Analysis of human visual experience data

Authors

Johannes Zauner^[1, 2] [0000-0003-2171-4566]

Aaron Nicholls [3] [0009-0001-6683-6826]

Lisa A. Ostrin [4] [0000-0001-8675-1478]

Manuel Spitschan[1,2,5] [0000-0002-8572-9268]

^ corresponding author

Affiliations

1 Technical University of Munich, TUM School of Medicine and Health, Department Health and Sport Sciences, Chronobiology & Health, Munich, Germany

2 Translational Sensory & Circadian Neuroscience, Max Planck Institute for Biological Cybernetics, Tübingen, Germany

3 Reality Labs Research, USA

4 University of Houston College of Optometry, Houston, Texas, USA

5 TUM Institute for Advanced Study (TUM-IAS), Technical University of Munich, Garching, Germany

Abstract

Exposure to the optical environment — often referred to as *visual experience* — profoundly influences human physiology and behavior across multiple time scales. In controlled laboratory settings, stimuli can be held constant or manipulated parametrically. However, such exposures rarely replicate real-world conditions, which are inherently complex and dynamic, generating high-dimensional datasets that demand rigorous and flexible analysis strategies. This tutorial presents an analysis pipeline for visual experience datasets, with a focus on reproducible workflows for human chronobiology and myopia research. Light exposure and its retinal encoding affect human physiology and behavior across multiple time scales. Here we provide step-by-step instructions for importing, visualizing, and processing viewing distance and light exposure data using the open-source R package LightLogR. This includes time-series analyses for working distance, biologically relevant light metrics, and spectral characteristics. By leveraging a modular approach, the tutorial supports researchers in building flexible and robust pipelines that accommodate diverse experimental paradigms and measurement systems.

Keywords: wearable, light logging, viewing-distance, near-work, visual experience, metrics, circadian, myopia, risk factors, time series, spectral analysis, spatial analysis, open-source, reproducibility

Introduction

Exposure to the optical environment — often referred to as *visual experience* — profoundly influences human physiology and behavior across multiple time scales. Two notable examples, from distinct research domains, can be understood through a common retinally-referenced framework.

The first example relates to the non-visual effects of light on human circadian and neuroendocrine physiology. The light–dark cycle entrains the circadian clock, and light exposure at night suppresses melatonin production (Brown et al. 2022; Blume, Garbazza, and Spitschan 2019). The second example concerns the influence of visual experience on ocular development, particularly myopia. Time spent outdoors — which features distinct optical environments — has been consistently associated with protective effects on ocular growth and health outcomes (Dahlmann-Noor et al. 2025).

In controlled laboratory settings, light exposure can be held constant or manipulated parametrically. However, such exposures rarely replicate real-world conditions, which are inherently complex and dynamic. As people move in and between spaces (indoors and outdoors) and move their body, head, and eyes, exposure to the optical environment varies significantly (Webler et al. 2019) and is modulated by behavior (Biller, Balakrishnan, and Spitschan 2024). Wearable devices for measuring light exposure have thus emerged as vital tools to capture the richness of ecological visual experience. These tools generate high-dimensional datasets that demand rigorous and flexible analysis strategies.

Starting in the 1980s (Okudaira, Kripke, and Webster 1983), technology to measure optical exposure has matured, with miniaturized illuminance sensors now (in 2025) very common in consumer wearables. In research, several devices are available that differ in functionality, ranging from small pins measuring ambient illuminance (Mohamed et al. 2021) to head-mounted multi-modal devices capturing nearly all relevant aspects of visual experience (Gibaldi et al. 2024). Increased capabilities in wearables bring complex, dense datasets. These go hand-in-hand with a proliferation of metrics, as highlighted by recent review papers in both circadian and myopia research.

At present, the analysis processes to derive metrics are often implemented on a per-laboratory or even per-researcher basis. This fragmentation is a potential source of errors and inconsistencies between studies, consumes considerable researcher time (Hartmeyer, Webler, and Andersen 2022), and these bespoke processes and formats hinder harmonization or meta-analysis across multiple studies. Too often, more time is spent preparing data than gaining insights through rigorous statistical analysis. These preparation tasks are best handled, or at least facilitated, by standardized, transparent, community-based analysis pipelines (Zauner, Udovicic, and Spitschan 2024).

In circadian research, the R package LightLogR was developed to address this need (Zauner, Hartmeyer, and Spitschan 2025). LightLogR is an open-source, MIT-licensed, community-driven package specifically designed for data from wearable light loggers and optical radiation dosimeters. It contains functions to calculate over sixty different metrics used in the field (Hartmeyer and Andersen 2023). In the current version (v0.9.2 Sunrise), the package also handles modalities beyond illuminance, such as viewing distance and light spectra—capabilities highly relevant for myopia research (Hönekopp and Weigelt 2023).

In this article, we demonstrate that LightLogR's analysis pipelines and metric functions apply broadly across the field of visual experience research, not just to circadian rhythms and chronobiology. Our approach is modular and extensible, allowing researchers to adapt it to a variety of devices and research questions. Emphasis is placed on clarity, transparency, and reproducibility, aligning with best practices in scientific computing and open science. We use example data from two devices to showcase the LightLogR workflow with metrics relevant to myopia research, covering working distance, (day)light exposure, and spectral analysis. Furthermore, the **Supplementary material S1** contains vital steps for importing, cleaning, and preprocessing the data. The supplement further dives into the significance of these steps on the outcomes, such as how irregularly spaced data is handled or "valid days" are determined. Readers are encouraged to recreate the analysis using the provided code. All necessary data and code are openly available in the GitHub repository.

Method and Materials

Software

This tutorial was built with Quarto (www.quarto.org), an open-source scientific and technical publishing system that integrates text, code, and code output into a single document. The source code to reproduce all results is included and accessible via the Quarto document's code tool menu on the accompanying website (https://tscnlab.github.io/ZaunerEtAl_JVis_2025/). All analyses were conducted in R (version 4.4.3, "Trophy Case") using LightLogR (version 0.9.2 "Sunrise"). We also used the tidyverse suite (version 2.0.0) for data manipulation (which LightLogR follows in its design), and the gt package (version 1.0.0) for generating summary tables. A comprehensive overview of the R computing environment is provided in the session info on the webpage.

Code in this tutorial is shown in grey boxes, and is taken verbatim from R scripts. Code is highlighted for easier distinction of functions (ending with brackets ()), argument names (ending with equal sign =), strings (in quotes ""), and comments (preceding hashtag #). Portions of code that are not central to a given section are greyed out.

Metric selection and definitions

In March 2025, two workshops with myopia researchers — initiated by the Research Data Alliance (RDA) Working Group on Optical Radiation Exposure and Visual Experience Data — focused on current needs and future opportunities in data analysis, including the development and standardization of metrics. Based on expert input from these workshops, the authors of this tutorial compiled a list of visual experience metrics, shown in **Table 1**, with definitions of terms in **Table 2**. These include many currently used metrics and definitions (Wen et al. 2020, 2019; Bhandari and Ostrin 2020; Williams et al. 2019), as well as new metrics enabled by spectrally-resolved measurements.

No.	Name	Implementation[1]
Distance		
1	Total wear time daily	durations()*

2	Duration of Near work, Intermediate Work, Near + Intermediate Work, or per each *Distance range* (10cm steps)	filter for distance range + durations()* (for single ranges) or grouping by distance range + durations()* (for all ranges)
3	Frequency of Continuous near work	extract_clusters()* + summarize_numeric()*
4	Frequency, duration, and distances of *Near Work episodes*	extract_clusters()* + extract_metric()* + summarize_numeric()*
5	Frequency and duration of *Visual breaks*	extract_clusters()* + filter
Light		
6	Light exposure (in lux)	summarize_numeric()*
7	Duration per *Outdoor range*	grouping by Outdoor range + durations()*
8	The number of times light level changes from indoor (<1000 lx) to outdoor (>1000 lx)	extract_states()* + summarize_numeric()*
9	Longest period above 1000 lx	period_above_threshold()*
Spectrum		
10	Ratio of short vs. long wavelength light	spectral_integration()* + summarize_numeric()*
11	Short-wavelength light at certain times of day	spectral_integration()* + filter_Time()* (for defined times) or cut_Datetime()* (for regular time intervals) or add_photoperiod()* (for solar times) + grouping by time state + summarize_numeric()*

[1] Functions from `LightLogR` are presented with an *. Non-asterisk functions refer to tidyverse functions.

Table 1: Overview of metrics. In all cases, the averages for weekday, weekend, and the mean daily value are calculated. Definitions are shown in ***Table 2****.*

Table 2 provides definitions for the terms used in **Table 1**. Note that specific definitions may vary depending on the research question or device capabilities.

Term / metric	**Description / pseudo formula**
Total wear time	$\sum(t) * dt$, where t: valid observations
Mean daily	$\frac{5 * \text{weekday} + 2 * \text{weekend}}{7}$
Near work	working distance, $[10,60)cm$
Intermediate Work	working distance, $[60,100)cm$
Total work	working distance, $[10,120)cm$
Distance range	working distance, $[10,20)cm$, Extremely near $[20,30)cm$, Very near $[30,40)cm$, Fairly near $[40,50)cm$, Near $[50,60)cm$, Moderately near $[60,70)cm$, Near intermediate $[70,80)cm$, Intermediate $[80,90)cm$, Moderately intermediate $[90,100)cm$, Far intermediate
Continuous near work	$T_t = \sum(t) * dt$, where working distance, $[20,60)cm$, $T_t \geq 30minutes$, $T_{interruptions} \leq 1minute$
Near work episodes	$T_t = \sum(t) * dt$, where working distance, $[20,60)cm$, $T_{\text{interruptions}} \leq 20seconds$
Ratio of daily near work	$\frac{T_{\text{near work}}}{T_{\text{total wear}}}$
Visual break	$T_t = \sum(t) * dt$, where working distance $\geq 100cm$, $T_t \geq 20$ seconds, $T_{t-1\ (\text{working distance}<100\,cm)} \leq 20$ minutes
Outdoor range	illuminance, $[1000,2000)lx$, Outdoor bright $[2000,3000)lx$, Outdoor very bright $[3000,\infty)lx$, Outdoor extremely bright
Light exposure[1]	$Illuminance$
Spectral bands	spectral irradiance, $[400,500]nm$, short wavelength light $[600,700]nm$, long wavelength light

Ratio of short vs. long wavelength light	$\frac{E_{e,\text{short wavelength}}}{E_{e,\text{long wavelength}}}$

[1] This deviates from the common definition of luminous exposure, which is the sum of illuminance measurements scaled to hourly observation intervals

Table 2: Definitions of terms and metrics

Devices

Data from two wearable devices are used in this analysis:

- `Clouclip`: A wearable device that measures viewing distance and ambient light [Glasson Technology Co., Ltd, Hangzhou, China; Wen et al. (2021); Wen et al. (2020)]. The Clouclip provides a simple data output with only Distance (working distance, in centimeters) and Illuminance (ambient light, in lux). Data in our example were recorded at 5-second intervals. Approximately one week of data (~120,960 observations) is about 1.6 MB in size.
- `Visual Environment Evaluation Tool` (VEET): A head-mounted multi-modal device that logs multiple data streams [Meta Platforms, Inc., Menlo Park, CA, USA; Sah, Narra, and Ostrin (2025); Sullivan et al. (2024)]. The VEET dataset used here contains simultaneous measurements of distance (via a time-of-flight sensor), ambient light (illuminance), activity (accelerometer & gyroscope), and spectral irradiance (multi-channel light sensor). Data were recorded at 2-second intervals, yielding a very dense dataset (~270 MB per week).

Data processing summary

The Results section uses imported and pre-processed data from the two devices to calculate metrics. Supplement 1 contains the annotated code and description for the steps involved. The following summarizes the steps involved:

Data import: We imported raw data from the `Clouclip` and `VEET` devices using LightLogR's built-in import functions, which automatically handle device-specific formats and idiosyncrasies. The `Clouclip` export file (provided as a tab-delimited text file) contains timestamped records of distance (cm) and illuminance (lux). LightLogR's `import$Clouclip` function reads this file, after specifying the device's recording timezone, and converts device-specific sentinel codes into proper missing

values. For instance, the Clouclip uses special numeric codes to indicate when it is in “sleep mode” or when a reading is out of the sensor’s range, rather than recording a normal value. LightLogR identifies `-1` (for both distance and illuminance) as indicating the device’s sleep mode and `204` (for distance) as indicating the object was beyond the measurable range, replacing these with `NA` and logging their status in separate columns. The import routine also provides an initial summary of the dataset, including start and end times and any irregular sampling intervals or gaps.

For the `VEET` device, data were provided as CSV logs (zipped on Github, due to size). We focused on the ambient light sensor modality first. Using `import$VEET(..., modality = "ALS")`, we extracted the illuminance (`Lux`) data stream and its timestamps. The raw VEET data similarly can contain irregular intervals or missing periods (e.g., if the device stopped recording or was reset); the import summary flags these issues.

Irregular intervals and gaps: Both datasets showed irregular timing and missing data, i.e., gaps. Irregular data means that some observations did not align with the nominal sampling interval (e.g., slight timing drift or pauses in recording). For the `Clouclip`‘s 5-second data, we detected irregular timestamps spanning all but the first and last day of the recording. Handling such irregularities is important because many downstream analyses assume a regular time series. We evaluated strategies to address this, including:

- Removing an initial portion of data if irregularities occur mainly during device start-up.
- Rounding all timestamps to the nearest regular interval (5 s in this case).
- Aggregating to a coarser time interval (with some loss of temporal resolution).

Based on the import summary and visual inspection of the time gaps, we chose to round the observation times to the nearest 5-second mark, as this addressed the minor offsets without significant data loss. After rounding timestamps, we added an explicit date column for convenient grouping by day.

We then generated a summary of missing data for each day. Implicit gaps (intervals where the device should have recorded data but did not) were converted into explicit missing entries using LightLogR’s gap-handling functions. We also removed days

that had very little data (in our `Clouclip` example, days with <1 hour of recordings were dropped) to focus on days with substantial wear time.

After these preprocessing steps, the `Clouclip` dataset had no irregular timestamps remaining and contained explicit markers for all periods of missing data (e.g., times when the device was off or not worn). The distance and illuminance values were now ready for metric calculations.

The `VEET` illuminance data underwent a similar cleaning procedure. To make the `VEET`'s 2-second illuminance data more comparable to the `Clouclip`'s and to reduce computational load, we aggregated the illuminance time series to 5-second intervals. We then inserted explicit missing entries for any gaps and removed days with more than one hour of missing illuminance data. After cleaning, six days of VEET illuminance data with good coverage remained for analysis (see Supplementary Material for details).

