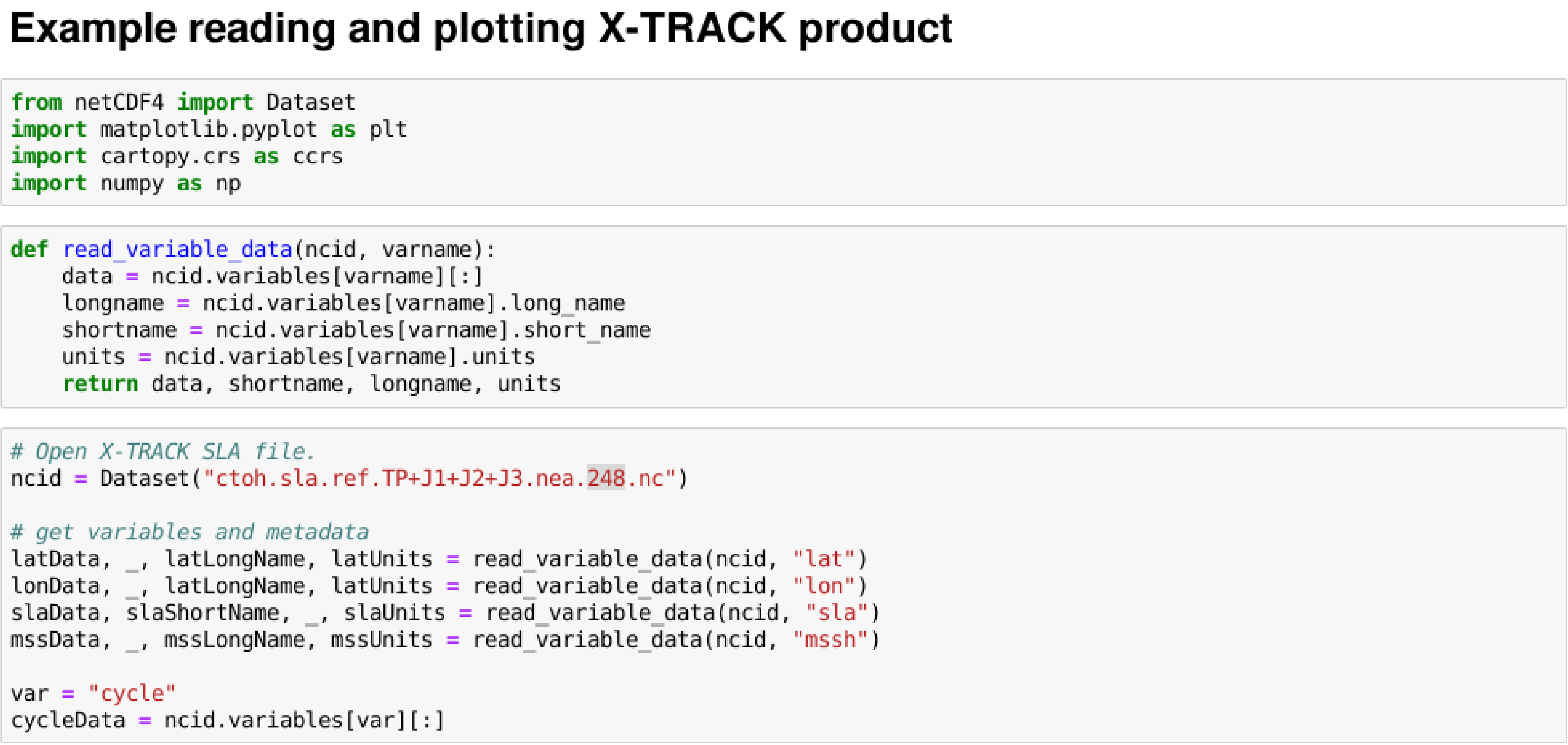
Python Example

Here you can find a basic example for reading and plotting the X-TRACK product using Python version 3.7. The output and code are shown below. Python is a free, general purpose programming language that is available on multiple operating systems including Linux, Windows and Mac OS.

