UNIVERSITY OF CALIFORNIA

Santa Barbara

Turning Data into Action. An R Package for the Wildfire Resilience Index

Technical Documentation

A capstone project submitted in partial satisfaction of the requirements for the degree of Master of Environmental Data Science
for the
Bren School of Environmental Science & Management

Authors:

Ixel Medrano
Emily Miller
Hylaea Miller
Kaiju Morquecho

Advisors:

Dr. Max Czapanskiy
Dr. Carmen Galaz García

Client:

Dr. Caitlin Fong
National Center of Ecological Analysis and Synthesis (NCEAS)

Spring 2026

TURNING DATA INTO ACTION. AN R PACKAGE FOR THE WILDFIRE RESILIENCE INDEX

As developers of this Capstone Project documentation, we archive this documentation on the Bren School's website such that the results of our research are available for all to read. Our signatures on the document signify our joint responsibility to fulfill the archiving standards set by the Bren School of Environmental Science & Management.

_____

IXEL MEDRANO

_____

EMILY MILLER

_____

HYLAEA MILLER

_____

KAIJU MORQUECHO

The Bren School of Environmental Science & Management produces professionals with unrivaled training in environmental science and management who will devote their unique skills to the diagnosis, assessment, mitigation, prevention, and remedy of the environmental problems of today and the future. A guiding principle of the School is that the analysis of environmental problems requires quantitative training in more than one discipline and an awareness of the physical, biological, social, political, and economic consequences that arise from scientific or technological decisions.

The Capstone Project is required of all students in the Master of Environmental Data Science (MEDS) Program. The project is a six-month-long activity in which small groups of students contribute to data science practices, products or analyses that address a challenge or need related to a specific environmental issue. This MEDS Capstone Project Technical Documentation is authored by MEDS students and has been reviewed and approved by:

_____

CARMEN GALAZ GARCÍA

_____

MAX CZAPANSKIY

_____

DATE

Acknowledgements

We thank Dr. Caitlin Fong of the National Center for Ecological Analysis and Synthesis (NCEAS) for serving as our project client and for her guidance throughout the year.

We thank our faculty advisor, Dr. Max Czapanskiy, and capstone instructor Dr. Carmen Galaz García for their technical input and feedback on package design, infrastructure, and documentation.

We thank the Wildfire Resilience Index (WRI) team at NCEAS for producing the underlying dataset and for their responsiveness to our questions about layer provenance and methodology. The WRI is funded by the Gordon and Betty Moore Foundation as part of the Wildfire Resilience Initiative.

We thank the staff at the Knowledge Network for Biocomplexity (KNB) for hosting the COG archive that powers the package.

We thank the faculty and staff of the Bren School Master of Environmental Data Science program for their instruction and support across the year.

Abstract

The Wildfire Resilience Index (WRI), hosted at the National Center for Ecological Analysis and Synthesis and funded by the Gordon and Betty Moore Foundation, is an open-access geospatial tool designed to measure social and environmental dimensions of wildfire resilience across western North America. The dataset combines social and ecological domains into composite resilience scores at 90 meter resolution, covering twelve western U.S. states, British Columbia, and the Yukon Territory, and comprises 82 raster layers totaling approximately 25 gigabytes. This size places the data out of practical reach for many intended users, including resource managers, decision makers, and community leaders without access to high-performance computing resources. To address this barrier, the `ignitR` project converted the source rasters to Cloud-Optimized GeoTIFF format, hosted them on the Knowledge Network for Biocomplexity, and developed `firex`, an R package that lets users query and download spatial subsets of WRI layers without downloading the full dataset. The package supports bounding box and polygon inputs, with named-region queries documented as a planned extension. The `firex` package makes the WRI usable on any machine that can run R, opening the dataset to a broader community of decision-makers and researchers.

Table of Contents

Acknowledgements ---- **iii**
Abstract ---- **iv**
Table of Contents ---- **v**
Executive Summary ---- **1**
Background and Motivation ---- 1
Problem statement ---- 1
Objectives ---- 1
Summary of Methods and Results ---- 2
References ---- 3
Approach ---- **3**
Methods ---- **4**
Data Description and Provenance ---- 4
COG Conversion ---- 5
STAC Catalog Creation ---- 5
Hosting ---- 6
Tools, Libraries, and Infrastructure ---- 6
Programming Languages and Libraries ---- 6
Version Control and Code Management ---- 7
Computational Resources and Infrastructure ---- 7
Quality Control and Validation ---- 7
Dependencies, Installation, and Testing ---- **8**
Dependencies ---- 8
Package Installation ---- 8
Unit Testing ---- 9
User Testing ---- 10
Package Description ---- **10**
Design and Functionality ---- 10
Performance and Limitations ---- 11
User Manual ---- **11**
User Guide: Installation and quick start ---- 11
Navigation and Key Features (vignettes) ---- 12
Troubleshooting and Support: package maintenance, support contact information 22
Archive Access ---- **23**
Code ---- 23
Data ---- 23
References ---- **23**
Appendix ---- **26**

Executive Summary

Background and Motivation

Wildfire is a natural and necessary part of western North American landscapes (Bond & Keeley, 2005). Unfortunately, decades of fire suppression, land use change, and climate-driven shifts in temperature and precipitation have altered fire regimes across the region (Abatzoglou & Williams, 2016; Moritz et al., 2014). Wildfires are now burning more frequently, at greater intensity, and across a broader geographic extent than at any point in the modern record, posing escalating risks to communities, infrastructure, ecosystems, and air and water quality across 12 western US states, British Columbia, and the Yukon Territory (Cunningham et al., 2024).

As wildfire activity increases across the western part of North America, resource managers and community leaders need tools that reveal where those risks exist and how well social and ecological systems are positioned to resist and recover from fire. The Wildfire Resilience Index (WRI) was created to meet that need by integrating 82 geospatial data layers at 90-meter resolution into a holistic measure of resilience across eight social and environmental domains (Fong et al., 2026). However, the size and complexity of the dataset make it difficult for users to access, as downloading multi-gigabytes of raster data, managing geospatial formats, and extracting relevant layers require technical capacity that many agencies and practitioners lack.

Problem statement

This project aims to bridge the accessibility gap for resource managers, researchers, and community organizations who are limited by the storage, bandwidth, and geospatial processing requirements of the original WRI archive.

Objectives

1. Make the Full Dataset Cloud-Accessible

Goal: Convert and host all WRI raster layers in a format that supports efficient cloud-native access.

Implementation: Reproducible scripts in a dedicated data processing repository for Tagged Image File Format (TIFF) to Cloud-Optimized GeoTIFF (COG) (Cloud-Native Geospatial Forum, 2021) conversion and SpatioTemporal Asset Catalog (STAC) (STAC Specification, 2021) generation. The resulting COG archive and catalog are accessed by the `firex` R package, described in Objective 4.

2. Enable Programmatic Layer Discovery

Goal: Let users discover and filter any of the 82 WRI layers by domain and metadata without manual file inspection.

Implementation: JavaScript Object Notation (JSON)-formatted catalog that organizes all WRI layers by eight domain categories with standardized metadata, enabling efficient data discovery and programmatic querying.

3. Eliminate Local Storage Requirements

Goal: Remove the need for users to store or manage multi-gigabyte files locally.

Implementation: Complete set of 82 WRI raster layers converted to COG format and hosted on Knowledge Network for Biocomplexity (KNB).

4. Provide an Accessible R Interface

Goal: Provide a simple R package for discovering layers and retrieving spatial subsets without requiring users to manage raw rasters.

Implementation: User-facing functions for catalog discovery (`wri_overview()`, `wri_overview_df()`, `layer_info()`) and data retrieval (`get_layer()`) with bounding box and polygon area-of-interest support, plus package documentation and vignettes.

5. Ensure Long-Term Usability and Maintainability

Goal: Support long-term usability by both end users and the National Center for Ecological Analysis and Synthesis team responsible for maintenance after project completion.

Implementation: User guide for end users, technical documentation for maintainers, and handoff materials with known limitations and maintenance recommendations.

Summary of Methods and Results

The ignitR project comprises two components: a data infrastructure backend that prepares and hosts the WRI archive in a cloud-accessible format, and `firex`, an R package that provides a user-facing interface to that archive. Together, they form an end-to-end system in which the backend makes efficient spatial access possible, and the package makes it simple.

