Data_hydrology_BF_Watersupply

February 28, 2023

0.0.1 Rainfall data

First part is repretition of the drought analysis but will be shown for completeness

[1]:
```
%matplotlib inline
import glob
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
from matplotlib import image as mpimg
from matplotlib.pyplot import figure
from datetime import timedelta
import scipy.stats as stats
import os

# plotting
from shapely.geometry import Polygon
import geopandas as gpd
from geospatial_functions import get_background_map
import rasterio
from rasterio.plot import show as rioshow
import folium


path = os.getcwd()
home_path = os.path.dirname(os.path.dirname(path))
gis_folder = f'{home_path}\\QGIS project'
```

[2]:
```
rainfall = pd.DataFrame()
Black_volta = pd.
 ↪read_csv(f'{home_path}\\data\\Volta_ERA5_lat_lon\ichirps_20_25_-2.75E_9.
 ↪50N_i.dat.txt', parse_dates = [0],
                delimiter = ' ', index_col=[0], skiprows=31,
                skipinitialspace = True, header = None, usecols=[0,1], names =␣
 ↪['Date','precipitation'])
```

```
Lake_Volta = pd.
 ↪read_csv(f'{home_path}\\data\\Volta_ERA5_lat_lon\ichirps_20_25_0.0E_6.5N_n.
 ↪dat.txt', parse_dates = [0],
             delimiter = ' ', index_col=[0], skiprows=31,
             skipinitialspace = True, header = None, usecols=[0,1], names =␣
 ↪['Date','precipitation'])
Mouhoun = pd.read_csv(f'{home_path}\\data\\Volta_ERA5_lat_lon\ichirps_20_25_-4.
 ↪00E_12.00N_n.dat.txt', parse_dates = [0],
          delimiter = ' ', index_col=[0], skiprows=31,
          skipinitialspace = True, header = None, usecols=[0,1], names =␣
 ↪['Date','precipitation'])
Nakambe = pd.read_csv(f'{home_path}\\data\\Volta_ERA5_lat_lon\ichirps_20_25_-2.
 ↪0E_13.5N_n.dat.txt', parse_dates = [0],
          delimiter = ' ', index_col=[0], skiprows=31,
          skipinitialspace = True, header = None, usecols=[0,1], names =␣
 ↪['Date','precipitation'])
Oti = pd.read_csv(f'{home_path}\\data\\Volta_ERA5_lat_lon\ichirps_20_25_0.0E_8.
 ↪5N_n.dat.txt', parse_dates = [0],
      delimiter = ' ', index_col=[0], skiprows=31,
      skipinitialspace = True, header = None, usecols=[0,1], names =␣
 ↪['Date','precipitation'])
Penjari = pd.read_csv(f'{home_path}\\data\\Volta_ERA5_lat_lon\ichirps_20_25_1.
 ↪0E_11.0N_n.dat.txt', parse_dates = [0],
          delimiter = ' ', index_col=[0], skiprows=31,
          skipinitialspace = True, header = None, usecols=[0,1], names =␣
 ↪['Date','precipitation'])
```

[3]:

```
names_col = ['Black_Volta', 'Lake_Volta', 'Mouhoun', 'Nakambe', 'Oti',␣
 ↪'Penjari']
Rainfall_data = pd.concat([Black_volta, Lake_Volta, Mouhoun, Nakambe, Oti,␣
 ↪Penjari], axis = 1, keys = names_col, ignore_index=False)
Rainfall_data
```

[3]:

```
             Black_Volta    Lake_Volta       Mouhoun       Nakambe  \
           precipitation precipitation precipitation precipitation
Date
1981-01-01           0.0      0.000000           0.0           0.0
1981-01-02           0.0      0.000000           0.0           0.0
1981-01-03           0.0      0.000000           0.0           0.0
1981-01-04           0.0      2.502239           0.0           0.0
1981-01-05           0.0      0.000000           0.0           0.0
…                      …             …             …             …
2022-12-27           0.0      0.000000           0.0           0.0
2022-12-28           0.0      0.000000           0.0           0.0
2022-12-29           0.0      0.000000           0.0           0.0
2022-12-30           0.0      0.000000           0.0           0.0
```

```
2022-12-31            0.0      0.000000           0.0            0.0

                      Oti       Penjari
           precipitation precipitation
Date
1981-01-01            0.0           0.0
1981-01-02            0.0           0.0
1981-01-03            0.0           0.0
1981-01-04            0.0           0.0
1981-01-05            0.0           0.0
…                   …           …
2022-12-27            0.0           0.0
2022-12-28            0.0           0.0
2022-12-29            0.0           0.0
2022-12-30            0.0           0.0
2022-12-31            0.0           0.0

[15340 rows x 6 columns]
```

