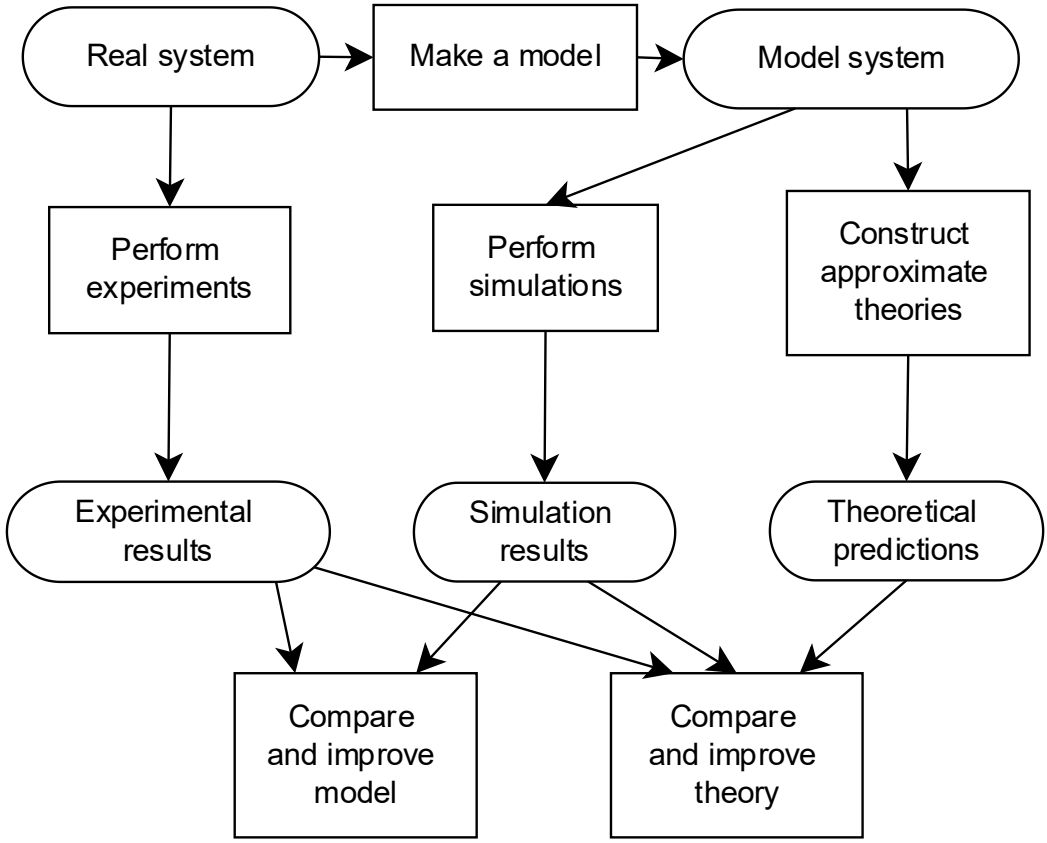
Geospatial Data Science
Content Block II: *Techniques*
Lab 7
Sampling Rasters & MultiD Arrays

Austin J. Brockmeier, Ph.D.

