

Threatened ecosystems of Ruritania

An IUCN Red List of Ecosystems Assessment

Jane Smith

José García

Wei Wang

1234-12-31

Table of contents

Home	3
Foreword	4
Preface	5
Acknowledgments	6
Coordinating Organizations	7
Organization #1	7
Organization #2	7
Summary	8
Introduction	9
Background	9
The ecosystems of Ruritania: an overview	9
Introduction to the IUCN Red List of Ecosystems	11
Definitions	12
IUCN Red List of Ecosystems Categories and Criteria	14
Assessment process	15
Terrestrial Ecosystems of Ruritania	17
Ecosystem description and assessment	27
T1.1.6 - Catatumbo moist forests	29
Assessment	30
Catatumbo moist forests	31
Criterion B Details	34
Load & Filter Ecosystem Data	34
Extent of Occurrence (EOO) (subcriterion B1)	35
Calculate EOO	35

Direct calculation of EOO	36
Area of Occupancy (AOO) (subcriterion B2)	37
AOO Calculation Details	37
AOO Calculation (direct call)	40
Criterion B Summary	41
T1.2.1 - Catatumbo Ocaña dry forests	42
Assessment	43
Catatumbo Ocaña dry forests	44
Criterion B Details	47
Load & Filter Ecosystem Data	47
Extent of Occurrence (EOO) (subcriterion B1)	48
Calculate EOO	48
Direct calculation of EOO	49
Area of Occupancy (AOO) (subcriterion B2)	50
AOO Calculation Details	50
AOO Calculation (direct call)	54
Criterion B Summary	55
T1.2.19 - Patia-Dagua dry forests	56
Assessment	57
Patia-Dagua dry forests	58
Criterion B Details	61
Load & Filter Ecosystem Data	61
Extent of Occurrence (EOO) (subcriterion B1)	62
Calculate EOO	62
Direct calculation of EOO	63
Area of Occupancy (AOO) (subcriterion B2)	64
AOO Calculation Details	64
AOO Calculation (direct call)	69
Criterion B Summary	69
T1.2.8 - Chicamocha Valley dry forests	70
Assessment	71

Chicamocha Valley dry forests	72
Criterion B Details	75
Load & Filter Ecosystem Data	75
Extent of Occurrence (EOO) (subcriterion B1)	76
Calculate EOO	76
Direct calculation of EOO	77
Area of Occupancy (AOO) (subcriterion B2)	78
AOO Calculation Details	78
AOO Calculation (direct call)	81
Criterion B Summary	82
T3.1.1 - Pantepui	83
Assessment	84
Pantepui	85
Criterion B Details	88
Load & Filter Ecosystem Data	88
Extent of Occurrence (EOO) (subcriterion B1)	89
Calculate EOO	89
Direct calculation of EOO	90
Area of Occupancy (AOO) (subcriterion B2)	91
AOO Calculation Details	91
AOO Calculation (direct call)	94
Criterion B Summary	95
T4.2.2 - Llanos	96
Assessment	97
Llanos	98
Criterion B Details	101
Load & Filter Ecosystem Data	101
Extent of Occurrence (EOO) (subcriterion B1)	102
Calculate EOO	102
Direct calculation of EOO	103
Area of Occupancy (AOO) (subcriterion B2)	104
AOO Calculation Details	104
AOO Calculation (direct call)	107

Criterion B Summary	107
T4.2.5 - Llanos	108
Assessment	109
Llanos	110
Criterion B Details	113
Load & Filter Ecosystem Data	113
Extent of Occurrence (EOO) (subcriterion B1)	114
Calculate EOO	114
Direct calculation of EOO	115
Area of Occupancy (AOO) (subcriterion B2)	116
AOO Calculation Details	116
AOO Calculation (direct call)	119
Criterion B Summary	120
T5.2.1 - Guajira-Barranquilla xeric scrub	121
Assessment	122
Guajira-Barranquilla xeric scrub	123
Criterion B Details	126
Load & Filter Ecosystem Data	126
Extent of Occurrence (EOO) (subcriterion B1)	127
Calculate EOO	127
Direct calculation of EOO	128
Area of Occupancy (AOO) (subcriterion B2)	129
AOO Calculation Details	129
AOO Calculation (direct call)	132
Criterion B Summary	133
T5.2.2 - Guajira-Barranquilla xeric scrub	134
Assessment	135
Guajira-Barranquilla xeric scrub	136
Criterion B Details	139
Load & Filter Ecosystem Data	139

Extent of Occurrence (EOO) (subcriterion B1)	140
Calculate EOO	140
Direct calculation of EOO	141
Area of Occupancy (AOO) (subcriterion B2)	142
AOO Calculation Details	142
AOO Calculation (direct call)	148
Criterion B Summary	149
T5.2.3 - Magdalena Valley dry forests	150
Assessment	151
Magdalena Valley dry forests	152
Criterion B Details	155
Load & Filter Ecosystem Data	155
Extent of Occurrence (EOO) (subcriterion B1)	156
Calculate EOO	156
Direct calculation of EOO	157
Area of Occupancy (AOO) (subcriterion B2)	158
AOO Calculation Details	158
AOO Calculation (direct call)	161
Criterion B Summary	162
References	163

Home

Jane Smith
José García
Wei Wang

1234 | Version 0.1

! How to customize the Home page

- Edit the `_quarto.yml` file:
 - Update the `version` value
 - Update the `book.title` value
- Edit the `index.qmd` file:
 - Update the `subtitle` value
 - Update the `authors` list

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

Foreword

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sagittis posuere ligula sit amet lacinia. Duis dignissim pellentesque magna, rhoncus congue sapien finibus mollis. Ut eu sem laoreet, vehicula ipsum in, convallis erat. Vestibulum magna sem, blandit pulvinar augue sit amet, auctor malesuada sapien. Nullam faucibus leo eget eros hendrerit, non laoreet ipsum lacinia. Curabitur cursus diam elit, non tempus ante volutpat a. Quisque hendrerit blandit purus non fringilla. Integer sit amet elit viverra ante dapibus semper. Vestibulum viverra rutrum enim, at luctus enim posuere eu. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

! How to customize the Foreword page

- To modify the foreword text, edit the `content/0.1_foreword.qmd` file.
- To add an author to the foreword, add the author's name to the `authors` list in the YAML header of the `0.1_foreword.qmd` file. For example:

```
authors:  
  - name: Ima Author
```

- To remove the foreword section, edit the `_quarto.yml` file and remove the `- content/0.1_foreword.qmd` line.

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

Preface

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sagittis posuere ligula sit amet lacinia. Duis dignissim pellentesque magna, rhoncus congue sapien finibus mollis. Ut eu sem laoreet, vehicula ipsum in, convallis erat. Vestibulum magna sem, blandit pulvinar augue sit amet, auctor malesuada sapien. Nullam faucibus leo eget eros hendrerit, non laoreet ipsum lacinia. Curabitur cursus diam elit, non tempus ante volutpat a. Quisque hendrerit blandit purus non fringilla. Integer sit amet elit viverra ante dapibus semper. Vestibulum viverra rutrum enim, at luctus enim posuere eu. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

! How to customize the Preface page

- To modify the preface text, edit the `content/0.2_preface.qmd` file.
- To add an author to the preface, add the author's name to the `authors` list in the YAML header of the `0.2_preface.qmd` file. For example:

```
authors:  
  - name: Ima Author
```

- To remove the preface section, edit the `_quarto.yml` file and remove the `- content/0.2_preface.qmd` line.

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

Acknowledgments

We gratefully acknowledge the efforts of hundreds of scientists and practitioners who have contributed to the understanding of Ruritania's terrestrial ecosystems. Our work is largely a synthesis exercise that would not have been possible without the dedication and contributions of those before us.

We also acknowledge all contributors to this project, particularly those who attended workshops in Ruritania. Without input from a large and diverse expert group, the development of the Ruritania Ecosystem Typology, the ecosystem descriptions and the application of the criteria would not have been possible. We also acknowledge the crucial support of ... Contributors to this assessment are listed within each ecosystem assessment and in Appendix 1.

The work to develop an IUCN Red List of Ecosystems of Ruritania was made possible through the funding and support of ...

! How to customize the Acknowledgments page

- To modify the acknowledgments text, edit the `content/0.3_acknowledgments.qmd` file.
- To remove the acknowledgments section, edit the `_quarto.yml` file and remove the `- content/0.3_acknowledgments.qmd` line.

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

Coordinating Organizations

Organization #1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sagittis posuere ligula sit amet lacinia. Duis dignissim pellentesque magna, rhoncus congue sapien finibus mollis. Ut eu sem laoreet, vehicula ipsum in, convallis erat. Vestibulum magna sem, blandit pulvinar augue sit amet, auctor malesuada sapien. Nullam faucibus leo eget eros hendrerit, non laoreet ipsum lacinia. Curabitur cursus diam elit, non tempus ante volutpat a. Quisque hendrerit blandit purus non fringilla. Integer sit amet elit viverra ante dapibus semper. Vestibulum viverra rutrum enim, at luctus enim posuere eu. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Organization #2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sagittis posuere ligula sit amet lacinia. Duis dignissim pellentesque magna, rhoncus congue sapien finibus mollis. Ut eu sem laoreet, vehicula ipsum in, convallis erat. Vestibulum magna sem, blandit pulvinar augue sit amet, auctor malesuada sapien. Nullam faucibus leo eget eros hendrerit, non laoreet ipsum lacinia. Curabitur cursus diam elit, non tempus ante volutpat a. Quisque hendrerit blandit purus non fringilla. Integer sit amet elit viverra ante dapibus semper. Vestibulum viverra rutrum enim, at luctus enim posuere eu. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

! How to customize the Coordinating Organizations page

- To modify the coordinating organizations text, edit the `content/0.4_coordinating_orgs.qmd` file.
- To remove the coordinating organizations section, edit the `_quarto.yml` file and remove the `- content/0.4_coordinating_orgs.qmd` line.

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

Summary

We gratefully acknowledge the efforts of hundreds of scientists and practitioners who have contributed to the understanding of Ruritania's terrestrial ecosystems over the past two centuries. Our work is largely a synthesis exercise that would not have been possible without the dedication and contributions of those before us.

We also acknowledge all contributors to this project, particularly those who attended workshops in Ruritania. Without input from a large and diverse expert group, the development of the Ruritania Ecosystem Typology, the ecosystem descriptions and the application of the criteria would not have been possible. We also acknowledge the crucial support of ... Contributors to this assessment are listed within each ecosystem assessment and in Appendix 1.

The work to develop an IUCN Red List of Ecosystems of Ruritania was made possible through the funding and support of ...

! How to customize the Summary page

- To modify the summary text, edit the `content/0.5_summary.qmd` file.
- To remove the summary section, edit the `_quarto.yml` file and remove the `- content/0.5_summary.qmd` line.

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

Introduction

Background

To support the Ruritania National Ecosystem Assessment, Ruritania's terrestrial ecosystems were assessed under the International Union for the Conservation of Nature (IUCN) Red List of Ecosystems Categories and Criteria.

This report describes the development of the IUCN Red List of Ecosystems for Ruritania, which included:

- A detailed literature review of all published and unpublished material relevant to the status of ecosystems in Ruritania;
- The development of an ecosystem typology for Ruritania suitable for conducting a national scale IUCN Red List of Ecosystems assessment;
- ...
- An expert review process for the ecosystem typology, descriptions and assessments.

! How to customize the Background section of the Introduction

- To modify the background text, edit the `content/1.1_background.qmd` file.
- To remove the background section, edit the `content/1.0_introduction.qmd` file and remove the `{{< include 1.1_background.qmd >}}` line.

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

The ecosystems of Ruritania: an overview

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sagittis posuere ligula sit amet lacinia. Duis dignissim pellentesque magna, rhoncus congue sapien finibus mollis. Ut eu sem laoreet, vehicula ipsum in, convallis erat. Vestibulum magna sem, blandit pulvinar augue sit amet, auctor malesuada sapien. Nullam faucibus leo eget eros hendrerit, non laoreet ipsum lacinia. Curabitur cursus diam elit, non tempus ante volutpat a. Quisque hendrerit blandit purus non fringilla. Integer sit amet elit viverra ante dapibus semper. Vestibulum viverra



Figure 1: A map of Ruritania showing the area of assessment in grey

rutrum enim, at luctus enim posuere eu. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Nunc ac dignissim magna. Vestibulum vitae egestas elit. Proin feugiat leo quis ante condimentum, eu ornare mauris feugiat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris cursus laoreet ex, dignissim bibendum est posuere iaculis. Suspendisse et maximus elit. In fringilla gravida ornare. Aenean id lectus pulvinar, sagittis felis nec, rutrum risus. Nam vel neque eu arcu blandit fringilla et in quam. Aliquam luctus est sit amet vestibulum eleifend. Phasellus elementum sagittis molestie. Proin tempor lorem arcu, at condimentum purus volutpat eu. Fusce et pellentesque ligula. Pellentesque id tellus at erat luctus fringilla. Suspendisse potenti.

! How to customize the Overview section of the Introduction

- To modify the overview text, edit the `content/1.2_overview.qmd` file.
- To remove the overview section, edit the `content/1.0_introduction.qmd` file and remove the `{{< include 1.2_overview.qmd >}}` line.

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

Introduction to the IUCN Red List of Ecosystems

The IUCN Red List of Ecosystems (RLE) aims to support conservation in resource use and management decisions by identifying ecosystems most at risk of loss or collapse (D. A. Keith et al. 2013, 2015). Similar to the IUCN Red List of Threatened Species, the outcome of an RLE assessment is a list of ecosystems and their status for a region (Figure 1.4; (Rodríguez et al. 2011)). Because the RLE was developed to promote a consistent framework suitable for assessing and monitoring the status of ecosystems, it enables comparisons of collapse risk between countries, locations and ecosystem types (D. A. Keith et al. 2013).

For further information on the development of the RLE protocol, the theory and scientific foundations upon which they were developed, and detailed information on the purpose of each of the five criteria refer to the Guidelines for the application of IUCN Red List of Ecosystems Categories and Criteria (L. Bland et al. 2017). More information on the IUCN Red List of Ecosystems, is available in multiple languages on the IUCN Red List of Ecosystems website (www.iucnrle.org).

Assessments of ecosystem types (commonly termed ‘assessment units’ within Red List of Ecosystems assessments) are conducted by applying five criteria and their associated thresholds, enabling each ecosystem type to be classified according to their risk of collapse (termed ‘status’). To ensure the assessment process is transparent and repeatable, each ecosystem type is clearly described according to the IUCN Red List of Ecosystems guidelines (L. Bland et al. 2017).

This standard approach of applying the IUCN Red List of Ecosystems Categories and Criteria to clearly described ecosystems is critical to allow for accurate, comparable and repeatable assessments of ecosystems status and to contribute to the global IUCN Red List of Ecosystems programme.

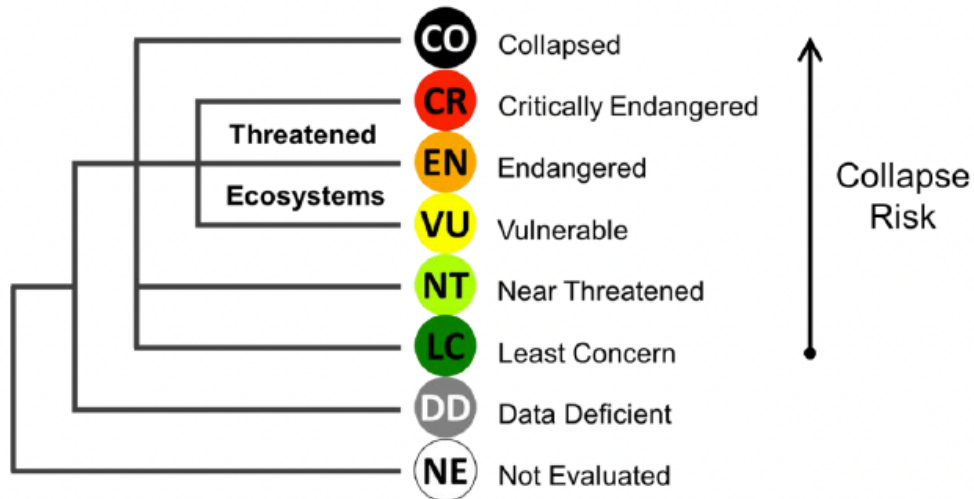


Figure 2: The IUCN Red List of Ecosystems categories, indicating the status of ecosystems. Threatened ecosystems are those assessed as Vulnerable, Endangered, or Critically Endangered. Source: (L. Bland et al. 2017)

Definitions

There are several key concepts that must be clearly defined to allow for repeatable ecosystem risk assessments:

Risk

Risk is defined as the probability of an adverse outcome over a specified time-frame. Here, the adverse outcome is the endpoint of ecosystem decline, which the RLE terms ecosystem collapse.

Ecosystem collapse

Understanding the concept of ecosystem collapse is critical for interpreting IUCN RLE assessments. For the purposes of the RLE, “an ecosystem is Collapsed when it is virtually certain that its defining biotic or abiotic features are lost from all occurrences, and the characteristic native

biota are no longer sustained. Collapse may occur when most of the diagnostic components of the characteristic native biota are lost from the system, or when functional components (biota that perform key roles in ecosystem organisation) are greatly reduced in abundance and lose the ability to recruit.” According to the IUCN guidelines (L. Bland et al. 2017), risks to ecosystems can be caused by a variety of threatening processes that are expressed through different symptoms of ecosystem collapse. The RLE risk model groups these symptoms into four major types, which ultimately form the RLE criteria (Figure 3).

