

Observation Planning for Radio Observations

This worksheet will introduce you to some concepts you'll need to plan out your observations with the radio telescope.

First, we are going to select our observatory. We will use astroplan plus astropy coordinates.

Import them and confirm they're installed:

```
In [1]: %pip install astroplan
```

```
Requirement already satisfied: astroplan in c:\programdata\anaconda3\lib\site-p
ackages (0.8)
Requirement already satisfied: numpy>=1.17 in c:\programdata\anaconda3\lib\site
-packages (from astroplan) (1.19.2)
Requirement already satisfied: pytz in c:\programdata\anaconda3\lib\site-packag
es (from astroplan) (2020.1)
Requirement already satisfied: six in c:\users\sydnee o'donnell\appdata\roaming
\python\python38\site-packages (from astroplan) (1.15.0)
Requirement already satisfied: astropy>=4 in c:\users\sydnee o'donnell\appdata
\roaming\python\python38\site-packages (from astroplan) (4.3.1)
Requirement already satisfied: pyerfa>=1.7.3 in c:\programdata\anaconda3\lib\si
te-packages (from astropy>=4->astroplan) (2.0.0)
Note: you may need to restart the kernel to use updated packages.
```

```
In [2]: import numpy as np
from astropy import units as u
from astropy import coordinates
from astroplan import Observer
```

Next, we set up the observer.

We are going to pick the observer based on the known latitude and longitude of our observatory in Gainesville in [geodetic \(https://en.wikipedia.org/wiki/Geographic_coordinate_system\)](https://en.wikipedia.org/wiki/Geographic_coordinate_system) coordinates, meaning we're specifying the location with two angles and one elevation (height above ellipsoid).

This will convert into some coordinate system used internally by astropy.

```
In [3]: gainesville_location = coordinates.EarthLocation.from_geodetic(lon=-82.3*u.deg, 1
gainesville_location
```

```
Out[3]: (743674.14, -5500332.5, 3131946.8) m
```

We then set up the "observer", which will be used as our reference when looking at the sky. We specify a time zone so we can use local times.

```
In [4]: observer = Observer(gainesville_location, timezone='US/Eastern')
observer
```

```
Out[4]: <Observer: location (lon, lat, el)=(-82.30000000000001 deg, 29.600000000000005
deg, 100.00000000232231 m),
        timezone=<DstTzInfo 'US/Eastern' LMT-1 day, 19:04:00 STD>>
```

Where are we pointed?

Now we can ask some questions: if we point at a specific altitude and azimuth, what's there now?
What's going to be there in 12 hours?

We define "now", but it requires more packages, `pytz` = "python timezone":

```
In [5]: import pytz
import datetime
now = pytz.timezone('US/Eastern').localize(datetime.datetime.now())
now
```

```
Out[5]: datetime.datetime(2022, 10, 26, 18, 21, 5, 789989, tzinfo=<DstTzInfo 'US/Easter
n' EDT-1 day, 20:00:00 DST>)
```

Quick review

Before running the next cell, answer:

What is at 90 degrees altitude?

directly overhead

What is at 90 degrees altitude, 180 degrees azimuth?

write in your answer here

```
In [6]: target = coordinates.SkyCoord(alt=90*u.deg, az=180*u.deg,
                                     location=observer.location,
                                     obstime=now,
                                     frame='altaz').transform_to(coordinates.Galactic())
target
```

```
Out[6]: <SkyCoord (Galactic): (l, b) in deg
        (61.65430086, 8.93281744)>
```

The "target" variable is now a Galactic coordinate that tells us where we're pointing in the galaxy *right now*.

If we want to know the RA/Dec, we can convert it:

```
In [7]: target.transform_to(coordinates.FK5())
```

```
Out[7]: <SkyCoord (FK5: equinox=J2000.000): (ra, dec) in deg
        (288.02255473, 29.55863803)>
```

Planning for future observations

What if we want to predict a future observation?

(specify time formats using <https://docs.python.org/3/library/datetime.html>
(<https://docs.python.org/3/library/datetime.html>))

```
In [8]: time = datetime.datetime.strptime('January 1, 2023 4:30:00 AM', "%B %d, %Y %I:%M:
        time
```

```
Out[8]: datetime.datetime(2023, 1, 1, 4, 30)
```

```
In [9]: target = coordinates.SkyCoord(alt=90*u.deg, az=180*u.deg,
        location=observer.location,
        obstime=time,
        frame='altaz').transform_to(coordinates.Galactic)
        target
```

```
Out[9]: <SkyCoord (Galactic): (l, b) in deg
        (178.98637743, -0.4019667)>
```

You can also change your altitude and azimuth.

Where is my target (where do I have to point)?

This is the fundamental question all observatories have to answer: given a target at a specific RA, Dec, where do we point? What altitude and azimuth do we point at?

(the question is easier if you have an equatorial mount, but you still need to know what Right Ascension to point at)

You want to know, for the purpose of observing gas in our Galaxy: Where do I have to point to see the Galactic Plane? If I want to observe the Galactic plane at different lines of longitude, then where do I have to point?

The Galactic Plane is simply the circle in Galactic coordinates where the latitude is zero degrees.

```
In [10]: # we pick a target at Galactic Longitude=45 degrees and Latitude=0 (the Galactic
        target = coordinates.SkyCoord(45*u.deg, 0*u.deg, frame='galactic')
```

```
In [11]: altaz_of_target = observer.altaz(time=time, target=target)
altaz_of_target
```

```
Out[11]: <SkyCoord (AltAz: obstime=2023-01-01 04:30:00, location=(743674.13504561, -5500
332.52750589, 3131946.80724394) m, pressure=0.0 hPa, temperature=0.0 deg_C, rel
ative_humidity=0.0, obswl=1.0 micron): (az, alt) in deg
(327.87766647, -43.99069283)>
```

```
In [12]: # print just the altitude
altaz_of_target.alt
```

```
Out[12]: -43°59'26.49418978"
```

Using the astroplan "observer", we can find the alt & az of our target at any time. In the above example, the target has an altitude of -44 degrees: not observable!

