


Baixando uma série temporal de vento do ERA5 do Copernicus pelo Jupyter

1) Acesse o site do Copernicus

← → ↻ cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview



Home Search Datasets Applications Toolbox Support Live

ERA5 hourly data on single levels from 1959 to present

NOTIFICATION 2022-12-16: Due to the update in the interpolation software since the move to the new HPC, for October 2022 the final ERA5 release slightly differs from the preliminary ERA5T release for isolated grid points (mainly at 90N and 90S). There is no meteorological impact.

Overview

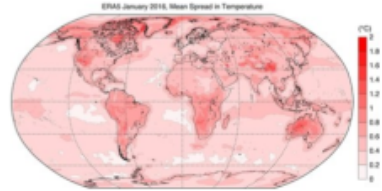
Download data

Quality assessment

Documentation

ERA5 is the fifth generation ECMWF reanalysis for the global climate and weather for the past 4 to 7 decades. Currently data is available from 1950, with Climate Data Store entries for 1950-1978 (preliminary back extension) and from 1959 onwards (final release plus timely updates, this page). ERA5 replaces the ERA-Interim reanalysis.

Reanalysis combines model data with observations from across the world into a globally complete and consistent dataset using the laws of physics. This principle, called data assimilation, is based on the method used by numerical weather prediction centres, where every so many hours (12 hours at ECMWF) a previous forecast is combined with newly available observations in an optimal way to produce a new best estimate of the state of the atmosphere, called analysis, from which an updated, improved forecast is issued. Reanalysis works in the same way, but at reduced resolution to allow for the provision of a dataset spanning back several decades. Reanalysis does not have the constraint of



2) No 'Download Data', escolher as opções... neste caso, somente as componentes u e v do vento a 10 m

3) Selecione a sub-região de interesse e escolha a opção do formato.

4) Clique no 'Show API request', e irá aparecer o código em Python da solicitação... copie e cole no Jupyter (célula abaixo).

Para funcionar, tem que instalar o pacote 'cdsapi', e ter uma conta e estar logado! Entendendo o código, faça modificações se achar necessário... neste caso estou baixando dados para somente um ponto da malha... a resolução é 0,5 graus, então é relativamente fácil achar um ponto próximo.

In [1]:

```
import cdsapi

c = cdsapi.Client()

c.retrieve(
    'reanalysis-era5-single-levels',
    {
        'product_type': 'reanalysis',
        'format': 'netcdf',
        'variable': [
            '10m_u_component_of_wind', '10m_v_component_of_wind',
        ],
        'year': [
            '2022', '2023',
        ],
    },
```

```

'month': [
    '01', '02', '03',
    '04', '05', '06',
    '07', '08', '09',
    '10', '11', '12',
],
'day': [
    '01', '02', '03',
    '04', '05', '06',
    '07', '08', '09',
    '10', '11', '12',
    '13', '14', '15',
    '16', '17', '18',
    '19', '20', '21',
    '22', '23', '24',
    '25', '26', '27',
    '28', '29', '30',
    '31',
],
'time': [
    '00:00', '01:00', '02:00',
    '03:00', '04:00', '05:00',
    '06:00', '07:00', '08:00',
    '09:00', '10:00', '11:00',
    '12:00', '13:00', '14:00',
    '15:00', '16:00', '17:00',
    '18:00', '19:00', '20:00',
    '21:00', '22:00', '23:00',
],
'area': [
    -21, -40.5, -21,
    -40.5,
],
},
'vento_ERA5.nc') # eu mudei o nome para minha conveniência... e outros parâmet

```

```

2023-01-21 16:27:01,226 INFO Welcome to the CDS
2023-01-21 16:27:01,229 INFO Sending request to https://cds.climate.copernicus.eu/api/v2/resources/reanalysis-era5-single-levels
2023-01-21 16:27:01,565 INFO Request is queued
2023-01-21 16:27:02,825 INFO Request is running
2023-01-21 16:37:25,933 INFO Request is completed
2023-01-21 16:37:25,938 INFO Downloading https://download-0005-clone.copernicus-climate.eu/cache-compute-0005/cache/data6/adaptor.mars.internal-1674329769.4357646-1514-9-21723732-0526-44b6-b5fd-64b956dbd1fb.nc to vento_ERA5.nc (108.5K)
2023-01-21 16:37:28,252 INFO Download rate 47K/s