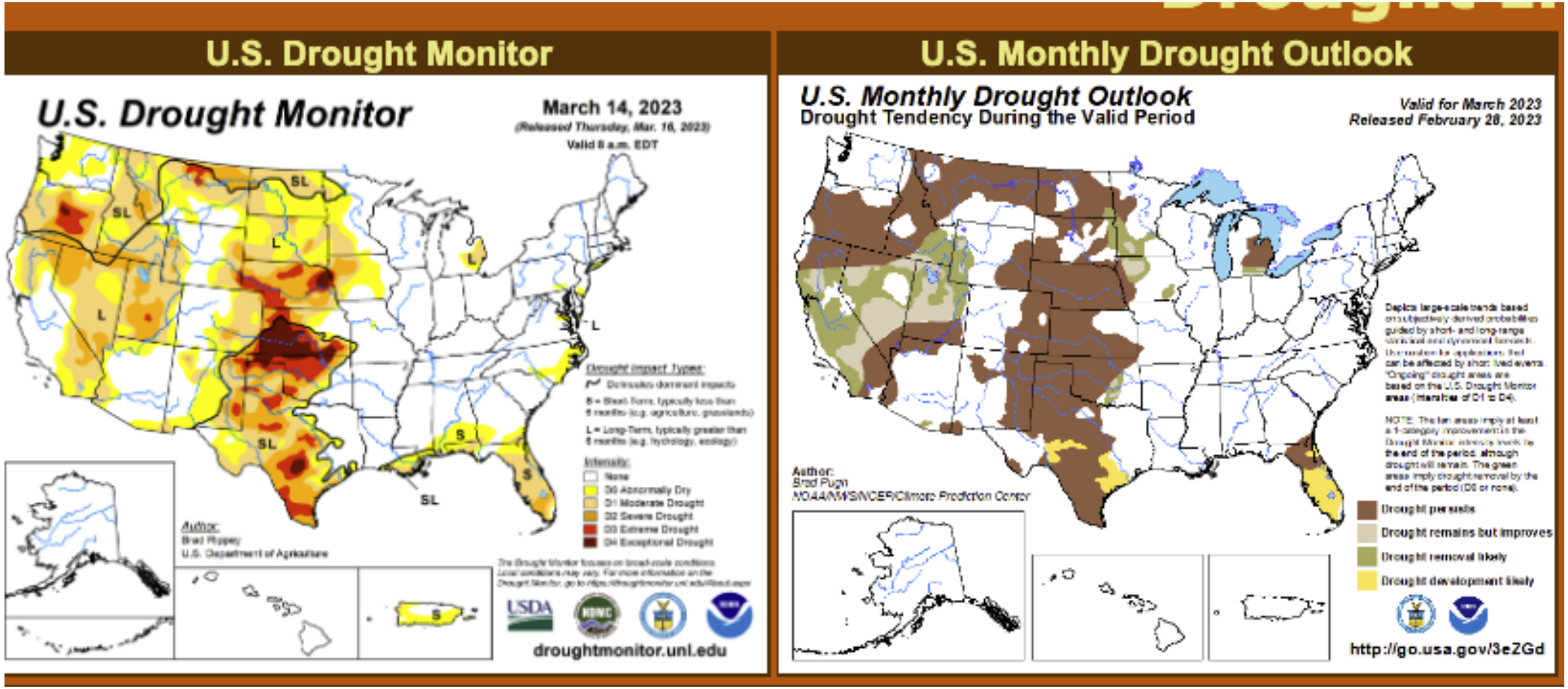
Wednesday, March 23rd , 2023

Probability and Spatial Statistics for Geospatial Systems and Data

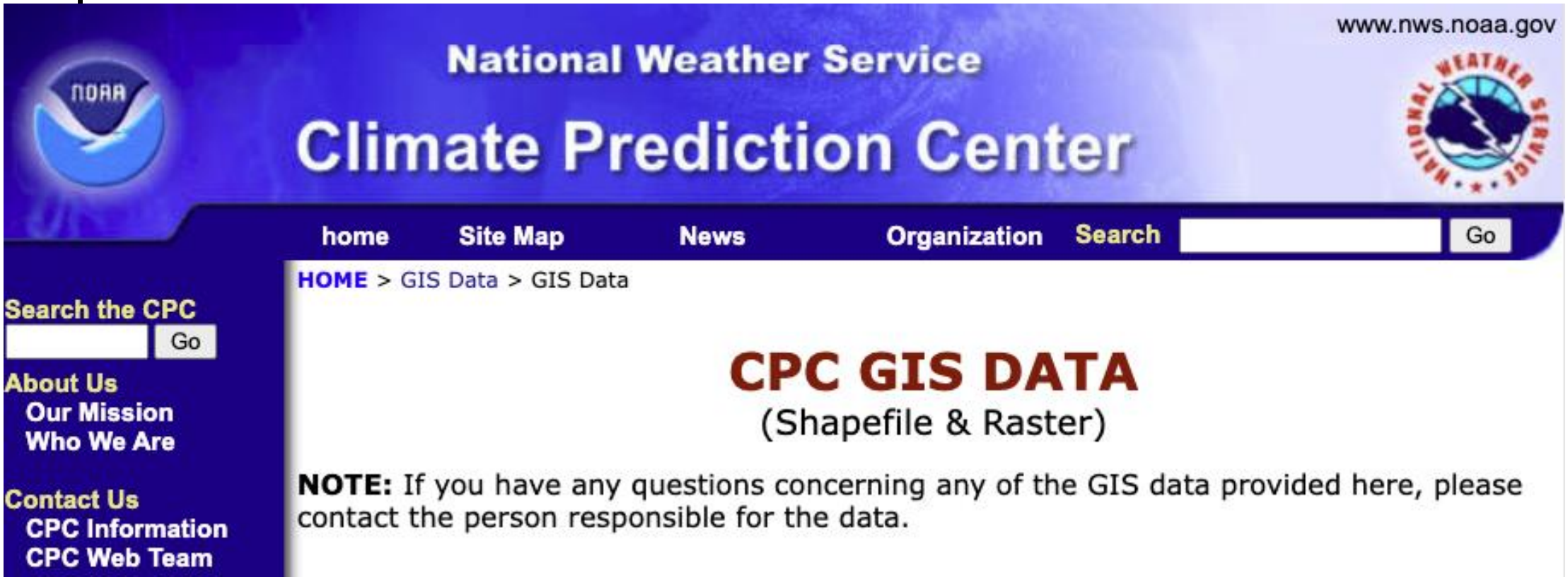
1. Statistical analysis
2. Generate data from random variables and processes through **computer simulation** (Lab 6)
3. **Analyze spatial patterns in data**