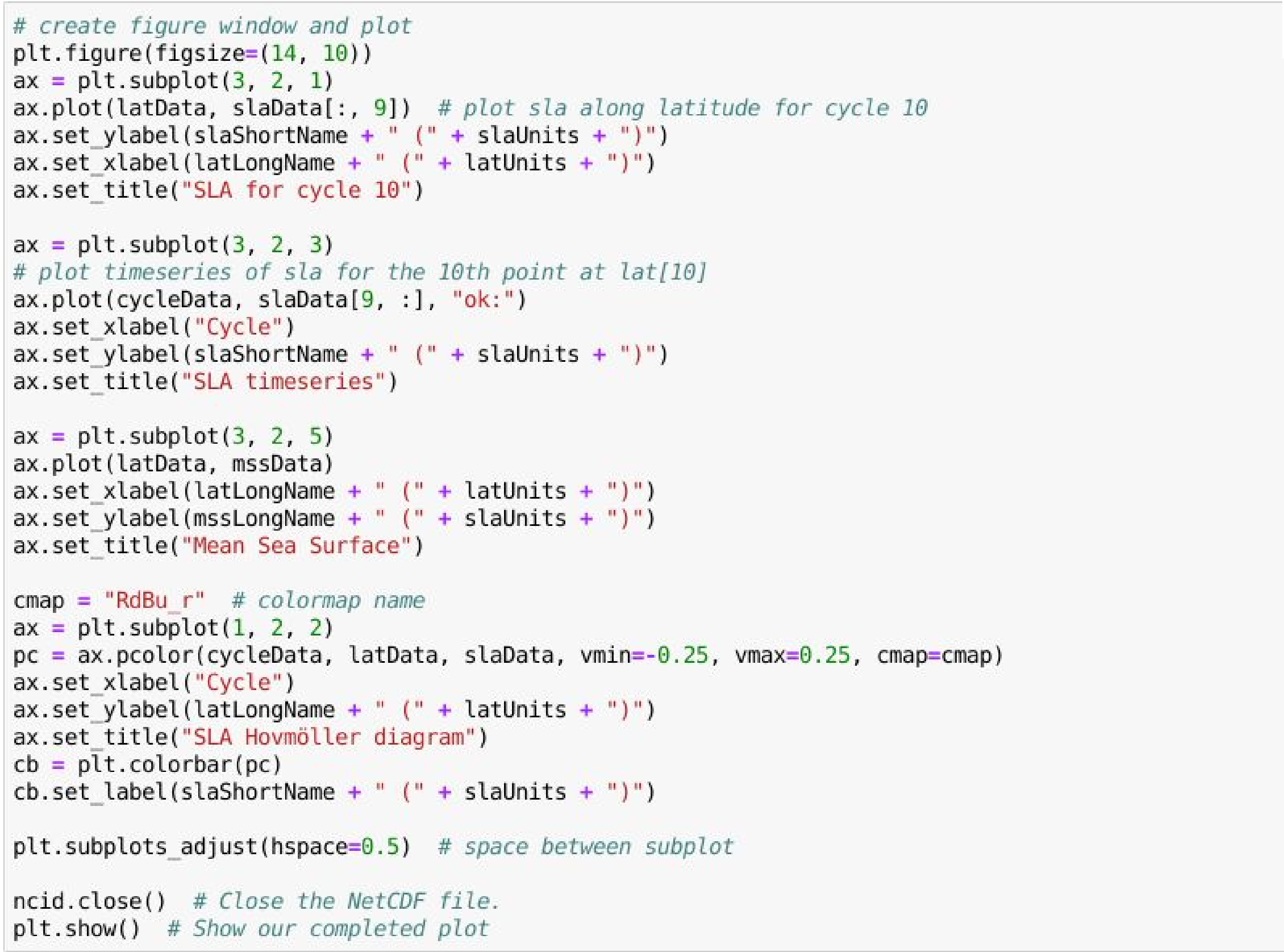


Example reading and plotting X-TRACK product

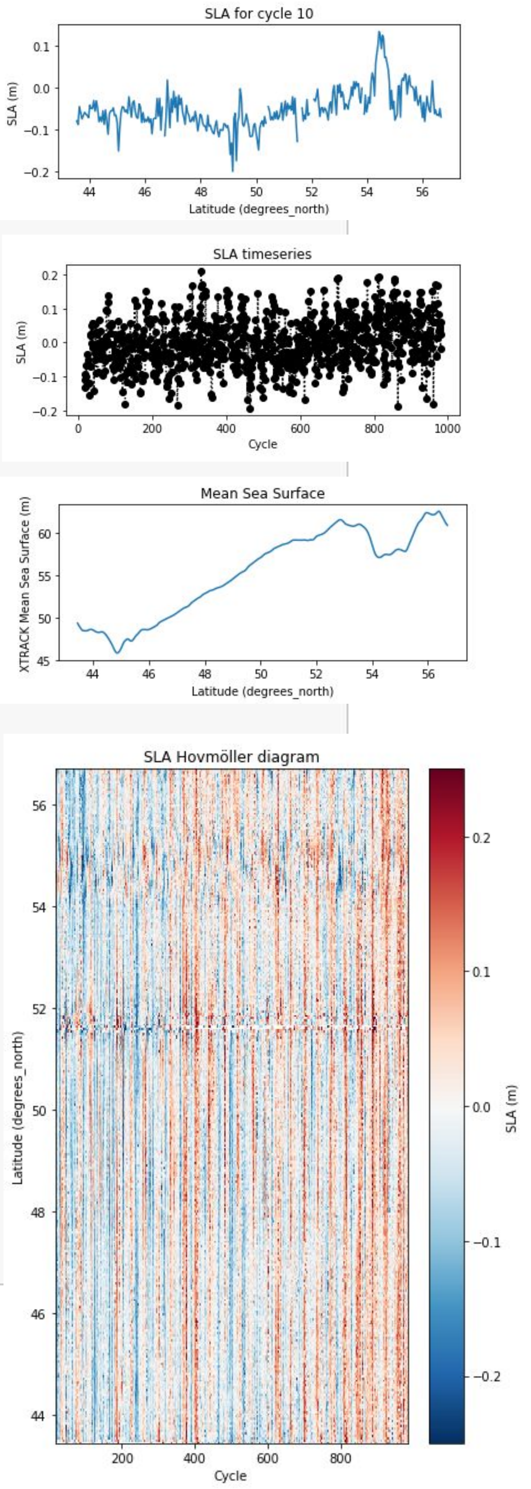
```python
from netCDF4 import Dataset
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import numpy as np
```

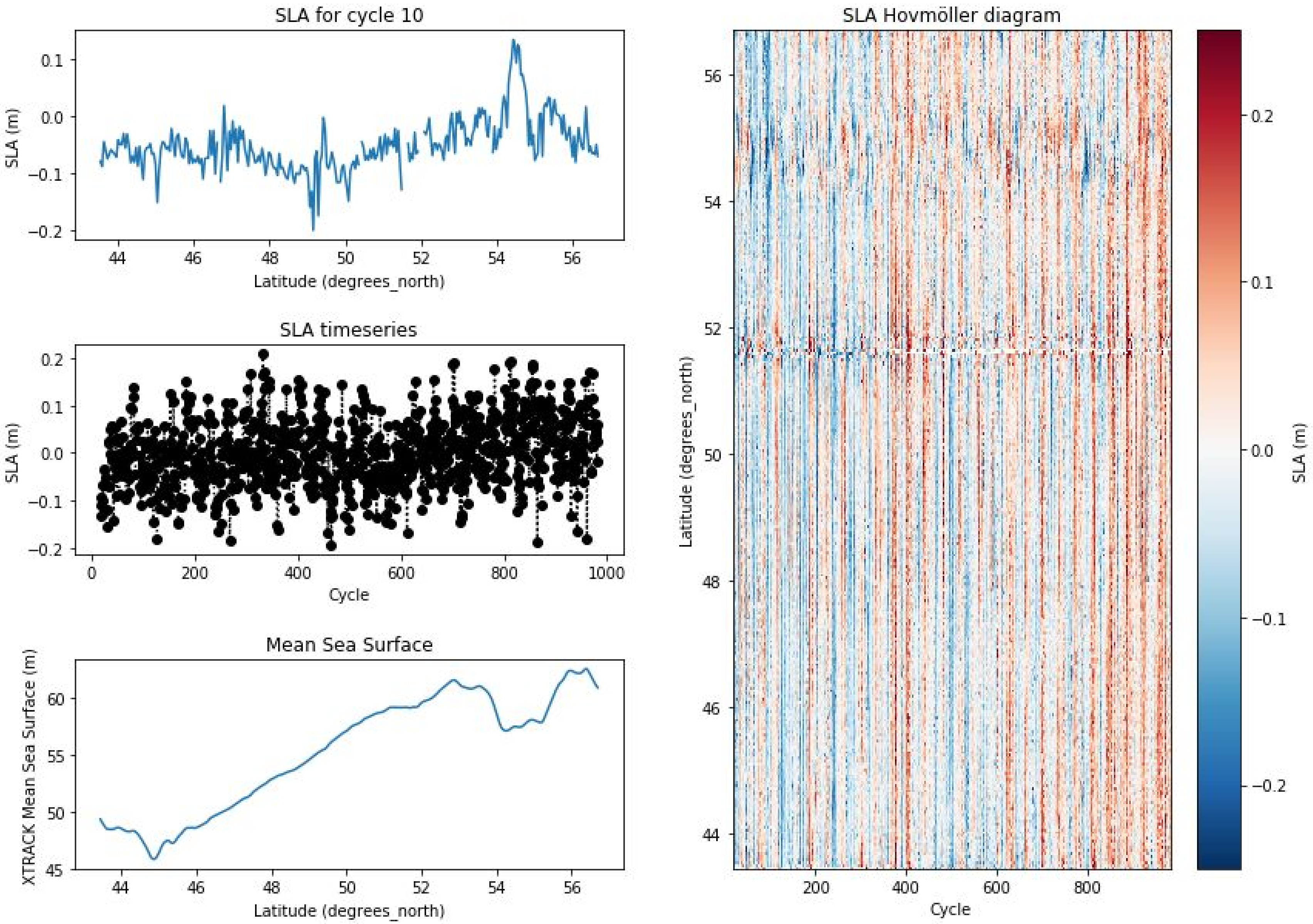
```python
def read_variable_data(ncid, varname):
    data = ncid.variables[varname][:]
    longname = ncid.variables[varname].long_name
    shortname = ncid.variables[varname].short_name
    units = ncid.variables[varname].units
    return data, shortname, longname, units
```

