QPAD bug explanation

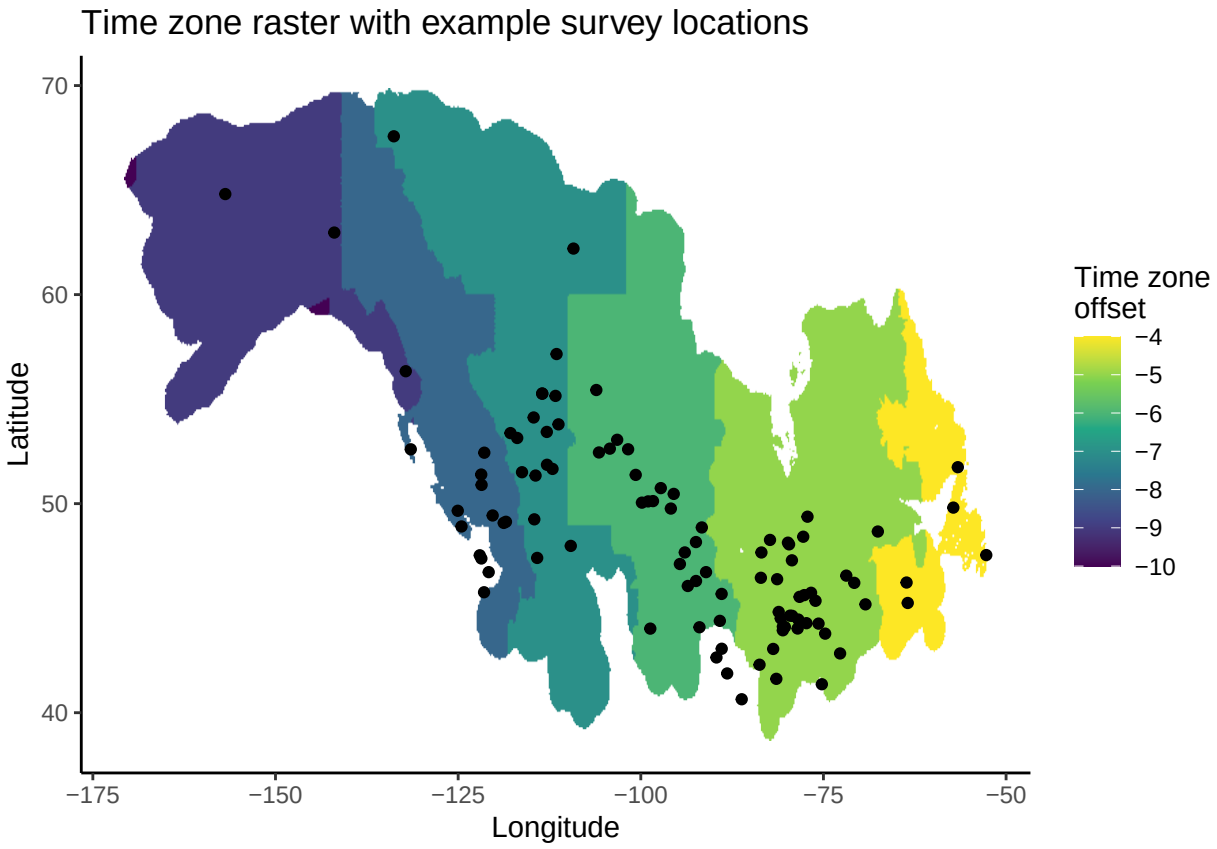
Elly Knight

May 7, 2025

Background

In QPAD, the set of removal models for availability for detection includes a term for time of day, measures as time since sunrise (TSSR). For each species, a set of models are fit and the best-fitting model is selected as the model with the lowest BIC.