[4]:
```
glob.glob(f'{home_path}\\data\\Volta_ERA5_lat_lon\*txt')
```

[4]:
```
['C:\\Users\\david\\Documents\\@@ Python\\Jaar 5\\Q3\\ENVM1400 Information and
advice\\Volta project\\Volta-burkina-
faso\\data\\Volta_ERA5_lat_lon\\ichirps_20_25_-2.0E_13.5N_n.dat.txt',
 'C:\\Users\\david\\Documents\\@@ Python\\Jaar 5\\Q3\\ENVM1400 Information and
advice\\Volta project\\Volta-burkina-
faso\\data\\Volta_ERA5_lat_lon\\ichirps_20_25_-2.75E_9.50N_i.dat.txt',
 'C:\\Users\\david\\Documents\\@@ Python\\Jaar 5\\Q3\\ENVM1400 Information and
advice\\Volta project\\Volta-burkina-
faso\\data\\Volta_ERA5_lat_lon\\ichirps_20_25_-4.00E_12.00N_n.dat.txt',
 'C:\\Users\\david\\Documents\\@@ Python\\Jaar 5\\Q3\\ENVM1400 Information and
advice\\Volta project\\Volta-burkina-
faso\\data\\Volta_ERA5_lat_lon\\ichirps_20_25_0.0E_6.5N_n.dat.txt',
 'C:\\Users\\david\\Documents\\@@ Python\\Jaar 5\\Q3\\ENVM1400 Information and
advice\\Volta project\\Volta-burkina-
faso\\data\\Volta_ERA5_lat_lon\\ichirps_20_25_0.0E_8.5N_n.dat.txt',
 'C:\\Users\\david\\Documents\\@@ Python\\Jaar 5\\Q3\\ENVM1400 Information and
advice\\Volta project\\Volta-burkina-
faso\\data\\Volta_ERA5_lat_lon\\ichirps_20_25_1.0E_11.0N_n.dat.txt']
```

[5]:
```
loc_data = []
for file in glob.glob(f'{home_path}\\data\\Volta_ERA5_lat_lon\*txt'):
    with open(file) as fin:
        for line in fin:
            process_coords = False
            if line.strip()[:15] == "# ave_region ::":
                coords = line.strip()[15:].strip()
```

```
                process_coords = True

            elif line.strip()[:22] == "# interpolating points":
                coords = line.strip()[22:].strip()
                process_coords = True

            if process_coords:
                lon, lat = coords.split(",")
                final_coord = [[float(j) for j in lon[4:].strip().split("   ")],
                               [float(k) for k in lat[5:].strip().split("   ")]]
                [xmin, xmax], [ymin, ymax] = final_coord
    loc_data.append(Polygon([(xmin, ymin), (xmax, ymin), (xmax, ymax), (xmin,␣
 ↪ymax)]))
```

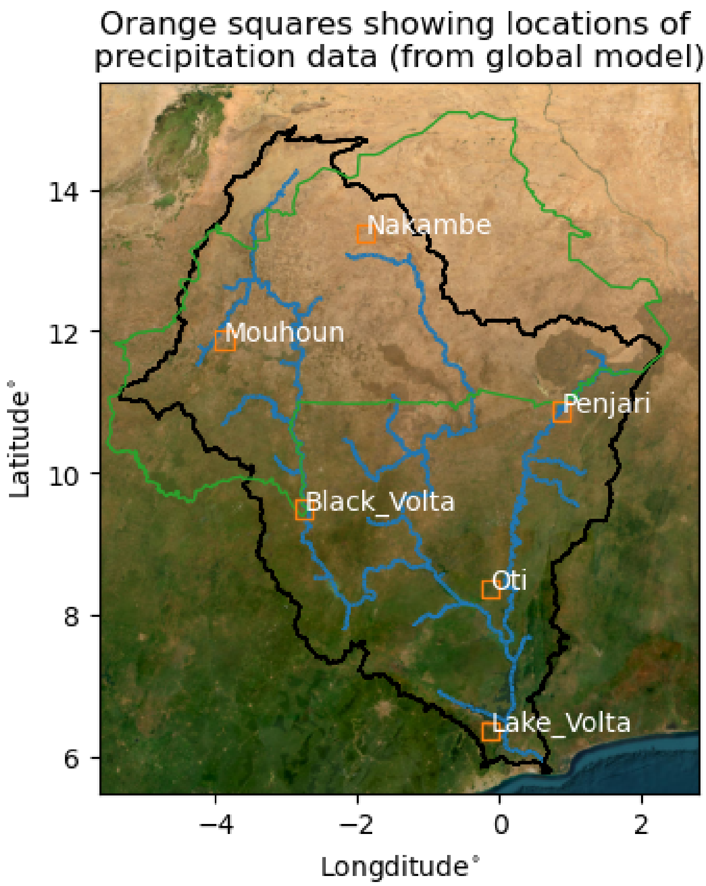
[6]:
```
names = ["Nakambe", "Black_Volta","Mouhoun","Lake_Volta","Oti","Penjari"]
geo_series_locations = gpd.GeoSeries(data=loc_data)

gdf_precip = gpd.GeoDataFrame(data=names,␣
 ↪columns=["name"],geometry=geo_series_locations)
```

