IHDEA 2020 Meeting

*The community-developed,
free and open-source solar data analysis
environment for Python.*

Laura A. Hayes[1]
on behalf of **The SunPy Community**

21 Oct 2020 PyHC

sunpy.org
https://github.com/sunpy/sunpy

[1]NASA Goddard Space Flight Center/USRA

SunPy

What is SunPy?

Python in Heliophysics (PyHC)

The SunPy project facilitates and promotes the use and development of several community-led, free, and open source data analysis software packages for solar physics based on the scientific Python environment.

Functionality

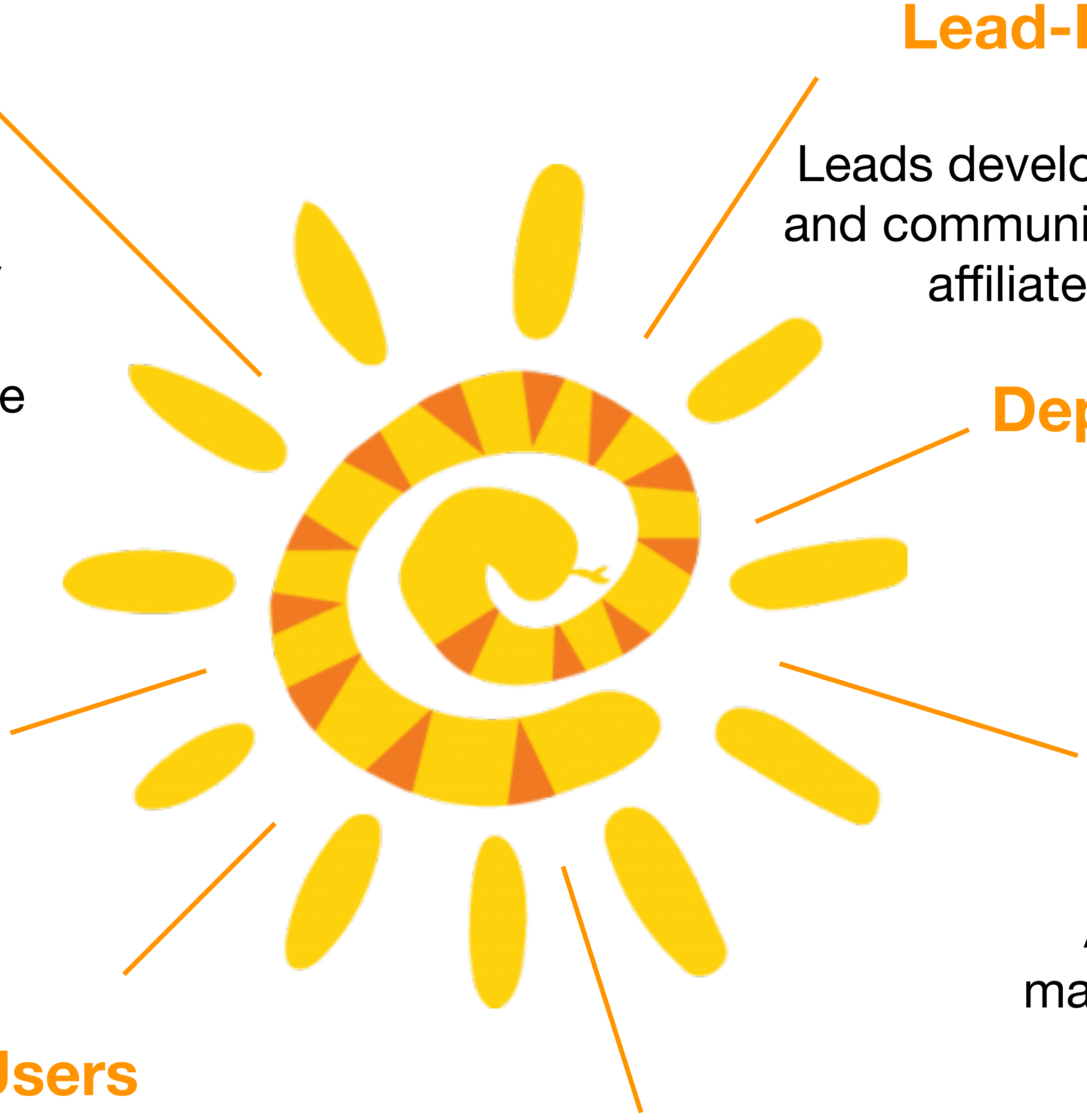
- Provide Python tools specific to solar data analysis - gateway into ecosystem
- Focus on calibrated high level data
- Leverage mature and maintained code from other field (e.g. `astropy`)
- Support other solar packages (affiliated) outside the scope of SunPy core

Cultural

- Open-source community and inclusive of everyone (anyone can contribute!)
- Coordinate development
- Code testing and code review
- Version control
- Standardized and discoverable documentation

SunPy Organization

The SunPy Community



SunPy Board

Up-to 10 people, guides overall direction of sunpy project by voting on SunPy Enhancement Proposals (SEPs) - anyone can propose

Lead-Developer

Leads development core team and community developers and affiliated packages

Deputy Lead-Developer

Assists lead developer

Community Roles

Sub-package maintainers

Assist lead devs with maintaining sub-packages

Summer of Code students

Users

Community Developers

https://sunpy.org/project/

Contributing to SunPy

What does this look like?

Anyone can contribute! 🌞

Doesn't have to just be code! - raising bugs/issues, providing feedback and suggestions, requesting features etc.

Lots of assistance given to newcomers!

Guidelines in place to ensure new code meets quality standard:

Style Guide

All code and docs must comply with widely used style guides (e.g. Pep8)

Docs

New features require documentation - code comments, docstrings and guide, examples

Unit tests

All code must provide unit tests that cover functionality

Within Scope

All code must be within scope and approved by at least 2 members of dev community

Automated testing on GitHub (incl. different op systems, building docs, plots, code-coverage etc).
Use of Azure, CircleCI, Codecov and Travis CI

Where is SunPy now?