A reproducible pipeline ingests the source WRI rasters, validates each layer, converts it to COG format with internal overviews and tiling, and the results are then uploaded to KNB. A second pipeline reads the published archive and generates a STAC catalog with layer-level metadata extracted directly from the COG headers. The `firex` package provides a consistent R interface to this archive. Users can browse the catalog, retrieve per-layer metadata, and download spatial subsets by bounding box or polygon boundary

through a shared `get_layer()` entry point. The function accepts numeric bounding boxes, file paths, and common spatial object classes (`sf`, `SpatVector`, `SpatRaster`, `SpatExtent` (Pebesma, 2018; Hijmans, 2025)), handling coordinate reference system validation and extent checking internally. No local data download or storage is required.

References

Abatzoglou, J. T., & Williams, A. P. (2016). Impact of anthropogenic climate change on wildfire across western US forests. Proceedings of the National Academy of Sciences, 113(42), 11770–11775. https://doi.org/10.1073/pnas.1607171113

Bond, W. J., & Keeley, J. E. (2005). Fire as a global 'herbivore': The ecology and evolution of flammable ecosystems. Trends in Ecology & Evolution, 20(7), 387–394. https://doi.org/10.1016/j.tree.2005.04.025

Cloud-Native Geospatial Forum (2021). Cloud-Optimized GeoTIFF: An Overview. https://cogeo.org

Cunningham, C. X., Williamson, G. J., & Bowman, D. M. J. S. (2024). Increasing frequency and intensity of the most extreme wildfires on Earth. Nature Ecology & Evolution, 8, 1420–1425. https://doi.org/10.1038/s41559-024-02452-2

Fong, C. R., Halpern, B. S., & the Western Wildfire Resilience Index team. (2026). Western Wildfire Resilience Index. National Center for Ecological Analysis and Synthesis, UC Santa Barbara. https://doi.org/10.5063/F1V40SQH

Hijmans, R.J. (2025). `terra`: Spatial Data Analysis. R package version 1.8. https://CRAN.R-project.org/package=terra

Moritz, M. A., Batllori, E., Bradstock, R. A., Gill, A. M., Handmer, J., Hessburg, P. F., Leonard, J., McCaffrey, S., Odion, D. C., Schoennagel, T., & Syphard, A. D. (2014). Learning to coexist with wildfire. Nature, 515, 58–66. https://doi.org/10.1038/nature13946

STAC Specification. (2021). SpatioTemporal Asset Catalog specification, v1.0.0. https://github.com/radiantearth/stac-spec

Approach

The project was completed in three sequential phases. The first phase produced a data processing pipeline that ingested the 82 source raster layers from the WRI team at NCEAS, validated each one, and converted it to COG format. The second phase published the resulting COG archive to the Knowledge Network for Biocomplexity and generated a SpatioTemporal Asset Catalog from the published archive. The third phase produced `firex`, an R package that reads the STAC catalog and provides user-facing functions for discovering and subsetting WRI layers. Package development was organized into three tiers of functionality, with each tier corresponding to a milestone reviewed with the project advisor and client.

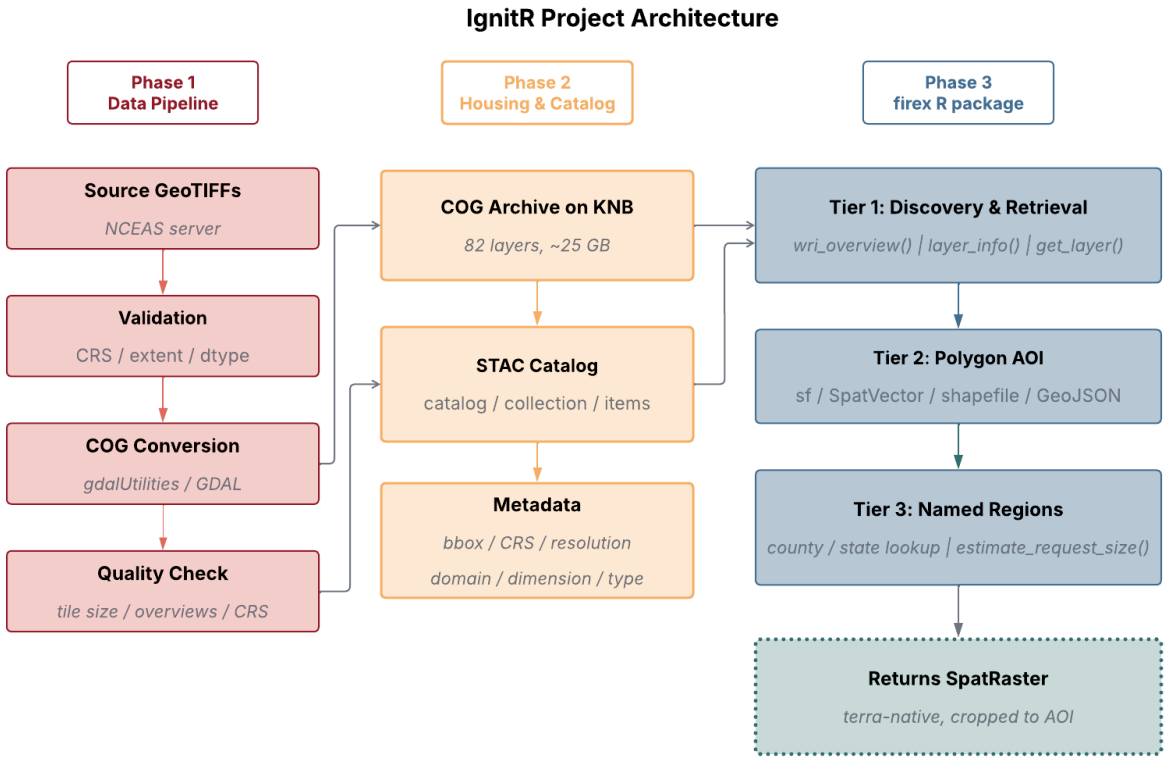

Figure 1. `ignitR` project architecture. Phase 1 converts source GeoTIFFs to COG format. Phase 2 hosts the archive on KNB and generates the STAC catalog. Phase 3 exposes the data through the `firex` R package across three tiers of functionality.

Methods

Data Description and Provenance

The WRI project is hosted at the NCEAS and funded by the Gordon and Betty Moore Foundation as part of the Wildfire Resilience Initiative. The WRI is the first open-access tool designed to measure social and environmental dimensions of wildfire resilience across western North America, and is intended to support resource managers, decision makers, community leaders, and researchers in understanding local opportunities and liabilities related to wildfire resistance and recovery.

The dataset consists of 82 raster layers organized into social and ecological domains, which are combined into composite resilience scores. All layers are aligned to the same 90 meter resolution grid and projected to EPSG:5070 (NAD83 / Conus Albers), and cover twelve western U.S. states, British Columbia, and the Yukon Territory.

Layers fall into two types. **Indicator** layers are derived directly from source datasets, such as census records, air quality monitoring, or species range maps, and represent a single measurable dimension of resilience. **Aggregate** layers are composite scores computed from multiple indicators within a domain; depending on the domain, these are aggregated into resistance, resilience, status, and/or an overall domain score. The final WRI score aggregates across all domain scores into a single composite value.

Each layer is also classified by dimension, which describes what aspect of resilience it measures. **Status** describes the current condition of a system relative to a reference or goal, for example, current streamflow relative to a 30-year historical baseline, or current species extinction risk. **Resistance** captures the ability of a system to withstand wildfire, and **recovery** captures the ability to regain function afterward. **Resilience** combines resistance and recovery into a single score, formulated such that high values in either component can confer resilience without requiring both. **Domain score** is the average of status and resilience for a given domain, and the final WRI score is the average of all domain scores. These three fields — domain, dimension, and type — form the metadata schema used to organize and query layers in the STAC catalog.