Monitoring (current) compared to outlook (future prediction from model)



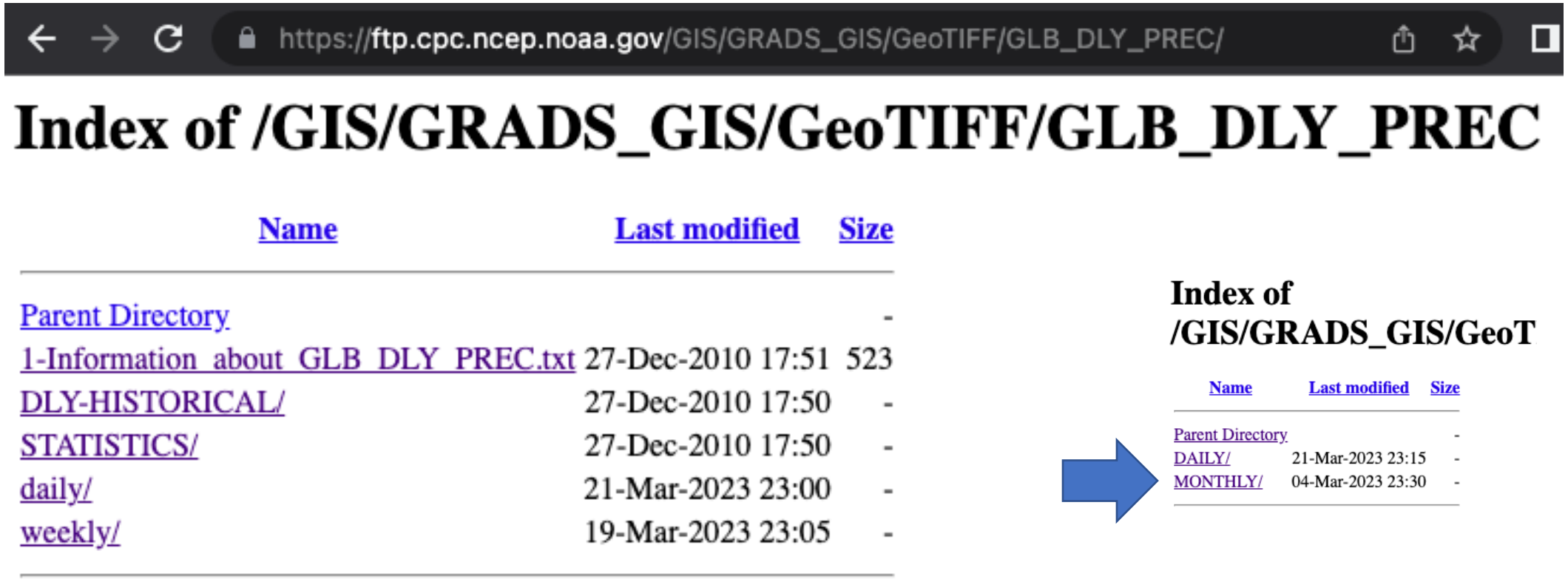
Open data from NWS NOAA



Daily Gridded Precipitation Analysis Contact: Wei.Shi@noaa.gov	
CPC Unified Global Gauge Daily Precipitation Analysis	Metadata