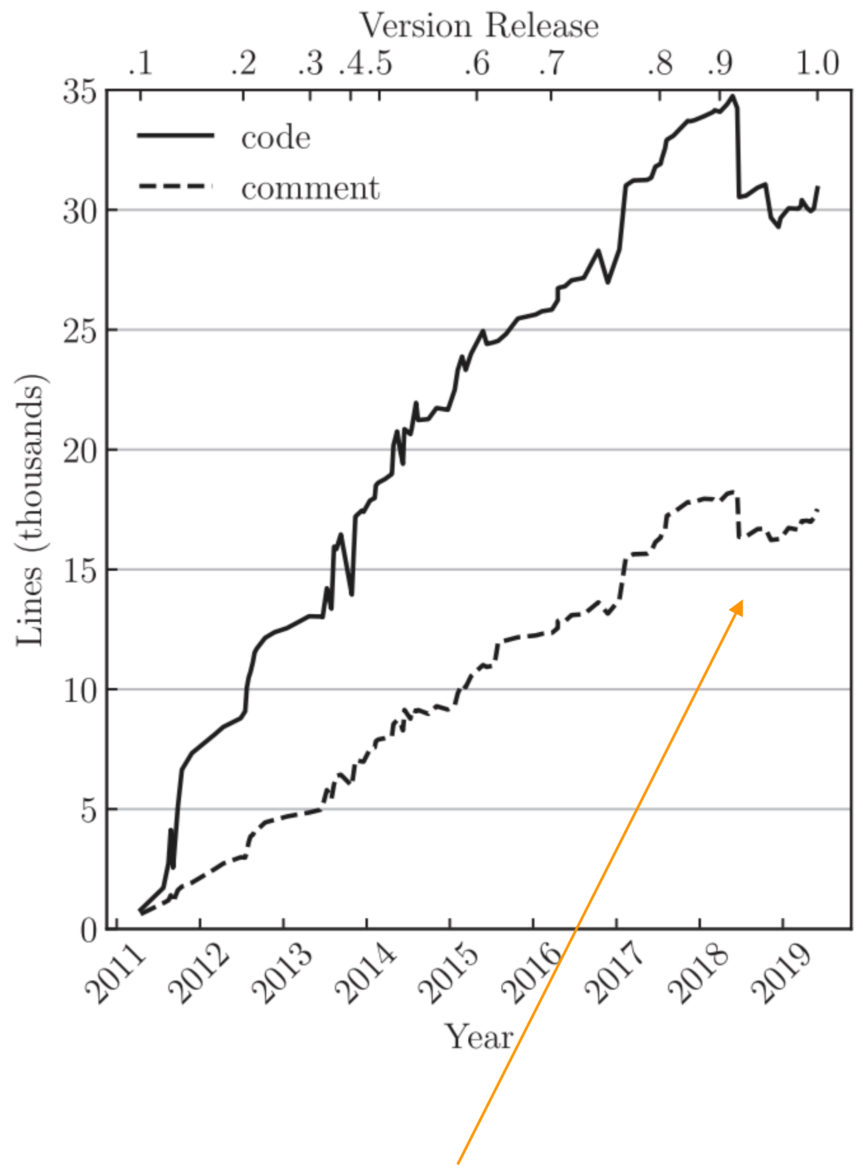
Current Status

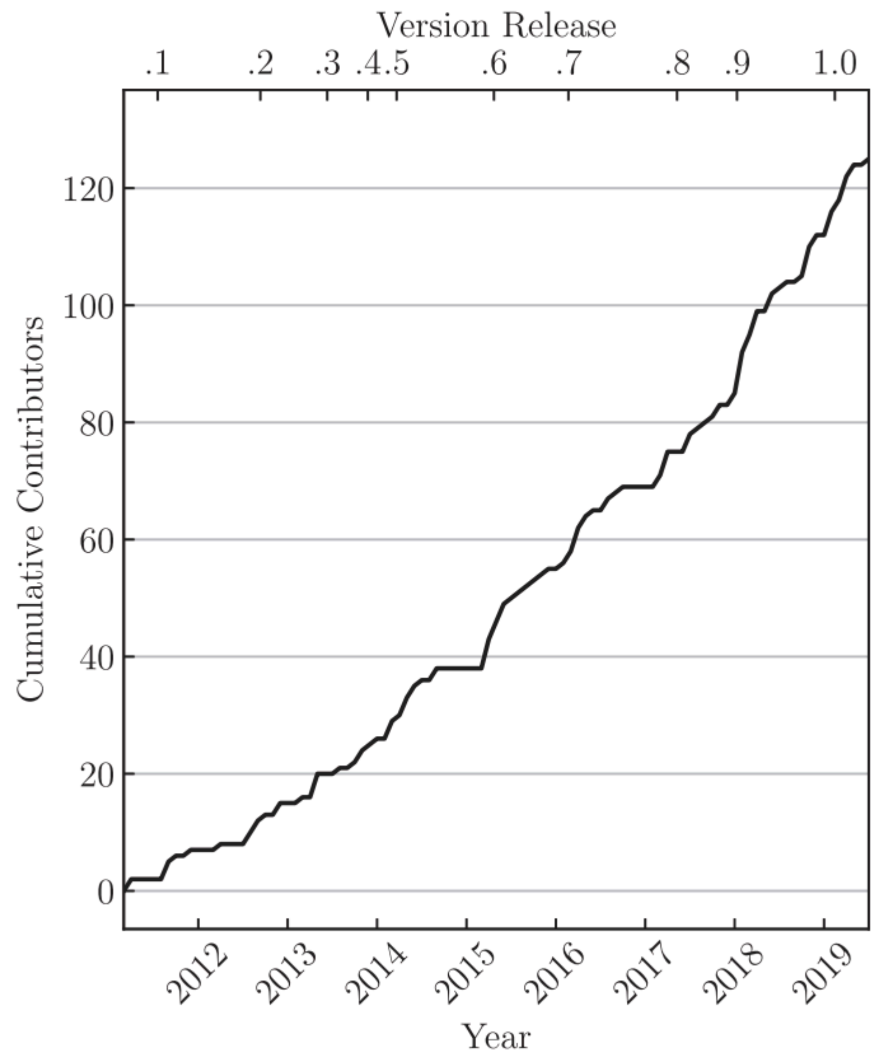
- Project officially founded in March 2014, began in March 2011 (9 years ago!)
- **Released SunPy 2.0** (1.0 first stable release 🎉 🥳) …currently on 2.0.3
- Now release schedule - twice a year
- ~50,000 lines of code (incl. comments & docs), 134 unique contributors
- Published paper and code:
  - *The SunPy Community, et al. "The sunpy project: Open source development and status of the version 1.0 core package." The Astrophysical Journal 890.1 (2020): 68.*
  - *Mumford, Stuart, et al. "SunPy: A Python package for Solar Physics." Journal of Open Source Software 5.46 (2020).*

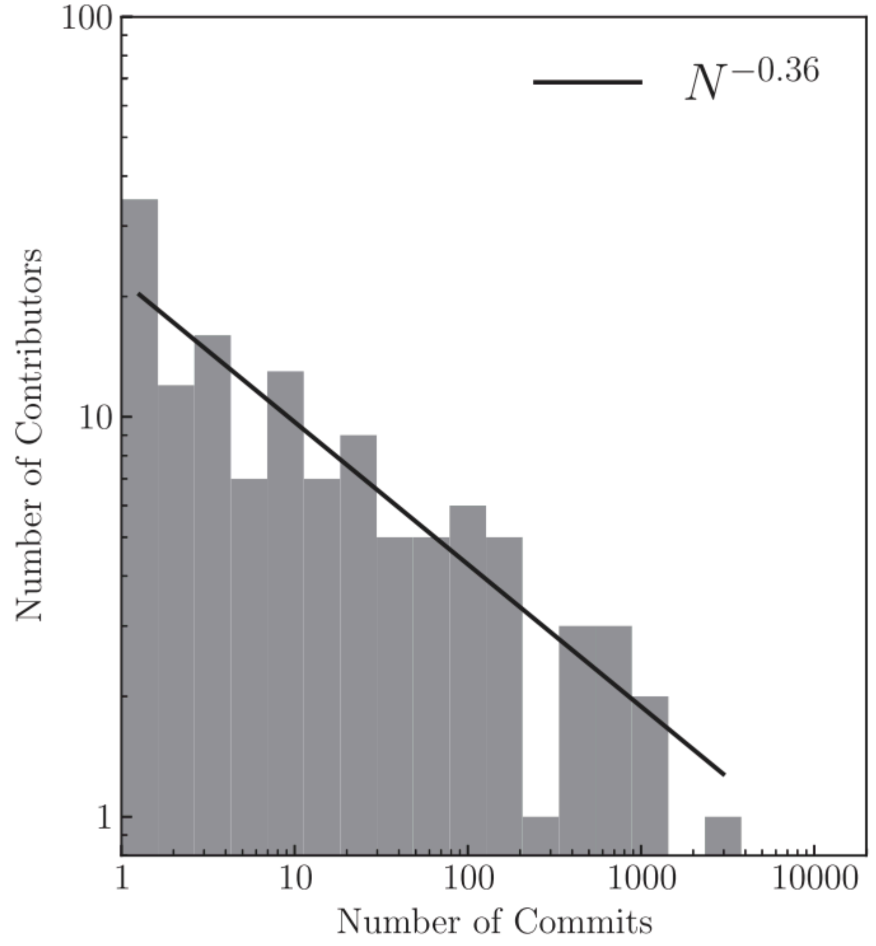
The Journal of Open Source Software

Where is SunPy now?

Current Status







Major tidy up for 1.0

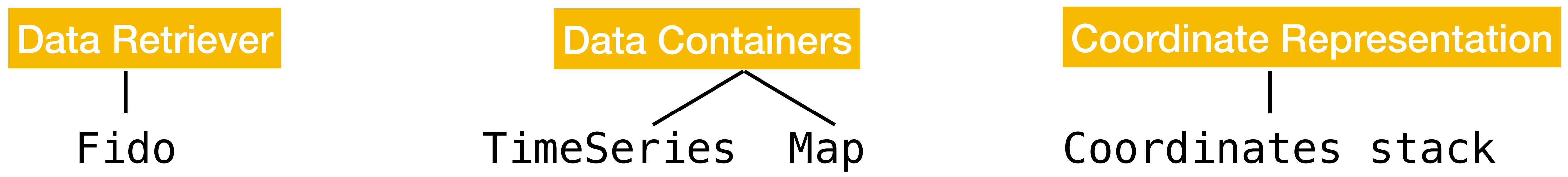
Slope quite steep
hope to flatten this

SunPy 2.0 +

Overview and Highlights

- **Headline core changes (from 1.0+)**:
  - Major clean up of core
  - Improved download capabilities (`parfive`) - parallel downloads
  - Improved coordinates functionality - Sun-specific transformation stack and coordinate frames
  - Full adoption of `astropy` time
  - Logging system

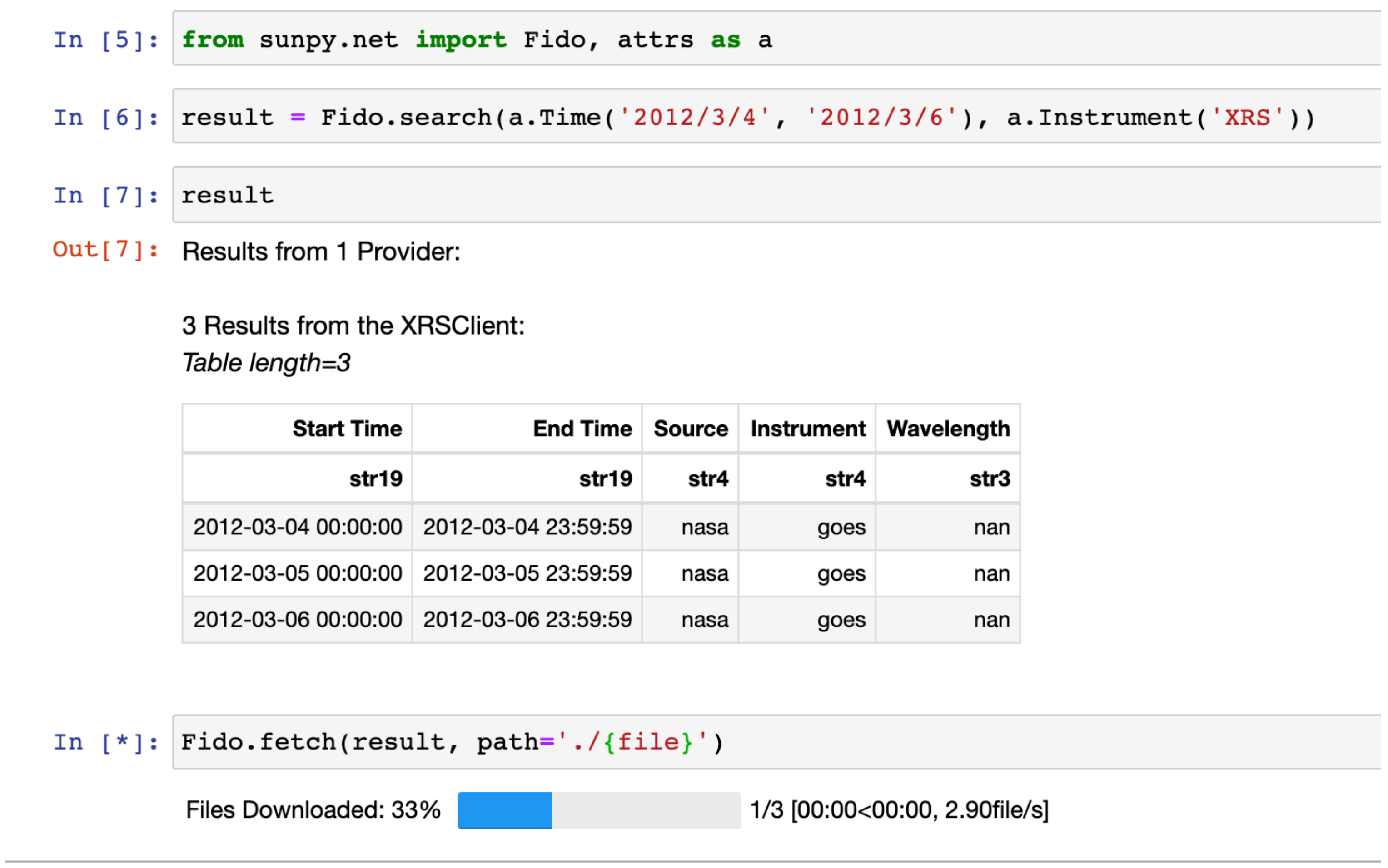
sunpy core overview:



