Solar System Birthday Calculator Analysis
C.J. Hapich

I have written a code to determine the date of a person's next birthday on 10 worlds of the solar system: the 8 planets, Ceres, and Pluto. The code also tells you when your next half-birthday will be, how old you are currently, how long the year is, and the length of your birth week on each world.

The length of a year on Earth in this code is defined as the length of the Gregorian calendar year, 365.2425 days, which is the average number of days in a year over the calendar's 400-year cycle of selectively adding leap days. This length is close to the length of the tropical year (a mere 0.0003 days or 26 seconds shorter) which is the length of time between successive equinoxes. This standard for Earth makes sense, as the historical function of the calendar has been primarily to track the seasons for agricultural purposes.

The length of the year on all other worlds is defined as the sidereal year, or the length of time it takes a world to orbit the sun relative to distant stars. This different definition was chosen because not all tropical years are known for other worlds of the solar system. Furthermore, when one thinks of having turned a year older, they typically think of having taken another lap around the sun which is best captured by the sidereal year.

The next birthday on every world besides Earth is calculated by rounding one's current age up, taking the difference between that age and one's current age, converting that time in years to a time in earth days, rounding down to a whole number of days, then determining what calendar day that new date will fall on. This procedure won't work on Earth, as is shown by the following figure:

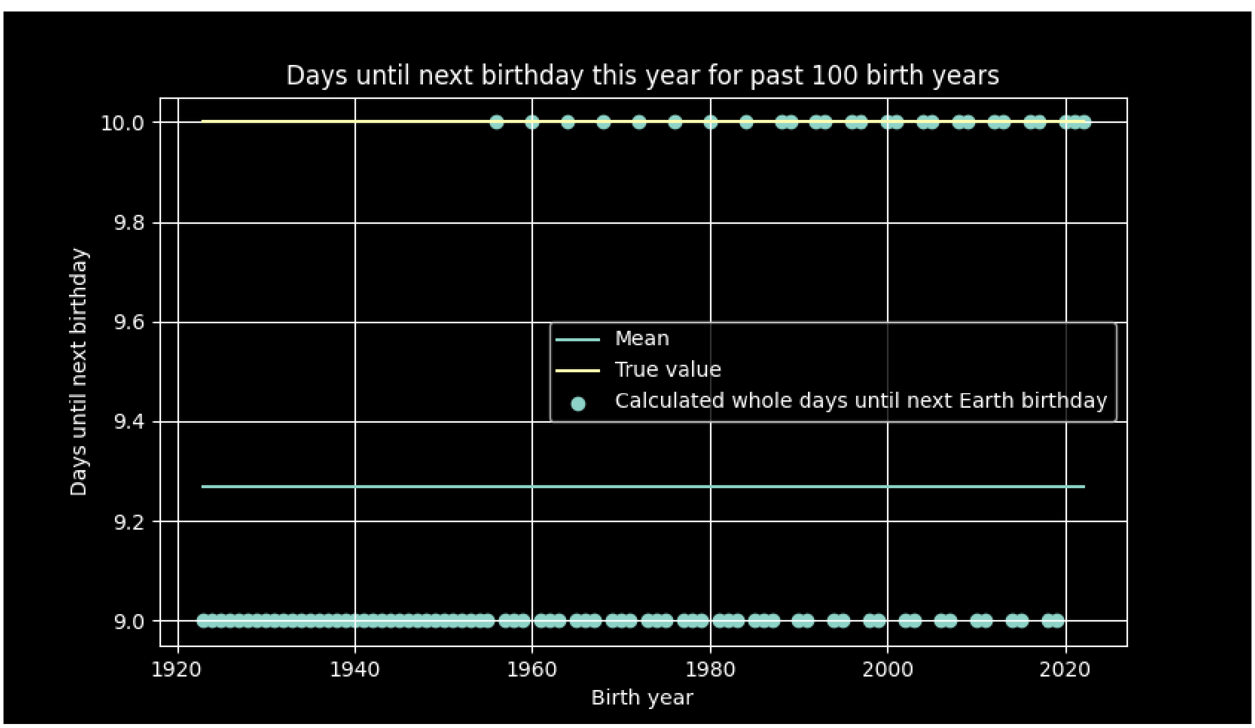

Figure 1. Days until next birthday this year provided the past 100 birth years.