[7]:
```
mid_points = gdf_precip.geometry.centroid
```

[8]:
```
country_outline = gpd.read_file(f"{gis_folder}\\country_outline_32630.gpkg")
volta_outline = gpd.read_file(f"{gis_folder}\\volta_watershed_vector_32630.
 ↪gpkg",crs="epsg:32630")
main_rivers = gpd.read_file(f"{gis_folder}\\main_rivers_volta.gpkg",crs="epsg:
 ↪32630")

country_outline = country_outline.set_geometry(country_outline.geometry.
 ↪to_crs('EPSG:4326'))
volta_outline = volta_outline.set_geometry(volta_outline.geometry.to_crs('EPSG:
 ↪4326'))
main_rivers = main_rivers.set_geometry(main_rivers.geometry.to_crs('EPSG:4326'))
```

[9]:
```
fig, ax = plt.subplots(1)
main_rivers.plot(ax=ax, color="C0",zorder=1)
# get the bounds for background
bounds_precip_measurements = (ax.get_xlim()[0], ax.get_ylim()[0], ax.
 ↪get_xlim()[1], ax.get_ylim()[1])

country_outline.plot(ax=ax, facecolor="none", edgecolor="C2",zorder=6)
volta_outline.plot(ax=ax,edgecolor="k", facecolor='none')
gdf_precip.plot(ax=ax, facecolor="none",edgecolor="C1",zorder=10)

# add labels
mid_points = gdf_precip.geometry.centroid
```

```
for index, name in enumerate(gdf_precip.name):
    ax.annotate(name, (mid_points.iloc[index].x,mid_points.iloc[index].
 ↪y),zorder=10, color="w")
# add background
with rasterio.open(get_background_map("precip_measurements",␣
 ↪bounds_precip_measurements)) as r:
    rioshow(r, ax=ax)

# crop a little
ax.set_ylim((5.5,15.5))
ax.set_title("Orange squares showing locations of \nprecipitation data (from␣
 ↪global model)")
ax.set_xlabel("Longditude$^{\circ}$");
ax.set_ylabel("Latitude$^{\circ}$");
```



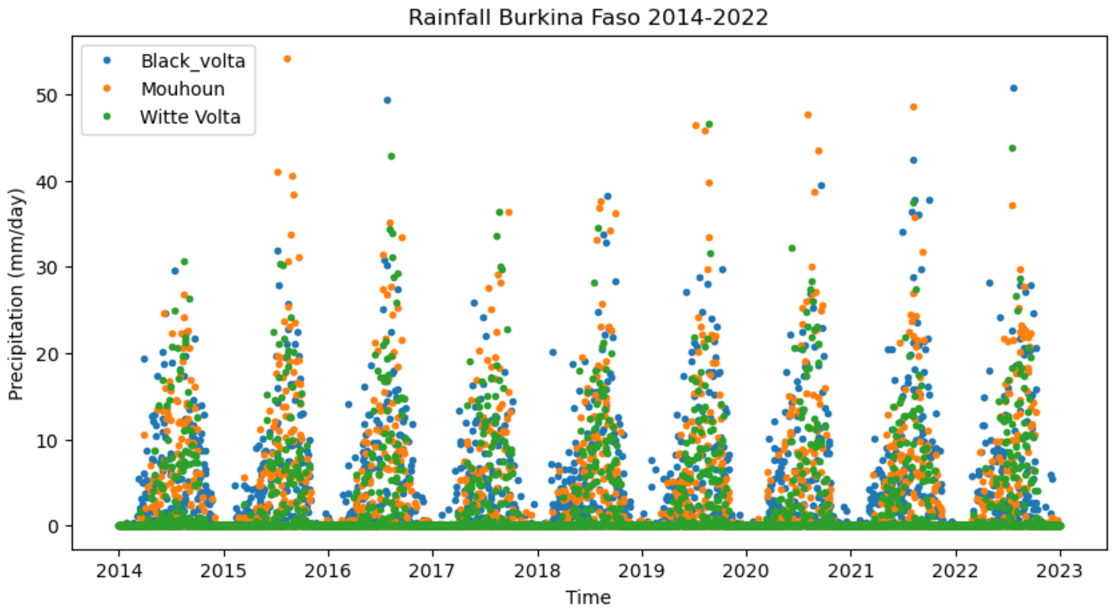

```
[10]: output = False
      if output:
```

```
    gdf_precip.to_file(f"{home_path}\\data\\Combining␣
↪data\\precipitation_data_client.geojson")
```

