

Mekong_e2o_selection

September 12, 2020

```
[1]: import xarray as xr
import geopandas as gpd
import pandas as pd
import cartopy
import cartopy.crs as ccrs
import matplotlib.pyplot as plt
%matplotlib inline
```

```
[2]: # enter the right file name below. To browse over files of v2 reanalysis, check
      ↪out
      # https://wci.earth2observe.eu/thredds/catalog-earth2observe-model-wrr2.html
      # once you select a file, choose the OPeNDAP link and copy paste that below.
      ↪The example below is WaterGAP model of Kassel, which
      # proved to be one of the best performing models globally.
      url = r'https://wci.earth2observe.eu/thredds/dodsC/univk/wrr2-monthly-agg.nc'

      # Point shapefile provided below
      fn = r'C:\Users\Okay Mert\Downloads\Mekong.shp'

      # variable name inside the OPeNDAP dataset
      var_name = 'RivOut'
```

```
[3]: # first open shapefile and plot
f = plt.figure(figsize=(10, 15))
ax = plt.subplot(projection=ccrs.PlateCarree())
gdf = gpd.read_file(fn)
ax.coastlines()
gdf.plot(ax=ax, transform=ccrs.PlateCarree())
ax.set_extent([-180, 180, -60, 85])
```



[4]: gdf

```
[4]:
```

	id	name	geometry
0	0	None	POINT (104.9752879112359 11.5470708383469)
1	1	None	POINT (105.4688605007835 11.9755686018459)
2	2	None	POINT (105.9999875264923 12.26580940724881)
3	3	None	POINT (105.9874693811052 13.20772085926176)
4	4	None	POINT (105.9713746227504 13.89443269206525)
5	5	None	POINT (105.856028854541 14.61372634674075)
6	6	None	POINT (105.6548443751058 15.28755202428502)
7	7	None	POINT (105.4107405400578 15.7902184473536)
8	8	None	POINT (105.0235721863004 16.18905327091771)
9	9	None	POINT (104.7387843787444 16.56908703174231)
10	10	None	POINT (104.7414668384702 16.97445769362399)
11	11	None	POINT (104.7969043394701 17.39132774493914)
12	12	None	POINT (104.4276190505513 17.70634066026602)
13	13	None	POINT (104.1043826535922 18.1253561527819)
14	14	None	POINT (103.9197400091328 18.32791185075669)
15	15	None	POINT (103.4932289127303 18.43058550398012)
16	16	None	POINT (103.1431679185132 18.1691145385544)
17	17	None	POINT (103.0264809204408 17.97232841291136)
18	18	None	POINT (102.7443755726106 17.88895745822364)
19	19	None	POINT (102.53603786724 17.9744547091478)
20	20	None	POINT (102.2195076195953 18.1232316806317)
21	21	None	POINT (101.9476850340474 18.09093653109315)
22	22	None	POINT (101.614165874806 17.89236112227667)
23	23	None	POINT (101.4228170810321 18.02165188837153)
24	24	None	POINT (101.4621598236772 18.28758939238045)
25	25	None	POINT (101.7304057962574 18.51878497533075)
26	26	None	POINT (101.8408337216362 18.77633751007903)
27	27	None	POINT (101.8086442049266 19.11800233427789)

```

28 28 None POINT (101.8023851322331 19.50490370433594)
29 29 None POINT (102.0151936038134 19.84673317472475)
30 30 None POINT (102.2159310066276 20.04593516887909)
31 31 None POINT (101.4854078079675 19.82780855114178)
32 32 None POINT (100.3938073677265 20.29782566936618)
33 33 None POINT (100.6466152257392 20.89562232514636)
34 34 None POINT (100.5107772814734 20.88571857613899)
35 35 None POINT (101.0497386264072 21.44652915722218)
36 36 None POINT (101.1478684519234 21.80952316810949)
37 37 None POINT (100.5794384706529 22.49655604604722)
38 38 None POINT (99.20478963033517 25.79952297142704)
39 39 None POINT (98.8321512667593 28.45829869536473)

```

```

[5]: # looks good, now we retrieve the xy coordinates in separate lists for later use
xs, ys = list(gdf.geometry.x.values), list(gdf.geometry.y.values)
print(ys)

