

Calculating potential binary stars' periods using data from the Sloan Digital Sky Survey (SDSS)

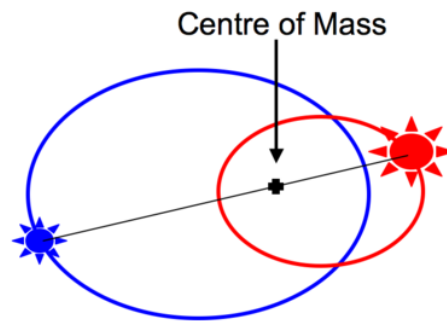
Short description

In this project, I calculated the period of likely binary stars using radial velocity time-series data from the SDSS's APOGEE survey. I acquired the data using SQL from the SDSS's Sky Server interface. The algorithm I used for finding the periods is the Lomb-Scargle periodogram. My code is written in Python and mainly uses the NumPy and AstroPy libraries. The period calculated shows that most binary stars complete their orbit in several days, while others do so in a few hours or many years.

Scientific background

1. Binary star

In astronomy, binary stars are 2 stars that are gravitationally bound to each other and revolve around one another. Their orbits can be either elliptical or circular, as shown in the figure below. The time in which each star in the system finishes its orbit is called the binary system's orbital period.

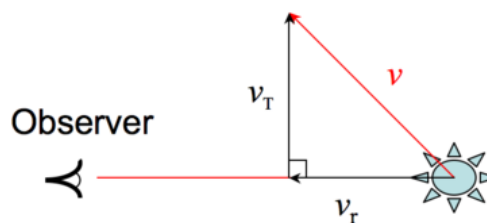


2 binary stars revolving around each other in elliptical orbits.

Source: <https://astronomy.swin.edu.au/cosmos/b/binary+star>

2. Radial velocity

Radial velocity is the vertical projection of the velocity vector on the line connecting the observer and the star.



v : velocity of the star. v_r : radial component of v .

Source: <https://astronomy.swin.edu.au/cosmos/r/Radial+velocity>

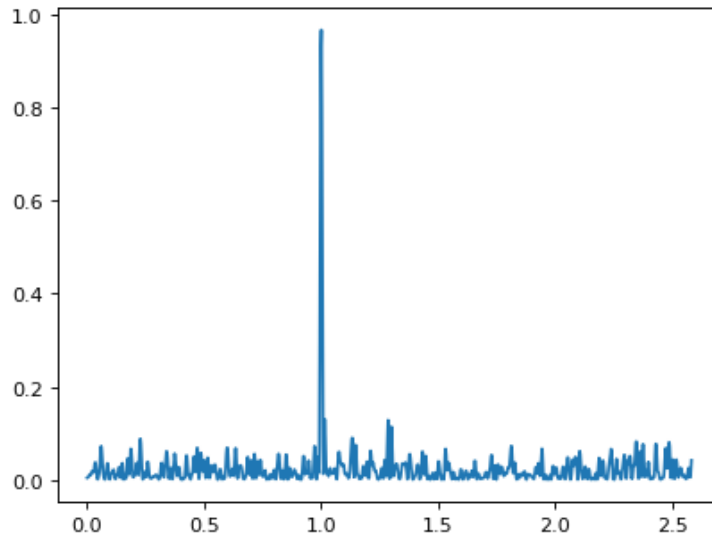
3. SDSS and APOGEE

SDSS⁽¹⁾ (Sloan Digital Sky Survey) is a major astronomical survey using a 2.5m telescope at the APO in New Mexico, US and the Irénée du Pont Telescope at Las Campanas Observatory, Chile. Beginning its operation in 2000, the SDSS has gone through 4 phases, and each comprises multiple surveys corresponding to different goals.

APOGEE⁽²⁾ (Apache Point Observatory (APO) Galactic Evolution Experiment) is one of those surveys. It contains two versions, which lasted from 2011 - 2014 and 2014 – 2021, respectively. APOGEE measures the chemical element abundance and dynamic motions of stars in the Milky Way. It does so by observing a star multiple times across many nights. We call each time a star is observed a *visit*. To systematically organize the data, each star in APOGEE is given an ID called *apogee_id*. There is also an ID for each visit, called *apvisit_id*.