Finally, for spectral analysis, we imported the `VEET`'s spectral sensor modality, and, for the distance analysis, the time-of-flight modality. This required additional processing: the raw spectral data consists of counts from 10 wavelength-specific channels (approximately 415 nm through 940 nm, plus two broadband clear channels and a dark channel) along with a sensor gain setting. We aggregated the spectral data to 5-minute intervals to focus on broader trends and reduce data volume. Each channel's counts were normalized by the appropriate gain at each moment, and the two clear channels were averaged. Using a calibration matrix provided by the manufacturer (specific to the spectral sensor model), we reconstructed full spectral power distributions for each 5-minute interval. The end result is a list-column in the dataset where each entry is the estimated spectral irradiance across wavelengths for that time interval. (Detailed spectral preprocessing steps, including the calibration and normalization, are provided in the Supplement.) After spectral reconstruction, the dataset was ready for calculating example spectrum-based metrics.

Similarly, the time-of-flight modality contains 256 values per observation, encoding an 8x8 grid of distance and confidence measurements for up to two objects (8x8 grid, times two objects, times distance + confidence column for each object and grid point -> 256 values). These data were pivoted into a long format, where each row

contains the distance and confidence data for both objects for a given position in the grid and a given datetime. After pivoting and converting grid positions into a deviation angle from central view, the dataswert was ready to be used for distance analysis.

This tutorial will start by importing a `Clouclip` dataset and providing an overview of the data. The `Clouclip` export is considerably simpler compared to the `VEET` export, only containing `Distance` and `Illuminance` measurements. The `VEET` data will be used later to discuss pipelines with the richer dataset.

```
# load libraries
library(LightLogR)
library(tidyverse)
library(gt)

load("data/cleaned/data.RData")
coordinates <- c(29.75, -95.36) #coordinates for Houston, Texas
#coordinates are important to calculate and visualize photoperiods later
```

Results

Distance

We first examine metrics related to viewing distance, using the processed `Clouclip` dataset. Many distance-based metrics are computed for each day and then averaged over weekdays, weekends, or across all days. To facilitate this, we define a helper function that produces a table:

```
to_mean_daily <- function(data, prefix = "average_") {
  data |>
    ungroup(Date) |>        # ungroup “by days”
    mean_daily(prefix = prefix) |>   # calculate the averages per grouping
    # remove underscores in names
    rename_with(.fn = \(x) str_replace_all(x,"_"," ")) |>
    gt()               # format as a gt table for display
}
```

Total wear time daily

Total wear time daily refers to the amount of time the device was actively collecting distance data each day (i.e. the time the device was worn and operational). We compute this by summing all intervals where a valid distance measurement is present, ignoring periods where data are missing or the device was off. The results

are shown in Table 3. The preprocessed `Clouclip` dataset is encoded in the variable `dataCC`.

```
dataCC |>
  durations(Dis) |>  # calculate total duration of data per day
  to_mean_daily("Total wear ")
```

	Total wear duration
Mean daily	31448s (~8.74 hours)
Weekday	34460s (~9.57 hours)
Weekend	23918s (~6.64 hours)

Table 3: Total wear time per day (average across days)

Duration within distance ranges

Many myopia-relevant metrics concern the time spent at certain viewing distances (e.g., “near work” vs. intermediate or far distances). We calculate the duration of time spent in specific distance ranges. **Table 4** shows the average daily *duration of near work*, defined here as time spent viewing at distances between 10–60 cm (a commonly used definition for near-work distance). **Table 5** provides a more detailed breakdown across multiple distance bands.

Duration of near work

```
dataCC |>
  filter(Dis >= 10, Dis < 60) |>  # consider only distances [10, 60) cm
  durations(Dis) |>               # total duration in that range per day
  to_mean_daily("Near work ")
```

	Near work duration
Mean daily	22586s (~6.27 hours)
Weekday	26343s (~7.32 hours)
Weekend	13192s (~3.66 hours)

Table 4: Daily duration of near work (10–60 cm viewing distance)

Duration within distance ranges

First, we define a set of distance breakpoints and descriptive labels for each range:

```
# defining distance ranges (in cm)
dist_breaks <- c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100, Inf)
dist_labels <- c(
    "Extremely near",          # [10, 20)
    "Very near",               # [20, 30)
    "Fairly near",             # [30, 40)
```

```
    "Near",                     # [40, 50)
    "Moderately near",          # [50, 60)
    "Near intermediate",        # [60, 70)
    "Intermediate",             # [70, 80)
    "Moderately intermediate", # [80, 90)
    "Far intermediate",         # [90, 100)
    "Far"                       # [100, Inf)
  )
```

Now we cut the distance data into these ranges and compute the daily duration spent in each range:

```
dataCC |>
  mutate(# categorize distances:
    Dis_range = cut(Dis, breaks = dist_breaks, labels = dist_labels)
  ) |>
  drop_na(Dis_range) |>                # remove intervals with no data
  group_by(Dis_range, .add = TRUE) |>  # group by distance range
  durations(Dis) |>                    # duration per range per day
  # wide format (ranges as columns):
  pivot_wider(names_from = Dis_range, values_from = duration) |>
  to_mean_daily("") |>
  fmt_duration(# convert seconds to minutes:
    input_units = "seconds", output_units = "minutes"
  )
```

	Mean daily	Weekday	Weekend
Far	16m	20m	5m
Far intermediate	11m	14m	2m
Moderately intermediate	5m	6m	3m
Intermediate	4m	6m	1m
Near intermediate	7m	7m	8m
Moderately near	13m	16m	5m
Near	27m	36m	5m
Fairly near	46m	60m	12m
Very near	102m	128m	38m
Extremely near	169m	180m	141m

Table 5: Daily duration in each viewing distance range. For better readability, the table was transposed (compared to the code results)

To visualize these data, **Figure 1** illustrates the relative proportion of time spent in each distance range:

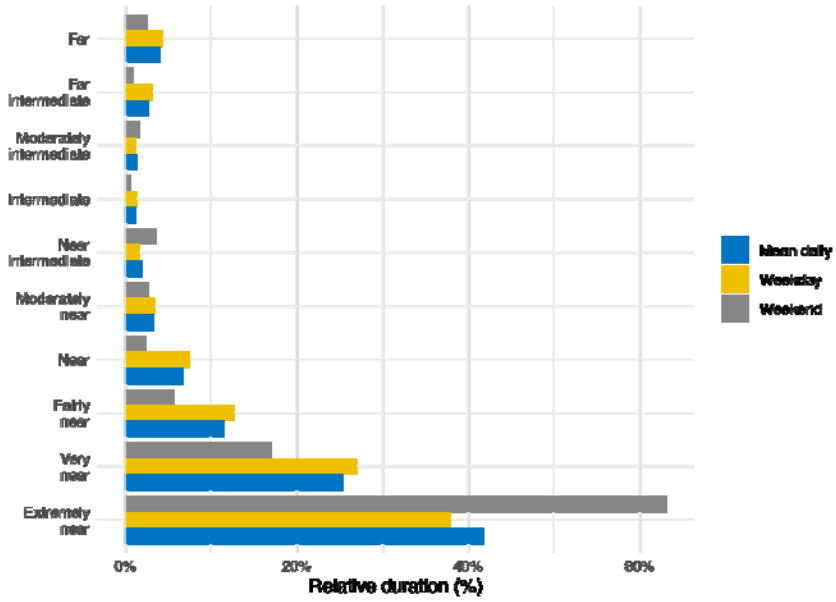


Figure 1: Percentage of total time spent in each viewing distance range for an average day (mean daily), average weekday, or weekend

Frequency of continuous near work

Continuous near-work is typically defined as sustained viewing within a near distance for some minimum duration, allowing only brief interruptions. We use LightLogR's `cluster` function to identify episodes of continuous near work. Here we define a near-work episode as viewing distance between 20 cm and 60 cm that lasts at least 30 minutes, with interruptions of up to 1 minute allowed (meaning short breaks ≤1 min do not end the episode). Using `extract_clusters()` with those parameters, we count how many such episodes occur per day.

Table 6 summarizes the average frequency of continuous near-work episodes per day, and **Figure 2** provides an example visualization of these episodes on the distance time series.

```
dataCC |>
  extract_clusters(# create a new table containing clusters
    Dis >= 20 & Dis < 60,            # condition: near-work distance
    cluster.duration = “30 mins”,    # minimum duration
    interruption.duration = “1 min”, # maximum gap allowed
    drop.empty.groups = FALSE        # list days with zero episodes
  ) |>
  summarize_numeric(remove = c(“start”, “end”, “epoch”, “duration”),
                    add.total.duration = FALSE) |> # count daily episodes
```

```
  mean_daily(prefix = “Frequency of “) |> # compute daily mean frequency
  gt() |> fmt_number()
```

	Frequency of episodes
Mean daily	0.86
Weekday	1.20
Weekend	0

Table 6: Frequency of continuous near-work episodes per day

```
dataCC |>
  add_clusters( # add a cluster column named “state” to the existing data
    Dis >= 20 & Dis < 60,
    cluster.duration = "30 mins",
    interruption.duration = "1 min"
  ) |>
  # Plotting the clusters:
  gg_day(y.axis = Dis, y.axis.label = "Distance (cm)", geom = "line",
    y.scale = "identity", y.axis.breaks = seq(0, 100, by = 20) |>
  gg_state(state, fill = "red") + # add state bands
  geom_hline(yintercept = c(20, 60), col = "red", linetype = "dashed") +
  coord_catesian(ylim = c(0, 100))
```

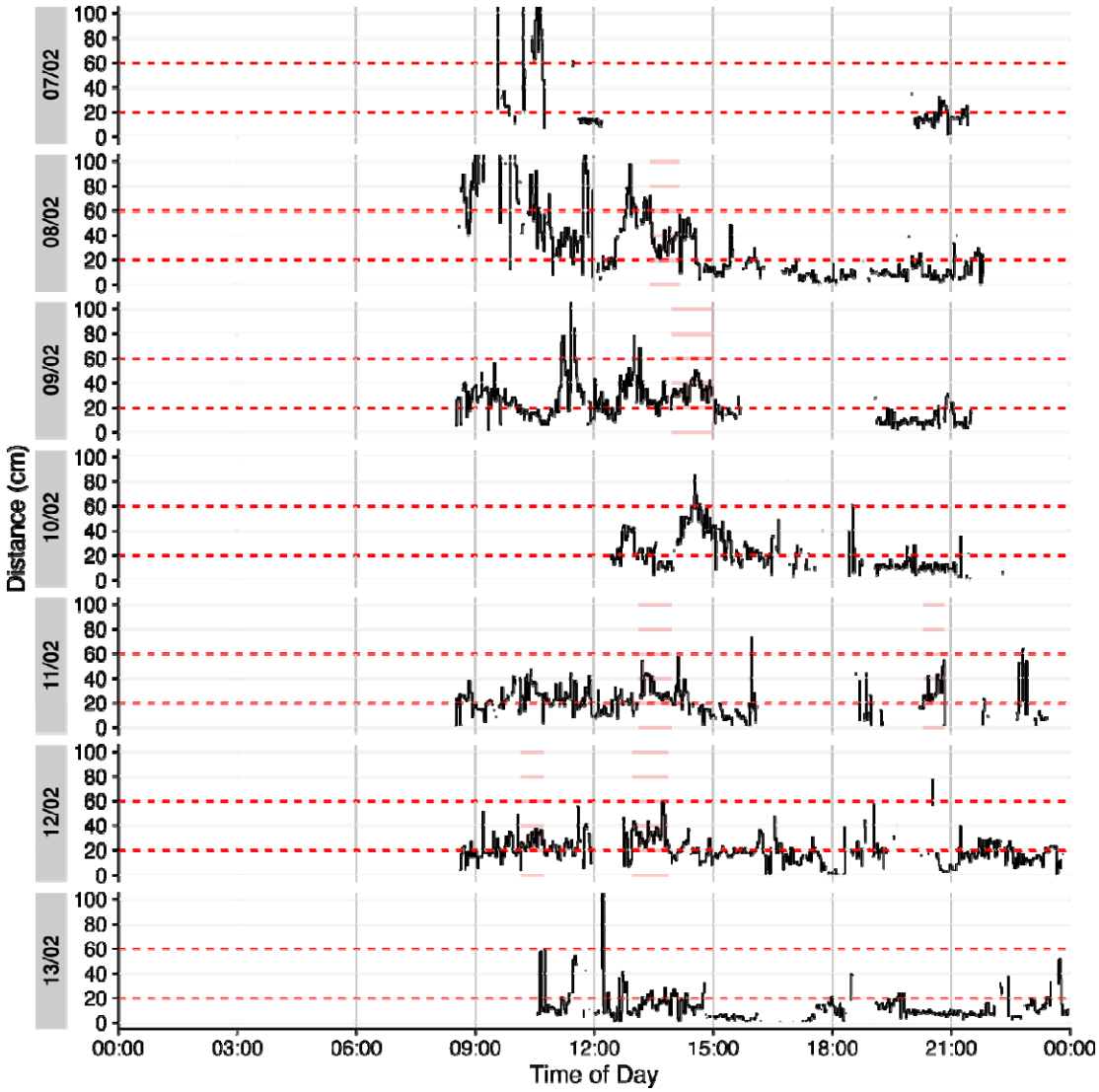


Figure 2: Example of continuous near-work episodes. Red shaded areas indicate periods of continuous near work (20–60 cm for ≥30 min, allowing ≤1 min interruptions). Black trace is viewing distance over time of day; red dashed lines mark the 20 cm and 60 cm boundaries.