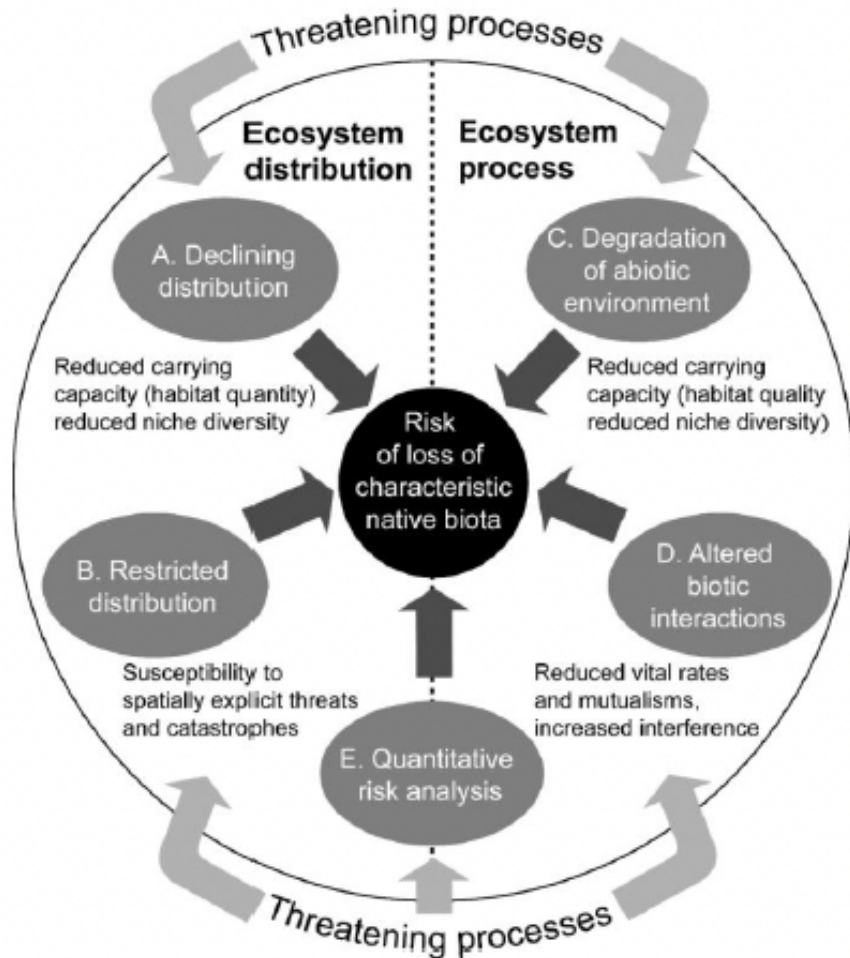


Figure 3: The IUCN Red List of Ecosystems risk assessment model. Source: (L. Bland et al. 2017).

For more information on the concept of collapse and how to identify when an ecosystem is collapsed, we recommend referring to the IUCN Red List of Ecosystems guidelines, which describes this in detail (Bland et al., 2017a; Bland et al., 2018). In this report and as

recommended by the guidelines, we explicitly define collapse for each ecosystem type in Myanmar in their ecosystem descriptions (See section 3).

Time frames

Because risks must be assessed over specified time frames, a standard set of time frames are carefully defined in the IUCN Red List of Ecosystems Categories and Criteria. There are four specified time frames used in the RLE:

- The historical past. We notionally use the year 1750, which marks the onset of industrial-scale exploitation of ecosystems in South-East Asia;
- The recent past. This is the past 50 years (1969-2019), which is considered long enough to distinguish directional change from natural variability;
- Any 50-year period including the recent past, present and future. Predictions and inferences based on past declines, simulation models and any other model considered suitable for assessing risks into the future may be used.
- The future. Again, predictions are required to assess risks over this time frame and are usually based on models that use information about the response of ecosystems to threatening processes.

IUCN Red List of Ecosystems Categories and Criteria

IUCN Red List of Ecosystems Criteria

To assess the risk of ecosystem collapse, each ecosystem is assessed under five rule-based criteria that form the IUCN Red List of Ecosystems Criteria. These criteria were developed following nearly a decade of scientific work focused on understanding pathways of ecosystem decline, degradation, loss and collapse (Nicholson, Keith, and Wilcove 2009; Rodríguez et al. 2011, 2015; D. A. Keith et al. 2013, 2015; L. M. Bland et al. 2017; Murray et al. 2017, 2018; D. A. Keith, Akçakaya, and Murray 2018). Importantly, they relate the symptoms of ecosystem decline with the risk that an ecosystem will lose its defining features. The five criteria were designed to target different symptoms of ecosystem collapse (Figure 3). These symptoms are both distributional and functional:

- **Criterion A:** declines in distribution, which reduce carrying capacity for dependent biota;
- **Criterion B:** restricted distribution, which predisposes the system to spatially explicit threats;
- **Criterion C:** degradation of the abiotic environment, reducing habitat quality or abiotic niche diversity for component biota; and
- **Criterion D:** disruption of biotic processes and interactions

- **Criterion E:** allows for the integration of the above four symptoms into a simulation model of ecosystem dynamics to allow quantitative estimates of the risk of ecosystem collapse.

For further information on the criteria refer to the Guidelines for the application of IUCN Red List of Ecosystems Categories and Criteria (L. Bland et al. 2017).

Categories

Applying thresholds (decision rules) for each of the IUCN RLE criteria enables each ecosystem to be assigned to a category of risk ('status'). An ecosystem assessed under the RLE criteria can be placed into eight categories: Collapsed (CO), Critically Endangered (CR), Endangered (EN), Vulnerable (VU), Near Threatened (NT), Least Concern (LC), Data Deficient (DD), and Not Evaluated (NE, Figure 2).

The first six categories (CO, CR, EN, VU, NT and LC) are ordered in decreasing risk of collapse. The categories Data Deficient and Not Evaluated do not indicate a level of risk. For further details of the categories refer to the Guidelines for the application of IUCN Red List of Ecosystems Categories and Criteria (L. Bland et al. 2017). We applied version 2.2 of the IUCN Red List of Ecosystems Criteria (Table 1.1).

 TODO: Add Table 1.1

The IUCN Red List of Ecosystems Criteria, Version 2.2. Source: (L. Bland et al. 2017).

Assessment process

Application of the IUCN Red List of Ecosystems Categories and Criteria follows a generic sequential process that includes:

- Adapting the newly developed global ecosystem typology (D. A. Keith, Ferrer-Paris, Nicholson, Bishop, et al. 2020; D. Keith et al. 2022) to the area of assessment (Myanmar's terrestrial environment, Figure 1). This process is guided by experts and the result is a list of ecosystem types for the area of assessment that will be assessed under the RLE protocol;
- Describing each of the ecosystem types in ecosystem typology following the standard approach detailed in the Guidelines for the application of IUCN Red List of Ecosystems Categories and Criteria (L. Bland et al. 2017);
- If no map data is available to support the assessment, an ecosystem mapping project is required to support the ecosystem descriptions and assessment of several of the RLE criteria;

- Applying the assessment criteria to each ecosystem type, which requires extensive data searches and analyses. The outcome of each ecosystem assessment consists of a status of the ecosystem under 5 criteria and 18 subcriteria of the IUCN Red List of Ecosystems categories and criteria;
- Compiling the results into a comprehensive IUCN Red List of Ecosystems for the area of assessment (this report), which describes each ecosystem and identifies ecosystems according to their risk of collapse.

This report details each of these steps in the following sections.

Terrestrial Ecosystems of Ruritania

The Ruritania ecosystem typology includes 161 ecosystem types across 21 biomes (Table 1). The ecosystem typology is consistent with the IUCN global ecosystem typology (D. A. Keith, Ferrer-Paris, Nicholson, and Kingsford 2020) to support crosswalks and comparisons with Red Lists of ecosystems in other countries and regions. The Ruritania ecosystem typology has a hierarchical structure, where each ecosystem type is assigned to a realm, biome and functional group (ecotype). Realm refers to one of the four component media in the biosphere (Figure 1), biome is the segment of the biosphere united by major functional traits and macro-environmental features (Figure 2), and functional group is a group of related ecosystems within a biome (Figure 3). The distribution of Ruritania’s ecosystem types is shown in Figure 4, and map data is made publicly available ([reference?](#)).

Table 1: List of the terrestrial ecosystems of Ruritania developed in this project.

ECO_CODE	COD	ECO_NAME	Glob_Typol
REALM: Terrestre (T)			
Bioma de bosques tropicales y subtropicales (T1)			
Bosques lluviosos tropicales y subtropicales de tierras bajas (T1.1)			
T1.1.1	B1b	Apure-Villavicencio dry forests	T1.1_Apure_Vill f_B1b
T1.1.2	B1a	Caqueta moist forests	T1.1_Caq mf_B1a
T1.1.3	B2a1	Caqueta moist forests	T1.1_Caq mf_B2a1
T1.1.4	B3a1	Caqueta moist forests	T1.1_Caq mf_B3a1
T1.1.5	B1d	Catatumbo moist forests	T1.1_Catatum mf_B1d
T1.1.6	B2d	Catatumbo moist forests	T1.1_Catatum mf_B2d
T1.1.7	B2c	Chocó-Darién moist forests	T1.1_ChoDar_mf_B2c
T1.1.8	B3c	Chocó-Darién moist forests	T1.1_ChoDar_mf_B3c
T1.1.9	B7	Chocó-Darién moist forests	T1.1_ChoDar_mf_B7
T1.1.10	B4a	Guaianan piedmont forestsi	T1.1_GuaiPiedm f_B4a
T1.1.11	B6	Guaianan piedmont forestsi	T1.1_GuaiPiedm f_B6
T1.1.12	B4b	Japurá-Solimões-Negro moist forests	T1.1_Jap-So-Ne mf_B4b
T1.1.13	B4c	Japurá-Solimões-Negro moist forests	T1.1_Jap-So-Ne mf_B4c
T1.1.14	B4d	Japurá-Solimões-Negro moist forests	T1.1_Jap-So-Ne mf_B4d
T1.1.15	B5	Llanos	T1.1_Llanos_B5
T1.1.16	B1c	Magdalena-Urabá moist forests	T1.1_Mag-Urab mf_B1c
T1.1.17	B2b	Magdalena-Urabá moist forests	T1.1_Mag-Urab mf_B2b

Table 1: List of the terrestrial ecosystems of Ruritania developed in this project.

ECO_CODE	COD	ECO_NAME	Glob_Typol
T1.1.18	B8	Magdalena-Urabá moist forests	T1.1_Mag-Urab mf_B8
T1.1.19	B9	Magdalena-Urabá moist forests	T1.1_Mag-Urab mf_B9
T1.1.20	B1a	Napo moist forests	T1.1_Napo mf_B1a
T1.1.21	B2a1	Napo moist forests	T1.1_Napo mf_B2a1
T1.1.22	B3a1	Napo moist forests	T1.1_Napo mf_B3a1
T1.1.23	B1a	Solimões-Japurá moist forests	T1.1_SolJapura mf_B1a
T1.1.24	B2a2	Solimões-Japurá moist forests	T1.1_SolJapura mf_B2a2
T1.1.25	B3a2	Solimões-Japurá moist forests	T1.1_SolJapura mf_B3a2
T1.1.26	B2c	Western Ecuador moist forests	T1.1_WEcua mf_B2c
T1.1.27	B3c	Western Ecuador moist forests	T1.1_WEcua mf_B3c
Bosques y matorrales secos tropicales/subtropicales (T1.2)			
T1.2.1	B10	Catatumbo Ocaña dry forests	T1.2_CatOcañ df_B10
T1.2.2	B12	Catatumbo Ocaña dry forests	T1.2_CatOcañ df_B12
T1.2.3	B23	Catatumbo Ocaña dry forests	T1.2_CatOcañ df_B23
T1.2.4	B10	Cauca Valley dry forests	T1.2_Cauca Val df_B10
T1.2.5	B11	Cauca Valley dry forests	T1.2_Cauca Val df_B11
T1.2.6	B12	Cauca Valley dry forests	T1.2_Cauca Val df_B12
T1.2.7	B23	Cauca Valley dry forests	T1.2_Cauca Val df_B23
T1.2.8	A5	Chicamocha Valley dry forests	T1.2_ChV df_A5
T1.2.9	B23	Chicamocha Valley dry forests	T1.2_ChV df_B23
T1.2.10	B23	Cordillera Oriental montane forests	T1.2_CordOri mf_B23
T1.2.11	B10	Guajira-Barranquilla xeric scrub	T1.2_Guaj-Barr xs_B10
T1.2.12	B11	Guajira-Barranquilla xeric scrub	T1.2_Guaj-Barr xs_B11
T1.2.13	B12	Guajira-Barranquilla xeric scrub	T1.2_Guaj-Barr xs_B12
T1.2.14	B13	Guajira-Barranquilla xeric scrub	T1.2_Guaj-Barr xs_B13
T1.2.15	B10	Magdalena Valley dry forests	T1.2_Magd Val df_B10
T1.2.16	B11	Magdalena Valley dry forests	T1.2_Magd Val df_B11
T1.2.17	B12	Magdalena Valley dry forests	T1.2_Magd Val df_B12
T1.2.18	B23	Magdalena Valley dry forests	T1.2_Magd Val df_B23
T1.2.19	A5	Patia-Dagua dry forests	T1.2_Pat Dag df_A5
T1.2.20	B10	Patia-Dagua dry forests	T1.2_Pat Dag df_B10
T1.2.21	B11	Patia-Dagua dry forests	T1.2_Pat Dag df_B11
T1.2.22	B12	Patia-Dagua dry forests	T1.2_Pat Dag df_B12
T1.2.23	B23	Patia-Dagua dry forests	T1.2_Pat Dag df_B23
T1.2.24	B10	Sinú Valley dry forests	T1.2_Sinú Valley df_B10
T1.2.25	B11	Sinú Valley dry forests	T1.2_Sinú Valley df_B11
T1.2.26	B12	Sinú Valley dry forests	T1.2_Sinú Valley df_B12
T1.2.27	B23	Sinú Valley dry forests	T1.2_Sinú Valley df_B23
T1.2.28	A6	Magdalena Valley montane forests	T1.2a_Magd Val mf_A6

Table 1: List of the terrestrial ecosystems of Ruritania developed in this project.

ECO_CODE	COD	ECO_NAME	Glob_Typol
Bosques lluviosos montanos tropicales/subtropicales (T1.3)			
T1.3.1	B19b	Catatumbo moist forests	T1.3_Catatum mf_B19b
T1.3.2	B20b	Catatumbo moist forests	T1.3_Catatum mf_B20b
T1.3.3	B19a	Cauca Valley montane forests	T1.3_Cauca V mf_B19a
T1.3.4	B20a	Cauca Valley montane forests	T1.3_Cauca V mf_B20a
T1.3.5	B20b	Cauca Valley montane forests	T1.3_Cauca V mf_B20b
T1.3.6	B21a	Cauca Valley montane forests	T1.3_Cauca V mf_B21a
T1.3.7	B21b	Cauca Valley montane forests	T1.3_Cauca V mf_B21b
T1.3.8	B21d	Cauca Valley montane forests	T1.3_Cauca V mf_B21d
T1.3.9	B19a	Cordillera Oriental montane forests	T1.3_CordOri mf_B19a
T1.3.10	B19b	Cordillera Oriental montane forests	T1.3_CordOri mf_B19b
T1.3.11	B20a	Cordillera Oriental montane forests	T1.3_CordOri mf_B20a
T1.3.12	B20b	Cordillera Oriental montane forests	T1.3_CordOri mf_B20b
T1.3.13	B21a	Cordillera Oriental montane forests	T1.3_CordOri mf_B21a
T1.3.14	B21b	Cordillera Oriental montane forests	T1.3_CordOri mf_B21b
T1.3.15	B21c	Cordillera Oriental montane forests	T1.3_CordOri mf_B21c
T1.3.16	B21d	Cordillera Oriental montane forests	T1.3_CordOri mf_B21d
T1.3.17	B19a	Eastern Cordillera Real montane forests	T1.3_EastCR mf_B19a
T1.3.18	B20a	Eastern Cordillera Real montane forests	T1.3_EastCR mf_B20a
T1.3.19	B21a	Eastern Cordillera Real montane forests	T1.3_EastCR mf_B21a
T1.3.20	B21b	Eastern Cordillera Real montane forests	T1.3_EastCR mf_B21b
T1.3.21	B22	Eastern Cordillera Real montane forests	T1.3_EastCR mf_B22
T1.3.22	B16	Macarena montane forests	T1.3_Macarena mf_B16
T1.3.23	B17	Macarena montane forests	T1.3_Macarena mf_B17
T1.3.24	B18	Macarena montane forests	T1.3_Macarena mf_B18
T1.3.25	B19a	Magdalena Valley montane forests	T1.3_Magd Val mf_B19a
T1.3.26	B19b	Magdalena Valley montane forests	T1.3_Magd Val mf_B19b
T1.3.27	B20a	Magdalena Valley montane forests	T1.3_Magd Val mf_B20a
T1.3.28	B20b	Magdalena Valley montane forests	T1.3_Magd Val mf_B20b
T1.3.29	B21a	Magdalena Valley montane forests	T1.3_Magd Val mf_B21a
T1.3.30	B21b	Magdalena Valley montane forests	T1.3_Magd Val mf_B21b
T1.3.31	B21c	Magdalena Valley montane forests	T1.3_Magd Val mf_B21c
T1.3.32	B21d	Magdalena Valley montane forests	T1.3_Magd Val mf_B21d
T1.3.33	B22	Magdalena Valley montane forests	T1.3_Magd Val mf_B22
T1.3.34	B19a	Northwest Andean montane forests	T1.3_NW And mf_B19a
T1.3.35	B19b	Northwest Andean montane forests	T1.3_NW And mf_B19b
T1.3.36	B20a	Northwest Andean montane forests	T1.3_NW And mf_B20a
T1.3.37	B20b	Northwest Andean montane forests	T1.3_NW And mf_B20b
T1.3.38	B21a	Northwest Andean montane forests	T1.3_NW And mf_B21a

Table 1: List of the terrestrial ecosystems of Ruritania developed in this project.