```

```

Out[1]: Result(content_length=111132,content_type=application/x-netcdf,location=https://download-0005-clone.copernicus-climate.eu/cache-compute-0005/cache/data6/adaptor.mars.internal-1674329769.4357646-1514-9-21723732-0526-44b6-b5fd-64b956dbd1fb.nc)

```

Explorando o conteúdo do arquivo descarregado

Isso é melhor feito em um outro notebook, mas vamos continuar assim por conveniência! Digo, se precisar reiniciar o kernel, terá que executar o download de novo, ou alternativamente comentar toda a célula ou transformar em markdown.

```
In [60]: import netCDF4
import matplotlib.pyplot as plt
import numpy as np
```

Carrega e mostra todos os campos do arquivo

```
In [67]: nc = netCDF4.Dataset('Vento_ERA5.nc')
```

```
# imprime o nome das variáveis
for i in nc.variables:
    print(i)
```

```
print()
print()
```

```
#imprime metadata
for i in nc.variables:
    print(nc.variables[i])
    print()
    print()
```

```
longitude
latitude
expver
time
u10
v10
```

```
<class 'netCDF4._netCDF4.Variable'>
float32 longitude(longitude)
    units: degrees_east
    long_name: longitude
unlimited dimensions:
current shape = (1,)
filling on, default _FillValue of 9.969209968386869e+36 used
```

```
<class 'netCDF4._netCDF4.Variable'>
float32 latitude(latitude)
    units: degrees_north
    long_name: latitude
unlimited dimensions:
current shape = (1,)
filling on, default _FillValue of 9.969209968386869e+36 used
```

```
<class 'netCDF4._netCDF4.Variable'>
int32 expver(expver)
    long_name: expver
unlimited dimensions:
current shape = (2,)
filling on, default _FillValue of -2147483647 used
```

```
<class 'netCDF4._netCDF4.Variable'>
int32 time(time)
    units: hours since 1900-01-01 00:00:00.0
    long_name: time
    calendar: gregorian
unlimited dimensions:
current shape = (9140,)
filling on, default _FillValue of -2147483647 used
```

```
<class 'netCDF4._netCDF4.Variable'>
int16 u10(time, expver, latitude, longitude)
    scale_factor: 0.0003555125781840113
    add_offset: 0.46690140630856425
    _FillValue: -32767
    missing_value: -32767
    units: m s**-1
    long_name: 10 metre U wind component
unlimited dimensions:
current shape = (9140, 2, 1, 1)
filling on
```

```
<class 'netCDF4._netCDF4.Variable'>
int16 v10(time, expver, latitude, longitude)
    scale_factor: 0.00034962355596667425
    add_offset: 1.2711035695696729
    _FillValue: -32767
    missing_value: -32767
    units: m s**-1
    long_name: 10 metre V wind component
unlimited dimensions:
current shape = (9140, 2, 1, 1)
filling on
```

Verificando as coordenas, e de fato, foi descarregado só um ponto da malha!

```
In [9]: lon = nc.variables['longitude'][:]
        lat = nc.variables['latitude'][:]

        print(lon, lat)
```

```
[-40.5] [-21.]
```

Convertendo o tempo para 'datetime'

```
In [26]: tempo = netCDF4.num2date(nc.variables['time'], nc.variables['time'].units, only_use_
        print(tempo[0:5])
```

```
[real_datetime(2023, 1, 16, 0, 0) real_datetime(2023, 1, 16, 1, 0)
 real_datetime(2023, 1, 16, 2, 0) real_datetime(2023, 1, 16, 3, 0)
 real_datetime(2023, 1, 16, 4, 0)]
```