The source data was provided by the WRI team as a set of GeoTIFF files. The full archive totals approximately 25 gigabytes, with individual layer sizes ranging from under 100 megabytes to nearly three gigabytes. (See Table 1 in the Appendix)

COG Conversion

Each source raster was converted to COG format using the GDAL command-line tools. The conversion process applied internal tiling, Lempel-Ziv-Welch lossless compression,

and pre-built overviews. Pre-built overviews are essential for cloud-native access, because without them, an overview request from a remote client triggers a download of the full file rather than a small range request.

The pipeline processes all WRI layers in a single reproducible batch run. Each output COG was validated against a checklist of structural properties, including correct tile size, presence of overviews, and matching CRS.

STAC Catalog Creation

After the COG archive was uploaded to KNB, a second pipeline read the published archive and generated a STAC. The catalog contains one item per layer, with metadata fields extracted directly from the COG headers, including bounding box, CRS, resolution, and band descriptions. Generating the catalog from the published archive, rather than from the source files, ensures that the catalog reflects exactly what users will retrieve.

Hosting

The COG archive and STAC catalog are hosted on KNB, a member node of the DataONE federation. KNB provides persistent identifiers, long-term storage, and a stable HTTPS endpoint that supports byte-range requests, which is the access pattern `firex` relies on for spatial subsetting.

Tools, Libraries, and Infrastructure

Programming Languages and Libraries

COG conversion was scripted in R using `gdalUtilities` (O'Brien, 2023), which wraps GDAL's (GDAL Contributors, 2024) `gdal_translate` command-line utility with the COG driver, enabling internal tiling and efficient downstream access. The conversion pipeline iterated over validated input files and produced one COG per layer in a single reproducible batch run. STAC catalog generation was handled using `readr` (Wickham et al., 2023) and `jsonlite` (Ooms, 2014). STAC Items require a `geometry` field as a GeoJSON polygon in WGS84; we used `sf` (Pebesma, 2018; Pebesma & Bivand, 2023) to transform each raster's footprint from its native `EPSG:5070` extent to `EPSG:4326` and construct the required `bbox` and `geometry` fields. Asset paths were assigned dynamically based on whether data were locally stored or remotely hosted. These

libraries are dependencies of the data processing pipeline only and are not required by the `firex` package: `gdalUtilities`, `readr`, `jsonlite`, and `sf`.

The `firex` package was deliberately designed with a minimal dependency footprint. Its only hard dependencies are `terra` (Hijmans et al., 2026), used for raster input/output, CRS normalization, and extent reprojection during AOI validation, and `rstac` (Appel & Pebesma, 2023) for reading the STAC catalog. Both are available on CRAN and installed automatically as package dependencies. The package leverages `/vsicurl/` from the GDAL Virtual File Systems to open hosted COGs via HTTP range requests, retrieving only the tiles that overlap a user's area of interest rather than downloading the full file.

Version Control and Code Management

Code was managed using **Git** and hosted on **GitHub**. The project was organized into two repositories:

- A **data processing repository** for generating COGs and building the STAC catalog
- A **package repository** containing the R package used to access and interact with the data

This separation improved modularity and reproducibility by clearly distinguishing between data engineering and package development components.

Computational Resources and Infrastructure

All computation and development were performed on local machines, including ThinkPad laptops and Apple Silicon MacBook Pro systems (M3). The workflow did not require cloud-based or high-performance computational infrastructure, as data processing tasks were handled efficiently within standard personal computing environments.

However, the system relies on remote data infrastructure for storage and access. Initial raw GeoTIFF datasets were accessed via the NCEAS server. After processing and conversion to COGs, datasets were uploaded to the KNB repository, where they became remotely accessible.

The package was designed to support a hybrid data access model, allowing it to retrieve data from either remotely hosted sources (KNB) or local file systems. This design enables seamless transitions between development and production workflows and supports incremental data publication.

To optimize performance, the KNB/NCEAS infrastructure team enabled HTTP range request support for hosted COGs. Using `/vsicurl/`, the package can request only specific portions (tiles) of large raster datasets rather than downloading entire files, significantly improving efficiency and reducing bandwidth usage.

Quality Control and Validation

Several quality control measures were implemented to ensure data integrity and code reliability. Metadata validation checks were incorporated into the STAC generation pipeline, including verification of required fields, detection of duplicate filenames, and enforcement of consistent spatial extent and coordinate reference system (CRS) information.

The COG conversion pipeline was designed to be rerunnable and fault-tolerant, with safeguards such as:

- Skipping files that had already been converted
- Checking for missing input files prior to processing
- Catching and logging errors to allow batch processing to continue

Code quality was further supported through collaborative code reviews conducted within the team and with a faculty advisor. Automated testing of the `firex` package was implemented using the `testthat` (Wickham, 2023) package, with tests developed and executed via `devtools`. In addition, standard R package validation workflows were used, including `R CMD build` and `R CMD check`, to ensure the package met structural, documentation, and dependency requirements (Wickham & Bryan, 2023).

Beyond automated testing, the package was also manually tested through end-to-end workflows, including loading STAC items, retrieving raster subsets, and verifying correct behavior for both locally stored and remotely hosted data.

To ensure metadata consistency and freshness, the package includes functionality to recreate the STAC catalog on demand when users request metadata. This design ensures that the most up-to-date catalog is used, particularly as new COGs are added or hosting status changes and reduces the risk of stale or inconsistent metadata.

Dependencies, Installation, and Testing

Dependencies

The `firex` package requires R version 4.1 or later. The package has two hard dependencies: `terra` for raster input/output, coordinate transformations, and spatial validation; and `rstac` for reading the STAC catalog. Both are installed automatically from CRAN when `firex` is installed.

Package Installation

The package is distributed through GitHub and can be installed using the `devtools` or `remotes` package:

```
install.packages("remotes")
remotes::install_github("ignitR-package/firex")
```

Once installed, the package is ready for use. No additional configuration, API keys, or credentials are required.

Unit Testing

Quality assurance occurred throughout package development, with unit tests implemented using the `testthat` package to validate individual function behavior. These tests provide controlled inputs to package functions and use expectations to confirm that returned values, attributes, classes, and error conditions match the intended outcomes. The package includes tests organized in `tests/testthat/` for area-of-interest handling, CRS validation, STAC catalog queries, metadata lookup, layer retrieval, and printed catalog summaries. Tests for `resolve_aoi()` verify invalid bounding boxes, coordinate ordering, unsupported input types, and accepted spatial inputs such as shapefiles, GeoJSON files, `sf` objects, and `terra` objects. Tests for `resolve_aoi_crs()` confirm CRS normalization, missing CRS errors, and CRS conflict handling. Tests for `check_extent_overlap()` verify fully contained, partially outside, and non-overlapping bounding boxes. Tests for `layer_info()`, `wri_overview()`, and `wri_overview_df()` check input validation, output classes, required field names, and that the local STAC catalog path is reachable. Tests for `get_layer()` cover input validation and area-of-interest checks; optional remote COG retrieval tests are guarded by the `FIREX_RUN_REMOTE_TESTS` environment variable and confirm that `get_layer()` returns a single-band `SpatRaster` in `EPSG:5070` for a small bounding box. The standard test suite runs locally with

`devtools::test()` and is executed automatically on push and pull requests via GitHub Actions.

User Testing

User testing was conducted with team members to ensure that the package is intuitive for users with moderate R experience. The testing focused on how users interacted with the functions, including entering different types of inputs, interpreting error messages, and navigating the documentation. Feedback from these sessions informed improvements to function naming, argument structure, error handling, and overall usability.

Package Description

Design and Functionality

The primary goal of the package is to enable customized, user-friendly access to the Wildfire Resilience Index geospatial data without requiring advanced knowledge of geospatial science, the dataset, or R. The package is designed to eliminate the need for manual data requests by acting as a direct interface between users and the dataset, effectively removing the client as an intermediary in the data access process. This replaces a workflow that previously required mediated access through the NCEAS server.

At its core, the package provides a streamlined workflow for discovering, querying, and retrieving spatial data layers stored as COGs. The main user-facing functions are:

- `wri_overview()`, which returns a structured summary of the STAC catalog
- `wri_overview_df()`, which provides a flattened data frame of all available layers and their metadata
- `layer_info()`, which retrieves metadata for a single data layer
- `get_layer()`, which retrieves the actual data and serves as the primary deliverable of the package
    - The user can request a specific layer and define their own area of interest.
    - When an extent is supplied, the function returns the layer cropped to that spatial region, rather than the full data layer, avoiding unnecessary data handling or computational overhead for the user.
    - The function accepts an extent in multiple formats (e.g., bounding boxes, polygons, and other spatial inputs).

