

[illegible]

Hazırlayan: Hidrosaf Yazılım Bilişim Danışmanlık LTD. ŞTİ.

## İçerik

Kısaltmalar.....	2
Kod Bloğu Listesi.....	3
1. GENEL BİLGİ.....	3
1.1. Açıklama .....	4
1.2. Ürünlerin Tanıtımı .....	4
1.2.1. ERA5 .....	4
2. KOD VE ÇALIŞMA PRENSİBİ.....	4
2.1. Import ve Datetime .....	4
2.2. netCDF ve Shapefile’ın okunması .....	5
2.3. Alansal İstatistik .....	6

## Kısaltmalar

<b>HSAF</b>	: Satellite Application Facility on Support to Operational Hydrology and Water Management (H-SAF)
<b>MGM</b>	: T.C. TARIM VE ORMAN BAKANLIĞI Meteoroloji Genel Müdürlüğü
<b>TSMS</b>	: Turkish State Meteorological Service
<b>EUMETSAT</b>	: European Organisation for the Exploitation of Meteorological Satellites
<b>FMI</b>	: Finnish Meteorological Institute
<b>UTC</b>	: Coordinated Universal Time
<b>SCA</b>	: Karla Kaplı Alan, Snow Covered Area
<b>FSC</b>	: Etikili Karla Kaplı Alan, Fractional Snow Covered Area
<b>SWE</b>	: Kar Su Eşdeğeri, Snow Water Equivalent
<b>HDF</b>	: Hierarchical Data Format
<b>GRIB2</b>	: GRIdded Binary or General Regularly-distributed Information in Binary
<b>GeoTiff</b>	: Georeferenced Tagged Image File Format
<b>MSG</b>	: Meteosat Second Generation
<b>NWCSAF</b>	: Nowcasting and Very Short Range Forecasting
<b>AVHRR</b>	: Advanced Very High Resolution Radiometer
<b>EPS</b>	: EUMETSAT Polar System (EPS)
<b>METOP</b>	: Meteorological Operational satellite programme
<b>M01</b>	: METOP – B- EPS Ürünü
<b>SSM/I/S</b>	: Special Sensor Microwave Imager/Sounder
<b>DMSP</b>	: Defense Meteorological Satellite Program
<b>ECMWF</b>	: European Centre for Medium-Range Weather Forecasts
<b>BUFR</b>	: Binary Universal Form for the Representation
<b>HUT</b>	: Helsinki University of Technology
<b>PNG</b>	: Portable Network Graphics

## Kod Bloğu Listesi

Kod Bloğu 1: Gerekli Kütüphaneler.....	4
Kod Bloğu 2: Datetime ve Gerekli Tarihlerin Listeleri.....	5
Kod Bloğu 3: netCDF ve Shapefile'ın yerleri.....	5
Kod Bloğu 4: Verilerin Okunması.....	6
Kod Bloğu 5: Zaman Serilerinin okunması.....	6
Kod Bloğu 6: Projeksiyonların Okunması.....	6
Kod Bloğu 7: Burdur Havzası'nın Yağış İstatistiği (Toplam/Sum).....	7
Kod Bloğu 8: Burdur Havzası'nın Sıcaklık İstatistiği (Ortalama/Average).....	7

## 1. GENEL BİLGİ

### 1.1. Açıklama

Bu kılavuz, Burdur Havzası için 1979-2020 arasında ERA5'ten alınan sıcaklık ve yağış verilerinin istatistiği için hazırlanmıştır.

### 1.2. Ürünlerin Tanıtımı

#### 1.2.1. ERA5

ERA5 bir reanalysis ürünü olup, 1979'dan günümüze kadar saatlik veya aylık biçimde, kar yoğunluğu, derinliği vb. parametreler dışında da oldukça geniş bir parameter seçeneğiyle netCDF datası olarak indirilebilir. Kar yoğunluğu verisinin çözünürlüğü 0.5 derecedir. Alan olarak bütün dünyayı kapsamaktadır, ve projeksiyonu WGS84'tür, ancak meridyen sistemi alıştığımız biçimde -180,180 değil 0,360 tır.

ERA5 aylık olarak aşağıdaki siteden indirilebilir:

<https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels-monthly-means?tab=overview>

Bu çalışma için 2m temperature ve total precipitation, 1979'dan 2020'nin sonuna kadar tüm yıl ve aylar,ve netCDF seçilmiştir.