https://www.cpc.ncep.noaa.gov/products/GIS/GIS_DATA/

`ftp` is File transfer protocol meant to access data remotely



https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/

Climate, anomalies, total

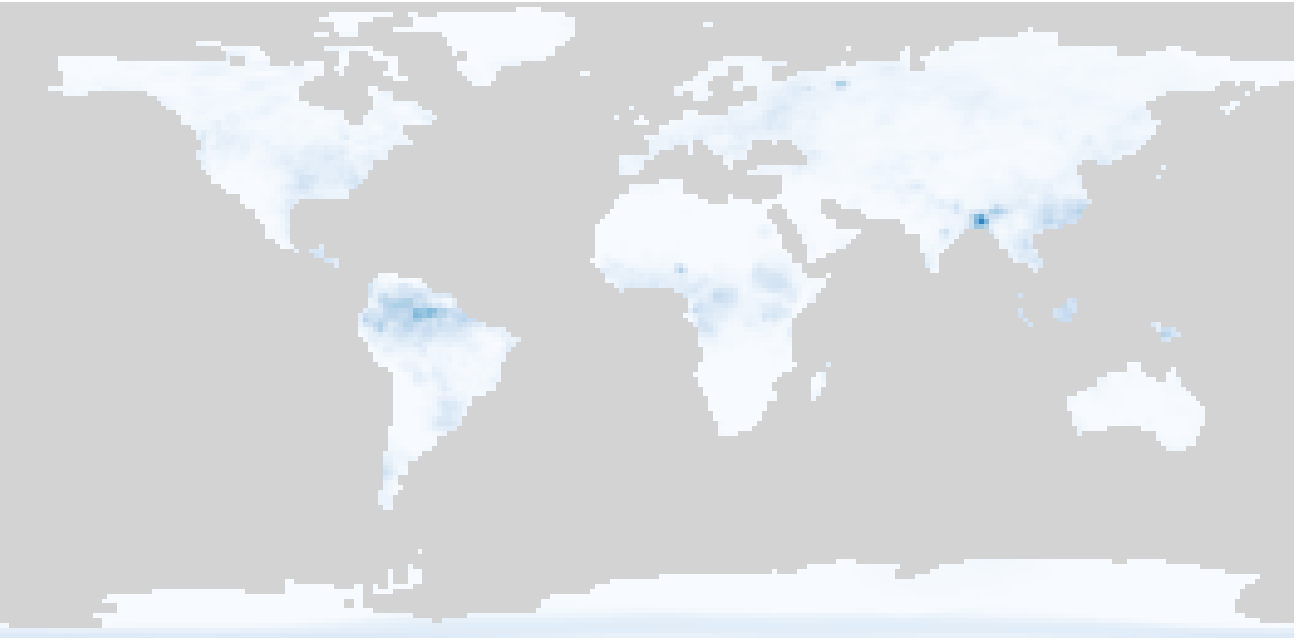
Name	Last modified	Size
Parent Directory		-
p_glb_anom_201201_float.tif	05-Feb-2012 00:30	1.0M
p_glb_anom_201202_float.tif	05-Mar-2012 00:30	1.0M
p_glb_anom_201203_float.tif	04-Apr-2012 23:30	1.0M
p_glb_anom_201204_float.tif	04-May-2012 23:30	1.0M
p_glb_anom_201205_float.tif	04-Jun-2012 23:30	1.0M

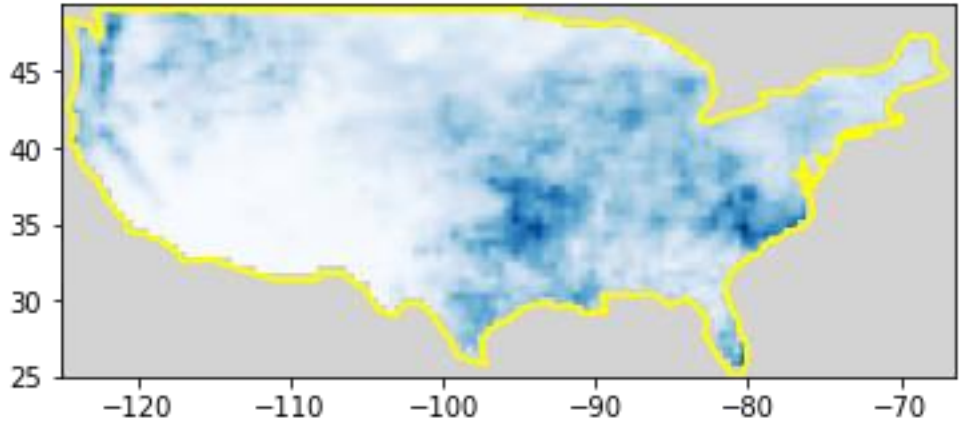
p_glb_pclim_201905_float.tif	04-Jun-2019 23:30	1.0M
p_glb_pclim_201906_float.tif	04-Jul-2019 23:30	1.0M
p_glb_pclim_201907_float.tif	04-Aug-2019 23:30	1.0M
p_glb_pclim_201908_float.tif	04-Sep-2019 23:30	1.0M
p_glb_pclim_201909_float.tif	04-Oct-2019 23:30	1.0M