Together, these functions provide an end-to-end workflow that supports both dataset discovery and targeted data retrieval. The design prioritizes user accessibility by abstracting complex geospatial operations such as coordinate transformations, spatial validation, and partial data retrieval into simple function calls.

Performance and Limitations

The performance optimization focused on improving query efficiency and minimizing data transfer requirements when accessing the Wildfire Resilience Index data. Some limitations may arise, such as slower processing times, depending on network speed when accessing database layers, the size of the subsetted data being retrieved, and the user's local computing resources.

User Manual

User Guide: Installation and quick start

This guide is intended for users who want to install `firex` and retrieve their first layer in under five minutes.

System requirements: R version 4.1 or later, an internet connection, and approximately 200 MB of free disk space for the package and its dependencies.

Install `firex` from GitHub (requires the `remotes` package):

```
# Install once
install.packages("remotes")
remotes::install_github("ignitR-package/firex")
```

Then load the package:

```
library(firex)
library(terra)

# 1. Browse all 82 available layers
wri_overview()

# 2. Get metadata for a specific layer
layer_info("WRI_score")

# 3. Retrieve the WRI composite score for Santa Barbara County, CA
wri <- get_layer("WRI_score",
                 aoi     = c(-120.7, 34.4, -119.4, 35.1),
```

```
                    aoi_crs = "EPSG:4326")
plot(wri, col = hcl.colors(100, "YlOrRd", rev = TRUE))
```

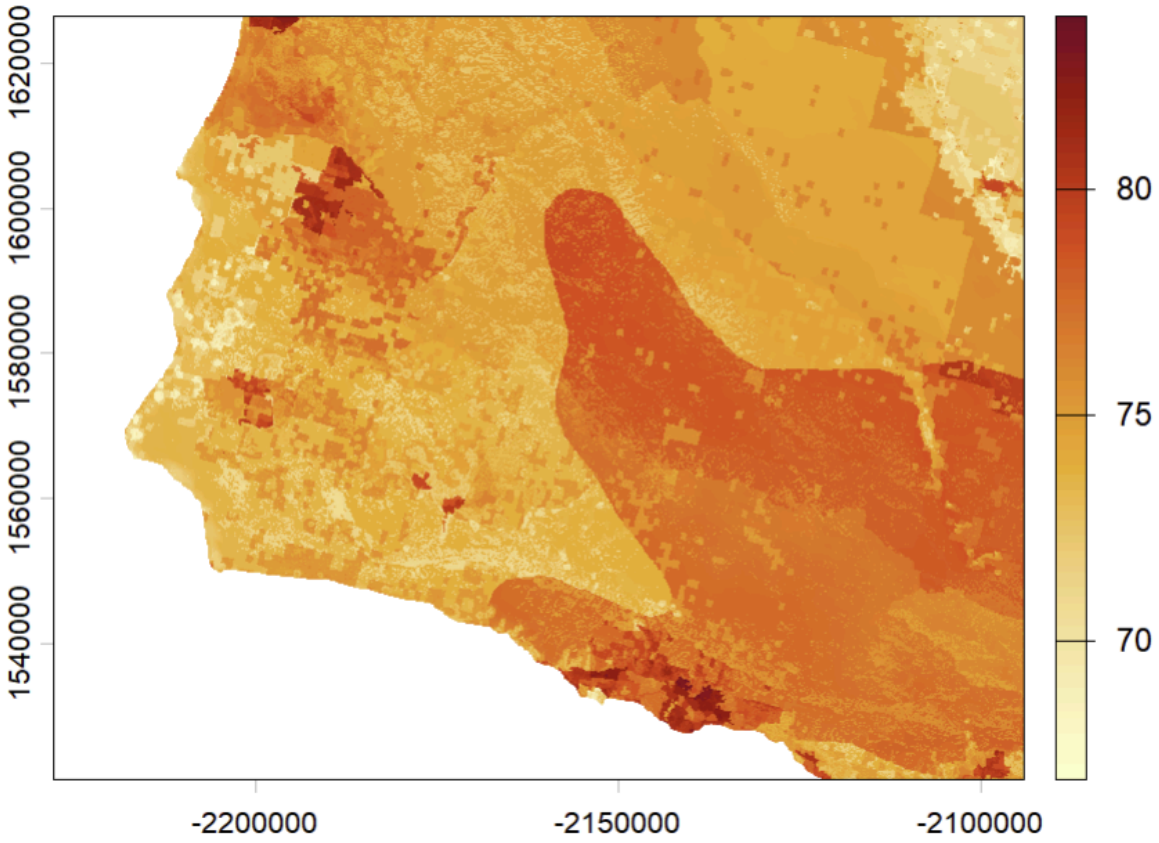

Figure 2. Output of `get_layer("WRI_score", aoi = c(-122, 37, -121, 38), aoi_crs = "EPSG:4326")`. The function returns a `terra SpatRaster` cropped to the requested bounding box, projected in `EPSG:5070` (NAD83 / Conus Albers).

Navigation and Key Features (vignettes)

This section walks through the main features of `firex` through a set of vignettes. Each vignette is also distributed with the package and accessible through `browseVignettes("firex")`.

Vignette 1: Discovering Available Layers

This vignette shows how to inspect the WRI catalog before downloading data. Users start by listing available layers, then narrow the catalog to the layer ID they need. Catalog browsing reads the local STAC index bundled with the package, so this step does not require a large data download.

Setup

Load `firex`. No other packages are needed for catalog browsing.

```
library(firex)
```

Browsing the full catalog

`wri_overview()` reads the local STAC index and prints a structured summary of all 82 available layers. Call it with no arguments to see all domains, dimensions, and data types at a glance:

```
> wri_overview()
WRI DATA SUMMARY
--------------------
Read at:      2026-05-11 16:07:50
Layer ID Naming Convention: wri_domain + wri_dimension + data_type
Collections (1): wri_ignitR
Unique wri_domain values (11): air_quality, communities, iconic_places,
iconic_species, infrastructure, livelihoods, natural_habitats,
sense_of_place, species, unknown, water
Unique data_type values (3): aggregate, final_score, indicator
Unique wri_dimension values (5): domain_score, recovery, resilience,
resistance, status
Total layers available: 82
```

Figure 3. Console output of `wri_overview()`. The object summarizes the 82 available layers grouped by domain, dimension, and data type.

Getting the catalog as a data frame

`wri_overview_df()` returns the same information as a flat data frame, one row per layer. This is useful for filtering with standard R tools:

```
df <- wri_overview_df()
nrow(df)
#> [1] 82

# Filter to water-domain aggregate layers
subset(df, wri_domain == "water" & data_type == "aggregate",
       select = c(id, wri_dimension, data_type))
#>                    id wri_dimension data_type
#> 75 water_domain_score  domain_score aggregate
#> 76   water_resilience    resilience aggregate
#> 77   water_resistance    resistance aggregate
```

```
#> 80       water_status        status aggregate
```

The ten wri_domain values are: `air_quality`, `communities`, `iconic_places`, `iconic_species`, `infrastructure`, `livelihoods`, `natural_habitats`, `sense_of_place`, `species`, and `water`. The five `wri_dimension` values are: `domain_score`, `recovery`, `resilience`, `resistance`, and `status`.

Inspecting a single layer

Once you have identified a layer ID, `layer_info()` retrieves its full metadata record, including the hosted COG URL, geographic extent, CRS, and classification properties. Transposing with `t()` makes it easier to read:

```
t(layer_info("WRI_score"))
#>            WRI_score
#> id         WRI_score
#> collection wri_ignitR
#> asset_href https://knb.ecoinformatics.org/.../WRI_score.tif
#> proj_code  EPSG:5070
#> data_type  final_score
#> wri_domain unknown
#> is_hosted  TRUE
#> xmin       -146.2082
#> ymin        19.1074
#> xmax       173.7109
#> ymax        54.8056
```

The `asset_href` field is the direct URL to the hosted COG file on KNB. `get_layer()` uses this URL to open a virtual connection via GDAL; you do not need to use it directly. The `xmin`/`ymin`/`xmax`/`ymax` fields give the layer's geographic extent in WGS84 (`EPSG:4326`); all COGs share the same extent.