When can we see it? Let's try 12 hours later...

```
In [13]: time = datetime.datetime.strptime('January 1, 2023 4:30:00 PM', "%B %d, %Y %I:%M:
altaz_of_target2 = observer.altaz(time=time, target=target)
altaz_of_target2.alt
```

```
Out[13]: 61°51'55.46107381"
```

+61 degrees is observable

So if you have the question, "What will I see if I look at the sky right now?", you can answer it - convert alt/az to ra/dec or Galactic coordinates.

If you have the question, "When can I see my target?", you can answer it by converting target coordinates to alt/az at any given time.

Generally, though, you want to know a *range* of times. Plots help with that.

Let's say we want to plot the observability over the course of a day. We can use astropy's time for this:

```
In [14]: from astropy.time import Time, TimeDelta
         atime = Time(time)
         atimes = atime + TimeDelta(1*u.h) * np.arange(24)
         atimes
```

```
Out[14]: <Time object: scale='utc' format='datetime' value=[datetime.datetime(2023, 1,
1, 16, 30)
datetime.datetime(2023, 1, 1, 17, 30)
datetime.datetime(2023, 1, 1, 18, 30)
datetime.datetime(2023, 1, 1, 19, 30)
datetime.datetime(2023, 1, 1, 20, 30)
datetime.datetime(2023, 1, 1, 21, 30)
datetime.datetime(2023, 1, 1, 22, 30)
datetime.datetime(2023, 1, 1, 23, 30)
datetime.datetime(2023, 1, 2, 0, 30) datetime.datetime(2023, 1, 2, 1, 30)
datetime.datetime(2023, 1, 2, 2, 30) datetime.datetime(2023, 1, 2, 3, 30)
datetime.datetime(2023, 1, 2, 4, 30) datetime.datetime(2023, 1, 2, 5, 30)
datetime.datetime(2023, 1, 2, 6, 30) datetime.datetime(2023, 1, 2, 7, 30)
datetime.datetime(2023, 1, 2, 8, 30) datetime.datetime(2023, 1, 2, 9, 30)
datetime.datetime(2023, 1, 2, 10, 30)
datetime.datetime(2023, 1, 2, 11, 30)
datetime.datetime(2023, 1, 2, 12, 30)
datetime.datetime(2023, 1, 2, 13, 30)
datetime.datetime(2023, 1, 2, 14, 30)
datetime.datetime(2023, 1, 2, 15, 30)]>
```

We can then compute where our target will be (altitude, azimuth) vs time:

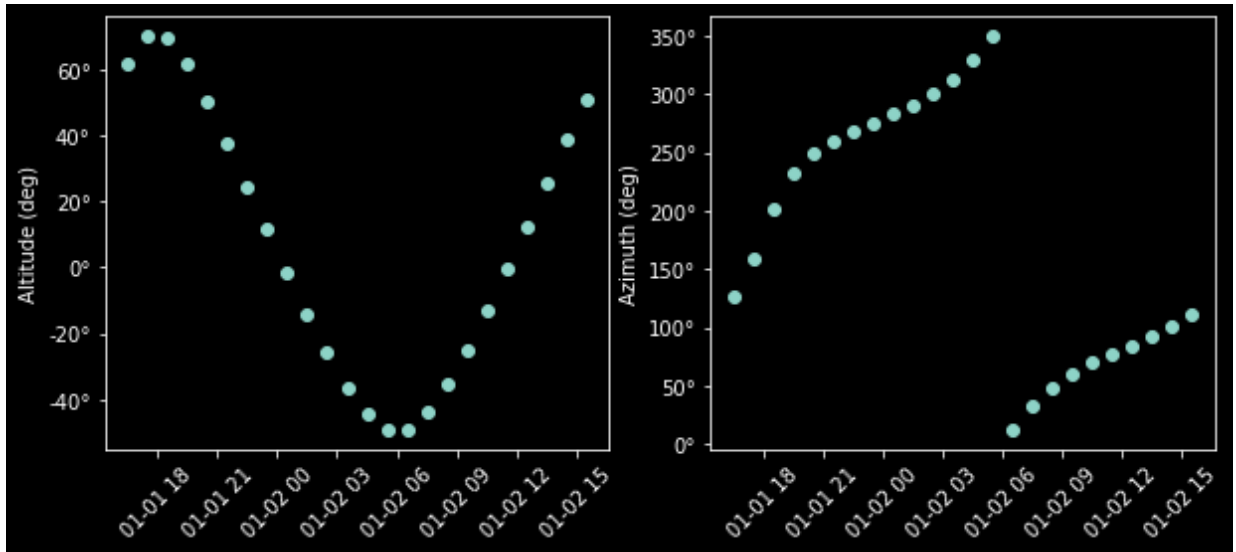
```
In [15]: altaz = observer.altaz(atimes, target=target)
altaz
```

```
Out[15]: <SkyCoord (AltAz: obstime=[datetime.datetime(2023, 1, 1, 16, 30)
datetime.datetime(2023, 1, 1, 17, 30)
datetime.datetime(2023, 1, 1, 18, 30)
datetime.datetime(2023, 1, 1, 19, 30)
datetime.datetime(2023, 1, 1, 20, 30)
datetime.datetime(2023, 1, 1, 21, 30)
datetime.datetime(2023, 1, 1, 22, 30)
datetime.datetime(2023, 1, 1, 23, 30)
datetime.datetime(2023, 1, 2, 0, 30) datetime.datetime(2023, 1, 2, 1, 30)
datetime.datetime(2023, 1, 2, 2, 30) datetime.datetime(2023, 1, 2, 3, 30)
datetime.datetime(2023, 1, 2, 4, 30) datetime.datetime(2023, 1, 2, 5, 30)
datetime.datetime(2023, 1, 2, 6, 30) datetime.datetime(2023, 1, 2, 7, 30)
datetime.datetime(2023, 1, 2, 8, 30) datetime.datetime(2023, 1, 2, 9, 30)
datetime.datetime(2023, 1, 2, 10, 30)
datetime.datetime(2023, 1, 2, 11, 30)
datetime.datetime(2023, 1, 2, 12, 30)
datetime.datetime(2023, 1, 2, 13, 30)
datetime.datetime(2023, 1, 2, 14, 30)
datetime.datetime(2023, 1, 2, 15, 30)], location=(743674.13504561, -5500332.52
750589, 3131946.80724394) m, pressure=0.0 hPa, temperature=0.0 deg_C, relative_
humidity=0.0, obswl=1.0 micron): (az, alt) in deg
[(127.36852382, 61.86540585), (158.39960251, 69.94308684),
(202.33895534, 69.85401987), (233.03468544, 61.67568521),
(249.78829477, 50.15114474), (260.48308916, 37.51096957),
(268.68920151, 24.49956428), (275.97769825, 11.4413596 ),
(283.24785201, -1.44564383), (291.23493615, -13.93219459),
(300.76988537, -25.68992216), (312.95116308, -36.17150761),
(329.07527965, -44.4388223 ), (349.63029372, -49.09499696),
( 12.22714365, -48.88038584), ( 32.45186939, -43.86343212),
( 48.20291099, -35.36233582), ( 60.11286902, -24.7445453 ),
( 69.48047555, -12.90886716), ( 77.37831138, -0.37823702),
( 84.62154171, 12.53114631), ( 91.94757984, 25.59424669),
(100.28652953, 38.58975189), (111.30754281, 51.17512469)]>
```