4. Lomb-Scargle periodogram algorithm

This algorithm is used for “detecting and characterizing periodic signals in unevenly sampled data”⁽³⁾. It uses a time series as its input and generates a power-vs-frequency graph. Here, *power* shows how much a sinusoidal curve with a particular frequency fits the available data. The higher the power, the more the curve fits the data.



An example of a Lomb-Scargle periodogram.

Source: <https://docs.astropy.org/en/stable/timeseries/lombscargle.html>

1. <https://www.sdss4.org/>
2. <https://www.sdss4.org/dr17/irspec/>
3. <https://iopscience.iop.org/article/10.3847/1538-4365/aab766>

Method

The project has 2 main steps: querying data of acceptable visits of likely binary stars, then calculating their orbital periods using the acquired data.

1. Querying data

I queried data from the SDSS database using SQL on the [SDSS's Sky Server](#). An overview of the database (e.g., tables, fields in a table) is given [here](#).

The *binaryStarVisit100000Query.sql* script first searches for binary stars from the *apogeeStar* table, based on the standard set out [here](#). According to this standard, I can confidently conclude that a star belongs to a binary system if the following 3 conditions are satisfied:

- +) RV's standard deviation (*vscatter*) is greater than 1 km/s
- +) RV's standard deviation (*vscatter*) is 10 times greater than its RV error (*verr*)
- +) The star has “many” acceptable visits. An acceptable visit is one which does NOT have a low signal-to-noise (SNR) ratio and does NOT have bitmask⁽¹⁾ #0, 4, 19, 22 turned on. The exact number to satisfy “many” is up to the analyst's discretion; the higher the number of visits, the more confident the analyst will be about their findings' accuracy (I personally used 8 visits as the boundary). These conditions are displayed in lines 12-16 of *binaryStarVisit100000Query.sql*

The above describes the code portion that finds potential binary stars. To calculate the orbital periods, I need to access the data from every visit of those stars. Accessing this data is done by joining *apogeeStar* via *apstar_id* with *apogeeStarVisit*, which is in turn joined via *visit_id* with *apogeeVisit*. Having accessed the relevant visits' data, I extracted necessary fields for the LS algorithm. They are

- +) *mjd* (which stands for Modified Julian Date, a time unit used in astronomy; 1 Julian Date = 1 Earth day)
- +) *vhelio* (derived RV in a visit)
- +) *vrelerr* (RV error)

After running the SQL query, I downloaded the data as a csv file:
binaryStarVisit100000.csv

1. Each bitmask, which can either be 0 or 1, indicates possible issues with each visit. For details, see <https://www.sdss4.org/dr17/irspec/apogee-bitmasks#StarBitMask>

2. Calculate orbital periods

This is done in the `computePeriod` function. The function takes the csv file mentioned above as input and outputs an array containing a star's *apogee_id* and its calculated period.

In this function, I first open the csv file and load data from it to an array. After that, there is a loop that traverses through every row of the data array. In each iteration, I separated every visit of a particular star, extracted the RV time-series data from those visits (`vhelio`, `vrelerr`, `mjd`), and used the Lomb-Scargle (LS) periodogram algorithm to find the frequency of the best-fit sinusoidal curve. From this frequency, I derive the period by taking the frequency's reciprocal. Function `csv_conversion` is used to save the period into a csv file.

Results

Results are saved in csv format in the file: `binaryStarPeriod100000.csv`. Period is mostly several days, while some are a few hours or thousands of days.

binaryStarPeriod1000

apogee_id	period (days)
2M00000662+7528598	1.03
2M00001071+6258172	1.03
2M00005196+0022164	1.06
2M00013842+7350554	1.81
2M00020962+0055374	1.0
2M00022269+6254032	1.51
2M00023036+8524194	1.04