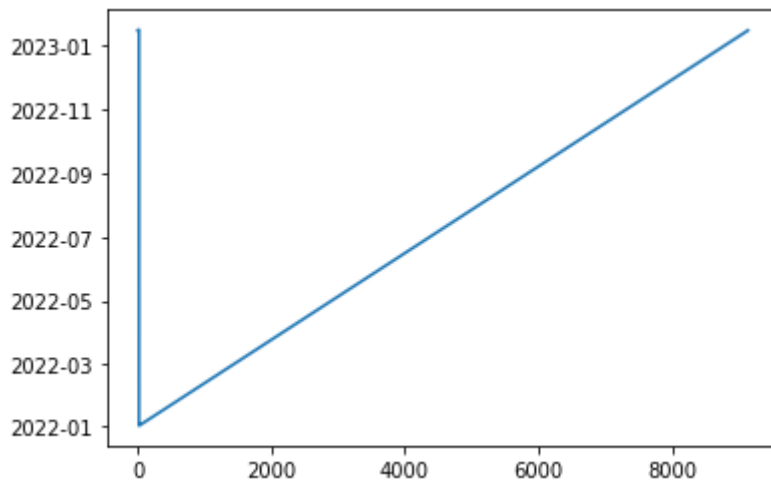
Deveria começar em 2022, 1, 1, 0, 0, 0 como solicitado no API - Algo estranho!

```
In [45]: print(tempo[0])
        print(tempo[-1])
```

```
2023-01-16 00:00:00
2023-01-15 23:00:00
```

```
In [27]: plt.plot(tempo)
```

```
Out[27]: [<matplotlib.lines.Line2D at 0x152794afdc0>]
```



De fato, bem estranho, e também considerando que V10 e U10 tem estas dimensões "int16 v10(time, expver, latitude, longitude)"

expver?

People also ask :

What is Expver in ERA5 data?

expver is **used to tell the difference between the initial release** (expver=5, called ERA5T) and validated ERA5 data (expver=1). Feb 18, 2020

<https://confluence.ecmwf.int/pages/viewpage.action>

[ERA5 and ERA5T - ECMWF Confluence Wiki](#)

```
In [68]: u = nc.variables['u10'][:]
u.shape
```

```
Out[68]: (9140, 2, 1, 1)
```

As últimas duas dimensões são referentes as coordenadas, mas que raios seria a 2.a?

Explorando...

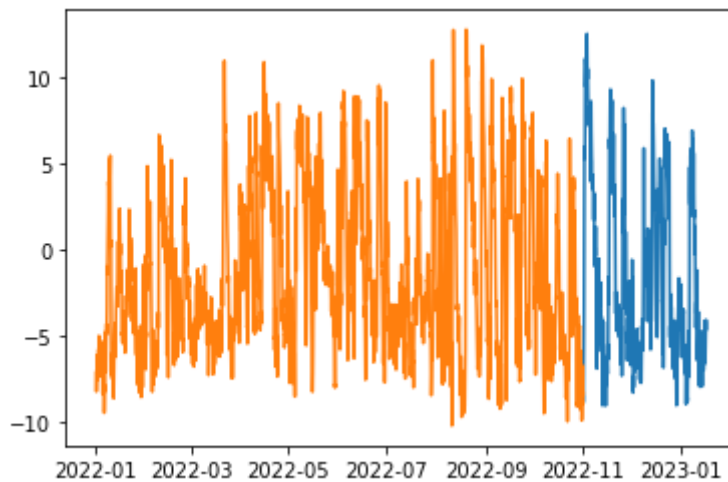
```
In [70]: u = nc.variables['u10'].squeeze()
v = nc.variables['v10'].squeeze()

print(type(u))

plt.plot(tempo,v[:,0])
plt.plot(tempo,v[:,1])
```

```
<class 'numpy.ma.core.MaskedArray'>
```

```
Out[70]: [<matplotlib.lines.Line2D at 0x1527a5d1c10>]
```



