Filtering and summarizing camera data for analysis

Kara Leimberger

Step 1: Import camera data + prepare for filtering

```
## 1. Create column for unique camera ID 2. Convert start/end columns into
## period objects, so can treat them as times when filtering out afternoon
## videos from covering days
data <- read.csv("../data/import/Camera_data_with_flower_counts_2016-2018_20220712.csv") %>%
    mutate(camera_id = paste(year, patch, plant_species, camera_num, sep = "_")) %>%
    mutate(across(c(video_start, video_end), ~lubridate::hm(.))) %>%
    mutate(across(c(sighting_start, sighting_end), ~lubridate::hms(.)))
```

Step 2: Run some initial checks before filtering

```
## Check strings that failed to parse in previous step
check01 <- data %>%
    filter(if_any(c(video_start, video_end), ~is.na(.)))

## Are there any missing start/end times for sightings within a video?  Filter
## to rows with sighting start/end times (no sightings = NA = no times to parse
## = warning message)
check02 <- data %>%
    filter(if_any(c(sighting_start, sighting_end), ~is.na(.))) %>%
    filter(sightings_yes_or_no == "Y")

## Are there videos from non-priority dates?  'Priority' = days in between
## capture sessions, omitting the day after cover day (for treatment sites) or
## its equivalent (for control sites)
check03 <- data %>%
    filter(video_priority != 1) %>%
    select(contains("day"), everything()) %>%
    arrange(exp_day, control_count_day, trmt_day_after_cover)

## Are there videos without any visible flowers?
check06 <- data %>%
    filter(flowers_camera_video == 0)

## It doesn't really make sense to analyze videos with no visible flowers, but
## out of curiosity...do hummingbirds ever visit anyway?  'Sighting' =
## hummingbird appears on camera, 'Visit' = hummingbird appears to drink from
## flower
check04 <- data %>%
    filter(flowers_camera_video == 0 & sightings_yes_or_no == "Y" & visit_type !=
        "none")

## Are there sightings from the afternoon of cover day?  Covering is a big
## disturbance to the site, so we should NOT use these data...and especially
## should not consider them 'pre' data!
check05 <- data %>%
    filter(cover_day == 1) %>%
    filter(video_end >= lubridate::hm("11:30")) %>%
    filter(sightings_yes_or_no == "Y")
```

Step 3: Filter data for analysis

Remove videos with no flowers on camera (videos were typically not reviewed if they had no flowers, but some were reviewed anyway).

```
## I will still include NAs, since some plants never had flowers visible on
## video (i.e., MARA-VER, GINGY)
data02 <- data %>%
    filter(flowers_camera_video > 0 | is.na(flowers_camera_video))
```

Remove sightings from afternoon of cover day (treatment sites only). Here, ‘afternoon’ = videos starting after 11:30 AM

```
## Videos that start (and end) after 11:30 (will just remove these)
treatment_videos_pm <- data02 %>%
    filter(cover_day == 1 & control_treatment == "treatment") %>%
    filter(video_start > hm("11:30"))

## Videos that span the cutoff time (will change end time to 11:30 and remove
## sightings after this time)
treatment_videos_am_pm <- data02 %>%
    filter(cover_day == 1 & control_treatment == "treatment") %>%
    filter(video_start < hm("11:30")) %>%
    filter(video_end > hm("11:30"))

## Sightings from afternoon of covering
treatment_sightings_pm <- treatment_videos_am_pm %>%
    filter(sighting_end > hm("11:30"))

## 1. Remove videos from afternoon of cover day (i.e., videos starting after
## 11:30AM) 2. Adjust end time of videos from afternoon of cover day 3.
## Recalculate video length with updated end times 4. Make 'adjusted' video
## length the new video length
data03 <- data02 %>%
    filter(!(file_id %in% treatment_videos_pm$file_id)) %>%
    filter(!(sighting_id %in% treatment_sightings_pm$sighting_id)) %>%
    mutate(video_end_adjust = dplyr::if_else(file_id %in% treatment_videos_am_pm$file_id,
        hm("11:30"), video_end)) %>%
    mutate(video_length_adjust = as.numeric(video_end_adjust - video_start, "hours")) %>%
    select(row_id:video_length, video_end_adjust, video_length_adjust, everything()) %>%
    select(-video_length) %>%
    rename(video_length = video_length_adjust)
```

Remove videos from non-priority dates

```
data04 <- data03 %>%
    filter(video_priority == 1)
```

Remove cameras without data pre and post (because for the experiment, I'm interested in the pre-to-post change over time. No data from a given time period means that I cannot look at that change!)

```
cameras_pre_and_post <- data04 %>%
    distinct(year, patch, control_treatment, exp_phase, camera_id, date_video) %>%
    group_by(year, patch, control_treatment, exp_phase, camera_id) %>%
    summarise(num_dates = n()) %>%
    ungroup() %>%
    pivot_wider(names_from = exp_phase, values_from = num_dates, values_fill = 0) %>%
    filter(post > 0 & pre > 0)

data05 <- data04 %>%
    filter(camera_id %in% cameras_pre_and_post$camera_id)

data_to_summarize <- data05
```

Step 4: Summarize data

To build interaction networks, I first need an interaction rate between each hummingbird species and plant species. Here, the interaction rate will be the sighting rate, i.e., number of hummingbird sightings per hour. A sighting occurs when a hummingbird is in the camera frame. However, it's possible that a hummingbird could just be flying through, without any intention of visiting the plant. So, I will subset sightings to those in which the bird seems to drink from ("visit") the flower. For this reason, I also call refer to the "sighting rate" as the "visitation rate".

To calculate the rate, I will total the number of hummingbirds seen and then divide by the observation effort (number of video hours).

At this point it's also good to think ahead about the different datasets I'll need:

1. To understand how hummingbird visitation changes as result of our experimental manipulation (Heliconia removal), I'll need to summarize data at the level of EXPERIMENTAL PERIOD (pre vs. post) and REPLICATE (i.e., site + year combination). To explore how sampling method (camera observations vs. pollen samples) influences network metrics from individual networks, I can use this same dataset, but filtered to the 'pre' period only.

2. To visualize 'normal' interactions within the study system, I'll need to just look at unmanipulated data from the 'pre' period. Here, I am interested in the interactions across sites and years. This is the "meta-network" (network of networks) approach.

Get files

```
## Source the summarizing function
source("../scripts/helper_functions/Summarize_camera_data.R")

## Get scientific names for birds and plants (not just abbreviations). Will add
## after summarizing
bird_names <- read.csv("../data/import/Hummingbird_codes_species_list.csv") %>%
    select(bird_species = code, bird_scientific_name = scientific_name)

## Plant names. Will use this for camera data only; for pollen data, do not
## always know exact plant species associated with each pollen morphotype
plant_names <- read.csv("../data/import/Plant_codes_species_list.csv") %>%
    select(plant_species = species_code, plant_scientific_name = species_name) %>%
    filter(!is.na(plant_species) & plant_species != "")  #Get rid of blank rows
```

Summarize data

```
## Pre-post networks
sighting_rates_for_pp_networks = data_to_summarize %>%
    calculate_sighting_rates(data = ., level_org = "plant_species", level_time = "exp_phase",
        level_bird = "camera_spp_separate", sightings = "with_visit", include_unknown_spp = FALSE) %>%
    left_join(bird_names) %>%
    left_join(plant_names)

## For meta-network, first filter to data from 'pre' period, which are not
## affected by experiment. Could probably also include data 'post' period of
## control sites, but just looking at pre period for simplicity
sighting_rates_for_metanetwork = data_to_summarize %>%
    filter(exp_phase == "pre") %>%
    calculate_sighting_rates(data = ., level_org = "plant_species_across_sites",
        level_time = "all", level_bird = "camera_spp_separate", sightings = "with_visit",
        include_unknown_spp = FALSE) %>%
    left_join(bird_names) %>%
    left_join(plant_names)
```

Step 5: Export

```
## Camera data, filtered but not summarized
write.csv(data_to_summarize, "../data/export/intermediate/Camera_data_filtered_for_analysis.csv")

## Camera data, summarized for analyses using pre and post networks from
## individual replicates
write.csv(sighting_rates_for_pp_networks, "../data/export/for_analysis/Camera_data_summarized_for_pp_ne

## Camera data, summarized for visualization of meta-network
write.csv(sighting_rates_for_metanetwork, "../data/export/for_analysis/Camera_data_summarized_for_metane
```