Data Retriever

sunpy.net and Fido

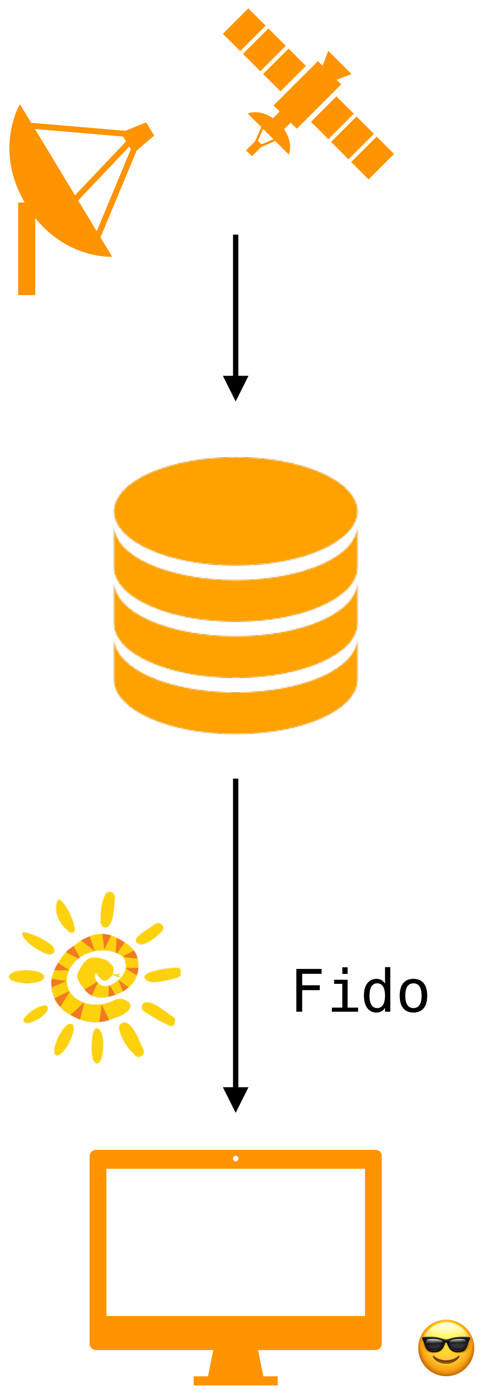
- Fido Unified API for searching and downloading solar data from various search engines and data sources (e.g. VSO, JSOC, https, ftp)



```
In [5]: from sunpy.net import Fido, attrs as a

In [6]: result = Fido.search(a.Time('2012/3/4', '2012/3/6'), a.Instrument('XRS'))

In [7]: result
Out[7]: Results from 1 Provider:

        3 Results from the XRSClient:
        Table length=3
```

Start Time	End Time	Source	Instrument	Wavelength
str19	str19	str4	str4	str3
2012-03-04 00:00:00	2012-03-04 23:59:59	nasa	goes	nan
2012-03-05 00:00:00	2012-03-05 23:59:59	nasa	goes	nan
2012-03-06 00:00:00	2012-03-06 23:59:59	nasa	goes	nan

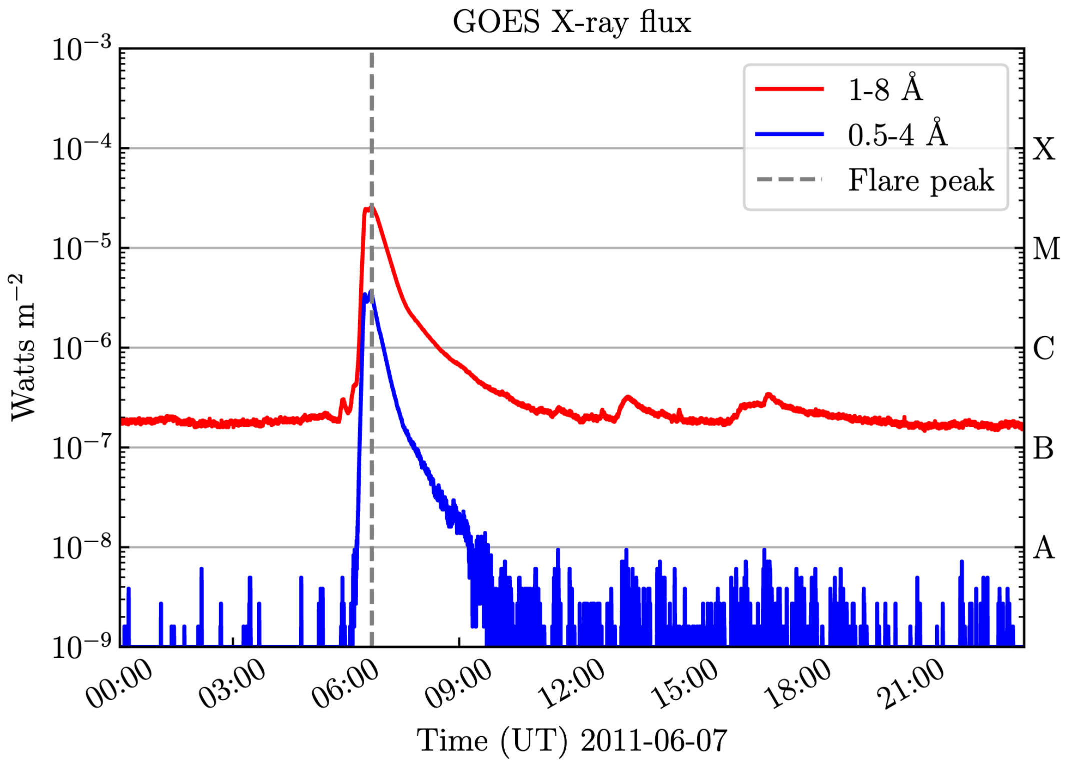
```
In [*]: Fido.fetch(result, path='./{file}')

        Files Downloaded: 33%  1/3 [00:00<00:00, 2.90file/s]
```



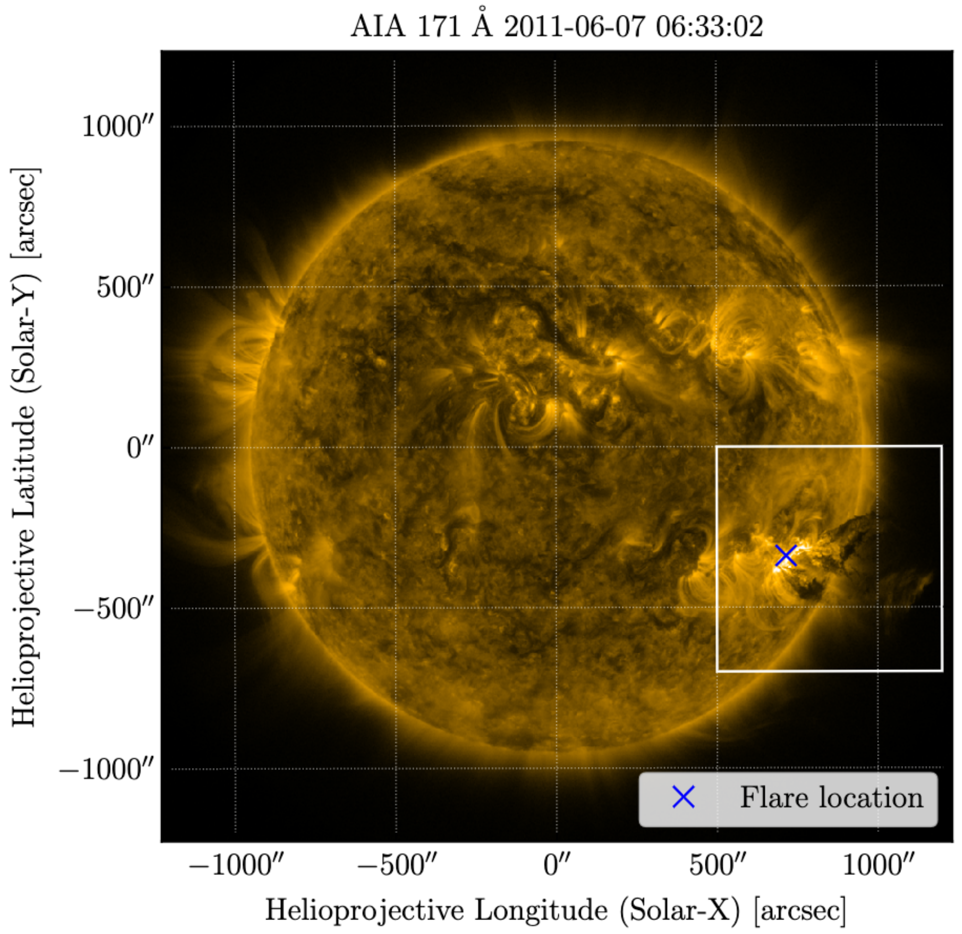
Data Containers

`TimeSeries` and `Map`

- SunPy provides general, **standard and consistent interface** for loading and representing solar data across different instruments and missions.



