--- title: "The Change in Phoenix, Arizona's Biodiversity Intactness" description: "Using satellite imagery to visualize fire scars" author: name: Joshua Ferrer-Lozano affiliation: Master of Environmental Data Science categories: [Python, MEDS, data-viz, Geospatial] date: 12-03-2025 format: html code-fold: true jupyter: eds220-env draft: false execute: warning: false message: false ---

Biodiversity Intactness Index Change in Phoenix, AZ

Author : Joshua Ferrer-Lozano Date: 12/5/2025

You can access the full analysis of the project in this Github repository

Purpose

This notebook investigates changes in the Biodiversity Intactness Index (BII) in the Phoenix subdivision between 2017 and 2020. The goal is to understand how rapid urban expansion has affected biodiversity.

Highlights

- Imported and explored BII raster datasets (2017, 2020) from the Microsoft Planetary Computer.
- Clipped rasters to the Phoenix subdivision shapefile.
- Calculated the percentage of area with BII ≥ 0.75 for both years.
- Visualized areas of biodiversity loss between 2017 and 2020.

About the Data

- **BII Time Series (2017, 2020):** Global 100m projections of biodiversity intactness, accessed via Microsoft Planetary Computer STAC catalog.
- **Phoenix Subdivision Shapefile:** Census County Subdivision shapefiles for Arizona, used to define the study area.
- **Contextual Data:** Optional basemap tiles from `contextily` for geographic context.

References

- Levitt, Z., & Eng, J. (2021). *Where America's developed areas are growing: 'Way off into the horizon'*. The Washington Post. https://www.washingtonpost.com/nation/interactive/2021/land-development-urban-growth-maps/
- Gassert, F., Mazzarello, J., & Hyde, S. (2022). *Global 100m Projections of Biodiversity Intactness for the years 2017-2020 [Technical Whitepaper].* https://ai4edatasetspublicassets.blob.core.windows.net/assets/pdfs/io-biodiversity/Biodiversity_Intactness_whitepaper.pdf

- Microsoft Planetary Computer, STAC Catalog. Biodiversity Intactness ('io-biodiversity'). [Dataset]. https://planetarycomputer.microsoft.com/dataset/io-biodiversity Accessed 6 December 2023.

Import Libraries

In [ ]:

```
# Load in Libraries
import os
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import matplotlib.patches as patch

# geospatial libraries
import xarray as xr
import rioxarray as rio
from shapely.geometry import box

# Planetary Computer Access
import contextily as ctx
import planetary_computer

# Check python environment
import sys
print(sys.executable)

from pystac_client import Client
```

```
c:\Users\joshu\.conda\envs\eds220-env\python.exe
```

Access Planetary Computer STAC Catalog: io-biodiversity collection

We connect to the Microsoft Planetary Computer STAC API to access the **io-biodiversity** collection. This dataset provides global 100m projections of biodiversity intactness for the years 2017–2020.

In [ ]:

```
# Connect to the Planetary Computer STAC API
catalog = Client.open(
    "https://planetarycomputer.microsoft.com/api/stac/v1",
    modifier=planetary_computer.sign_inplace,
)
# Exploring catalog metadata
print('Title', catalog.title)

# Pull out the io_biodiversity collection
io_collection = catalog.get_child('io_biodiversity')
print(io_collection)
```

```
Title Microsoft Planetary Computer STAC API
None
```

Define our parameters

In [4]:

```
# Temporal range of interest (2017–2020 drought period)
time_range = '2017-01-01/2020-01-01'

# Bounding box for Phoenix subdivision
bbox = [-112.826843, 32.974108, -111.184387, 33.863574]

# Get items from search
search = catalog.search(
    collections = ['io-biodiversity'],
    bbox = bbox,
    datetime = time_range)

# Retrieve items
items = search.item_collection()
print(len(items))
```

```
4
```

