ANGELA_Project

July 6, 2019

PRACTICA Nž1
Name : Carlos Enciso
Email : carlos.enciso.o@gmail.com
Aims:
Obtener la serie temporal de lluvia de un punto geográfico.
Presentar un mapa con la ubicación del punto seleccionado
Calcular las medidas de estadística descriptiva.
Obtener la Climatología.
Representar gráficamente.
Justificar la elección de la base de datos TRMM o PISCO para el punto escogido.
Redactar describiendo los resultados obtenidos.

[1]:
```
%%capture
#=====================================
# Importing Libraries
#=====================================
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import numpy as np
import xarray as xr
import geopandas as gpd
import cartopy.crs as ccrs
import cmocean as cmo
from rasterio import features
from affine import Affine
import mplleaflet
import os
import eofs
import string
import seaborn as sns
import cartopy.feature as cfeature
import cartopy.io.shapereader as shpreader
from cartopy.mpl.ticker import LongitudeFormatter, LatitudeFormatter
from cartopy.mpl.gridliner import LONGITUDE_FORMATTER, LATITUDE_FORMATTER
from mpl_toolkits.axes_grid1 import make_axes_locatable
import matplotlib.ticker as mticker
from cartopy.io.shapereader import Reader
```

```
from shapely.geometry.polygon import LinearRing
from pandas.tseries import converter
```

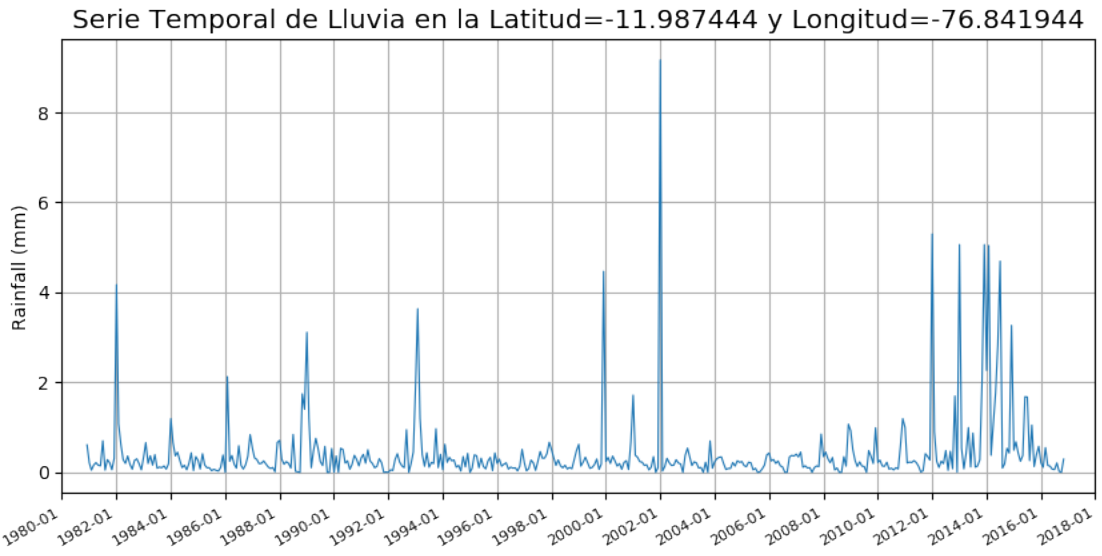
[2]:
```
#--------------------------
# Reading datasets
#--------------------------
fili = './DATASETS/new_PISCO2.nc'
ds = xr.open_dataset(fili, decode_cf=False)
ds = xr.decode_cf(ds)
```

Sol:
Obtener la serie temporal de lluvia de un punto geográfico.