`TimeSeries`: 1D temporal data

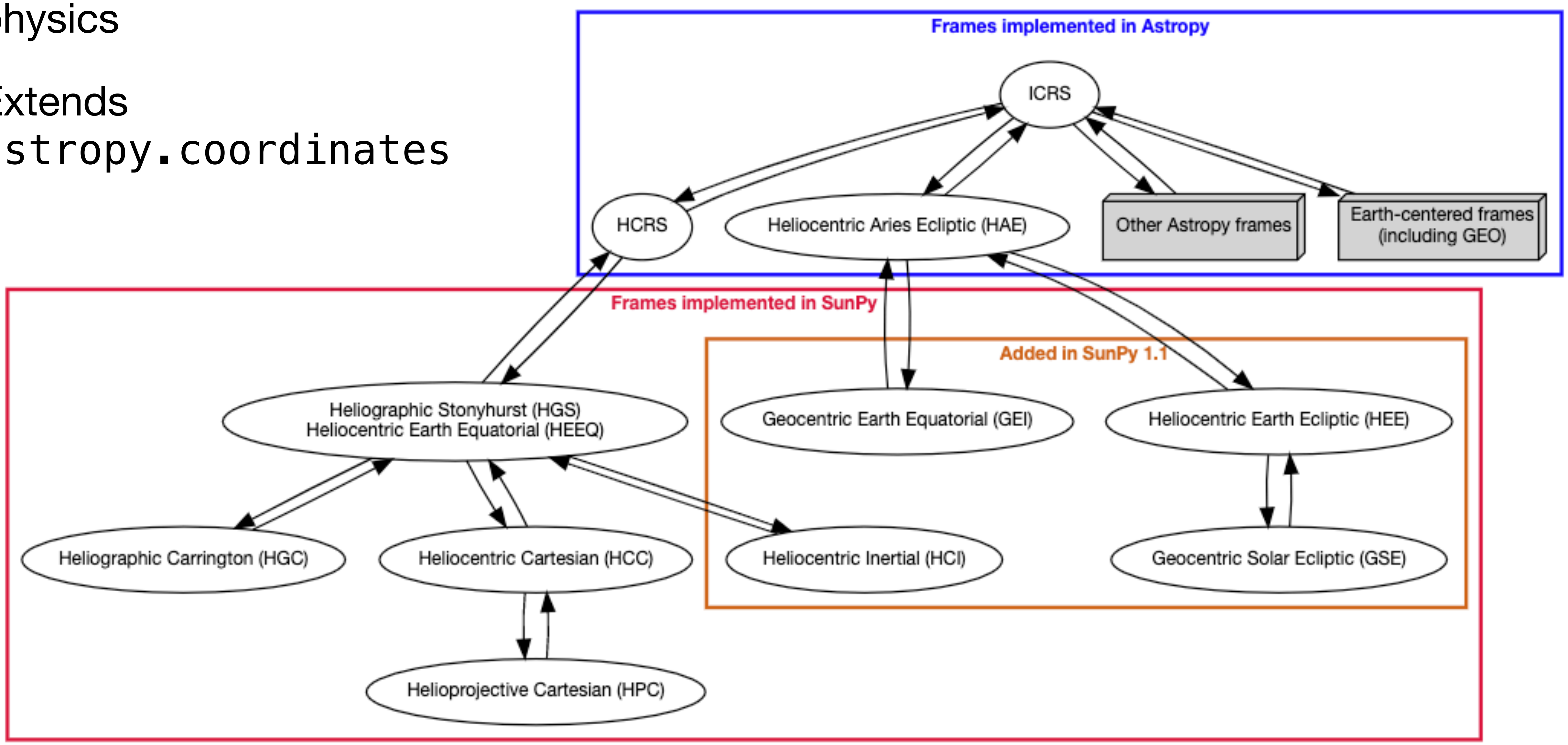


`Map`: 2D coordinate aware image data

Coordinates

Solar Coordinates

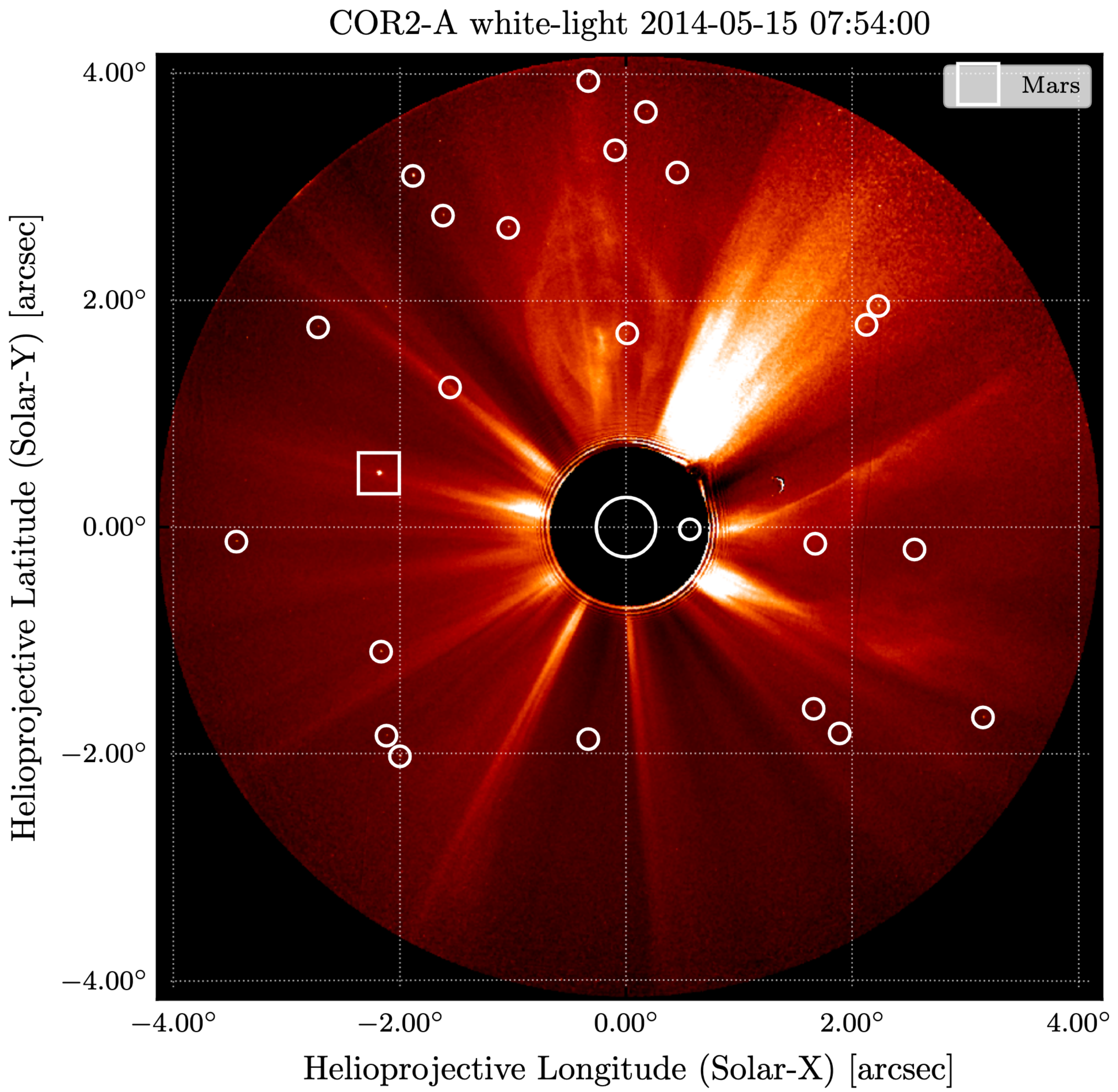
- `sunpy.coordinates` for representing and **transforming** coordinates used in solar physics
- Extends `astropy.coordinates`