We can then plot it....

```
In [16]: from astropy import visualization
import pylab as pl
visualization.quantity_support() # only need this once
pl.style.use('dark_background')
```

```
In [17]: pl.figure(figsize=(10,4))
pl.subplot(1,2,1).plot_date(atimes.plot_date, altaz.alt)
pl.xticks(rotation=45)
pl.ylabel("Altitude (deg)")
pl.subplot(1,2,2).plot_date(atimes.plot_date, altaz.az)
pl.xticks(rotation=45)
pl.ylabel("Azimuth (deg)");
```



Visualizing your plan

From the "Observation Planning to Map Making" lecture, you can grab an image of the Galaxy and overplot

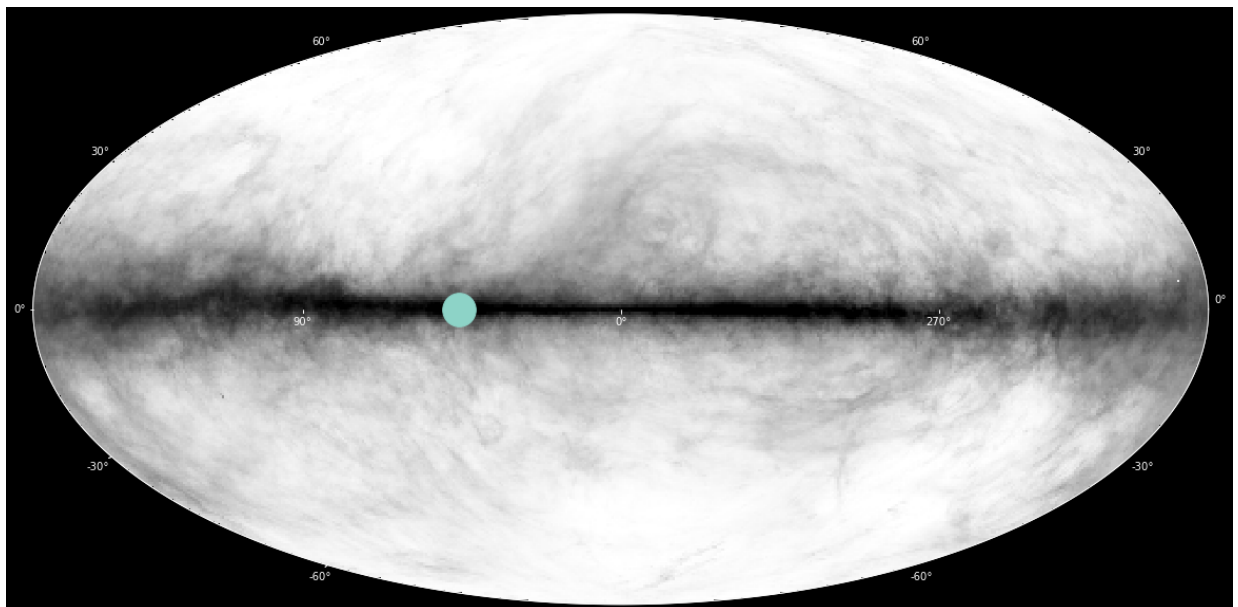
```
In [18]: from astropy.wcs import WCS
from astropy.io import fits
from astropy.utils.data import get_pkg_data_filename
from astropy.visualization.wcsaxes.frame import EllipticalFrame
from astropy.visualization import simple_norm
```

```
In [19]: # first step: we need to download an all-sky image
# Use the ROSAT image to get an all-sky header
filename = get_pkg_data_filename('allsky/allsky_rosat.fits')
hdu = fits.open(filename)[0]
wcs = WCS(hdu.header)
data = hdu.data

# let's also grab the HI data
from reproject import reproject_from_healpix, reproject_to_healpix
hihdu = fits.open('https://lambda.gsfc.nasa.gov/data/foregrounds/ebv_2017/mom0_-9
hiarray, hifootprint = reproject_from_healpix(hihdu[1], hdu.header)
```

C:\ProgramData\Anaconda3\lib\site-packages\astropy_healpix\core.py:586: Runtime Warning: invalid value encountered in bilinear_interpolation_weights
 result = _core.bilinear_interpolation_weights(lon, lat, nside)

```
In [20]: # next, make the figure with special attributes
plt.figure(figsize=(20,15))
ax = plt.subplot(projection=wcs, frame_class=EllipticalFrame)
im = ax.imshow(hiarray, origin='lower', cmap='gray_r', interpolation='none',
               norm=simple_norm(hiarray, stretch='asinh', max_percent=99.5, min_p
# and overplot a Location
ax.scatter(target.galactic.l, target.galactic.b, s=1000, transform=ax.get_transf
```



EXERCISE:

Repeat the above for three locations in the Galactic plane.

Try $l=0$, $l=90$, and $l=135$ degrees.

