

# Práctica 4 - GCOMP

Celia Rubio Madrigal

16 de marzo de 2022

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Material usado y metodología</b>	<b>2</b>
2.1. Apartado <i>i</i> ) . . . . .	2
2.2. Apartado <i>ii</i> ) . . . . .	2
2.3. Apartado <i>iii</i> ) . . . . .	2
<b>3. Resultados y conclusiones</b>	<b>2</b>
3.1. Apartado <i>i</i> ) . . . . .	2
<b>4. Código</b>	<b>3</b>

1. Introducción
2. Material usado y metodología
  - 2.1. Apartado *i*)
  - 2.2. Apartado *ii*)
3. Resultados y conclusiones
  - 3.1. Apartado *i*)
  - 3.2. Apartado *ii*)

## 4. Código

```
# -*- coding: utf-8 -*-
"""
Referencias:

Fuente primaria del reanálisis
https://psl.noaa.gov/data/gridded/data.ncep.reanalysis2.pressure.html

Altura geopotencial en niveles de presión
https://psl.noaa.gov/cgi-bin/db_search/DBListFiles.pl?did=59&tid=97457&vid=1498

Temperatura en niveles de presión:
https://psl.noaa.gov/cgi-bin/db_search/DBListFiles.pl?did=59&tid=97457&vid=4237

Temperatura en niveles de superficie:
https://psl.noaa.gov/cgi-bin/db_search/DBListFiles.pl?did=59&tid=97457&vid=1497

"""

import datetime as dt # Python standard library datetime module
import numpy as np
import matplotlib.pyplot as plt
from netCDF4 import Dataset
from sklearn.decomposition import PCA

f = Dataset("air.2021.nc", "r", format="NETCDF4")
level = f.variables['level'][:].copy()
lats = f.variables['lat'][:].copy()
lons = f.variables['lon'][:].copy()
air21 = f.variables['air'][:].copy()
f.close()

f = Dataset("hgt.2021.nc", "r", format="NETCDF4")
time21 = f.variables['time'][:].copy()
hgt21 = f.variables['hgt'][:].copy()
f.close()

f = Dataset("air.2022.nc", "r", format="NETCDF4")
air22 = f.variables['air'][:].copy()
f.close()

f = Dataset("hgt.2022.nc", "r", format="NETCDF4")
time22 = f.variables['time'][:].copy()
hgt22 = f.variables['hgt'][:].copy()
f.close()

""" Apartado i) """

"""
Distribución espacial de la temperatura en el nivel de 500hPa, para el primer día
"""

plt.contour(lons, lats, air21[0,0,:,:])
plt.savefig("1")

hgt21b = hgt21[:,level==500.,:,:].reshape(len(time21),len(lats)*len(lons))
n_components=4

Y = hgt21b.transpose()
pca = PCA(n_components=n_components)
```

```

pca.fit(Y)
print(pca.explained_variance_ratio_)
out = pca.singular_values_

Element_pca0 = pca.fit_transform(Y)
Element_pca0 = Element_pca0.transpose(1,0).reshape(n_components,len(lats),len(lons))
print(pca.explained_variance_ratio_)

fig = plt.figure()
fig.subplots_adjust(hspace=0.4, wspace=0.4)
for i in range(4):
    ax = fig.add_subplot(2, 2, i+1)
    ax.text(0.5, 90, 'PCA-'+str(i), fontsize=18, ha='center')
    plt.contour(lons, lats, Element_pca0[i,:,:])
plt.savefig("2")

""" Apartado ii) """

dt_time21 = np.array([dt.date(1800, 1, 1) + dt.timedelta(hours=t) for t in time21])
dt_time22 = np.array([dt.date(1800, 1, 1) + dt.timedelta(hours=t) for t in time22])
dia0 = dt.date(2022, 1, 11)
hgtc = np.ma.masked_where(((lats > 20) & (lats < -20), hgt22))
hgt0 = hgt22[dt_time22==dia0,:,:,:][0]
air0 = air22[dt_time22==dia0,:,:,:][0]
[ l>-20 and l>20 for l in lats],[ l>30 and l<50 for l in lons]

distancias = [ np.norm() ]

```