Near-work episodes

Beyond frequency, we can characterize near-work episodes by their duration and typical viewing distance. This section extracts all near-work episodes (using a shorter minimum duration to capture more routine near-work bouts) and summarizes three aspects: (1) frequency (count of episodes per day), (2) average duration of

episodes, and (3) average distance during those episodes. These results are combined in **Table 7**.

```
dataCC |>
  extract_clusters(
    Dis >= 20 & Dis < 60,
    cluster.duration = "5 secs", # minimal duration is set to interval
    interruption.duration = "20 secs", # up to 20 seconds interruption
    drop.empty.groups = FALSE # list empty groups, too
  ) |>
  # calculate mean distance during each episode and add to clusters:
  extract_metric(dataCC, distance = mean(Dis, na.rm = TRUE)) |>
  summarize_numeric(remove = c("start", "end", "epoch"), prefix = "",
                    add.total.duration = FALSE) |>
  mean_daily(prefix = "") |>    # daily averages for each metric
  gt() |> fmt_number(c(distance, episodes), decimals = 0) |> #table
  cols_units(distance = "cm")
```

	duration	distance, cm	episodes
Mean daily	233s (~3.88 minutes)	32	57
Weekday	284s (~4.73 minutes)	32	64
Weekend	104s (~1.73 minutes)	32	40

Table 7: Near-work episodes: frequency, mean duration, and mean viewing distance

In the above, `extract_metric(..., distance = mean(Dis, ...))` computes the mean viewing distance during each episode, and the subsequent `summarize_numeric()` and `mean_daily()` steps derive daily averages of episode count, duration, and distance.

Visual breaks

Visual breaks are a little different, compared to the previous metrics. The difference is that, in this case, the minimum break and the previous (non-break) episode is important. This leads to a two-step process, where we first extract instances of `Distance` above 100 cm for at least 20 seconds, before we filter for a previous duration of at maximum 20 minutes. **Table 8** provides the daily frequency of visual breaks.

```
dataCC |>
  extract_clusters(Dis >= 100, #define the condition, greater 100 cm away
                   cluster.duration = "20 secs", # minimum duration
                   return.only.clusters = FALSE, #return also non-clusters
                   drop.empty.groups = FALSE #list even days w/o cluster
```

```
                    ) |>
 # filter clusters with previous episode lengths of at maximum 20 minutes:
  filter((start - lag(end) <= duration("20 mins")), is.cluster) |>
  summarize_numeric(
    remove = c("start", "end", "epoch", "is.cluster", "duration"),
    prefix = "",
    add.total.duration = FALSE) |> #count the number of episodes
  mean_daily(prefix = "Daily ") |> #daily means
  gt() |> fmt_number(decimals = 1) #table
```

	Daily episodes
Mean daily	5.9
Weekday	6.2
Weekend	5.0

Table 8: Frequency of visual breaks (n per day)

```
dataCC |>
  extract_clusters(Dis >= 100, #define the condition, greater 100 cm away
                   cluster.duration = "20 secs", # minimum duration
                   return.only.clusters = FALSE, #return also non-clusters
                   drop.empty.groups = FALSE #list even days w/o cluster
                   ) |>
  # return only clusters with previous episode lengths of maximum 20
minutes:
  filter((start - lag(end) <= duration("20 mins")), is.cluster) %>%
  add_states(dataCC, ., ) |> # add these clusters to the dataset
  gg_day(y.axis = Dis, y.axis.label = "Distance (cm)", geom = "line") |>
  gg_photoperiod(coordinates) + # add photoperiod information to the plot
  geom_point(data = \(x) filter(x, is.cluster), col = "red")
```

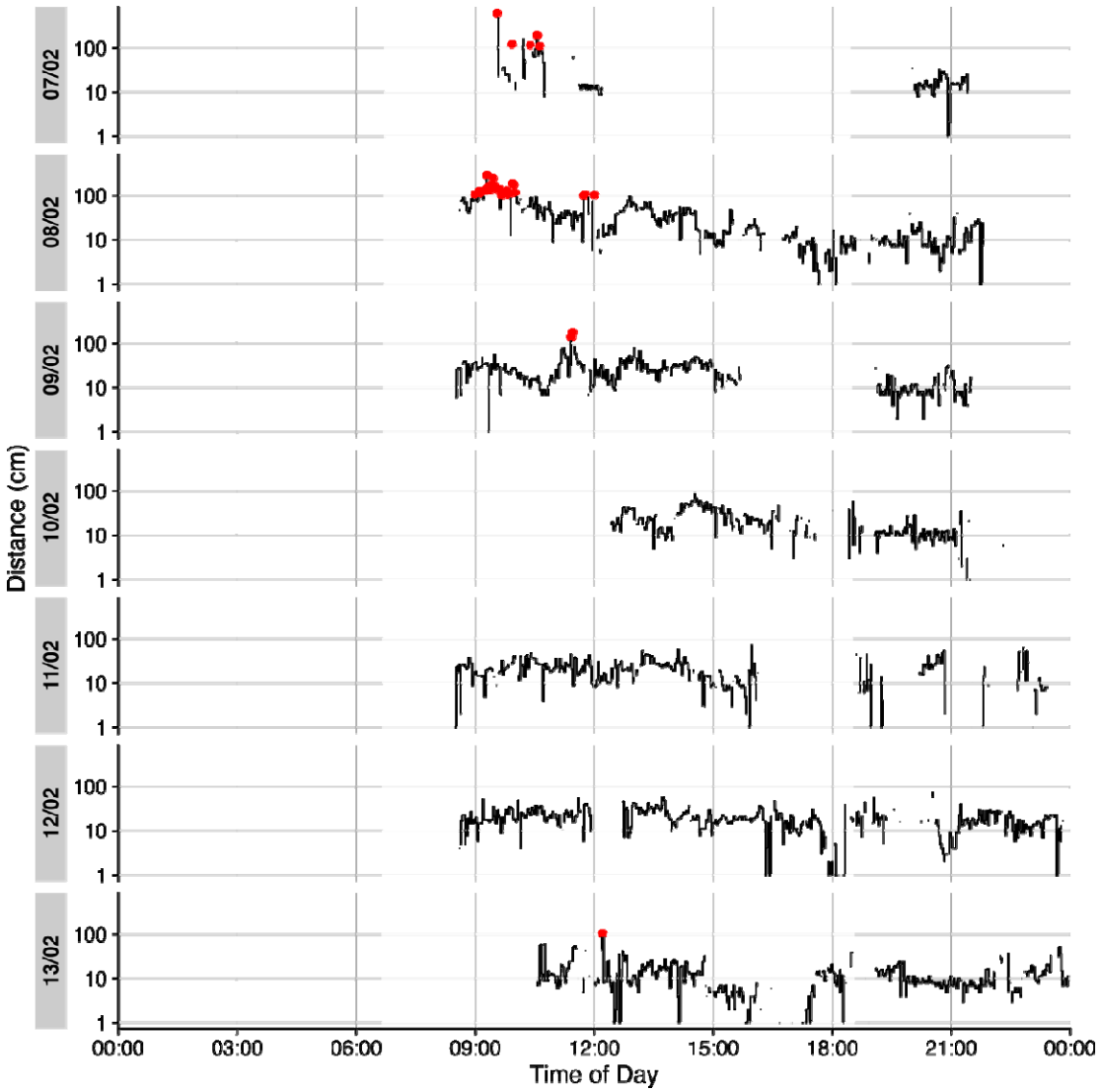


Figure 3: Plot of visual breaks (red dots). Black traces show distance measurement data. Grey shaded areas show nighttime between civil dusk and civil dawn (photoperiod)

Distance with spatial distribution

The `Clouclip` device outputs a singular measure for distance, while the visual environment in natural conditions contains many distances, depending on the solid angle and direction of the measurement. A device like the `VEET` increases the spatial resolution of measurements, allowing for more in-depth analyses of the size and position of an object within the field of view. In the case of the `VEET`, data are collected from an 8x8 measurement grid, spanning 52° vertically and 41°

horizontally. Here are exemplary observations from six different days at the same time:

```
# create helper function to allow choosing an observation:
slicer <- function(x){seq(min((x-1)*64+1), max(x*64), by = 1)}
#set visualization parameters:
extras <- list(
  geom_tile(), #colored tiles
  scale_fill_viridis_c(direction = -1, limits = c(0, 200), #color palette
                       oob = scales::oob_squish_any), #include infinity
  scale_color_manual(values = c("black", "white")), #colors for text
  theme_minimal(), #set minimal plot theme
  guides(colour = "none"), #remove scale for text
  geom_text( # add text labels for values
    aes(label = (dist1/10) |> round(0), colour = dist1>1000), size = 2.5),
  coord_fixed(), #keep x&y dimensions of plots consistent with scale
  labs(x = "X position (°)", y = "Y position (°)", #set labels
       fill = "Distance (cm)", alpha = "Confidence (0-255)")) #set labels

dataVEET3 |>
  slice(slicer(9530)) |>  #choose a particular observation with helper
  mutate(dist1 = ifelse(dist1 == 0, Inf, dist1)) |> #replace 0 -> infinity
  filter(conf1 >= 0.1 | dist1 == Inf) |> #remove data < 10% confidence
  ggplot(aes(x=x.pos, y=y.pos, fill = dist1/10))+ extras + #plot the data
  facet_wrap(~Datetime) #show one plot per day
```

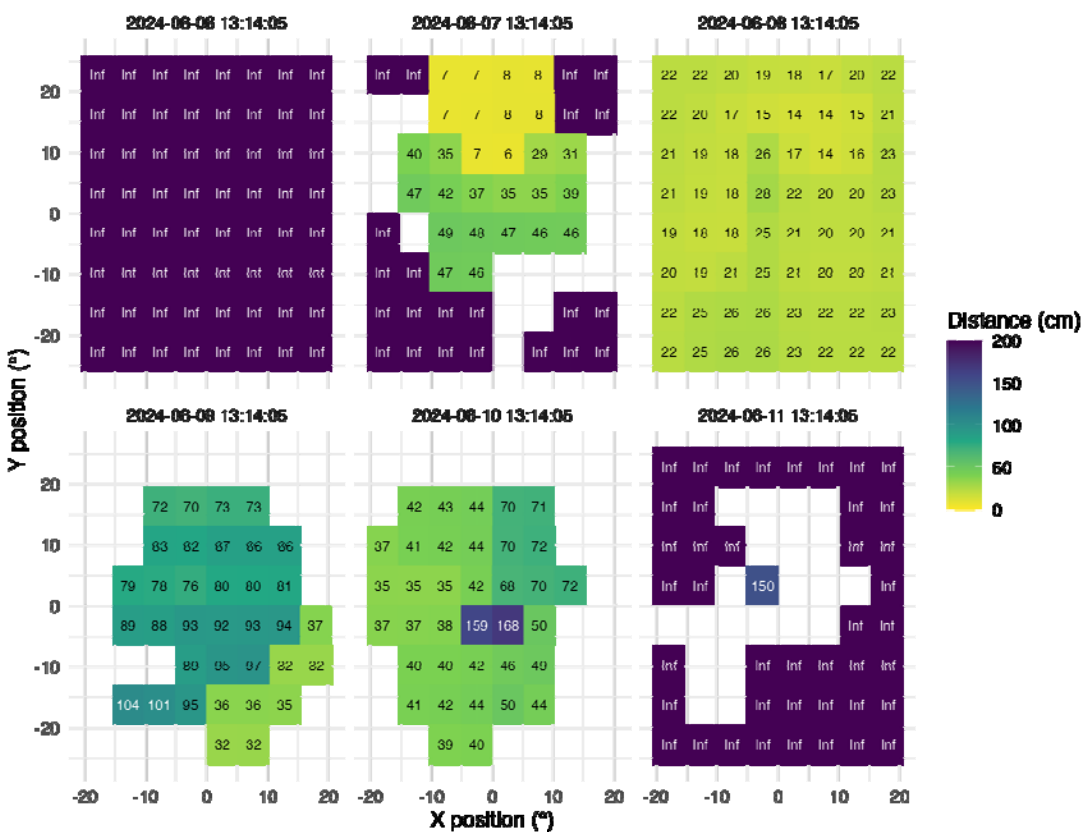


Figure 4: Example observations of the measurement grid at 1:14 p.m. for each measurement day. Text values show distance in cm. Empty grid points show values with low confidence. Zero-distance values were replaced with infinite distance and plotted despite low confidence.

To use these distance data in the framework shown above for the `Clouclip` device, a sensible method to condense the data has to be applied. This method has to be chosen based on theoretical assumptions about what a relevant distance within the field of view is. Possible methods include:

- average across all (high confidence) distance values within the grid
- closest (high confidence) distance within the grid
- (high confidence) values at or around a given grid position, e.g., ±10 degrees around the central view (0°)

Many more options are available based on the richer dataset, e.g., condensation rules based on the number of points in the grid with a given condition, or the variation within the grid.

We will demonstrate these three exemplary methods for a single day (`2024-06-10`), all leading to a data structure akin to the `Clouclip`, i.e., to be used for further calculation of visual experience metrics.

```
dataVEET3_part <- #filter one day:
dataVEET3 |>
  filter_Date(start = "2024-06-10", length = "1 day")

dataVEET3_condensed <- #condense the data:
dataVEET3_part |>
  group_by(Datetime, .add = TRUE) |> #group by every observation
  filter(conf1 >= 0.1) |> #remove data with low confidence
  summarize(
    distance_mean = mean(dist1), #average across all distance values,
    distance_min = min(dist1), #closest across all distance values,
    distance_central = #central distance:
      mean(dist1[between(x.pos, -10,10) & between(y.pos, -10,10)]),
    n = n(), #number of (valid) grid points
    .groups = "drop_last"
  )

dataVEET3_condensed |> #plot the data:
  aggregate_Datetime(
    "15 mins", #create 15 minute data
    numeric.handler = \(x) mean(x, na.rm = TRUE) #ignor NA datapoints
  ) |>
  remove_partial_data(by.date = TRUE) |> #remove midnight data point
```

```
  pivot_longer(contains("distance"), #put all methods into a long table
               names_to = c(".value", "method"),
               names_pattern = "(distance)_(mean|min|max|central)"
  ) |>
  gg_day(y.axis = distance/10, #plot timeline on a cm scale
         aes_col = method,
         geom = "line",
         group = method,
         linewidth = 1,
         alpha = 0.75,
         y.scale = "identity",
         y.axis.breaks = seq(0,150, by = 20),
         y.axis.label = "Distance (cm)"
         )
```

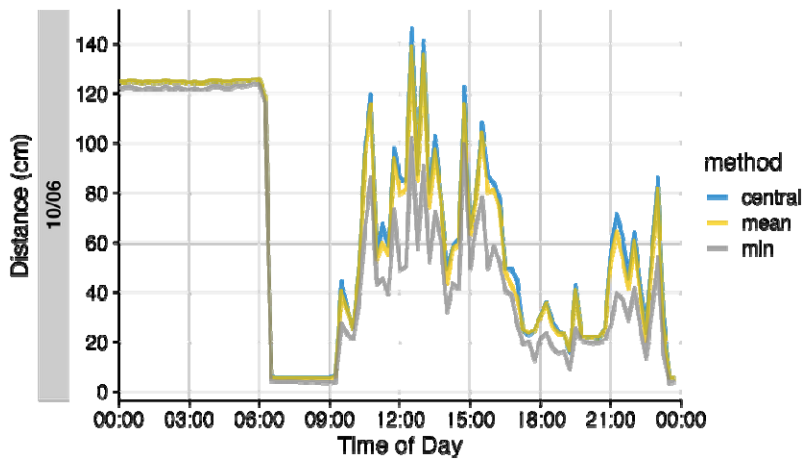


Figure 5: Comparison of condensation methods for spatial grid of distance measurements. The lines represent an average across all data points (yellow), the minimum distance (grey), or the central 10° (blue). Data points with confidence less than 10% were removed prior to calculation.