ECO_CODE	COD	ECO_NAME	Glob_Typol
T1.3.39	B21b	Northwest Andean montane forests	T1.3_NW And mf_B21b
T1.3.40	B21c	Northwest Andean montane forests	T1.3_NW And mf_B21c
T1.3.41	B21d	Northwest Andean montane forests	T1.3_NW And mf_B21d
T1.3.42	B19b	Santa Marta montane forests	T1.3_S Mart mf_B19b
T1.3.43	B20b	Santa Marta montane forests	T1.3_S Mart mf_B20b
T1.3.44	B21b	Santa Marta montane forests	T1.3_S Mart mf_B21b
Bosques tropicales esclerofilos sobre arenas blancas (T1.4)			
T1.4.1	B14	Japurá-Solimões-Negro moist forests	T1.4_Jap-So-Ne mf_B14
T1.4.2	B15	Negro-Branco moist forests	T1.4_Neg Branc mf_B15
T1.4.3	S1	Rio Negro Campinas	T1.4a_Negro camp_S1
Bioma de arbustales y matorrales (T3)			
Matorrales tropicales estacionalmente secos (T3.1)			
T3.1.1	A4	Pantepui	T3.1_Pantepui_A4
T3.1.2	S2	Pantepui	T3.1_Pantepui_S2
Bioma de sabanas y pastizales (T4)			
Sabanas de pastos piricos (T4.2)			
T4.2.1	S16	Chicamocha Valley dry forests	T4.2_ChV df_S16
T4.2.2	S5	Llanos	T4.2_Llanos_S5
T4.2.3	S3	Llanos	T4.2_Llanos_S3
T4.2.4	S4	Llanos	T4.2_Llanos_S4
T4.2.5	S5	Llanos	T4.2_Llanos_S5
T4.2.6	S6	Llanos	T4.2_Llanos_S6
T4.2.7	S7	Llanos	T4.2_Llanos_S7
T4.2.8	S8	Llanos	T4.2_Llanos_S8
T4.2.9	S11	Magdalena-Urabá moist forests	T4.2_Mag-Urab mf_S11
T4.2.10	S10	Llanos	T4.2a_Llanos_S10
T4.2.11	S9	Llanos	T4.2a_Llanos_S9
T4.2.12	S12	Sinú Valley dry forests	T4.2b_Sinú Valley df_S12
Bioma de desiertos y semidesiertos (T5)			
Desiertos y semidesiertos suculentos o espinosos (T5.2)			
T5.2.1	A1	Guajira-Barranquilla xeric scrub	T5.2_Guaj-Barr xs_A1
T5.2.2	A3	Guajira-Barranquilla xeric scrub	T5.2_Guaj-Barr xs_A3
T5.2.3	A2	Magdalena Valley dry forests	T5.2_Magd Val df_A2
Bioma polar/alpino (criogenico) (T6)			
Pastizales y herbazales de altas montanas tropicales (T6.5)			
T6.5.1	N	Northern Andean Paramo	T6.5_N And par_N
T6.5.2	S13	Northern Andean Paramo	T6.5_N And par_S13
T6.5.3	S14	Northern Andean Paramo	T6.5_N And par_S14
T6.5.4	S15	Northern Andean Paramo	T6.5_N And par_S15

Table 1: List of the terrestrial ecosystems of Ruritania developed in this project.

ECO_CODE	COD	ECO_NAME	Glob_Typol
T6.5.5	N	Santa Marta Paramo	T6.5_SM Paramo_N
T6.5.6	S14	Santa Marta Paramo	T6.5_SM Paramo_S14
REALM: Agua dulce (F)			
Bioma de rios y arroyos (F1)			
Rios permanentes de tierras altas (F1.1)			
F1.1.1	Agua	Catatumbo moist forests	F1.1_Catatum mf_Agua
F1.1.2	Agua	Cauca Valley montane forests	F1.1_Cauca V mf_Agua
F1.1.3	Agua	INDEF	F1.1_Indef_Agua
F1.1.4	Agua	Magdalena-Urabá moist forests	F1.1_Mag-Urab mf_Agua
F1.1.5	Agua	Magdalena Valley montane forests	F1.1_Magd Val mf_Agua
F1.1.6	Agua	Northwest Andean montane forests	F1.1_NW And mf_Agua
Rios permanentes de tierras bajas (F1.2)			
F1.2.1	Agua	INDEF	F1.2_Indef_Agua
F1.2.2	Agua	INDEF	F1.2a_Indef_Agua
Bioma de lagos (F2)			
Grandes lagos de agua dulce permanentes (F2.1)			
F2.1.1	Agua	INDEF	F2.1_Indef_Agua
Pequenos lagos de agua dulce permanentes (F2.2)			
F2.2.1	Agua	INDEF	F2.2_Indef_Agua
Lagos de agua dulce estacionales (F2.3)			
F2.3.1	Agua	INDEF	F2.3_Indef_Agua
Bioma de humedales artificiales (F3)			
Grandes embalses (F3.1)			
F3.1.1	Agua	INDEF	F3.1_Indef_Agua
REALM: Agua dulce-Marina (FM)			
Bioma de aguas de transicion semiconfinadas (FM1)			
Estuarios fluviales y bahias permanentemente conectados (FM1.2)			
FM1.2.1	Agua	INDEF	FM1.2_Indef_Agua
REALM: Marina-Agua dulce-Terrestre (MFT)			
Bioma de mareas salobres (MFT1)			
Deltas fluviales costeros (MFT1.1)			
MFT1.1.1	B31	Chocó-Darién moist forests	MFT1.1_ChoDar_mf_B31
MFT1.1.2	B32	Chocó-Darién moist forests	MFT1.1_ChoDar_mf_B32
MFT1.1.3	B32	Western Ecuador moist forests	MFT1.1_WEcua mf_B32
Bosques y matorrales intermareales (MFT1.2)			
MFT1.2.1	B36	Amazon-Orinoco-Southern Caribbean mangroves	MFT1.2_AmOriSCar_mg_B36
MFT1.2.2	B35	South American Pacific mangroves	MFT1.2_SAmPacif mg_B35
REALM: Terrestre-Agua dulce (TF)			
Bioma de humedales palustres (TF1)			

Table 1: List of the terrestrial ecosystems of Ruritania developed in this project.

ECO_CODE	COD	ECO_NAME	Glob_Typol
Bosques tropicales inundados y bosques de turba (TF1.1)			
TF1.1.1	B27	Amazon-Orinoco Alluvial forests	TF1.1_Am-Ori alvf_B27
TF1.1.2	B28	Amazon-Orinoco Alluvial forests	TF1.1_Am-Ori alvf_B28
TF1.1.3	B29	Amazon-Orinoco Alluvial forests	TF1.1_Am-Ori alvf_B29
TF1.1.4	B30	Amazon-Orinoco Alluvial forests	TF1.1_Am-Ori alvf_B30
TF1.1.5	B24	Andean Alluvial forests	TF1.1_And alvf_B24
TF1.1.6	B25	Andean Alluvial forests	TF1.1_And alvf_B25
TF1.1.7	B33	Andean Alluvial forests	TF1.1_And alvf_B33
TF1.1.8	B24	Andean Pacific Alluvial forests	TF1.1_AndPac alvf_B24
TF1.1.9	B26	Andean Pacific Alluvial forests	TF1.1_AndPac alvf_B26
Pantanos permanentes (TF1.3)			
TF1.3.1	P1	Eastern Cordillera Real montane forests	TF1.3_EastCR mf_P1
TF1.3.2	P1	Magdalena Valley montane forests	TF1.3_Magd Val mf_P1
TF1.3.3	P1	Northwest Andean montane forests	TF1.3_NW And mf_P1
Ciénagas estacionales de llanuras de inundacion (TF1.4)			
TF1.4.1	P4	Chocó-Darién moist forests	TF1.4_ChoDar_mf_P4
TF1.4.2	P2	Sinu-Magdalena wetlands	TF1.4_Sin-Mag w_P2
TF1.4.3	P3	Sinu-Magdalena wetlands	TF1.4_Sin-Mag w_P3
Llanuras de inundacion aridas episodicas (TF1.5)			
TF1.5.1	B34	Guajira-Barranquilla xeric scrub	TF1.5_Guaj-Barr xs_B34
TF1.5.2	B34	Magdalena Valley dry forests	TF1.5_Magd Val df_B34
TF1.5.3	B34	Sinú Valley dry forests	TF1.5_Sinú Valley df_B34

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/iucn_get_data/ecosystem_map.py:435: UserWarning: Tight layout not applied. The bott
fig.tight_layout(rect=(0, 0.18, 1, 1))
```

! How to customize the Ecosystem Overview page

- To modify the ecosystem overview text, edit the `content/2_ecosystem_overview.qmd` file.
- To remove the ecosystem overview section, edit the `_quarto.yml` file and remove the `- content/2_ecosystem_overview.qmd` line.

To hide these instructions on all pages, set `show-instructions: false` in the `_quarto.yml` file.

Realm



- F — Freshwater
- FM — Freshwater-Marine
- MFT —
Marine-Freshwater-Terrestrial
- T — Terrestrial
- TF — Terrestrial-Freshwater

Figure 1: The distribution of the realms of Ruritania.

Biome














-  F1 — Rivers and streams
-  F2 — Lakes
-  F3 — Artificial wetlands
-  FM1 — Semi-confined transitional waters
-  MFT1 — Brackish tidal
-  T1 — Tropical-subtropical forests
-  T3 — Shrublands and shrubby woodlands
-  T4 — Savannas and grasslands
-  T5 — Deserts and semi-deserts
-  T6 — Polar/alpine (cryogenic)
-  TF1 — Palustrine wetlands

Figure 2: The distribution of the biomes of Ruritania.

Functional Group



- F1.1 — Permanent upland streams
- F1.2 — Permanent lowland rivers
- F2.1 — Large permanent freshwater lakes
- F2.2 — Small permanent freshwater lakes
- F2.3 — Seasonal freshwater lakes
- F3.1 — Large reservoirs
- FM1.2 — Permanently open riverine estuaries and bays
- MFT1.1 — Coastal river deltas
- MFT1.2 — Intertidal forests and shrublands
- T1.1 — Tropical/Subtropical lowland rainforests
- T1.2 — Tropical/Subtropical dry forests and thickets
- T1.3 — Tropical/Subtropical montane rainforests
- T1.4 — Tropical heath forests
- T3.1 — Seasonally dry tropical shrublands
- T4.2 — Pyric tussock savannas
- T5.2 — Succulent or Thorny deserts and semi-deserts
- T6.5 — Tropical alpine grasslands and herbfields
- TF1.1 — Tropical flooded forests and peat forests
- TF1.3 — Permanent marshes
- TF1.4 — Seasonal floodplain marshes
- TF1.5 — Episodic arid floodplains

Figure 3: The distribution of the biomes of Ruritania.

(Heat map of the distribution of natural ecosystem types not implemented yet.)

Figure 4

Ecosystem description and assessment

Ecosystem accounts provided in this section consist of two components, an ecosystem description and a detailed summary of the application of the IUCN Red List of Ecosystem criteria. The ecosystem description follows the standard format suggested by the IUCN. Ecosystem descriptions are an essential component of IUCN Red List of Ecosystems assessments and serve the purpose of clearly defining each assessment unit. This allows repeated application of the categories and criteria to a single defined unit and supports red listing and cross-walking at the global scale. The associated assessment section provides the details the Red List assessments, including the data sources, methods of analysis, evidence statements and key references.

Table 1: Components of ecosystem description

Component	Description
Authors	Authors of the description and the assessment.
Myanmar ecosystem names	Alternative names of the ecosystem.
Biome	Biome membership according to the global ecosystem typology.
Functional group	Functional group membership according to the global ecosystem typology.
Global classification	Classification code according to the global ecosystem typology.
Description	General overview of the principal components and dynamics of the ecosystem. Includes a photograph of the ecosystem.
Distribution	Short written description and range map of the spatial distribution of the ecosystem.
Characteristic native biota	Identifies the defining biotic features of the ecosystem, including diagnostic native taxa, functional components of the characteristic biota.
Abiotic environment	Identifies the defining abiotic features of the ecosystem, including descriptions of the characteristic states or summary of values of the key abiotic variables.
Key processes and interactions	Describes the key ecosystem drivers and interactions among biota and the abiotic environment.

Component	Description
Major threats	Short summary of the major threats and impacts to the ecosystem.
Ecosystem collapse definition	Short description of the collapsed state of the ecosystem and any associated thresholds.
Assessment summary	Short summary of the red list of ecosystems assessment.
Assessment outcome	Short format assessed status of the ecosystem.
IUCN Red List of Ecosystems Categories and Criteria	Version of the IUCN Red List of Ecosystems criteria used in the assessment.
Year published	Publication year.
Date assessed	The date the ecosystem assessment was completed.
Assessment credits	Names the authors, reviewers and contributors to the assessment.
Assessment summary	Short summary of the red list of ecosystems assessment.
Criterion A	Description of data, analysis and methods used to assess the criterion and the status outcome.
Criterion B	Description of data, analysis and methods used to assess the criterion and the status outcome.
Criterion C	Description of data, analysis and methods used to assess the criterion and the status outcome.
Criterion D	Description of data, analysis and methods used to assess the criterion and the status outcome.
Criterion E	Description of data, analysis and methods used to assess the criterion and the status outcome.

T1.1.6 - Catatumbo moist forests

Assessment

Catatumbo moist forests

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T1.1

Global classification: T1.1.6

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 77 (10 km x 10 km) grid cells . EOO was measured as 9324 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.  
# This line is replaced by build_ecosystem_pages.py for each ecosystem.  
ecosystem_code = 'T1.1.6'
```

Import Python modules.

```
import os  
import yaml  
from pathlib import Path  
from lonboard import Map  
from rle_python_gee.ecosystems import Ecosystems  
from rle_python_gee.eoo import make_eoo  
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))  
config_path = Path(project_root) / 'config' / 'country_config.yaml'  
with open(config_path) as f:  
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']  
ecosystems = Ecosystems.from_file(  
    source['data'],  
    ecosystem_column=source['ecosystem_code_column'],  
    ecosystem_name_column=source['ecosystem_name_column'],  
    functional_group_column=source['functional_group_column']  
)
```

Filter by the T1.1.6 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 128
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7fa470fbb590>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 9323.9 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```

```

if has_data:
    cmap = LinearSegmentedColormap.from_list("white_red", ["white", "red"])
    values = aoo_grid_filtered.grid_cells[ecosystem_column].values
    normalized = (values - values.min()) / (values.max() - values.min())
    colors = apply_continuous_cmap(normalized, cmap)
    display(smart_map([(aoo_grid_filtered, {"get_fill_color": colors}), ecosystem]))

```

```

/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
warnings.warn(

```

```

VBox(children=(<lonboard._map.Map object at 0x7fa470e0e9d0>, VBox(children=(ErrorOutput(), E

```

Calculate grid cell area and total area

- (2) Calculate extent of the ecosystem type in each grid cell (area) and sum these areas to obtain the total ecosystem area (total area).

```

if has_data:
    keep = ['geometry', 'grid_col', 'grid_row', ecosystem_column]
    gdf = aoo_grid_filtered.grid_cells[keep]
    display(gdf)

```

	geometry	grid_col	grid_row	T1_1_6
0	POLYGON ((-73.30253 8.8061, -73.30253 8.89759,...	-817	97	0.029385
1	POLYGON ((-73.30253 8.89759, -73.30253 8.98911...	-817	98	0.118214
2	POLYGON ((-73.30253 8.98911, -73.30253 9.08066...	-817	99	0.126984
3	POLYGON ((-73.30253 9.08066, -73.30253 9.17222...	-817	100	0.004873
4	POLYGON ((-73.2127 8.44032, -73.2127 8.53173, ...	-816	93	0.046072
...
90	POLYGON ((-72.40421 8.44032, -72.40421 8.53173...	-807	93	0.008839
91	POLYGON ((-72.31438 7.98358, -72.31438 8.07488...	-806	88	0.197302
92	POLYGON ((-72.31438 8.07488, -72.31438 8.16621...	-806	89	0.246987
93	POLYGON ((-72.31438 8.25756, -72.31438 8.34893...	-806	91	0.023974
94	POLYGON ((-72.31438 8.34893, -72.31438 8.44032...	-806	92	0.012926

The column T1_1_6 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```

if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)

```

```
np.float64(30.072950569410207)
```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```

if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)

```

	geometry	grid_col	grid_row	T1_1_6	cumulative_area
84	POLYGON ((-72.40421 7.89229, -72.40421 7.98358...	-807	87	0.000055	0.000055
71	POLYGON ((-72.67371 9.08066, -72.67371 9.17222...	-810	100	0.000075	0.000130
29	POLYGON ((-73.03303 9.26381, -73.03303 9.35542...	-814	102	0.000916	0.001046
11	POLYGON ((-73.2127 9.17222, -73.2127 9.26381, ...	-816	101	0.001589	0.002634
30	POLYGON ((-72.9432 8.34893, -72.9432 8.44032, ...	-813	92	0.002880	0.005514
...
67	POLYGON ((-72.67371 8.71462, -72.67371 8.8061,...	-810	96	0.847337	26.148114
66	POLYGON ((-72.67371 8.62317, -72.67371 8.71462...	-810	95	0.924837	27.072951
57	POLYGON ((-72.76354 8.71462, -72.76354 8.8061,...	-811	96	1.000000	28.072951
58	POLYGON ((-72.76354 8.8061, -72.76354 8.89759,...	-811	97	1.000000	29.072951
59	POLYGON ((-72.76354 8.89759, -72.76354 8.98911...	-811	98	1.000000	30.072951

Calculate cumulative proportion

- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)

```

if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)

```

	geometry	grid_col	grid_row	T1_1_6	cumulative_area
84	POLYGON ((-72.40421 7.89229, -72.40421 7.98358...	-807	87	0.000055	0.000055
71	POLYGON ((-72.67371 9.08066, -72.67371 9.17222...	-810	100	0.000075	0.000130
29	POLYGON ((-73.03303 9.26381, -73.03303 9.35542...	-814	102	0.000916	0.001046
11	POLYGON ((-73.2127 9.17222, -73.2127 9.26381, ...	-816	101	0.001589	0.002634
30	POLYGON ((-72.9432 8.34893, -72.9432 8.44032, ...	-813	92	0.002880	0.005514
...
67	POLYGON ((-72.67371 8.71462, -72.67371 8.8061,...	-810	96	0.847337	26.148114
66	POLYGON ((-72.67371 8.62317, -72.67371 8.71462...	-810	95	0.924837	27.072951
57	POLYGON ((-72.76354 8.71462, -72.76354 8.8061,...	-811	96	1.000000	28.072951
58	POLYGON ((-72.76354 8.8061, -72.76354 8.89759,...	-811	97	1.000000	29.072951
59	POLYGON ((-72.76354 8.89759, -72.76354 8.98911...	-811	98	1.000000	30.072951

Count AOO cells

- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'A00 is {aoo} cells')
```

A00 is 77 cells

AOO Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'A00: {aoo_count} grid cells')
```

A00: 77 grid cells

```
if has_data:
    display(smart_map([aoo_grid_filtered, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geowarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
  warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7fa40022e950>, VBox(children=(ErrorOutput(), E
```

Criterion B Summary

Ecosystem Code	Ecosystem Name	EOO	AOO
T1.1.6	Catatumbo moist forests	9324 km ²	77 cells

T1.2.1 - Catatumbo Ocaña dry forests

Assessment

Catatumbo Ocaña dry forests

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T1.2

Global classification: T1.2.1

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 21 (10 km x 10 km) grid cells . EOO was measured as 2012 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.
# This line is replaced by build_ecosystem_pages.py for each ecosystem.
ecosystem_code = 'T1.2.1'
```

Import Python modules.