```

```

[11.5470708383469, 11.975568601845904, 12.265809407248812, 13.207720859261755,
13.894432692065246, 14.613726346740746, 15.287552024285016, 15.790218447353597,
16.189053270917707, 16.569087031742306, 16.974457693623993, 17.39132774493914,
17.706340660266015, 18.1253561527819, 18.32791185075669, 18.43058550398012,
18.169114538554403, 17.97232841291136, 17.888957458223636, 17.9744547091478,
18.1232316806317, 18.09093653109315, 17.89236112227667, 18.02165188837153,
18.287589392380454, 18.51878497533075, 18.776337510079035, 19.118002334277893,
19.50490370433594, 19.846733174724747, 20.04593516887909, 19.82780855114178,
20.297825669366176, 20.895622325146363, 20.885718576138995, 21.446529157222177,
21.809523168109493, 22.496556046047225, 25.799522971427038, 28.458298695364732]

```

```

[6]: # now open the OPeNDAP link and plot one time slice just to see if everything
      ↪ works as expected
ds = xr.open_dataset(url)
ds

```

```

[6]: <xarray.Dataset>
Dimensions:  (bnds: 2, lat: 720, lon: 1440, time: 420)
Coordinates:
  * lon      (lon) float32 -179.875 -179.625 -179.375 ... 179.625 179.875
  * lat      (lat) float32 -89.875 -89.625 -89.375 ... 89.375 89.625 89.875
  * time     (time) datetime64[ns] 1980-01-16 1980-02-15 ... 2014-12-16
Dimensions without coordinates: bnds
Data variables:
  time_bnds  (time, bnds) datetime64[ns] ...
  CanopInt   (time, lat, lon) float32 ...
  Evap       (time, lat, lon) float32 ...
  LAI        (time, lat, lon) float32 ...
  PotEvap    (time, lat, lon) float32 ...
  Precip     (time, lat, lon) float32 ...

```

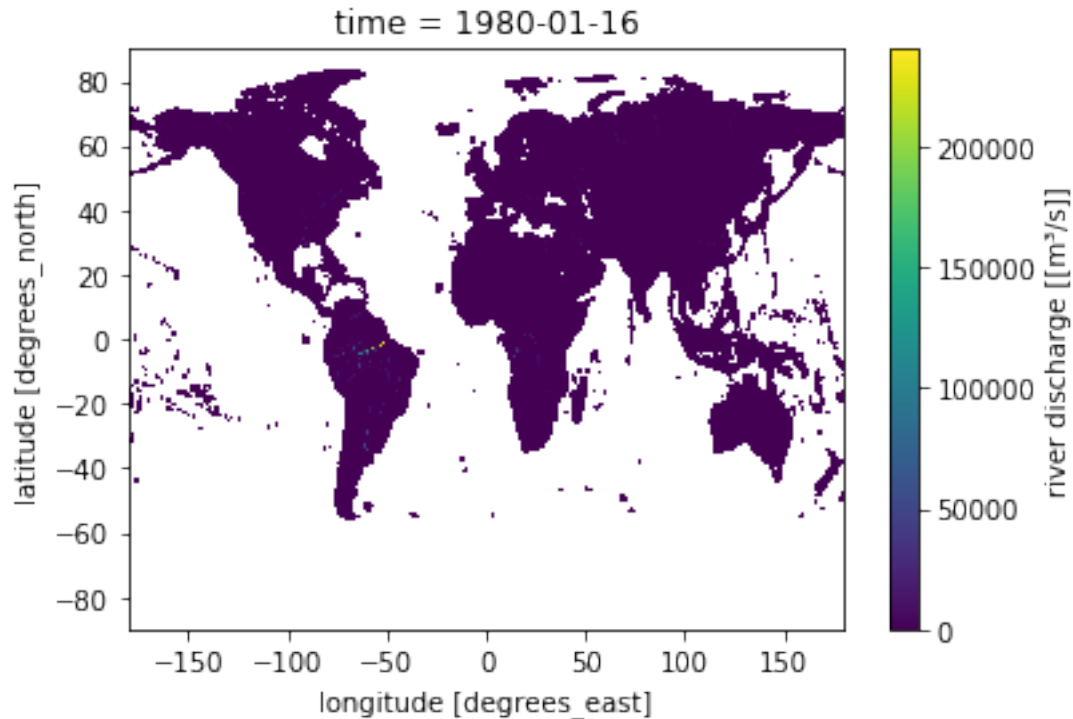
```

Qs          (time, lat, lon) float32 ...
Qsb         (time, lat, lon) float32 ...
Qsm         (time, lat, lon) float32 ...
RivOut      (time, lat, lon) float32 ...
RootMoist   (time, lat, lon) float32 ...
Runoff      (time, lat, lon) float32 ...
SWE         (time, lat, lon) float32 ...
SnowFrac    (time, lat, lon) float32 ...
SurfStor    (time, lat, lon) float32 ...
Attributes:
  _NCProperties:      version=1|netcdfversion=4.4.1.1|hdf5libversion=...
  CDI:              Climate Data Interface version 1.8.2 (http://mpime...
  history:          Tue May 16 16:39:10 2017: cdo mergetime /data/data...
  institution:      Center for Environmental Systems Research CESR, Un...
  Conventions:      CF-1.6
  title:            Earth2Observe water resources reanalysis version 2...
  project:          EARTH2OBSERVE
  contact:          www.earth2observe.eu
  id:               e2o_univk_wrr2_glob15_day_CanopInt_1980-1989.nc4
  naming_authority: earth2observe.eu
  Metadata_Conventions: Unidata Dataset Discovery v1.0
  summary:          Water resources analysis version 2 preformed by Wa...
  keywords:         Earth2Observe, reanalysis, ERA-Interim, WFDEI, Wat...
  comment:          -
  creator_name:     Gabriel Fink
  creator_url:      http://www.usf.uni-kassel.de/cesr/
  creator_email:    fink@cesr.de
  acknowledgment:   The Earth2Observe project has received funding fro...
  publisher_name:   Center for Environmental Systems Research CESR, Un...
  publisher_url:    http://www.usf.uni-kassel.de/cesr/
  publisher_email:  fink@cesr.de
  cdm_data_type:    grid
  references:       -
  license:          This file is made available under the Open Databas...
  NCO:              "4.6.2"
  CDO:              Climate Data Operators version 1.8.2rc1 (http://mp...
  _CoordSysBuilder: ucar.nc2.dataset.conv.CF1Convention
  frequency:        mon