[3]:
```
#------------------------
# Setting coordinates
#------------------------
lats,lons = [-11.987444, -76.841944]
ds_point = ds.sel(lat=lats,lon=lons, method='nearest')
```

[4]:
```
fig, ax = plt.subplots(figsize=(10,5), dpi=100)
plt.style.use('ggplot')
ds_series = ds_point.P.to_series()
ds_series.plot(ax=ax, linewidth=.8)
dates_rng = pd.date_range('1980-01-01', '2018-12-01', freq='24M')
plt.xticks(dates_rng, [dtz.strftime('%Y-%m') for dtz in dates_rng],␣
 ↪rotation=30, size=8)
plt.title('Serie Temporal de Lluvia en la Latitud={} y Longitud={}'.
 ↪format(lats,lons))
plt.xlabel('')
plt.ylabel('Rainfall (mm)')
```

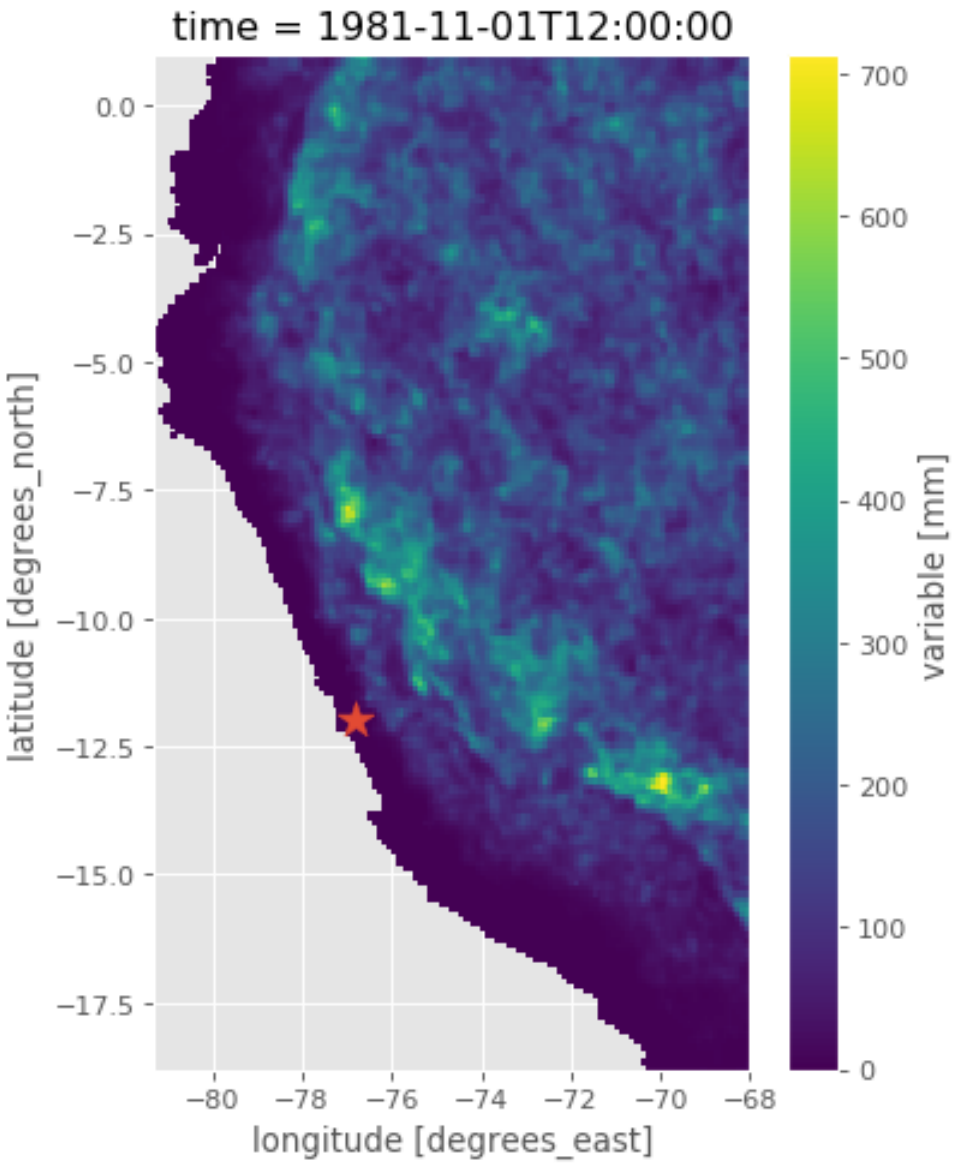
[4]:
```
Text(0, 0.5, 'Rainfall (mm)')
```



Sol:

Presentar un mapa con la ubicación del punto seleccionado.