[11]:
```
#making dataframe smaller to 2014-2018
Rainfall_data_2014 = Rainfall_data.loc['2014-01-01':, :]
Rainfall_data_2014.loc['2017-01-01':'2022-12-31'].head()
```

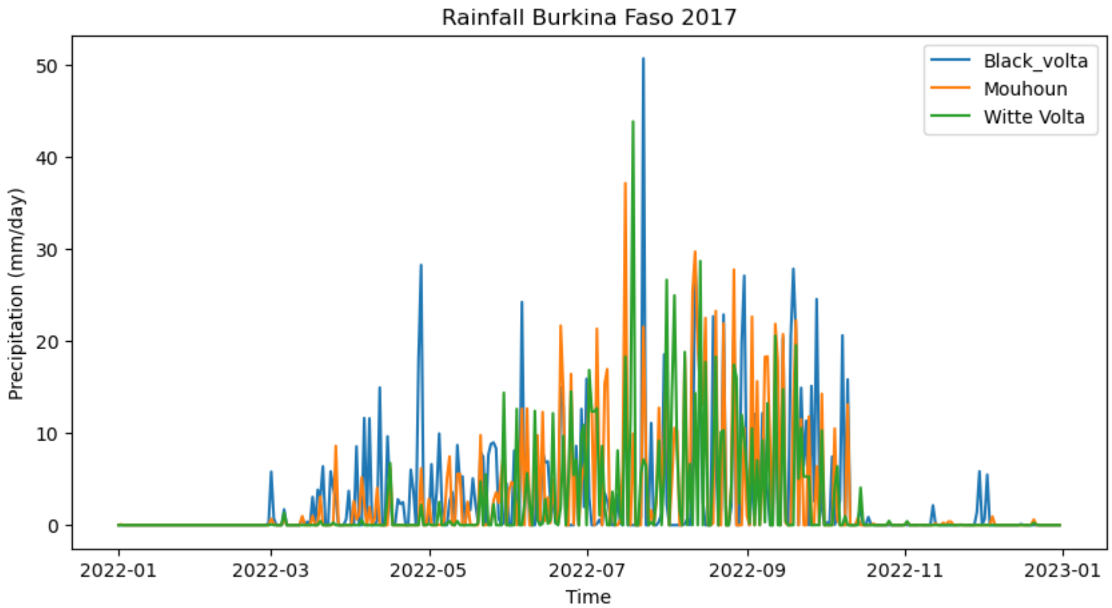
[11]:
```
             Black_Volta    Lake_Volta       Mouhoun       Nakambe  \
           precipitation precipitation precipitation precipitation
Date
2017-01-01           0.0           0.0           0.0           0.0
2017-01-02           0.0           0.0           0.0           0.0
2017-01-03           0.0           0.0           0.0           0.0
2017-01-04           0.0           0.0           0.0           0.0
2017-01-05           0.0           0.0           0.0           0.0

                     Oti       Penjari
           precipitation precipitation
Date
2017-01-01           0.0           0.0
2017-01-02           0.0           0.0
2017-01-03           0.0           0.0
2017-01-04           0.0           0.0
2017-01-05           0.0           0.0
```