```

```
[7]: ds[var_name][0].plot()
```

```
[7]: <matplotlib.collections.QuadMesh at 0x619357c828>
```

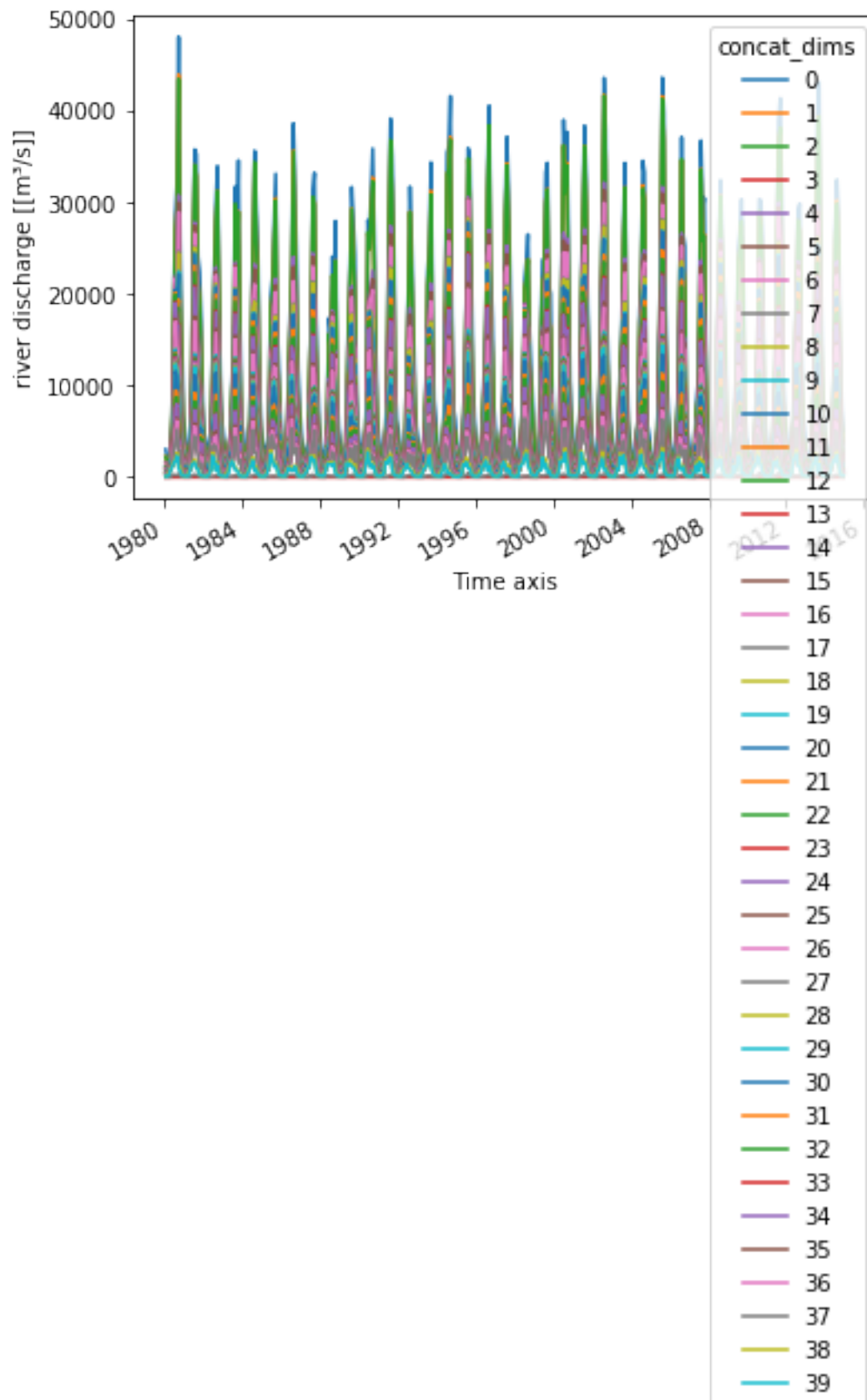


```
[8]: ds_sel = xr.concat([ds[var_name].sel(lon=x, lat=y, method='nearest') for x, y_
    ↪in zip(xs, ys)])
    ds_sel.plot.line(x='time')
```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
FutureWarning: the `dim` argument to `concat` will be required in a future
version of xarray; for now, setting it to the old default of 'concat_dim'
 """Entry point for launching an IPython kernel.

```
[8]: [<matplotlib.lines.Line2D at 0x6193604518>,
    <matplotlib.lines.Line2D at 0x61981c4518>,
    <matplotlib.lines.Line2D at 0x619826f240>,
    <matplotlib.lines.Line2D at 0x619826f390>,
    <matplotlib.lines.Line2D at 0x619826f4e0>,
    <matplotlib.lines.Line2D at 0x61981f7a20>,
    <matplotlib.lines.Line2D at 0x61981fd5c0>,
    <matplotlib.lines.Line2D at 0x61981fcac8>,
    <matplotlib.lines.Line2D at 0x61981fccf8>,
    <matplotlib.lines.Line2D at 0x61981fe518>,
    <matplotlib.lines.Line2D at 0x61981c4278>,
    <matplotlib.lines.Line2D at 0x61981fec50>,
    <matplotlib.lines.Line2D at 0x61981fef28>,
    <matplotlib.lines.Line2D at 0x61981fe2b0>]
```

```
<matplotlib.lines.Line2D at 0x61981fe6a0>,  
<matplotlib.lines.Line2D at 0x61981fea90>,  