Can you observe all three locations in the same observing session?


```
In [21]: target0 = coordinates.SkyCoord(0*u.deg, 0*u.deg, frame='galactic')
target90 = coordinates.SkyCoord(90*u.deg, 0*u.deg, frame='galactic')
target135 = coordinates.SkyCoord(135*u.deg, 0*u.deg, frame='galactic')
```

```
In [22]: altaz_of_target0 = observer.altaz(time=time, target=target0)
print("altaz I = 0", altaz_of_target0)

altaz_of_target90 = observer.altaz(time=time, target=target90)
print("altaz I = 0", altaz_of_target90)

altaz_of_target135 = observer.altaz(time=time, target=target135)
print("altaz I = 0", altaz_of_target135)
```

```
altaz I = 0 <SkyCoord (AltAz: obstime=2023-01-01 16:30:00, location=(743674.135
04561, -5500332.52750589, 3131946.80724394) m, pressure=0.0 hPa, temperature=0.
0 deg_C, relative_humidity=0.0, obswl=1.0 micron): (az, alt) in deg
(179.49284572, 31.45274954)>
altaz I = 0 <SkyCoord (AltAz: obstime=2023-01-01 16:30:00, location=(743674.135
04561, -5500332.52750589, 3131946.80724394) m, pressure=0.0 hPa, temperature=0.
0 deg_C, relative_humidity=0.0, obswl=1.0 micron): (az, alt) in deg
(49.36342044, 46.49627604)>
altaz I = 0 <SkyCoord (AltAz: obstime=2023-01-01 16:30:00, location=(743674.135
04561, -5500332.52750589, 3131946.80724394) m, pressure=0.0 hPa, temperature=0.
0 deg_C, relative_humidity=0.0, obswl=1.0 micron): (az, alt) in deg
(21.57904426, 8.27429756)>
```

```
In [23]: # print just the altitude
print("altitude I = 0", altaz_of_target0.alt)
print("altitude I = 90", altaz_of_target90.alt)
print("altitude I = 135", altaz_of_target135.alt)
```

```
altitude I = 0 31d27m09.89833971s
altitude I = 90 46d29m46.5937419s
altitude I = 135 8d16m27.4712337s
```

```
In [24]: atime = Time(time)
          atimes = atime + TimeDelta(1*u.h) * np.arange(24)
          atimes
```

```
Out[24]: <Time object: scale='utc' format='datetime' value=[datetime.datetime(2023, 1,
1, 16, 30)
          datetime.datetime(2023, 1, 1, 17, 30)
          datetime.datetime(2023, 1, 1, 18, 30)
          datetime.datetime(2023, 1, 1, 19, 30)
          datetime.datetime(2023, 1, 1, 20, 30)
          datetime.datetime(2023, 1, 1, 21, 30)
          datetime.datetime(2023, 1, 1, 22, 30)
          datetime.datetime(2023, 1, 1, 23, 30)
          datetime.datetime(2023, 1, 2, 0, 30) datetime.datetime(2023, 1, 2, 1, 30)
          datetime.datetime(2023, 1, 2, 2, 30) datetime.datetime(2023, 1, 2, 3, 30)
          datetime.datetime(2023, 1, 2, 4, 30) datetime.datetime(2023, 1, 2, 5, 30)
          datetime.datetime(2023, 1, 2, 6, 30) datetime.datetime(2023, 1, 2, 7, 30)
          datetime.datetime(2023, 1, 2, 8, 30) datetime.datetime(2023, 1, 2, 9, 30)
          datetime.datetime(2023, 1, 2, 10, 30)
          datetime.datetime(2023, 1, 2, 11, 30)
          datetime.datetime(2023, 1, 2, 12, 30)
          datetime.datetime(2023, 1, 2, 13, 30)
          datetime.datetime(2023, 1, 2, 14, 30)
          datetime.datetime(2023, 1, 2, 15, 30)]>
```

```
In [25]: altaz0 = observer.altaz(atimes, target=target0)
altaz0
```

```
Out[25]: <SkyCoord (AltAz: obstime=[datetime.datetime(2023, 1, 1, 16, 30)
datetime.datetime(2023, 1, 1, 17, 30)
datetime.datetime(2023, 1, 1, 18, 30)
datetime.datetime(2023, 1, 1, 19, 30)
datetime.datetime(2023, 1, 1, 20, 30)
datetime.datetime(2023, 1, 1, 21, 30)
datetime.datetime(2023, 1, 1, 22, 30)
datetime.datetime(2023, 1, 1, 23, 30)
datetime.datetime(2023, 1, 2, 0, 30) datetime.datetime(2023, 1, 2, 1, 30)
datetime.datetime(2023, 1, 2, 2, 30) datetime.datetime(2023, 1, 2, 3, 30)
datetime.datetime(2023, 1, 2, 4, 30) datetime.datetime(2023, 1, 2, 5, 30)
datetime.datetime(2023, 1, 2, 6, 30) datetime.datetime(2023, 1, 2, 7, 30)
datetime.datetime(2023, 1, 2, 8, 30) datetime.datetime(2023, 1, 2, 9, 30)
datetime.datetime(2023, 1, 2, 10, 30)
datetime.datetime(2023, 1, 2, 11, 30)
datetime.datetime(2023, 1, 2, 12, 30)
datetime.datetime(2023, 1, 2, 13, 30)
datetime.datetime(2023, 1, 2, 14, 30)
datetime.datetime(2023, 1, 2, 15, 30)], location=(743674.13504561, -5500332.52
750589, 3131946.80724394) m, pressure=0.0 hPa, temperature=0.0 deg_C, relative_
humidity=0.0, obswl=1.0 micron): (az, alt) in deg
[(179.49284572, 31.45274954), (194.67660771, 29.83026406),
(208.47276663, 24.99917358), (220.1626541, 17.60428357),
(229.76462076, 8.34734086), (237.64354981, -2.20190686),
(244.21825367, -13.63662536), (249.85051705, -25.67915719),
(254.83290309, -38.14016482), (259.41912004, -50.88728748),
(263.91040991, -63.82364662), (269.14650012, -76.87123139),
(359.88921938, -89.34520498), ( 90.85216252, -76.87375015),
( 96.0886313, -63.82616293), (100.57993988, -50.88979306),
(105.1660956, -38.14264791), (110.14836723, -25.68159822),
(115.7804595, -13.63899288), (122.35492243, -2.20415182),
(130.23352495, 8.34529289), (139.83507425, 17.60254123),
(151.52448984, 24.99788212), (165.32024247, 29.82958339)]>
```

```
In [26]: altaz90 = observer.altaz(atimes, target=target90)
altaz90
```