[12]:
```
#plotting data from 2014 - 2022
plt.figure(figsize=(10,5))
plt.plot(Rainfall_data_2014.loc[:, 'Black_Volta'], label =␣
↪'Black_volta',marker=".",lw=0)
plt.plot(Rainfall_data_2014.loc[:, 'Mouhoun'], label = 'Mouhoun', marker='.
↪',lw=0)
plt.plot(Rainfall_data_2014.loc[:, 'Nakambe'], label = 'Witte Volta', marker='.
↪',lw=0)
plt.xlabel('Time')
plt.ylabel('Precipitation (mm/day)')
plt.title('Rainfall Burkina Faso 2014-2022');
plt.legend();
```



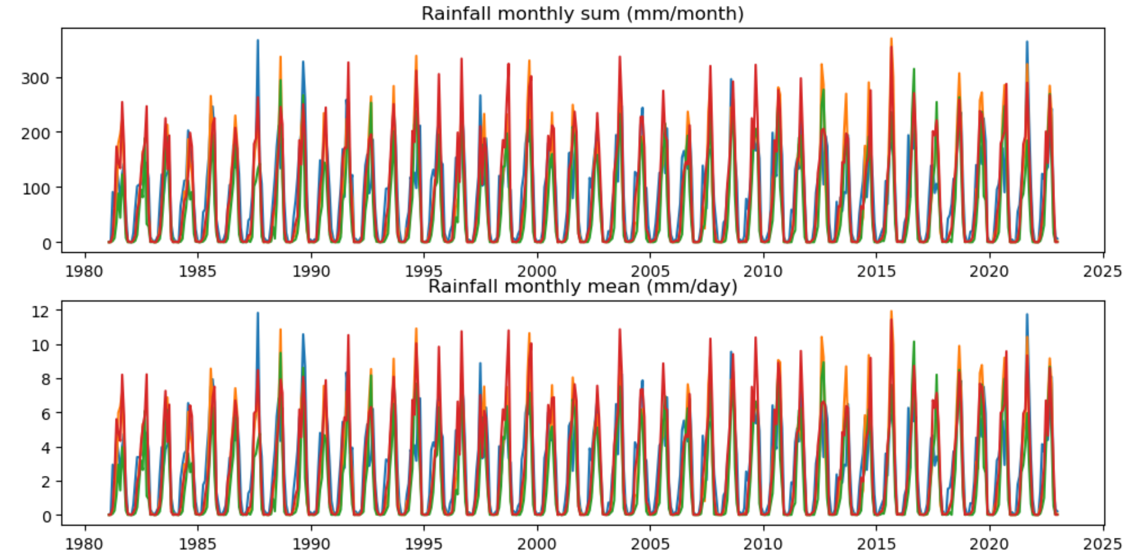
Notes: - Rainfall has become more extreme - high seasonality: dry and wet season - High peaks and low lows - Maxima are around 50 to 60 mm/day

[13]:
```
#plotting data in 2017
plt.figure(figsize=(10,5))
plt.plot(Rainfall_data_2014.loc['2022-01-01':'2022-12-31', 'Black_Volta'],␣
 ↪label = 'Black_volta')
plt.plot(Rainfall_data_2014.loc['2022-01-01':'2022-12-31', 'Mouhoun'], label =␣
 ↪'Mouhoun')
plt.plot(Rainfall_data_2014.loc['2022-01-01':'2022-12-31', 'Nakambe'], label =␣
 ↪'Witte Volta')
plt.title('Rainfall Burkina Faso 2017')
plt.xlabel('Time')
plt.ylabel('Precipitation (mm/day)')
plt.legend();
```



Notes: - Very dry season from october untill may
- Wet season from may to october

[35]:
```
# sorting to only B.F
Rainfall_sorted_BF = Rainfall_data.loc[:,['Black_Volta','Mouhoun', 'Nakambe',␣
↪'Penjari']]
Rainfall_BF_msum = Rainfall_sorted_BF.resample('M').sum()
Rainfall_BF_mmean = Rainfall_sorted_BF.resample('M').mean()

#plotting monthly sum and mean
fig, ax = plt.subplots(2,1,figsize= (10,5))
fig.tight_layout(h_pad=1)
ax[0].set_title('Rainfall monthly sum (mm/month)')
ax[0].plot(Rainfall_BF_msum)
ax[1].set_title('Rainfall monthly mean (mm/day)')
ax[1].plot(Rainfall_BF_mmean);
```



[37]:
```
Rainfall_BF_msum.columns = Rainfall_BF_msum.columns.droplevel(level=1)
```

[46]:
```
output = False
if output:
    Rainfall_BF_msum.to_excel(f"{home_path}\\data\\Combining␣
 ↪data\\Monthly_sum_rainfall.xlsx")
```