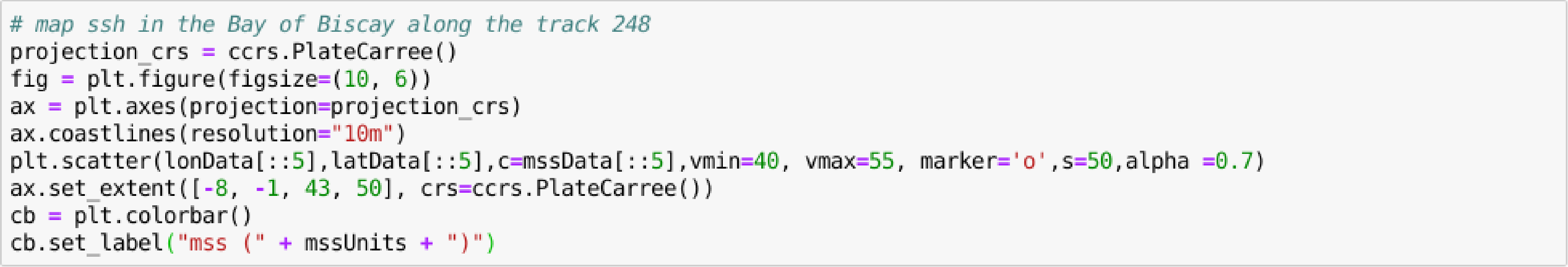
```python
# Open X-TRACK SLA file.
ncid = Dataset("ctoh.sla.ref.TP+J1+J2+J3.nea.248.nc")

# get variables and metadata
latData, _, latLongName, latUnits = read_variable_data(ncid, "lat")
lonData, _, latLongName, latUnits = read_variable_data(ncid, "lon")
slaData, slaShortName, _, slaUnits = read_variable_data(ncid, "sla")
mssData, _, mssLongName, mssUnits = read_variable_data(ncid, "mssh")

var = "cycle"
cycleData = ncid.variables[var][:]
```