[5]:
```
fig, ax = plt.subplots(figsize=(5,7), dpi=80)
ds.P.isel(time=10).plot(ax=ax)
ax.scatter(lons,lats, marker='*', s=200)
plt.grid(True)
```



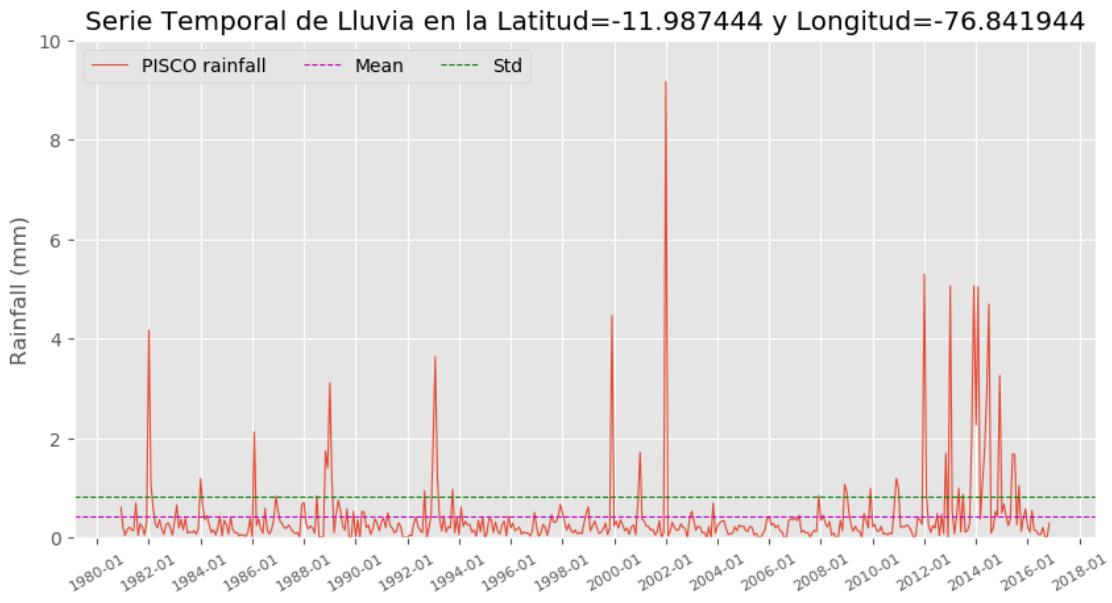
Sol:

Calcular las medidas de estadística descriptiva.

[6]:
```
[mean, std] = [ds_point.mean(), ds_point.std()]
```

[7]:
```
fig, ax = plt.subplots(figsize=(10,5), dpi=100)
plt.style.use('ggplot')
ds_series = ds_point.P.to_series()
ax.plot(ds_series, linewidth=.8, label='PISCO rainfall')
ax.axhline(y=mean.P.values, linewidth=.8, color='m', linestyle='--',␣
 ↪label='Mean')
ax.axhline(y=std.P.values, linewidth=.8, color='green', linestyle='--',␣
 ↪label='Std')
dates_rng = pd.date_range('1980-01-01', '2018-12-01', freq='24M')
plt.xticks(dates_rng, [dtz.strftime('%Y-%m') for dtz in dates_rng],␣
 ↪rotation=30, size=8)
plt.title('Serie Temporal de Lluvia en la Latitud={} y Longitud={}'.
 ↪format(lats,lons))
plt.xlabel('')
plt.ylabel('Rainfall (mm)')
ax.set_ylim(0, 10)
plt.legend(ncol=3)
```