[32]:
```
Rainfall_BF_msum["month"] = Rainfall_BF_msum.apply(lambda x: x.name.month,␣
 ↪axis=1)
```

Growing season in Burkina Faso is between May and November

[17]:
```
mask = ((Rainfall_BF_msum.month >= 5) & (Rainfall_BF_msum.month <= 11))
Rainfall_BF_msum_growing_season = Rainfall_BF_msum[mask]
Rainfall_BF_msum_growing_season
```

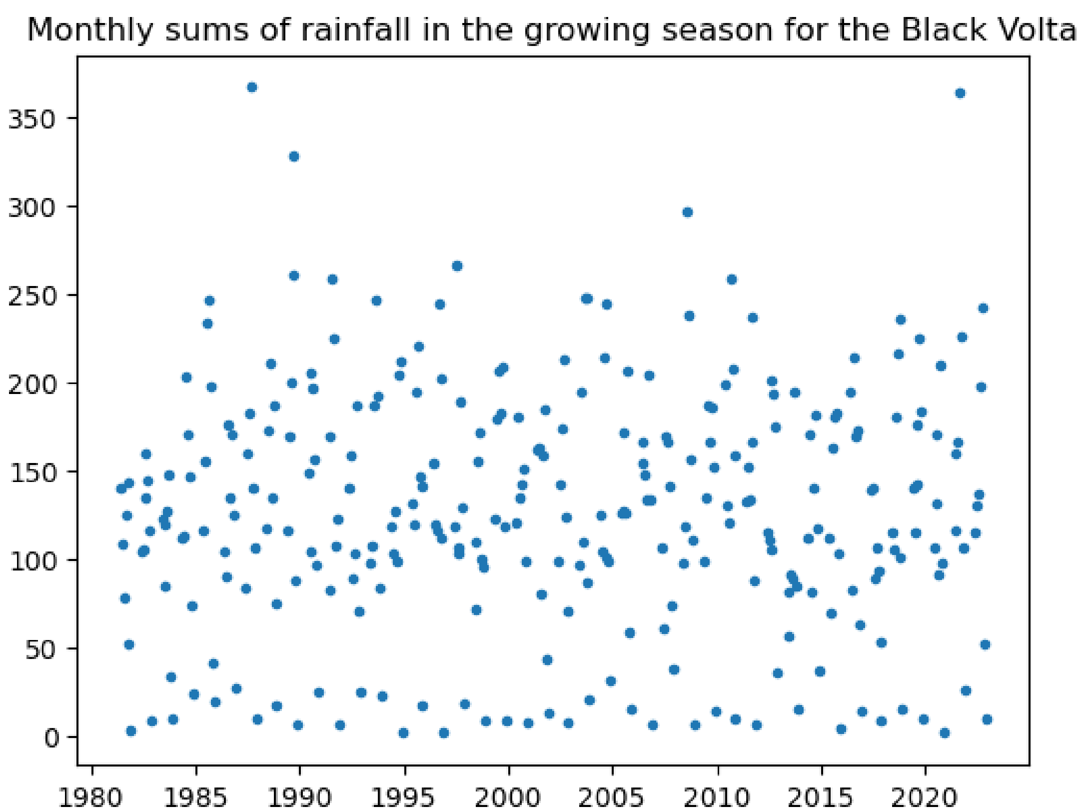
[17]:

	Black_Volta precipitation	Mouhoun precipitation	Nakambe precipitation	Penjari precipitation	month
Date					
1981-05-31	140.261632	77.489709	35.860808	173.526988	5
1981-06-30	107.978451	179.285071	88.180126	143.640896	6
1981-07-31	78.178869	196.273572	44.308429	134.048451	7
1981-08-31	124.400695	223.085272	122.409677	254.216096	8
1981-09-30	142.819855	129.306107	74.593623	174.523889	9
…	…	…	…	…	…
2022-07-31	136.009999	209.163020	189.988094	143.766882	7
2022-08-31	197.515479	283.810280	268.714974	266.400101	8
2022-09-30	242.311991	229.312312	143.299004	180.332295	9
2022-10-31	51.457809	25.429949	16.014090	84.981922	10

```
2022-11-30      9.665133      1.552652      0.417778      0.000000    11

[294 rows x 5 columns]
```

[18]:
```
fig, ax = plt.subplots(1)
ax.plot(Rainfall_BF_msum_growing_season.
 ↪index,Rainfall_BF_msum_growing_season["Black_Volta"],marker=".",lw=0)
ax.set_title("Monthly sums of rainfall in the growing season for the Black␣
 ↪Volta")
```