Advanced: querying STAC items directly

For filtering on combinations of properties not easily expressed as data frame queries, `query_stac_flexible()` filters the underlying STAC item list by any property value:

```
catalog <- wri_overview()
items   <- lapply(catalog$data$items, function(x) x$item)

# All resistance indicators in the water domain
water_resist <- query_stac_flexible(items,
```

```
                                    wri_domain    = "water",
                                    wri_dimension = "resistance",
                                    data_type     = "indicator")
vapply(water_resist, function(x) x$id, character(1))
#> [1] "water_resistance_drought_plans"
#> [2] "water_resistance_water_treatment"
```

Once users identify a layer ID, they can pass that ID to `get_layer()` in Vignette 2 or use an existing spatial object as the area of interest in Vignette 3.

Vignette 2: Retrieving a Layer by Bounding Box

This vignette shows the simplest data retrieval workflow: request one layer for one rectangular area of interest. `get_layer()` opens the hosted COG through GDAL and uses range requests so the user does not need to download the full raster file.

Setup

```
library(firex)
library(terra)
```

The basic call

`get_layer()` takes a layer ID, an `aoi` argument, and for numeric bounding boxes an explicit `aoi_crs`. Supply the bounding box as `c(xmin, ymin, xmax, ymax)`. The function checks overlap against the layer extent and returns a `terra` raster in the COG native projection.

```
# Santa Barbara County, CA in WGS84
sb  <- c(-120.7, 34.4, -119.4, 35.1)
wri <- get_layer("WRI_score", aoi = sb, aoi_crs = "EPSG:4326")
wri
#> class       : SpatRaster
#> dimensions  : 482, 756, 1  (nrow, ncol, nlyr)
#> resolution  : 90, 90  (x, y)
#> extent      : -1766635, -1698535, 527375, 570755
#> coord. ref. : NAD83 / Conus Albers (EPSG:5070)
#> name        : WRI_score
```

Plotting the result

```
plot(wri,
     main = "WRI Score -- Santa Barbara County",
```

```
    col  = hcl.colors(100, "YlOrRd", rev = TRUE))
```

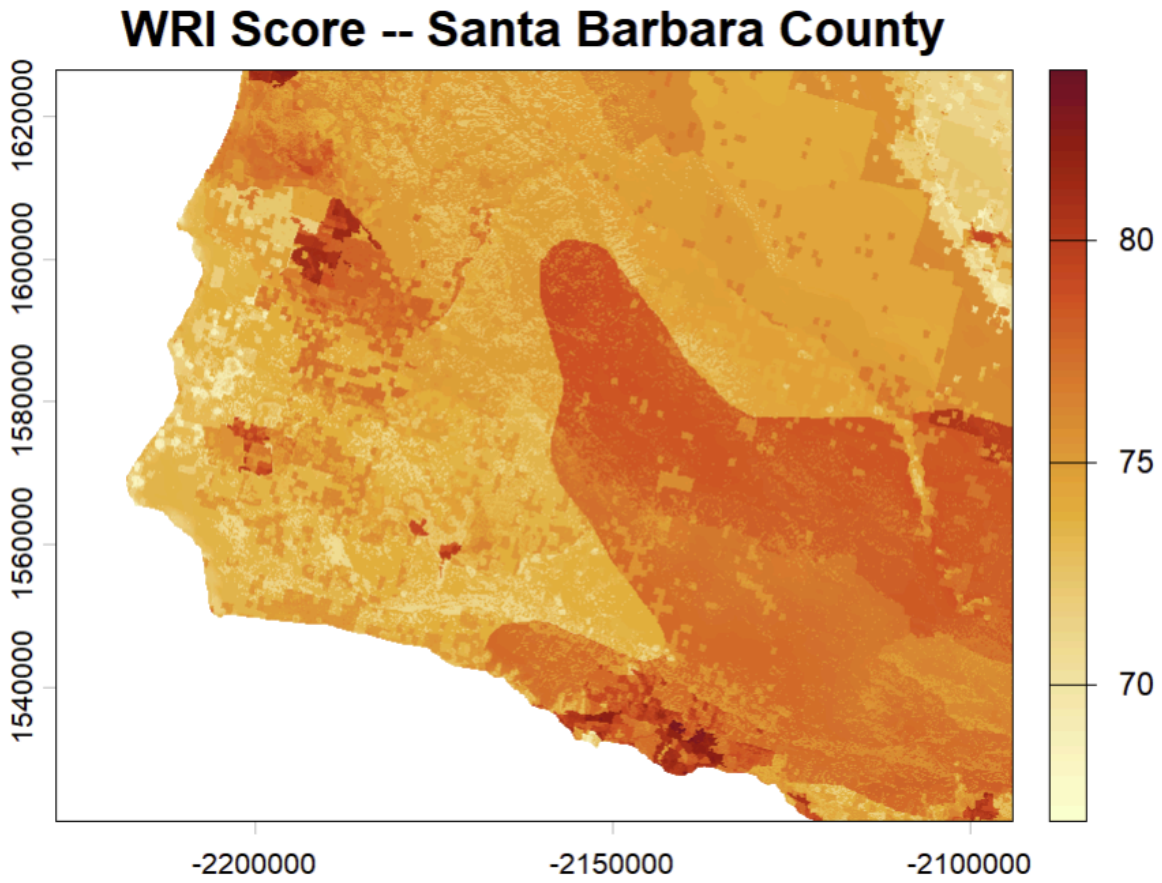

Figure 4. WRI score for a 1 degree by 1 degree bounding box centered on northern California, returned by `get_layer("WRI_score", aoi = c(-122, 37, -121, 38), aoi_crs = "EPSG:4326")`. Higher values (green) indicate greater overall wildfire resilience.

Working with the result

Because `get_layer()` returns a standard `SpatRaster`, the full `terra` ecosystem is immediately available for analysis:

```
# Summary statistics
global(wri, fun = c("mean", "sd", "min", "max"), na.rm = TRUE)
#>    mean    sd  min   max
#>   0.531 0.112 0.18 0.894

# Write to disk as a GeoTIFF
writeRaster(wri, "wri_score_sb.tif")

# Reproject to WGS84 for export to other tools
wri_wgs84 <- project(wri, "EPSG:4326")
```

CRS shorthand

`aoi_crs` accepts any format that `terra` recognizes: an integer EPSG code, a character string, or a PROJ or WKT string. These are all equivalent:

```
get_layer("WRI_score", aoi = sb, aoi_crs = 4326)
get_layer("WRI_score", aoi = sb, aoi_crs = "EPSG:4326")
get_layer("WRI_score", aoi = sb, aoi_crs = "WGS84")
```

Retrieving multiple layers for the same area

Use `lapply()` to iterate over a vector of layer IDs. All layers share the same extent and resolution once retrieved, so the results can be stacked directly with `terra::c()`:

```
water_ids <- subset(wri_overview_df(),
                    wri_domain == "water" & data_type == "aggregate")$id

water_layers <- lapply(water_ids, get_layer,
                       aoi = sb, aoi_crs = "EPSG:4326")
names(water_layers) <- water_ids

stack <- do.call(c, water_layers)
plot(stack, col = hcl.colors(100, "YlOrRd", rev = TRUE))
```

Extent overlap and error handling

If the bounding box extends partly outside the COG coverage area, `get_layer()` returns the overlapping portion and issues a warning naming the sides that overflowed:

```
wri_big <- get_layer("WRI_score",
                     aoi     = c(-130, 30, -55, 55),
                     aoi_crs = "EPSG:4326")
#> Warning: Requested bbox extends outside the layer extent to the east.
```

If the bounding box falls entirely outside the covered area, the function stops with an informative error before making any network request:

```
tryCatch(
  get_layer("WRI_score",
            aoi = c(-80, -90, -60, -70), aoi_crs = "EPSG:4326"),
  error = function(e) message("Error: ", conditionMessage(e))
)
#> Error: Requested bbox does not overlap the layer extent.
```

See Vignette 3 to use a polygon boundary in place of a rectangular bounding box, which reduces the returned data to only the cells inside your study area.