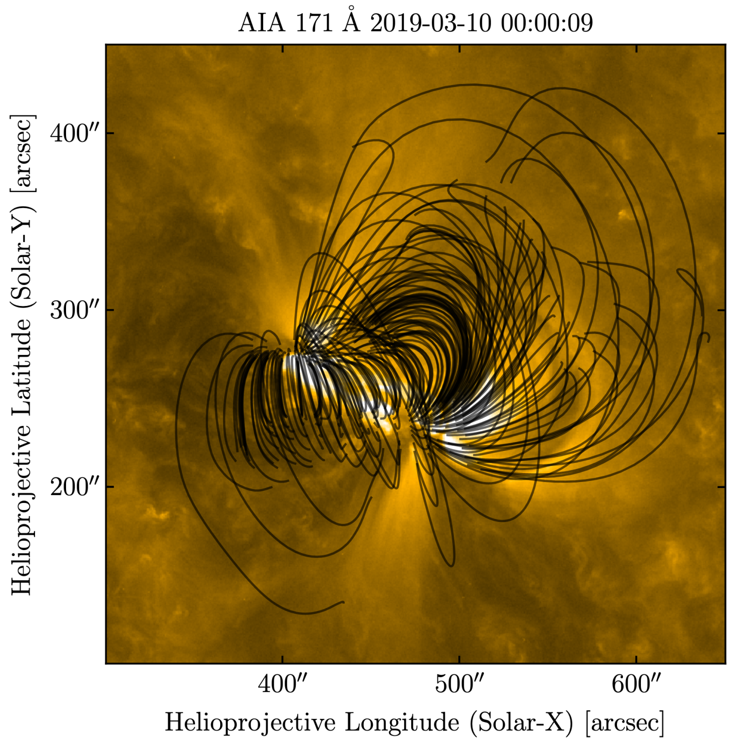


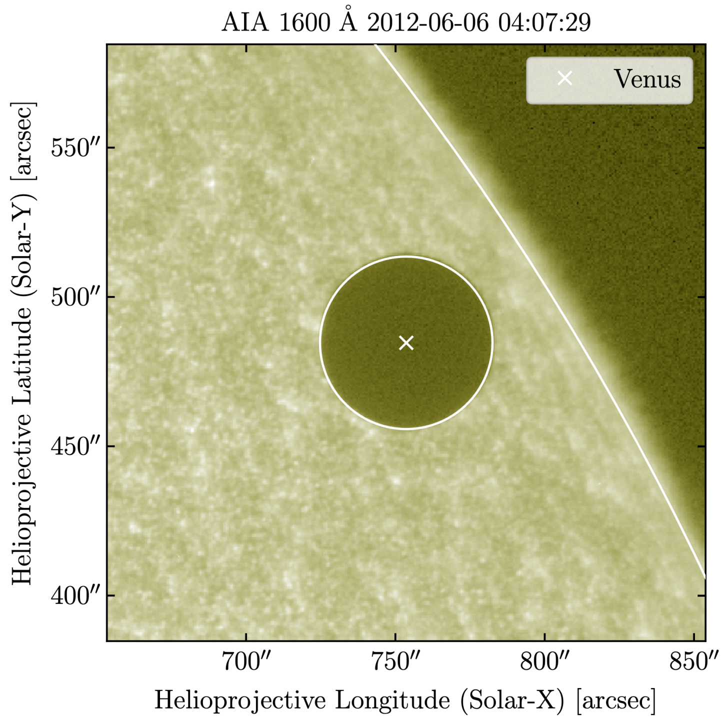
Frames implemented in Astropy
ICRS
HCRS
Heliocentric Aries Ecliptic (HAE)
Other Astropy frames
Earth-centered frames (including GEO)
Frames implemented in SunPy
Added in SunPy 1.1
Heliographic Stonyhurst (HGS) Heliocentric Earth Equatorial (HEEQ)
Geocentric Earth Equatorial (GEI)
Heliocentric Earth Ecliptic (HEE)
Heliographic Carrington (HGC)
Heliocentric Cartesian (HCC)
Heliocentric Inertial (HCI)
Geocentric Solar Ecliptic (GSE)
Helioprojective Cartesian (HPC)