```
import os
import yaml
from pathlib import Path
from lonboard import Map
from rle_python_gee.ecosystems import Ecosystems
from rle_python_gee.eoo import make_eoo
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))
config_path = Path(project_root) / 'config' / 'country_config.yaml'
with open(config_path) as f:
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']
ecosystems = Ecosystems.from_file(
    source['data'],
    ecosystem_column=source['ecosystem_code_column'],
    ecosystem_name_column=source['ecosystem_name_column'],
    functional_group_column=source['functional_group_column']
)
```

Filter by the T1.2.1 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 84
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f7eac6cd750>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 2011.7 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```

```

if has_data:
    cmap = LinearSegmentedColormap.from_list("white_red", ["white", "red"])
    values = aoo_grid_filtered.grid_cells[ecosystem_column].values
    normalized = (values - values.min()) / (values.max() - values.min())
    colors = apply_continuous_cmap(normalized, cmap)
    display(smart_map([(aoo_grid_filtered, {"get_fill_color": colors}), ecosystem]))

```

```

/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.11/
packages/lonboard/_gearrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPSG:4326
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(

```

```

VBox(children=(<lonboard._map.Map object at 0x7f7e58c02f50>, VBox(children=(ErrorOutput(), ErrorOutput()),

```

Calculate grid cell area and total area

- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).

```

if has_data:
    keep = ['geometry', 'grid_col', 'grid_row', ecosystem_column]
    gdf = aoo_grid_filtered.grid_cells[keep]
    display(gdf)

```

	geometry	grid_col	grid_row	T1	2	1
0	POLYGON ((-72.76354 7.98358, -72.76354 8.07488...	-811	88	0.001224		
1	POLYGON ((-72.76354 8.07488, -72.76354 8.16621...	-811	89	0.057771		
2	POLYGON ((-72.67371 7.80102, -72.67371 7.89229...	-810	86	0.047898		
3	POLYGON ((-72.67371 7.89229, -72.67371 7.98358...	-810	87	0.032588		
4	POLYGON ((-72.67371 7.98358, -72.67371 8.07488...	-810	88	0.008510		
5	POLYGON ((-72.67371 8.07488, -72.67371 8.16621...	-810	89	0.005267		
6	POLYGON ((-72.58387 7.61854, -72.58387 7.70977...	-809	84	0.024333		
7	POLYGON ((-72.58387 7.80102, -72.58387 7.89229...	-809	86	0.124427		
8	POLYGON ((-72.58387 7.89229, -72.58387 7.98358...	-809	87	0.348859		
9	POLYGON ((-72.58387 7.98358, -72.58387 8.07488...	-809	88	0.314078		
10	POLYGON ((-72.58387 8.07488, -72.58387 8.16621...	-809	89	0.057531		
11	POLYGON ((-72.58387 8.16621, -72.58387 8.25756...	-809	90	0.021904		
12	POLYGON ((-72.49404 7.61854, -72.49404 7.70977...	-808	84	0.009261		

	geometry	grid_col	grid_row	T1_2_1
13	POLYGON ((-72.49404 7.70977, -72.49404 7.80102...	-808	85	0.170308
14	POLYGON ((-72.49404 7.80102, -72.49404 7.89229...	-808	86	0.259769
15	POLYGON ((-72.49404 7.89229, -72.49404 7.98358...	-808	87	0.043424
16	POLYGON ((-72.49404 7.98358, -72.49404 8.07488...	-808	88	0.205809
17	POLYGON ((-72.49404 8.07488, -72.49404 8.16621...	-808	89	0.219096
18	POLYGON ((-72.49404 8.16621, -72.49404 8.25756...	-808	90	0.046407
19	POLYGON ((-72.40421 7.61854, -72.40421 7.70977...	-807	84	0.006006
20	POLYGON ((-72.40421 7.70977, -72.40421 7.80102...	-807	85	0.060013
21	POLYGON ((-72.40421 7.80102, -72.40421 7.89229...	-807	86	0.349297
22	POLYGON ((-72.40421 7.89229, -72.40421 7.98358...	-807	87	0.133761
23	POLYGON ((-72.40421 7.98358, -72.40421 8.07488...	-807	88	0.118924
24	POLYGON ((-72.31438 7.98358, -72.31438 8.07488...	-806	88	0.029849

The column T1_2_1 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```
if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)
```

```
np.float64(2.6963111050247246)
```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```
if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)
```

	geometry	grid_col	grid_row	T1_2_1	cumulative_area
0	POLYGON ((-72.76354 7.98358, -72.76354 8.07488...	-811	88	0.001224	0.001224
5	POLYGON ((-72.67371 8.07488, -72.67371 8.16621...	-810	89	0.005267	0.006491
19	POLYGON ((-72.40421 7.61854, -72.40421 7.70977...	-807	84	0.006006	0.012497

	geometry	grid_col	grid_row	T1_2_1	cumulative_area
4	POLYGON ((-72.67371 7.98358, -72.67371 8.07488...	-810	88	0.008510	0.021006
12	POLYGON ((-72.49404 7.61854, -72.49404 7.70977...	-808	84	0.009261	0.030267
11	POLYGON ((-72.58387 8.16621, -72.58387 8.25756...	-809	90	0.021904	0.052171
6	POLYGON ((-72.58387 7.61854, -72.58387 7.70977...	-809	84	0.024333	0.076504
24	POLYGON ((-72.31438 7.98358, -72.31438 8.07488...	-806	88	0.029849	0.106353
3	POLYGON ((-72.67371 7.89229, -72.67371 7.98358...	-810	87	0.032588	0.138940
15	POLYGON ((-72.49404 7.89229, -72.49404 7.98358...	-808	87	0.043424	0.182365
18	POLYGON ((-72.49404 8.16621, -72.49404 8.25756...	-808	90	0.046407	0.228772
2	POLYGON ((-72.67371 7.80102, -72.67371 7.89229...	-810	86	0.047898	0.276669
10	POLYGON ((-72.58387 8.07488, -72.58387 8.16621...	-809	89	0.057531	0.334200
1	POLYGON ((-72.76354 8.07488, -72.76354 8.16621...	-811	89	0.057771	0.391971
20	POLYGON ((-72.40421 7.70977, -72.40421 7.80102...	-807	85	0.060013	0.451984
23	POLYGON ((-72.40421 7.98358, -72.40421 8.07488...	-807	88	0.118924	0.570908
7	POLYGON ((-72.58387 7.80102, -72.58387 7.89229...	-809	86	0.124427	0.695335
22	POLYGON ((-72.40421 7.89229, -72.40421 7.98358...	-807	87	0.133761	0.829096
13	POLYGON ((-72.49404 7.70977, -72.49404 7.80102...	-808	85	0.170308	0.999404
16	POLYGON ((-72.49404 7.98358, -72.49404 8.07488...	-808	88	0.205809	1.205212
17	POLYGON ((-72.49404 8.07488, -72.49404 8.16621...	-808	89	0.219096	1.424309
14	POLYGON ((-72.49404 7.80102, -72.49404 7.89229...	-808	86	0.259769	1.684077
9	POLYGON ((-72.58387 7.98358, -72.58387 8.07488...	-809	88	0.314078	1.998156
8	POLYGON ((-72.58387 7.89229, -72.58387 7.98358...	-809	87	0.348859	2.347014
21	POLYGON ((-72.40421 7.80102, -72.40421 7.89229...	-807	86	0.349297	2.696311

Calculate cumulative proportion

- (4) Calculate cumulative proportion by dividing cumulative area by total area (cumulative proportion takes values between 0 and 1)

```
if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)
```

	geometry	grid_col	grid_row	T1_2_1	cumulative_area
0	POLYGON ((-72.76354 7.98358, -72.76354 8.07488...	-811	88	0.001224	0.001224
5	POLYGON ((-72.67371 8.07488, -72.67371 8.16621...	-810	89	0.005267	0.006491
19	POLYGON ((-72.40421 7.61854, -72.40421 7.70977...	-807	84	0.006006	0.012497
4	POLYGON ((-72.67371 7.98358, -72.67371 8.07488...	-810	88	0.008510	0.021006
12	POLYGON ((-72.49404 7.61854, -72.49404 7.70977...	-808	84	0.009261	0.030267

	geometry	grid_col	grid_row	T1_2_1	cumulative_area
11	POLYGON ((-72.58387 8.16621, -72.58387 8.25756...	-809	90	0.021904	0.052171
6	POLYGON ((-72.58387 7.61854, -72.58387 7.70977...	-809	84	0.024333	0.076504
24	POLYGON ((-72.31438 7.98358, -72.31438 8.07488...	-806	88	0.029849	0.106353
3	POLYGON ((-72.67371 7.89229, -72.67371 7.98358...	-810	87	0.032588	0.138940
15	POLYGON ((-72.49404 7.89229, -72.49404 7.98358...	-808	87	0.043424	0.182365
18	POLYGON ((-72.49404 8.16621, -72.49404 8.25756...	-808	90	0.046407	0.228772
2	POLYGON ((-72.67371 7.80102, -72.67371 7.89229...	-810	86	0.047898	0.276669
10	POLYGON ((-72.58387 8.07488, -72.58387 8.16621...	-809	89	0.057531	0.334200
1	POLYGON ((-72.76354 8.07488, -72.76354 8.16621...	-811	89	0.057771	0.391971
20	POLYGON ((-72.40421 7.70977, -72.40421 7.80102...	-807	85	0.060013	0.451984
23	POLYGON ((-72.40421 7.98358, -72.40421 8.07488...	-807	88	0.118924	0.570908
7	POLYGON ((-72.58387 7.80102, -72.58387 7.89229...	-809	86	0.124427	0.695335
22	POLYGON ((-72.40421 7.89229, -72.40421 7.98358...	-807	87	0.133761	0.829096
13	POLYGON ((-72.49404 7.70977, -72.49404 7.80102...	-808	85	0.170308	0.999404
16	POLYGON ((-72.49404 7.98358, -72.49404 8.07488...	-808	88	0.205809	1.205212
17	POLYGON ((-72.49404 8.07488, -72.49404 8.16621...	-808	89	0.219096	1.424309
14	POLYGON ((-72.49404 7.80102, -72.49404 7.89229...	-808	86	0.259769	1.684077
9	POLYGON ((-72.58387 7.98358, -72.58387 8.07488...	-809	88	0.314078	1.998156
8	POLYGON ((-72.58387 7.89229, -72.58387 7.98358...	-809	87	0.348859	2.347014
21	POLYGON ((-72.40421 7.80102, -72.40421 7.89229...	-807	86	0.349297	2.696311

Count AOO cells

- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'A00 is {aoo} cells')
```

A00 is 21 cells

A00 Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'A00: {aoo_count} grid cells')
```

AOO: 21 grid cells

```
if has_data:  
    display(smart_map([aoo_grid_filtered, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1  
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS  
Lonboard is only able to render data in EPSG:4326 projection.  
warnings.warn(  

```

```
VBox(children=(<lonboard._map.Map object at 0x7f7eac6cda10>, VBox(children=(ErrorOutput(), E
```

Criterion B Summary

Ecosystem Code	Ecosystem Name	EOO	AOO
T1.2.1	Catatumbo Ocaña dry forests	2012 km ²	21 cells

T1.2.19 - Patia-Dagua dry forests

Assessment

Patà-Dagua dry forests

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T1.2

Global classification: T1.2.19

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 28 (10 km x 10 km) grid cells . EOO was measured as 2926 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.
# This line is replaced by build_ecosystem_pages.py for each ecosystem.
ecosystem_code = 'T1.2.19'
```

Import Python modules.

```
import os
import yaml
from pathlib import Path
from lonboard import Map
from rle_python_gee.ecosystems import Ecosystems
from rle_python_gee.eoo import make_eoo
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))
config_path = Path(project_root) / 'config' / 'country_config.yaml'
with open(config_path) as f:
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']
ecosystems = Ecosystems.from_file(
    source['data'],
    ecosystem_column=source['ecosystem_code_column'],
    ecosystem_name_column=source['ecosystem_name_column'],
    functional_group_column=source['functional_group_column']
)
```

Filter by the T1.2.19 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 113
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f6d0f2bbbd0>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 2926.2 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```


	geometry	grid_col	grid_row	T1_2_19
13	POLYGON ((-77.43478 0.90441, -77.43478 0.99486...	-863	10	0.304329
14	POLYGON ((-77.43478 0.99486, -77.43478 1.08531...	-863	11	0.609044
15	POLYGON ((-77.43478 1.08531, -77.43478 1.17576...	-863	12	0.429132
16	POLYGON ((-77.43478 1.17576, -77.43478 1.26622...	-863	13	0.264533
17	POLYGON ((-77.34495 0.99486, -77.34495 1.08531...	-862	11	0.277829
18	POLYGON ((-77.34495 1.08531, -77.34495 1.17576...	-862	12	0.173535
19	POLYGON ((-77.34495 1.17576, -77.34495 1.26622...	-862	13	0.002867
20	POLYGON ((-77.34495 1.26622, -77.34495 1.35668...	-862	14	0.119695
21	POLYGON ((-77.25511 1.08531, -77.25511 1.17576...	-861	12	0.082091
22	POLYGON ((-77.25511 1.17576, -77.25511 1.26622...	-861	13	0.641881
23	POLYGON ((-77.25511 1.26622, -77.25511 1.35668...	-861	14	0.709896
24	POLYGON ((-77.25511 1.35668, -77.25511 1.44714...	-861	15	0.267268
25	POLYGON ((-77.25511 1.44714, -77.25511 1.53761...	-861	16	0.193222
26	POLYGON ((-77.16528 1.08531, -77.16528 1.17576...	-860	12	0.030285
27	POLYGON ((-77.16528 1.17576, -77.16528 1.26622...	-860	13	0.173905
28	POLYGON ((-77.16528 1.26622, -77.16528 1.35668...	-860	14	0.236492
29	POLYGON ((-77.16528 1.35668, -77.16528 1.44714...	-860	15	0.416053
30	POLYGON ((-77.16528 1.44714, -77.16528 1.53761...	-860	16	0.293780
31	POLYGON ((-77.07545 1.35668, -77.07545 1.44714...	-859	15	0.183151
32	POLYGON ((-77.07545 1.44714, -77.07545 1.53761...	-859	16	0.115833

The column T1_2_19 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```
if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)
```

```
np.float64(6.44686965700011)
```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (cumulative area).