Vignette 3: Using Spatial Objects as Areas of Interest

In this vignette, you will learn to pass a spatial object directly to `get_layer()` as the area of interest. This is useful when users already have a study boundary, such as a shapefile, GeoJSON file, sf object, or `terra` object, and do not want to manually enter bounding-box coordinates.

For spatial objects, `get_layer()` reads the object's coordinate reference system, derives its extent, and retrieves the overlapping COG tiles. The returned raster is cropped to the area-of-interest extent. It is not masked to the exact polygon boundary. Users who need exact polygon clipping can apply `terra::mask()` after retrieval.

Setup

```
library(firex)
library(terra)
library(sf)     # only needed if using sf objects
```

Accepted input types

`get_layer()` accepts several area-of-interest input types, including numeric bounding boxes, spatial files, `sf` objects, and `terra` objects. Inputs without stored coordinate reference system information, such as numeric bounding boxes and `SpatExtent` objects, require `aoi_crs`.

Example: shapefile path

Pass the file path as a character string. For a shapefile, the coordinate reference system is read from the .prj sidecar file, so `aoi_crs` is not needed.

```
shp <- system.file("demos/data/Eaton_Perimeter_20250121.shp",
                   package = "firex")

wri_eaton <- get_layer("WRI_score", aoi = shp)

wri_eaton
#> class       : SpatRaster
#> dimensions  : 193, 256, 1  (nrow, ncol, nlyr)
#> resolution  : 90, 90  (x, y)
#> coord. ref. : NAD83 / Conus Albers (EPSG:5070)
```

The returned raster covers the extent of the Eaton Fire perimeter. This avoids manually entering coordinates while keeping the result compatible with terra workflows.

Example: `sf` object

```
eaton   <- sf::st_read(shp, quiet = TRUE)
wri_sf  <- get_layer("WRI_score", aoi = eaton)

# The shapefile path and sf object define the same area of interest.
all.equal(terra::ext(wri_eaton), terra::ext(wri_sf))
#> [1] TRUE
```

This shows that users can choose the input format that best matches their workflow. The package handles the spatial object and derives the retrieval extent internally.

Example: SpatExtent (explicit CRS required)

A `terra::SpatExtent` stores only the bounding coordinates, not the coordinate reference system. Because of this, `aoi_crs` must be supplied explicitly.

```
eaton_v <- terra::vect(shp)
eaton_crs <- terra::crs(eaton_v)
eaton_ext <- terra::ext(eaton_v)

wri_ext <- get_layer("WRI_score",
                     aoi = eaton_ext,
                     aoi_crs = eaton_crs)
```

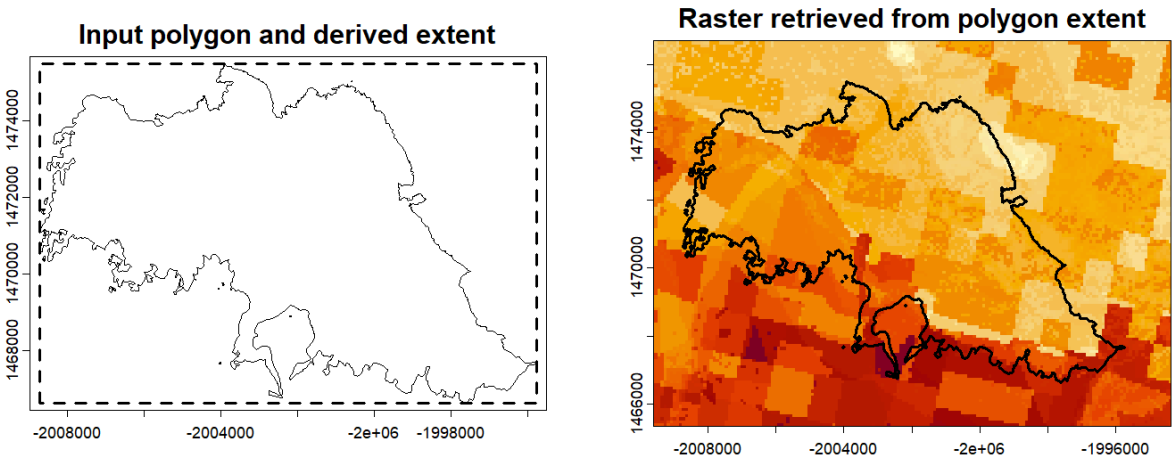

Figure 5. Polygon-based area-of-interest input. Left: the Eaton Fire perimeter and the bounding extent derived from that geometry. Right: `get_layer()` retrieves the `WRI_score` raster over that extent, with the original perimeter overlaid for context. Figure generated from the reproducible script `inst/figures/fig5_bbox_vs_polygon.R`.

Vignette 4: Comparing Layers Across Domains

The WRI measures resilience across eight domains: air quality, communities, infrastructure, livelihoods, natural habitats, sense of place, species, and water. Each domain has status, resistance, resilience, and recovery layers. This vignette shows how to retrieve several layers for the same area and compare them.

Start by browsing the catalog to identify the layers you want:

```
library(firex)
library(terra)

# Get the full catalog as a data frame
df <- wri_overview_df()

# List all water-domain layers
subset(df, wri_domain == "water", select = c(id, wri_dimension, data_type))
#>                              id wri_dimension data_type
#> water_domain_score              domain_score aggregate
#> water_resilience                   resilience aggregate
#> water_resistance                   resistance aggregate
#> water_status                           status aggregate
#> water_status_surface_water_quantity  status indicator
#> water_status_surface_water_timing    status indicator
```

Retrieve two domain-level layers for the same area and compare them side by side. The output of `get_layer()` is a `terra` `SpatRaster`, so standard `terra` operations work directly:

```
# Santa Barbara County, CA (WGS 84)
sb_bbox <- c(-120.7, 34.4, -119.4, 35.1)

water <- get_layer(
"water_domain_score"
, aoi = sb_bbox, aoi_crs =
"EPSG:4326"
)
air   <- get_layer(
"air_quality_domain_score"
, aoi = sb_bbox, aoi_crs =
"EPSG:4326"
)
```

```
# Stack layers and plot together
stack <- c(water, air)
names(stack) <- c("Water domain", "Air quality domain")
plot(stack, col = hcl.colors(100,
"YlOrRd"
, rev = TRUE))

# Compute pixel-level correlation between domains
cor(values(water), values(air), use =
"complete.obs"
)
#> [1]  0.41
```

You can retrieve any number of layers this way. For systematic comparisons, use `lapply()` to iterate over a vector of layer IDs:

```
# Retrieve all aggregate water-domain layers
water_ids <- subset(df, wri_domain ==
"water"
 & data_type ==
"aggregate"
)$id

water_layers <- lapply(water_ids, get_layer, aoi = sb_bbox, aoi_crs =
"EPSG:4326"
)
names(water_layers) <- water_ids

# Inspect dimensions to confirm all layers are aligned
sapply(water_layers, terra::dim)
```