Explore the search

In [5]:

```
# Get the io-biodiversity collection
io_collection = catalog.get_collection('io-biodiversity')

# Print the collection metadata
print("\nID:", io_collection.id)
print("Title:", io_collection.title)
print("Description:", io_collection.description)
print("Extent:", io_collection.extent)

# Convert items to a list
all_items = list(search.items())

# Print item IDs and properties
for item in all_items:
    print(item.id, item.properties)
```

```
ID: io-biodiversity
Title: Biodiversity Intactness
Description: Generated by [Impact Observatory](https://www.impactobservatory.com/),
in collaboration with [Vizzuality](https://www.vizzuality.com/), these datasets esti
mate terrestrial Biodiversity Intactness as 100-meter gridded maps for the years 201
7-2020.

Maps depicting the intactness of global biodiversity have become a critical tool for
spatial planning and management, monitoring the extent of biodiversity across Earth,
and identifying critical remaining intact habitat. Yet, these maps are often years o
ut of date by the time they are available to scientists and policy-makers. The datas
ets in this STAC Collection build on past studies that map Biodiversity Intactness u
sing the [PREDICTS database](https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.2
579) of spatially referenced observations of biodiversity across 32,000 sites from o
ver 750 studies. The approach differs from previous work by modeling the relationshi
p between observed biodiversity metrics and contemporary, global, geospatial layers
of human pressures, with the intention of providing a high resolution monitoring pro
duct into the future.

Biodiversity intactness is estimated as a combination of two metrics: Abundance, the
quantity of individuals, and Compositional Similarity, how similar the composition o
f species is to an intact baseline. Linear mixed effects models are fit to estimate
the predictive capacity of spatial datasets of human pressures on each of these metr
ics and project results spatially across the globe. These methods, as well as compar
isons to other leading datasets and guidance on interpreting results, are further ex
plained in a methods [white paper](https://ai4edatasetspublicassets.blob.core.window
s.net/assets/pdfs/io-biodiversity/Biodiversity_Intactness_whitepaper.pdf) entitled
“Global 100m Projections of Biodiversity Intactness for the years 2017-2020.”

All years are available under a Creative Commons BY-4.0 license.

Extent: <pystac.collection.Extent object at 0x000001E690C13F10>
bii_2020_34.74464974521749_-115.38597824385106_cog {'datetime': None, 'proj:shape':
[7992, 7992], 'end_datetime': '2020-12-31T23:59:59Z', 'proj:transform': [0.000898315
2841195215, 0.0, -115.38597824385106, 0.0, -0.0008983152841195215, 34.7446497452174
9, 0.0, 0.0, 1.0], 'start_datetime': '2020-01-01T00:00:00Z', 'proj:code': 'EPSG:432
6'}
bii_2019_34.74464974521749_-115.38597824385106_cog {'datetime': None, 'proj:shape':
[7992, 7992], 'end_datetime': '2019-12-31T23:59:59Z', 'proj:transform': [0.000898315
2841195215, 0.0, -115.38597824385106, 0.0, -0.0008983152841195215, 34.7446497452174
9, 0.0, 0.0, 1.0], 'start_datetime': '2019-01-01T00:00:00Z', 'proj:code': 'EPSG:432
6'}
bii_2018_34.74464974521749_-115.38597824385106_cog {'datetime': None, 'proj:shape':
[7992, 7992], 'end_datetime': '2018-12-31T23:59:59Z', 'proj:transform': [0.000898315
2841195215, 0.0, -115.38597824385106, 0.0, -0.0008983152841195215, 34.7446497452174
9, 0.0, 0.0, 1.0], 'start_datetime': '2018-01-01T00:00:00Z', 'proj:code': 'EPSG:432
6'}
bii_2017_34.74464974521749_-115.38597824385106_cog {'datetime': None, 'proj:shape':
[7992, 7992], 'end_datetime': '2017-12-31T23:59:59Z', 'proj:transform': [0.000898315
2841195215, 0.0, -115.38597824385106, 0.0, -0.0008983152841195215, 34.7446497452174
9, 0.0, 0.0, 1.0], 'start_datetime': '2017-01-01T00:00:00Z', 'proj:code': 'EPSG:432
6'}
```

In [6]:

```
# Retrieve search items
items = search.item_collection()
print(f"There is {len(items)} items in the search.")
```

```
There is 4 items in the search.
```