A supplementary code was written to determine why incorrect next birthdays were calculated for certain birth years. This code used the next birthday procedure described above to predict how many days until a hypothetical next birthday would occur this year for someone who was born 10 calendar days from now but on each year in the past 100 years. For example, running the code on 8/10/2022 would find how many days until this year's birthday for a person born on 8/20/2021, 8/20/2020, 8/20/2019, and so on, back to a date of birth of 8/20/1922. The results in Figure 1 show that the procedure is incorrect for approximately 75% of these days of birth. To try and figure out why, let's take a look at the unrounded days:

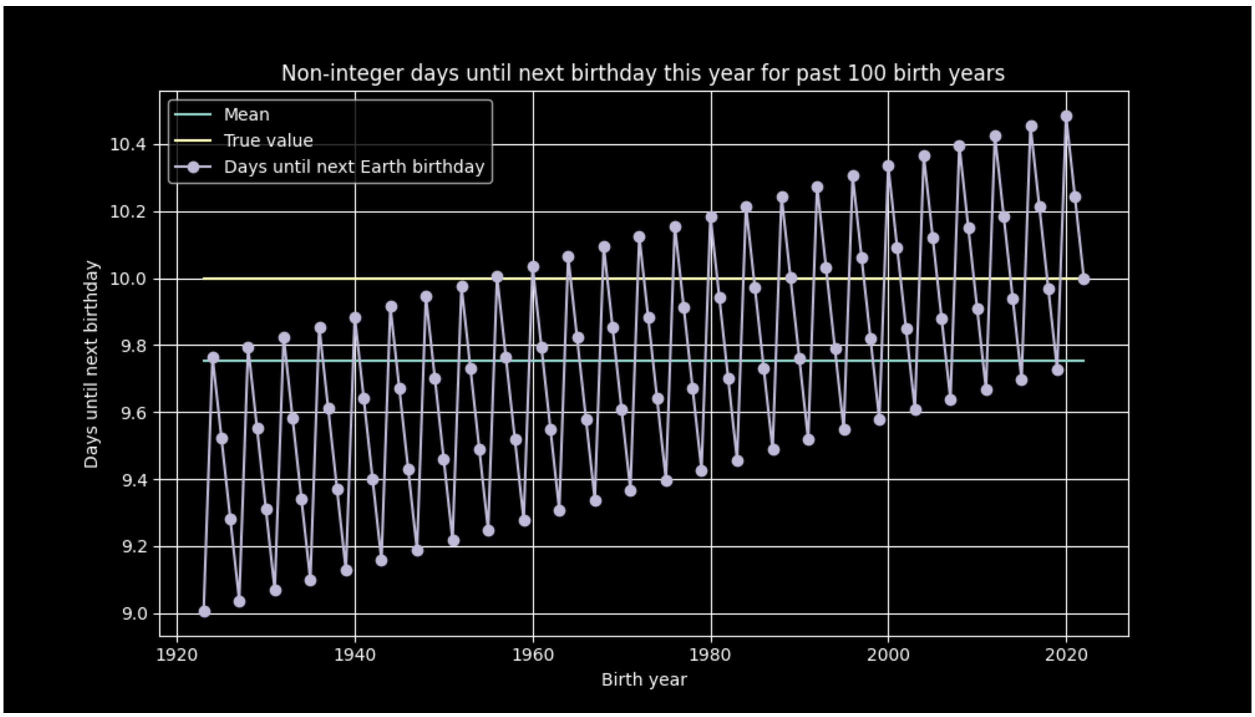

Figure 2. Unrounded days until next birthday for past 100 birth years.