```
if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)
```

	geometry	grid_col	grid_row	T1_2_19	cumulative_area
4	POLYGON ((-77.61444 0.90441, -77.61444 0.99486...	-865	10	0.001692	0.001692
19	POLYGON ((-77.34495 1.17576, -77.34495 1.26622...	-862	13	0.002867	0.004560
11	POLYGON ((-77.43478 0.72351, -77.43478 0.81396...	-863	8	0.010432	0.014992
0	POLYGON ((-77.7941 0.81396, -77.7941 0.90441, ...	-867	9	0.019752	0.034743
2	POLYGON ((-77.61444 0.72351, -77.61444 0.81396...	-865	8	0.022897	0.057640
10	POLYGON ((-77.52461 1.17576, -77.52461 1.26622...	-864	13	0.027672	0.085312
26	POLYGON ((-77.16528 1.08531, -77.16528 1.17576...	-860	12	0.030285	0.115597
9	POLYGON ((-77.52461 1.08531, -77.52461 1.17576...	-864	12	0.039459	0.155056
12	POLYGON ((-77.43478 0.81396, -77.43478 0.90441...	-863	9	0.046522	0.201578
7	POLYGON ((-77.52461 0.90441, -77.52461 0.99486...	-864	10	0.052011	0.253589
1	POLYGON ((-77.70427 0.81396, -77.70427 0.90441...	-866	9	0.063122	0.316711
21	POLYGON ((-77.25511 1.08531, -77.25511 1.17576...	-861	12	0.082091	0.398802
3	POLYGON ((-77.61444 0.81396, -77.61444 0.90441...	-865	9	0.094620	0.493423
8	POLYGON ((-77.52461 0.99486, -77.52461 1.08531...	-864	11	0.095877	0.589299
32	POLYGON ((-77.07545 1.44714, -77.07545 1.53761...	-859	16	0.115833	0.705133
20	POLYGON ((-77.34495 1.26622, -77.34495 1.35668...	-862	14	0.119695	0.824828
5	POLYGON ((-77.52461 0.72351, -77.52461 0.81396...	-864	8	0.157733	0.982561
18	POLYGON ((-77.34495 1.08531, -77.34495 1.17576...	-862	12	0.173535	1.156097
27	POLYGON ((-77.16528 1.17576, -77.16528 1.26622...	-860	13	0.173905	1.330002
31	POLYGON ((-77.07545 1.35668, -77.07545 1.44714...	-859	15	0.183151	1.513153
25	POLYGON ((-77.25511 1.44714, -77.25511 1.53761...	-861	16	0.193222	1.706375
28	POLYGON ((-77.16528 1.26622, -77.16528 1.35668...	-860	14	0.236492	1.942867
16	POLYGON ((-77.43478 1.17576, -77.43478 1.26622...	-863	13	0.264533	2.207400
24	POLYGON ((-77.25511 1.35668, -77.25511 1.44714...	-861	15	0.267268	2.474669
17	POLYGON ((-77.34495 0.99486, -77.34495 1.08531...	-862	11	0.277829	2.752498
6	POLYGON ((-77.52461 0.81396, -77.52461 0.90441...	-864	9	0.290258	3.042756
30	POLYGON ((-77.16528 1.44714, -77.16528 1.53761...	-860	16	0.293780	3.336536
13	POLYGON ((-77.43478 0.90441, -77.43478 0.99486...	-863	10	0.304329	3.640865
29	POLYGON ((-77.16528 1.35668, -77.16528 1.44714...	-860	15	0.416053	4.056918
15	POLYGON ((-77.43478 1.08531, -77.43478 1.17576...	-863	12	0.429132	4.486050
14	POLYGON ((-77.43478 0.99486, -77.43478 1.08531...	-863	11	0.609044	5.095094
22	POLYGON ((-77.25511 1.17576, -77.25511 1.26622...	-861	13	0.641881	5.736974
23	POLYGON ((-77.25511 1.26622, -77.25511 1.35668...	-861	14	0.709896	6.446870

Calculate cumulative proportion

- (4) Calculate cumulative proportion by dividing cumulative area by total area (cumulative proportion takes values between 0 and 1)

```

if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)

```

	geometry	grid_col	grid_row	T1_2_19	cumulative_area
4	POLYGON ((-77.61444 0.90441, -77.61444 0.99486...	-865	10	0.001692	0.001692
19	POLYGON ((-77.34495 1.17576, -77.34495 1.26622...	-862	13	0.002867	0.004560
11	POLYGON ((-77.43478 0.72351, -77.43478 0.81396...	-863	8	0.010432	0.014992
0	POLYGON ((-77.7941 0.81396, -77.7941 0.90441, ...	-867	9	0.019752	0.034743
2	POLYGON ((-77.61444 0.72351, -77.61444 0.81396...	-865	8	0.022897	0.057640
10	POLYGON ((-77.52461 1.17576, -77.52461 1.26622...	-864	13	0.027672	0.085312
26	POLYGON ((-77.16528 1.08531, -77.16528 1.17576...	-860	12	0.030285	0.115597
9	POLYGON ((-77.52461 1.08531, -77.52461 1.17576...	-864	12	0.039459	0.155056
12	POLYGON ((-77.43478 0.81396, -77.43478 0.90441...	-863	9	0.046522	0.201578
7	POLYGON ((-77.52461 0.90441, -77.52461 0.99486...	-864	10	0.052011	0.253589
1	POLYGON ((-77.70427 0.81396, -77.70427 0.90441...	-866	9	0.063122	0.316711
21	POLYGON ((-77.25511 1.08531, -77.25511 1.17576...	-861	12	0.082091	0.398802
3	POLYGON ((-77.61444 0.81396, -77.61444 0.90441...	-865	9	0.094620	0.493423
8	POLYGON ((-77.52461 0.99486, -77.52461 1.08531...	-864	11	0.095877	0.589299
32	POLYGON ((-77.07545 1.44714, -77.07545 1.53761...	-859	16	0.115833	0.705133
20	POLYGON ((-77.34495 1.26622, -77.34495 1.35668...	-862	14	0.119695	0.824828
5	POLYGON ((-77.52461 0.72351, -77.52461 0.81396...	-864	8	0.157733	0.982561
18	POLYGON ((-77.34495 1.08531, -77.34495 1.17576...	-862	12	0.173535	1.156097
27	POLYGON ((-77.16528 1.17576, -77.16528 1.26622...	-860	13	0.173905	1.330002
31	POLYGON ((-77.07545 1.35668, -77.07545 1.44714...	-859	15	0.183151	1.513153
25	POLYGON ((-77.25511 1.44714, -77.25511 1.53761...	-861	16	0.193222	1.706375
28	POLYGON ((-77.16528 1.26622, -77.16528 1.35668...	-860	14	0.236492	1.942867
16	POLYGON ((-77.43478 1.17576, -77.43478 1.26622...	-863	13	0.264533	2.207400
24	POLYGON ((-77.25511 1.35668, -77.25511 1.44714...	-861	15	0.267268	2.474669
17	POLYGON ((-77.34495 0.99486, -77.34495 1.08531...	-862	11	0.277829	2.752498
6	POLYGON ((-77.52461 0.81396, -77.52461 0.90441...	-864	9	0.290258	3.042756
30	POLYGON ((-77.16528 1.44714, -77.16528 1.53761...	-860	16	0.293780	3.336536
13	POLYGON ((-77.43478 0.90441, -77.43478 0.99486...	-863	10	0.304329	3.640865
29	POLYGON ((-77.16528 1.35668, -77.16528 1.44714...	-860	15	0.416053	4.056918
15	POLYGON ((-77.43478 1.08531, -77.43478 1.17576...	-863	12	0.429132	4.486050
14	POLYGON ((-77.43478 0.99486, -77.43478 1.08531...	-863	11	0.609044	5.095094
22	POLYGON ((-77.25511 1.17576, -77.25511 1.26622...	-861	13	0.641881	5.736974
23	POLYGON ((-77.25511 1.26622, -77.25511 1.35668...	-861	14	0.709896	6.446870

T1.2.8 - Chicamocha Valley dry forests

Assessment

Chicamocha Valley dry forests

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T1.2

Global classification: T1.2.8

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 69 (10 km x 10 km) grid cells . EOO was measured as 10506 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.  
# This line is replaced by build_ecosystem_pages.py for each ecosystem.  
ecosystem_code = 'T1.2.8'
```

Import Python modules.

```
import os  
import yaml  
from pathlib import Path  
from lonboard import Map  
from rle_python_gee.ecosystems import Ecosystems  
from rle_python_gee.eoo import make_eoo  
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))  
config_path = Path(project_root) / 'config' / 'country_config.yaml'  
with open(config_path) as f:  
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']  
ecosystems = Ecosystems.from_file(  
    source['data'],  
    ecosystem_column=source['ecosystem_code_column'],  
    ecosystem_name_column=source['ecosystem_name_column'],  
    functional_group_column=source['functional_group_column']  
)
```

Filter by the T1.2.8 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 55
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f3c30b6dbd0>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 10506.1 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```

```

if has_data:
    cmap = LinearSegmentedColormap.from_list("white_red", ["white", "red"])
    values = aoo_grid_filtered.grid_cells[ecosystem_column].values
    normalized = (values - values.min()) / (values.max() - values.min())
    colors = apply_continuous_cmap(normalized, cmap)
    display(smart_map([(aoo_grid_filtered, {"get_fill_color": colors}), ecosystem]))

```

```

/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.

```

```
warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f3c30bcc090>, VBox(children=(ErrorOutput(), E
```

Calculate grid cell area and total area

- (2) Calculate extent of the ecosystem type in each grid cell (area) and sum these areas to obtain the total ecosystem area (total area).

```

if has_data:
    keep = ['geometry', 'grid_col', 'grid_row', ecosystem_column]
    gdf = aoo_grid_filtered.grid_cells[keep]
    display(gdf)

```

	geometry	grid_col	grid_row	T1_2_8
0	POLYGON ((-73.30253 6.25222, -73.30253 6.34319...	-817	69	0.065599
1	POLYGON ((-73.30253 6.34319, -73.30253 6.43418...	-817	70	0.089367
2	POLYGON ((-73.30253 6.43418, -73.30253 6.52518...	-817	71	0.082105
3	POLYGON ((-73.30253 6.52518, -73.30253 6.6162,...	-817	72	0.036622
4	POLYGON ((-73.30253 6.6162, -73.30253 6.70724,...	-817	73	0.000056
...
79	POLYGON ((-72.40421 6.07032, -72.40421 6.16126...	-807	67	0.048667
80	POLYGON ((-72.40421 6.16126, -72.40421 6.25222...	-807	68	0.181465
81	POLYGON ((-72.40421 6.34319, -72.40421 6.43418...	-807	70	0.161152
82	POLYGON ((-72.40421 6.43418, -72.40421 6.52518...	-807	71	0.401867
83	POLYGON ((-72.40421 6.52518, -72.40421 6.6162,...	-807	72	0.008742

The column T1_2_8 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```

if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)

```

```
np.float64(27.954203129344013)
```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```

if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)

```

	geometry	grid_col	grid_row	T1_2_8	cumulative_area
4	POLYGON ((-73.30253 6.6162, -73.30253 6.70724,...	-817	73	0.000056	0.000056
46	POLYGON ((-72.76354 5.61584, -72.76354 5.70671...	-811	62	0.000111	0.000167
7	POLYGON ((-73.2127 6.25222, -73.2127 6.34319, ...	-816	69	0.000872	0.001039
6	POLYGON ((-73.30253 6.79829, -73.30253 6.88936...	-817	75	0.001521	0.002561
51	POLYGON ((-72.76354 6.07032, -72.76354 6.16126...	-811	67	0.001988	0.004549
...
34	POLYGON ((-72.9432 6.70724, -72.9432 6.79829, ...	-813	74	0.908254	24.134543
42	POLYGON ((-72.85337 6.52518, -72.85337 6.6162,...	-812	72	0.912845	25.047389
77	POLYGON ((-72.49404 6.43418, -72.49404 6.52518...	-808	71	0.954000	26.001389
68	POLYGON ((-72.58387 6.25222, -72.58387 6.34319...	-809	69	0.963909	26.965297
69	POLYGON ((-72.58387 6.34319, -72.58387 6.43418...	-809	70	0.988906	27.954203

Calculate cumulative proportion

- (4) Calculate cumulative proportion by dividing cumulative area by total area (cumulative proportion takes values between 0 and 1)

```

if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)

```

	geometry	grid_col	grid_row	T1_2_8	cumulative_area
4	POLYGON ((-73.30253 6.6162, -73.30253 6.70724,...	-817	73	0.000056	0.000056
46	POLYGON ((-72.76354 5.61584, -72.76354 5.70671...	-811	62	0.000111	0.000167
7	POLYGON ((-73.2127 6.25222, -73.2127 6.34319, ...	-816	69	0.000872	0.001039
6	POLYGON ((-73.30253 6.79829, -73.30253 6.88936...	-817	75	0.001521	0.002561
51	POLYGON ((-72.76354 6.07032, -72.76354 6.16126...	-811	67	0.001988	0.004549
...
34	POLYGON ((-72.9432 6.70724, -72.9432 6.79829, ...	-813	74	0.908254	24.134543
42	POLYGON ((-72.85337 6.52518, -72.85337 6.6162,...	-812	72	0.912845	25.047389
77	POLYGON ((-72.49404 6.43418, -72.49404 6.52518...	-808	71	0.954000	26.001389
68	POLYGON ((-72.58387 6.25222, -72.58387 6.34319...	-809	69	0.963909	26.965297
69	POLYGON ((-72.58387 6.34319, -72.58387 6.43418...	-809	70	0.988906	27.954203

Count AOO cells

- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'A00 is {aoo} cells')
```

A00 is 69 cells

AOO Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'A00: {aoo_count} grid cells')
```

A00: 69 grid cells

```
if has_data:
    display(smart_map([aoo_grid_filtered, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geopandas/ops/reproject.py:116: UserWarning: Input being reprojected to EPSG:4326
Lonboard is only able to render data in EPSG:4326 projection.
  warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f3bb8e87e90>, VBox(children=(ErrorOutput(), E
```

Criterion B Summary

Ecosystem Code	Ecosystem Name	EOO	AOO
T1.2.8	Chicamocha Valley dry forests	10506 km ²	69 cells

T3.1.1 - Pantepui

Assessment

Pantepui

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T3.1

Global classification: T3.1.1

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 134 (10 km x 10 km) grid cells . EOO was measured as 129229 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.  
# This line is replaced by build_ecosystem_pages.py for each ecosystem.  
ecosystem_code = 'T3.1.1'
```

Import Python modules.

```
import os  
import yaml  
from pathlib import Path  
from lonboard import Map  
from rle_python_gee.ecosystems import Ecosystems  
from rle_python_gee.eoo import make_eoo  
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))  
config_path = Path(project_root) / 'config' / 'country_config.yaml'  
with open(config_path) as f:  
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']  
ecosystems = Ecosystems.from_file(  
    source['data'],  
    ecosystem_column=source['ecosystem_code_column'],  
    ecosystem_name_column=source['ecosystem_name_column'],  
    functional_group_column=source['functional_group_column']  
)
```

Filter by the T3.1.1 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 50
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f30385a7990>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 129228.7 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```

```

if has_data:
    cmap = LinearSegmentedColormap.from_list("white_red", ["white", "red"])
    values = aoo_grid_filtered.grid_cells[ecosystem_column].values
    normalized = (values - values.min()) / (values.max() - values.min())
    colors = apply_continuous_cmap(normalized, cmap)
    display(smart_map([(aoo_grid_filtered, {"get_fill_color": colors}), ecosystem]))

```

```

/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
warnings.warn(

```

```

VBox(children=(<lonboard._map.Map object at 0x7f30385a49d0>, VBox(children=(ErrorOutput(), E

```

Calculate grid cell area and total area

- (2) Calculate extent of the ecosystem type in each grid cell (area) and sum these areas to obtain the total ecosystem area (total area).