The coefficients from that best-fitting model are then used to calculate the mean cue rate for a species as part of the QPAD offset calculation. The functions within the `qpad-offsets` repository and the `wildrtrax` R package that prepare the covariates for cue rate calculation calculate TSSR for the mountain time zone and then use the coordinates of the survey to extract a time zone value for that location from a raster and use that time zone value to adjust the TSSR. We use this raster-based correction approach because it is much more efficient for large datasets than other available time zone lookup functions (see 'Comparison to alternate approach' below)

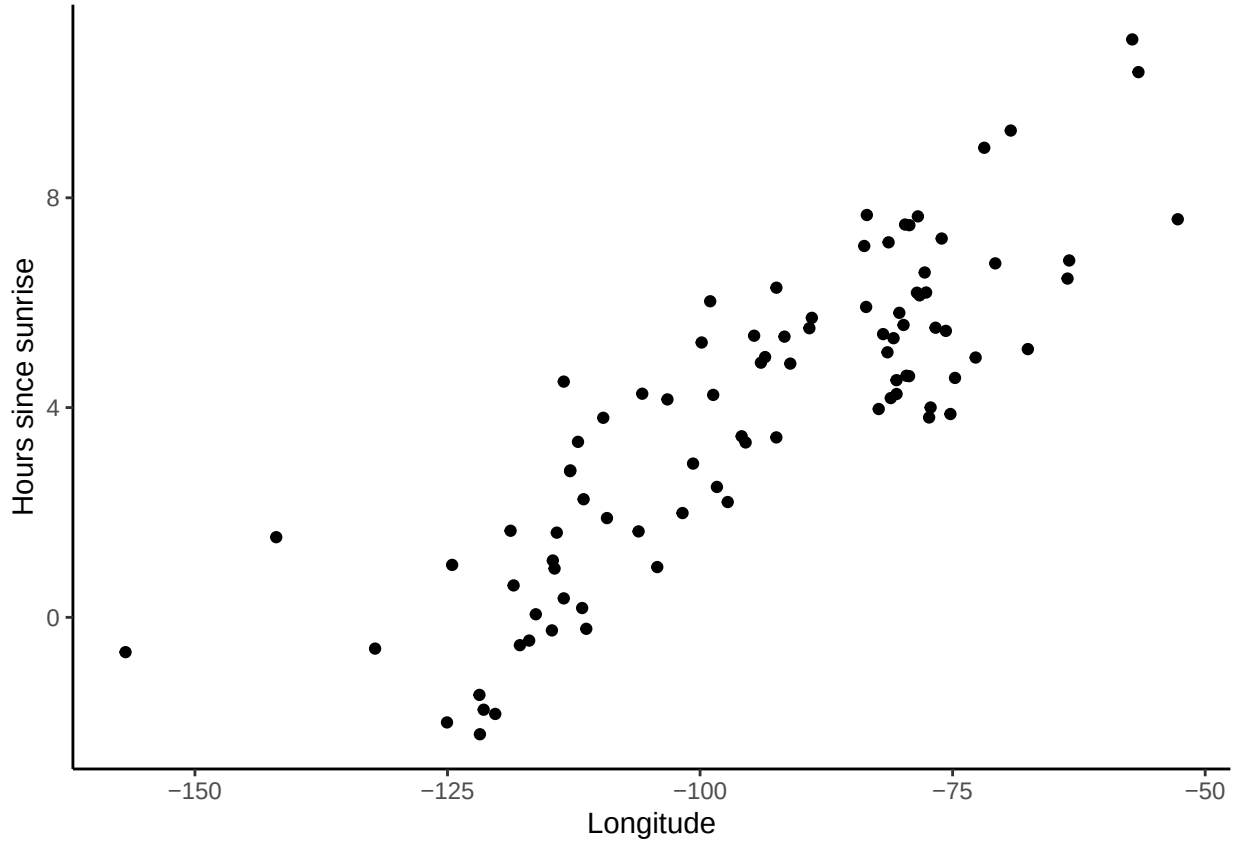


Code bug

In the original code, the adjustment was made by subtracting sunrise from the survey time and adding the time zone offset.

```
# get time zone offset
ltz <- raster::extract(rtz, xy) + 7

# get sunrise in mountain time
sr <- sunriset(cbind("X"=lon, "Y"=lat),
               as.POSIXct(dtm, tz="America/Edmonton"),
               direction="sunrise", POSIXct.out=FALSE) * 24

# calculate TSSR
TSSR_bug <- hour - sr$day_frac + ltz
```

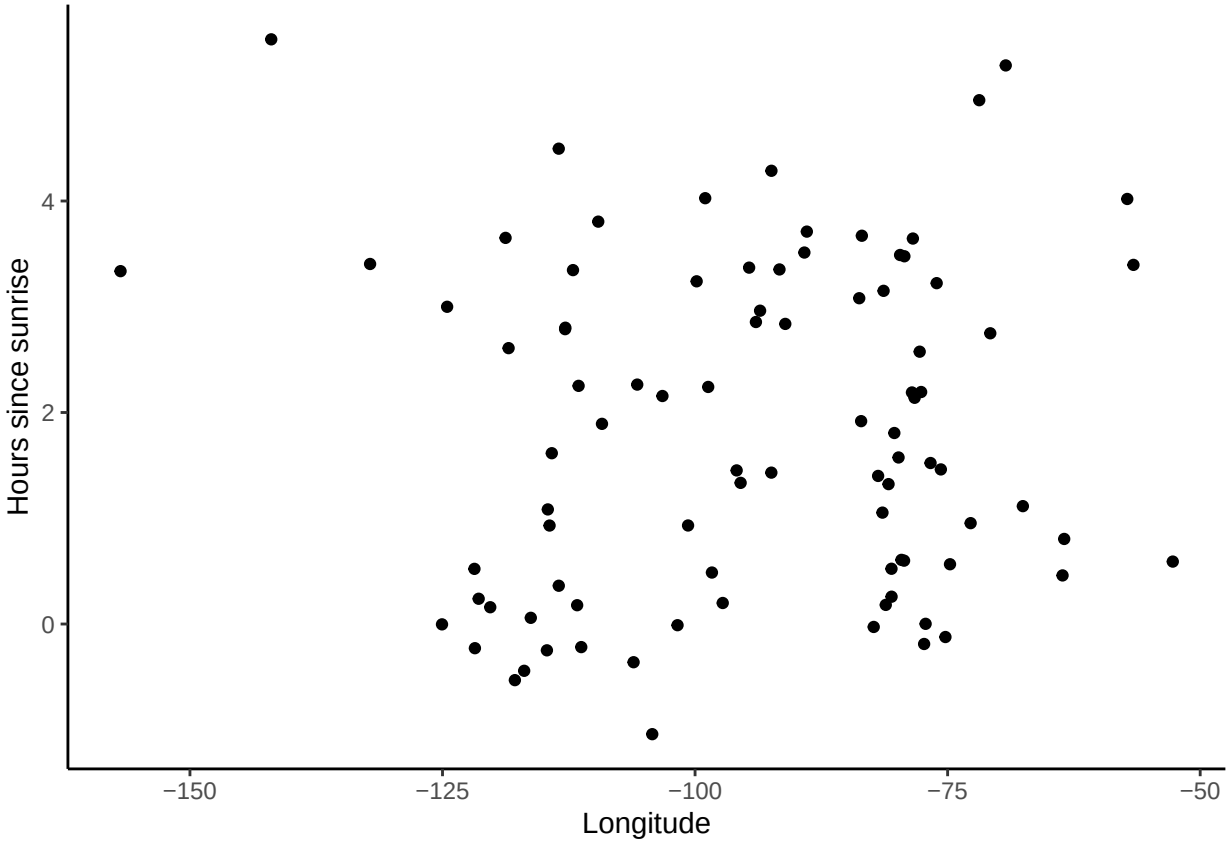
However, the addition of the time zone offset doubles the time zone difference relative to mountain time zone instead of zeroing it. We can detect this bug in the TSSR as a linear relationship between TSSR and longitude.



Bug Correction

If we instead subtract the time zone offset, we see the relationship is nullified.

```
# calculate TSSR
TSSR_fix <- hour - sr$day_frac - ltz
```



Comparison to alternate approach

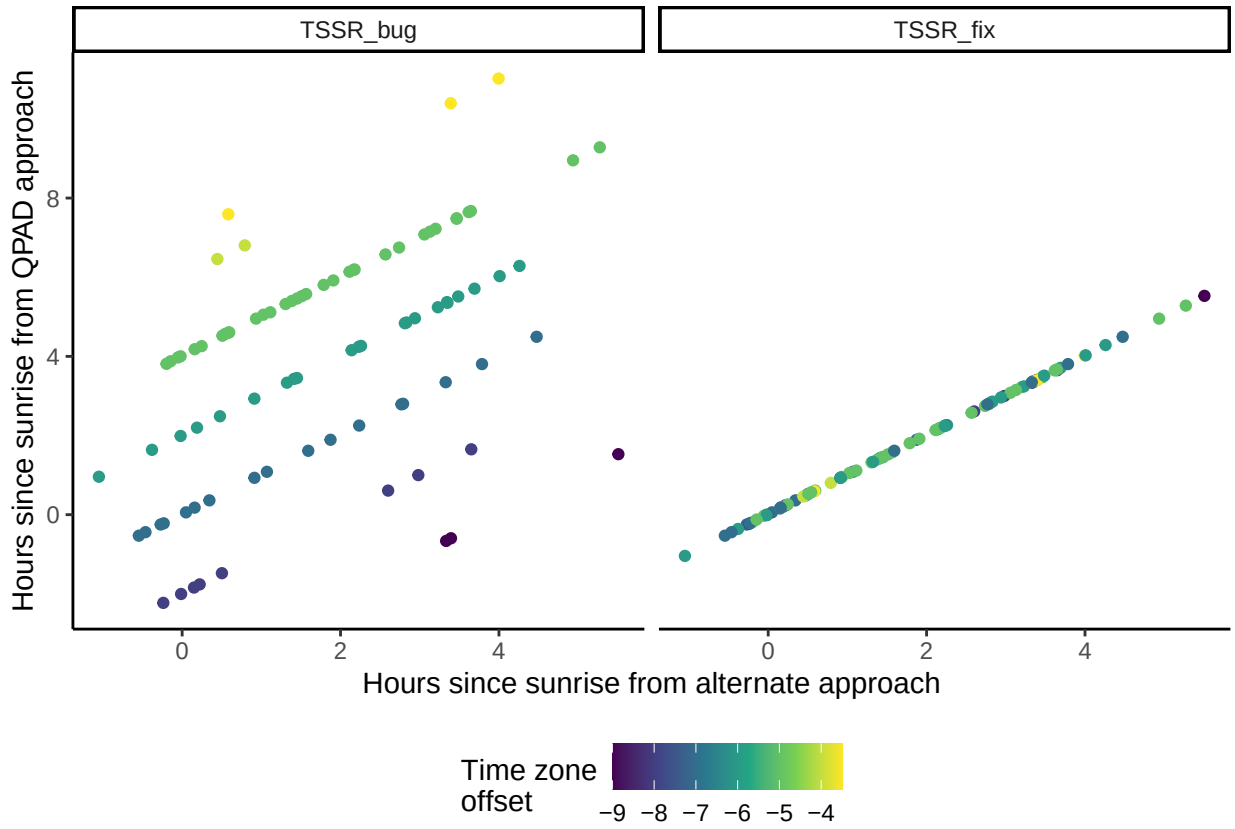
We can double check that subtraction of the time zone offset gives us the correct TSSR by comparing to a slower alternate approach. Here, we use the `lutz` package to look up timezone using the coordinates and use that timezone in the sunrise calculation directly.

```
#look up timezones
dat$tz <- tz_lookup_coords(lat, lon, method="accurate")

#calculate sunrise for each timezone
tzs <- unique(dat$tz)
out <- data.frame()
for(i in 1:length(tzs)){

    #get just the data for that timezone
    dat.i <- dplyr::filter(dat, tz==tzs[i]) |>
        mutate(datetime = ymd_hms(datetime, tz=tzs[i]),
               date = as.Date(datetime, tz=tzs[i]))

    #get sunrise time
    dat.i$sr2 <- getSunlightTimes(data=dat.i, keep="sunrise", tz=tzs[i])$sunrise

    #fix America/Regina
    if(tzs[i]=="America/Regina"){dat.i$sr2 <- dat.i$sr2 + 3600}
```

```
    #output
    out <- rbind(out, dat.i)
}

#calculate tssr
out$TSSR_alternate <- as.numeric(difftime(out$datetime, out$sr), units="hours")
```

Let's compare `TSSR_bug` and `TSSR_fix` to the alernate approach to double check.

```
## Joining with `by = join_by(id, lat, lon, tz)`
```



Looks good, onward!