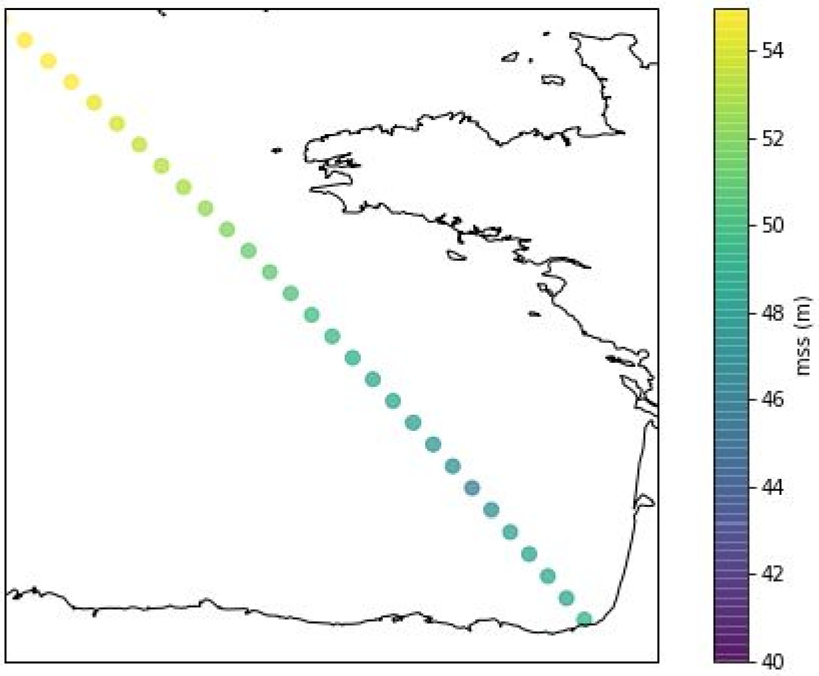
```
# create figure window and plot
plt.figure(figsize=(14, 10))
ax = plt.subplot(3, 2, 1)
ax.plot(latData, slaData[:, 9])  # plot sla along latitude for cycle 10
ax.set_ylabel(slaShortName + " (" + slaUnits + ")")
ax.set_xlabel(latLongName + " (" + latUnits + ")")
ax.set_title("SLA for cycle 10")

ax = plt.subplot(3, 2, 3)
# plot timeseries of sla for the 10th point at lat[10]
ax.plot(cycleData, slaData[9, :], "ok:")
ax.set_xlabel("Cycle")
ax.set_ylabel(slaShortName + " (" + slaUnits + ")")
ax.set_title("SLA timeseries")

ax = plt.subplot(3, 2, 5)
ax.plot(latData, mssData)
ax.set_xlabel(latLongName + " (" + latUnits + ")")
ax.set_ylabel(mssLongName + " (" + slaUnits + ")")
ax.set_title("Mean Sea Surface")

cmap = "RdBu_r"  # colormap name
ax = plt.subplot(1, 2, 2)
pc = ax.pcolor(cycleData, latData, slaData, vmin=-0.25, vmax=0.25, cmap=cmap)
ax.set_xlabel("Cycle")
ax.set_ylabel(latLongName + " (" + latUnits + ")")
ax.set_title("SLA Hovmöller diagram")
cb = plt.colorbar(pc)
cb.set_label(slaShortName + " (" + slaUnits + ")")

plt.subplots_adjust(hspace=0.5)  # space between subplot

ncid.close()  # Close the NetCDF file.
plt.show()  # Show our completed plot
```




SLA for cycle 10
SLA (m)
0.1
0.0
−0.1
−0.2
44
46
48
50
52
54
56
Latitude (degrees_north)
SLA timeseries
SLA (m)
0.2
0.1
0.0
−0.1
−0.2
0
200
400
600
800
1000
Cycle
Mean Sea Surface
XTRACK Mean Sea Surface (m)
60
55
50
45
44
46
48
50
52
54
56
Latitude (degrees_north)
SLA Hovmöller diagram
Latitude (degrees_north)
56
54
52
50
48
46
44
200
400
600
800
Cycle
0.2
0.1
0.0
−0.1
−0.2
SLA (m)