```

if has_data:
    keep = ['geometry', 'grid_col', 'grid_row', ecosystem_column]
    gdf = aoo_grid_filtered.grid_cells[keep]
    display(gdf)

```

	geometry	grid_col	grid_row	T3_1_1
0	POLYGON ((-73.2127 0.54263, -73.2127 0.63307, ...	-816	6	0.123263
1	POLYGON ((-73.2127 0.63307, -73.2127 0.72351, ...	-816	7	0.059392
2	POLYGON ((-73.12286 0.27131, -73.12286 0.36175...	-815	3	0.102231
3	POLYGON ((-73.12286 0.36175, -73.12286 0.45219...	-815	4	0.139359
4	POLYGON ((-73.12286 0.45219, -73.12286 0.54263...	-815	5	0.349538
...
170	POLYGON ((-68.27196 2.17099, -68.27196 2.2615,...	-761	24	0.007212
171	POLYGON ((-68.18213 2.08049, -68.18213 2.17099...	-760	23	0.381188
172	POLYGON ((-68.18213 2.17099, -68.18213 2.2615,...	-760	24	0.564259
173	POLYGON ((-68.0923 2.08049, -68.0923 2.17099, ...	-759	23	0.206402
174	POLYGON ((-68.0923 2.17099, -68.0923 2.2615, -...	-759	24	0.000011

The column T3_1_1 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```

if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)

```

```
np.float64(37.59302761708342)
```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```

if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)

```

	geometry	grid_col	grid_row	T3_1_1	cumulative_
9	POLYGON ((-73.03303 0.54263, -73.03303 0.63307...	-814	6	2.836323e-07	2.836323e-0
21	POLYGON ((-72.76354 0.36175, -72.76354 0.45219...	-811	4	5.195489e-06	5.479121e-0
174	POLYGON ((-68.0923 2.17099, -68.0923 2.2615, -...	-759	24	1.069085e-05	1.616997e-0
66	POLYGON ((-72.31438 -0.18087, -72.31438 -0.090...	-806	-2	5.918903e-05	7.535900e-0
139	POLYGON ((-70.60758 0.36175, -70.60758 0.45219...	-787	4	7.005047e-05	1.454095e-0
...
112	POLYGON ((-70.78724 0.09044, -70.78724 0.18087...	-789	1	7.638137e-01	3.435782e+
132	POLYGON ((-70.69741 0.72351, -70.69741 0.81396...	-788	8	7.833165e-01	3.514114e+
108	POLYGON ((-70.87708 0.99486, -70.87708 1.08531...	-790	11	8.050010e-01	3.594614e+
72	POLYGON ((-71.41607 1.62808, -71.41607 1.71855...	-796	18	8.211107e-01	3.676725e+
131	POLYGON ((-70.69741 0.63307, -70.69741 0.72351...	-788	7	8.257762e-01	3.759303e+

Calculate cumulative proportion

- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)

```

if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)

```

	geometry	grid_col	grid_row	T3_1_1	cumulative_
9	POLYGON ((-73.03303 0.54263, -73.03303 0.63307...	-814	6	2.836323e-07	2.836323e-0
21	POLYGON ((-72.76354 0.36175, -72.76354 0.45219...	-811	4	5.195489e-06	5.479121e-0
174	POLYGON ((-68.0923 2.17099, -68.0923 2.2615, -...	-759	24	1.069085e-05	1.616997e-0
66	POLYGON ((-72.31438 -0.18087, -72.31438 -0.090...	-806	-2	5.918903e-05	7.535900e-0
139	POLYGON ((-70.60758 0.36175, -70.60758 0.45219...	-787	4	7.005047e-05	1.454095e-0
...
112	POLYGON ((-70.78724 0.09044, -70.78724 0.18087...	-789	1	7.638137e-01	3.435782e+
132	POLYGON ((-70.69741 0.72351, -70.69741 0.81396...	-788	8	7.833165e-01	3.514114e+
108	POLYGON ((-70.87708 0.99486, -70.87708 1.08531...	-790	11	8.050010e-01	3.594614e+
72	POLYGON ((-71.41607 1.62808, -71.41607 1.71855...	-796	18	8.211107e-01	3.676725e+
131	POLYGON ((-70.69741 0.63307, -70.69741 0.72351...	-788	7	8.257762e-01	3.759303e+

Count AOO cells

- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'A00 is {aoo} cells')
```

A00 is 134 cells

AOO Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'A00: {aoo_count} grid cells')
```

A00: 134 grid cells

```
if has_data:
    display(smart_map([aoo_grid_filtered, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
  warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f2fc3cf9490>, VBox(children=(ErrorOutput(), E
```

Criterion B Summary

Ecosystem Code	Ecosystem Name	EOO	AOO
T3.1.1	Pantepui	129229 km ²	134 cells

T4.2.2 - Llanos

Assessment

Llanos

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T4.2

Global classification: T4.2.2

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 1 (10 km x 10 km) grid cells . EOO was measured as 2 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.  
# This line is replaced by build_ecosystem_pages.py for each ecosystem.  
ecosystem_code = 'T4.2.2'
```

Import Python modules.

```
import os  
import yaml  
from pathlib import Path  
from lonboard import Map  
from rle_python_gee.ecosystems import Ecosystems  
from rle_python_gee.eoo import make_eoo  
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))  
config_path = Path(project_root) / 'config' / 'country_config.yaml'  
with open(config_path) as f:  
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']  
ecosystems = Ecosystems.from_file(  
    source['data'],  
    ecosystem_column=source['ecosystem_code_column'],  
    ecosystem_name_column=source['ecosystem_name_column'],  
    functional_group_column=source['functional_group_column']  
)
```

Filter by the T4.2.2 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 1
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f5bec0d6f90>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 1.8 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```

```

if has_data:
    cmap = LinearSegmentedColormap.from_list("white_red", ["white", "red"])
    values = aoo_grid_filtered.grid_cells[ecosystem_column].values
    normalized = (values - values.min()) / (values.max() - values.min())
    colors = apply_continuous_cmap(normalized, cmap)
    display(smart_map([(aoo_grid_filtered, {"get_fill_color": colors}), ecosystem]))

```

```

/tmp/ipykernel_4346/3217409409.py:9: RuntimeWarning: invalid value encountered in divide
    normalized = (values - values.min()) / (values.max() - values.min())
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(

```

```

VBox(children=(<lonboard._map.Map object at 0x7f5bf75fcd10>, VBox(children=(ErrorOutput(), E

```

Calculate grid cell area and total area

- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).

```

if has_data:
    keep = ['geometry', 'grid_col', 'grid_row', ecosystem_column]
    gdf = aoo_grid_filtered.grid_cells[keep]
    display(gdf)

```

	geometry	grid_col	grid_row	T4_2_2
0	POLYGON ((-70.69741 5.07091, -70.69741 5.1617,...	-788	56	0.009767

The column T4_2_2 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```

if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)

```

```

np.float64(0.009767005260212538)

```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```
if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)
```

	geometry	grid_col	grid_row	T4_2_2	cumulative_area
0	POLYGON ((-70.69741 5.07091, -70.69741 5.1617,...	-788	56	0.009767	0.009767

Calculate cumulative proportion

- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (**cumulative proportion** takes values between 0 and 1)

```
if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)
```

	geometry	grid_col	grid_row	T4_2_2	cumulative_area
0	POLYGON ((-70.69741 5.07091, -70.69741 5.1617,...	-788	56	0.009767	0.009767

Count AOO cells

- (5) Calculate **AOO** by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'AOO is {aoo} cells')
```

AOO is 1 cells

AOO Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'AOO: {aoo_count} grid cells')
```

AOO: 1 grid cells

```
if has_data:
    display(smart_map([aoo_grid_filtered, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoparrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f5bf8606d50>, VBox(children=(ErrorOutput(), E
```

Criterion B Summary

Ecosystem Code	Ecosystem Name	EOO	AOO
T4.2.2	Llanos	2 km ²	1 cells

T4.2.5 - Llanos

Assessment

Llanos

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T4.2

Global classification: T4.2.5

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 505 (10 km x 10 km) grid cells . EOO was measured as 129384 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.  
# This line is replaced by build_ecosystem_pages.py for each ecosystem.  
ecosystem_code = 'T4.2.5'
```

Import Python modules.

```
import os  
import yaml  
from pathlib import Path  
from lonboard import Map  
from rle_python_gee.ecosystems import Ecosystems  
from rle_python_gee.eoo import make_eoo  
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))  
config_path = Path(project_root) / 'config' / 'country_config.yaml'  
with open(config_path) as f:  
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']  
ecosystems = Ecosystems.from_file(  
    source['data'],  
    ecosystem_column=source['ecosystem_code_column'],  
    ecosystem_name_column=source['ecosystem_name_column'],  
    functional_group_column=source['functional_group_column']  
)
```

Filter by the T4.2.5 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 521
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f33309d8610>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 129383.8 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```

```

if has_data:
    cmap = LinearSegmentedColormap.from_list("white_red", ["white", "red"])
    values = aoo_grid_filtered.grid_cells[ecosystem_column].values
    normalized = (values - values.min()) / (values.max() - values.min())
    colors = apply_continuous_cmap(normalized, cmap)
    display(smart_map([(aoo_grid_filtered, {"get_fill_color": colors}), ecosystem]))

```

```

/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.

```

```
warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f33a8483010>, VBox(children=(ErrorOutput(), E
```

Calculate grid cell area and total area

- (2) Calculate extent of the ecosystem type in each grid cell (area) and sum these areas to obtain the total ecosystem area (total area).

```

if has_data:
    keep = ['geometry', 'grid_col', 'grid_row', ecosystem_column]
    gdf = aoo_grid_filtered.grid_cells[keep]
    display(gdf)

```

	geometry	grid_col	grid_row	T4_2_5
0	POLYGON ((-73.84152 1.71855, -73.84152 1.80903...	-823	19	3.939652e-08
1	POLYGON ((-72.9432 3.98235, -72.9432 4.073, -7...	-813	44	3.233195e-03
2	POLYGON ((-72.85337 4.073, -72.85337 4.16367, ...	-812	45	3.464122e-01
3	POLYGON ((-72.85337 4.16367, -72.85337 4.25434...	-812	46	1.972495e-01
4	POLYGON ((-72.76354 3.98235, -72.76354 4.073, ...	-811	44	2.147361e-01
...
596	POLYGON ((-68.0923 6.16126, -68.0923 6.25222, ...	-759	68	1.578826e-01
597	POLYGON ((-68.00247 6.07032, -68.00247 6.16126...	-758	67	4.670375e-01
598	POLYGON ((-68.00247 6.16126, -68.00247 6.25222...	-758	68	3.093356e-02
599	POLYGON ((-67.91264 6.07032, -67.91264 6.16126...	-757	67	1.125859e-01
600	POLYGON ((-67.91264 6.16126, -67.91264 6.25222...	-757	68	3.389178e-02

The column T4_2_5 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```

if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)

```

```

np.float64(258.5065204836777)

```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```

if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)

```

	geometry	grid_col	grid_row	T4_2_5	cumulative_
0	POLYGON ((-73.84152 1.71855, -73.84152 1.80903...	-823	19	3.939652e-08	3.939652e-0
158	POLYGON ((-70.42792 4.25434, -70.42792 4.34503...	-785	47	3.796168e-06	3.835565e-0
576	POLYGON ((-68.27196 5.79759, -68.27196 5.88849...	-761	64	1.218001e-05	1.601558e-0
534	POLYGON ((-68.54146 5.70671, -68.54146 5.79759...	-764	63	1.248884e-05	2.850441e-0
397	POLYGON ((-69.17028 4.79862, -69.17028 4.88937...	-771	53	2.216783e-05	5.067224e-0
...
234	POLYGON ((-69.97876 5.61584, -69.97876 5.70671...	-780	62	9.477456e-01	2.546492e+
252	POLYGON ((-69.88893 5.70671, -69.88893 5.79759...	-779	63	9.561567e-01	2.556053e+
292	POLYGON ((-69.70927 5.70671, -69.70927 5.79759...	-777	63	9.591442e-01	2.565645e+
155	POLYGON ((-70.51775 5.2525, -70.51775 5.34332,...	-786	58	9.632359e-01	2.575277e+
519	POLYGON ((-68.63129 5.9794, -68.63129 6.07032,...	-765	66	9.788116e-01	2.585065e+

Calculate cumulative proportion

- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)

```

if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)

```

	geometry	grid_col	grid_row	T4_2_5	cumulative_
0	POLYGON ((-73.84152 1.71855, -73.84152 1.80903...	-823	19	3.939652e-08	3.939652e-0
158	POLYGON ((-70.42792 4.25434, -70.42792 4.34503...	-785	47	3.796168e-06	3.835565e-0
576	POLYGON ((-68.27196 5.79759, -68.27196 5.88849...	-761	64	1.218001e-05	1.601558e-0
534	POLYGON ((-68.54146 5.70671, -68.54146 5.79759...	-764	63	1.248884e-05	2.850441e-0
397	POLYGON ((-69.17028 4.79862, -69.17028 4.88937...	-771	53	2.216783e-05	5.067224e-0
...
234	POLYGON ((-69.97876 5.61584, -69.97876 5.70671...	-780	62	9.477456e-01	2.546492e+
252	POLYGON ((-69.88893 5.70671, -69.88893 5.79759...	-779	63	9.561567e-01	2.556053e+
292	POLYGON ((-69.70927 5.70671, -69.70927 5.79759...	-777	63	9.591442e-01	2.565645e+
155	POLYGON ((-70.51775 5.2525, -70.51775 5.34332,...	-786	58	9.632359e-01	2.575277e+
519	POLYGON ((-68.63129 5.9794, -68.63129 6.07032,...	-765	66	9.788116e-01	2.585065e+

Count AOO cells

- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'A00 is {aoo} cells')
```

A00 is 505 cells

AOO Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'A00: {aoo_count} grid cells')
```

A00: 505 grid cells

```
if has_data:
    display(smart_map([aoo_grid_filtered, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
  warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f33311e1050>, VBox(children=(ErrorOutput(), E
```

Criterion B Summary

Ecosystem Code	Ecosystem Name	EOO	AOO
T4.2.5	Llanos	129384 km ²	505 cells

T5.2.1 - Guajira-Barranquilla xeric scrub

Assessment

Guajira-Barranquilla xeric scrub

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T5.2

Global classification: T5.2.1

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 93 (10 km x 10 km) grid cells . EOO was measured as 11310 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.  
# This line is replaced by build_ecosystem_pages.py for each ecosystem.  
ecosystem_code = 'T5.2.1'
```

Import Python modules.

```
import os  
import yaml  
from pathlib import Path  
from lonboard import Map  
from rle_python_gee.ecosystems import Ecosystems  
from rle_python_gee.eoo import make_eoo  
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))  
config_path = Path(project_root) / 'config' / 'country_config.yaml'  
with open(config_path) as f:  
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']  
ecosystems = Ecosystems.from_file(  
    source['data'],  
    ecosystem_column=source['ecosystem_code_column'],  
    ecosystem_name_column=source['ecosystem_name_column'],  
    functional_group_column=source['functional_group_column']  
)
```

Filter by the T5.2.1 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 68
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7fd48cc370d0>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 11310.3 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```

```

if has_data:
    cmap = LinearSegmentedColormap.from_list("white_red", ["white", "red"])
    values = aoo_grid_filtered.grid_cells[ecosystem_column].values
    normalized = (values - values.min()) / (values.max() - values.min())
    colors = apply_continuous_cmap(normalized, cmap)
    display(smart_map([(aoo_grid_filtered, {"get_fill_color": colors}), ecosystem]))

```

```

/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
warnings.warn(

```

```

VBox(children=(<lonboard._map.Map object at 0x7fd4277267d0>, VBox(children=(ErrorOutput(), E

```

Calculate grid cell area and total area

- (2) Calculate extent of the ecosystem type in each grid cell (area) and sum these areas to obtain the total ecosystem area (total area).

```

if has_data:
    keep = ['geometry', 'grid_col', 'grid_row', ecosystem_column]
    gdf = aoo_grid_filtered.grid_cells[keep]
    display(gdf)

```

	geometry	grid_col	grid_row	T5_2_1
0	POLYGON ((-72.85337 11.56174, -72.85337 11.654...	-812	127	0.006011
1	POLYGON ((-72.76354 11.56174, -72.76354 11.654...	-811	127	0.046523
2	POLYGON ((-72.76354 11.65401, -72.76354 11.746...	-811	128	0.041689
3	POLYGON ((-72.67371 11.56174, -72.67371 11.654...	-810	127	0.000567
4	POLYGON ((-72.67371 11.65401, -72.67371 11.746...	-810	128	0.117268
...
109	POLYGON ((-71.14657 12.2083, -71.14657 12.3007...	-793	134	0.247815
110	POLYGON ((-71.14657 12.30079, -71.14657 12.393...	-793	135	0.001373
111	POLYGON ((-71.05674 11.93102, -71.05674 12.023...	-792	131	0.040519
112	POLYGON ((-71.05674 12.02342, -71.05674 12.115...	-792	132	0.309626
113	POLYGON ((-71.05674 12.11584, -71.05674 12.208...	-792	133	0.017712

The column T5_2_1 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```

if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)

```

```
np.float64(56.1608246206109)
```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```

if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)

```

	geometry	grid_col	grid_row	T5_2_1	cumulative_area
33	POLYGON ((-72.04489 11.46949, -72.04489 11.561...	-803	126	0.000037	0.000037
32	POLYGON ((-72.04489 11.37728, -72.04489 11.469...	-803	125	0.000177	0.000214
23	POLYGON ((-72.13472 11.28509, -72.13472 11.377...	-804	124	0.000313	0.000527
31	POLYGON ((-72.04489 11.28509, -72.04489 11.377...	-803	124	0.000437	0.000965
3	POLYGON ((-72.67371 11.56174, -72.67371 11.654...	-810	127	0.000567	0.001531
...
20	POLYGON ((-72.22455 11.65401, -72.22455 11.746...	-805	128	0.982051	52.161253
58	POLYGON ((-71.77539 11.93102, -71.77539 12.023...	-800	131	0.999571	53.160825
88	POLYGON ((-71.41607 12.02342, -71.41607 12.115...	-796	132	1.000000	54.160825
81	POLYGON ((-71.5059 12.11584, -71.5059 12.2083,...	-797	133	1.000000	55.160825
74	POLYGON ((-71.59573 12.11584, -71.59573 12.208...	-798	133	1.000000	56.160825

Calculate cumulative proportion

- (4) Calculate cumulative proportion by dividing cumulative area by total area (cumulative proportion takes values between 0 and 1)

```

if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)

```

	geometry	grid_col	grid_row	T5_2_1	cumulative_area
33	POLYGON ((-72.04489 11.46949, -72.04489 11.561...	-803	126	0.000037	0.000037
32	POLYGON ((-72.04489 11.37728, -72.04489 11.469...	-803	125	0.000177	0.000214
23	POLYGON ((-72.13472 11.28509, -72.13472 11.377...	-804	124	0.000313	0.000527
31	POLYGON ((-72.04489 11.28509, -72.04489 11.377...	-803	124	0.000437	0.000965
3	POLYGON ((-72.67371 11.56174, -72.67371 11.654...	-810	127	0.000567	0.001531
...
20	POLYGON ((-72.22455 11.65401, -72.22455 11.746...	-805	128	0.982051	52.161253
58	POLYGON ((-71.77539 11.93102, -71.77539 12.023...	-800	131	0.999571	53.160825
88	POLYGON ((-71.41607 12.02342, -71.41607 12.115...	-796	132	1.000000	54.160825
81	POLYGON ((-71.5059 12.11584, -71.5059 12.2083,...	-797	133	1.000000	55.160825
74	POLYGON ((-71.59573 12.11584, -71.59573 12.208...	-798	133	1.000000	56.160825

Count AOO cells

- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'A00 is {aoo} cells')
```

A00 is 93 cells

A00 Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'A00: {aoo_count} grid cells')
```

A00: 93 grid cells

```
if has_data:
    display(smart_map([aoo_grid_filtered, ecosystem]))
```


T5.2.2 - Guajira-Barranquilla xeric scrub

Assessment

Guajira-Barranquilla xeric scrub

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T5.2

Global classification: T5.2.2

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 42 (10 km x 10 km) grid cells . EOO was measured as 8839 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.
# This line is replaced by build_ecosystem_pages.py for each ecosystem.
ecosystem_code = 'T5.2.2'
```

Import Python modules.