In [69]:

```
junta = []
for i in range(len(v[:,0])):
    print(i, tempo[i], v[i,0], v[i,1])

    if i == 30:
        break
```

```
0 2023-01-16 00:00:00 -6.452780028846094 --
1 2023-01-16 01:00:00 -6.576197144102331 --
2 2023-01-16 02:00:00 -6.335306514041292 --
3 2023-01-16 03:00:00 -6.106652708439087 --
4 2023-01-16 04:00:00 -5.888137985959915 --
5 2023-01-16 05:00:00 -5.79408924940488 --
6 2023-01-16 06:00:00 -5.729758515107012 --
7 2023-01-16 07:00:00 -5.33783050886837 --
8 2023-01-16 08:00:00 -5.03820312140493 --
9 2023-01-16 09:00:00 -4.608515771121889 --
10 2023-01-16 10:00:00 -4.291057582304148 --
11 2023-01-16 11:00:00 -4.165892349268079 --
12 2023-01-16 12:00:00 -4.08058420161221 --
13 2023-01-16 13:00:00 -4.223929859558547 --
14 2023-01-16 14:00:00 -4.186520139070112 --
15 2023-01-16 15:00:00 -4.088625543399443 --
16 2023-01-16 16:00:00 -4.08373081361591 --
17 2023-01-16 17:00:00 -4.138621711902678 --
18 2023-01-16 18:00:00 -4.359583799273617 --
19 2023-01-16 19:00:00 -4.593481958215321 --
20 2022-01-01 00:00:00 -- -7.114267796735042
21 2022-01-01 01:00:00 -- -7.650939955143887
22 2022-01-01 02:00:00 -- -7.879943384302059
23 2022-01-01 03:00:00 -- -8.211036891802499
24 2022-01-01 04:00:00 -- -8.136567074381597
25 2022-01-01 05:00:00 -- -8.108247566348297
26 2022-01-01 06:00:00 -- -8.037273984487062
27 2022-01-01 07:00:00 -- -7.7995299664297235
28 2022-01-01 08:00:00 -- -7.476827424272484
29 2022-01-01 09:00:00 -- -7.15866998834281
30 2022-01-01 10:00:00 -- -7.267402914248445
```

Creio que isso, (tempo truncado e 2 colunas para a mesma variável, alternando) tem a ver porque eu pedi todos os dados até a data da solicitação. O link abaixo descreve O que é o 'expver' e porque isto é feito.

<https://confluence.ecmwf.int/pages/viewpage.action?pageId=173385064>

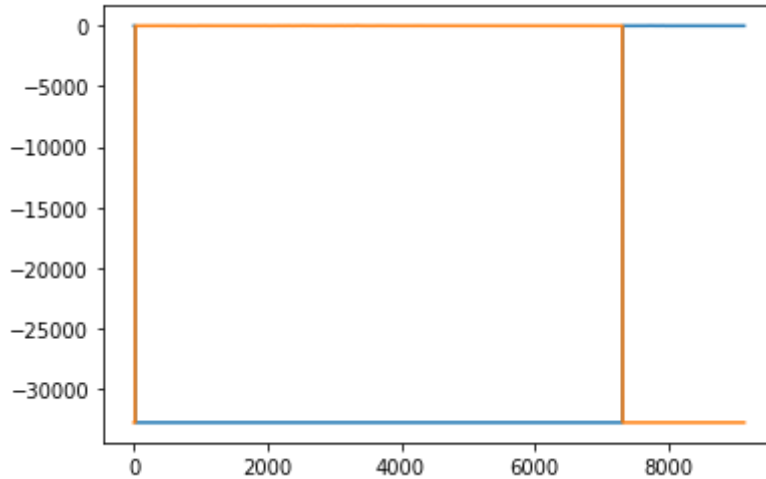
Então o caso é juntar em uma única coluna e colocar em ordem cronológica!

Transformando em np.array, o símbolo '--' vira um flag numérico de ausência e pode ser usado para juntar os dados

```
In [72]: u = np.array(u)
v = np.array(v)

plt.plot(u)
```

```
Out[72]: [<matplotlib.lines.Line2D at 0x1527b89ba00>,
<matplotlib.lines.Line2D at 0x1527b89ba30>]
```