[7]:
```
<matplotlib.legend.Legend at 0x7f7129646b70>
```

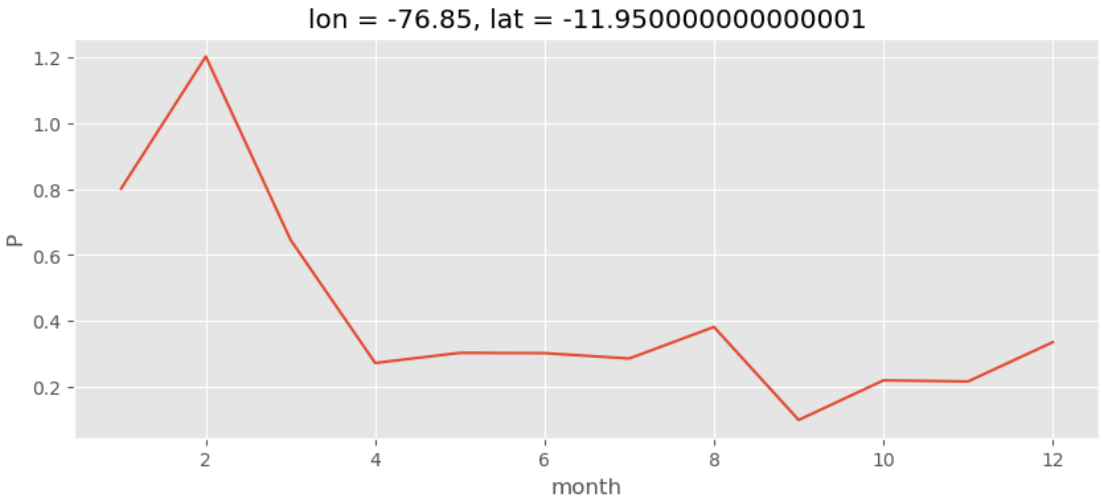


Sol:
Obtener la Climatología.

[8]:
```
ds_climatology = ds_point.groupby('time.month').mean('time')
```

[9]:
```
fig, ax = plt.subplots(figsize=(10,4), dpi=100)
ds_climatology.P.plot(ax=ax)
```

[9]: [<matplotlib.lines.Line2D at 0x7f71295f7240>]



Sol:
Justificar la elección de la base de datos TRMM o PISCO para el punto escogido.

[10]:
```
df = pd.read_pickle('./DATASETS/ÑAÑA_stations_SENAMHI')
gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.Lon, df.Lat))
```

[11]:
```
#-----------------------------------------
# Plotting point from SENAMHI database
#-----------------------------------------
def plotting_station(stations_points):
    fig, ax = plt.subplots(figsize=(8,10), dpi=60)
    stations_points.plot(ax=ax, facecolor='m')
    crs = {'init' :'epsg:32717'}
    return mplleaflet.display(fig,crs=stations_points.crs)
```

[12]:
```
plotting_station(gdf)
```

```
/home/carlos/miniconda3/envs/py37/lib/python3.6/site-
packages/IPython/core/display.py:689: UserWarning: Consider using
IPython.display.IFrame instead
  warnings.warn("Consider using IPython.display.IFrame instead")
```

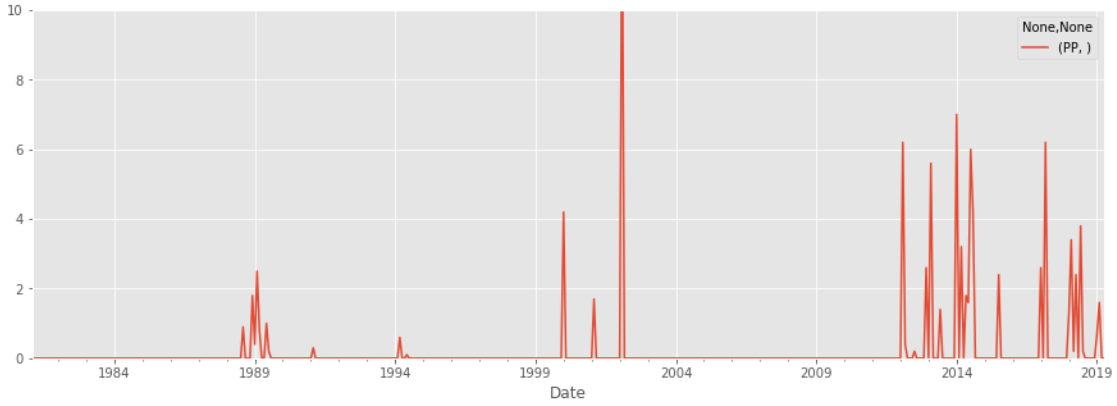
[12]: <IPython.core.display.HTML object>