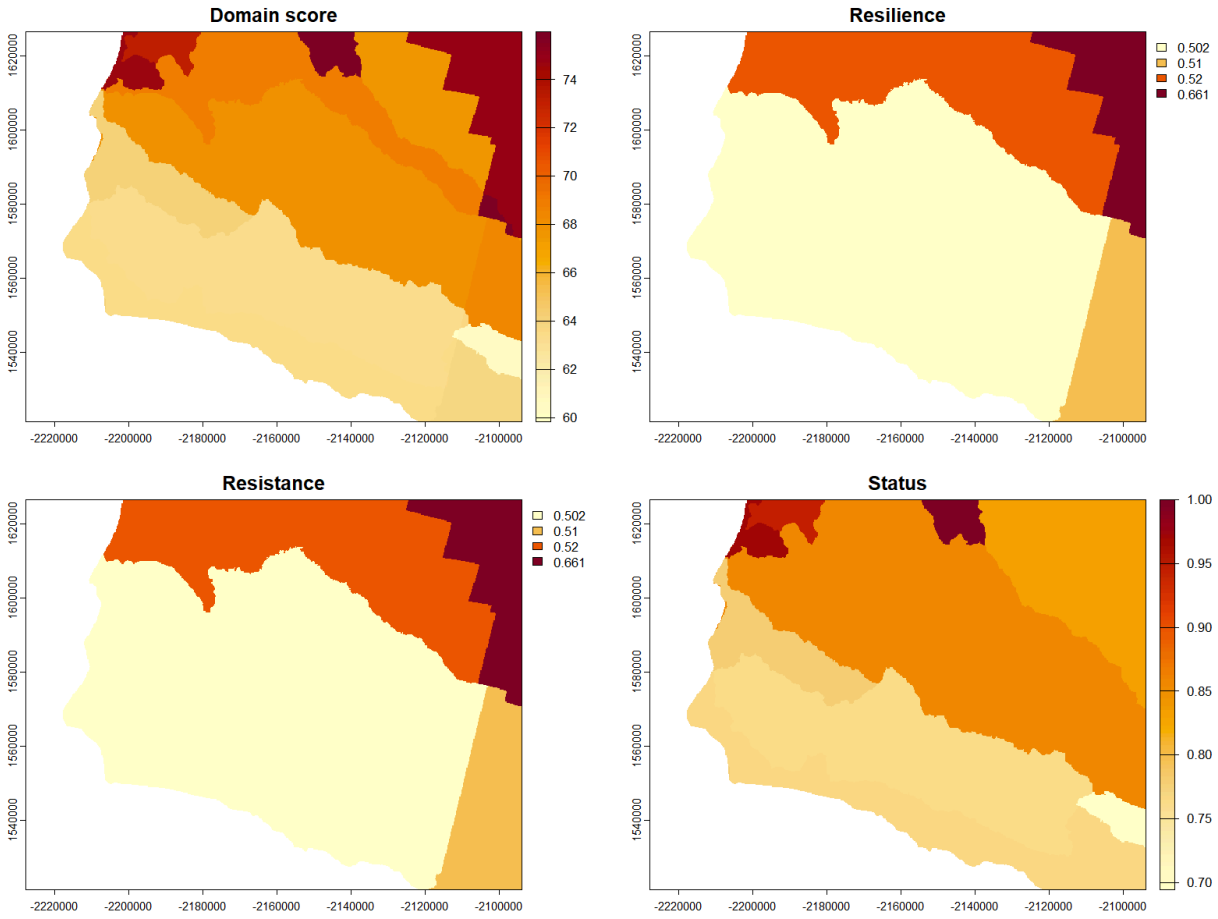

Figure 6. Four aggregate water-domain layers retrieved for Santa Barbara County, CA using the same bounding box and iterated with `lapply()`. Each panel is a `SpatRaster` in `EPSG:5070`; colour scales differ by layer. Figure generated from the reproducible script `inst/figures/fig6_water_domains.R`.

Troubleshooting and Support: package maintenance, support contact information

Common issues and resolutions:

- Network errors during `get_layer()`: check the internet connection and confirm that the KNB archive is reachable. The archive URL can be tested with `wri_overview()`, which performs only a small metadata request.
- Invalid bounding box errors: confirm that coordinates are in `EPSG:4326` (longitude, latitude) and ordered as `c(xmin, ymin, xmax, ymax)`.
- Bounding box does not overlap layer extent: the requested area falls entirely outside the WRI coverage region. Use `wri_overview()` to confirm the layer's extent.

Bug reports and feature requests should be filed on the `firex` GitHub repository issue tracker.

Package maintenance is handled by the project team through Spring Quarter 2026. After project completion, maintenance will transition to NCEAS under the supervision of Dr. Caitlin Fong.

Archive Access

Code

The `firex` package source code is hosted on GitHub at ignitR-package/firex. The repository includes the package source, tests, vignettes, and documentation. The data processing pipeline that produces the COG archive and STAC catalog is hosted at ignitR-package/wri-data-processing.

The `firex` package is released under the MIT License.

Data

The COG archive and STAC catalog are hosted on KNB.

A persistent Digital Object Identifier (DOI) has not yet been assigned because publication depends on formal release of the underlying dataset by the client. Until then, the archive is accessible through the KNB-hosted WRI data-processing directory at https://knb.ecoinformatics.org/data/wri-data-processing/.

References

Abatzoglou, J. T., & Williams, A. P. (2016). Impact of anthropogenic climate change on wildfire across western US forests. Proceedings of the National Academy of Sciences, 113(42), 11770–11775. https://doi.org/10.1073/pnas.1607171113

Appel, M., & Pebesma, E. (2023). rstac: Client library for SpatioTemporal Asset Catalog. R package. https://CRAN.R-project.org/package=rstac

Bond, W. J., & Keeley, J. E. (2005). Fire as a global 'herbivore': The ecology and evolution of flammable ecosystems. Trends in Ecology & Evolution, 20(7), 387–394. https://doi.org/10.1016/j.tree.2005.04.025

Cloud-Native Geospatial Forum. (2021). Cloud-optimized GeoTIFF: An overview. https://cogeo.org

Cunningham, C. X., Williamson, G. J., & Bowman, D. M. J. S. (2024). Increasing frequency and intensity of the most extreme wildfires on Earth. Nature Ecology & Evolution, 8, 1420–1425. https://doi.org/10.1038/s41559-024-02452-2

Fong, C. R., Halpern, B. S., & the Western Wildfire Resilience Index team. (2026). Western Wildfire Resilience Index. National Center for Ecological Analysis and Synthesis, UC Santa Barbara. https://doi.org/10.5063/F1V40SQH

GDAL Contributors. (2024). GDAL: Geospatial Data Abstraction Library. Open Source Geospatial Foundation. https://gdal.org/

Hijmans, R., Brown, A., & Barbosa, M. (2026). terra: Spatial data analysis (R package version 1.9-27). https://CRAN.R-project.org/package=terra

Knowledge Network for Biocomplexity. (2026, April 18). Index of /data/wri-data-processing/cogs. https://knb.ecoinformatics.org/data/wri-data-processing/cogs/

Moritz, M. A., Batllori, E., Bradstock, R. A., Gill, A. M., Handmer, J., Hessburg, P. F., Leonard, J., McCaffrey, S., Odion, D. C., Schoennagel, T., & Syphard, A. D. (2014). Learning to coexist with wildfire. Nature, 515, 58–66. https://doi.org/10.1038/nature13946

O'Brien, J. (2023). gdalUtilities: Wrappers for 'GDAL' utilities executables (R package version 1.2.5). https://CRAN.R-project.org/package=gdalUtilities

Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between JSON data and R objects. arXiv:1403.2805. https://arxiv.org/abs/1403.2805

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. The R Journal, 10(1), 439–446. https://doi.org/10.32614/RJ-2018-009

Pebesma, E., & Bivand, R. (2023). Spatial data science: With applications in R. Chapman and Hall/CRC. https://doi.org/10.1201/9780429459016

R Core Team. (2023). R: A language and environment for statistical computing. R Foundation for Statistical Computing. https://www.R-project.org/

STAC Specification. (2021). SpatioTemporal Asset Catalog specification, v1.0.0. https://github.com/radiantearth/stac-spec

Wickham, H. (2023). testthat: Unit testing for R. R package. https://CRAN.R-project.org/package=testthat

Wickham, H., & Bryan, J. (2023). R packages (2nd ed.). O'Reilly. https://r-pkgs.org/

Wickham, H., Hester, J., & Bryan, J. (2023). readr: Read rectangular text data (R package version 2.1.4). https://CRAN.R-project.org/package=readr

Wildfire Resilience Index. (2026). National Center for Ecological Analysis and Synthesis. https://www.nceas.ucsb.edu/wildfire-resilience-index

Appendix

Table 1. Layer inventory for the Western Wildfire Resilience Index (WRI) as hosted through `firex`. All 82 layers are COGs at 90 m resolution in `EPSG:5070` (NAD83 / Conus Albers), hosted on KNB. Abbreviated layer IDs beginning with "..." continue the `sense_of_place` prefix. Source abbreviations: WRI/NCEAS = processed by NCEAS; CDC = Centers for Disease Control and Prevention; AHA = American Hospital Association; EPA = Environmental Protection Agency; AQS = Air Quality System; BLS = Bureau of Labor Statistics; ACS = American Community Survey; ICC = International Code Council; NLCD = National Land Cover Database; USFS = US Forest Service; NPS = National Park Service; GAP = Gap Analysis Program; IUCN = International Union for Conservation of Nature; USGS = US Geological Survey.

