

Ashy Storm-Petrel Catch-Per-Unit-Effort Estimation for Channel Islands National Park

Amelia J. DuVall & Emma Kelsey

8/27/2020

This document builds on the metadata file ("sessions.csv") that documents mist-netting efforts for Ashy Storm-Petrels for the Channel Islands National Park (CHIS) Seabird Monitoring Program Database. Several new fields are added to the data file, including * Apparent Sunset ("app_sunset") * Standard Ending 5.3 hours Post-Sunset ("std_ending") * Total Effort in Minutes ("min") * Total Effort in Minutes with Standard Ending ("min_std") * Number of ASSP Captures ("ASSP") * Number of ASSP Captures with Standard Ending ("ASSPstd") * Catch-Per-Unit-Effort ("CPUEraw") * Catch-Per-Unit-Effort with Standard Ending ("CPUEstd")

This is v.2020-09-08

Read-in data

```
sessions <- readRDS(here::here("Working", "sessions.RDS"))
captures <- readRDS(here::here("Working", "captures.RDS"))

# Check for duplicated session_IDs in sessions. This will affect joining data frames by session_ID.
any(duplicated(sessions$session_ID))

## [1] FALSE

# Check for duplicated catch_IDs in captures.
any(duplicated(captures$catch_ID))

## [1] FALSE
```

Add new fields

Apparent Sunset, Standard Ending, Minutes, Minutes (Standard)

```
# create dataframe to run sunset function on
sun_vec <- sessions %>%
  transmute(date = session_date, lat = lat, lon = lon, site_code = site_code, session_ID = session_ID)

# get apparent sunset times, calculate standard ending based on sunset and join back with CPUE database
cpue_v1 <- getSunlightTimes(data = sun_vec,
  keep = c("sunset"), tz = "PST8PDT") %>%
  mutate(std_ending = sunset + lubridate::hours(5) + lubridate::minutes(18)) %>% # standard ending = 5.3 hours
  rename(app_sunset = sunset) %>%
  left_join(sun_vec, by = c("date", "lat", "lon")) %>%
  dplyr::select(-date, -lat, -lon) %>%
  left_join(sessions, by = c("session_ID", "site_code")) %>%
  # calculate total number of minutes net was open, as well as total number of minutes before standard
```

```

mutate(min_1 = net_close_1 - net_open_1,
       min_2 = net_close_2 - net_open_2,
       min_3 = net_close_3 - net_open_3,
       min_4 = net_close_4 - net_open_4,
       min_5 = net_close_5 - net_open_5) %>%
mutate_at(c("min_1", "min_2", "min_3", "min_4", "min_5"), .funs = ~replace_na(., 0)) %>%
mutate(net_close_1_std = if_else(net_close_1 > std_ending, std_ending, net_close_1),
       net_close_2_std = if_else(net_close_2 > std_ending, std_ending, net_close_2),
       net_close_3_std = if_else(net_close_3 > std_ending, std_ending, net_close_3),
       net_close_4_std = if_else(net_close_4 > std_ending, std_ending, net_close_4),
       net_close_5_std = if_else(net_close_5 > std_ending, std_ending, net_close_5),
       min_1_std = net_close_1_std - net_open_1,
       min_2_std = net_close_2_std - net_open_2,
       min_3_std = net_close_3_std - net_open_3,
       min_4_std = net_close_4_std - net_open_4,
       min_5_std = net_close_5_std - net_open_5) %>%
mutate_at(c("min_1_std", "min_2_std", "min_3_std", "min_4_std", "min_5_std"), .funs = ~replace(., .<0, 0)) %>%
mutate_at(c("min_1_std", "min_2_std", "min_3_std", "min_4_std", "min_5_std"), .funs = ~replace_na(., 0)) %>%
mutate(min = as.numeric(min_1 + min_2 + min_3 + min_4 + min_5),
       min_std = min_1_std + min_2_std + min_3_std + min_4_std + min_5_std,
       min = as.double(if_else(min < 0, "0", as.character(min))),
       min = as.double(if_else(as.character(net_open_1) == "NA", "0", as.character(min))),
       min_std = as.double(if_else(min_std < 0, "0", as.character(min_std))),
       min_std = as.double(if_else(as.character(net_open_1) == "NA", "0", as.character(min_std)))) %>%
select(-min_1:-min_5_std) %>%
arrange(session_year) %>%
filter(TRUE)

```

ASSP, ASSP, CPUEraw, CPUEstd

```

catches_std <- captures %>%
  dplyr::select(-c("flagged", "notes")) %>% # remove these fields since they exist in sessions w/ diff
  left_join(cpue_v1, by = c("site_code", "session_ID", "lat", "long", "session_date", "session_month", "session_year")) %>%
  mutate(std = if_else(std_ending > capture_date, "1", "0"), # if bird was caught before std_ending = 1
         SNR = mosaic::derivedFactor(
           "Y" = (recapture == "SNR"), # identify same night recaptures to be removed from CPUE
           "N" = (recapture == "N" | recapture == "Y"), # recaps from other nights still count
           .default = NA),
         catchPastSS = capture_date - app_sunset) %>% # catchPastSS = capture time (min) past sunset
  filter(TRUE)

# sum catches for each species and night
cpue <- catches_std %>%
  filter(species == "ASSP") %>% # filters to 3861 observations
  group_by(session_ID) %>%
  summarise(ASSP = n(),
            ASSPstd = sum(std == "1")) %>%
  ungroup() %>%
  right_join(cpue_v1, by = c("session_ID")) %>%
  mutate(CPUEraw = ASSP/min,
         CPUEstd = ASSPstd/min_std) %>%
  dplyr::select("session_ID", "island_code", "subisland_code", "site_code", "site_name", "lat", "lon", "long", "session_year", "session_month", "session_day", "series_ID", "app_sunset", "std")

```

```

      "net_close_1", "net_open_2", "net_close_2", "net_open_3", "net_close_3", "net_op
      "net_open_5", "net_close_5", "min", "min_std", "ASSP", "ASSPstd", "CPUEraw", "CP
      "net_dim", "spp_audio_file", "dB_level", "speaker_system", "org", "notes", "flagge

## 'summarise()' ungrouping output (override with '.groups' argument)
# Check new fields
chk <- cpue %>%
  filter(is.na(ASSP))

sIDs <- unique(chk$session_ID)

chkcaps <- captures %>%
  filter(session_ID %in% c(sIDs))
## There were no ASSP captures during these sessions

```

Export out

```

# Exporting as RDS will preserve date/time fields
saveRDS(cpue, here("Working", "cpue.RDS"))
write.csv(cpue, here("Working", "cpue.csv"), row.names = FALSE)

```