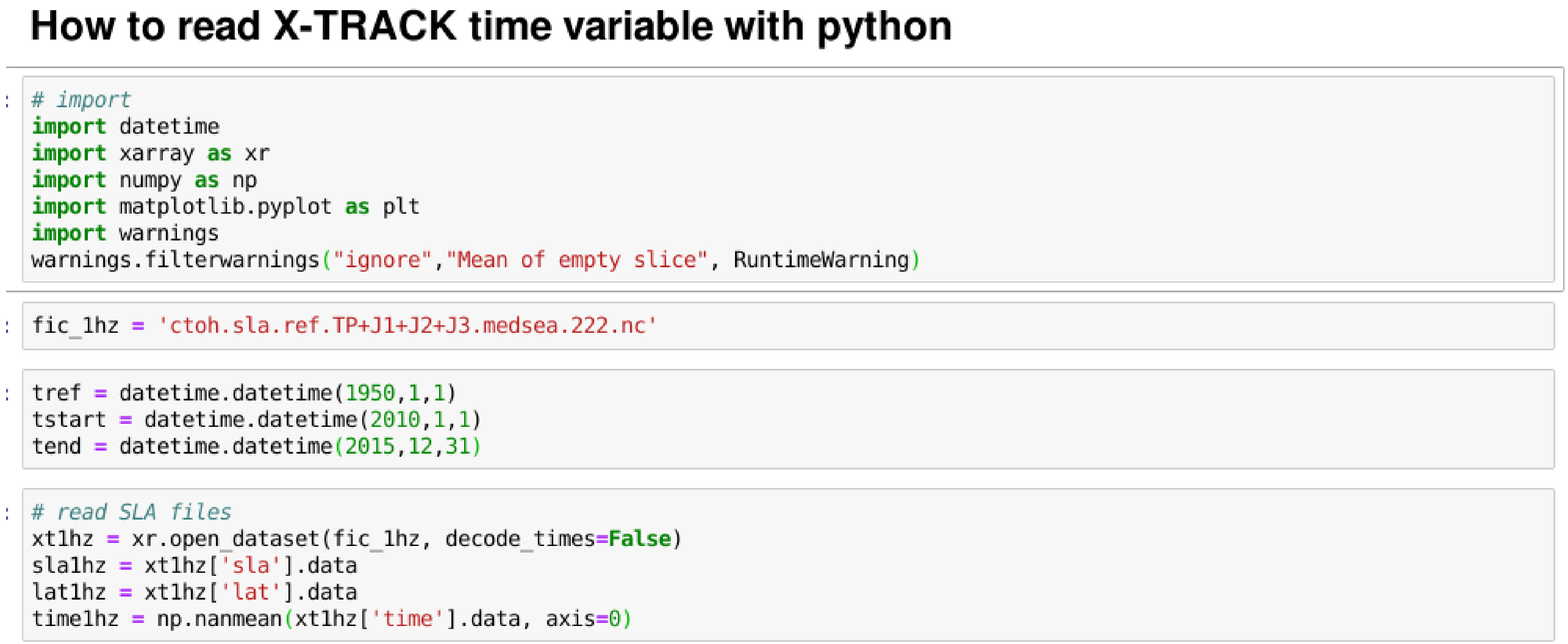


```
# map ssh in the Bay of Biscay along the track 248
projection_crs = ccrs.PlateCarree()
fig = plt.figure(figsize=(10, 6))
ax = plt.axes(projection=projection_crs)
ax.coastlines(resolution="10m")
plt.scatter(lonData[::5],latData[::5],c=mssData[::5],vmin=40, vmax=55, marker='o',s=50,alpha =0.7)
ax.set_extent([-8, -1, 43, 50], crs=ccrs.PlateCarree())
cb = plt.colorbar()
cb.set_label("mss (" + mssUnits + ")")
```



As written in the handbook time units is «days since 1950/01/01 »

- The **datetime** module supplies classes for manipulating dates and times.



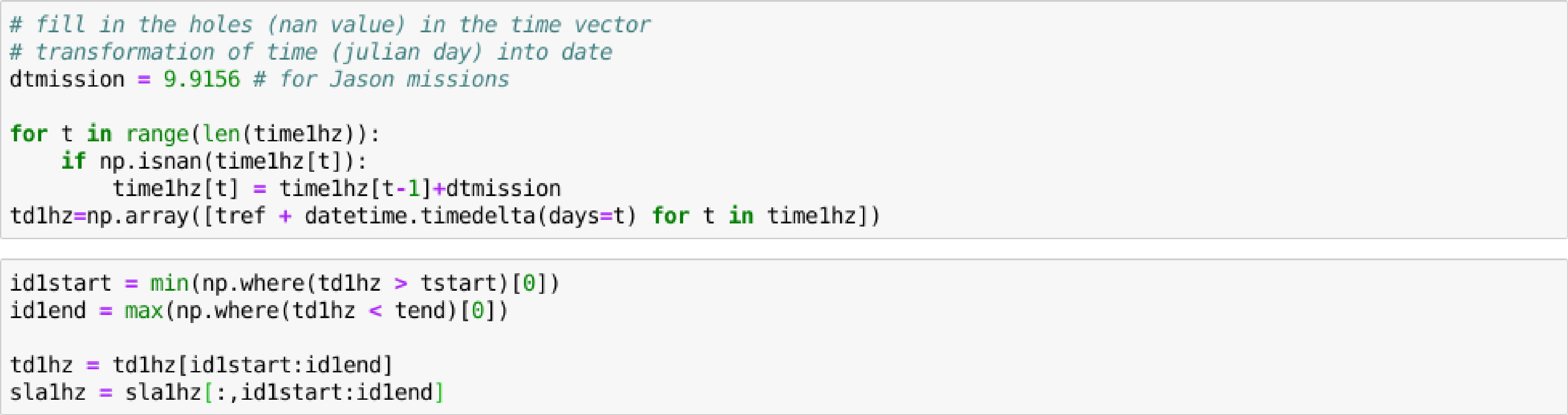
How to read X-TRACK time variable with python

```
# import
import datetime
import xarray as xr
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore","Mean of empty slice", RuntimeWarning)
```

```
fic_1hz = 'ctoh.sla.ref.TP+J1+J2+J3.medsea.222.nc'
```