Coordinates

Some Examples

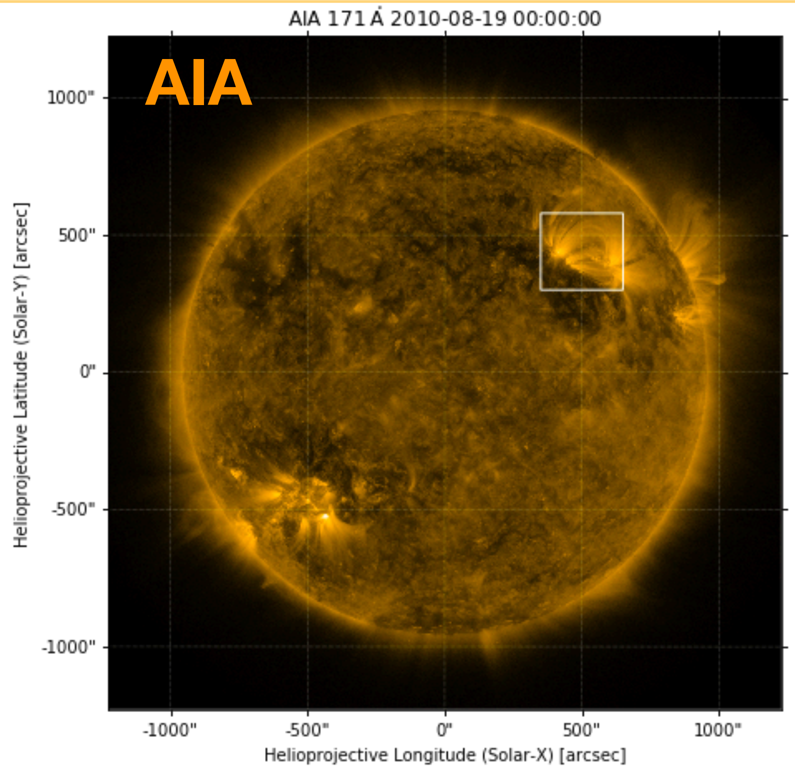


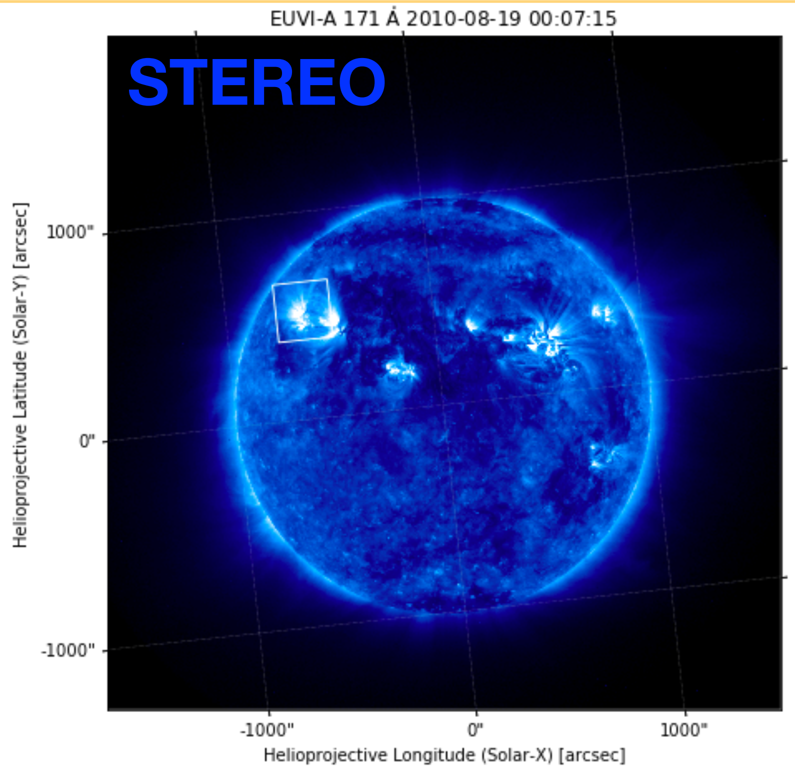


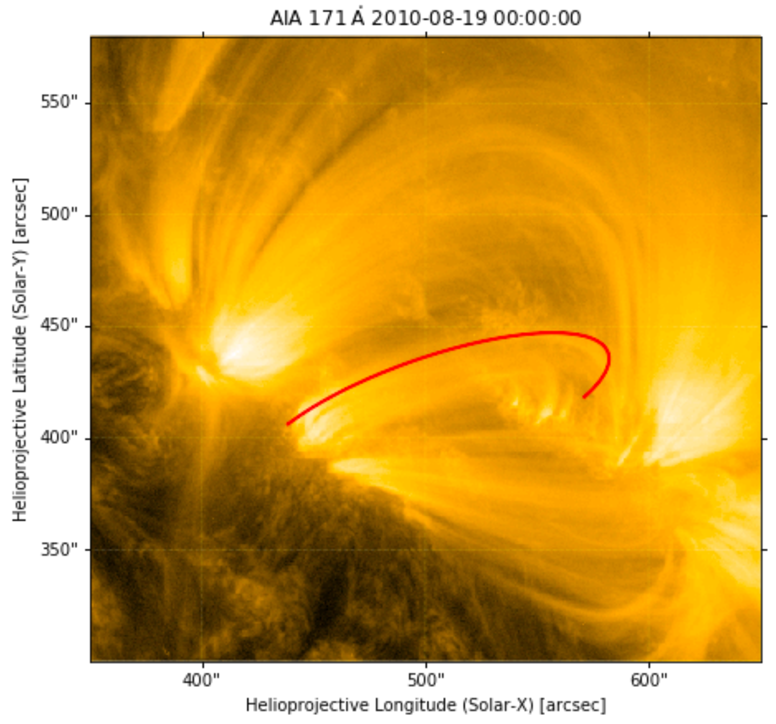


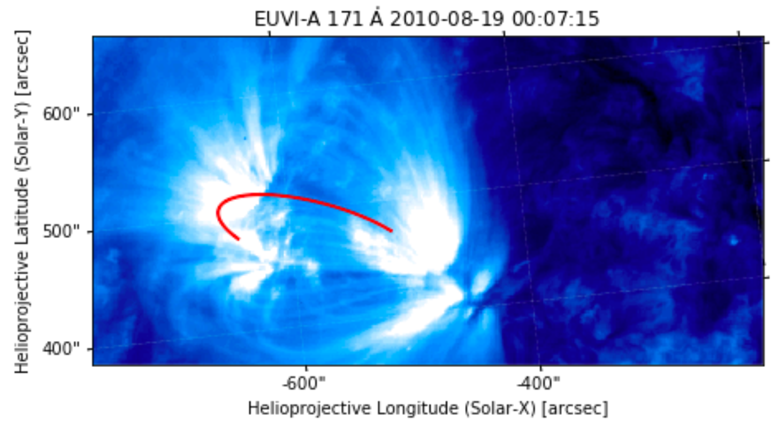
Coordinates

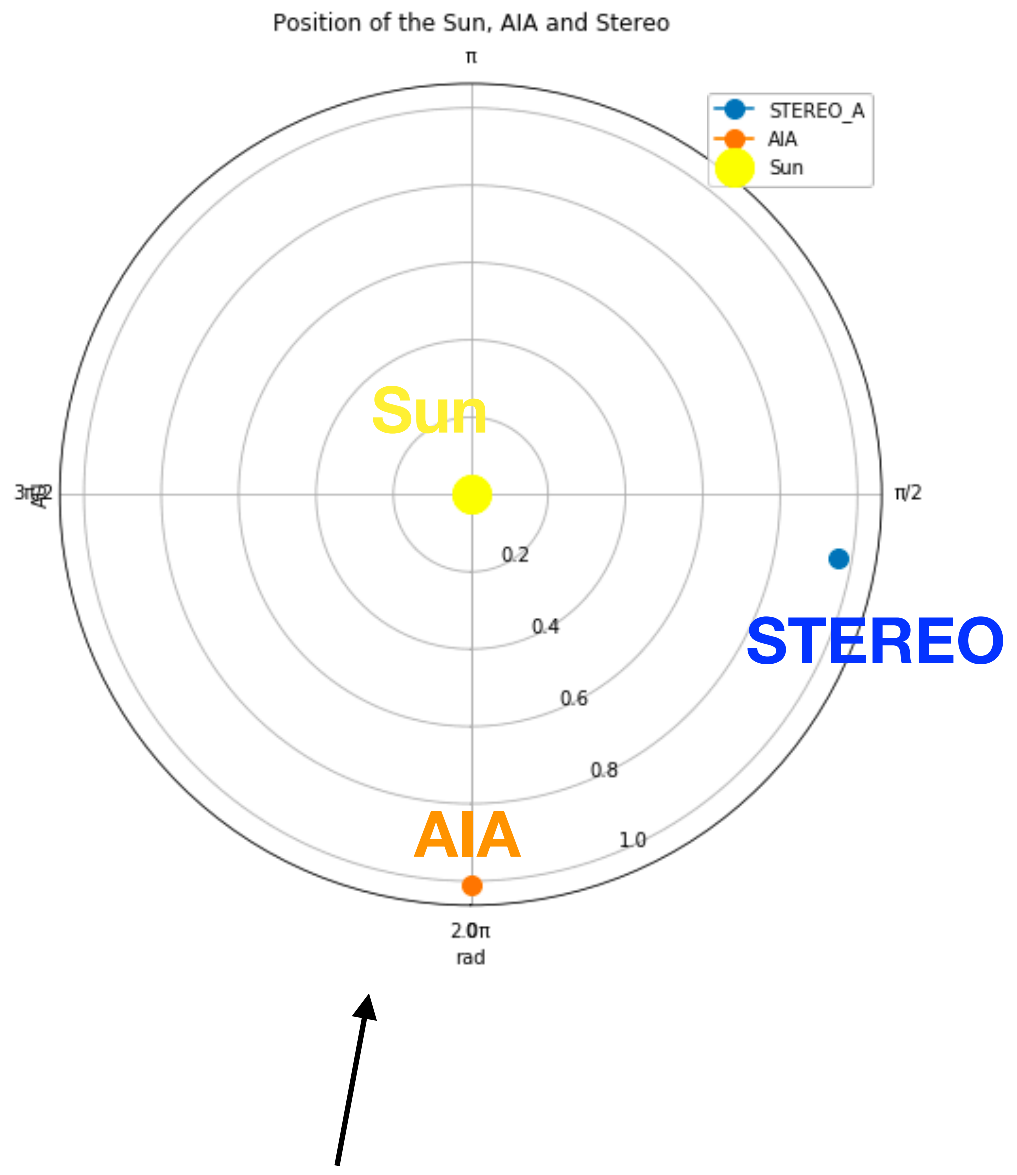
Finding regions of interest - two different fields of view


AIA 171 Å 2010-08-19 00:00:00
AIA
Helioprojective Latitude (Solar-Y) [arcsec]
Helioprojective Longitude (Solar-X) [arcsec]
1000"
500"
0"
-500"
-1000"


EUVI-A 171 Å 2010-08-19 00:07:15
STEREO
Helioprojective Latitude (Solar-Y) [arcsec]
Helioprojective Longitude (Solar-X) [arcsec]
1000"
0"
-1000"


AIA 171 Å 2010-08-19 00:00:00
Helioprojective Latitude (Solar-Y) [arcsec]
Helioprojective Longitude (Solar-X) [arcsec]
550"
500"
450"
400"
350"
400"
500"
600"


EUVI-A 171 Å 2010-08-19 00:07:15
Helioprojective Latitude (Solar-Y) [arcsec]
Helioprojective Longitude (Solar-X) [arcsec]
600"
500"
400"
-600"
-400"