Layer ID	Description	Type	Dimension	Source	License
WRI Score — 1 layer					
`WRI_score`	Final composite score	`final_score`		WRI / NCEAS	MIT
Air Quality — 11 layers					
`air_quality_domain_score`	Domain composite	aggregate	domain_score	WRI / NCEAS	MIT
`air_quality_resilience`	Resilience	aggregate	resilience	WRI / NCEAS	MIT
`air_quality_resistance`	Resistance	aggregate	resistance	WRI / NCEAS	MIT
`air_quality_status`	Status	aggregate	status	WRI / NCEAS	MIT
`air_quality_resistance_asthma`	Asthma rate	indicator	resistance	CDC	Public
`air_quality_resistance_copd`	COPD rate	indicator	resistance	CDC	Public
`air_quality_resistance_hospital_density`	Hospital density	indicator	resistance	AHA	Public
`air_quality_resistance_vulnerable_populations`	Vulnerable populations	indicator	resistance	US Census	Public
`air_quality_resistance_vulnerable_workers`	Vulnerable workers	indicator	resistance	US Census	Public
`air_quality_status_aqi_100`	AQI > 100 days/year	indicator	status	EPA AQS	Public
`air_quality_status_aqi_300`	AQI > 300 days/year	indicator	status	EPA AQS	Public
Communities — 13 layers					
`communities_domain_score`	Domain composite	aggregate	domain_score	WRI / NCEAS	MIT
`communities_resilience`	Resilience	aggregate	resilience	WRI / NCEAS	MIT
`communities_resistance`	Resistance	aggregate	resistance	WRI / NCEAS	MIT
`communities_recovery_income`	Income recovery	indicator	recovery	US Census ACS	Public
`communities_recovery_incorporation`	Incorporation recovery	indicator	recovery	US Census ACS	Public
`communities_recovery_owners`	Home ownership	indicator	recovery	US Census ACS	Public
`communities_resistance_age_65_plus`	Age 65+ share	indicator	resistance	US Census	Public

Layer ID	Description	Type	Dimension	Source	License
communities_resistance_cwpps	CWPPs presence	indicator	resistance	NFPA	Public
communities_resistance_disability	Disability share	indicator	resistance	US Census	Public
communities_resistance_egress	Egress routes	indicator	resistance	US Census	Public
communities_resistance_firewise_communities	Firewise communities	indicator	resistance	NFPA	Public
communities_resistance_no_vehicle	No-vehicle households	indicator	resistance	US Census	Public
communities_resistance_volunteer_fire_stations	Volunteer fire stations	indicator	resistance	NFPA	Public
Infrastructure — 9 layers					
infrastructure_domain_score	Domain composite	aggregate	domain_score	WRI / NCEAS	MIT
infrastructure_resilience	Resilience	aggregate	resilience	WRI / NCEAS	MIT
infrastructure_resistance	Resistance	aggregate	resistance	WRI / NCEAS	MIT
infrastructure_status	Status	aggregate	status	WRI / NCEAS	MIT
infrastructure_resistance_building_codes	Building codes adopted	indicator	resistance	ICC	Public
infrastructure_resistance_d_space	Defensible space	indicator	resistance	USFS	Public
infrastructure_resistance_egress	Egress routes	indicator	resistance	USFS	Public
infrastructure_resistance_fire_resource_density	Fire resource density	indicator	resistance	USFS	Public
infrastructure_resistance_wildland_urban_interface	WUI classification	indicator	resistance	NLCD / USFS	Public
Livelihoods — 9 layers					
livelihoods_domain_score	Domain composite	aggregate	domain_score	WRI / NCEAS	MIT
livelihoods_resilience	Resilience	aggregate	resilience	WRI / NCEAS	MIT
livelihoods_resistance	Resistance	aggregate	resistance	WRI / NCEAS	MIT
livelihoods_status	Status	aggregate	status	WRI / NCEAS	MIT
livelihoods_recovery_diversity_of_jobs	Job diversity	indicator	recovery	BLS	Public
livelihoods_resistance_job_vulnerability	Job vulnerability	indicator	resistance	BLS	Public
livelihoods_status_housing_burden	Housing cost burden	indicator	status	US Census ACS	Public
livelihoods_status_median_income	Median household income	indicator	status	US Census ACS	Public

Layer ID	Description	Type	Dimension	Source	License
livelihoods_status_unemployment	Unemployment rate	indicator	status	BLS	Public
Natural Habitats — 4 layers					
natural_habitats_domain_score	Domain composite	aggregate	domain_score	WRI / NCEAS	MIT
natural_habitats_resilience	Resilience	aggregate	resilience	WRI / NCEAS	MIT
natural_habitats_resistance	Resistance	aggregate	resistance	WRI / NCEAS	MIT
natural_habitats_status	Status	aggregate	status	WRI / NCEAS	MIT
Sense of Place — Iconic Places — 11 layers					
sense_of_place_iconic_places_domain_score	Domain composite	aggregate	domain_score	NPS / USFS	Public
sense_of_place_iconic_places_resilience	Resilience	aggregate	resilience	NPS / USFS	Public
sense_of_place_iconic_places_resistance	Resistance	aggregate	resistance	NPS / USFS	Public
...recovery_degree_of_protection	Degree of protection	indicator	recovery	GAP / PAD-US	Public
...recovery_national_parks	National parks proximity	indicator	recovery	NPS	Public
...resistance_egress	Egress routes	indicator	resistance	USFS	Public
...resistance_fire_resource_density	Fire resource density	indicator	resistance	USFS	Public
...resistance_national_parks	National parks resistance	indicator	resistance	NPS	Public
...resistance_structures	Historic structures	indicator	resistance	NPS	Public
...resistance_wui	WUI classification	indicator	resistance	NLCD / USFS	Public
...status_presence	Presence/absence	indicator	status	NPS	Public
Sense of Place — Iconic Species — 8 layers					
sense_of_place_iconic_species_domain_score	Domain composite	aggregate	domain_score	IUCN / NatureServe	Public
...iconic_species_resilience	Resilience	aggregate	resilience	IUCN / NatureServe	Public
...iconic_species_resistance	Resistance	aggregate	resistance	IUCN / NatureServe	Public
...iconic_species_status	Status	aggregate	status	IUCN / NatureServe	Public
...iconic_species_area_recovery	Species range area recovery	indicator	recovery	IUCN	Public
...iconic_species_status_75_extinction_rescaled	75% extinction risk score	indicator	status	IUCN	Public
...iconic_species_traits_recovery	Trait-based recovery	indicator	recovery	NatureServe	Public

Layer ID	Description	Type	Dimension	Source	License
...iconic_species_traits_resistance	Trait-based resistance	indicator	resistance	NatureServe	Public
Species — 7 layers					
species_domain_score	Domain composite	aggregate	domain_score	NatureServe / IUCN	Public
species_resilience	Resilience	aggregate	resilience	NatureServe / IUCN	Public
species_resistance	Resistance	aggregate	resistance	NatureServe / IUCN	Public
species_status	Status	aggregate	status	NatureServe / IUCN	Public
species_recovery_range_area	Species range area	indicator	recovery	NatureServe	Public
species_recovery_traits	Trait-based recovery	indicator	recovery	NatureServe	Public
species_resistance_traits	Trait-based resistance	indicator	resistance	NatureServe	Public
Water — 8 layers					
water_domain_score	Domain composite	aggregate	domain_score	WRI / NCEAS	MIT
water_resilience	Resilience	aggregate	resilience	WRI / NCEAS	MIT
water_resistance	Resistance	aggregate	resistance	WRI / NCEAS	MIT
water_status	Status	aggregate	status	WRI / NCEAS	MIT
water_resistance_drought_plans	Drought contingency plans	indicator	resistance	EPA	Public
water_resistance_water_treatment	Water treatment capacity	indicator	resistance	EPA	Public
water_status_surface_water_quantity	Surface water quantity	indicator	status	USGS	Public
water_status_surface_water_timing	Surface water timing	indicator	status	USGS	Public