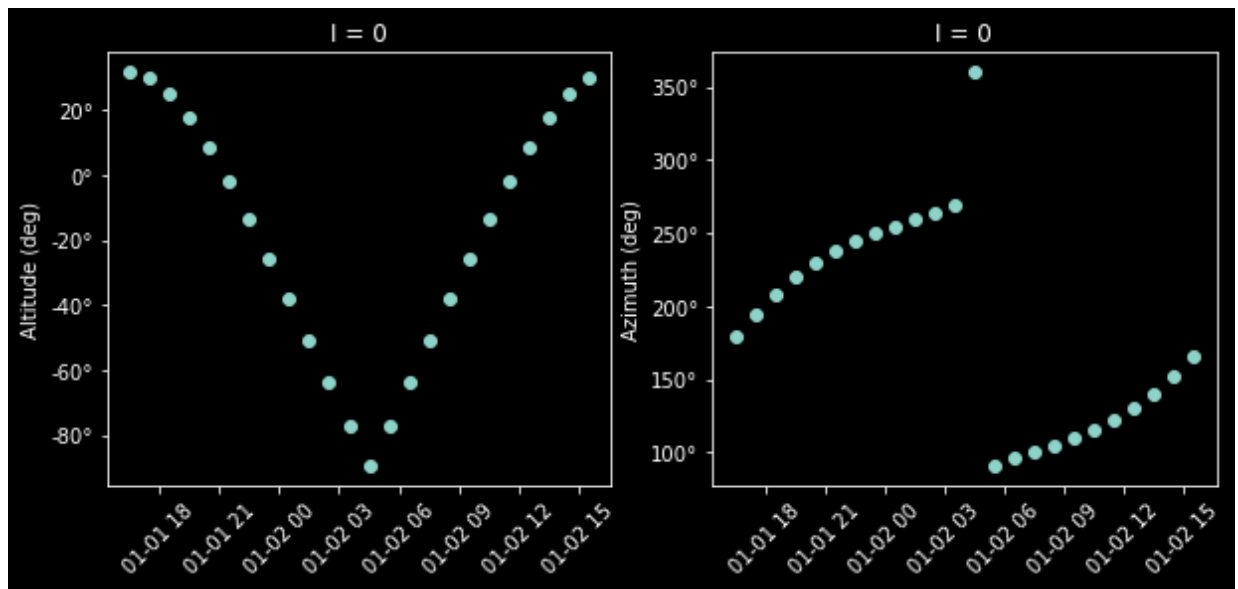
```
Out[26]: <SkyCoord (AltAz: obstime=[datetime.datetime(2023, 1, 1, 16, 30)
datetime.datetime(2023, 1, 1, 17, 30)
datetime.datetime(2023, 1, 1, 18, 30)
datetime.datetime(2023, 1, 1, 19, 30)
datetime.datetime(2023, 1, 1, 20, 30)
datetime.datetime(2023, 1, 1, 21, 30)
datetime.datetime(2023, 1, 1, 22, 30)
datetime.datetime(2023, 1, 1, 23, 30)
datetime.datetime(2023, 1, 2, 0, 30) datetime.datetime(2023, 1, 2, 1, 30)
datetime.datetime(2023, 1, 2, 2, 30) datetime.datetime(2023, 1, 2, 3, 30)
datetime.datetime(2023, 1, 2, 4, 30) datetime.datetime(2023, 1, 2, 5, 30)
datetime.datetime(2023, 1, 2, 6, 30) datetime.datetime(2023, 1, 2, 7, 30)
datetime.datetime(2023, 1, 2, 8, 30) datetime.datetime(2023, 1, 2, 9, 30)
datetime.datetime(2023, 1, 2, 10, 30)
datetime.datetime(2023, 1, 2, 11, 30)
datetime.datetime(2023, 1, 2, 12, 30)
datetime.datetime(2023, 1, 2, 13, 30)
datetime.datetime(2023, 1, 2, 14, 30)
datetime.datetime(2023, 1, 2, 15, 30)], location=(743674.13504561, -5500332.52
750589, 3131946.80724394) m, pressure=0.0 hPa, temperature=0.0 deg_C, relative_
humidity=0.0, obswl=1.0 micron): (az, alt) in deg
[( 49.36342044,  46.49627604), ( 45.72928332,  56.20313619),
( 35.4958877 ,  64.83382598), ( 13.59798239,  70.46627363),
(343.74325614,  70.14484544), (323.10564348,  64.09714685),
(313.70689796,  55.30570198), (310.50456639,  45.54880572),
(310.65195689,  35.58429233), (312.91931041,  25.81075572),
(316.79574298,  16.5221276 ), (322.09002623,   8.00179421),
(328.74276691,   0.56170373), (336.70240621,  -5.44843686),
(345.81317716,  -9.66447593), (355.72728662, -11.77005424),
(   5.90177568, -11.58342833), ( 15.71900073,  -9.12142632),
( 24.66234099,  -4.59309198), ( 32.41817755,   1.67188784),
( 38.85463767,   9.3085008 ), ( 43.9266083 ,  17.97250533),
( 47.562542   ,  27.35790664), ( 49.5355239 ,  37.18317413)]>
```

```
In [27]: altaz135 = observer.altaz(atimes, target=target135)
altaz135
```

```
Out[27]: <SkyCoord (AltAz: obstime=[datetime.datetime(2023, 1, 1, 16, 30)
datetime.datetime(2023, 1, 1, 17, 30)
datetime.datetime(2023, 1, 1, 18, 30)
datetime.datetime(2023, 1, 1, 19, 30)
datetime.datetime(2023, 1, 1, 20, 30)
datetime.datetime(2023, 1, 1, 21, 30)
datetime.datetime(2023, 1, 1, 22, 30)
datetime.datetime(2023, 1, 1, 23, 30)
datetime.datetime(2023, 1, 2, 0, 30) datetime.datetime(2023, 1, 2, 1, 30)
datetime.datetime(2023, 1, 2, 2, 30) datetime.datetime(2023, 1, 2, 3, 30)
datetime.datetime(2023, 1, 2, 4, 30) datetime.datetime(2023, 1, 2, 5, 30)
datetime.datetime(2023, 1, 2, 6, 30) datetime.datetime(2023, 1, 2, 7, 30)
datetime.datetime(2023, 1, 2, 8, 30) datetime.datetime(2023, 1, 2, 9, 30)
datetime.datetime(2023, 1, 2, 10, 30)
datetime.datetime(2023, 1, 2, 11, 30)
datetime.datetime(2023, 1, 2, 12, 30)
datetime.datetime(2023, 1, 2, 13, 30)
datetime.datetime(2023, 1, 2, 14, 30)
datetime.datetime(2023, 1, 2, 15, 30)], location=(743674.13504561, -5500332.52
750589, 3131946.80724394) m, pressure=0.0 hPa, temperature=0.0 deg_C, relative_
humidity=0.0, obswl=1.0 micron): (az, alt) in deg
[( 21.57904426,  8.27429756), ( 26.72172301, 13.64167783),
( 30.7089491 , 19.94462898), ( 33.34833398, 26.90342846),
( 34.37069179, 34.21821138), ( 33.3455376 , 41.54123522),
( 29.6025148 , 48.41901405), ( 22.28446767, 54.2045365 ),
( 10.93989433, 58.00681353), (356.93630615, 58.92429062),
(343.73502714, 56.68472143), (334.09109598, 51.91186986),
(328.40783705, 45.56307654), (325.9278951 , 38.42883649),
(325.84664268, 31.06171026), (327.60296536, 23.86059024),
(330.84645041, 17.14755574), (335.35042045, 11.21194709),
(340.93287616,  6.32729147), (347.39273196,  2.74618475),
(354.4672191 ,  0.67719548), ( 1.82263586,  0.25227383),
(  9.08467684,  1.49969817), ( 15.89312042,  4.33762109)]>
```