<matplotlib.lines.Line2D at 0x61981fee80>,  
<matplotlib.lines.Line2D at 0x619821b2e8>,  
<matplotlib.lines.Line2D at 0x61981ff1d0>,  
<matplotlib.lines.Line2D at 0x61981ff160>,  
<matplotlib.lines.Line2D at 0x61981ff2b0>,  
<matplotlib.lines.Line2D at 0x61981ff4e0>,  
<matplotlib.lines.Line2D at 0x61981ff630>,  
<matplotlib.lines.Line2D at 0x61981ff780>,  
<matplotlib.lines.Line2D at 0x61981ff8d0>,  
<matplotlib.lines.Line2D at 0x61981ffa20>,  
<matplotlib.lines.Line2D at 0x61981ffb70>,  
<matplotlib.lines.Line2D at 0x61981ffcc0>,  
<matplotlib.lines.Line2D at 0x61981ffe10>,  
<matplotlib.lines.Line2D at 0x61981fff60>,  
<matplotlib.lines.Line2D at 0x61981ec0f0>,  
<matplotlib.lines.Line2D at 0x61981ec240>,  
<matplotlib.lines.Line2D at 0x61981ec390>,  
<matplotlib.lines.Line2D at 0x61981ec4e0>,  
<matplotlib.lines.Line2D at 0x61981ec630>,  
<matplotlib.lines.Line2D at 0x61981ec780>,  
<matplotlib.lines.Line2D at 0x61981ec8d0>,  
<matplotlib.lines.Line2D at 0x61981eca20>,  
<matplotlib.lines.Line2D at 0x61981ecb70>,  
<matplotlib.lines.Line2D at 0x61981eccc0>]
```



```
[10]: for n in range(len(ds_sel)):
        fn_out = 'timeseries_{:02d}.csv'.format(n + 1)
        df = ds_sel[n].drop(['lat', 'lon']).to_dataframe().rename(columns={'RivOut':
↪ 'flow_{:02d}'.format(0)})
        df.to_csv(fn_out)
```

```
[ ]:
```