```
import os
import yaml
from pathlib import Path
from lonboard import Map
from rle_python_gee.ecosystems import Ecosystems
from rle_python_gee.eoo import make_eoo
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))
config_path = Path(project_root) / 'config' / 'country_config.yaml'
with open(config_path) as f:
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']
ecosystems = Ecosystems.from_file(
    source['data'],
    ecosystem_column=source['ecosystem_code_column'],
    ecosystem_name_column=source['ecosystem_name_column'],
    functional_group_column=source['functional_group_column']
)
```

Filter by the T5.2.2 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 47
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7f2dd7ed5dd0>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 8839.1 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```


	geometry	grid_col	grid_row	T5_2_2
13	POLYGON ((-71.95505 12.11584, -71.95505 12.208...	-802	133	0.672293
14	POLYGON ((-71.95505 12.2083, -71.95505 12.3007...	-802	134	0.358193
15	POLYGON ((-71.86522 12.02342, -71.86522 12.115...	-801	132	0.380588
16	POLYGON ((-71.86522 12.11584, -71.86522 12.208...	-801	133	0.512544
17	POLYGON ((-71.86522 12.2083, -71.86522 12.3007...	-801	134	0.230348
18	POLYGON ((-71.86522 12.30079, -71.86522 12.393...	-801	135	0.156713
19	POLYGON ((-71.77539 11.65401, -71.77539 11.746...	-800	128	0.000786
20	POLYGON ((-71.77539 12.02342, -71.77539 12.115...	-800	132	0.177099
21	POLYGON ((-71.77539 12.11584, -71.77539 12.208...	-800	133	0.305242
22	POLYGON ((-71.77539 12.2083, -71.77539 12.3007...	-800	134	0.903159
23	POLYGON ((-71.77539 12.30079, -71.77539 12.393...	-800	135	0.348547
24	POLYGON ((-71.68556 11.65401, -71.68556 11.746...	-799	128	0.000049
25	POLYGON ((-71.68556 11.74632, -71.68556 11.838...	-799	129	0.000009
26	POLYGON ((-71.68556 11.83865, -71.68556 11.931...	-799	130	0.026525
27	POLYGON ((-71.68556 11.93102, -71.68556 12.023...	-799	131	0.214035
28	POLYGON ((-71.68556 12.11584, -71.68556 12.208...	-799	133	0.115390
29	POLYGON ((-71.68556 12.2083, -71.68556 12.3007...	-799	134	0.400309
30	POLYGON ((-71.68556 12.30079, -71.68556 12.393...	-799	135	0.652381
31	POLYGON ((-71.68556 12.39331, -71.68556 12.485...	-799	136	0.107051
32	POLYGON ((-71.59573 11.83865, -71.59573 11.931...	-798	130	0.198667
33	POLYGON ((-71.59573 11.93102, -71.59573 12.023...	-798	131	0.450032
34	POLYGON ((-71.59573 12.02342, -71.59573 12.3007...	-798	134	0.258278
35	POLYGON ((-71.59573 12.30079, -71.59573 12.393...	-798	135	0.932666
36	POLYGON ((-71.59573 12.39331, -71.59573 12.485...	-798	136	0.430782
37	POLYGON ((-71.5059 11.74632, -71.5059 11.83865...	-797	129	0.020141
38	POLYGON ((-71.5059 11.83865, -71.5059 11.93102...	-797	130	0.543172
39	POLYGON ((-71.5059 11.93102, -71.5059 12.02342...	-797	131	0.308959
40	POLYGON ((-71.5059 12.02342, -71.5059 12.30079,...	-797	134	0.193402
41	POLYGON ((-71.5059 12.30079, -71.5059 12.39331...	-797	135	0.749183
42	POLYGON ((-71.5059 12.39331, -71.5059 12.48587...	-797	136	0.358634
43	POLYGON ((-71.41607 11.74632, -71.41607 11.838...	-796	129	0.026162
44	POLYGON ((-71.41607 11.83865, -71.41607 11.931...	-796	130	0.414513
45	POLYGON ((-71.41607 11.93102, -71.41607 12.023...	-796	131	0.571442
46	POLYGON ((-71.41607 12.02342, -71.41607 12.393...	-796	135	0.124567
47	POLYGON ((-71.41607 12.39331, -71.41607 12.485...	-796	136	0.027081
48	POLYGON ((-71.32623 11.74632, -71.32623 11.838...	-795	129	0.013314
49	POLYGON ((-71.32623 11.83865, -71.32623 11.931...	-795	130	0.819897
50	POLYGON ((-71.32623 11.93102, -71.32623 12.023...	-795	131	0.274259
51	POLYGON ((-71.2364 11.83865, -71.2364 11.93102...	-794	130	0.136918
52	POLYGON ((-71.2364 11.93102, -71.2364 12.02342...	-794	131	0.469495

The column T5_2_2 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```
if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)
```

```
np.float64(15.037707238541868)
```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```
if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)
```

	geometry	grid_col	grid_row	T5_2_2	cumulative_area
25	POLYGON ((-71.68556 11.74632, -71.68556 11.838...	-799	129	0.000009	0.000009
24	POLYGON ((-71.68556 11.65401, -71.68556 11.746...	-799	128	0.000049	0.000058
2	POLYGON ((-72.13472 11.28509, -72.13472 11.377...	-804	124	0.000272	0.000330
1	POLYGON ((-72.13472 11.19293, -72.13472 11.285...	-804	123	0.000324	0.000654
19	POLYGON ((-71.77539 11.65401, -71.77539 11.746...	-800	128	0.000786	0.001441
48	POLYGON ((-71.32623 11.74632, -71.32623 11.838...	-795	129	0.013314	0.014755
37	POLYGON ((-71.5059 11.74632, -71.5059 11.83865...	-797	129	0.020141	0.034897
0	POLYGON ((-72.22455 11.83865, -72.22455 11.931...	-805	130	0.025347	0.060244
43	POLYGON ((-71.41607 11.74632, -71.41607 11.838...	-796	129	0.026162	0.086406
26	POLYGON ((-71.68556 11.83865, -71.68556 11.931...	-799	130	0.026525	0.112931
47	POLYGON ((-71.41607 12.39331, -71.41607 12.485...	-796	136	0.027081	0.140012
8	POLYGON ((-72.04489 11.83865, -72.04489 11.931...	-803	130	0.043618	0.183630
12	POLYGON ((-71.95505 12.02342, -71.95505 12.115...	-802	132	0.044158	0.227788
9	POLYGON ((-72.04489 12.02342, -72.04489 12.115...	-803	132	0.072893	0.300680
31	POLYGON ((-71.68556 12.39331, -71.68556 12.485...	-799	136	0.107051	0.407731
7	POLYGON ((-72.13472 12.2083, -72.13472 12.3007...	-804	134	0.108972	0.516703
6	POLYGON ((-72.13472 12.11584, -72.13472 12.208...	-804	133	0.111997	0.628700
28	POLYGON ((-71.68556 12.11584, -71.68556 12.208...	-799	133	0.115390	0.744091
46	POLYGON ((-71.41607 12.30079, -71.41607 12.393...	-796	135	0.124567	0.868658

	geometry	grid_col	grid_row	T5_2_2	cumulative_area
51	POLYGON ((-71.2364 11.83865, -71.2364 11.93102...	-794	130	0.136918	1.005576
18	POLYGON ((-71.86522 12.30079, -71.86522 12.393...	-801	135	0.156713	1.162289
20	POLYGON ((-71.77539 12.02342, -71.77539 12.115...	-800	132	0.177099	1.339388
5	POLYGON ((-72.13472 12.02342, -72.13472 12.115...	-804	132	0.189220	1.528609
40	POLYGON ((-71.5059 12.2083, -71.5059 12.30079,...	-797	134	0.193402	1.722011
32	POLYGON ((-71.59573 11.83865, -71.59573 11.931...	-798	130	0.198667	1.920678
3	POLYGON ((-72.13472 11.83865, -72.13472 11.931...	-804	130	0.200753	2.121431
27	POLYGON ((-71.68556 11.93102, -71.68556 12.023...	-799	131	0.214035	2.335466
4	POLYGON ((-72.13472 11.93102, -72.13472 12.023...	-804	131	0.221869	2.557336
17	POLYGON ((-71.86522 12.2083, -71.86522 12.3007...	-801	134	0.230348	2.787684
34	POLYGON ((-71.59573 12.2083, -71.59573 12.3007...	-798	134	0.258278	3.045962
50	POLYGON ((-71.32623 11.93102, -71.32623 12.023...	-795	131	0.274259	3.320221
11	POLYGON ((-72.04489 12.2083, -72.04489 12.3007...	-803	134	0.294374	3.614596
21	POLYGON ((-71.77539 12.11584, -71.77539 12.208...	-800	133	0.305242	3.919837
39	POLYGON ((-71.5059 11.93102, -71.5059 12.02342...	-797	131	0.308959	4.228797
23	POLYGON ((-71.77539 12.30079, -71.77539 12.393...	-800	135	0.348547	4.577344
14	POLYGON ((-71.95505 12.2083, -71.95505 12.3007...	-802	134	0.358193	4.935537
42	POLYGON ((-71.5059 12.39331, -71.5059 12.48587...	-797	136	0.358634	5.294171
15	POLYGON ((-71.86522 12.02342, -71.86522 12.115...	-801	132	0.380588	5.674759
29	POLYGON ((-71.68556 12.2083, -71.68556 12.3007...	-799	134	0.400309	6.075068
44	POLYGON ((-71.41607 11.83865, -71.41607 11.931...	-796	130	0.414513	6.489581
36	POLYGON ((-71.59573 12.39331, -71.59573 12.485...	-798	136	0.430782	6.920363
33	POLYGON ((-71.59573 11.93102, -71.59573 12.023...	-798	131	0.450032	7.370395
52	POLYGON ((-71.2364 11.93102, -71.2364 12.02342...	-794	131	0.469495	7.839890
16	POLYGON ((-71.86522 12.11584, -71.86522 12.208...	-801	133	0.512544	8.352434
38	POLYGON ((-71.5059 11.83865, -71.5059 11.93102...	-797	130	0.543172	8.895606
45	POLYGON ((-71.41607 11.93102, -71.41607 12.023...	-796	131	0.571442	9.467047
30	POLYGON ((-71.68556 12.30079, -71.68556 12.393...	-799	135	0.652381	10.119428
13	POLYGON ((-71.95505 12.11584, -71.95505 12.208...	-802	133	0.672293	10.791721
41	POLYGON ((-71.5059 12.30079, -71.5059 12.39331...	-797	135	0.749183	11.540904
49	POLYGON ((-71.32623 11.83865, -71.32623 11.931...	-795	130	0.819897	12.360802
10	POLYGON ((-72.04489 12.11584, -72.04489 12.208...	-803	133	0.841081	13.201882
22	POLYGON ((-71.77539 12.2083, -71.77539 12.3007...	-800	134	0.903159	14.105041
35	POLYGON ((-71.59573 12.30079, -71.59573 12.393...	-798	135	0.932666	15.037707

Calculate cumulative proportion

- (4) Calculate cumulative proportion by dividing cumulative area by total area (cumulative proportion takes values between 0 and 1)

```

if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)

```

	geometry	grid_col	grid_row	T5_2_2	cumulative_area
25	POLYGON ((-71.68556 11.74632, -71.68556 11.838...	-799	129	0.000009	0.000009
24	POLYGON ((-71.68556 11.65401, -71.68556 11.746...	-799	128	0.000049	0.000058
2	POLYGON ((-72.13472 11.28509, -72.13472 11.377...	-804	124	0.000272	0.000330
1	POLYGON ((-72.13472 11.19293, -72.13472 11.285...	-804	123	0.000324	0.000654
19	POLYGON ((-71.77539 11.65401, -71.77539 11.746...	-800	128	0.000786	0.001441
48	POLYGON ((-71.32623 11.74632, -71.32623 11.838...	-795	129	0.013314	0.014755
37	POLYGON ((-71.5059 11.74632, -71.5059 11.83865...	-797	129	0.020141	0.034897
0	POLYGON ((-72.22455 11.83865, -72.22455 11.931...	-805	130	0.025347	0.060244
43	POLYGON ((-71.41607 11.74632, -71.41607 11.838...	-796	129	0.026162	0.086406
26	POLYGON ((-71.68556 11.83865, -71.68556 11.931...	-799	130	0.026525	0.112931
47	POLYGON ((-71.41607 12.39331, -71.41607 12.485...	-796	136	0.027081	0.140012
8	POLYGON ((-72.04489 11.83865, -72.04489 11.931...	-803	130	0.043618	0.183630
12	POLYGON ((-71.95505 12.02342, -71.95505 12.115...	-802	132	0.044158	0.227788
9	POLYGON ((-72.04489 12.02342, -72.04489 12.115...	-803	132	0.072893	0.300680
31	POLYGON ((-71.68556 12.39331, -71.68556 12.485...	-799	136	0.107051	0.407731
7	POLYGON ((-72.13472 12.2083, -72.13472 12.3007...	-804	134	0.108972	0.516703
6	POLYGON ((-72.13472 12.11584, -72.13472 12.208...	-804	133	0.111997	0.628700
28	POLYGON ((-71.68556 12.11584, -71.68556 12.208...	-799	133	0.115390	0.744091
46	POLYGON ((-71.41607 12.30079, -71.41607 12.393...	-796	135	0.124567	0.868658
51	POLYGON ((-71.2364 11.83865, -71.2364 11.93102...	-794	130	0.136918	1.005576
18	POLYGON ((-71.86522 12.30079, -71.86522 12.393...	-801	135	0.156713	1.162289
20	POLYGON ((-71.77539 12.02342, -71.77539 12.115...	-800	132	0.177099	1.339388
5	POLYGON ((-72.13472 12.02342, -72.13472 12.115...	-804	132	0.189220	1.528609
40	POLYGON ((-71.5059 12.2083, -71.5059 12.30079,...	-797	134	0.193402	1.722011
32	POLYGON ((-71.59573 11.83865, -71.59573 11.931...	-798	130	0.198667	1.920678
3	POLYGON ((-72.13472 11.83865, -72.13472 11.931...	-804	130	0.200753	2.121431
27	POLYGON ((-71.68556 11.93102, -71.68556 12.023...	-799	131	0.214035	2.335466
4	POLYGON ((-72.13472 11.93102, -72.13472 12.023...	-804	131	0.221869	2.557336
17	POLYGON ((-71.86522 12.2083, -71.86522 12.3007...	-801	134	0.230348	2.787684
34	POLYGON ((-71.59573 12.2083, -71.59573 12.3007...	-798	134	0.258278	3.045962
50	POLYGON ((-71.32623 11.93102, -71.32623 12.023...	-795	131	0.274259	3.320221
11	POLYGON ((-72.04489 12.2083, -72.04489 12.3007...	-803	134	0.294374	3.614596
21	POLYGON ((-71.77539 12.11584, -71.77539 12.208...	-800	133	0.305242	3.919837
39	POLYGON ((-71.5059 11.93102, -71.5059 12.02342...	-797	131	0.308959	4.228797
23	POLYGON ((-71.77539 12.30079, -71.77539 12.393...	-800	135	0.348547	4.577344
14	POLYGON ((-71.95505 12.2083, -71.95505 12.3007...	-802	134	0.358193	4.935537

	geometry	grid_col	grid_row	T5_2_2	cumulative_area
42	POLYGON ((-71.5059 12.39331, -71.5059 12.48587...	-797	136	0.358634	5.294171
15	POLYGON ((-71.86522 12.02342, -71.86522 12.115...	-801	132	0.380588	5.674759
29	POLYGON ((-71.68556 12.2083, -71.68556 12.3007...	-799	134	0.400309	6.075068
44	POLYGON ((-71.41607 11.83865, -71.41607 11.931...	-796	130	0.414513	6.489581
36	POLYGON ((-71.59573 12.39331, -71.59573 12.485...	-798	136	0.430782	6.920363
33	POLYGON ((-71.59573 11.93102, -71.59573 12.023...	-798	131	0.450032	7.370395
52	POLYGON ((-71.2364 11.93102, -71.2364 12.02342...	-794	131	0.469495	7.839890
16	POLYGON ((-71.86522 12.11584, -71.86522 12.208...	-801	133	0.512544	8.352434
38	POLYGON ((-71.5059 11.83865, -71.5059 11.93102...	-797	130	0.543172	8.895606
45	POLYGON ((-71.41607 11.93102, -71.41607 12.023...	-796	131	0.571442	9.467047
30	POLYGON ((-71.68556 12.30079, -71.68556 12.393...	-799	135	0.652381	10.119428
13	POLYGON ((-71.95505 12.11584, -71.95505 12.208...	-802	133	0.672293	10.791721
41	POLYGON ((-71.5059 12.30079, -71.5059 12.39331...	-797	135	0.749183	11.540904
49	POLYGON ((-71.32623 11.83865, -71.32623 11.931...	-795	130	0.819897	12.360802
10	POLYGON ((-72.04489 12.11584, -72.04489 12.208...	-803	133	0.841081	13.201882
22	POLYGON ((-71.77539 12.2083, -71.77539 12.3007...	-800	134	0.903159	14.105041
35	POLYGON ((-71.59573 12.30079, -71.59573 12.393...	-798	135	0.932666	15.037707

Count AOO cells

- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'A00 is {aoo} cells')
```

A00 is 42 cells

AOO Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'A00: {aoo_count} grid cells')
```

A00: 42 grid cells

T5.2.3 - Magdalena Valley dry forests

Assessment

Magdalena Valley dry forests

Authors: John Smith, Jane Smith

Biome: TODO

Functional Group: T5.2

Global classification: T5.2.3

IUCN Status: TODO

Description: TODO

Distribution: TODO

Characteristic Native Biota: TODO

Abiotic environment: TODO

Key processes and interactions: TODO

Major threats: TODO

Ecosystem collapse definition: TODO

Assessment summary: **TODO**

Assessment information:

Criteria

Status

Criterion A

A1

NE

A2a

NE

A2b

NE

A3

NE

Criterion B

B1

NE

B2

NE

subcriteria

NE

B3

NE

Criterion C

C1

NE

C2a

NE

C2b

NE

C3

NE

Criterion D

D1

NE

D2a

NE

D2b

NE

D3

NE

Criterion E

E

NE

Assessment outcome: **TODO**

Year published: TODO

Date assessed: TODO

Assessment credits:

Assessed by: TODO

Reviewed by: TODO

Contributions by: TODO

Criterion A: TODO

Criterion B: AOO was measured as 12 (10 km x 10 km) grid cells . EOO was measured as 774 km².

Criterion C: TODO

Criterion D: TODO

Criterion E: TODO

Criterion B Details

```
# Default ecosystem code for template development.
# This line is replaced by build_ecosystem_pages.py for each ecosystem.
ecosystem_code = 'T5.2.3'
```

Import Python modules.

