

# species-working-v1

October 4, 2024

## 0.1 Species Distribution

### 0.1.1 Working code assembled from segments

```
[1]: # bring in libraries
# some spec by CGPT
# zipfile is downloads on windows
#
# This code checks and prints out the ecoregions of the world

import os
import pathlib
import requests
import zipfile
import geopandas as gpd
import glob
import matplotlib.pyplot as plt

# Create data directory in the home folder
data_dir = os.path.join(
    # Home directory
    pathlib.Path.home(),
    # Earth analytics data directory
    'earth-analytics',
    'data',
    # Project directory
    'migrationstudy',
)
os.makedirs(data_dir, exist_ok=True)
print(data_dir)

# Create data directory in the home folder
data_dir = os.path.join(
    # Home directory
    pathlib.Path.home(),
    # Earth analytics data directory
    'earth-analytics',
    'data',
```

```

    # Project directory
    'migrationstudy',
)
os.makedirs(data_dir, exist_ok=True)

# Set up the ecoregion boundary URL
url = "https://storage.googleapis.com/teow2016/Ecoregions2017.zip"

# Set up a path to save the zip file and extracted data
ecoregion_zip_path = os.path.join(data_dir, 'Ecoregions2017.zip')
ecoregion_extract_path = os.path.join(data_dir, 'ecoregions')

# Download the ZIP file if it doesn't exist
if not os.path.exists(ecoregion_zip_path):
    response = requests.get(url)
    with open(ecoregion_zip_path, 'wb') as file:
        file.write(response.content)

# Extract the zip file if the directory doesn't exist
if not os.path.exists(ecoregion_extract_path):
    with zipfile.ZipFile(ecoregion_zip_path, 'r') as zip_ref:
        zip_ref.extractall(ecoregion_extract_path)

# Path to the shapefile
shapefile_path = os.path.join(ecoregion_extract_path, 'Ecoregions2017.shp')

# Only read the shapefile once it is extracted
if not os.path.exists(shapefile_path):
    print("Shapefile not found after extraction!")
else:
    # Read the shapefile with geopandas
    my_gdf = gpd.read_file(shapefile_path)
    # Optionally save the shapefile to a new location if needed
    my_gdf.to_file(os.path.join(data_dir, 'ecoregions_shape.shp'))
    print("Ecoregions shapefile saved successfully!")

# Function to find all shapefiles in the species-distribution directory
def find_shapefiles(directory):
    # Use glob to find all .shp files recursively in the specified directory
    shapefiles = glob.glob(os.path.join(directory, '**', '*.shp'),
↪ recursive=True)
    return shapefiles

# Define the species distribution directory
species_distribution_dir = os.path.join(
    pathlib.Path.home(),
    'earth-analytics',

```

```

    'data',
    'species-distribution'
)

# Find all shapefiles in the species distribution directory
shapefiles = find_shapefiles(species_distribution_dir)

# Print all found shapefiles
print("Found shapefiles:")
for shp in shapefiles:
    print(shp)

# Path to the shapefile - update this with the actual file path
data_dir = os.path.join(
    pathlib.Path.home(),
    'earth-analytics',
    'data',
    'migrationstudy',
    'ecoregions' # Assuming the shapefile was extracted here
)

shapefile_path = os.path.join(data_dir, 'Ecoregions2017.shp') # Ensure this
↳ path points to the actual shapefile

# Check if the file exists before proceeding
if not os.path.exists(shapefile_path):
    print(f"Shapefile not found at {shapefile_path}")
else:
    # Open up the ecoregions boundaries
    gdf = gpd.read_file(shapefile_path)

    # Name the index so it will match the other data later on
    gdf.index.name = 'ecoregion'

    # Examine the ecoregion GeoDataFrame
    print(gdf.head())

    # Plot the ecoregions to check the download
    # Customize the plot with a title and a color map
    fig, ax = plt.subplots(figsize=(20, 16)) # Adjust the figure size for
↳ better visibility
    gdf.plot(ax=ax, column='BIOME_NAME', legend=True, cmap='Set3') # Assuming
↳ 'BIOME_NAME' is a valid column

    # Set a title for the plot
    ax.set_title("Ecoregions of the World", fontsize=12)

```

```
# Show the plot
plt.show()
```

/home/jovyan/earth-analytics/data/migrationstudy

Ecoregions shapefile saved successfully!

Found shapefiles:

	OBJECTID	ECO_NAME \
ecoregion		
0	1.0	Adelie Land tundra
1	2.0	Admiralty Islands lowland rain forests
2	3.0	Aegean and Western Turkey sclerophyllous and m...
3	4.0	Afghan Mountains semi-desert
4	5.0	Ahklun and Kilbuck Upland Tundra

	BIOME_NUM	BIOME_NAME \
ecoregion		
0	11.0	Tundra
1	1.0	Tropical & Subtropical Moist Broadleaf Forests
2	12.0	Mediterranean Forests, Woodlands & Scrub
3	13.0	Deserts & Xeric Shrublands
4	11.0	Tundra

	REALM	ECO_BIOME_	NNH	ECO_ID	SHAPE LENG	SHAPE_AREA \
ecoregion						
0	Antarctica	AN11	1	117	9.749780	0.038948
1	Australasia	AU01	2	135	4.800349	0.170599
2	Palearctic	PA12	4	785	162.523044	13.844952
3	Palearctic	PA13	4	807	15.084037	1.355536
4	Nearctic	NE11	1	404	22.590087	8.196573

	NNH_NAME	COLOR	COLOR_BIO	COLOR_NNH \
ecoregion				
0	Half Protected	#63CFAB	#9ED7C2	#257339
1	Nature Could Reach Half Protected	#70A800	#38A700	#7BC141
2	Nature Imperiled	#FF7F7C	#FE0000	#EE1E23
3	Nature Imperiled	#FA774D	#CC6767	#EE1E23
4	Half Protected	#4C82B6	#9ED7C2	#257339

	LICENSE	geometry
ecoregion		
0	CC-BY 4.0	MULTIPOLYGON (((158.7141 -69.60657, 158.71264 ...
1	CC-BY 4.0	MULTIPOLYGON (((147.28819 -2.57589, 147.2715 -...
2	CC-BY 4.0	MULTIPOLYGON (((26.88659 35.32161, 26.88297 35...
3	CC-BY 4.0	MULTIPOLYGON (((65.48655 34.71401, 65.52872 34...
4	CC-BY 4.0	MULTIPOLYGON (((-160.26404 58.64097, -160.2673...