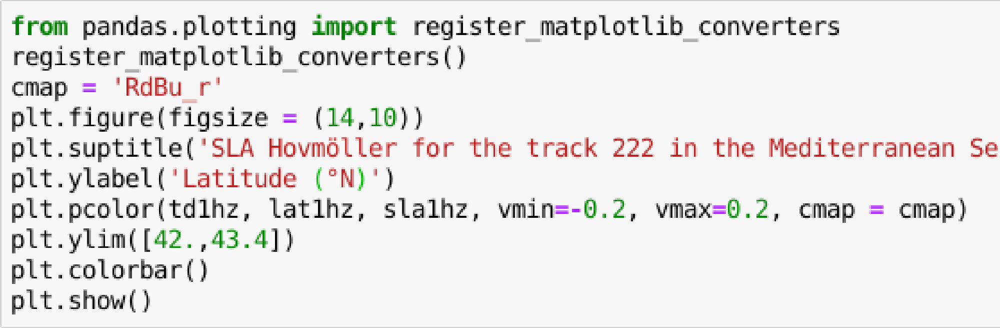
```
tref = datetime.datetime(1950,1,1)
tstart = datetime.datetime(2010,1,1)
tend = datetime.datetime(2015,12,31)
```

```
# read SLA files
xt1hz = xr.open_dataset(fic_1hz, decode_times=False)
sla1hz = xt1hz['sla'].data
lat1hz = xt1hz['lat'].data
time1hz = np.nanmean(xt1hz['time'].data, axis=0)
```