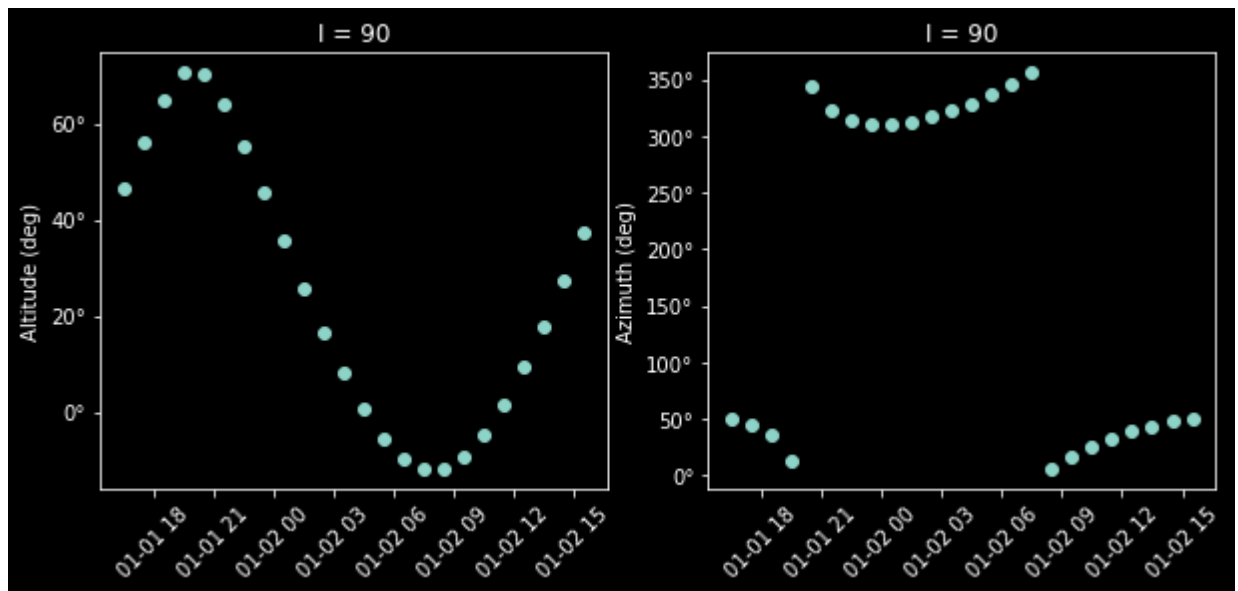
```
In [31]: pl.figure(figsize=(10,4))
pl.subplot(1,2,1).plot_date(atimes.plot_date, altaz0.alt)
pl.xticks(rotation=45)
pl.ylabel("Altitude (deg)")
pl.title("I = 0")
pl.subplot(1,2,2).plot_date(atimes.plot_date, altaz0.az)
pl.xticks(rotation=45)
pl.ylabel("Azimuth (deg)")
pl.title("I = 0")
```

Out[31]: Text(0.5, 1.0, 'I = 0')