[18]:
```
Text(0.5, 1.0, 'Monthly sums of rainfall in the growing season for the Black
Volta')
```



1 Yearly

[19]:
```
# resample yearly
Rainfall_BF_ysum = Rainfall_sorted_BF.resample('Y').sum()
Rainfall_BF_ymean = Rainfall_sorted_BF.resample('Y').mean()
Rainfall_BF_ysum.head(5)
```

[19]:

	Black_Volta precipitation	Mouhoun precipitation	Nakambe precipitation	Penjari precipitation
Date				
1981-12-31	806.372730	847.056155	382.767435	992.835201
1982-12-31	956.491056	728.852786	499.778791	917.659482
1983-12-31	732.098916	723.026033	579.453776	808.718699
1984-12-31	997.025082	701.638042	407.425785	782.821328
1985-12-31	1124.828382	823.212610	511.587376	817.543045

Notes: - rainfall sum is extremely high - monthly mean is also high, but seems less extreme

1.0.1 Analysis of water suply from precipitation

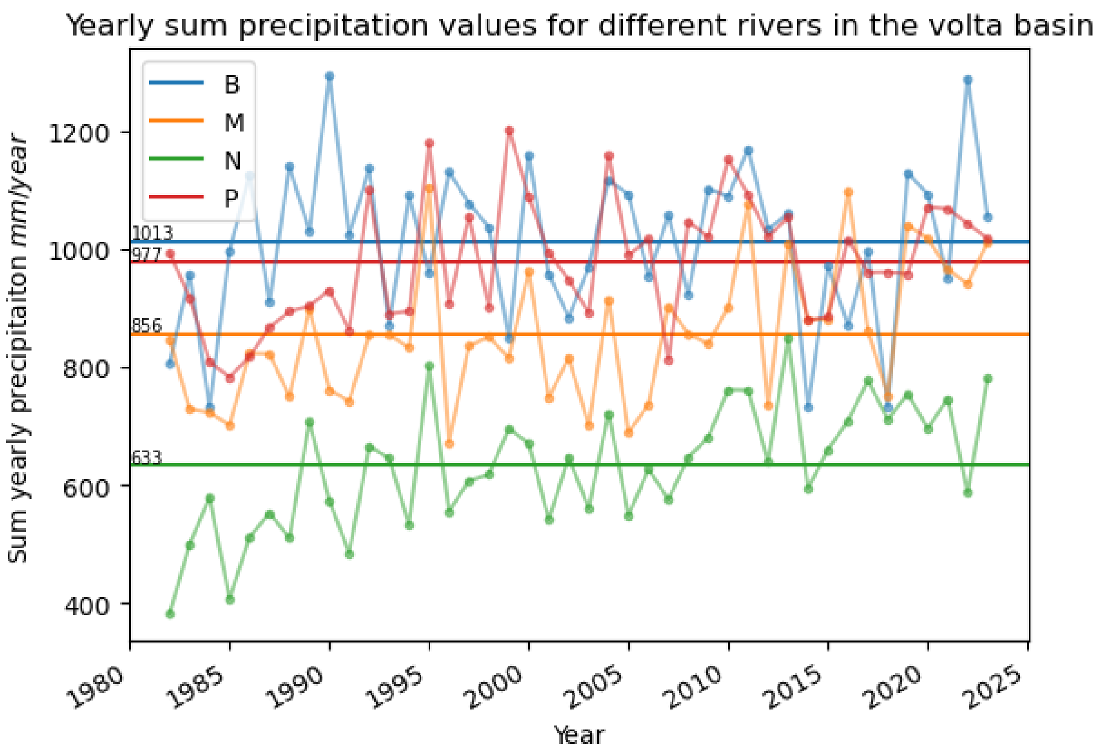
[20]:
```
df_yearly_means = pd.DataFrame(data=Rainfall_BF_ysum.mean(),columns=["P"])
df_yearly_means.index = df_yearly_means.apply(lambda x: x.name[0], axis=1)
df_yearly_means
```

[20]:

	P
Black_Volta	1013.079354
Mouhoun	856.154086
Nakambe	632.943870
Penjari	977.379929