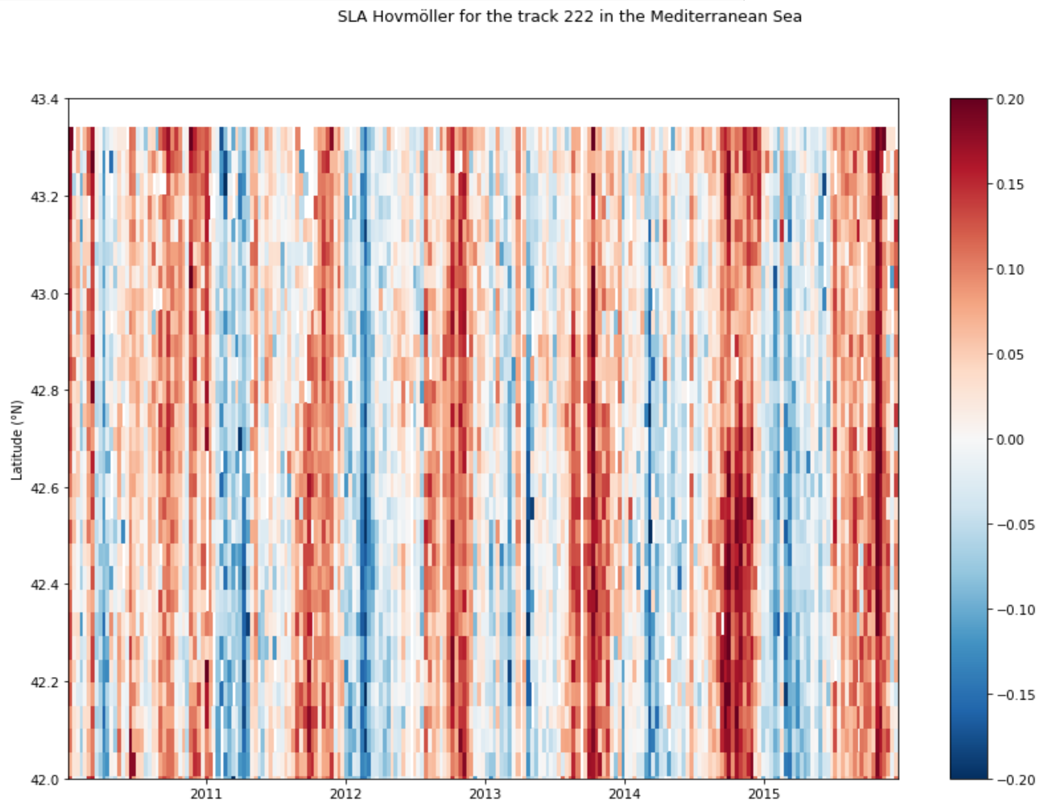


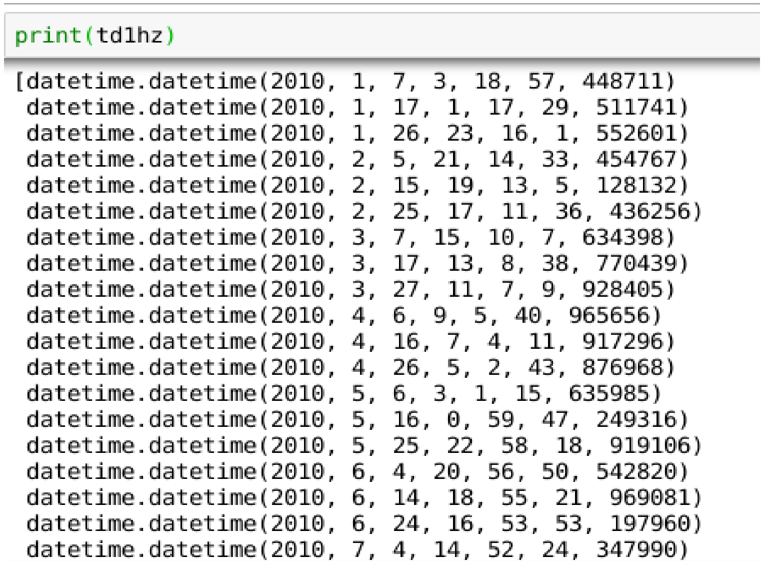
```
# fill in the holes (nan value) in the time vector
# transformation of time (julian day) into date
dtmission = 9.9156 # for Jason missions

for t in range(len(time1hz)):
    if np.isnan(time1hz[t]):
        time1hz[t] = time1hz[t-1]+dtmission
td1hz=np.array([tref + datetime.timedelta(days=t) for t in time1hz])
```

```
id1start = min(np.where(td1hz > tstart)[0])
id1end = max(np.where(td1hz < tend)[0])

td1hz = td1hz[id1start:id1end]
sla1hz = sla1hz[:,id1start:id1end]
```



```
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
cmap = 'RdBu_r'
plt.figure(figsize = (14,10))
plt.suptitle('SLA Hovmöller for the track 222 in the Mediterranean Sea')
plt.ylabel('Latitude (°N)')
plt.pcolor(td1hz, lat1hz, sla1hz, vmin=-0.2, vmax=0.2, cmap = cmap)
plt.ylim([42.,43.4])
plt.colorbar()
plt.show()
```





```
print(td1hz)
```

```
[datetime.datetime(2010, 1, 7, 3, 18, 57, 448711)
 datetime.datetime(2010, 1, 17, 1, 17, 29, 511741)
 datetime.datetime(2010, 1, 26, 23, 16, 1, 552601)
 datetime.datetime(2010, 2, 5, 21, 14, 33, 454767)
 datetime.datetime(2010, 2, 15, 19, 13, 5, 128132)
 datetime.datetime(2010, 2, 25, 17, 11, 36, 436256)
 datetime.datetime(2010, 3, 7, 15, 10, 7, 634398)
 datetime.datetime(2010, 3, 17, 13, 8, 38, 770439)
 datetime.datetime(2010, 3, 27, 11, 7, 9, 928405)
 datetime.datetime(2010, 4, 6, 9, 5, 40, 965656)
 datetime.datetime(2010, 4, 16, 7, 4, 11, 917296)
 datetime.datetime(2010, 4, 26, 5, 2, 43, 876968)
 datetime.datetime(2010, 5, 6, 3, 1, 15, 635985)
 datetime.datetime(2010, 5, 16, 0, 59, 47, 249316)
 datetime.datetime(2010, 5, 25, 22, 58, 18, 919106)
 datetime.datetime(2010, 6, 4, 20, 56, 50, 542820)
 datetime.datetime(2010, 6, 14, 18, 55, 21, 969081)
 datetime.datetime(2010, 6, 24, 16, 53, 53, 197960)
 datetime.datetime(2010, 7, 4, 14, 52, 24, 347990)
```