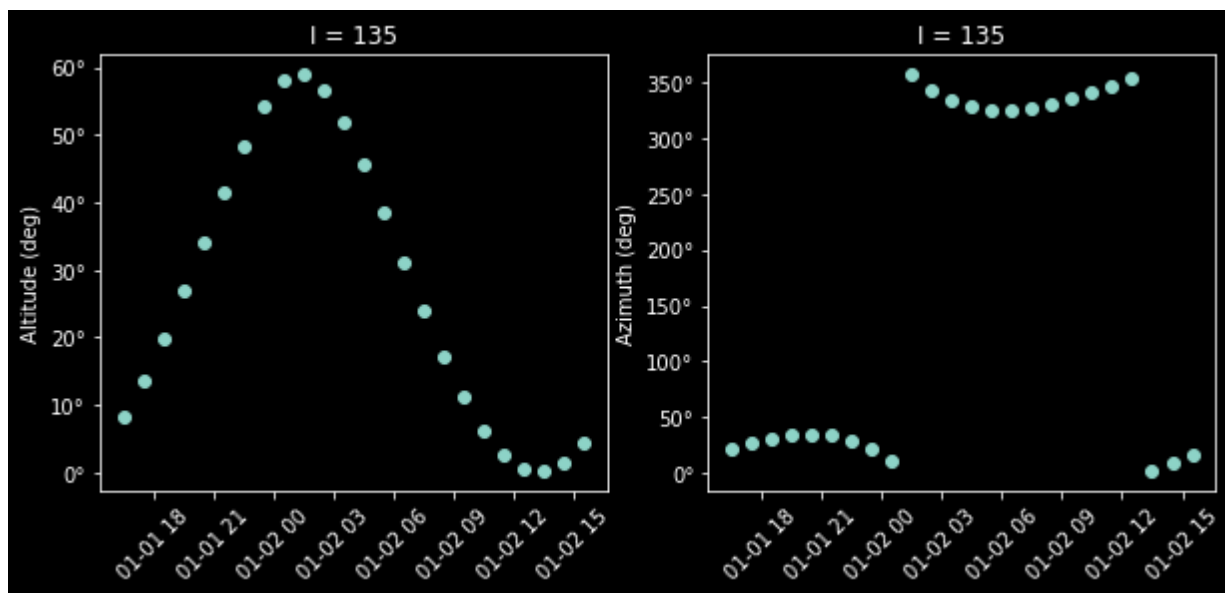
```
In [32]: pl.figure(figsize=(10,4))
pl.subplot(1,2,1).plot_date(atimes.plot_date, altaz90.alt)
pl.xticks(rotation=45)
pl.ylabel("Altitude (deg)")
pl.title("I = 90")
pl.subplot(1,2,2).plot_date(atimes.plot_date, altaz90.az)
pl.xticks(rotation=45)
pl.ylabel("Azimuth (deg)")
pl.title("I = 90")
```

Out[32]: Text(0.5, 1.0, 'I = 90')



```
In [33]: pl.figure(figsize=(10,4))
pl.subplot(1,2,1).plot_date(atimes.plot_date, altaz135.alt)
pl.xticks(rotation=45)
pl.ylabel("Altitude (deg)")
pl.title("I = 135")
pl.subplot(1,2,2).plot_date(atimes.plot_date, altaz135.az)
pl.xticks(rotation=45)
pl.ylabel("Azimuth (deg)")
pl.title("I = 135")
```

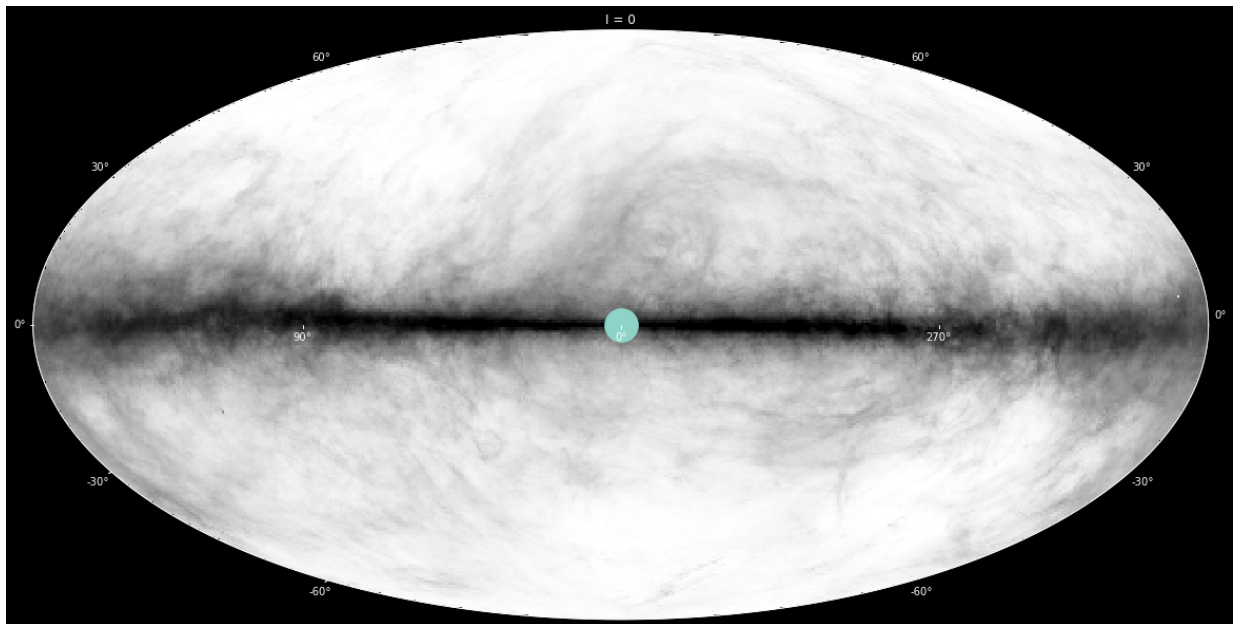
Out[33]: Text(0.5, 1.0, 'I = 135')




```
In [38]: # next, make the figure with special attributes
pl.figure(figsize=(20,15))
ax = pl.subplot(projection=wcs, frame_class=EllipticalFrame)
im = ax.imshow(hiarray, origin='lower', cmap='gray_r', interpolation='none',
               norm=simple_norm(hiarray, stretch='asinh', max_percent=99.5, min_p

# and overplot a Location
ax.scatter(target0.galactic.l, target0.galactic.b, s=1000, transform=ax.get_trans
ax.set_title("I = 0")
```