[13]:
```
df_prec = df[['PP','Date']]
```

[14]:
```
df_prec_monthly = df_prec.resample('M', on='Date').sum()
```

[15]:
```
df_prec_monthly.plot(figsize=(15,5), ylim=(0,10))
```

[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7125825390>

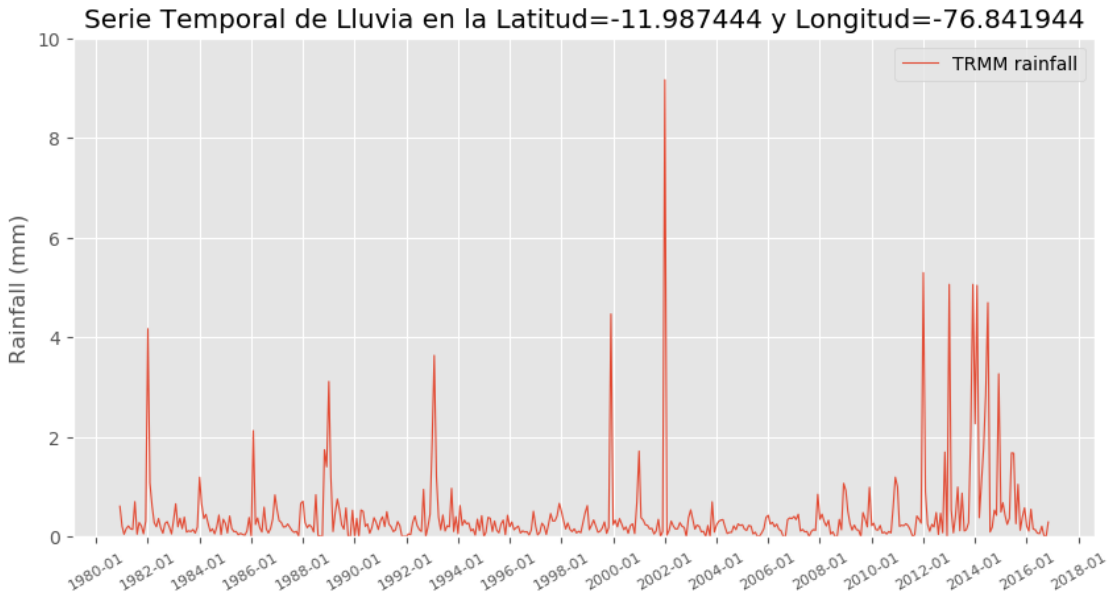


[16]:
```
ds_trmm = xr.open_dataset('./DATASETS/TRMM_3B432_2001_2014.nc', decode_cf=False)
ds_trmm = xr.decode_cf(ds_trmm)
```

[17]:
```
#------------------------
# Setting coordinates
#------------------------
ds_trmm_point = ds_trmm.sel(latitude=lats,longitude=lons, method='nearest')
```

[18]:
```
fig, ax = plt.subplots(figsize=(10,5), dpi=100)
plt.style.use('ggplot')
ds_trmm_series = ds_trmm_point.pcp.to_series()
ax.plot(ds_series, linewidth=.8, label='TRMM rainfall')
dates_rng = pd.date_range('1980-01-01', '2018-12-01', freq='24M')
plt.xticks(dates_rng, [dtz.strftime('%Y-%m') for dtz in dates_rng],␣
 ↪rotation=30, size=8)
plt.title('Serie Temporal de Lluvia en la Latitud={} y Longitud={}'.
 ↪format(lats,lons))
plt.xlabel('')
plt.ylabel('Rainfall (mm)')
ax.set_ylim(0, 10)
plt.legend(ncol=3)
```

[18]:
```
<matplotlib.legend.Legend at 0x7f7124c25c88>
```


Serie Temporal de Lluvia en la Latitud=-11.987444 y Longitud=-76.841944
TRMM rainfall
Rainfall (mm)
0
2
4
6
8
10
1980-01
1982-01
1984-01
1986-01
1988-01
1990-01
1992-01
1994-01
1996-01
1998-01
2000-01
2002-01
2004-01
2006-01
2008-01
2010-01
2012-01
2014-01
2016-01
2018-01

[ ]: