

Every Eclipse Visible from Your Current Location for the Rest of Your Life

Contents

<i>Submission Information</i>	1
<i>Abstract</i>	1
<i>Broad Topic Covered / Learning Goals</i>	4
Timezone Preference(s)	5
Duration	5
Language	5
Intended Audience	5
Pedagogical Method of Teaching	5
Maximum Number of Attendees	6
Motivation to Teach this Tutorial at userR!	6
Supplemental Material Links	6
Web Pages	6
Prerequisites / requirements	6
License, Material sharing, recording consent	6
<i>Biography</i>	6

Submitted: 2025-03-10 Author: Timothy J. Bender Email: bendertj@gmail.com Phone: +1 (513) 477-8924

Submission Information

- Author: Timothy J. Bender
- Submitted on: March 10, 2025
- email: bendertj@gmail.com
- phone: +1 (513) 477-8924

Abstract

One hobbyist's attempt (using R) to never again miss a solar eclipse:

Background

Since 2017 there have been 3 major solar eclipse events visible in North Carolina

- a total solar eclipse: August 21, 2017 (73% obscuration),
- an annular solar eclipse: October 14, 2023 (23% obscuration), and
- a total solar eclipse: April 8, 2024 (14% obscuration)

My kids and I witnessed the 2017 total eclipse¹ from our home in Durham, NC, In 2024 we drove to Hudson, OH to experience around 3 minutes of totality² and view the sun's corona³. But the 2023 eclipse slipped right by without us noticing. Maybe because the 2023 eclipse was an annular⁴ eclipse lacking the conditions necessary to view the sun's corona or experience totality, and thus less widely shared on social media.

However, I remembered a day in the fall of 2023 when the sky began to darken for no apparent reason and realized later - while making travel plans for the 2024 eclipse - that I actually had experienced the 2023 solar eclipse without even realizing it. The day the sky darkened was in October, and the only good explanation was the annular eclipse that month which would've blocked approximately 25% of the sun at it's peak.

That's the moment I knew I would build the tool I wanted - a way to never miss a solar (or lunar) eclipse again, no matter where I'm at on Earth. No more trekking across the interweb cross-checking various websites for generalized eclipse information. One website, any date, any lon/lat, every eclipse date from now until anyone alive right now can reasonably expect to still be alive.

A shiny webapp that calculates all solar and lunar eclipses for up to the next 75 years visible at your current lon/lat coordinate location. 100 miles distance is enough for 2 locations to have vastly different experiences.

Solution - Build a shiny app

- A shiny app that will use your current location (or a specific address input by user) to identify every solar and lunar eclipse viewable from that location for the next (n) years.
- Current functioning webapp: Future Solar and Lunar Eclipses Visible from Your Current Location (Shiny webapp)
- Relying primarily on the following packages/libraries:
 - *sweephR* (via CRAN): High Precision Swiss Ephemeris package. Used to determine when, where and how solar and lunar eclipses will occur (among other novel celestial events).
 - *leaflet* (via CRAN): Create Interactive Web Maps with the JavaScript 'Leaflet' Library
 - *shiny* (via CRAN): Web Application Framework for R.
 - *censusxy* (via github): Designed to provide easy access to the U.S. Census Bureau Geocoding Tools
 - *qrcode* (via CRAN): Create static QR codes in R. Implemented so that users could share between mobile devices.

¹A total solar eclipse happens when the Moon passes between the Sun and Earth, completely blocking the face of the Sun. Weather permitting, people in the path of a total solar eclipse can see the Sun's corona, the outer atmosphere, which is otherwise usually obscured by the bright face of the Sun. (Source: NASA)

²Totality is when the Moon obscures the entire disk of the Sun and only the solar corona (the sun's outer atmosphere) is visible (Source: Wikipedia)

³A stellar corona is a star's outer atmosphere. In the case of our solar system's star, the sun, the corona is only visible during a brief period of totality during a total solar eclipse when the moon completely blocks the disk of the sun (Source: Wikipedia)

⁴An annular solar eclipse happens when the Moon passes between the Sun and Earth, but when it is at or near its farthest point from Earth. Because the Moon is farther away from Earth, it appears smaller than the Sun and does not completely cover the Sun. As a result, the Moon appears as a dark disk on top of a larger, bright disk, creating what looks like a ring around the Moon. (Source: NASA)

The following code demonstrates in a simple way how the `sweephR` package is used to calculate the next global solar eclipse to occur on or after the start of our conference. You could update this code with different `input_` variables and begin to explore how outcomes differ based on time and geographic location. Extended documentation for the package is available [here](#).

```
# SOME CODE to capture solar eclipse information
library(sweephR)
library(lubridate)
library(scales)

## Example Input Variables
input_lonlat <- c("lon" = -78.938, "lat" = 36.001) # Duke Univ
input_gregtime <- ymd_hms("2025-08-08 01:00:00", # Start of useR! 2025
                        tz = "America/New_York") |>
  with_tz("UTC")

## Example Solar Eclipse Calculation
# convert gregorian time to julian time (ephemeral time i.e. "ET")
var_jultime <- swe_utc_to_jd(year = year(input_gregtime),
                           month = month(input_gregtime),
                           day = mday(input_gregtime),
                           houri = hour(input_gregtime),
                           min = minute(input_gregtime),
                           sec = second(input_gregtime),
                           gregflag = SE$GREG_CAL)$dret[2]

# calculate the next solar eclipse visible anywhere on earth
nextGlobEcl_Jd <- swe_sol_eclipse_when_glob(jd_start = var_jultime,
                                           ephe_flag = SE$FLG_MOSEPH,
                                           ifltype = 0,
                                           backward = F)$tret[c(1)]

# convert to gregorian time
nextEcl.temp <- swe_jdet_to_utc(jd_et = nextGlobEcl_Jd,
                              gregflag = SE$GREG_CAL)

output_gregtime <- paste(nextEcl.temp$year_out, "-",
                        nextEcl.temp$month_out, "-",
                        nextEcl.temp$day_out, " ",
                        nextEcl.temp$hour_out, ":",
                        nextEcl.temp$min_out, ":",
                        nextEcl.temp$sec_out,
                        sep = "") |>
  ymd_hms()

# Will the eclipse be visible from our location?
output_visible <- swe_sol_eclipse_how(jd_ut = nextGlobEcl_Jd,
                                     ephe_flag = SE$FLG_MOSEPH,
                                     geopos = c(input_lonlat["lon"],
                                                input_lonlat["lat"], 0))

output_obscuratation <- output_visible$attr[3]

output_visible <- ifelse(output_visible$return == 0 | # "no eclipse visible"
                        output_visible$attr[3] == 0, # % of sun blocked by moon == 0.0
```

```

        yes = F, no = T) |>
  ifelse(yes = "Visible from input location",
        no = "Not visible from input location")

# Where is the best viewing location geometrically?
output_xy <- swe_sol_eclipse_where(jd_ut      = nextGlobEcl_Jd,
                                  ephe_flag = SE$FLG_MOSEPH)$pathpos[c(1,2)]
names(output_xy) <- c("lon", "lat")

## Example Returned Values
print(list("Eclipse_DateTime"      = output_gregtime,
           "Visible_Locally"       = output_visible,
           "Obscuration_Locally"   = percent(output_obscurtion,
                                              accuracy = 0.1),
           "Maximal_View.Globally" = output_xy))

## $Eclipse_DateTime
## [1] "2025-09-21 19:40:51 UTC"
##
## $Visible_Locally
## [1] "Not visible from input location"
##
## $Obscuration_Locally
## [1] "0.0%"
##
## $Maximal_View.Globally
##      lon      lat
## 153.38805 -61.03134

```

Broad Topic Covered / Learning Goals

Julius Caesar vs Pope Gregory XIII: The Battle for Space-Time

- While most of the western world adopted the Gregorian calendar (365.2425 days long) by the 20th century for agricultural and cultural reasons, astronomers track time off-earth using the older Julian calendar (365.25 days long). We will briefly touch on why this is (it's not wrong), how to convert back-and-forth between Gregorian and Julian to perform astronomical calculations, and other notable phenomenon to keep in mind when using R in space.

User Input: Flexibility vs Ease-of-Use

- Specifically with regards to earthly longitude/latitude coordinate inputs (required for calculating the alignment of celestial bodies to a Earth-bound viewer), a decision had to be made affecting usability of the shiny app for the end-user based on their ability to easily enter an input location:
 - If required to enter lon/lat coordinates manually, that may be a barrier for some users that would prevent them from using the tool. Users may also explore other locations around the world.
 - If the app pulled the lon/lat coordinates automatically from the user's i.p. address this would likely be the easiest use-case for most people but would prevent most people from exploring other locations around the world.

- Using a forward geocoding service could be a compromise between these two, allowing users to enter a postal address that gets converted into lon/lat coordinates. However, there are issues with geocoding services including both geographic coverage and cost constraints that make this difficult to QA, maintain and implement at the level desired.

Managing Time while Prioritizing Project Features/Outcomes

- This project evolved over a period of a few weeks from a simple script that I could use myself into a fully-deployed shiny app that I could use to share with friends and family across the country. Throughout the process I balanced the end-goal of the project and the timeline I had with my desire to learn new skills (evolving from ggmap to leaflet, for example) and add useful features.
 - I'd like to use this presentation time to briefly cover the packages I relied on, and the **many** celestial calculations available in the swephR library beyond eclipse predictions.

Timezone Preference(s)

- I live about 15 minutes from Duke University so any presentation I would prefer in-person if possible. Otherwise America/New_York <-> America/Los_Angeles

Duration

- Open to any of the following:
 - Talk (15-20 minutes) - (in-person)
 - Lightning Talk (5 minutes) - (in-person)
 - Poster / visual presentation

Language

- English is the only language I speak.

Intended Audience

- Eclipse seekers
- Beginners and hobbyists looking for:
 - project idea inspiration,
 - a better understanding on how to approach the project life cycle, and
 - managing project creep

Pedagogical Method of Teaching

- For this I would approach in a lecture-style while allowing for interaction. I'm a seasoned public speaker in front of local elected and appointed officials and in engaging public meetings.
- Rather than having a technical presentation that gets into the weeds with code and deployment steps I'll approach the project evolution part of the presentation with the philosophy that it's more important to know what to do than how to do it. Writing out an outline in plain English helped me conceptualize the workflow for this project quicker and more accurately than trying to code-on-the-go.

Maximum Number of Attendees

- I would imagine the number of attendees interested would be relatively low due to the field intersections - astronomy x geospatial x public_communication. But if you are asking how many people I would be comfortable presenting in front of, there is no lower or upper bounds.

Motivation to Teach this Tutorial at useR!

- I enjoy making things in R (mostly as a hobby right now), I made a thing that's pretty neat and would like to share my experience in the hope that there are others who could benefit from either technical or general information shared, or maybe just be entertained by attending a talk on celestial calculations.
- I became VERY MOTIVATED when I saw that useR! 2025 was at Duke University in Durham as I live only a few miles away. It just seemed like the opportunity was right.

Supplemental Material Links

- Shiny app: “Future Solar and Lunar Eclipses Visible from Your Current Location” link
- Additional shiny app I developed specifically for the April 2024 eclipse day-of planning: “Total Eclipse of April 8, 2024 - What to Expect from Your Location” link

Tutorial materials N/A

Web Pages

https://tim-bender.shinyapps.io/shiny_all_eclipses/ https://github.com/benda18/eclipse/blob/main/shiny_all_eclipsesV2/app.R

Prerequisites / requirements

- None

License, Material sharing, recording consent

- None

Biography

Tim Bender, (title)

- Bachelor of Urban Planning from the University of Cincinnati.
- ~15 years local government experience as an urban planner and transit planner. Tim was part of a team that helped deploy Google Transit for his transit agency in Kentucky in 2008, being among the first 50-ish agencies worldwide to go live.

- Journey with R began with a desire to log transit vehicle real-time location data from an api for analysis but I had no programming experience or knowledge of how to approach the problem. I wouldn't successfully solve this problem until after about 5 years of self-guided learning.
- LinkedIn
- github