Code to Generate Input Data from Weather Dataset

https://colab.research.google.com/drive/1bswqCdNeJrgfCVVBSD-AqI-dg_OIMhLb?usp=sharing

```
import pandas as pd
import numpy as np
from math import radians, cos, sin, asin, sqrt
import requests
from io import StringIO


# Important library for many geopython libraries
!apt install gdal-bin python-gdal python3-gdal
# Install rtree - Geopandas requirment
!apt install python3-rtree
# Install Geopandas
!pip install git+git://github.com/geopandas/geopandas.git
# Install descartes - Geopandas requirment
!pip install descartes
# Install Folium for Geographic data visualization
!pip install folium
# Install plotlyExpress
!pip install plotly_express

# ca bioregions shape file
import geopandas as gpd
!unzip /content/data.zip
```

Retrieve Raw Weather data from NOAA

```
x = requests.get('https://nathanpersonalbucket.s3-us-west-2.amazonaws.com/2366872.csv')
data = StringIO(x.text)
weatherStat = pd.read_csv(data)
weatherStat.NAME.unique().size #73 stations
weatherStat = weatherStat.drop_duplicates(subset=['NAME'])
# Renaming the column names
weatherStat=weatherStat.rename(columns = {'LATITUDE':'lat','LONGITUDE':'lon'})
weatherStat
```

	STATION	NAME	lat	lon	ELEVATION	DATE	AWND	PRCP
0	USW00023129	LONG BEACH DAUGHERTY AIRPORT, CA US	33.81160	-118.14630	9.4	1970-01-01	NaN	0.00
18263	USW00023293	SAN JOSE INTERNATIONAL AIRPORT, CA US	37.35910	-121.92400	15.5	1998-07-04	8.50	0.00
26113	USW00003167	HAWTHORNE MUNICIPAL AIRPORT, CA US	33.92278	-118.33417	19.2	1998-04-01	5.82	0.07
33866	USW00023174	LOS ANGELES INTERNATIONAL AIRPORT, CA US	33.93800	-118.38880	29.6	1970-01-01	NaN	0.00
52129	USW00023130	VAN NUYS	34.20972	-118.48917	234.7	1998-	4.70	0.11

Useful Functions

		SANTA MONICA				1998-		

```
def dist(lat1, long1, lat2, long2):
    # calc distance between 2 points on earth - using haversine formula
    # convert decimal degrees to radians
    lat1, long1, lat2, long2 = map(radians, [lat1, long1, lat2, long2])
    # haversine formula
    dlon = long2 - long1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * asin(sqrt(a))
    km = 6371* c # Radius of earth in kilometers is 6371
    return km

def find_nearest(lat, long):
  #given lat, lon of fire - find nearest weather station
    distances = weatherStat.apply(
        lambda row: dist(lat, long, row['lat'], row['lon']), axis=1)
    return weatherStat.loc[distances.idxmin(), 'NAME']
```

Create 3 Bioregions

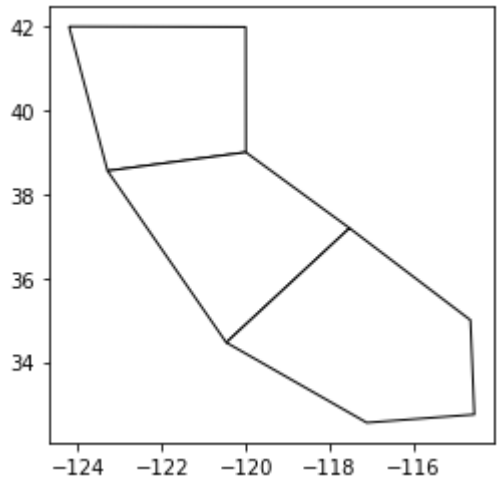
```
import geopandas as gpd
from shapely.geometry import Polygon
import matplotlib.pyplot as plt

poly1_lons = [41.996221, 41.985584,38.99915, 38.556104]
poly1_lats = [-124.216027, -120.003800, -120.00038, -123.298092]
poly2_lons = [38.556104, 38.99915, 37.190380, 34.463602]
```

```
poly2_lats = [-123.298092, -120.00038, -117.528050, -120.46687]
reg1 = Polygon(zip(poly1_lats, poly1_lons))
reg2 = Polygon(zip(poly2_lats, poly2_lons))
reg3 = Polygon([(-117.528050,37.190380), (-120.46687,34.463602), (-117.116231,32.552450), (-1

df_bioregions = gpd.GeoDataFrame(geometry=[reg1, reg2,reg3])
df_bioregions['region'] = [1,2,3]
fig, ax = plt.subplots()
ax.set_aspect('equal')
df_bioregions.plot(ax=ax, color='white', edgecolor='black')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5c3b3ca898>
```