Select 2017 and 2020 Raster

In [7]:

```
# Create a lookup table
lookup = {int(item.properties["start_datetime"][:4]): item for item in items} # yea

# Store 2017 and 2020 raster
phoenix_2017 = lookup[2017]
phoenix_2020 = lookup[2020]
```

Arizona Data

In [9]:

```
# Import Arizona Census Subdivision data

# display all columns when looking at dataframes
pd.set_option("display.max.columns", None)

arizona_fp = os.path.join('data','tl_2022_04_cousub','tl_2022_04_cousub.shp')
arizona = gpd.read_file(arizona_fp)

# preliminary exploration
arizona.head(5)
```

Out[9]:

	STATEFP	COUNTYFP	COUSUBFP	COUSUBNS	GEOID	NAME	NAMELSAD	LSAI
0	04	005	91198	01934931	0400591198	Flagstaff	Flagstaff CCD	2
1	04	005	91838	01934953	0400591838	Kaibab Plateau	Kaibab Plateau CCD	2
2	04	005	91683	01934950	0400591683	Hualapai	Hualapai CCD	2
3	04	023	92295	01934961	0402392295	Nogales	Nogales CCD	2
4	04	023	92550	01934966	0402392550	Patagonia	Patagonia CCD	2

Map the subdivision

In [11]:

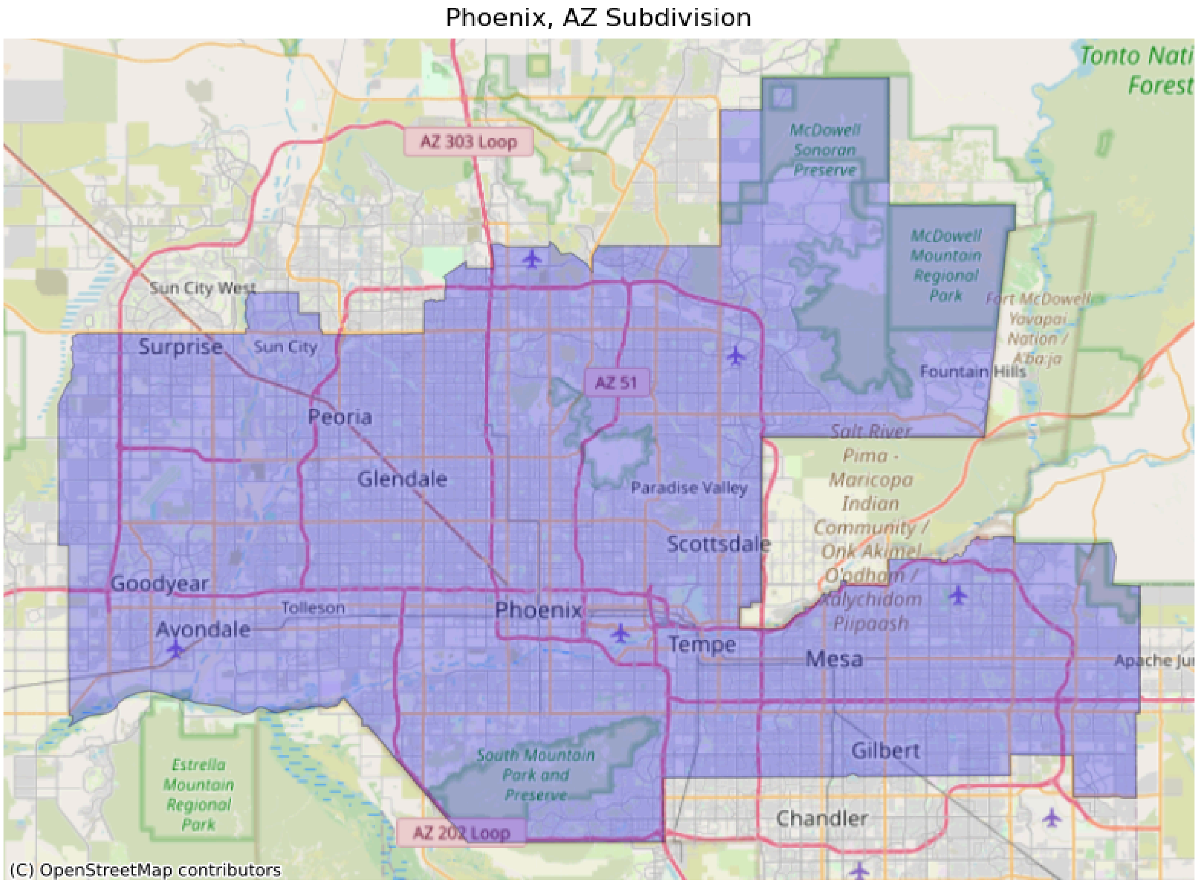
```
# Select the Phoenix Subdivision
phoenix = arizona[arizona["NAME"] == "Phoenix"]

print(phoenix.crs)
```

```
EPSG:4269
```