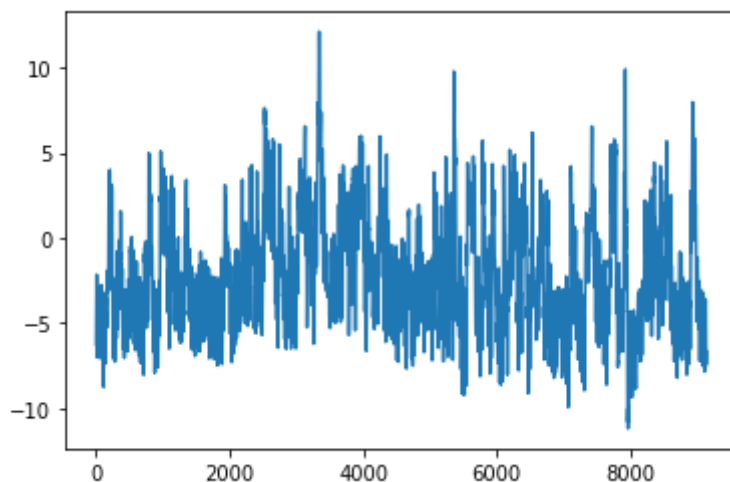
```
In [76]: flag = np.min(u)
print(flag)

u2 = []
v2 = []
for i in range(len(tempo)):
    if u[i,0] != flag:
        u2.append(u[i,0])
        v2.append(v[i,0])
    else:
        u2.append(u[i,1])
        v2.append(v[i,1])

plt.plot(u2)
```

```
-32767.0
```

```
Out[76]: [<matplotlib.lines.Line2D at 0x1527b991670>]
```



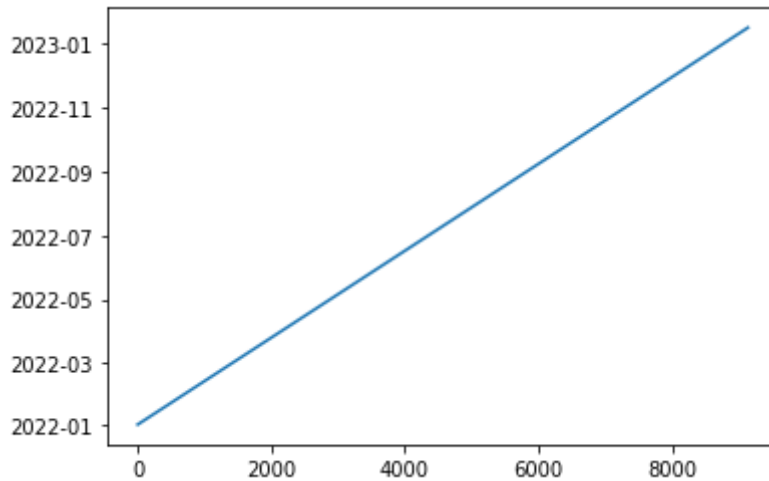
Colocando em ordem cronológica...

```
In [81]: # monta a matriz
junta = np.hstack((
    np.atleast_2d(np.array(tempo)).T,
    np.atleast_2d(np.array(u2)).T,
    np.atleast_2d(np.array(v2)).T
))

# ordena usando o tempo
junta = junta[junta[:,0].argsort()]

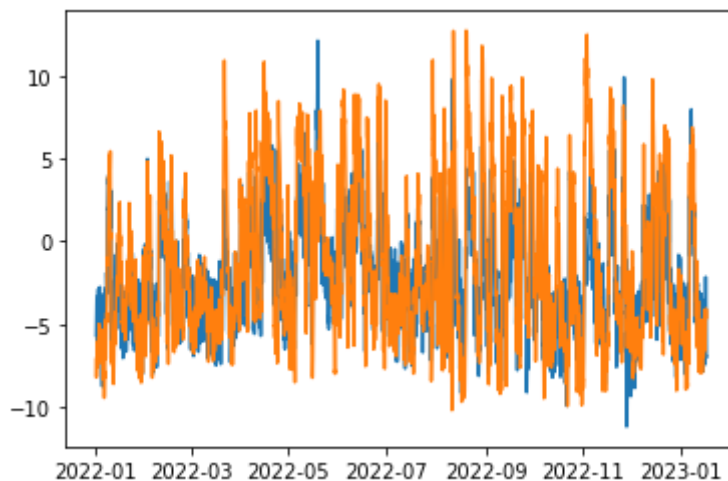
plt.plot(junta[:,0])
```

Out[81]: [<matplotlib.lines.Line2D at 0x1527bb75fd0>]



```
In [82]: plt.plot(junta[:,0], junta[:,1])
plt.plot(junta[:,0], junta[:,2])
```

Out[82]: [<matplotlib.lines.Line2D at 0x1527b9a3580>]



```
In [86]: import pickle

# [tempo, u, v]
with open('Vento_ERA5_2022_2023.pkl', 'wb') as io:
    pickle.dump(junta, io)
```