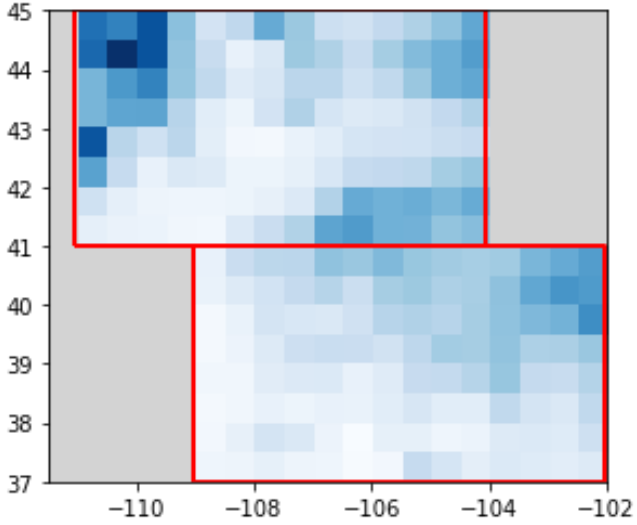
p_glb_clim_201712_float.tif	04-Jan-2018 23:30	1.0M
p_glb_clim_201801_float.tif	04-Feb-2018 23:30	1.0M
p_glb_clim_201802_float.tif	04-Mar-2018 23:30	1.0M
p_glb_clim_201803_float.tif	04-Apr-2018 23:30	1.0M
p_glb_clim_201804_float.tif	04-May-2018 23:30	1.0M

p_glb_total_201501_float.tif	04-Feb-2015 23:30	1.0M
p_glb_total_201502_float.tif	04-Mar-2015 23:30	1.0M
p_glb_total_201503_float.tif	04-Apr-2015 23:30	1.0M
p_glb_total_201504_float.tif	04-May-2015 23:30	1.0M

Raster windows/subsets for focused analysis







Rasters are arrays

```
no_data_pixels=np.sum(raster1.read() ==-999)
```

```
raster1_sub, window_transform = rasterio.mask.mask(raster1, s_states, crop=True)
rasterio.plot.show(raster1_sub,transform=window_transform)
```

Downloading in the face of 404s with try/except

```
rasters = []
for i in range(2012,2023+1):
  for j in range(1,12+1):
    file_name = url_str+file_strs[0]+"{}{:02}".format(i,j)+file_strs[1]
    print(file_name)
    try:
       with rasterio.open(file_name, 'r') as raster:
         raster_sub = rasterio.mask.mask(raster, s_states, crop=True,nodata=np.nan)
         rasters.append(raster_sub[0])
    except:
       rasters.append(np.full(raster1_sub.shape, np.nan))
```

```
https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/STATISTICS/MONTHLY/p_glb_total_201706_float.tif
https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/STATISTICS/MONTHLY/p_glb_total_201707_float.tif
https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/STATISTICS/MONTHLY/p_glb_total_201708_float.tif
https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/STATISTICS/MONTHLY/p_glb_total_201709_float.tif
```

Making a stack of matrices from a list

```
rasters = []
for i in range(2012,2023+1):
  for j in range(1,12+1):
        …
       rasters.append(raster_sub[0])
    except:
       rasters.append(np.full(raster1_sub.shape, np.nan))
```