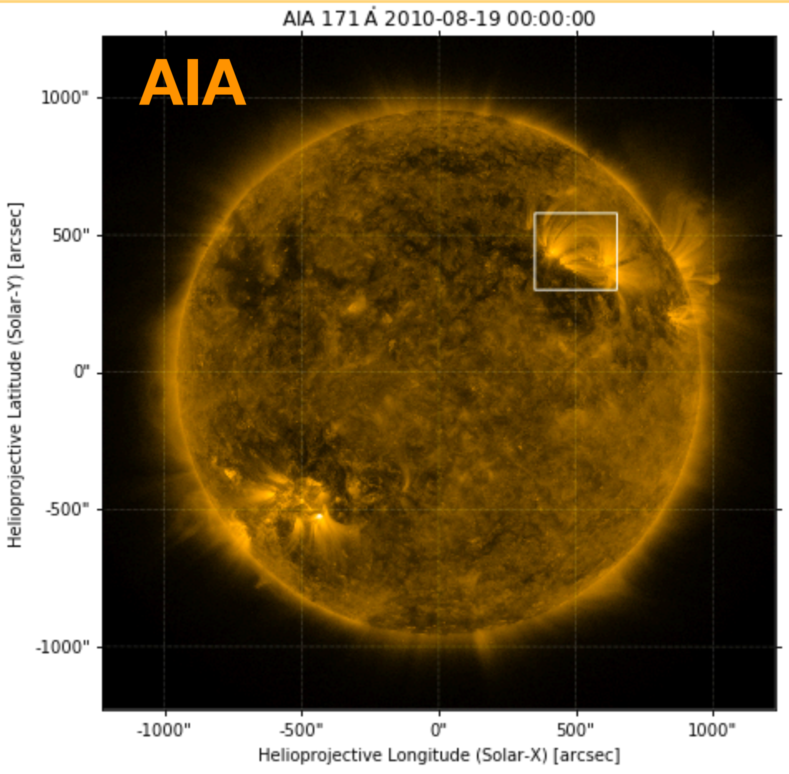


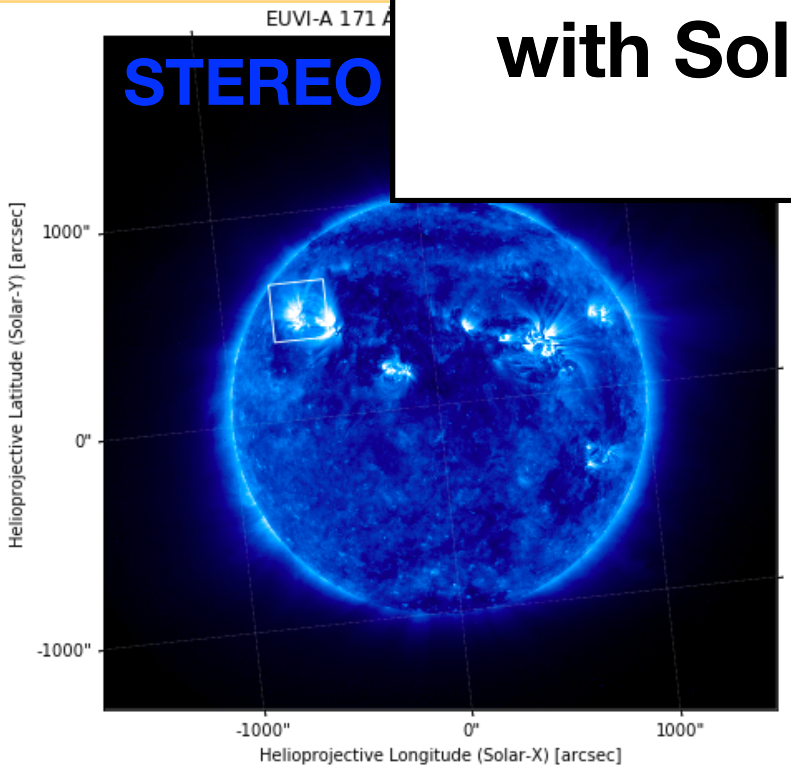
Positions of satellites

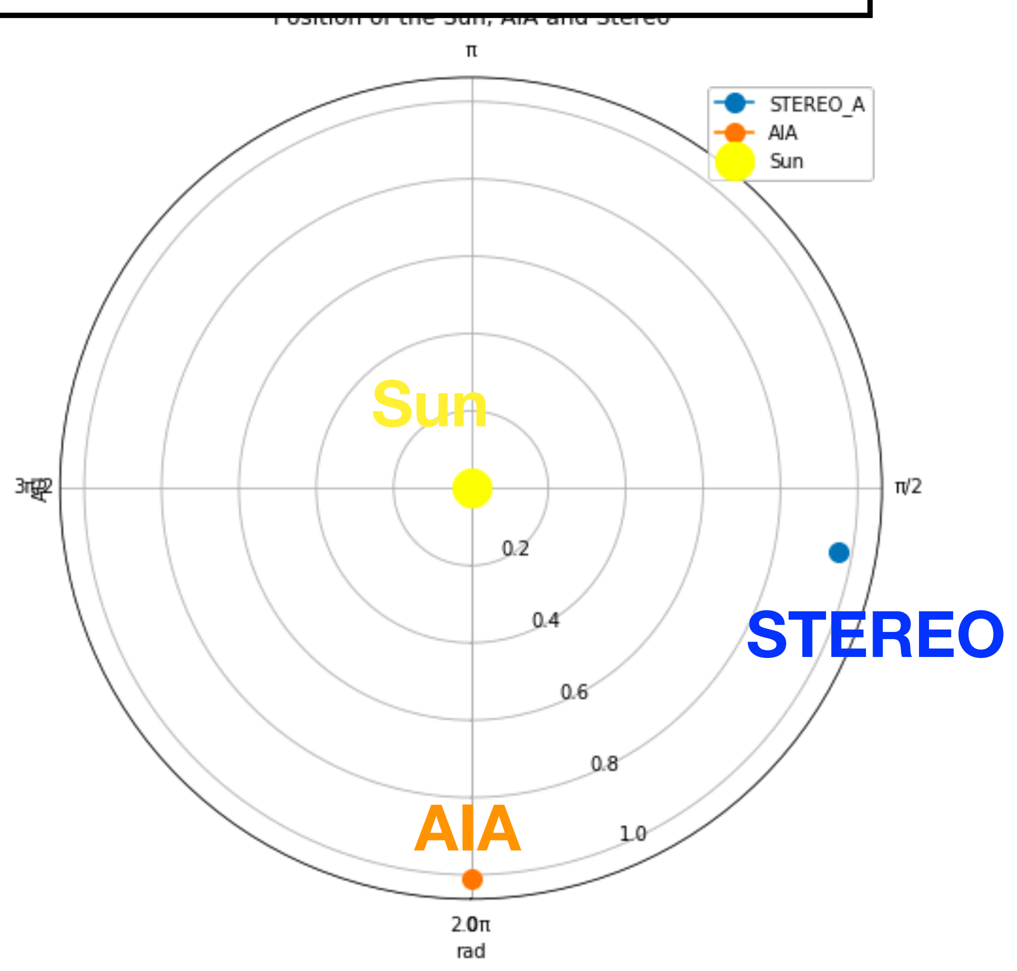
Coordinates

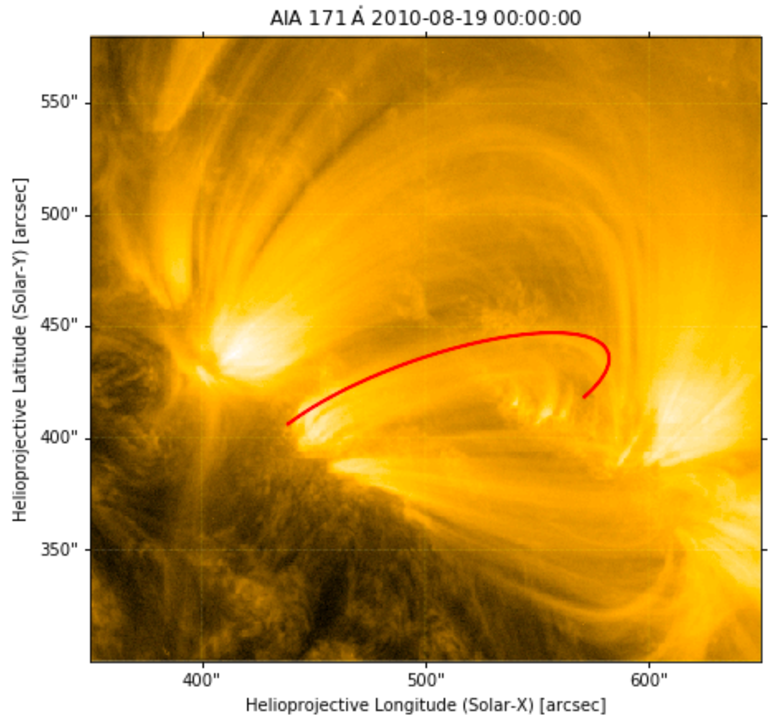
Finding regions of interest

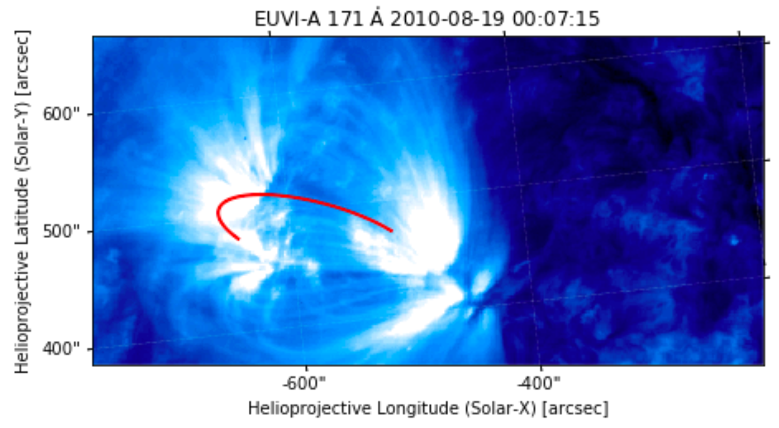
Really important to have this functionality for new observations with Solar Orbiter and Parker Solar Probe!


AIA 171 Å 2010-08-19 00:00:00
AIA
Helioprojective Longitude (Solar-X) [arcsec]
Helioprojective Latitude (Solar-Y) [arcsec]


EUVI-A 171 Å
STEREO
Helioprojective Longitude (Solar-X) [arcsec]
Helioprojective Latitude (Solar-Y) [arcsec]




AIA 171 Å 2010-08-19 00:00:00
Helioprojective Longitude (Solar-X) [arcsec]
Helioprojective Latitude (Solar-Y) [arcsec]

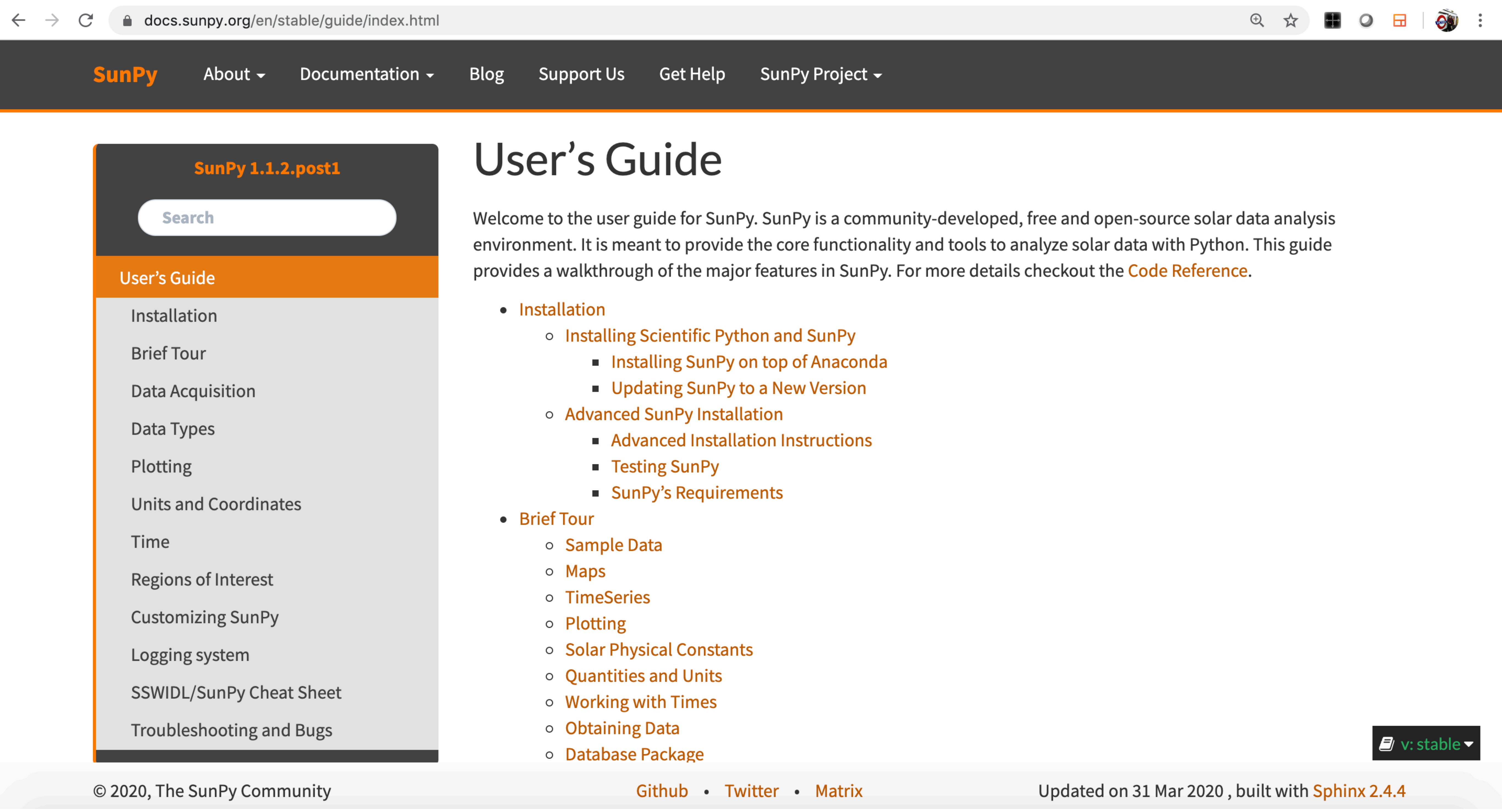

EUVI-A 171 Å 2010-08-19 00:07:15
Helioprojective Longitude (Solar-X) [arcsec]
Helioprojective Latitude (Solar-Y) [arcsec]

Positions of satellites

Documentation

User Docs

https://docs.sunpy.org/en/stable/


docs.sunpy.org/en/stable/guide/index.html

SunPy | About | Documentation | Blog | Support Us | Get Help | SunPy Project

SunPy 1.1.2.post1

Search

User's Guide
- Installation
- Brief Tour
- Data Acquisition
- Data Types
- Plotting
- Units and Coordinates
- Time
- Regions of Interest
- Customizing SunPy
- Logging system
- SSWIDL/SunPy Cheat Sheet
- Troubleshooting and Bugs

User's Guide

Welcome to the user guide for SunPy. SunPy is a community-developed, free and open-source solar data analysis environment. It is meant to provide the core functionality and tools to analyze solar data with Python. This guide provides a walkthrough of the major features in SunPy. For more details checkout the Code Reference.