Most noticeable in Figure 2 is a four-year trend. As one moves backwards in date of birth, the number of days until the next birthday go up as multiples of 0.2425 pile up. Each time a leap year is passed (e.g., 2020, 2016, etc.), however, one day is subtracted from four instances of 0.2425 excess days—an overcorrection. This accounts for the downward trend of the 100-year graph.

The full Gregorian calendar cycle is 400 years, as years that are divisible by 100 omit leap years unless they are also divisible by 400. The full cycle shows the long-term correcting behavior of the leap year:

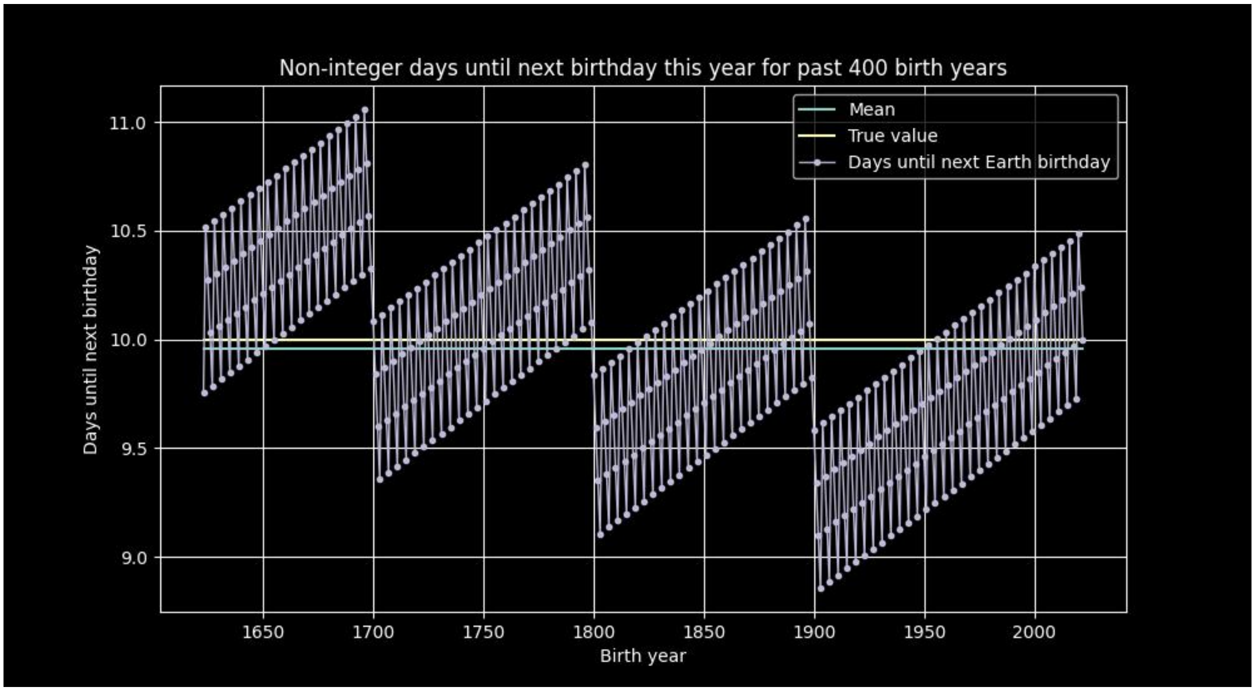

Figure 3. Unrounded days until next birthday for past 400 birth years.

Over 400 years, the selective placement of leap years does place the number of days per year at the mathematical average of the Gregorian calendar. However, calculating the elapsed time between the date of birth and the next birthday using the Gregorian calendar year length may lead to errors as high as plus or minus 1 day (or minus 2 days when rounding down some of the most extreme discrepancies). For this reason, my code cheats by using a separate function for Earth to simply return the provided birth month and day with the appropriate next birthday year. While it may have been fun to instead try and build a better calendar, this solution seems to be the least computationally taxing!