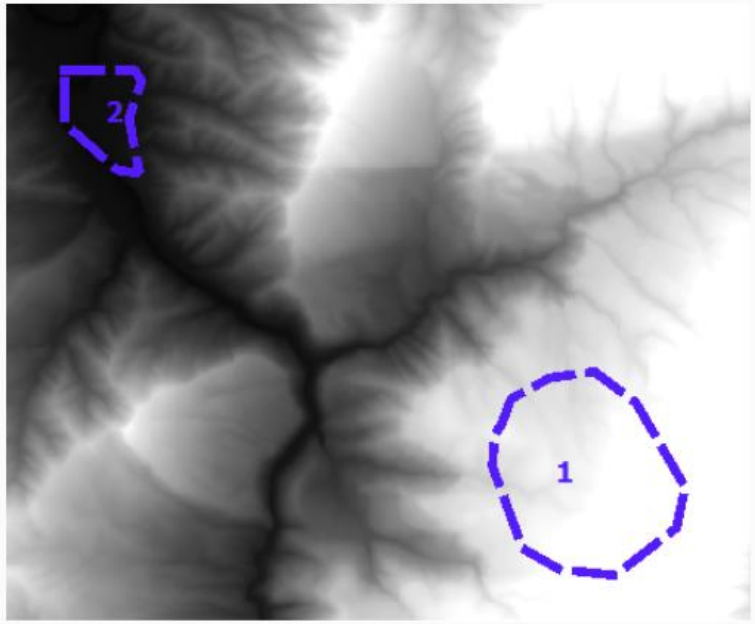
https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/STATISTICS/MONTHLY/p_glb_total_201706_float.tif
https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/STATISTICS/MONTHLY/p_glb_total_201707_float.tif
https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/STATISTICS/MONTHLY/p_glb_total_201708_float.tif
https://ftp.cpc.ncep.noaa.gov/GIS/GRADS_GIS/GeoTIFF/GLB_DLY_PREC/STATISTICS/MONTHLY/p_glb_total_201709_float.tif

```
X = np.vstack(rasters)
print(X.shape)
M = np.nanmean(X,axis=0)*12 # average annual rainfall ... assuming missing at random
M[np.isnan(X).all(axis=0)] = np.nan
```

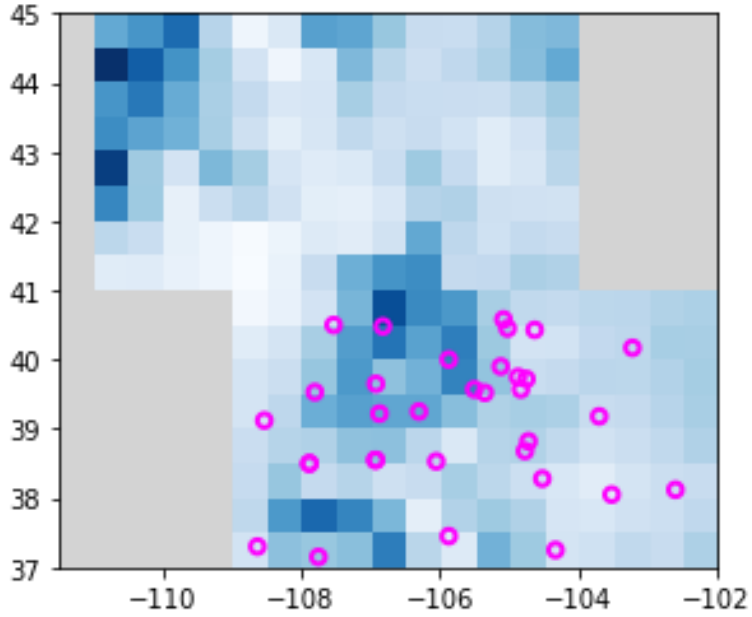
Computing the mean across all months

Get raster data at vector defined sets with

zonal statistics



point values




```
stats = zonal_stats(s_cities, X[0,:,:], transform=transform, nodata=np.nan)
mean_stats = np.array([x['mean'] for x in stats])
print(mean_stats)
```

```
[11.90853691 1.19109964 10.70276642 51.79537201 11.90853691
5.77838135 5.5387764 20.57175636 18.27160645 5.77838135
```

Organizing data across cities with GeoPandas

```
m = 0
list_of_dfs = []
for i in range(2012,2023+1):
    for j in range(1,12+1):
        col_name = '{}{:02}'.format(i,j)
#         arr = point_query(s_cities, X[m,:,:], affine=window_transform, nodata=np.nan)
        stats = zonal_stats(s_cities, X[m,:,:], transform=transform, nodata=np.nan)
        arr = np.array([x['mean'] for x in stats])
        df =  pd.DataFrame( arr,
                            columns=[col_name])
        list_of_dfs.append(df)
        m += 1
```

```
cities_gdf = gpd.GeoDataFrame(pd.concat(list_of_dfs, axis=1), geometry=s_cities )
cities_gdf
cities_gdf = cities_gdf.fillna(value=np.nan)
```