- Installation
  - Installing Scientific Python and SunPy
    - Installing SunPy on top of Anaconda
    - Updating SunPy to a New Version
  - Advanced SunPy Installation
    - Advanced Installation Instructions
    - Testing SunPy
    - SunPy's Requirements
- Brief Tour
  - Sample Data
  - Maps
  - TimeSeries
  - Plotting
  - Solar Physical Constants
  - Quantities and Units
  - Working with Times
  - Obtaining Data
  - Database Package

v: stable

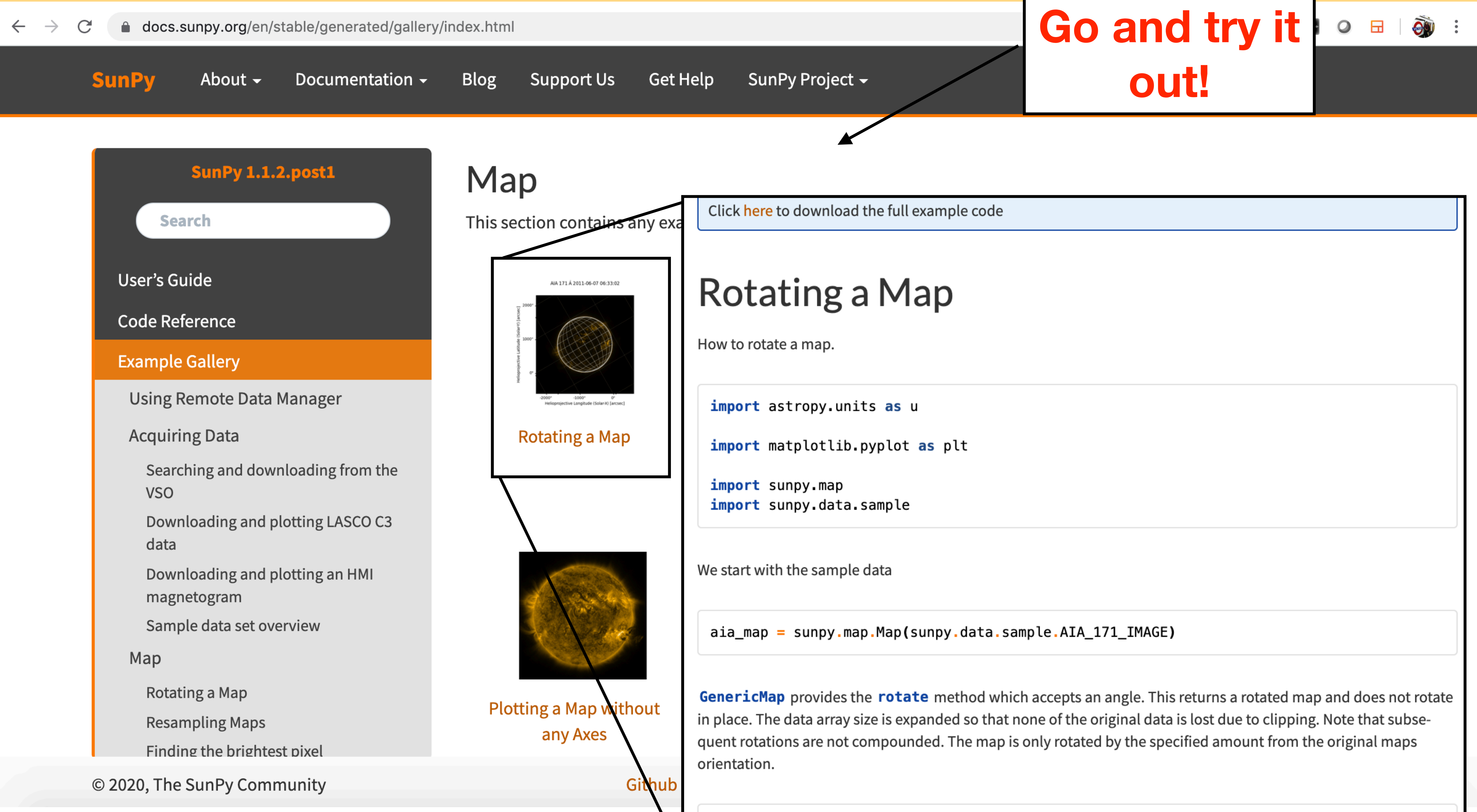
© 2020, The SunPy Community   Github • Twitter • Matrix   Updated on 31 Mar 2020 , built with Sphinx 2.4.4

Documentation

Example Gallery

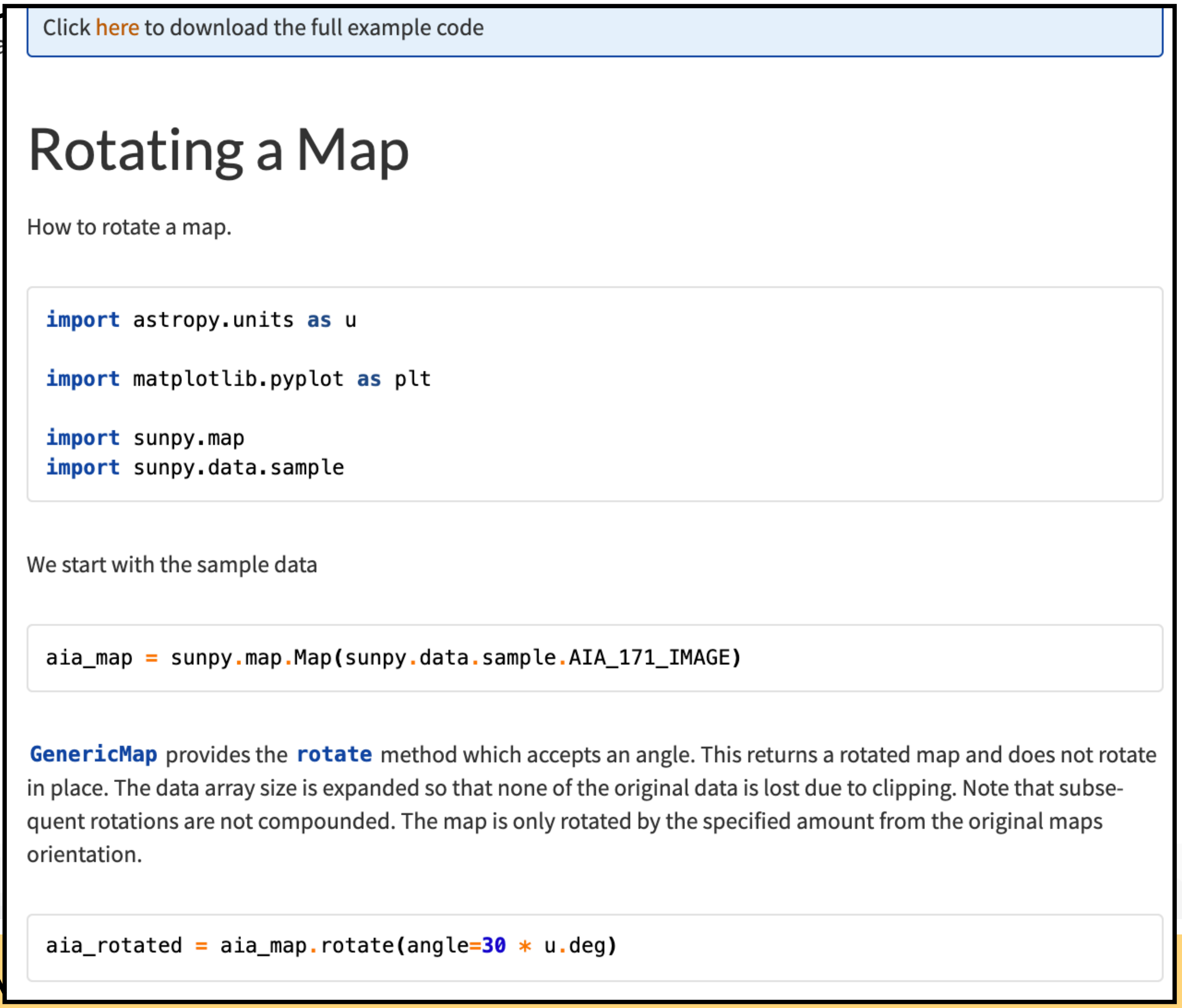
https://docs.sunpy.org/en/stable/

Go and try it out!


docs.sunpy.org/en/stable/generated/gallery/index.html
SunPy
About
Documentation
Blog
Support Us
Get Help
SunPy Project
SunPy 1.1.2.post1
Search
User's Guide
Code Reference
Example Gallery
Using Remote Data Manager
Acquiring Data
Searching and downloading from the VSO
Downloading and plotting LASCO C3 data
Downloading and plotting an HMI magnetogram
Sample data set overview
Map
Rotating a Map
Resampling Maps
Finding the brightest pixel
Rotating a Map
Plotting a Map without any Axes
© 2020, The SunPy Community
Github

Map

This section contains any exa


Click here to download the full example code

Rotating a Map

How to rotate a map.

```
import astropy.units as u

import matplotlib.pyplot as plt

import sunpy.map
import sunpy.data.sample
```

We start with the sample data

```
aia_map = sunpy.map.Map(sunpy.data.sample.AIA_171_IMAGE)
```

`GenericMap` provides the `rotate` method which accepts an angle. This returns a rotated map and does not rotate in place. The data array size is expanded so that none of the original data is lost due to clipping. Note that subsequent rotations are not compounded. The map is only rotated by the specified amount from the original maps orientation.

```
aia_rotated = aia_map.rotate(angle=30 * u.deg)
```

Looking ahead

Roadmap and future plans

- Two new releases now planned per year 🌞
- Future development and roadmap plan:
  - `NDCube` for `Map` – upgrade to N-dim coordinate aware data
  - Improved support for data with spectral axes and multi-dimensional data sets
  - standardized approach to metadata
  - Package template for **affiliated** packages - e.g. incubator for instrument teams
- Hope to **grow community involvement** - feedback from users, users —> contributors