```
import os
import yaml
from pathlib import Path
from lonboard import Map
from rle_python_gee.ecosystems import Ecosystems
from rle_python_gee.eoo import make_eoo
from rle_python_gee.aoo import make_aoo_grid
```

Load the country config file.

```
project_root = os.environ.get('PIXI_PROJECT_ROOT', str(Path('.').resolve()))
config_path = Path(project_root) / 'config' / 'country_config.yaml'
with open(config_path) as f:
    config = yaml.safe_load(f)
```

Load & Filter Ecosystem Data

Load data for all the ecosystems.

```
source = config['ecosystem_source']
ecosystems = Ecosystems.from_file(
    source['data'],
    ecosystem_column=source['ecosystem_code_column'],
    ecosystem_name_column=source['ecosystem_name_column'],
    functional_group_column=source['functional_group_column']
)
```

Filter by the T5.2.3 and check the number of features.

```
ecosystem = ecosystems.filter(ecosystem_code)
has_data = ecosystem.size() > 0
print(f'{ecosystem.size() = }')
if not has_data:
    from IPython.display import Markdown, display
    display(Markdown(
        f'**No spatial data found for {ecosystem_code}.** '
        f'Criterion B calculations are skipped.'
    ))
```

```
ecosystem.size() = 6
```

Extent of Occurrence (EOO) (subcriterion B1)

Extent of occurrence (EOO). The EOO of an ecosystem is the area (km²) of a minimum convex polygon – the smallest polygon in which no internal angle exceeds 180° that encompasses all known current spatial occurrences of the ecosystem type.

The minimum convex polygon (also known as a convex hull) must not exclude any areas, discontinuities or disjunctions, regardless of whether the ecosystem can occur in those areas or not. Regions such as oceans (for terrestrial ecosystems), land (for coastal or marine ecosystems), or areas outside the study area (such as in a different country) must remain included within the minimum convex polygon to ensure that this standardised method is comparable across ecosystem types. In addition, these features contribute to spreading risks across the distribution of the ecosystem by making different parts of its distribution more spatially independent.

Calculate EOO

Start by calculating the convex hull of the ecosystem's distribution.

```
import geopandas as gpd

if has_data:
    ecosystem_geometry = ecosystem.geometry.union_all()
    gdf_ecosystem_polygons = gpd.GeoDataFrame(geometry=[ecosystem_geometry], crs=ecosystem.g
    hull = ecosystem_geometry.convex_hull
    gdf_hull = gpd.GeoDataFrame(geometry=[hull], crs=ecosystem.geometry.crs)
```

Display the ecosystem's distribution and the convex hull.

```
from lonboard import Map, PolygonLayer
from rle_python_gee.viz import smart_map

if has_data:
    eoo_hull = make_eoo(ecosystem).compute()
    display(smart_map([eoo_hull, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7effe4f09990>, VBox(children=(ErrorOutput(), E
```

```
if has_data:
    hull_ea = gdf_hull.to_crs("ESRI:54034")
    eoo = hull_ea.geometry.iloc[0].area / 1e6
    print(f'E00 is {eoo:.1f} km2')
```

```
E00 is 773.5 km2
```

Then calculate the area of the convex hull polygon.

Direct calculation of EOO

EOO can also be calculated directly using ...

```
if has_data:
    ecosystem.eoo
```

Verify that the area returned by calling `make_eoo(ecosystem).compute().area_km2` is the same as the area of the convex hull polygon.

```
if has_data:
    assert ecosystem.eoo == eoo
```

Area of Occupancy (AOO) (subcriterion B2)

The protocol for this adjustment includes the following steps:

- (1) Intersect AOO grid with the ecosystem's distribution map.
- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).
- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).
- (4) Calculate **cumulative proportion** by dividing **cumulative area** by **total area** (cumulative proportion takes values between 0 and 1)
- (5) Calculate AOO by counting the number of cells with a **cumulative proportion** greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

AOO Calculation Details

Intersect AOO grid and ecosystem map

- (1) Intersect AOO grid with the ecosystem's distribution map

```
from pathlib import Path
from rle_python_gee.aoo import make_aoo_grid_cached

if has_data:
    cache_path = Path(project_root) / '.cache' / 'aoo_grid.parquet'
    aoo_grid = make_aoo_grid_cached(ecosystems, cache_path=cache_path)
    aoo_grid_filtered = aoo_grid.filter_by_ecosystem(ecosystem_code)
```

Visualize variations in the AOO grid.

```
from matplotlib.colors import LinearSegmentedColormap
from lonboard.colormap import apply_continuous_cmap
from rle_python_gee.aoo import slugify_ecosystem_name

ecosystem_column = slugify_ecosystem_name(ecosystem_code)
```

```

if has_data:
    cmap = LinearSegmentedColormap.from_list("white_red", ["white", "red"])
    values = aoo_grid_filtered.grid_cells[ecosystem_column].values
    normalized = (values - values.min()) / (values.max() - values.min())
    colors = apply_continuous_cmap(normalized, cmap)
    display(smart_map([(aoo_grid_filtered, {"get_fill_color": colors}), ecosystem]))

```

/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.11/site-packages/lonboard/_gearrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPSG:4326. Lonboard is only able to render data in EPSG:4326 projection.

```
warnings.warn(
```

```
VBox(children=(<lonboard._map.Map object at 0x7eff90cf4cd0>, VBox(children=(ErrorOutput(),
```

Calculate grid cell area and total area

- (2) Calculate extent of the ecosystem type in each grid cell (**area**) and sum these areas to obtain the total ecosystem area (**total area**).

```

if has_data:
    keep = ['geometry', 'grid_col', 'grid_row', ecosystem_column]
    gdf = aoo_grid_filtered.grid_cells[keep]
    display(gdf)

```

	geometry	grid_col	grid_row	T5_2_3
0	POLYGON ((-75.27882 3.07629, -75.27882 3.16686...	-839	34	0.148599
1	POLYGON ((-75.27882 3.16686, -75.27882 3.25744...	-839	35	0.361386
2	POLYGON ((-75.18899 3.07629, -75.18899 3.16686...	-838	34	0.146667
3	POLYGON ((-75.18899 3.16686, -75.18899 3.25744...	-838	35	0.500675
4	POLYGON ((-75.18899 3.25744, -75.18899 3.34802...	-838	36	0.315243
5	POLYGON ((-75.18899 3.34802, -75.18899 3.43861...	-838	37	0.005295
6	POLYGON ((-75.09916 3.07629, -75.09916 3.16686...	-837	34	0.461403
7	POLYGON ((-75.09916 3.16686, -75.09916 3.25744...	-837	35	0.859556
8	POLYGON ((-75.09916 3.25744, -75.09916 3.34802...	-837	36	0.853786
9	POLYGON ((-75.09916 3.34802, -75.09916 3.43861...	-837	37	0.227379
10	POLYGON ((-75.00933 3.07629, -75.00933 3.16686...	-836	34	0.137939
11	POLYGON ((-75.00933 3.16686, -75.00933 3.25744...	-836	35	0.200385
12	POLYGON ((-75.00933 3.25744, -75.00933 3.34802...	-836	36	0.087718

The column T5_2_3 contains the (fractional) area of the ecosystem in each grid cell.

Sum up the areas of each grid cell to get the total area.

```
if has_data:
    total_area = gdf[ecosystem_column].sum()
    display(total_area)
```

```
np.float64(4.306029834949537)
```

Calculate cumulative area

- (3) Arrange grid cells in ascending order based on their area (smaller first). Calculate accumulated sum of area per cell (**cumulative area**).

```
if has_data:
    gdf = gdf.sort_values(by=ecosystem_column)
    gdf["cumulative_area"] = gdf[ecosystem_column].cumsum()
    display(gdf)
```

	geometry	grid_col	grid_row	T5_2_3	cumulative_area
5	POLYGON ((-75.18899 3.34802, -75.18899 3.43861...	-838	37	0.005295	0.005295
12	POLYGON ((-75.00933 3.25744, -75.00933 3.34802...	-836	36	0.087718	0.093013
10	POLYGON ((-75.00933 3.07629, -75.00933 3.16686...	-836	34	0.137939	0.230952
2	POLYGON ((-75.18899 3.07629, -75.18899 3.16686...	-838	34	0.146667	0.377619
0	POLYGON ((-75.27882 3.07629, -75.27882 3.16686...	-839	34	0.148599	0.526218
11	POLYGON ((-75.00933 3.16686, -75.00933 3.25744...	-836	35	0.200385	0.726603
9	POLYGON ((-75.09916 3.34802, -75.09916 3.43861...	-837	37	0.227379	0.953982
4	POLYGON ((-75.18899 3.25744, -75.18899 3.34802...	-838	36	0.315243	1.269224
1	POLYGON ((-75.27882 3.16686, -75.27882 3.25744...	-839	35	0.361386	1.630611
6	POLYGON ((-75.09916 3.07629, -75.09916 3.16686...	-837	34	0.461403	2.092014
3	POLYGON ((-75.18899 3.16686, -75.18899 3.25744...	-838	35	0.500675	2.592689
8	POLYGON ((-75.09916 3.25744, -75.09916 3.34802...	-837	36	0.853786	3.446474
7	POLYGON ((-75.09916 3.16686, -75.09916 3.25744...	-837	35	0.859556	4.306030

Calculate cumulative proportion

- (4) Calculate cumulative proportion by dividing cumulative area by total area (cumulative proportion takes values between 0 and 1)

```
if has_data:
    gdf["cumulative_proportion"] = gdf["cumulative_area"] / total_area
    display(gdf)
```

	geometry	grid_col	grid_row	T5_2_3	cumulative_area
5	POLYGON ((-75.18899 3.34802, -75.18899 3.43861...	-838	37	0.005295	0.005295
12	POLYGON ((-75.00933 3.25744, -75.00933 3.34802...	-836	36	0.087718	0.093013
10	POLYGON ((-75.00933 3.07629, -75.00933 3.16686...	-836	34	0.137939	0.230952
2	POLYGON ((-75.18899 3.07629, -75.18899 3.16686...	-838	34	0.146667	0.377619
0	POLYGON ((-75.27882 3.07629, -75.27882 3.16686...	-839	34	0.148599	0.526218
11	POLYGON ((-75.00933 3.16686, -75.00933 3.25744...	-836	35	0.200385	0.726603
9	POLYGON ((-75.09916 3.34802, -75.09916 3.43861...	-837	37	0.227379	0.953982
4	POLYGON ((-75.18899 3.25744, -75.18899 3.34802...	-838	36	0.315243	1.269224
1	POLYGON ((-75.27882 3.16686, -75.27882 3.25744...	-839	35	0.361386	1.630611
6	POLYGON ((-75.09916 3.07629, -75.09916 3.16686...	-837	34	0.461403	2.092014
3	POLYGON ((-75.18899 3.16686, -75.18899 3.25744...	-838	35	0.500675	2.592689
8	POLYGON ((-75.09916 3.25744, -75.09916 3.34802...	-837	36	0.853786	3.446474
7	POLYGON ((-75.09916 3.16686, -75.09916 3.25744...	-837	35	0.859556	4.306030

Count AOO cells

- (5) Calculate AOO by counting the number of cells with a cumulative proportion greater than 0.01 (i.e. exclude cells that in combination account for up to 1% of the total mapped extent of the ecosystem type).

```
if has_data:
    aoo = len(gdf[gdf["cumulative_proportion"] > 0.01])
    print(f'AOO is {aoo} cells')
```

AOO is 12 cells

AOO Calculation (direct call)

```
if has_data:
    aoo_count = ecosystem.aoo
    print(f'A00: {aoo_count} grid cells')
```

A00: 12 grid cells

```
if has_data:
    display(smart_map([aoo_grid_filtered, ecosystem]))
```

```
/home/runner/work/rle-ruritania-thomas/rle-ruritania-thomas/.pixi/envs/default/lib/python3.1
packages/lonboard/_geoarrow/ops/reproject.py:116: UserWarning: Input being reprojected to EPS
Lonboard is only able to render data in EPSG:4326 projection.
    warnings.warn(
```

VBox(children=(<lonboard._map.Map object at 0x7eff90cf4110>, VBox(children=(ErrorOutput(), E

Criterion B Summary

Ecosystem Code	Ecosystem Name	EOO	AOO
T5.2.3	Magdalena Valley dry forests	774 km ²	12 cells

References

- Bland, LM, DA Keith, RM Miller, NJ Murray, and JP Rodríguez. 2017. “Guidelines for the Application of IUCN Red List of Ecosystems Categories and Criteria, Version 1.1.” *International Union for the Conservation of Nature, Gland, Switzerland*.
- Bland, Lucie M, Tracey J Regan, Minh Ngoc Dinh, Renata Ferrari, David A Keith, Rebecca Lester, David Mouillot, Nicholas J Murray, Hoang Anh Nguyen, and Emily Nicholson. 2017. “Using Multiple Lines of Evidence to Assess the Risk of Ecosystem Collapse.” *Proceedings of the Royal Society B: Biological Sciences* 284 (1863): 20170660.
- Keith, DA, JR Ferrer-Paris, E Nicholson, MJ Bishop, BA Polidoro, E Ramirez-Llodra, et al. 2022. “Earth’s Ecosystems: A Function-Based Typology for Conservation and Sustainability.” *Nature* 610: 513–18.
- Keith, David A, H Resit Akçakaya, and Nicholas J Murray. 2018. “Scaling Range Sizes to Threats for Robust Predictions of Risks to Biodiversity.” *Conservation Biology* 32 (2): 322–32.
- Keith, David A, Jose R Ferrer-Paris, Emily Nicholson, Melanie J Bishop, Beth A Polidoro, Eva Ramirez-Llodra, Mark G Tozer, et al. 2020. “Indicative Distribution Maps for Ecological Functional Groups-Level 3 of IUCN Global Ecosystem Typology.”
- Keith, David A, José R Ferrer-Paris, Emily Nicholson, and Richard Kingsford. 2020. *IUCN Global Ecosystem Typology 2.0: Descriptive Profiles for Biomes and Ecosystem Functional Groups*. Gland, Switzerland: IUCN. <https://doi.org/10.2305/IUCN.CH.2020.13.en>.
- Keith, David A, Jon Paul Rodríguez, Thomas M Brooks, Mark A Burgman, Edmund G Barrow, Lucie Bland, Patrick J Comer, et al. 2015. “The IUCN Red List of Ecosystems: Motivations, Challenges, and Applications.” *Conservation Letters* 8 (3): 214–26.
- Keith, David A, Jon Paul Rodríguez, Kathryn M Rodríguez-Clark, Emily Nicholson, Kaisu Aapala, Alfonso Alonso, Marianne Asmussen, et al. 2013. “Scientific Foundations for an IUCN Red List of Ecosystems.” *PLOS One* 8 (5): e62111.
- Murray, Nicholas J, David A Keith, Lucie M Bland, Renata Ferrari, Mitchell B Lyons, Richard Lucas, Nathalie Pettorelli, and Emily Nicholson. 2018. “The Role of Satellite Remote Sensing in Structured Ecosystem Risk Assessments.” *Science of the Total Environment* 619: 249–57.
- Murray, Nicholas J, David A Keith, Lucie M Bland, Emily Nicholson, Tracey J Regan, Jon Paul Rodríguez, and Michael Bedward. 2017. “The Use of Range Size to Assess Risks to Biodiversity from Stochastic Threats.” *Diversity and Distributions* 23 (5): 474–83.
- Nicholson, Emily, David A Keith, and David S Wilcove. 2009. “Assessing the Threat Status of Ecological Communities.” *Conservation Biology* 23 (2): 259–74.

- Rodríguez, Jon Paul, David A Keith, Kathryn M Rodríguez-Clark, Nicholas J Murray, Emily Nicholson, Tracey J Regan, Rebecca M Miller, et al. 2015. "A Practical Guide to the Application of the IUCN Red List of Ecosystems Criteria." *Philosophical Transactions of the Royal Society B: Biological Sciences* 370 (1662): 20140003.
- Rodríguez, Jon Paul, KATHRYN M RODRÍGUEZ-CLARK, Jonathan EM Baillie, Neville Ash, John Benson, Timothy Boucher, Claire Brown, et al. 2011. "Establishing IUCN Red List Criteria for Threatened Ecosystems." *Conservation Biology* 25 (1): 21–29.