[ ]:
```
```

[21]:
```
fig, ax = plt.subplots(1)
for index, row in enumerate(df_yearly_means.iterrows()):
    ax.axhline(row[1].P,label=row[0][0],color=f"C{index}")
    ax.annotate(f'{row[1].P:.0f}',(3660,row[1].P+5),fontsize="x-small")
ax.legend()
Rainfall_BF_ysum.plot(ax=ax,legend=False,alpha=0.5,marker='.')

ax.set_ylabel("Sum yearly precipitaiton $mm/year$")
ax.set_xlabel("Year")
ax.set_title("Yearly sum precipitation values for different rivers in the volta␣
↪basin");
```



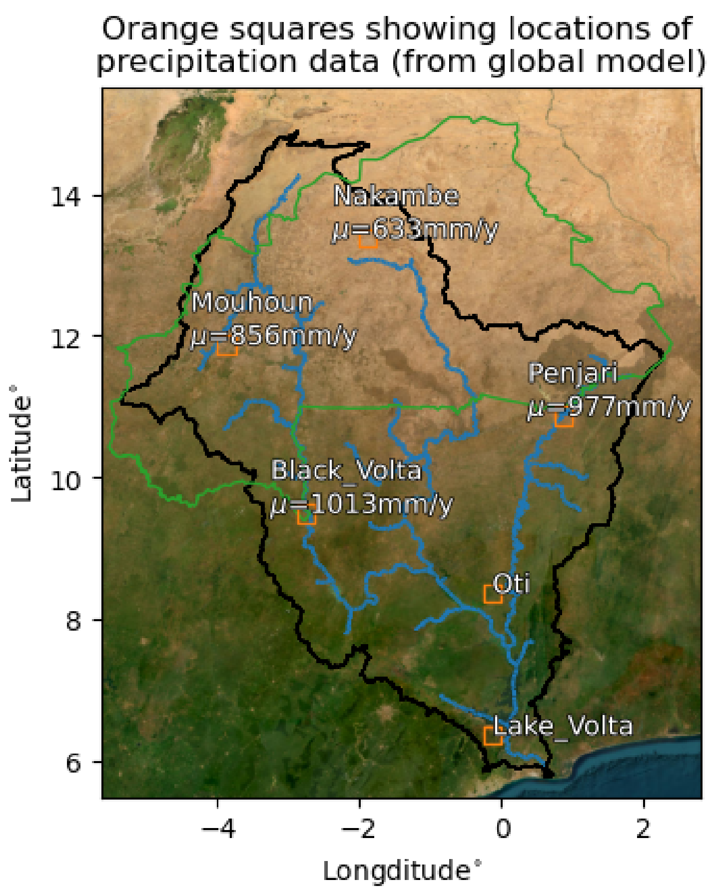
[22]:
```
df_yearly_means
```

[22]:

	P
Black_Volta	1013.079354
Mouhoun	856.154086
Nakambe	632.943870
Penjari	977.379929

[23]:
```
fig, ax = plt.subplots(1)
main_rivers.plot(ax=ax, color="C0",zorder=1)
# get the bounds for background
bounds_precip_measurements = (ax.get_xlim()[0], ax.get_ylim()[0], ax.
 ↪get_xlim()[1], ax.get_ylim()[1])

country_outline.plot(ax=ax, facecolor="none", edgecolor="C2",zorder=6)
volta_outline.plot(ax=ax,edgecolor="k", facecolor='none')
gdf_precip.plot(ax=ax, facecolor="none",edgecolor="C1",zorder=10)

# add labels
mid_points = gdf_precip.geometry.centroid
for index, name in enumerate(gdf_precip.name):
    if name in df_yearly_means.index:
```

```python
        ax.annotate(f"{name} \n$\mu$={df_yearly_means.loc[name,'P']:.0f}mm/y" ,
                    (mid_points.iloc[index].x-0.5,mid_points.iloc[index].y),zorder=10, color="w",
                     path_effects=[matplotlib.patheffects.withStroke(linewidth=1, foreground="k")])
    else:
        ax.annotate(name, (mid_points.iloc[index].x,mid_points.iloc[index].y),zorder=10, color="w",
                     path_effects=[matplotlib.patheffects.withStroke(linewidth=1, foreground="k")])
# add background
with rasterio.open(get_background_map("precip_measurements", bounds_precip_measurements)) as r:
    rioshow(r, ax=ax)

# crop a little
ax.set_ylim((5.5,15.5))
ax.set_title("Orange squares showing locations of \nprecipitation data (from global model)")
ax.set_xlabel("Longditude$^{\circ}$");
ax.set_ylabel("Latitude$^{\circ}$");
```



[ ]:

```
```