## 2. KOD VE ÇALIŞMA PRENSİBİ

### 2.1. Import ve Datetime

Öncelikle gerekli kütüphaneleri indirmek için aşağıdaki kod bloğu yazılmıştır:

```
import pandas as pd
import datetime
from datetime import timedelta
import xarray as xr
import rasterio as rio
import geopandas as gpd
import rasterstats as rstats
```

*Kod Bloğu 1: Gerekli Kütüphaneler*

netCDF verileri zaman serilerinden oluşmaktadır, ve bu çalışma için 1979'dan 2020'ye kadarki tüm yılların bir listesi gerekmektedir. Bunun içinse aşağıdaki kod yazılmıştır:

```

start = datetime.datetime.strptime("19790101", "%Y%m%d")
end = datetime.datetime.strptime("20201231", "%Y%m%d")
date_list2 = [start + timedelta(n) for n in range(int((end - start).days) + 1)]

df3 = pd.DataFrame({'date':date_list2})

df3['month_year'] = df3['date'].dt.to_period('M')
df3['year'] = df3['date'].dt.year
df3['months'] = df3['date'].dt.month

dateList3 = df3['month_year'].astype(str).values.tolist()
dateList3 = sorted(list(set(dateList3)))
yearList = df3['year'].astype(str).values.tolist()
monthList1 = df3['months'].astype(str).values.tolist()
uniquemonthList = sorted(map(int, (list(set(monthList1)))))
uniqueyearList = sorted(list(set(yearList)))

yearList = [ele for ele in uniqueyearList for i in range(12)]

monthList = sorted(list(map(int, uniquemonthList)))*42

```

*Kod Bloğu 2: Datetime ve Gerekli Tarihlerin Listeleri*

## 2.2. netCDF ve Shapefile'in okunması

Aşağıdaki kodda kullanılacak olan her iki netCDF verisinin (2m temperature ve total precipitation), ve Burdur Havzasının yerleri belirlenmiştir. ERA5 ve shapefile'in projeksiyonu WGS84'tür.

```

shp_fo = "D:/Drivers/Hidrosaf_notes/havza/burdur_wgs.shp"
nc_fo = "D:/Drivers/Hidrosaf_notes/havza/temperature.nc"
nc_fo2 = "D:/Drivers/Hidrosaf_notes/havza/total_precipitation.nc"

```

*Kod Bloğu 3: netCDF ve Shapefile'in yerleri*

Daha sonrasında hem shapefile hem de netCDF verileri aşağıdaki gibi okunmuştur:

```

# load and read shp-file with geopandas

shp_df = gpd.read_file(shp_fo)

# load and read netCDF-file to dataset and get datarray for variable

#total precipitation
nc_ds = xr.open_dataset(nc_fo)
nc_var = nc_ds['tp']
nc_var = nc_var[:,0,:,:]

#2m temperature
nc_ds2 = xr.open_dataset(nc_fo2)
nc_var2 = nc_ds2['t2m']
nc_var2 = nc_var2[:,0,:,:]

```

Kod Bloğu 4: Verilerin Okunması

Daha sonrasında her iki netCDF dosyasının da zaman serileri aşağıdaki gibi okunmuştur:

```

years_ = nc_ds['time'] #precipitation
years_2 = nc_ds2['time'] #temperature

```

Kod Bloğu 5: Zaman Serilerinin okunması

Ve rasterio modülü ile her iki netCDF dosyasının projeksiyonu da okunmuştur:

```

# get affine of nc-file with rasterio
affine = rio.open(nc_fo).transform
affine2 = rio.open(nc_fo2).transform

```

Kod Bloğu 6: Projeksiyonların Okunması

### 2.3. Alansal İstatistik

Her iki netCDF dosyası da Burdur Havzası shapefile'inin belirlediği alanda kesilmiş, alansal istatistikleri çıkartılmış, ve bu alansal istatistikler önce pandas dataframe, daha sonrasında ise .csv dosyası olarak kaydedilmiştir.

```
# # go through all years for precipitation (sum)
pd = pd.DataFrame(columns=['sum'])
for year in years_:
    # get values of variable pear year
    nc_arr = nc_var.sel(time=year)
    nc_arr_vals = nc_arr.values
    # go through all geometries and compute zonal statistics
    for i in range(len(shp_df)):
        stat = rstats.zonal_stats(shp_df.geometry, nc_arr_vals, affine=affine, stats="sum")
        stats = stat[0]

        pd = pd.append(
            {'sum':stats['sum']}, ignore_index=True)

pd_ = pd.assign(years=yearList)
pd_['months'] = monthList
pd_.to_csv('prec_son.csv', index=False)
```

*Kod Bloğu 7: Burdur Havzası'nın Yağış İstatistiği (Toplam/Sum)*

```
## # go through all years for temperature (avg)

pd2 = pd.DataFrame(columns=['mean'])
for year in years_:
    # get values of variable pear year
    nc_arr2 = nc_var2.sel(time=year)
    nc_arr_vals2 = nc_arr2.values
    # go through all geometries and compute zonal statistics
    for i in range(len(shp_df)):
        stat = rstats.zonal_stats(shp_df.geometry, nc_arr_vals2, affine=affine2, stats="mean")
        stats = stat[0]

        pd = pd.append(
            {'sum':stats['sum']}, ignore_index=True)

pd_2 = pd2.assign(years=yearList)
pd_2['months'] = monthList
pd_2.to_csv('temp_son.csv', index=False)
```

*Kod Bloğu 8: Burdur Havzası'nın Sıcaklık İstatistiği (Ortalama/Average)*