Get Monthly Averages from Daily Temperature Data from NOAA Weather Dataset

```
x = requests.get('https://nathanpersonalbucket.s3-us-west-2.amazonaws.com/2366872.csv')
data = StringIO(x.text)

df = pd.read_csv(data)
table = df.dropna(subset=['TAVG'])
table = table.drop(columns=['AWND'])
table['DATE'] = pd.to_datetime(table['DATE'])

table['YEAR']=table['DATE'].dt.year
table['MONTH']=table['DATE'].dt.month
table = table.groupby(['NAME','YEAR','MONTH']).mean()


table
gdf_weather = gpd.GeoDataFrame(table,   geometry=gpd.points_from_xy(table.LONGITUDE, table.LA
table
```

NAME	YEAR	MONTH	LATITUDE	LONGITUDE	ELEVATION	PRCP	TAVG	TM
ALTURAS MUNICIPAL AIRPORT, CA US	1998	6	41.49139	-120.56444	1333.5	0.068333	57.500000	71.5333
		7	41.49139	-120.56444	1333.5	0.045484	68.290323	88.1290
		9	41.49139	-120.56444	1333.5	0.066333	60.733333	78.2333
		10	41.49139	-120.56444	1333.5	0.010000	45.533333	61.7000
		11	41.49139	-120.56444	1333.5	0.119667	36.033333	45.0000

Spatial Join Bioregions with Weather Data

```
joinWDF=gpd.sjoin(df_bioregions, gdf_weather, how='inner', op='intersects')
WEATHER_REGIONS = joinWDF.drop(columns=['index_right','LATITUDE','LONGITUDE','ELEVATION'])
WEATHER_REGIONS
```

	geometry	region	NAME	YEAR	MONTH	PRCP	TAVG	TMAX	TMI
0	POLYGON ((-124.21603 41.99622, -120.00380 41.9...	1	MONTAGUE SISKIYOU AIRPORT, CA US	2002	6	0.001154	64.423077	82.038462	46.30769
0	POLYGON ((-124.21603 41.99622, -120.00380 41.9...	1	MONTAGUE SISKIYOU AIRPORT, CA US	2005	2	0.016429	39.714286	53.535714	25.42857
0	POLYGON ((-124.21603 41.99622, -120.00380 41.9...	1	MONTAGUE SISKIYOU AIRPORT, CA US	2003	2	0.026429	38.821429	50.678571	26.53571
0	POLYGON ((-124.21603 41.99622, -120.00380	1	MONTAGUE SISKIYOU AIRPORT,	2002	2	0.035714	40.821429	53.714286	27.42857

Get Weather Averages from the Past 3 Months, then Save Results to CSV

```
WEATHER_3MONTH = WEATHER_REGIONS.groupby(by=['region','YEAR','MONTH']).mean().reset_index()
WEATHER_3MONTH

PRCP3 = []
TAVG3 = []
TMIN3 = []
TMAX3 = []
def threeMonthPRCP(data):
  for i in range(3, len(data)):
    threeMON = data.iloc[i-3]
    twoMON = data.iloc[i-2]
    oneMON = data.iloc[i-1]
    df = pd.DataFrame([threeMON, twoMON, oneMON])
    PRCP3.append(df['PRCP'].mean())
    TAVG3.append(df['TAVG'].mean())
    TMIN3.append(df['TMIN'].mean())
    TMAX3.append(df['TMAX'].mean())

threeMonthPRCP(WEATHER_3MONTH)
WEATHER_3MONTH = WEATHER_3MONTH.drop([0,1,2])
WEATHER_3MONTH['PRCP3'] = PRCP3
WEATHER_3MONTH['TAVG3'] = TAVG3
WEATHER_3MONTH['TMIN3'] = TMIN3
WEATHER_3MONTH['TMAX3'] = TMAX3
WEATHER_3MONTH.to_csv('weather3MONTH.csv', index=False)
```