In [12]:

```
# Create map of Phoenix subdivision with basemap
fig, ax = plt.subplots(1, 1, figsize=(9, 6))

# plot Phenix subdivision polygon
phoenix.plot(ax=ax, facecolor='blue', edgecolor='black', linewidth=1, alpha=0.25)

# Add OpenStreetMap basemap
ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik, crs = phoenix.crs)

# Add title and remove axes
ax.set_title('Phoenix, AZ Subdivision')
```

```
# Remove axes
ax.axis('off')

# Display the map
plt.tight_layout()
plt.show()
```



Phoenix, AZ Subdivision

In [15]:

```
# Select 2017 and 2020 rasters from search
item_2020 = items[0]
item_2017 = items[3]

# Extract 2017 and 2020 rasters using open_rasterio()
phoenix_2017 = rio.open_rasterio(planetary_computer.sign(item_2017.assets["data"]).
phoenix_2020 = rio.open_rasterio(planetary_computer.sign(item_2020.assets["data"]).
```

In [17]:

```
# Align Phoenix subdivision with raster CRS
phoenix = phoenix.to_crs(phoenix_2020.rio.crs)


# Clip rasters to Phoenix subdivision boundary
phoenix_2017_clip = phoenix_2017.rio.clip(phoenix.geometry, phoenix.crs)
phoenix_2020_clip = phoenix_2020.rio.clip(phoenix.geometry, phoenix.crs)
```

In [18]:

```
# Identify area with biodiversity intactness index is > .75
phoenix_mask_2017 = (phoenix_2017_clip >= 0.75)
phoenix_mask_2020 = (phoenix_2020_clip >= 0.75)
```

In [19]:

```
# Calculate percentage of Phoenix area with BII > .75
pct_bii_2017 = float(phoenix_mask_2017.mean()) * 100
pct_bii_2020 = float(phoenix_mask_2020.mean()) * 100

print(f"Phoenix subdivision BII ≥ 0.75: {pct_bii_2017:.2f}% in 2017, {pct_bii_2020:
```

```
Phoenix subdivision BII ≥ 0.75: 4.18% in 2017, 3.81% in 2020
```

In [23]:

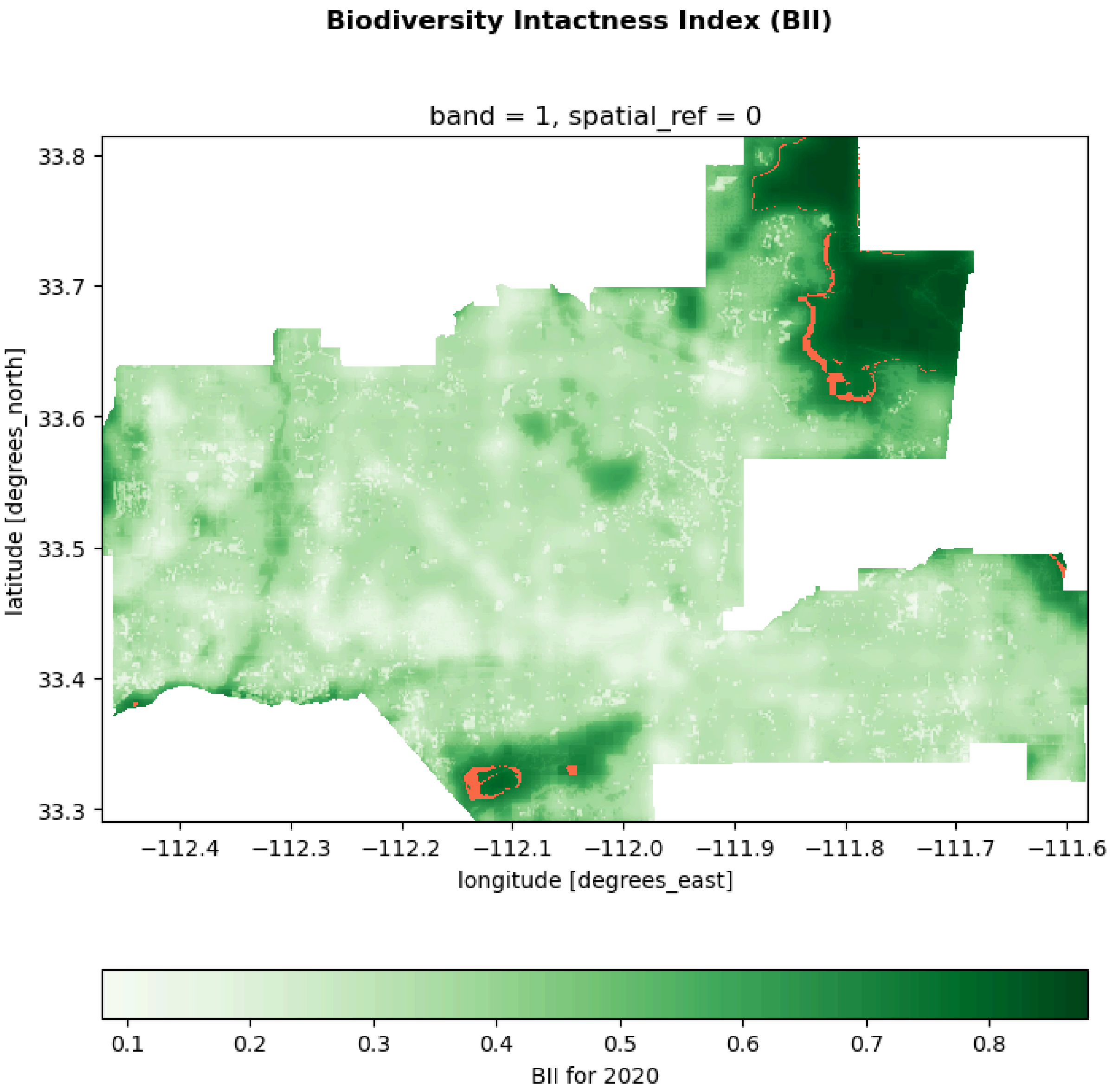
```
# Identify areas that were greater than .75 in 2017 but dropped below .75 in 2020
bii_loss_mask = phoenix_mask_2017 & (~phoenix_mask_2020)

# Convert 0 values to NAs if the conditions are not met
bii_loss_mask = bii_loss_mask.where(bii_loss_mask!=0)
```

In [24]:

```
# Plot mask visualization
fig, ax = plt.subplots(figsize=(8, 8))

# Create choropleth map
phoenix_2020_clip.plot(cmap = 'Greens', cbar_kwargs={'location':'bottom', # Legend
                                                     'label':'BII for 2020'})

# Plot the lost mask
bii_loss_mask.plot(ax=ax, cmap='Reds', add_colorbar = False)

# Add title
plt.suptitle('Biodiversity Intactness Index (BII)', fontsize=12, weight = 'bold')
```

Out[24]:

```
Text(0.5, 0.98, 'Biodiversity Intactness Index (BII)')
```


Biodiversity Intactness Index (BII)
band = 1, spatial_ref = 0
33.8
33.7
33.6
33.5
33.4
33.3
latitude [degrees_north]
−112.4
−112.3
−112.2
−112.1
−112.0
−111.9
−111.8
−111.7
−111.6
longitude [degrees_east]
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
BII for 2020