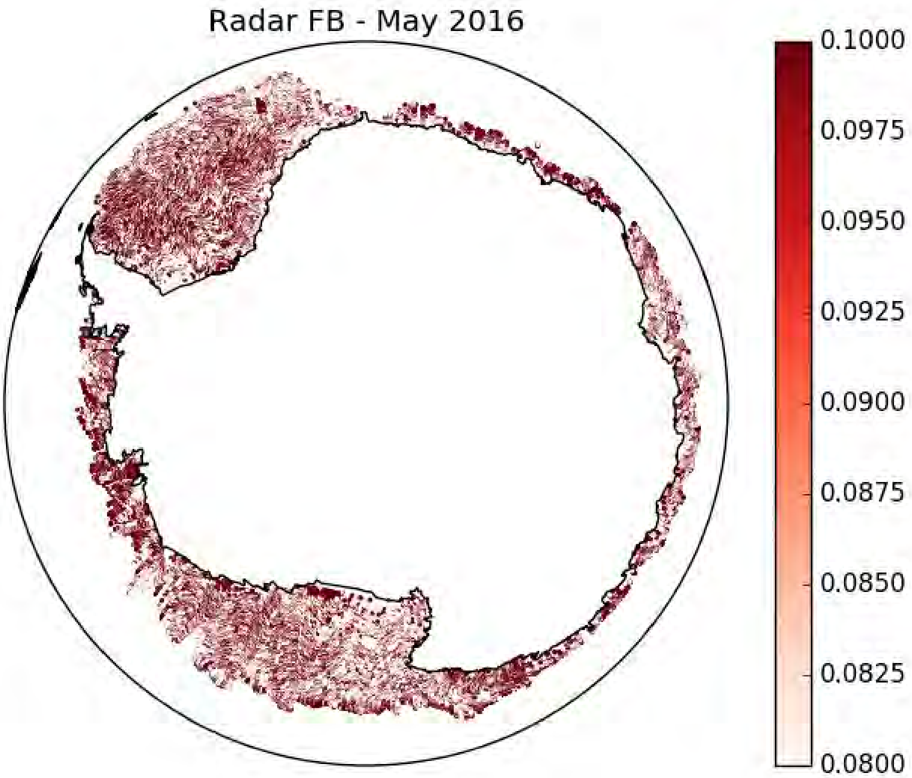
There is no significant trend. The challenge of CS2 time series is that it is really short, and the annual (and interannual variability) is large.

6 Bonus - Helping out the ship insurance agent

By this time, you've gained a reputation of being rather savvy with thickness products. You meet an insurance agent at a sea ice conference, who asks if you could produce a maps of the Antarctic which will show the areas of very thick ice - just in case a ship gets in trouble, it would be good to have the background information of heavy ice areas in the past. Can you? Would it make sense to use freeboard instead of thickness for the Antarctic?

This, actually is a pretty broad question and leaves a lot open for the student.





Example code:

```
from glob import glob
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from netCDF4 import Dataset


filelist = glob('../CCICDR/L2P/cryosat2/v2.0/SH/2016/05/*nc')

filelist.sort()

m =
Basemap(projection='nsper',lon_0=0,lat_0=-90,satellite_height=2000*1000.,resolution='l')

for str_filename in filelist:

    print "Now processing: " + str_filename
    D = Dataset(str_filename)

    lat = D.variables['lat'][:]
    lon = D.variables['lon'][:]
    thi = D.variables['sea_ice_thickness']
```

```
        x,y = m(lon, lat)
        m.scatter(x[::5],y[::5],c=thi[::5],s=5,vmin=0.9,vmax=1,edgecolor='none',cmap='Blues')



m.drawcoastlines()
cbar = plt.colorbar()

plt.title('Sea ice thickness from CCI Sea Ice - May 2016')



plt.figure()
for str_filename in filelist:

        print "Now processing: " + str_filename
        D = Dataset(str_filename)

        lat = D.variables['lat'][:]
        lon = D.variables['lon'][:]
        fb = D.variables['radar_freeboard']


        x,y = m(lon, lat)

m.scatter(x[::5],y[::5],c=fb[::5],s=5,vmin=0.08,vmax=0.10,edgecolor='none',cmap='Reds')



m.drawcoastlines()
cbar = plt.colorbar()
plt.title('Radar FB - May 2016')

plt.show()
```