As can be seen in **Figure 5**, while the overall pattern is similar regardless of the used method, there are notable differences between the methods, which will consequently affect downstream analyses. Most importantly, the process of condensation has to be well documented and reproducible, as shown above. Any of these data could now be used to calculate the `frequency of continuous near work`, `visual breaks`, or `near-work episodes` as described above.

Light

The `Clouclip` illuminance data in our example are extremely low (the device was mostly used in dim conditions), which would make certain light exposure summaries

trivial or not meaningful. To better illustrate light exposure metrics, we turn to the exemplary `VEET` device's illuminance data, which capture a broader range of lighting conditions. We briefly examine its distribution.

Illuminance distribution: The illuminance values from the `Clouclip` were almost always near zero (see **Figure 6**), while the `VEET` data include outdoor exposures up to several thousand lux. The contrast is evident from comparing histograms of the two datasets' lux values (`Clouclip` vs. `VEET`). The `VEET` illuminance histogram (see **Figure 7**) shows a heavily skewed distribution with a spike at zero (indicating many intervals of complete darkness or the sensor being covered) and a long tail extending to very high lux values. Such zero-inflated and skewed data are common in wearable light measurements (Zauner, Guidolin, and Spitschan, 2025).

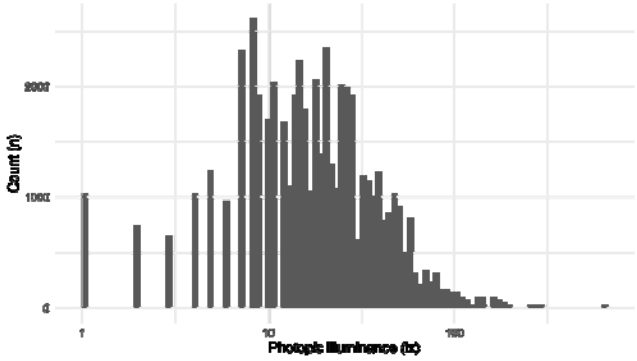


Figure 6: Histogram of illuminance values from the Clouclip dataset (5-second data). The values are very low and are typical of indoor conditions.

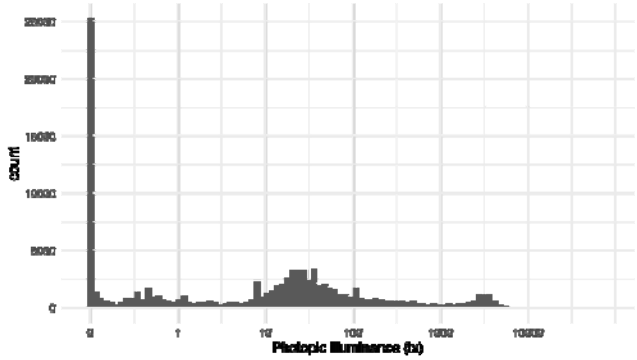


Figure 7: Histogram of illuminance values from the VEET dataset (aggregated to 5 s). Note the logarithmic x-axis: the distribution is highly skewed with many low values (including zeros) and a long tail of high lux readings.

After confirming that the `VEET` data cover a broad dynamic range of lighting, we proceed with calculating light exposure metrics. (The `VEET` data had been cleaned for gaps and irregularities as described earlier; see Supplement 1 for the summary table.)

Average light exposure

A basic metric is the average illuminance over the day. **Table 9** shows the mean illuminance (in lux) for weekdays, weekends, and the overall daily mean, calculated directly from the raw lux values.

```
dataVEET |>
  select(Id, Date, Datetime, Lux) |>
  # create averages for all numeric columns:
  summarize_numeric(prefix = "mean ", remove = c("Datetime")) |>
  to_mean_daily() |>      # compute mean for weekday, weekend, all days
  fmt_number(decimals = 1) |>
  cols_hide(`average episodes`) |>  # hide an irrelevant column
  cols_label(`average mean Lux` = "Mean photopic illuminance (lx)")
```

	Mean photopic illuminance (lx)
Mean daily	304.1
Weekday	357.8
Weekend	169.8

Table 9: Mean light exposure (illuminance) per day

However, because illuminance data tend to be extremely skewed and contain many zero values (periods of darkness), the arithmetic mean can be misleading. A common approach is to apply a logarithmic transform to illuminance before averaging, which down-weighs extreme values and accounts for the multiplicative nature of light intensity effects. LightLogR provides helper functions `log_zero_inflated()` and its inverse `exp_zero_inflated()` to handle log-transformation when zeros are present (by adding a small offset before log, and back-transforming after averaging). Using this approach, we recompute the daily mean illuminance. The results in **Table 10** show that the log-transformed mean (back-transformed to lux) is much lower, reflecting the fact that for much of the time

illuminance was near zero. This transformed mean is often more representative of typical exposure for skewed data.

```
dataVEET |>
  select(Id, Date, Datetime, Lux) |>
  mutate(Lux = Lux |> log_zero_inflated()) |> # log-transform with zeros
  summarize_numeric(prefix = "mean ", remove = c("Datetime")) |>
  mean_daily(prefix = "") |> # get daily mean of log-lux
  mutate(`mean Lux` = `mean Lux` |> exp_zero_inflated()) |> #back to lux
  gt() |> fmt_number(decimals = 1) |> cols_hide(episodes) |> #table
  cols_label(`mean Lux` = "Mean photopic illuminance (lx)")
```

	Mean photopic illuminance (lx)
Mean daily	6.3
Weekday	7.9
Weekend	3.5

Table 10: Mean light exposure per day (after logarithmic transformation to account for zero inflation and skewness)

Duration in high-light (outdoor) conditions

Another important metric is the amount of time spent under bright light, often used as a proxy for outdoor exposure. We define thresholds corresponding to outdoor light levels (e.g., 1000 lx and above). Here, we categorize each 5-second interval of illuminance into bands: Outdoor bright (≥1000 lx), Outdoor very bright (≥2000 lx), and Outdoor extremely bright (≥3000 lx). We then sum the duration in each category per day. We first create a categorical variable for illuminance range:

```
# Define outdoor illuminance thresholds (in lux)
out_breaks <- c(1e3, 2e3, 3e3, Inf)
out_labels <- c(
    "Outdoor bright",          # [1000, 2000) lx
    "Outdoor very bright",     # [2000, 3000) lx
    "Outdoor extremely bright" # [3000, ∞) lx
  )
# Divide data by thresholds (in lux)
dataVEET <- dataVEET |>
  mutate(Lux_range = cut(Lux, breaks = out_breaks, labels = out_labels))
```

Now we compute the mean daily duration spent in each of these outdoor light ranges (**Table 11**):

```r
dataVEET |>
  drop_na(Lux_range) |>
  group_by(Lux_range, .add = TRUE) |>
  durations(Lux) |> # total duration per range per day
  pivot_wider(names_from = Lux_range, values_from = duration) |>
  to_mean_daily("") |>
  fmt_duration(input_units = "seconds", output_units = "minutes")
```

Outdoor	bright	very bright	extremely bright
Mean daily	24m	32m	55m
Weekday	29m	41m	65m
Weekend	10m	10m	30m

Table 11: Average daily duration in outdoor-equivalent light conditions

It is also informative to visualize when these bright-light conditions occurred. **Figure 8** shows a timeline plot with periods of outdoor-level illuminance highlighted in color. In this example, violet denotes ≥1000 lx, green ≥2000 lx, and yellow ≥3000 lx. Grey shading indicates nighttime (from civil dusk to dawn) for context.

```r
dataVEET |>
  gg_day(y.axis = Lux, y.axis.label = "Photopic illuminance (lx)",
         geom = "line", jco_color = FALSE) |>
  gg_state(Lux_range, aes_fill = Lux_range, alpha = 0.75) |>
  gg_photoperiod(coordinates) +
  scale_fill_viridis_d() +
  labs(fill = "Illuminance range") +
  theme(legend.position = "bottom")
```

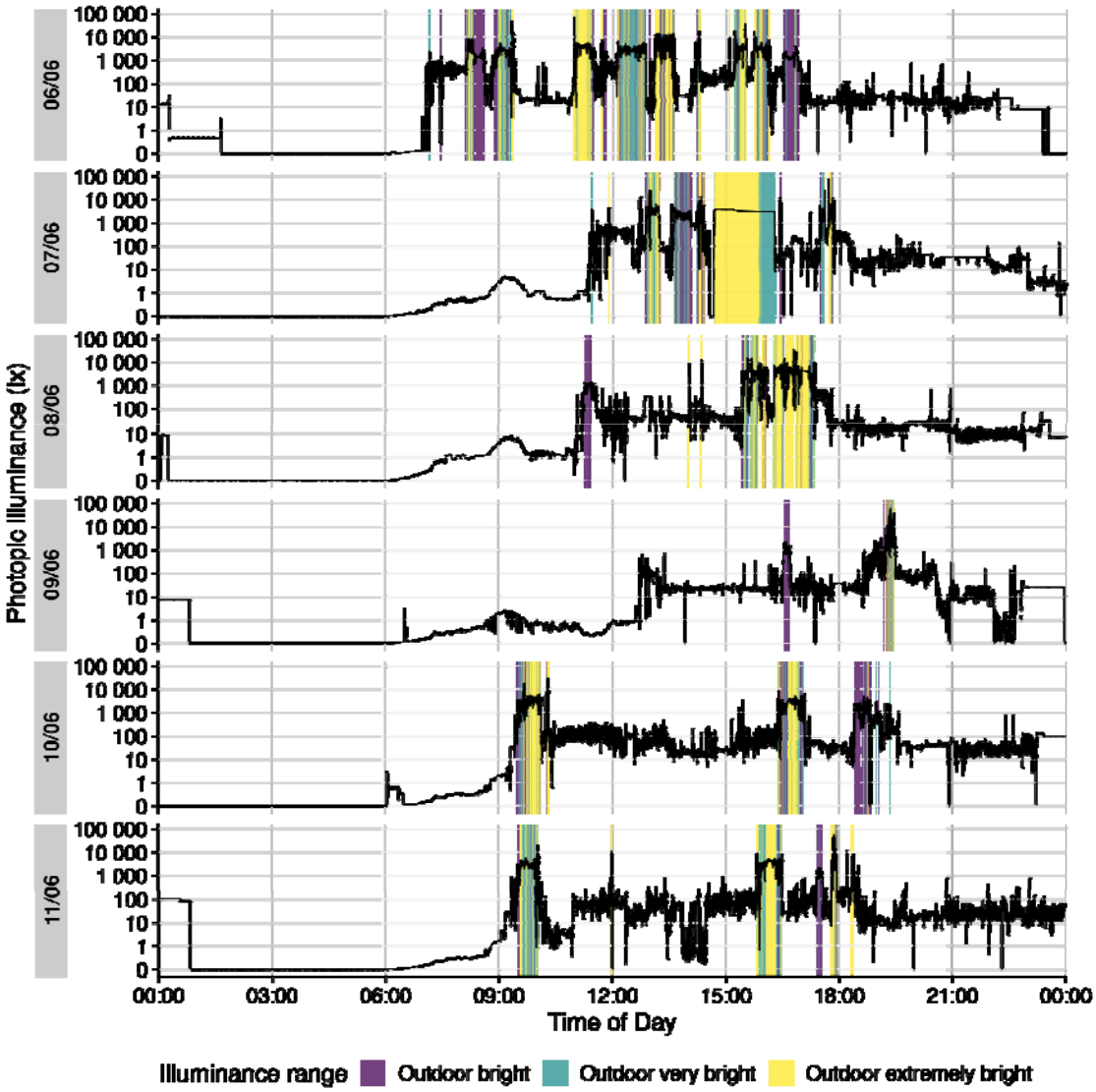


Figure 8: Outdoor light exposure over time. Colored bands indicate periods when illuminance exceeded outdoor thresholds: violet for ≥1000 lx, green for ≥2000 lx, and yellow for ≥3000 lx. Grey shaded regions denote night (from civil dusk to dawn).

Frequency of transitions from indoor to outdoor light

We next consider how often the subject moved from an indoor light environment to an outdoor-equivalent environment. We operationally define an "outdoor transition" as a change from <1000 lx to ≥1000 lx. Using the cleaned `VEET` data, we extract all instances where illuminance crosses that threshold from below to above.

Table 12 shows the average number of such transitions per day. Note that if data are recorded at a fine temporal resolution (5 s here), very brief excursions above 1000 lx could count as transitions and inflate this number. Indeed, the initial count is fairly high, reflecting fleeting spikes above 1000 lx that might not represent meaningful outdoor exposures.

```
dataVEET |>
  # label each interval as Outdoor (Lux≥1000) or not:
  extract_states(Outdoor, Lux >= 1000, group.by.state = FALSE) |>
  # filter where the previous state was "indoor" and current is "outdoor":
  filter(!lead(Outdoor) & Outdoor) |>
  summarize_numeric(prefix = "mean ", # average and provide counts
    remove = c("Datetime", "Outdoor", "start", "end", "duration"),
    add.total.duration = FALSE) |>
  mean_daily(prefix = "") |>
  gt() |> fmt_number(episodes, decimals = 0) |>
  fmt_duration(`mean epoch`, input_units = "seconds",
               output_units = "seconds")
```

	mean epoch	episodes
Mean daily	5s	64
Weekday	5s	72
Weekend	5s	46

Table 12: Average daily count of transitions from indoor (<1000 lx) to outdoor (≥1000 lx) lighting when looking at 5-second epochs

To obtain a more meaningful measure, we can require that the outdoor state persists for some minimum duration to count as a true transition (filtering out momentary fluctuations around the 1000 lx mark). For example, we can require that once ≥1000 lx is reached, it continues for at least 5 minutes (allowing short interruptions up to 20 s). **Table 13** applies this criterion, resulting in a lower, more plausible transition count.

```
dataVEET |>
  extract_clusters(Lux >= 1000,
                   cluster.duration = "5 min",
                   interruption.duration = "20 secs",
                   return.only.clusters = FALSE,
                   drop.empty.groups = FALSE) |>
  filter(!lead(is.cluster) & is.cluster) |>
  summarize_numeric(prefix = "mean ",
```

```
    remove = c("Datetime", "start", "end", "duration"),
    add.total.duration = FALSE) |>
  mean_daily(prefix = "") |>
  gt() |> fmt_number(episodes, decimals = 0)
```

	mean epoch	episodes
Mean daily	5s	5
Weekday	5s	6
Weekend	5s	4

Table 13: Daily indoor-to-outdoor transition count (requiring ≥5 min duration of ≥1000 lx to count)

Longest sustained bright-light period

The final light exposure metric we illustrate is the longest continuous period above a certain illuminance threshold (often termed Longest Period Above Threshold, e.g. PAT1000 for 1000 lx). This gives a sense of the longest outdoor exposure in a day. Along with it, one might report the total duration above that threshold in the day (TAT1000). While we could derive these from the earlier analyses, LightLogR provides dedicated metric functions for such calculations, which can compute multiple related metrics at once.

Using the function `period_above_threshold()` for PAT and `duration_above_threshold()` for TAT, we calculate both metrics for the 1000 lx threshold. **Table 14** shows the mean of these metrics across days (i.e., average longest bright period and average total bright time per day).

```
dataVEET |>
  summarize(
    period_above_threshold(Lux, # specify which variable is of interest
                           Datetime, # which variable carries datetimes
                           threshold = 1000, # what is threshold
                           na.rm = TRUE, # whether NA(missing) are ignored
                           as.df = TRUE # automated naming of output cols
                           ),
    # consistent interface across functions:
    duration_above_threshold(Lux, Datetime, threshold = 1000,
                             na.rm = TRUE, as.df = TRUE),
    .groups = "keep"
  ) |>
  to_mean_daily("")
```

	period above 1000	duration above 1000
Mean daily	1987s (~33.12 minutes)	6709s (~1.86 hours)
Weekday	2501s (~41.68 minutes)	8164s (~2.27 hours)
Weekend	702s (~11.7 minutes)	3070s (~51.17 minutes)

Table 14: Longest period and total duration above 1000 lx (PAT1000 and TAT1000)

Spectrum

The `VEET` device's spectral sensor provides rich data beyond simple lux values, but it requires reconstruction of the actual light spectrum from raw sensor counts. We processed the spectral sensor data in order to compute two example spectrum-based metrics. Detailed data import, normalization, and spectral reconstruction steps are given in **Supplement 1**; here we present the resulting metrics. Briefly, the `VEET`'s spectral sensor recorded counts in ten wavelength bands (roughly 415 nm to 910 nm), plus a `Dark` and a `Clear` channel. After normalizing by sensor gain and applying the calibration matrix, we obtained an estimated spectral irradiance distribution for each interval in the recording. With these reconstructed spectra, we can derive novel metrics that consider spectral content of the light.

Note that spectrum-based metrics in wearable data are relatively new and less established compared to distance or broadband light metrics. The following examples illustrate potential uses of spectral data in a theoretical sense, which can be adapted as needed for specific research questions.

Ratio of short- vs. long-wavelength light

Our first spectral metric is the ratio of short-wavelength light to long-wavelength light, which is relevant, for example, in assessing the blue-light content of exposure. We define "short" wavelengths here as 400–500 nm and "long" as 600–700 nm. Using the list-column of spectra in our dataset, we integrate each spectrum over these ranges (using `spectral_integration()`), and then compute the ratio short/long for each time interval. We then summarize these ratios per day.

```
dataVEET <- dataVEET2 |>
  select(Id, Date, Datetime, Spectrum) |> # focus on these variables
```

```
  mutate(
    short = Spectrum |>
            map_dbl(spectral_integration, wavelength.range = c(400, 500)),
    long  = Spectrum |>
            map_dbl(spectral_integration, wavelength.range = c(600, 700)),
    `sl ratio` = ifelse(is.nan(short / long), NA, short / long)
  )
```

Table 15 shows the average short/long wavelength ratio, averaged over each day (and then as weekday/weekend means if applicable). In this dataset, the values give an indication of the spectral balance of the light the individual was exposed to (higher values mean relatively more short-wavelength content).

```
dataVEET |>
  summarize_numeric(prefix = "", remove = c("Datetime", "Spectrum")) |>
  # mean_daily(prefix = "") |>
  gt() |>
  fmt_number(decimals = 1, scale_by = 1000) |>
  fmt_number(`sl ratio`, decimals = 3) |>
  cols_hide(episodes)
```

date	short	long	sl-ratio
2025-06-18	44.1	42.2	0.524
2025-06-20	69.2	49.1	0.336

Table 15: Average (mW/m²) and ratio of short-wavelength (400–500 nm) to long-wavelength (600–700 nm) light

Short-wavelength light at specific times of day

The second spectral example examines short-wavelength light exposure as a function of time of day. Certain studies might be interested in, for instance, blue-light exposure during midday versus morning or night. We demonstrate three approaches: (a) filtering the data to a specific local time window, and (b) aggregating by hour of day to see a daily profile of short-wavelength exposure. Additionally, we (c) look at differences between day and night periods.

Local morning exposure

Table 16 isolates the time window between 7:00 and 11:00 each day and computes the average short-wavelength irradiance in that interval. This represents a

straightforward query: “How much blue light does the subject get in the morning on average?”

```
dataVEET |>
  filter_Time(start = "7:00:00", end = "11:00:00") |>  # filter times
  select(-c(Spectrum, long, `sl ratio`, Time, Datetime)) |>
  summarize_numeric(prefix = "") |>
  # mean_daily()|> # not used, as these data do not cover weekends
  gt() |> fmt_number(short, scale_by = 1000) |>
  cols_label(short = "Short-wavelength irradiance (mW/m²)") |>
  cols_hide(episodes)
```

Date	Short-wavelength irradiance (mW/m²)
2025-06-18	5.44
2025-06-20	0.95

Table 16: Average short-wavelength light (400–500 nm) exposure between 7:00 and 11:00 each day

Hourly profile across the day

To visualize short-wavelength exposure over the course of a day, we aggregate the data into hourly bins. We cut the timeline into 1-hour segments (using local time), compute the mean short-wavelength irradiance in each hour for each day. **Figure 9** shows the resulting diurnal profile, with short-wavelength exposure expressed as a fraction of the daily maximum for easier comparison.

```
# Prepare hourly binned data
dataVEETtime <- dataVEET |>
  cut_Datetime(unit = "1 hour", type = "floor", group_by = TRUE) |>
  select(-c(Spectrum, long, `sl ratio`, Datetime)) |>
  summarize_numeric(prefix = "") |>
  add_Time_col(Datetime.rounded)  |>   # add a Time column (hour of day)
  mutate(rel_short = short / max(short))

#creating the plot
dataVEETtime |>
  ggplot(aes(x=Time, y = rel_short)) +
  geom_col(aes(fill = factor(Date)), position = "dodge") +
  ggsci::scale_fill_jco() +
  theme_minimal() +
  labs(y = "Normalized short-wavelength irradiance",
       x = "Local time (HH:MM)",
       fill = "Date") +
```

```
  scale_y_continuous(labels = scales::label_percent()) +
  scale_x_time(labels = scales::label_time(format = "%H:%M"))
```

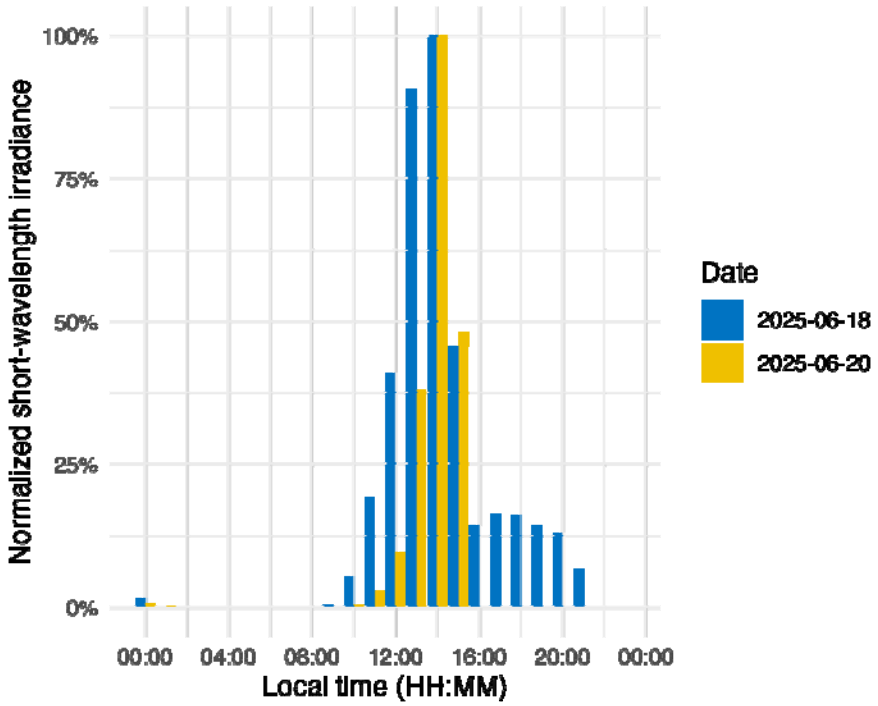


Figure 9: Diurnal profile of short-wavelength light exposure. Each bar represents the average short-wavelength irradiance at that hour of the day (0–23 h), normalized to the daily maximum.

Day vs. night (photoperiod)

Finally, we compare short-wavelength exposure during daytime vs. nighttime. Using civil dawn and dusk information (based on geographic coordinates, here set for Houston, TX, USA), we label each measurement as day or night and then compute the total short-wavelength exposure in each period. **Table 17** summarizes the daily short-wavelength dose received during the day vs. during the night.

```
dataVEET |>
  select(-c(Spectrum, long, `sl ratio`)) |>
  add_photoperiod(coordinates) |>
  group_by(photoperiod.state, .add = TRUE) |>
  summarize_numeric(
    prefix = "", remove = c("dawn", "dusk", "photoperiod", "Datetime")) |>
  group_by(Id, photoperiod.state) |>
#   mean_daily()|> # not used, as these data do not cover weekends
  select(-episodes) |>
```

```
  pivot_wider(names_from =photoperiod.state, values_from = short) |>
  gt() |> fmt_number(scale_by = 1000, decimals = 1)
```

date	day	night
2025-06-18	73.9	2.6
2025-06-20	112.0	1.0

Table 17: Short wavelength light exposure (mW/m²) during the day and at night

In the code section above, `add_photoperiod(coordinates)` is used as a convenient way to add columns to the data frame, indicating for each timestamp whether it was day or night, given the latitude/longitude.

Discussion and conclusion

This tutorial demonstrates a standardized, step-by-step pipeline to calculate a variety of visual experience metrics. We illustrated how a combination of LightLogR functions and tidyverse workflows can yield clear and reproducible analyses for wearable device data. While the full pipeline is detailed, each metric is computed through a dedicated sequence of well-documented steps. By leveraging LightLogR's framework alongside common data analysis approaches, the process remains transparent and relatively easy to follow. The overall goal is to make analysis transparent (with open-source functions), accessible (through thorough documentation, tutorials, and human-readable function naming, all under an MIT license), robust (the package includes over 800 unit tests and continuous integration with bug tracking on GitHub), and community-driven (open feature requests and contributions via GitHub).

Even with standardized pipelines, researchers must still make and document many decisions during data cleaning, time-series handling, and metric calculations — especially for complex metrics that involve grouping data in multiple ways (for example, grouping by distance range as well as by duration for cluster metrics). We have highlighted these decision points in the tutorial (such as how to handle irregular intervals, choosing thresholds for "near" distances or "outdoor" light, and deciding on minimum durations for sustained events). Explicitly considering and reporting these choices is important for reproducibility and for comparing results across studies.

The broad set of features in LightLogR — ranging from data import and cleaning tools (for handling time gaps and irregularities) to visualization functions and metric calculators — make it a powerful toolkit for visual experience research. Our examples spanned myopia-related metrics and also circadian-related ones, demonstrating the versatility of a unified analysis approach. By using community-supported tools and workflows, researchers in vision science, chronobiology, myopia, and related fields can reduce time spent on low-level data wrangling and focus more on interpreting results and advancing scientific understanding.

Statements

Data availability statement

All data in this tutorial and Supplement 1 are available from the GitHub repository: https://github.com/tscnlab/ZaunerEtAl_JVis_2025/, archived on Zenodo: https://doi.org/10.5281/zenodo.16566014 under a CC-BY license (Code under MIT license).

Funding statement

JZ's position is funded by the MeLiDos project. The project has received funding from the European Partnership on Metrology (22NRM05 MeLiDos), co-financed from the European Union's Horizon Europe Research and Innovation Programme and by the Participating States. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or EURAMET. Neither the European Union nor the granting authority can be held responsible for them. JZ, LAO, and MS received research funding from Reality Labs Research. The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

Conflict of interest statement

JZ declares the following potential conflict of interest in the past five years (2021-2025). Funding: Received research funding from Reality Labs Research.

AN declares the following potential conflicts of interest in the past five years (2021-2025). none

LAO declares the following potential conflict of interest in the past five years (2021-2025). Consultancy: Zeiss, Alcon, EssilorLuxottica; Research support: Topcon, Meta, LLC; Patents: US 11375890 B2

MS declares the following potential conflicts of interest in the past five years (2021–2025). Academic roles: Member of the Board of Directors, Society of Light, Rhythms, and Circadian Health (SLRCH); Chair of Joint Technical Committee 20 (JTC20) of the International Commission on Illumination (CIE); Member of the Daylight Academy; Chair of Research Data Alliance Working Group Optical Radiation and

Visual Experience Data. Remunerated roles: Speaker of the Steering Committee of the Daylight Academy; Ad-hoc reviewer for the Health and Digital Executive Agency of the European Commission; Ad-hoc reviewer for the Swedish Research Council; Associate Editor for LEUKOS, journal of the Illuminating Engineering Society; Examiner, University of Manchester; Examiner, Flinders University; Examiner, University of Southern Norway. Funding: Received research funding and support from the Max Planck Society, Max Planck Foundation, Max Planck Innovation, Technical University of Munich, Wellcome Trust, National Research Foundation Singapore, European Partnership on Metrology, VELUX Foundation, Bayerisch-Tschechische Hochschulagentur (BTHA), BayFrance (Bayerisch-Französisches Hochschulzentrum), BayFOR (Bayerische Forschungsallianz), and Reality Labs Research. Honoraria for talks: Received honoraria from the ISGlobal, Research Foundation of the City University of New York and the Stadt Ebersberg, Museum Wald und Umwelt. Travel reimbursements: Daimler und Benz Stiftung. Patents: Named on European Patent Application EP23159999.4A (“System and method for corneal-plane physiologically-relevant light logging with an application to personalized light interventions related to health and well-being”). With the exception of the funding source supporting this work, M.S. declares no influence of the disclosed roles or relationships on the work presented herein.

Statement of generative AI and AI-assisted technologies in the writing process

The authors used ChatGPT during the preparation of this work. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Use of AI in contributor roles based on the CRediT taxonomy. Funding acquisition, investigation, project administration, resources, and supervision were deemed irrelevant in this context and thus removed:

Conceptualization: no

Data curation: no

Formal analysis: bug fixing

Methodology: no

Software: bug fixing

Validation: no

Visualization: tweaking of options

Writing – original draft: abstract refinement

Writing – review & editing: improve readability and language

Contributions

Conceptualisation: JZ

Data curation: JZ, LAO

Formal analysis: JZ

Funding acquisition: MS, AN

Methodology: LAO, AN, JZ, MS

Project administration: MS, JZ

Software: JZ

Resources: JZ, MS

Supervision: MS, LAO, AN

Validation: LAO, AN

Visualisation: JZ

Writing – original draft: JZ

Writing – review & editing: JZ, AN, LAO, MS

References

Bhandari, Khob R, and Lisa A Ostrin. 2020. “Validation of the Clouclip and Utility in Measuring Viewing Distance in Adults.” *Ophthalmic and Physiological Optics* 40 (6): 801–14. https://doi.org/https://doi.org/10.1111/opo.12735.

Biller, A. M., P. Balakrishnan, and M. Spitschan. 2024. “Behavioural Determinants of Physiologically-Relevant Light Exposure.” Journal Article. *Commun Psychol* 2 (1): 114. https://doi.org/10.1038/s44271-024-00159-5.

Blume, C., C. Garbazza, and M. Spitschan. 2019. “Effects of Light on Human Circadian Rhythms, Sleep and Mood.” Journal Article. *Somnologie (Berl)* 23 (3): 147–56. https://doi.org/10.1007/s11818-019-00215-x.

Brown, T. M., G. C. Brainard, C. Cajochen, C. A. Czeisler, J. P. Hanifin, S. W. Lockley, R. J. Lucas, et al. 2022. “Recommendations for Daytime, Evening, and Nighttime Indoor Light Exposure to Best Support Physiology, Sleep, and Wakefulness in Healthy Adults.” Journal Article. *PLoS Biol* 20 (3): e3001571. https://doi.org/10.1371/journal.pbio.3001571.

Dahlmann-Noor, A. H., D. Bokre, M. Khazova, and L. L. A. Price. 2025. “Measuring the Visual Environment of Children and Young People at Risk of Myopia: A Scoping Review.” Journal Article. *Graefes Arch Clin Exp Ophthalmol*. https://doi.org/10.1007/s00417-024-06719-z.

Gibaldi, A., E. N. Harb, C. F. Wildsoet, and M. S. Banks. 2024. “A Child-Friendly Wearable Device for Quantifying Environmental Risk Factors for Myopia.” Journal Article. *Transl Vis Sci Technol* 13 (10): 28. https://doi.org/10.1167/tvst.13.10.28.

Hartmeyer, S. L., and M. Andersen. 2023. “Towards a Framework for Light-Dosimetry Studies: Quantification Metrics.” Journal Article. *Lighting Research & Technology* 56 (4): 337–65. https://doi.org/10.1177/14771535231170500.

Hartmeyer, S. L., F. S. Webler, and M. Andersen. 2022. “Towards a Framework for Light-Dosimetry Studies: Methodological Considerations.” Journal Article. *Lighting Research & Technology* 55 (4-5): 377–99. https://doi.org/10.1177/14771535221103258.

Hönekopp, A., and S. Weigelt. 2023. “Using Light Meters to Investigate the Light-Myopia Association - a Literature Review of Devices and Research Methods.” Journal Article. *Clin Ophthalmol* 17: 2737–60. https://doi.org/10.2147/OPTH.S420631.

Mohamed, A., V. Kalavally, S. W. Cain, A. J. K. Phillips, E. M. McGlashan, and C. P. Tan. 2021. “Wearable Light Spectral Sensor Optimized for Measuring Daily Alpha-Opic Light Exposure.” Journal Article. *Opt Express* 29 (17): 27612–27. https://doi.org/10.1364/OE.431373.

Okudaira, N., D. F. Kripke, and J. B. Webster. 1983. “Naturalistic Studies of Human Light Exposure.” Journal Article. *Am J Physiol* 245 (4): R613–5. https://doi.org/10.1152/ajpregu.1983.245.4.R613.

Sah, Raman Prasad, Pavan Kalyan Narra, and Lisa A. Ostrin. 2025. “A Novel Wearable Sensor for Objective Measurement of Distance and Illumination.” *Ophthalmic and Physiological Optics* 00 (n/a): 1–13. https://doi.org/https://doi.org/10.1111/opo.13523.

Sullivan, David, Aaron Nicholls, George Hatoun, Samuel Thompson, Cory Schwarzmiller, Fathollah Memarzanjany, Alyssa Gunderson, et al. 2024. “The Visual Experience Evaluation Tool: A Myopia Research Instrument for Quantifying Visual Experience.” *bioRxiv*. https://doi.org/10.1101/2024.09.20.614212.

Webler, F. S., M. Spitschan, R. G. Foster, M. Andersen, and S. N. Peirson. 2019. “What Is the ’Spectral Diet’ of Humans?” Journal Article. *Curr Opin Behav Sci* 30: 80–86. https://doi.org/10.1016/j.cobeha.2019.06.006.

Wen, Longbo, Yingpin Cao, Qian Cheng, Xiaoning Li, Lun Pan, Lei Li, HaoGang Zhu, Weizhong Lan, and Zhikuan Yang. 2020. “Objectively Measured Near Work, Outdoor Exposure and Myopia in Children.” *British Journal of Ophthalmology* 104 (11): 1542–47. https://doi.org/10.1136/bjophthalmol-2019-315258.

Wen, Longbo, Qian Cheng, Yingpin Cao, Xiaoning Li, Lun Pan, Lei Li, Haogang Zhu, Ian Mogran, Weizhong Lan, and Zhikuan Yang. 2021. “The Clouclip, a Wearable Device for Measuring Near-Work and Outdoor Time: Validation and Comparison of Objective Measures with Questionnaire Estimates.” *Acta Ophthalmologica* 99 (7): e1222–35. https://doi.org/https://doi.org/10.1111/aos.14785.

Wen, Longbo, Qian Cheng, Weizhong Lan, Yingpin Cao, Xiaoning Li, Yiqiu Lu, Zhenghua Lin, Lun Pan, Haogang Zhu, and Zhikuan Yang. 2019. “An Objective Comparison of Light Intensity and Near-Visual Tasks Between Rural and Urban School Children in China by a Wearable Device Clouclip.”

Translational Vision Science & Technology 8 (6): 15–15. https://doi.org/10.1167/tvst.8.6.15.

Williams, Rachel, Suyash Bakshi, Edwin J Ostrin, and Lisa A Ostrin. 2019. "Continuous Objective Assessment of Near Work." *Scientific Reports* 9 (1): 6901.

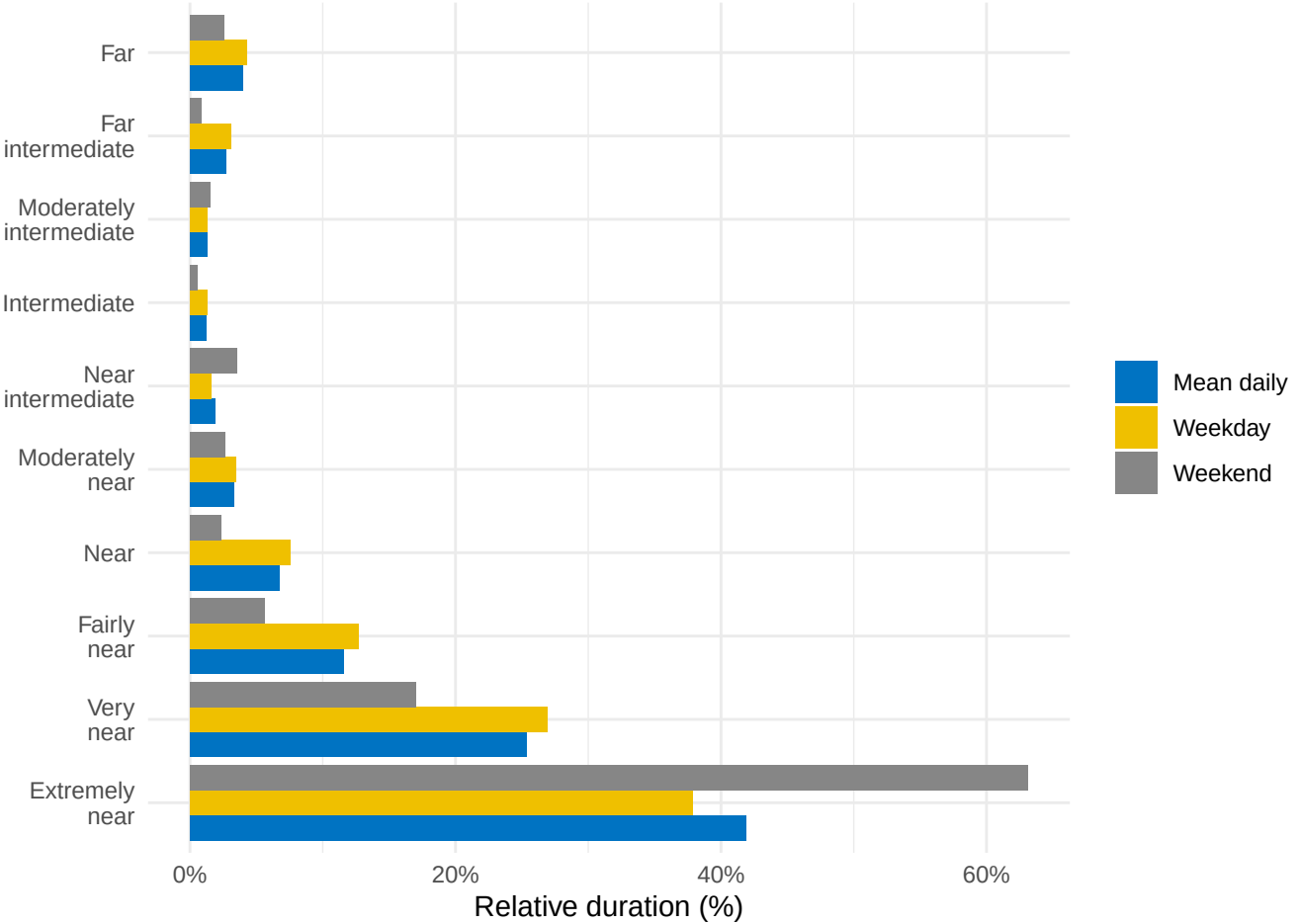
Zauner, J., C. Guidolin, and M. Spitschan. "How to Deal with Darkness: Modeling and Visualization of Zero-Inflated Personal Light Exposure Data on a Logarithmic Scale." *Journal of Biological Rhythms* 0 (0): 07487304251336624. https://doi.org/10.1177/07487304251336624.

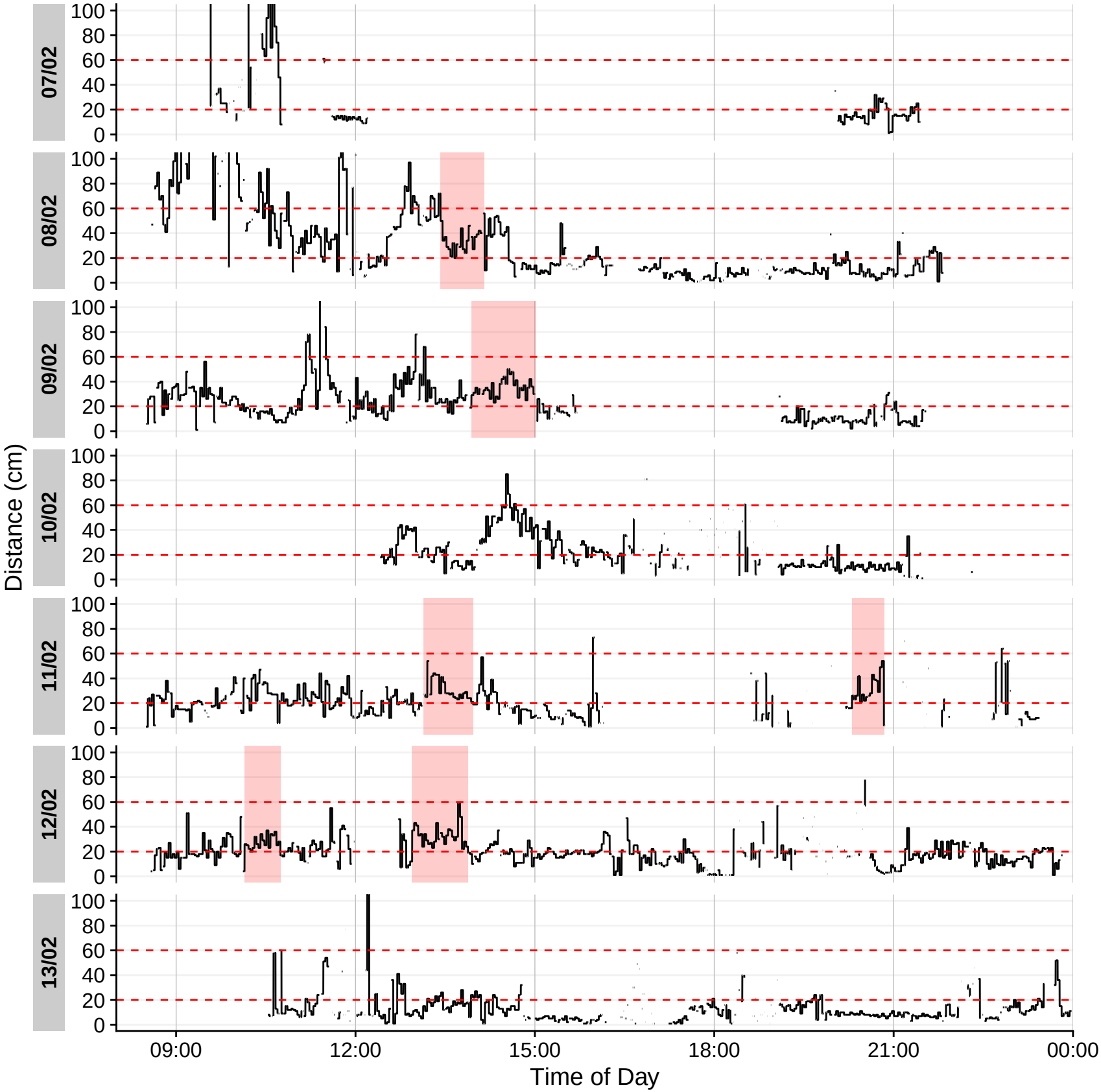
Zauner, J., S. Hartmeyer, and M. Spitschan. 2025. "LightLogR: Reproducible Analysis of Personal Light Exposure Data." Journal Article. *J Open Source Softw* 10 (107): 7601. https://doi.org/10.21105/joss.07601.

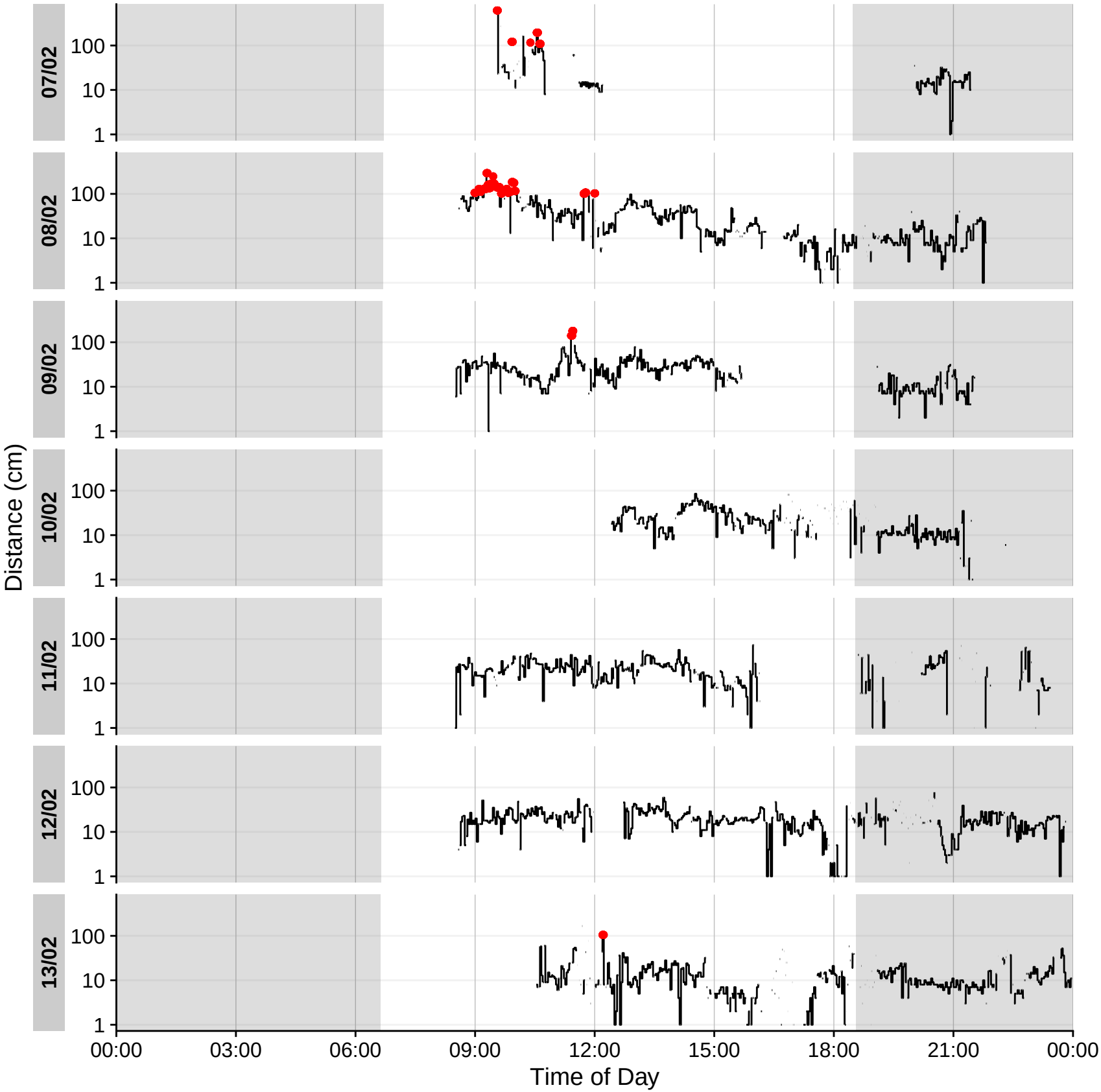
Zauner, J., L. Udovicic, and M. Spitschan. 2024. "Power Analysis for Personal Light Exposure Measurements and Interventions." *PLOS ONE* 19 (12): 1–15. https://doi.org/10.1371/journal.pone.0308768.

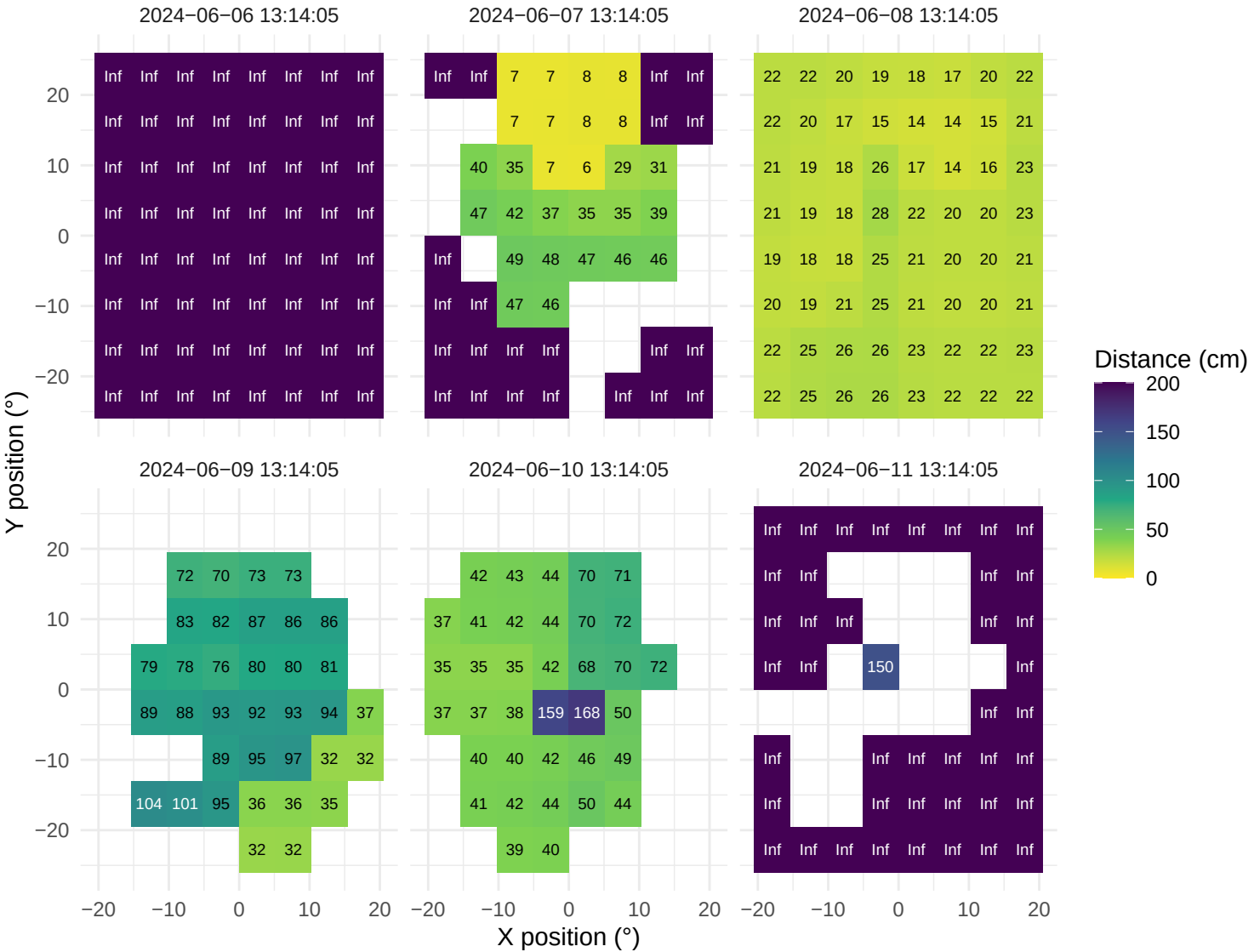
Supplementary Material

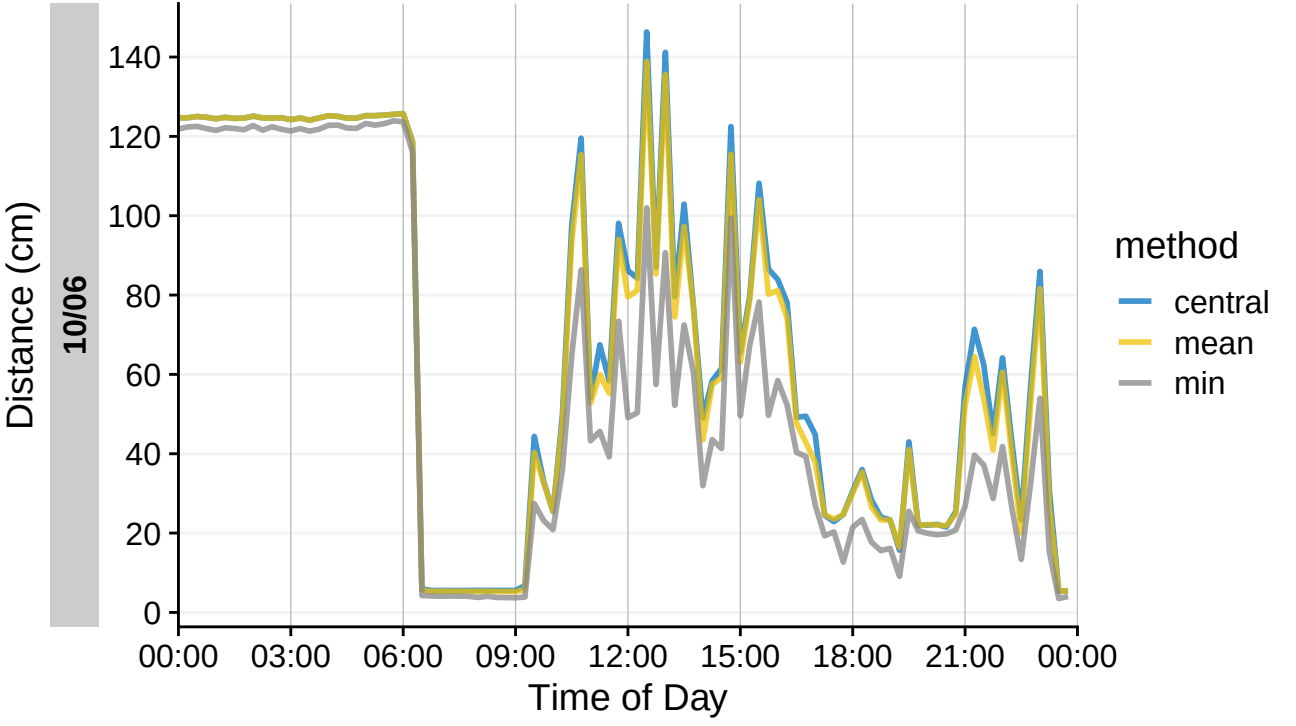
S1. Import and preprocessing file. An HTML file that contains all the code and code results to prepare the data for the analysis in the main manuscript.

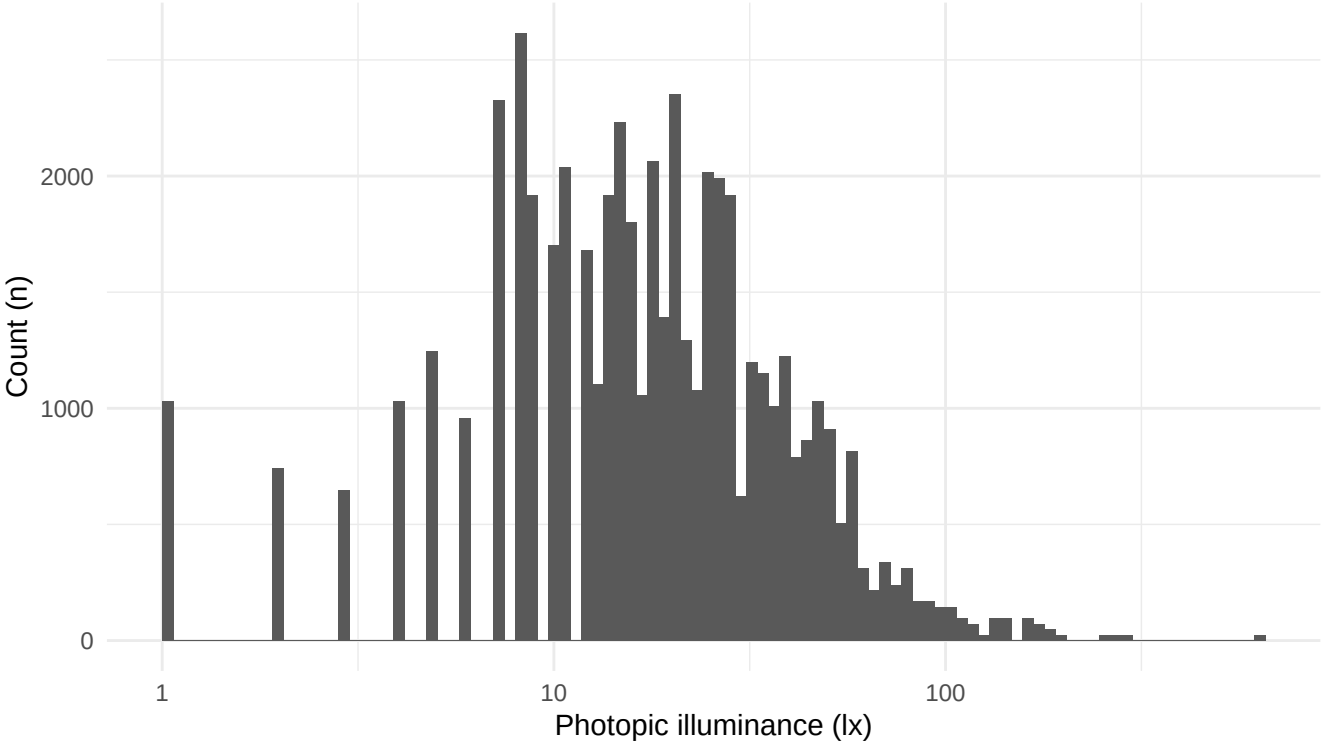

Far
Far intermediate
Moderately intermediate
Intermediate
Near intermediate
Moderately near
Near
Fairly near
Very near
Extremely near
0%
20%
40%
60%
Relative duration (%)
Mean daily
Weekday
Weekend

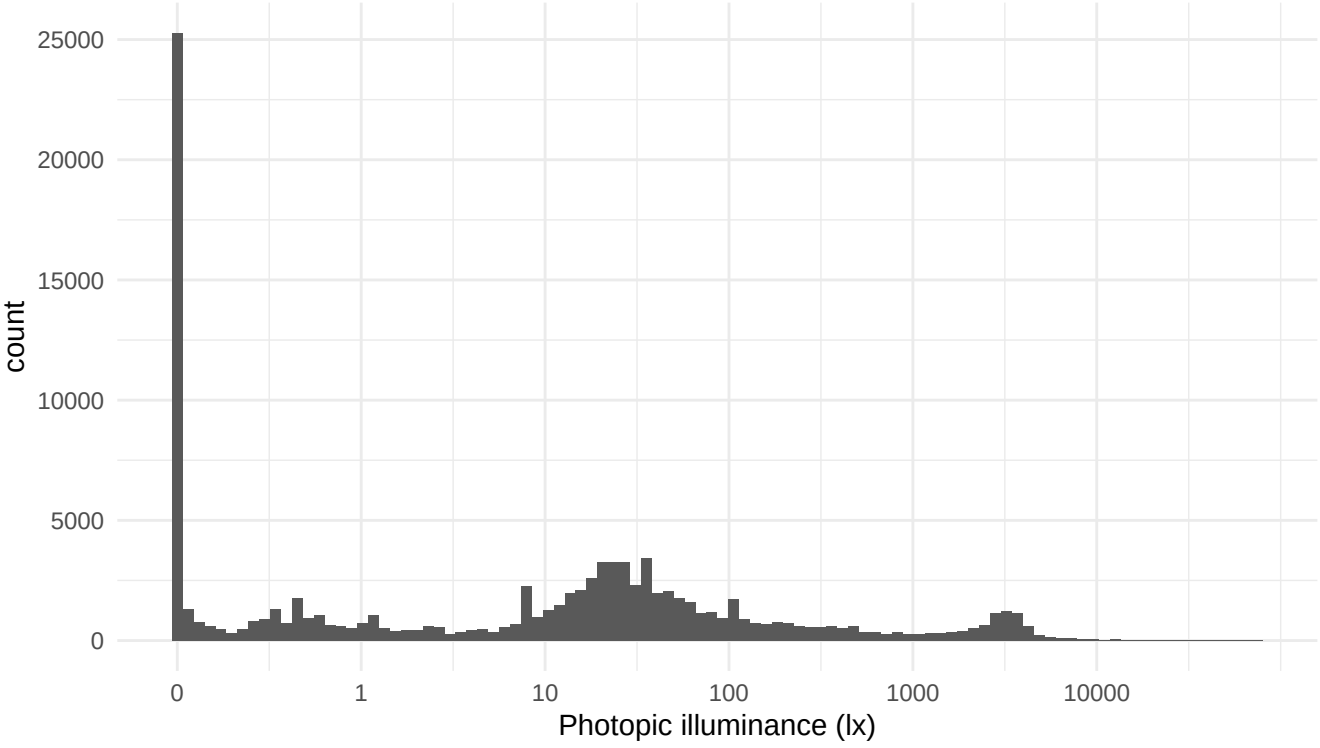

07/02
08/02
09/02
10/02
11/02
12/02
13/02
Distance (cm)
100
80
60
40
20
0
09:00
12:00
15:00
18:00
21:00
00:00
Time of Day

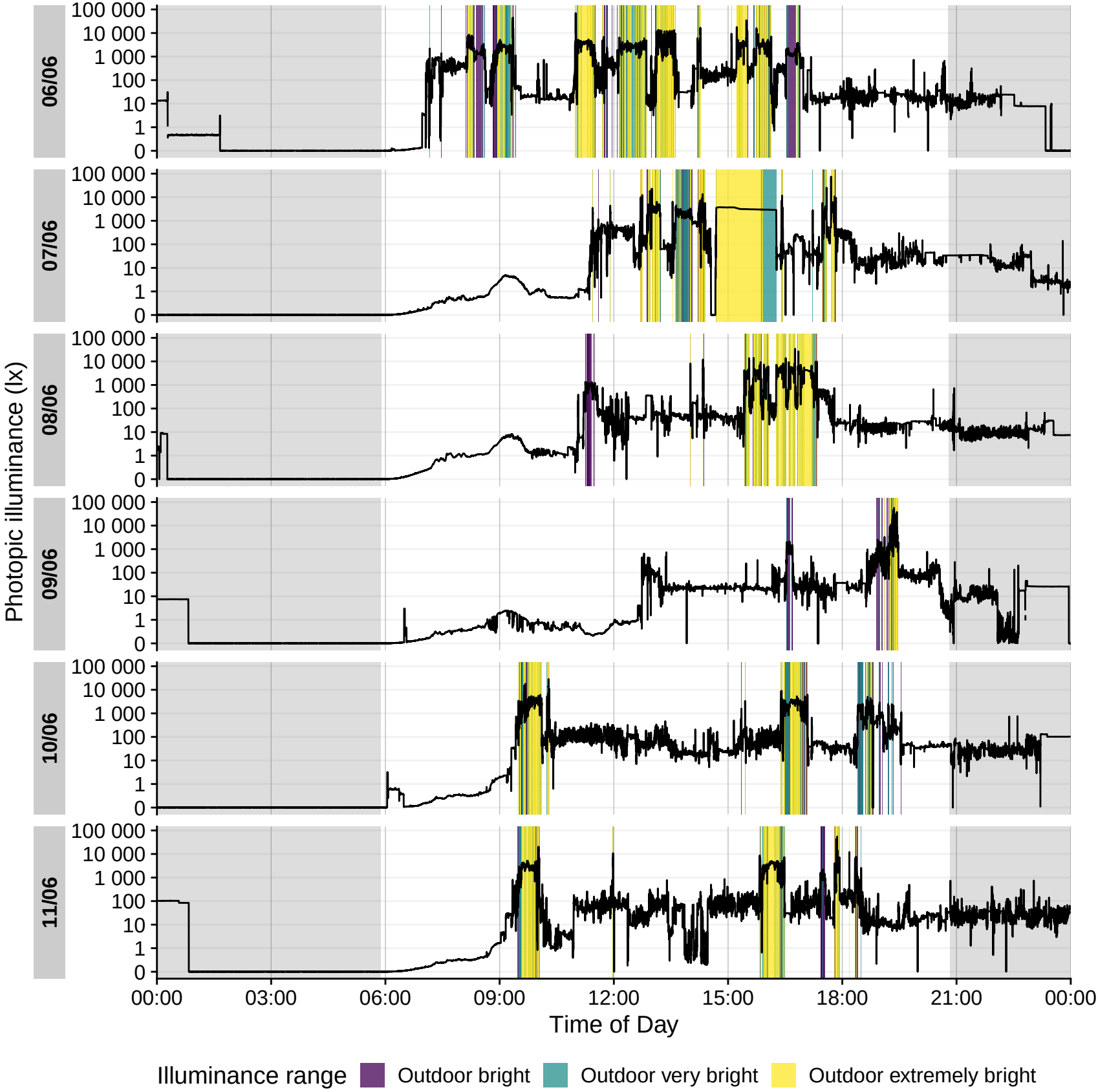

07/02
08/02
09/02
10/02
11/02
12/02
13/02
100
10
1
Distance (cm)
00:00
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00
Time of Day

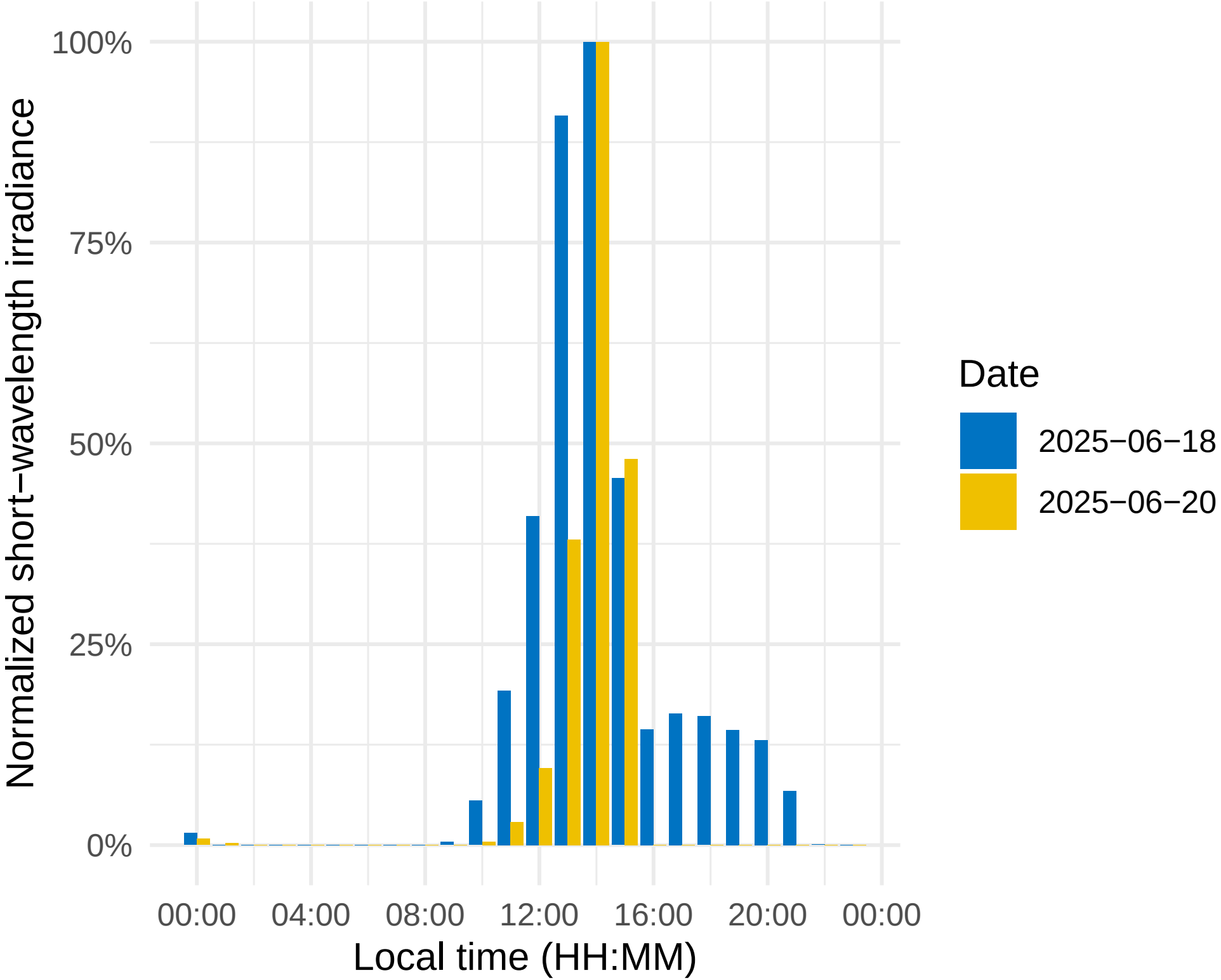

2024−06−06 13:14:05
Inf Inf Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf Inf
2024−06−07 13:14:05
Inf Inf 7 7 8 8 Inf Inf
7 7 8 8 Inf Inf
40 35 7 6 29 31
47 42 37 35 35 39
Inf 49 48 47 46 46
Inf Inf 47 46
Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf
2024−06−08 13:14:05
22 22 20 19 18 17 20 22
22 20 17 15 14 14 15 21
21 19 18 26 17 14 16 23
21 19 18 28 22 20 20 23
19 18 18 25 21 20 20 21
20 19 21 25 21 20 20 21
22 25 26 26 23 22 22 23
22 25 26 26 23 22 22 22
2024−06−09 13:14:05
72 70 73 73
83 82 87 86 86
79 78 76 80 80 81
89 88 93 92 93 94 37
89 95 97 32 32
104 101 95 36 36 35
32 32
2024−06−10 13:14:05
42 43 44 70 71
37 41 42 44 70 72
35 35 35 42 68 70 72
37 37 38 159 168 50
40 40 42 46 49
41 42 44 50 44
39 40
2024−06−11 13:14:05
Inf Inf Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf
Inf Inf Inf Inf Inf
Inf Inf 150 Inf
Inf Inf
Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf
Inf Inf Inf Inf Inf Inf Inf Inf
Y position (°)
X position (°)
−20 −10 0 10 20
Distance (cm)
200
150
100
50
0


10/06
Distance (cm)
140
120
100
80
60
40
20
0
00:00
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00
Time of Day
method
central
mean
min


Count (n)
2000
1000
0
1
10
100
Photopic illuminance (lx)


25000
20000
15000
10000
5000
0
count
0
1
10
100
1000
10000
Photopic illuminance (lx)


Photopic illuminance (lx)
06/06
07/06
08/06
09/06
10/06
11/06
100 000
10 000
1 000
100
10
1
0
00:00
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00
Time of Day
Illuminance range
Outdoor bright
Outdoor very bright
Outdoor extremely bright


100%
75%
50%
25%
0%
Normalized short−wavelength irradiance
00:00
04:00
08:00
12:00
16:00
20:00
00:00
Local time (HH:MM)
Date
2025−06−18
2025−06−20