Organizing data across cities with xarray

```
time = pd.date_range(start_date, periods=12*nyears+add_months,freq='M')

mon_precip_xr = xr.DataArray(data=cities_gdf.values[:,0:X.shape[0]],
                 dims=["city_index","time"],
                coords = dict(time=time,
                reference_time=reference_time))
```

xarray.DataArray (city_index: 33, time: 144)

```
array([[11.908536911010742, 39.55228805541992, 1.1004893779754639, ...,
        nan, nan, nan],
       [1.1910996437072754, 17.534093856811523, 3.8624894618988037, ...,
```

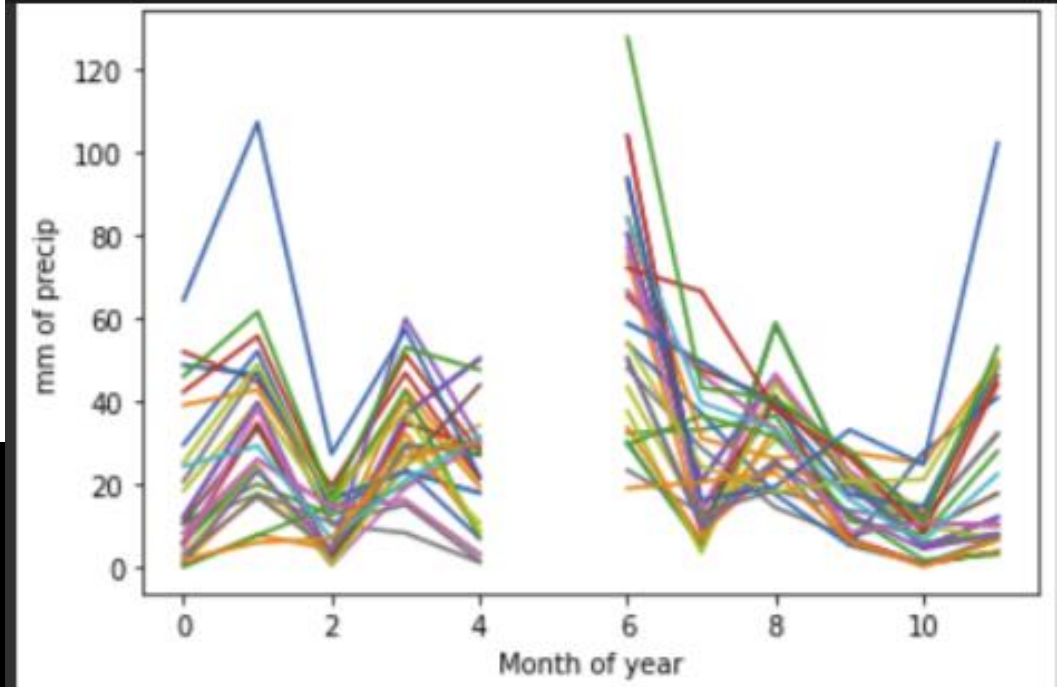
▼ Coordinates:

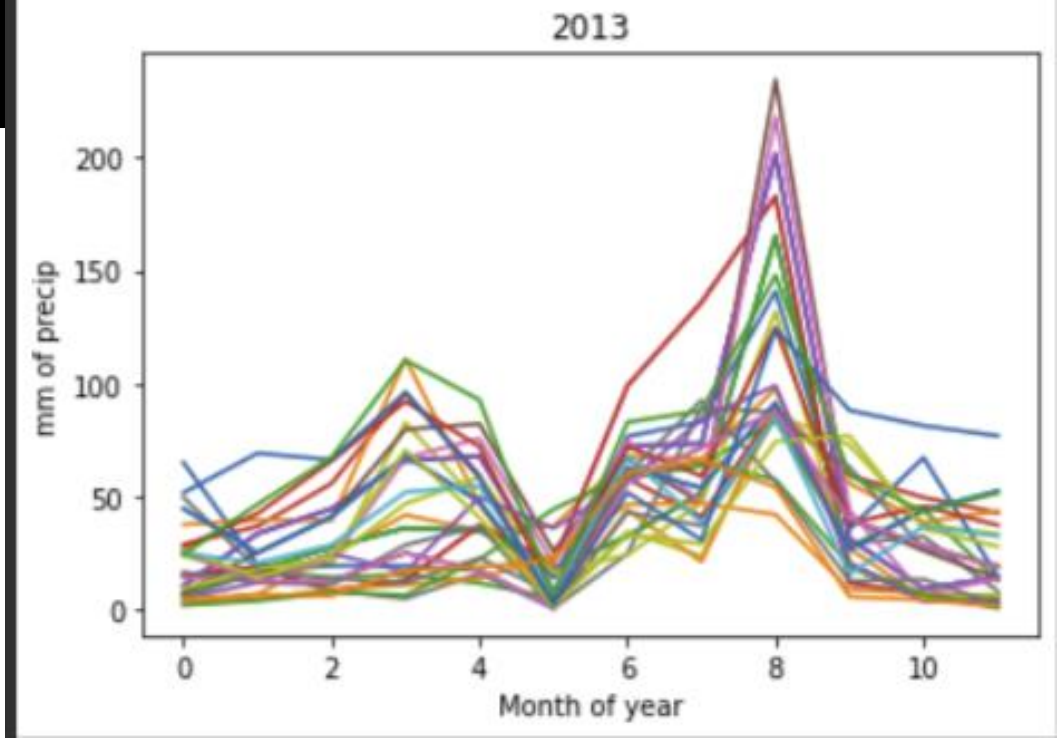
time	(time)	datetime64[ns]	2012-01-31 ... 2023-12-31
reference_time	()	datetime64[ns]	2012-01-01

▶ Indexes: (1)

▶ Attributes: (0)

```
for y in range(2012,2023+1):
  plt.figure()
  plt.plot(mon_precip_xr.groupby("time.year")[y].T)
  plt.title('{}'.format(y))
  plt.xlabel('Month of year’)
  plt.ylabel('mm of precip’)
  plt.show()
```

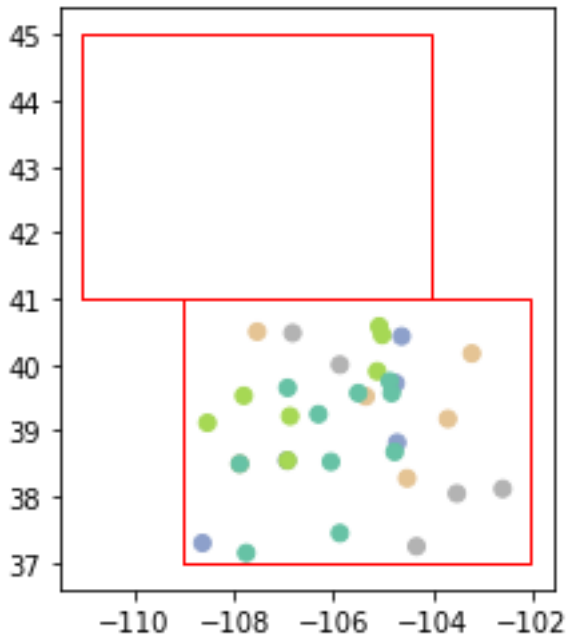




```
from sklearn.cluster import KMeans

x = np.reshape(precip_monthly.data,(12,-1)).T
print(x.shape)
kmeans = KMeans(n_clusters=5, random_state=200, n_init="auto").fit(x)
print(kmeans.labels_)
cities_gdf['cluster'] = kmeans.labels_
```

```
ax1 = cities_gdf.plot(column='cluster',cmap="Set2")
s_states.plot(facecolor='None',ax=ax1,edgecolor='r')
```



Quiz - Question 1 (private/direct message me in Zoom; if you need more time per DSS send me your answers by email)

- Please describe in your own words (and or code) how you would find the set of households that are not 2 units from any park and are not 3 units from any grocery store.

Quiz - Question 2
(private message me in Zoom)

- Please name or describe three techniques or concepts that you have learned so far in this module.

Quiz - Question 3
(private message me in Zoom)

Consider the following observational experiment.

Survey northbound vehicles on I-95 between 7:00–8:00 am on 3/1/2023.

Attributes collected (4 in total)

(time-stamp, type, mass in kg, state of license)

$\in \mathbb{R}_{>0} \times$ {car, bus, semi,…}$\times \mathbb{R}_{>0} \times$ {AL, AK,..}

4 attributes are collected per vehicle.

Which could be modeled as continuous random variables, and which are discrete random variables?

Quiz - Question 4
(private message me in Zoom)

Please give an example of two variables that show spatial covariance.

At what scale does this covariance break down or lessen?