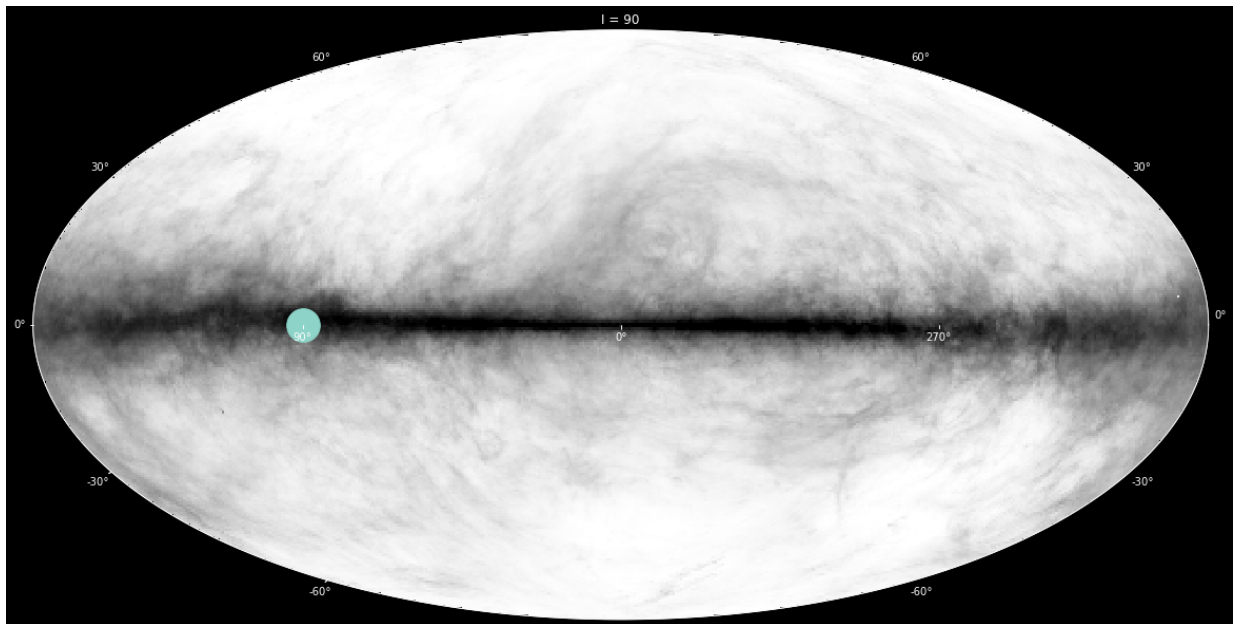
Out[38]: Text(0.5, 1.0, 'I = 0')



```
In [39]: # next, make the figure with special attributes
pl.figure(figsize=(20,15))
ax = pl.subplot(projection=wcs, frame_class=EllipticalFrame)
im = ax.imshow(hiarray, origin='lower', cmap='gray_r', interpolation='none',
               norm=simple_norm(hiarray, stretch='asinh', max_percent=99.5, min_p

# and overplot a Location
ax.scatter(target90.galactic.l, target90.galactic.b, s=1000, transform=ax.get_tra
ax.set_title("I = 90")
```

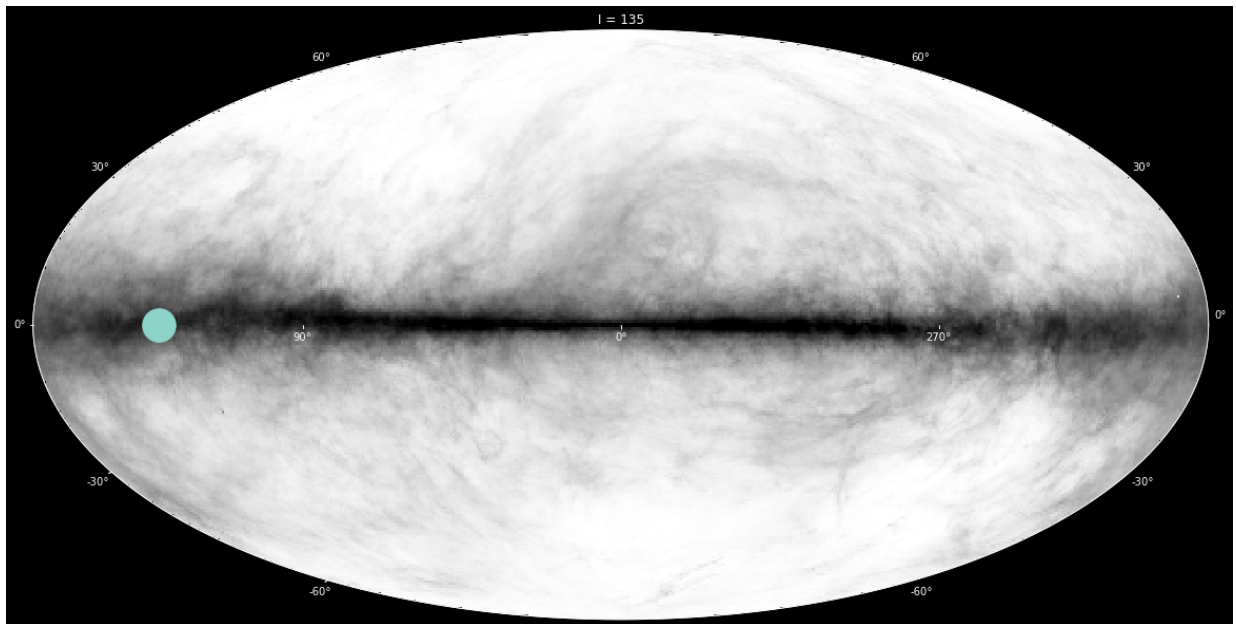
Out[39]: Text(0.5, 1.0, 'I = 90')



```
In [40]: # next, make the figure with special attributes
pl.figure(figsize=(20,15))
ax = pl.subplot(projection=wcs, frame_class=EllipticalFrame)
im = ax.imshow(hiarray, origin='lower', cmap='gray_r', interpolation='none',
               norm=simple_norm(hiarray, stretch='asinh', max_percent=99.5, min_p

# and overplot a Location
ax.scatter(target135.galactic.l, target135.galactic.b, s=1000, transform=ax.get_t
ax.set_title("I = 135")
```

Out[40]: Text(0.5, 1.0, 'I = 135')



you can probably observe $l = 90$ and $l = 135$ in the same sessions

In []: