

CHIS ASSP Mistnetting Database QA/QC

Amelia DuVall (ajduvall@uw.edu) & Emma Kelsey (ekelsey@usgs.gov)

24 August 2020

This is v.2020-08-24

This document explains the QA/QC conducted on the Channel Islands National Park (CHIS) Ashy Storm-Petrel (ASSP) Mistnetting database (1994 - 2018). The database is an Excel file that contains (6) sheets: Banding, Banding_Data_Dictionary, CPUE, CPUE_Data_Dictionary, Mistnetting_Locations, Participant_Initials. The purpose of this exercise is to capture incongruent values in the database due errors in original data entry or bugs from data compilation and to improve the functioning of this document as a long-term database. We cannot verify observer errors (e.g., if errors were made while taking morphometric measurements and recorded as such). Whenever possible, we cross-referenced raw data (e.g., field notebooks or scanned data sheets) to verify and/or fix incongruent values. This document is divided into two categories: Banding QAQC and CPUE QAQC. Amelia DuVall conducted QAQC on the Banding data and Emma Kelsey conducted QAQC on the CPUE metadata.

Load libraries

```
library(tidyverse)
library(lubridate)
library(ggplot2)
library(readxl)
library(here)
```

Read-in data and set-up for analysis

```
captures.raw <- read.csv(here("Working", "captures.csv"), na.strings = c("NA", "ND"))
cpue.raw <- read.csv(here("Working", "cpue.csv"), na.strings = c("NA", "ND"))

# Fix day/times
captures <- captures.raw %>%
  mutate(session_date = as_date(session_date, format = "%m/%d/%Y")) %>%
  mutate(capture_date = as_datetime(capture_date, format = "%m/%d/%Y %H:%M")) %>%
  mutate(release_date = as_datetime(release_date, format = "%m/%d/%Y %H:%M")) %>%
  dplyr::select(-1)

cpue <- cpue.raw %>%
  mutate(session_date = as_date(session_date, format = "%Y-%m-%d")) %>%
  mutate(app_sunset = as_datetime(app_sunset, format = "%m/%d/%Y %H:%M")) %>%
  mutate(std_ending = as_datetime(std_ending, format = "%m/%d/%Y %H:%M")) %>%
  mutate_at((c("net_open_1", "net_open_2", "net_open_3", "net_open_4", "net_open_5",
               "net_close_1", "net_close_2", "net_close_3", "net_close_4", "net_close_5")),
            ~ as_datetime(., format = "%m/%d/%Y %H:%M")) %>%
  dplyr::select(-1)
```

Banding Data QAQC

Filter data to ASSP species and banded individuals only.

```
ASSP <- group_by(.data = captures) %>%  
  filter(species == "ASSP") %>%  
  filter(band_no != "notbanded") %>%  
  ungroup()
```

Summarize band numbers and capture rates.

Summarize data by band number and determine capture rate for each band number (e.g., each individual).

```
summary <- group_by(.data = ASSP, band_no) %>%  
  summarise(no_captures = n()) %>%  
  ungroup()
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

There are 3645 unique band numbers.

Summarize capture rates.

```
summarycaptures <- group_by(summary, no_captures) %>%  
  summarise(count = n()) %>%  
  ungroup()
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
show(summarycaptures)
```

```
## # A tibble: 4 x 2  
##   no_captures count  
##       <int> <int>  
## 1         1  3471  
## 2         2   155  
## 3         3    16  
## 4         4     3
```

Summarize recapture rates.

```
recap <- group_by(.data = ASSP, recapture) %>%  
  summarise(no_captures = n()) %>%  
  ungroup()
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
show(recap)
```

```
## # A tibble: 3 x 2  
##   recapture no_captures  
##   <chr>       <int>  
## 1 N           3604  
## 2 SNR           42  
## 3 Y           195
```

The recapture rates will not necessarily match the unique band numbers because we encountered some individuals only once as a recapture (i.e., we did not band them). These bands need to be cross-referenced against BBL data to determine when/where they were first banded. This database does not contain all ASSP mistnetting banding records from CHIS.

Another way to check the recapture field is to sort the band numbers sequentially and look for outliers. In theory, these would be on someone else's banding permit (and a recapture).

```
seq <- arrange(.data = ASSP, band_no)
```

We visually inspected these data and flagged any bands that were out of order to cross-reference against raw data.

Capture time

```
# Which unique hour values are represented in the data?
unique(ASSP$capture_hour)

## [1] 21 22 2 23 0 1 3 4 20

unique(captures$capture_hour)

## [1] 21 22 2 23 0 1 3 20 4

# Isolate questionable data.
cap_hour_chk <- group_by(.data = ASSP, capture_hour) %>%
  filter(capture_hour > 2 & capture_hour < 20) %>%
  ungroup()

# Export to csv and cross-reference raw data.
#write.csv(cap_hour_chk, "captimeQAQC.csv")
```

Release time

```
# Isolate hour as new field.
ASSP$rel_hour <- hour(ASSP$release_date)

# Which unique hour values are represented in the data?
unique(ASSP$rel_hour)

## [1] 21 22 2 23 NA 0 1 3 20 4

# There are NA's because the release time was often unrecorded.

#Isolate questionable data
rel_hour_chk <- group_by(.data = ASSP, rel_hour) %>%
  filter(rel_hour > 2 & rel_hour < 20) %>%
  ungroup()

# Export to csv and cross-reference raw data.
#write.csv(rel_hour_chk, "reltimeQAQC.csv")
```

Brood patch

```
# What are the unique BP values represented in the data?
unique(ASSP$BP)

## [1] "4.5" "1" "1.5" "3" "5" "2" "4" NA "0" "B" "PD" "D"

# All these values fall within the range of values included in data dictionary.
```

Mass

There are three mass values recorded in the database: mass (uncorrected), mass (tare), and mass (corrected). The mass (corrected) field is the remainder of mass (uncorrected) - mass (tare).

```
unique(ASSP$mass_corr)
```

```
## [1] 38 30 37 39 33 34 40 23 32 35 NA 36 31 42 41 43 62 60 58 61 26 44 17 29 48  
## [26] 27 28 45 46 22
```

```
summary(ASSP$mass_corr)
```

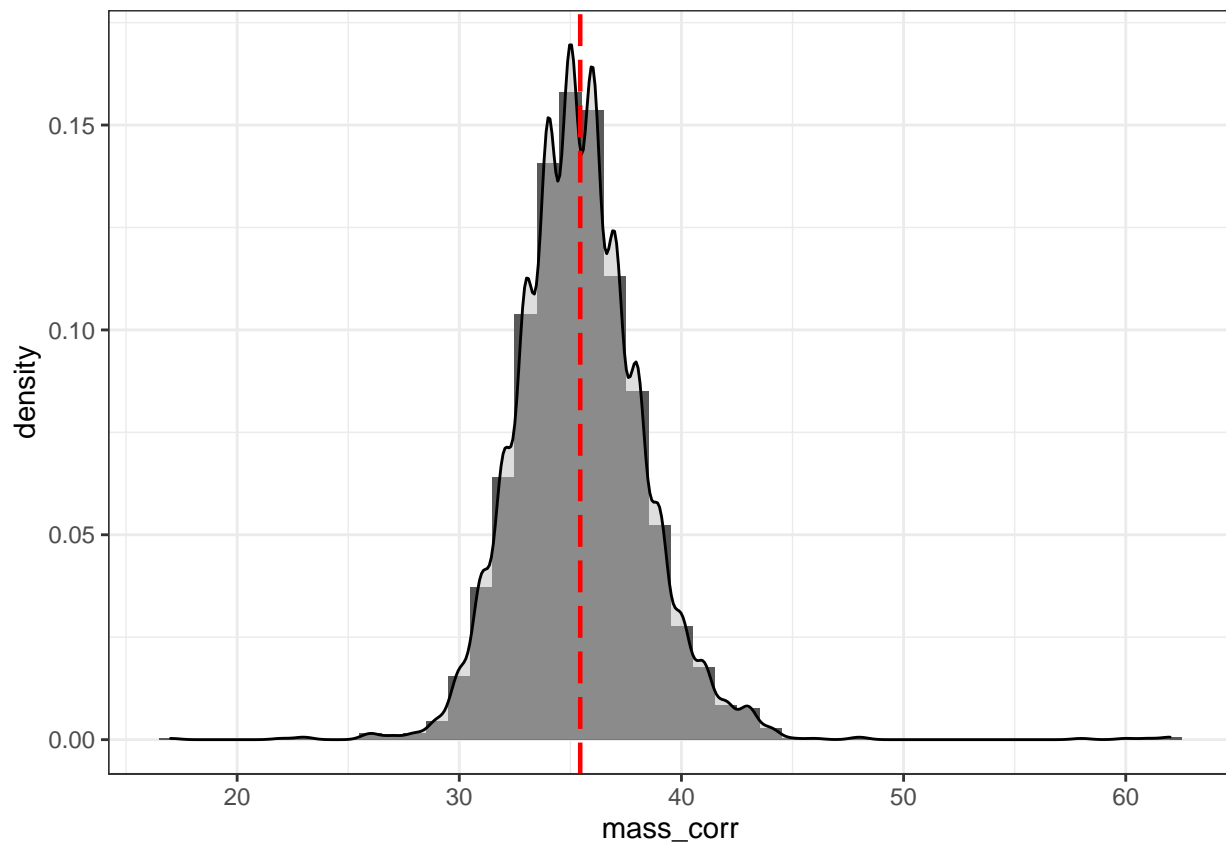
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's  
##    17.00   34.00   35.00   35.44   37.00   62.00     491
```

```
# Histogram
```

```
ggplot(data = ASSP, aes(x = mass_corr), na.rm = TRUE) +  
  geom_histogram(aes(y = ..density..), binwidth = 1) +  
  geom_density(alpha = .5, fill = "gray") +  
  geom_vline(aes(xintercept = mean(mass_corr, na.rm = TRUE)),  
             colour = "red", linetype = "longdash", size = .8) +  
  theme_bw()
```

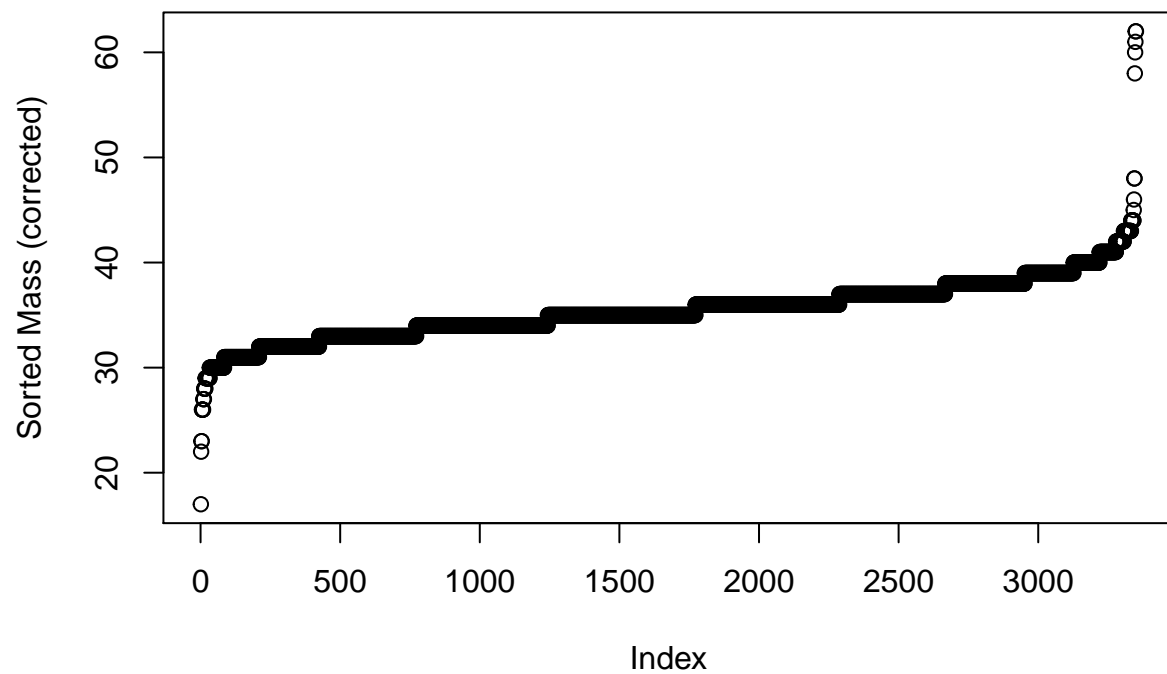
```
## Warning: Removed 491 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 491 rows containing non-finite values (stat_density).
```



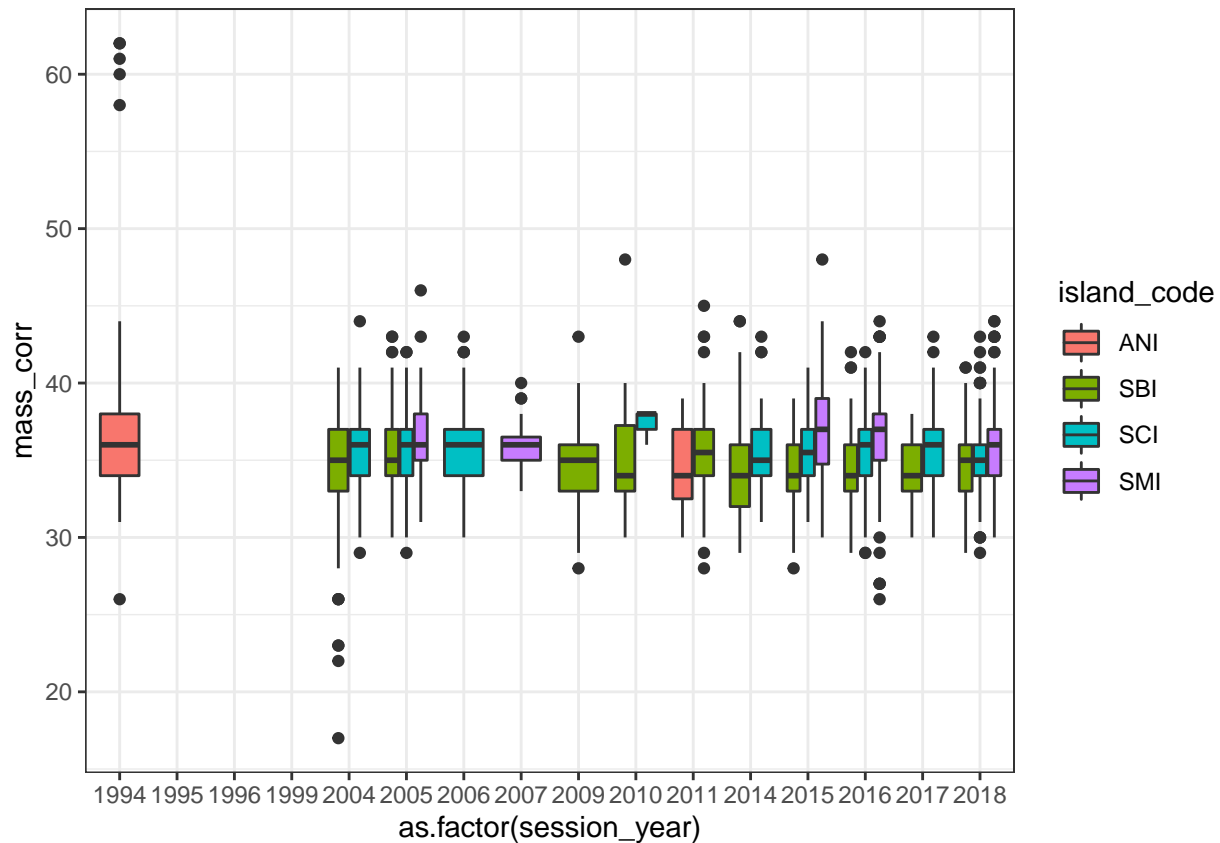
```
# Sorted values plot
```

```
plot(sort(ASSP$mass_corr), ylab = "Sorted Mass (corrected)")
```



```
# Boxplot of values
ggplot(ASSP, aes(x = as.factor(session_year), y = mass_corr, fill = island_code)) +
  geom_boxplot() +
  theme_bw()
```

```
## Warning: Removed 491 rows containing non-finite values (stat_boxplot).
```



```
# Mass (g) values reported in Adams (2016) paper: 36.1 +/- 2.8 (female) and 34.7 +/- 2.1 (male).
# Isolate questionable data.
mass_corr_chk <- filter(ASSP, mass_corr < 25 | mass_corr > 50)

# Export to csv and cross-reference raw data.
#write.csv(mass_corr_chk, "masscorrQAQC.csv")
```

Culmen

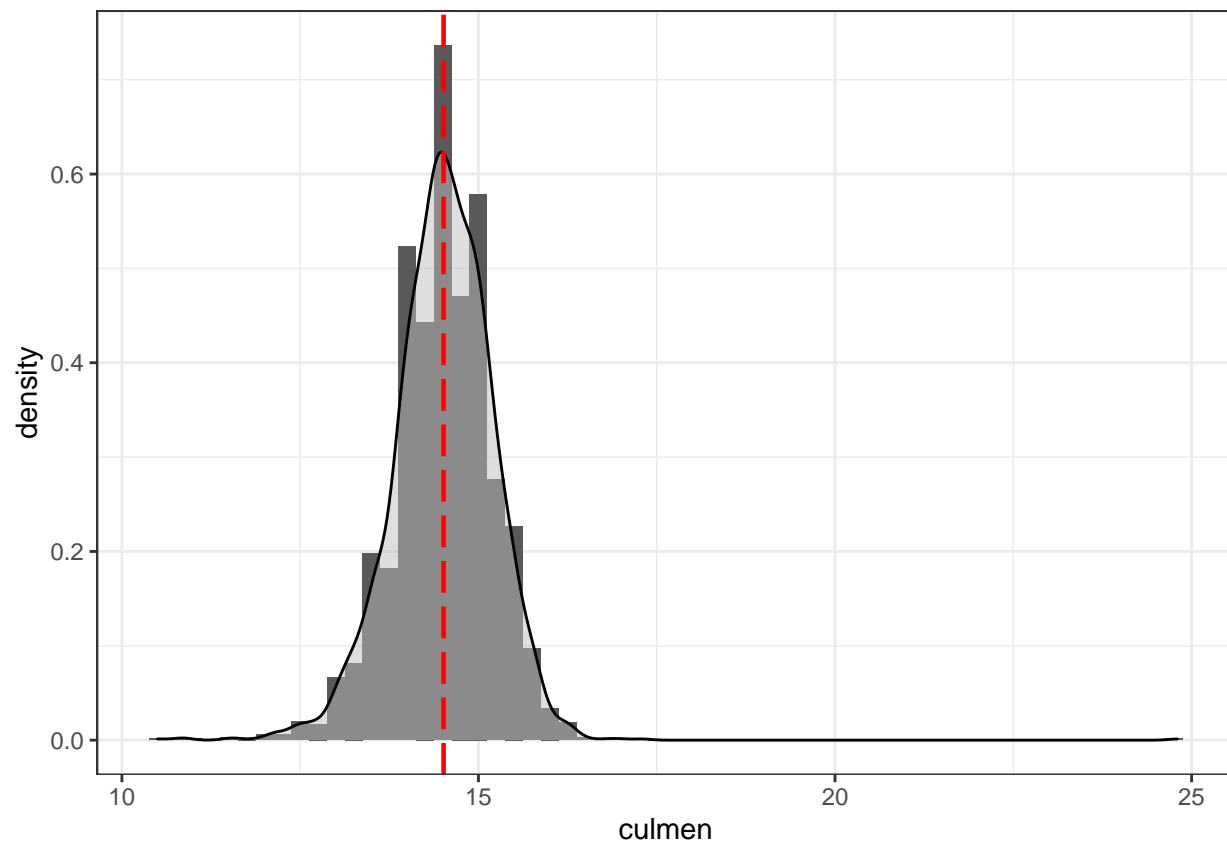
```
summary(ASSP$culmen)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      10.50   14.10   14.50   14.51   14.99   24.80    1250
```

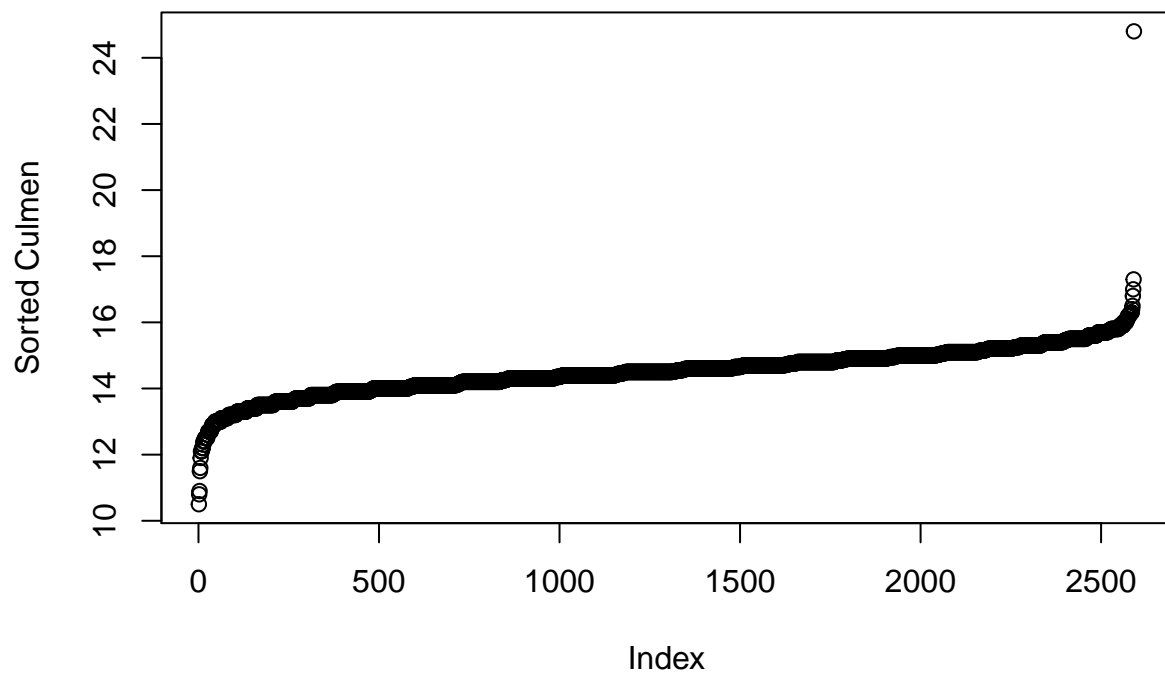
```
# Histogram
ggplot(data = ASSP, aes(x = culmen, na.rm = TRUE)) +
  geom_histogram(aes(y = ..density..), binwidth = .25) +
  geom_density(alpha = .5, fill = "gray") +
  geom_vline(aes(xintercept = mean(culmen, na.rm = TRUE)),
             colour = "red", linetype = "longdash", size = .8) +
  theme_bw()
```

```
## Warning: Removed 1250 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 1250 rows containing non-finite values (stat_density).
```

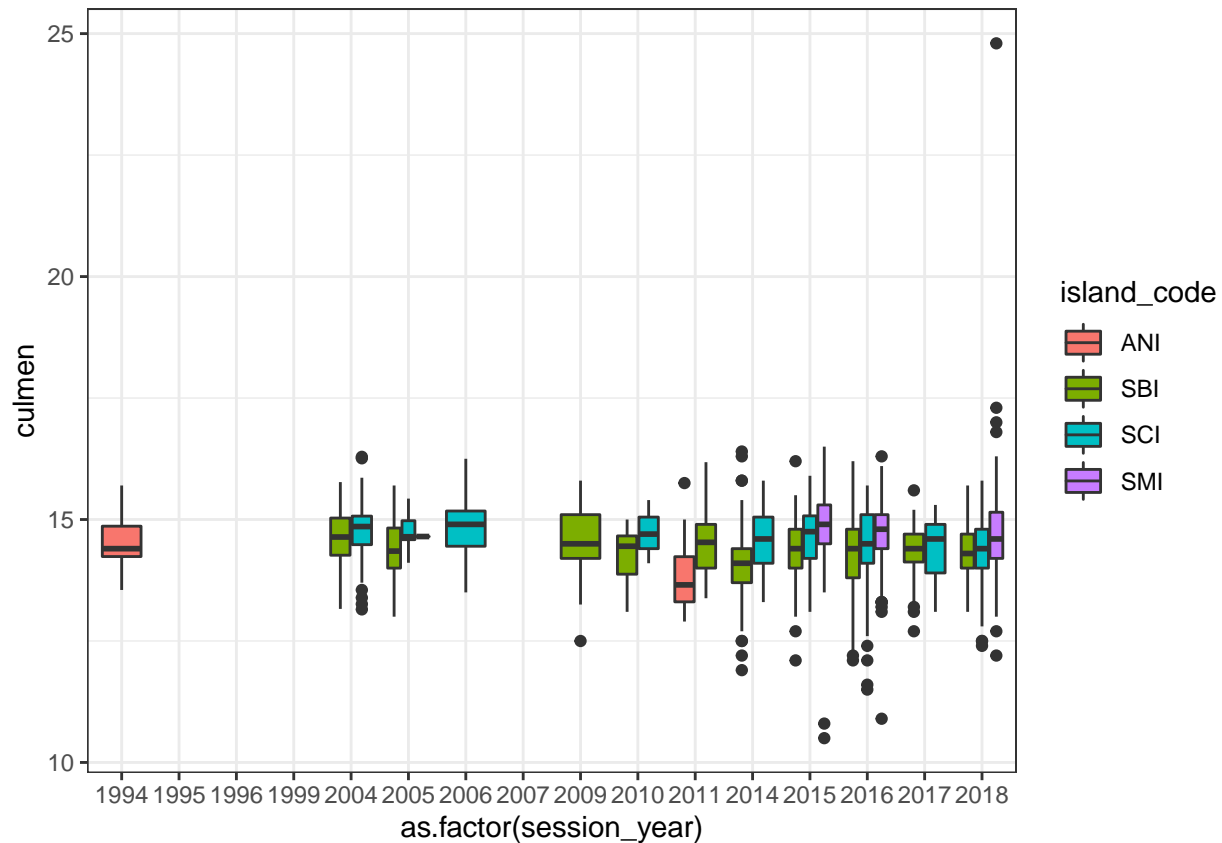


```
# Sorted values plot  
plot(sort(ASSP$culmen), ylab = "Sorted Culmen")
```



```
# Boxplot of values  
ggplot(ASSP, aes(x = as.factor(session_year), y = culmen, fill = island_code)) +  
  geom_boxplot() +  
  theme_bw()
```

```
## Warning: Removed 1250 rows containing non-finite values (stat_boxplot).
```

*# Bill length (mm) values reported in Adams (2016) paper: 14.9 +/- 0.5 (f) and 14.6 +/- 0.8 (m).
Values reported in Pyle guide: 13.1-15.2 (95% CI).*

```
#Isolate questionable data
culmen_chk <- filter(ASSP, culmen < 13 | culmen > 16)

# Export to csv and cross-reference raw data.
#write.csv(culmen_chk, "culmenQAQC.csv")
# Note: The filter range (13-16) might be too narrow. Most were not typos.
```

Skull length

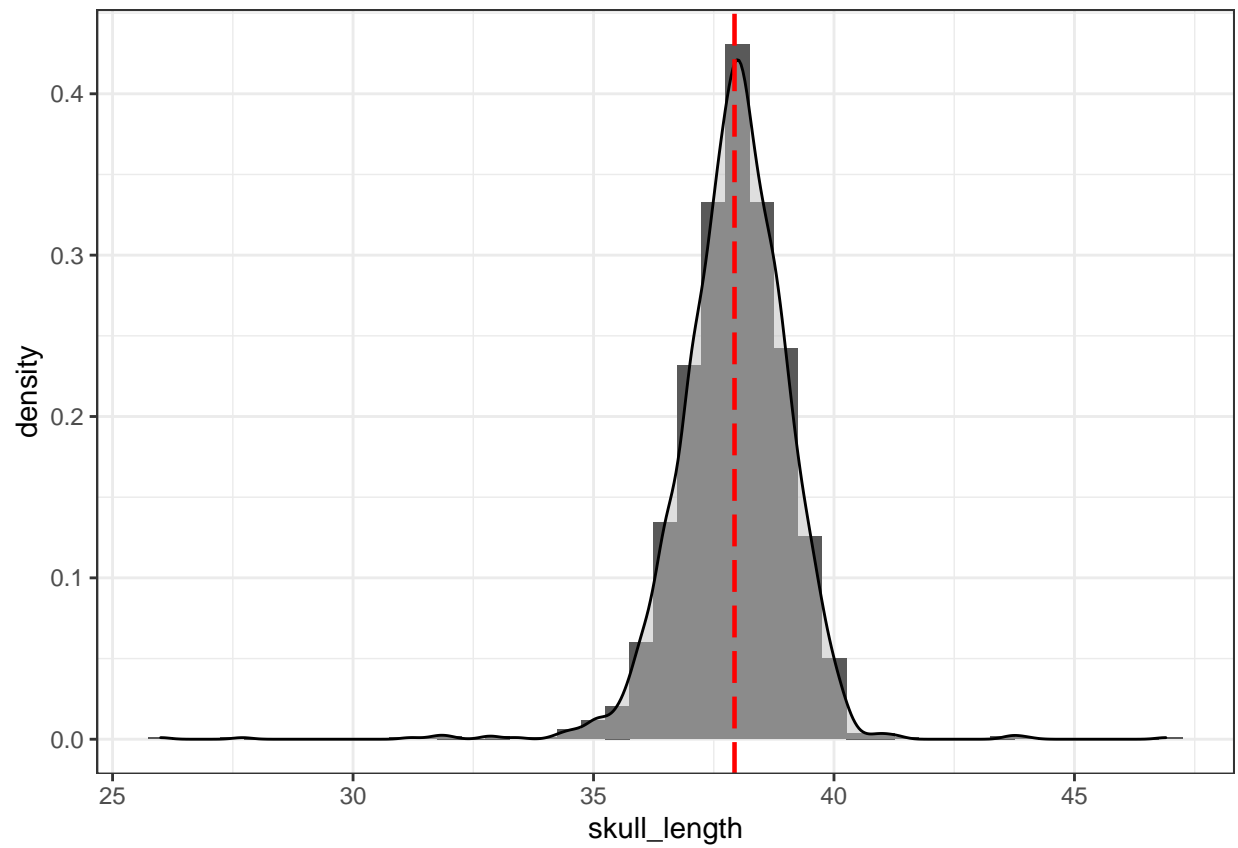
```
summary(ASSP$skull_length)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      26.00   37.30   38.00   37.93   38.65   46.90    1802
```

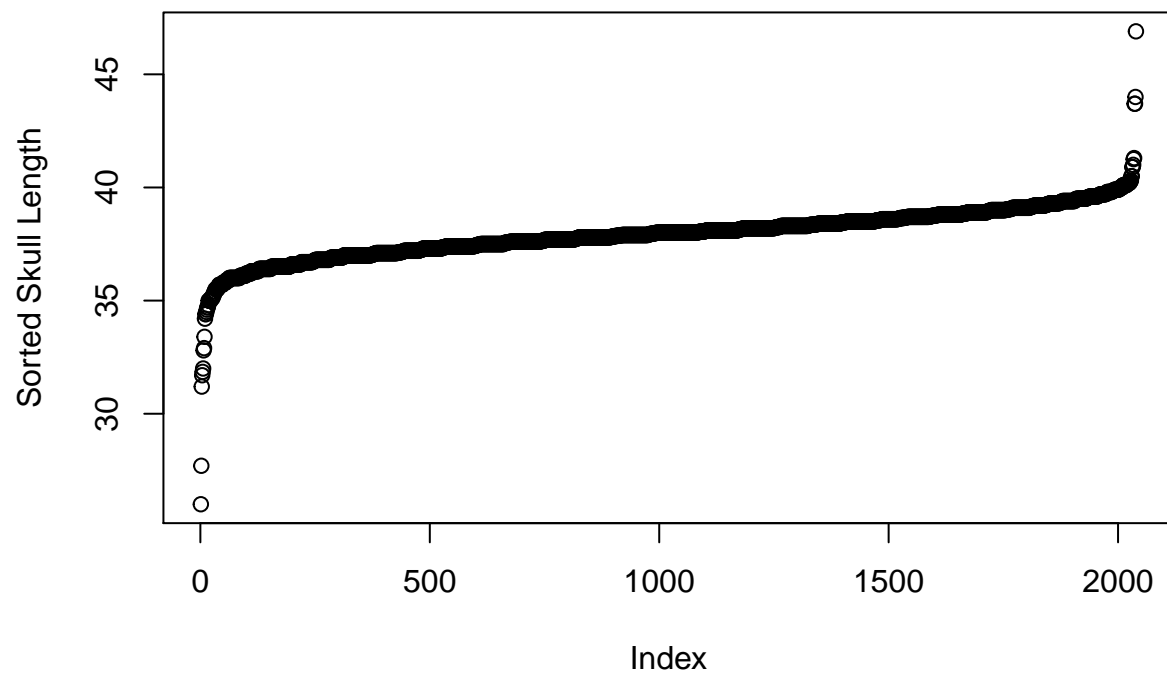
```
# Histogram
ggplot(data = ASSP, aes(x = skull_length, na.rm = TRUE)) +
  geom_histogram(aes(y = ..density..), binwidth = .5) +
  geom_density(alpha = .5, fill = "gray") +
  geom_vline(aes(xintercept = mean(skull_length, na.rm = TRUE)),
             colour = "red", linetype="longdash", size = .8) +
  theme_bw()
```

```
## Warning: Removed 1802 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 1802 rows containing non-finite values (stat_density).
```

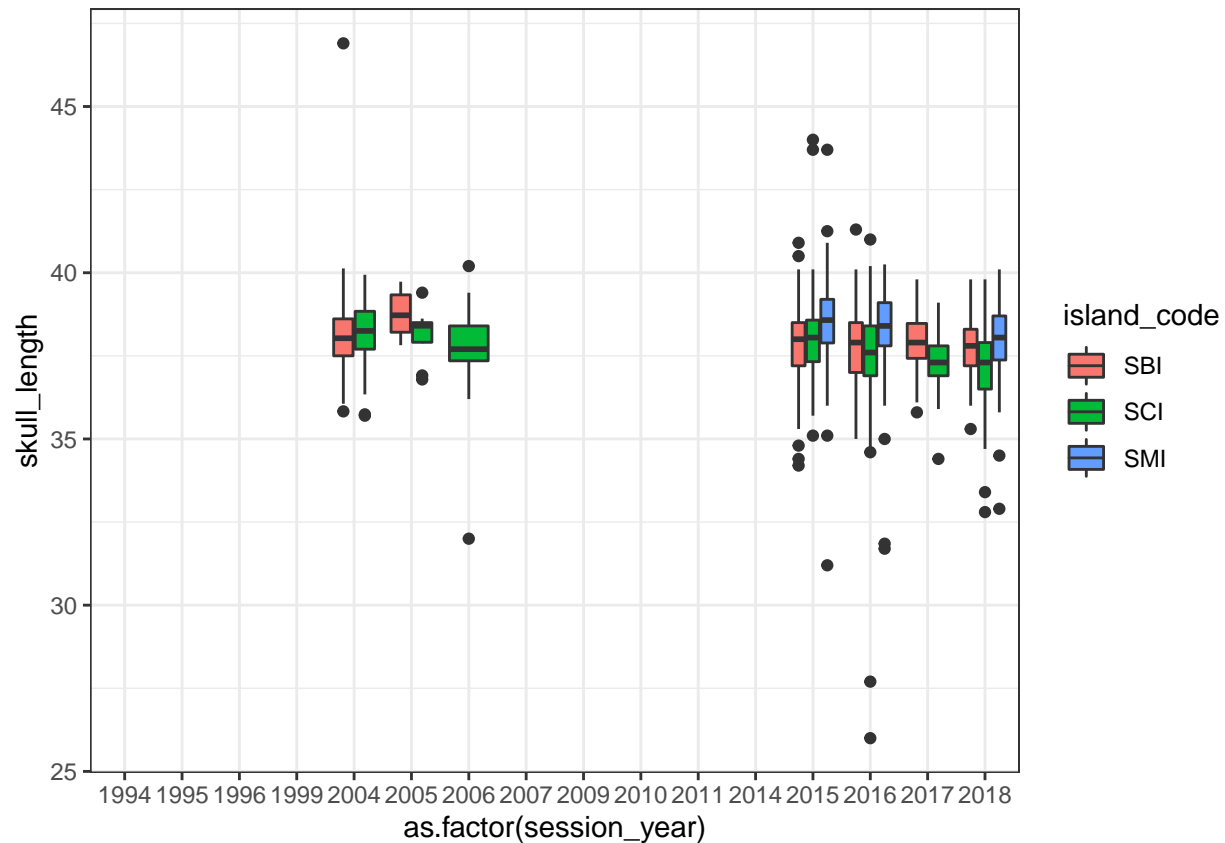


```
# Sorted values plot  
plot(sort(ASSP$skull_length), ylab = "Sorted Skull Length")
```



```
# Boxplot of values  
ggplot(ASSP, aes(x = as.factor(session_year), y = skull_length, fill = island_code)) +  
  geom_boxplot() +  
  theme_bw()
```

```
## Warning: Removed 1802 rows containing non-finite values (stat_boxplot).
```



*# Skull length (mm) values reported in Adams (2016): 38.1 +/- 1.1 (f) and 37.9 +/- 0.8 (m).
No information on skull length in Pyle guide.*

Isolate questionable data.

```
skull_chk <- filter(ASSP, skull_length < 35 | skull_length > 41)
```

Export to csv and cross-reference raw data.

```
#write.csv(skull_chk, "skullQAQC.csv")
```

Tarsus

```
summary(ASSP$tarsus)
```

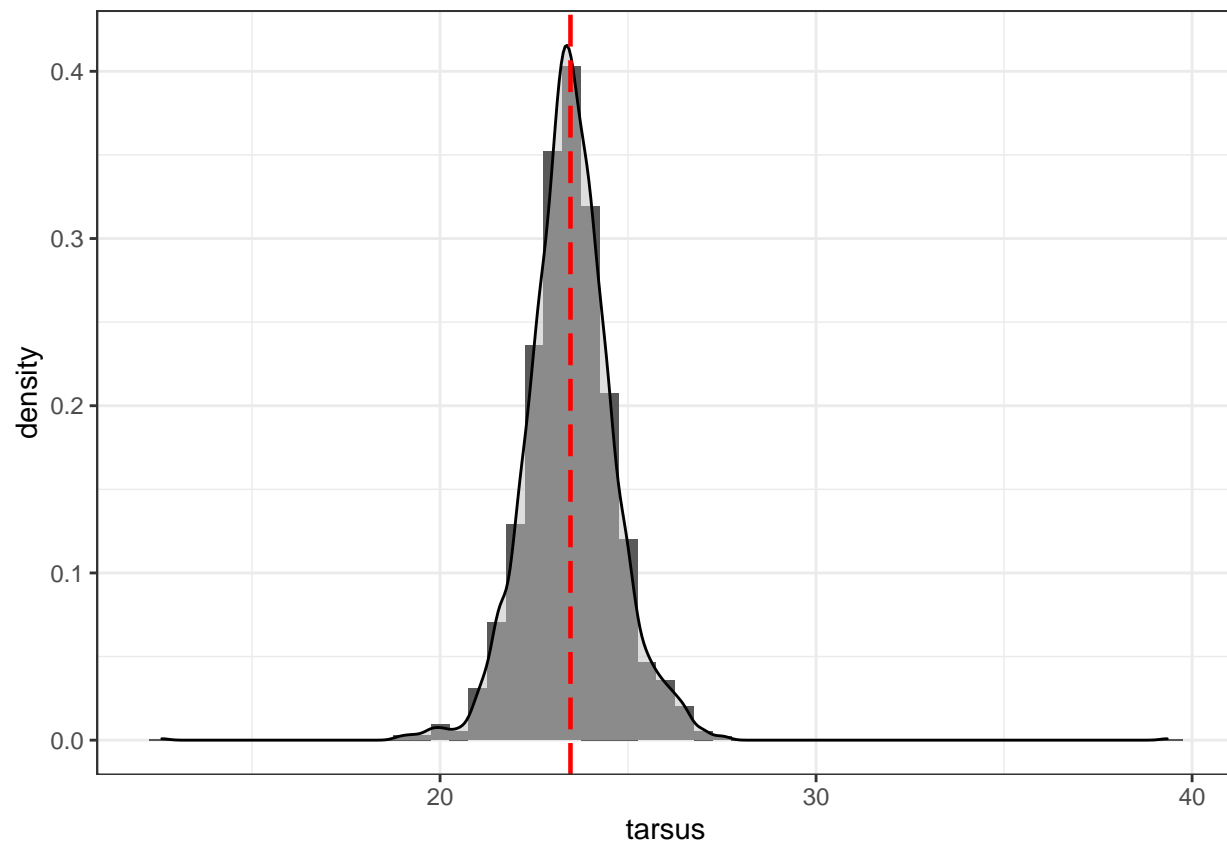
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##    12.60  22.80   23.45   23.47  24.10   39.34   1249
```

Histogram

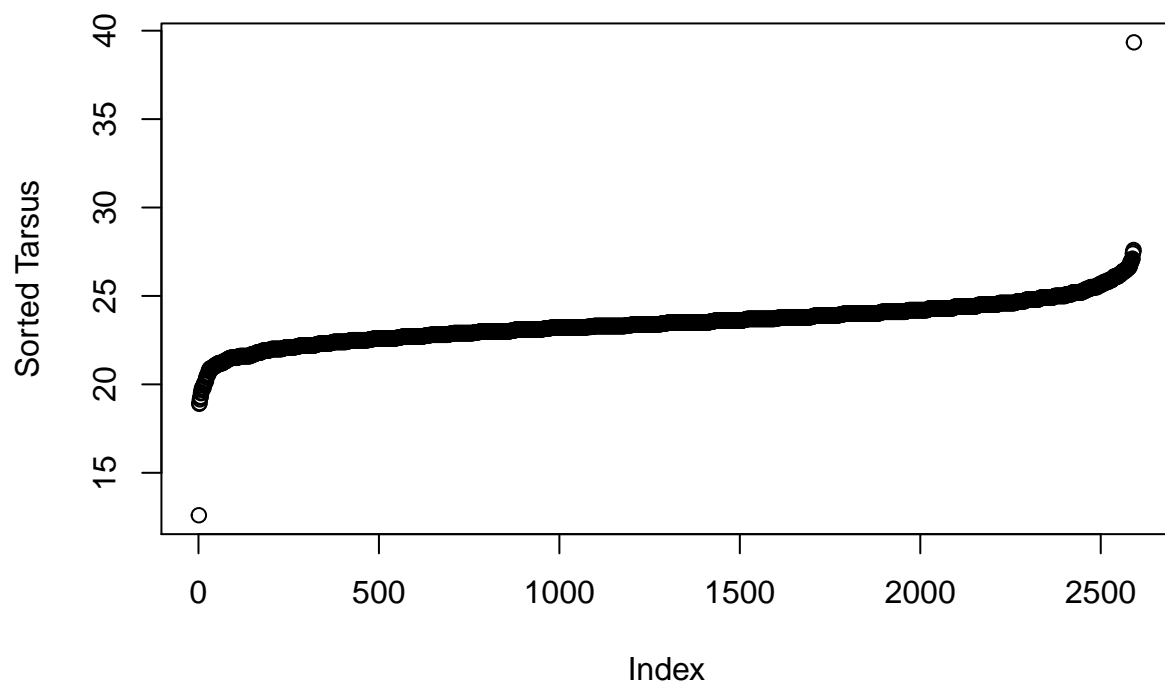
```
ggplot(data = ASSP, aes(x = tarsus, na.rm = TRUE)) +
  geom_histogram(aes(y = ..density..), binwidth = .5) +
  geom_density(alpha = .5, fill = "gray") +
  geom_vline(aes(xintercept = mean(tarsus, na.rm = TRUE)),
             colour = "red", linetype = "longdash", size = .8) +
  theme_bw()
```

```
## Warning: Removed 1249 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 1249 rows containing non-finite values (stat_density).
```

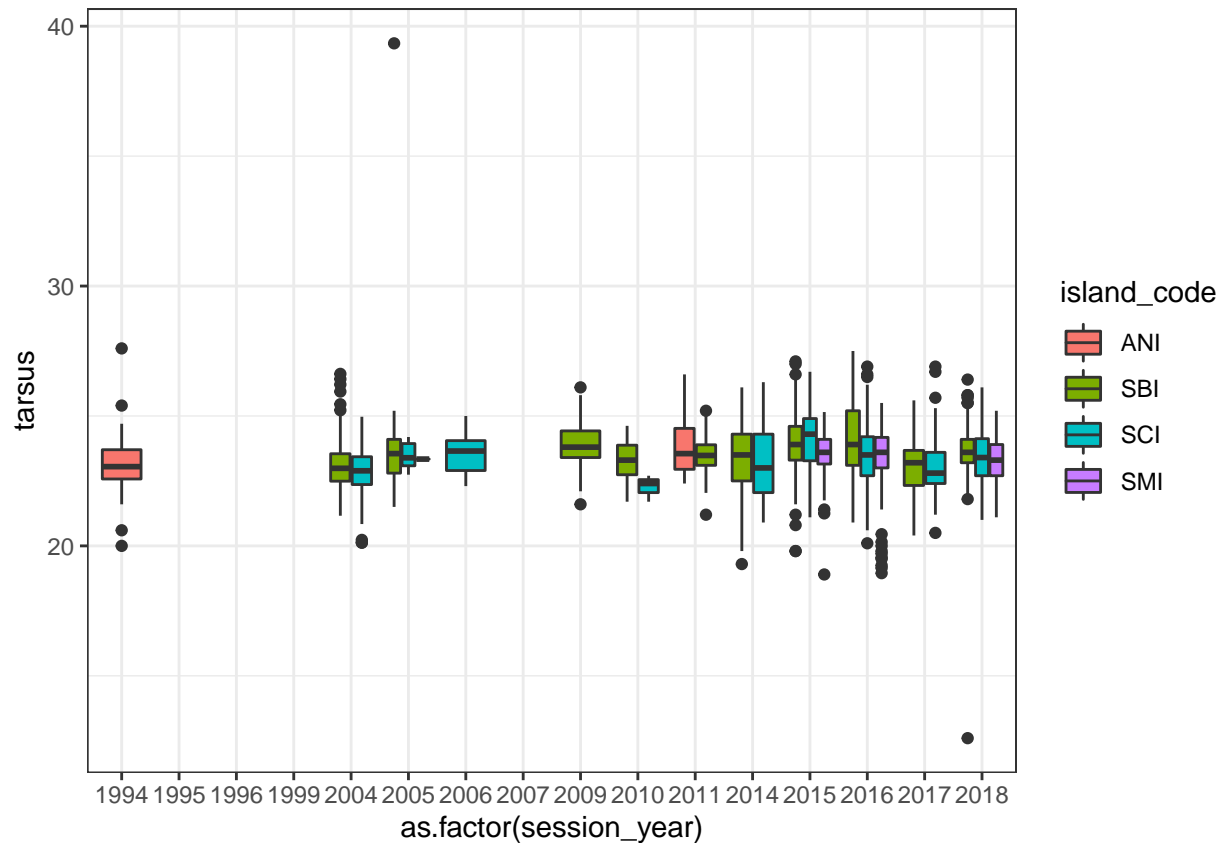


```
# Sorted values plot  
plot(sort(ASSP$tarsus), ylab = "Sorted Tarsus")
```



```
# Boxplot of values  
ggplot(ASSP, aes(x = as.factor(session_year), y = tarsus, fill = island_code)) +  
  geom_boxplot() +  
  theme_bw()
```

```
## Warning: Removed 1249 rows containing non-finite values (stat_boxplot).
```



Tarsus (mm) values reported in Adams (2016): 23.2 +/- 0.9 (f) and 23.1 +/- 0.8 (m).
Values reported in Pyle guide: 21-25 (95% CI).

Isolate questionable data.

```
tarsus_chk <- filter(ASSP, tarsus < 19 | tarsus > 27)
```

Export to csv and cross-reference raw data.

```
#write.csv(tarsus_chk, "tarsusQAQC.csv")
```

Wing chord

```
summary(ASSP$wing)
```

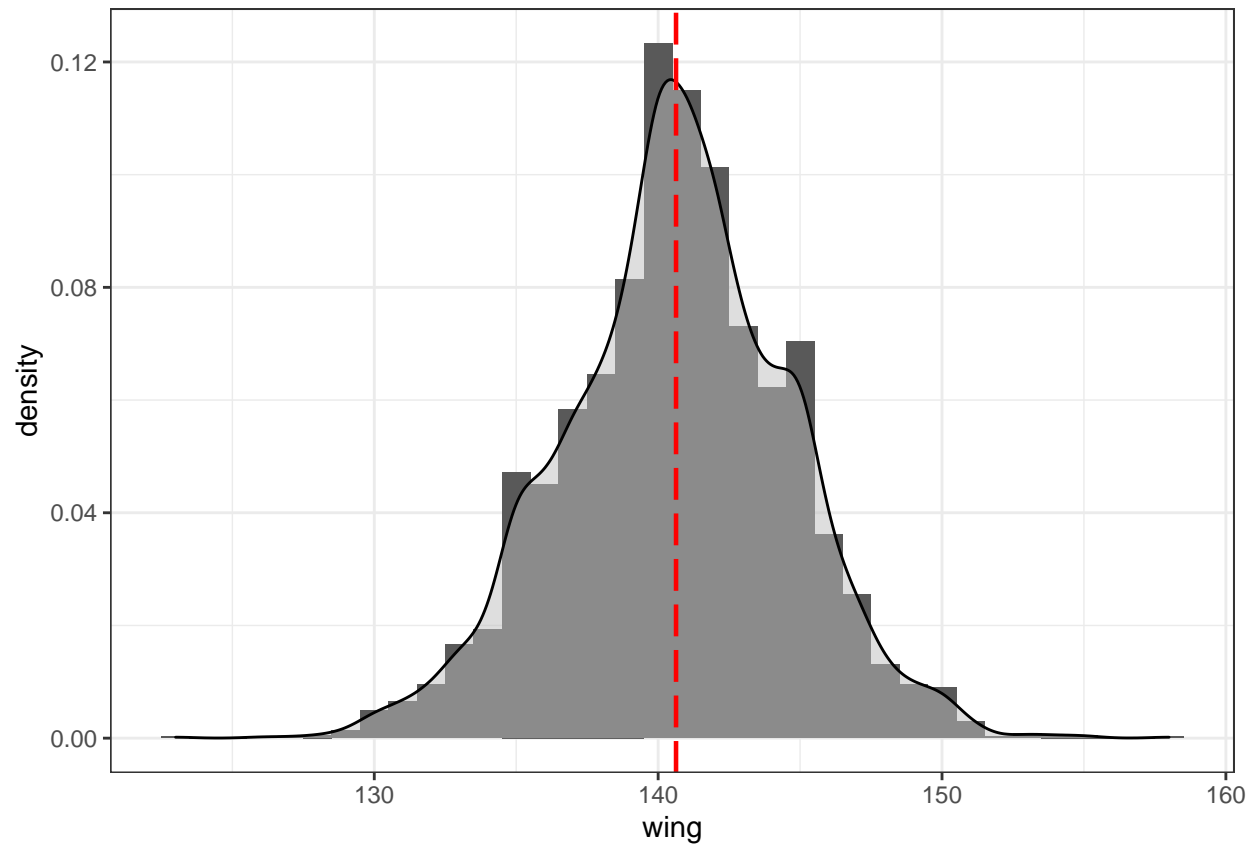
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##      123.0   138.0   141.0   140.6   143.0   158.0    159
```

Histogram

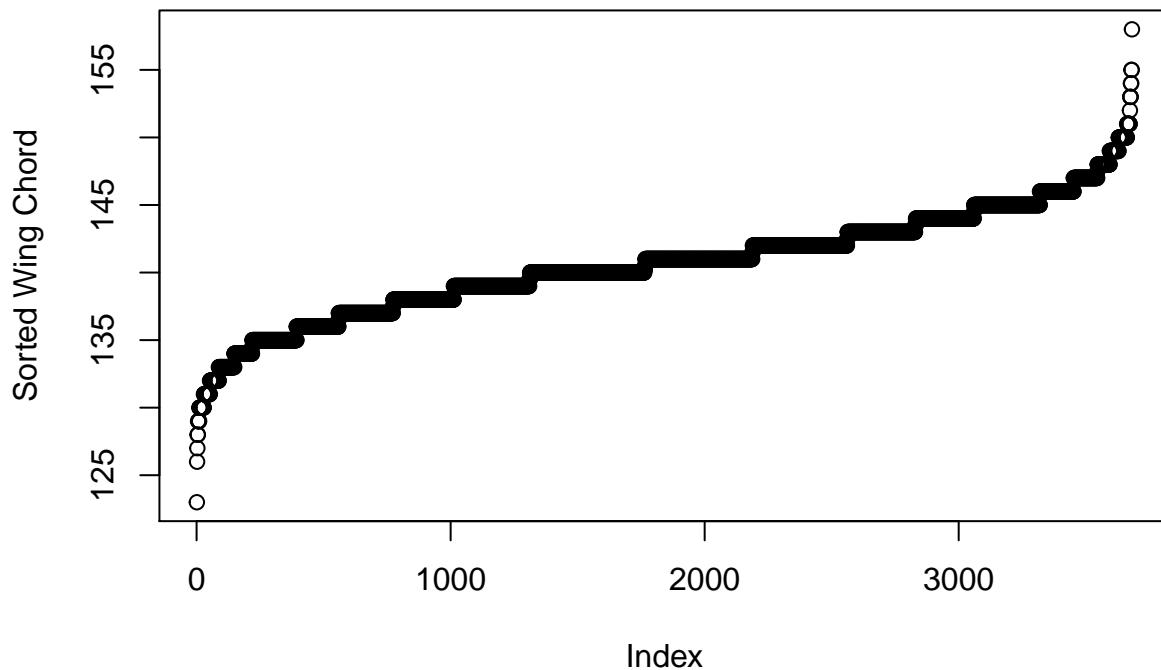
```
ggplot(data = ASSP, aes(x = wing, na.rm = TRUE)) +
  geom_histogram(aes(y = ..density..), binwidth = 1) +
  geom_density(alpha = .5, fill = "gray") +
  geom_vline(aes(xintercept = mean(wing, na.rm = TRUE)),
             colour = "red", linetype = "longdash", size = .8) +
  theme_bw()
```

```
## Warning: Removed 159 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 159 rows containing non-finite values (stat_density).
```



```
# Sorted values plot  
plot(sort(ASSP$wing), ylab = "Sorted Wing Chord")
```

```
# Max flat wing (mm) values reported in Adams (2016): 142.7 +/- 2.8 (f) and 140.4 +/- 3.3 (m).
# Values reported in Pyle guide: 132-148 (95% CI).
```

```
# Isolate questionable data.
wing_chk <- filter(ASSP, wing < 130 | wing > 150)

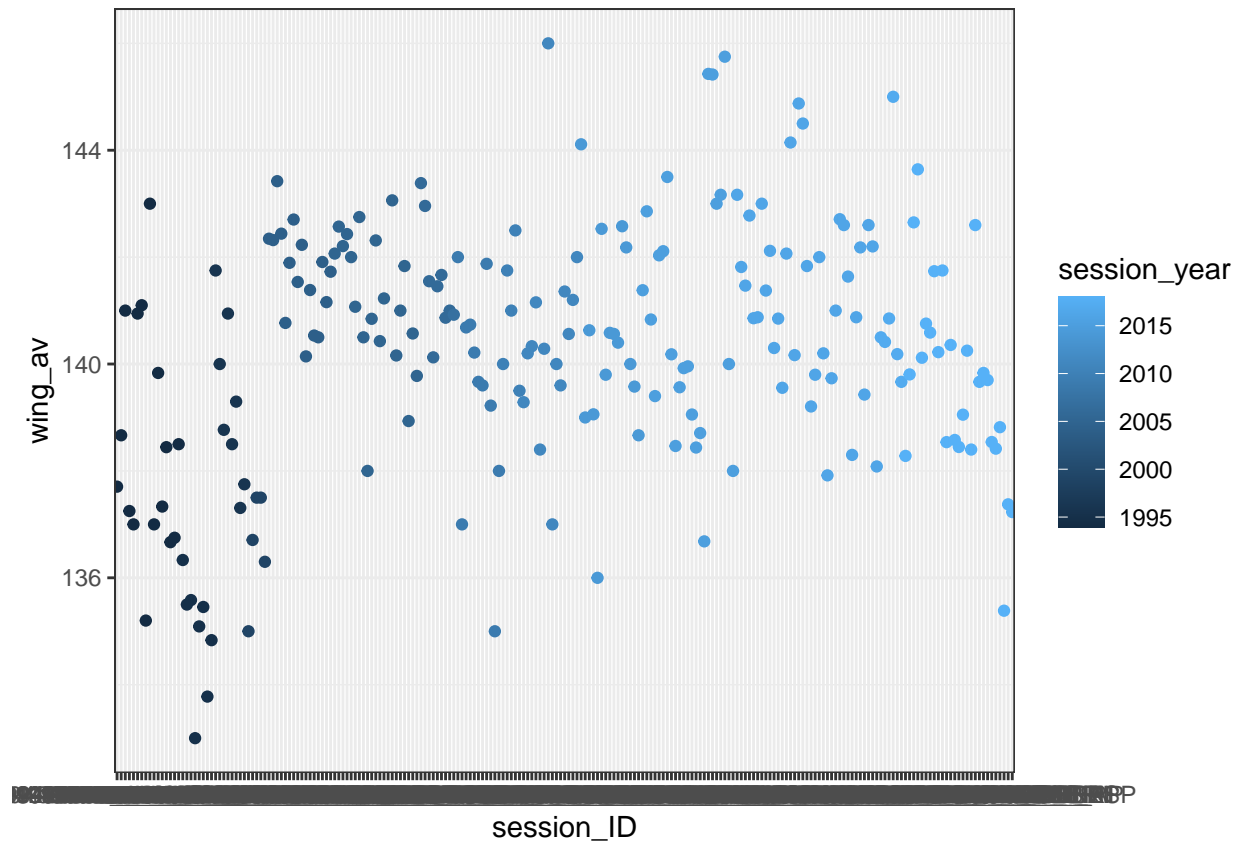
# Export to csv and cross-reference raw data.
# write.csv(wing_chk, "wingQAQC.csv")
```

We decided to check for a bimodal distribution of the wing chord morphometric data once the data entry errors had been rectified. A bimodal distribution could be an indication of different methods used to measure wing chord (e.g, flattened wing chord versus relaxed wing chord).

```
# How about mean wing chord observed per session?
WC_ses <- group_by(.data = ASSP, session_ID, session_year) %>%
  summarise(wing_av = mean(wing, na.rm = TRUE)) %>%
  ungroup()
```

```
## 'summarise()' regrouping output by 'session_ID' (override with '.groups' argument)
ggplot(WC_ses) +
  geom_point(aes(x = session_ID, y = wing_av, color = session_year)) +
  theme_bw()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



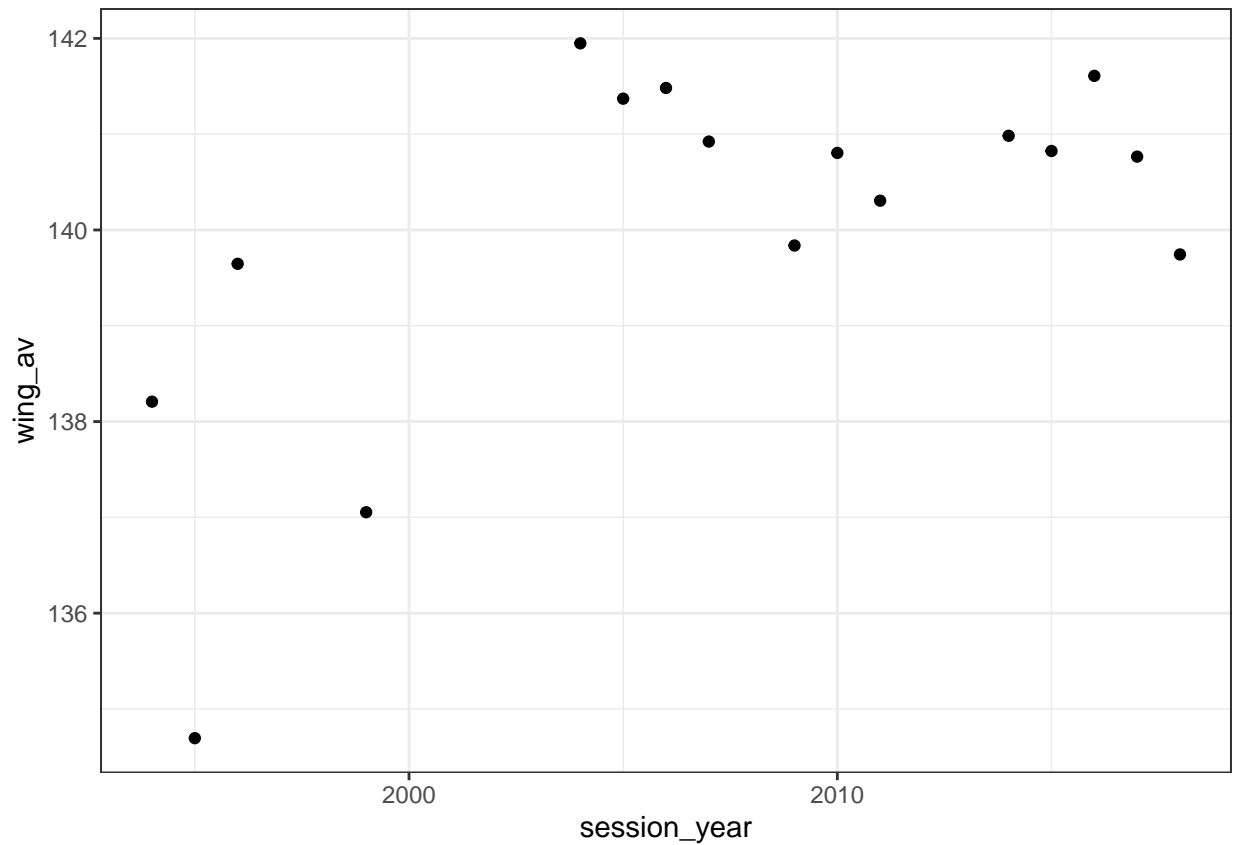
```
# Values in earlier years (~1990s) appear smaller.
```

```
# How about mean wing chord observed per year?
```

```
WC_yr <- group_by(.data = ASSP, session_year) %>%
  summarise(wing_av = mean(wing, na.rm = TRUE)) %>%
  ungroup()
```

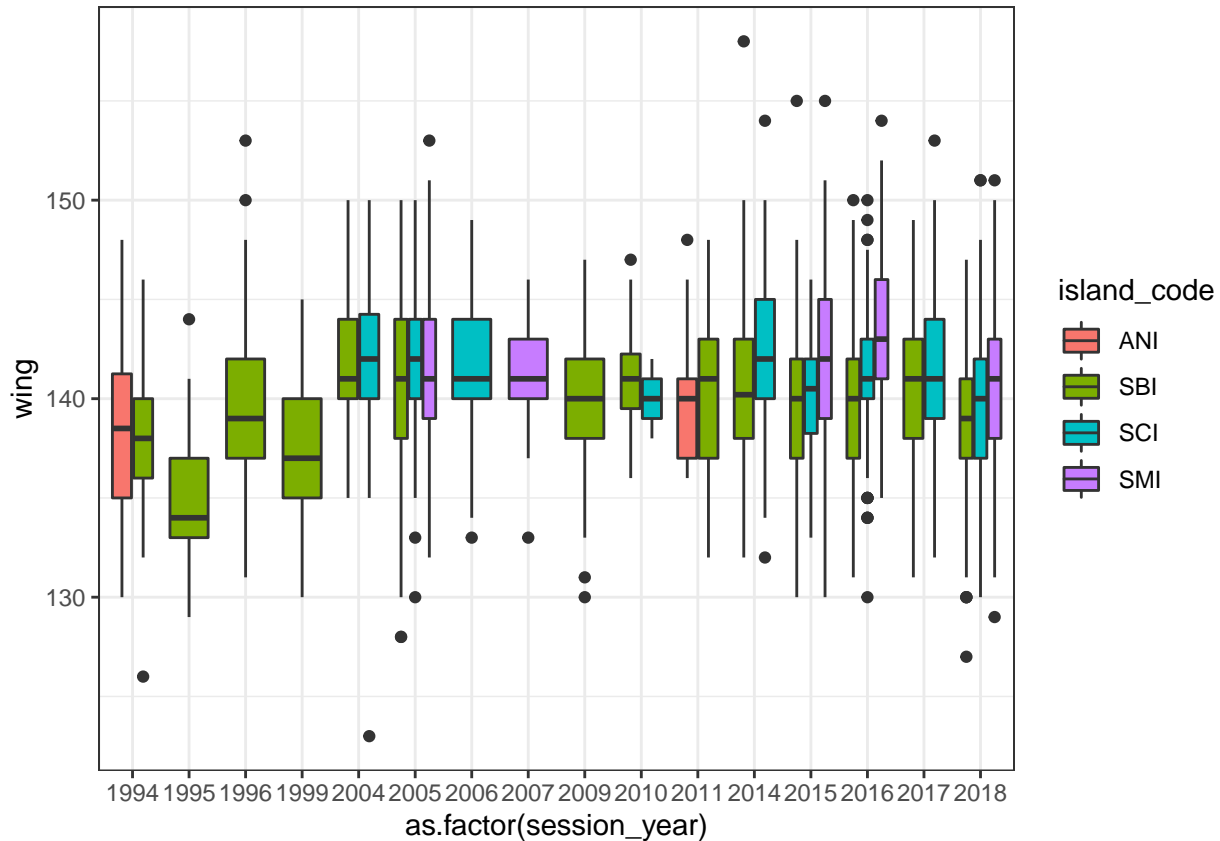
```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
ggplot(WC_yr) +
  geom_point(aes(x = session_year, y = wing_av)) +
  theme_bw()
```



```
# Boxplot of wing chord values  
ggplot(ASSP, aes(x = as.factor(session_year), y = wing, fill = island_code)) +  
  geom_boxplot() +  
  theme_bw()
```

```
## Warning: Removed 159 rows containing non-finite values (stat_boxplot).
```



It does look like the wing chord values recorded in the 1990s on Santa Barbara Island (PRBO) and Anacapa Island (Harry Carter) are significantly smaller than the values recorded by USGS and CHIS collaborators in the 2000-2010s. It's possible they used a relaxed wing measurement as opposed to a flattened wing measurement used later on. We made a note in the Banding Data Dictionary under "wing" that a different method might have been used to take wing chord measurements over time.

Tail

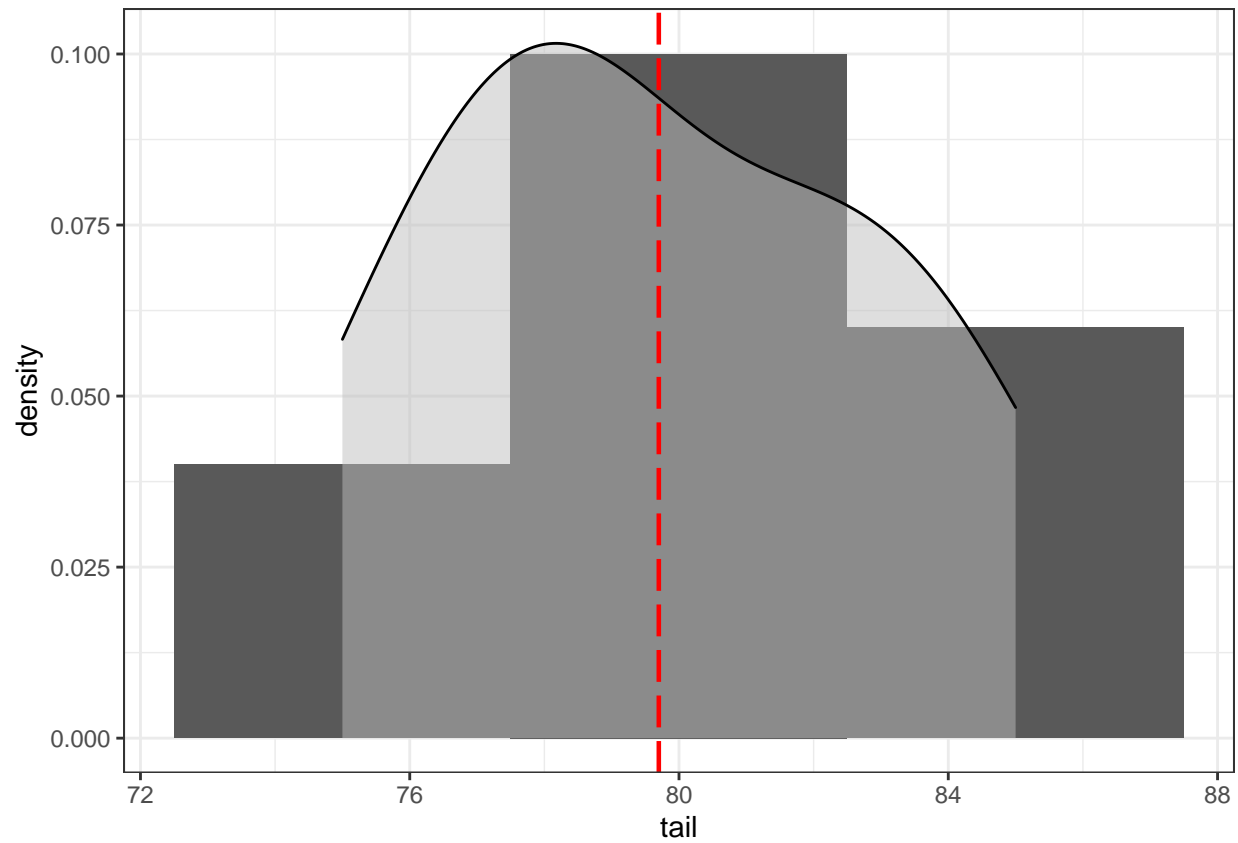
```
summary(ASSP$tail)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      75.0   78.0    79.0   79.7   82.5   85.0    3831
```

```
# Histogram
ggplot(data = ASSP, aes(x = tail, na.rm = TRUE)) +
  geom_histogram(aes(y = ..density..), binwidth = 5) +
  geom_density(alpha = .5, fill = "gray") +
  geom_vline(aes(xintercept = mean(tail, na.rm = TRUE)),
    colour = "red", linetype = "longdash", size = .8) +
  theme_bw()
```

```
## Warning: Removed 3831 rows containing non-finite values (stat_bin).
```

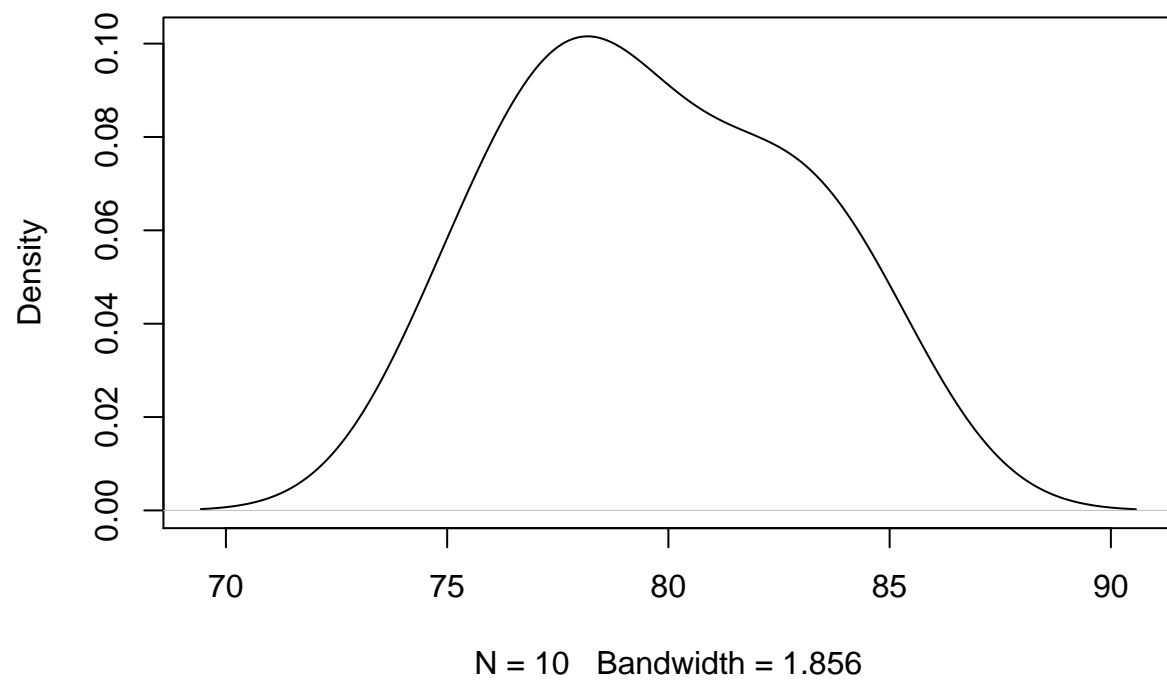
```
## Warning: Removed 3831 rows containing non-finite values (stat_density).
```



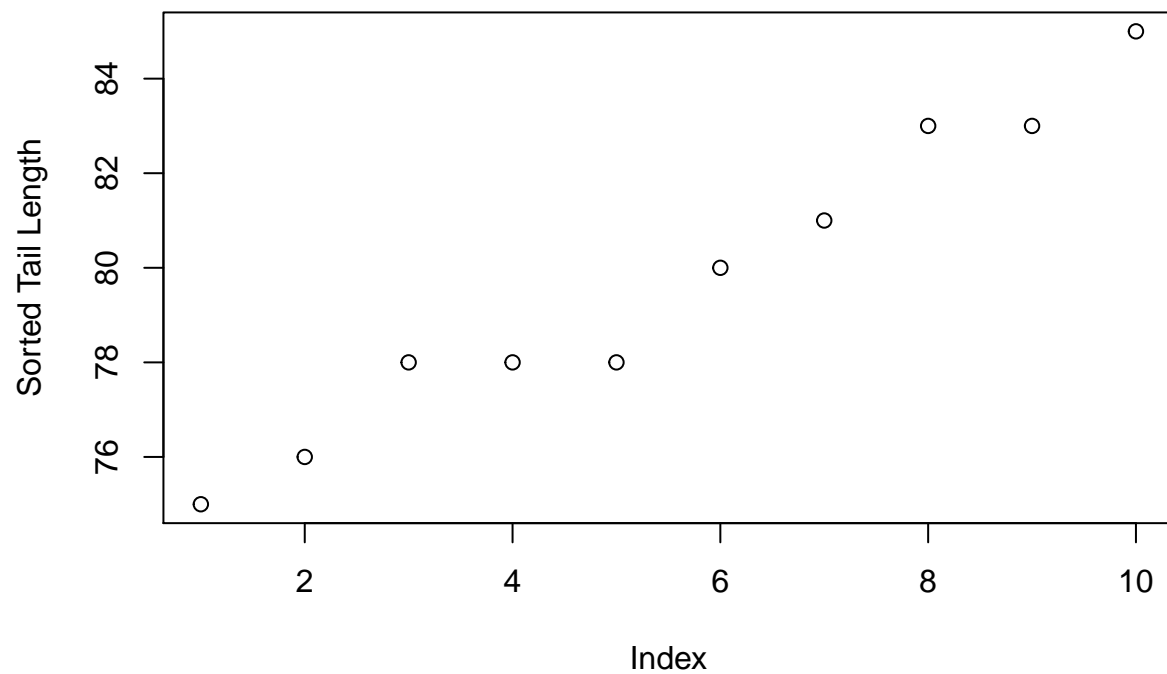
```
# There are very few values for this field.
```

```
# Density plot
```

```
plot(density(ASSP$tail, na.rm = TRUE), main = "")
```

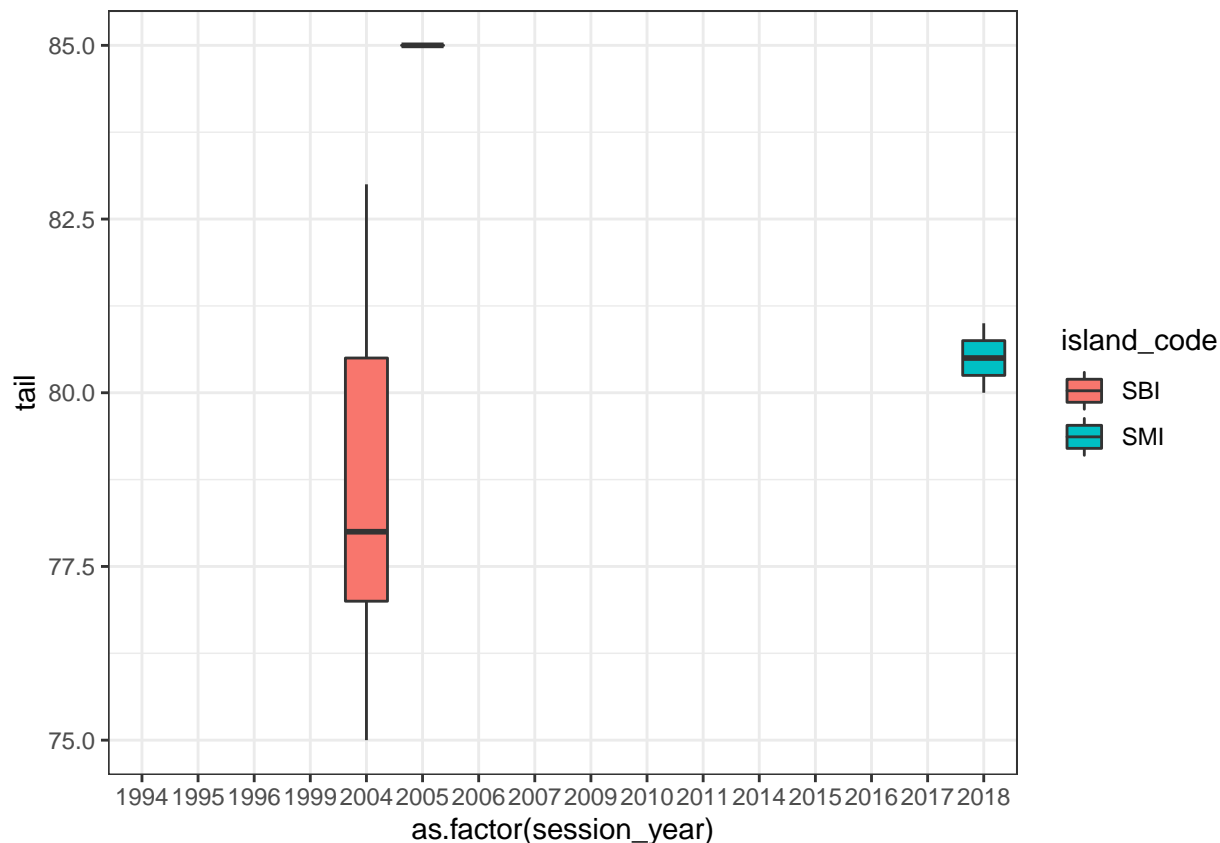


```
# Sorted values plot  
plot(sort(ASSP$tail), ylab = "Sorted Tail Length")
```



```
# Boxplot of values  
ggplot(ASSP, aes(x = as.factor(session_year), y = tail, fill = island_code)) +  
  geom_boxplot() +  
  theme_bw()
```

```
## Warning: Removed 3831 rows containing non-finite values (stat_boxplot).
```



```
# Values reported in Pyle guide: 72-84 (95% CI).
```

```
#Isolate questionable data.
```

```
tail_chk <- filter(ASSP, tail < 71 | tail > 85)
```

```
# Export to csv and cross-reference raw data.
```

```
#write.csv(tail_chk, "tailQAQC.csv")
```

We reviewed the exported data and cross-reference it with raw data (if available). If there was a data entry error, we rectified the mistake in the database and tracked changes in a separate csv file.

CPUE

Set-up data for analysis

```
metadata <- cpue %>%
  mutate_at(c("app_sunset", "std_ending", "net_open_1", "net_close_1"),
    .funs = list(time = ~ hms::as_hms(.))) %>%
  mutate(CPUE_ratio = CPUEstd/CPUEraw,
    month = as.character(session_month))

metadata_effort <- metadata %>%
  dplyr::select(session_ID, site_code, app_sunset, std_ending, lat, long)

catches <- captures %>%
  left_join(metadata_effort, by = c("session_ID", "site_code", "lat", "long")) %>%
```



```

mutate_at(c("app_sunset", "std_ending", "capture_date", "release_date"),
  .funs = ~as.POSIXct(., format="%Y-%m-%d %H:%M:%S")) %>%
filter(species == "ASSP") %>%
mutate(std = if_else(std_ending > capture_date, "1", "0"),
  catchPastSS = capture_date - app_sunset,
  assumeBreed = mosaic::derivedFactor(
    "Y" = (BP == "B" | BP == "2" | BP == "3" | BP == "4"),
    "N" = (BP == "D" | BP == "0" | BP == "5" | BP == "PD" | BP == "1" | BP == "1.5" | BP == "4.5"),
    .default = "ND")) %>%
filter(TRUE)

```

```
## Registered S3 method overwritten by 'mosaic':
```

```
##   method                                from
```

```
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```

metadata_BP <- catches %>%
  group_by(session_ID, site_code) %>%
  summarise(ASSP = n(),
    ASSPstd = sum(std == "1"),
    BPct = n(),
    BP_Y = sum(assumeBreed == "Y"), # number of birds that have a broodpatch (2-4, B)
    BP_N = sum(assumeBreed == "N")) %>% # number of birds that don't have a broodpatch (1, 1.5,
right_join(metadata, by= c("session_ID", "site_code")) %>%
mutate(BPfreq_Y = BP_Y/BPct, # frequency of birds that have a broodpatch
  BPfreq_N = BP_N/BPct)

```

```
## 'summarise()' regrouping output by 'session_ID' (override with '.groups' argument)
```

Time and Mistnetting Effort

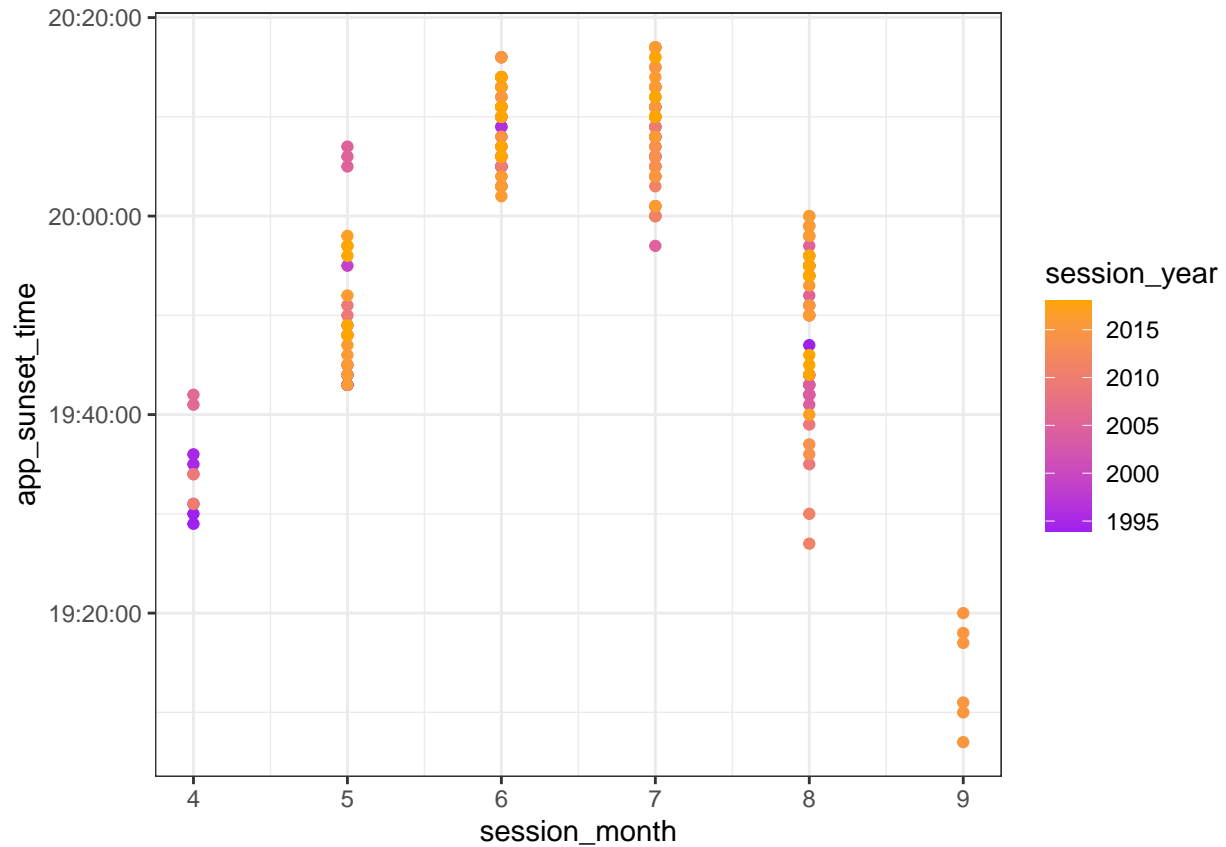
App_sunset

```

ggplot(metadata, aes(session_month, app_sunset_time)) +
  geom_point(aes(color = session_year)) +
  scale_color_gradient(low="purple", high="orange") +
  theme_bw()

```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```

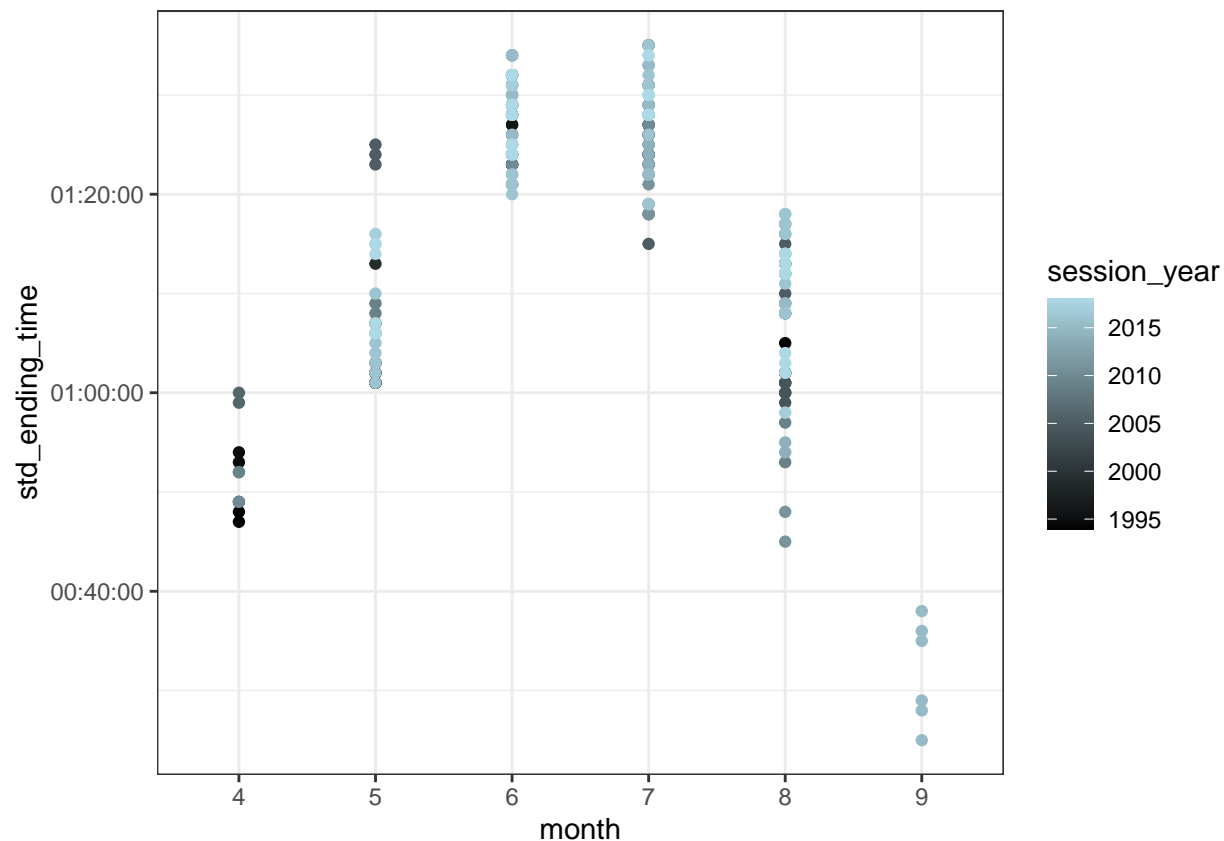


This graph shows the time of apparent sunset for netting sessions each month. The range and timing for that time of year is as we would expect. Thus we conclude that the `suncalc` function was used effectively to get the sunset times associated with each mistnetting session.

Std_ending

```
ggplot(metadata, aes(month, std_ending_time)) +
  geom_point(aes(color = session_year)) +
  scale_color_gradient(low="black", high="light blue") +
  theme_bw()
```

Warning: Removed 4 rows containing missing values (geom_point).



This graph shows the time of standard ending (5.3 hours after sunset) for netting sessions each month. The range and timing of the standard ending track with sunset time as we would expect.

net_open and net_close

```
# first (and usually only) net open time
```

```
summary(as.POSIXct(metadata$net_open_1_time))
```

```
##                Min.                1st Qu.                Median
## "1970-01-01 00:00:00" "1970-01-01 20:45:00" "1970-01-01 21:02:00"
##                Mean                3rd Qu.                Max.
## "1970-01-01 20:33:25" "1970-01-01 21:35:15" "1970-01-01 23:35:00"
##                NA's
##                "22"
```

```
# first (and usually only) net close time
```

```
summary(as.POSIXct(metadata$net_close_1_time))
```

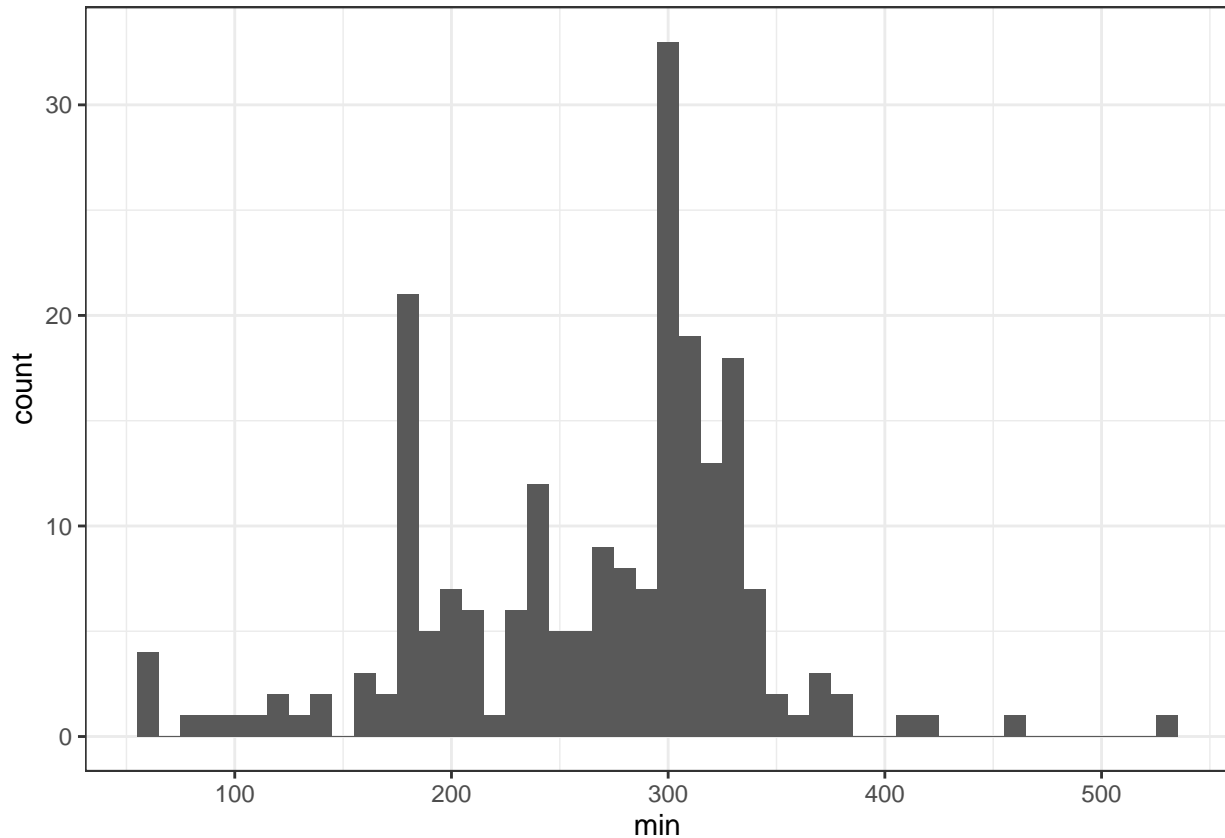
```
##                Min.                1st Qu.                Median
## "1970-01-01 00:00:00" "1970-01-01 01:25:00" "1970-01-01 02:00:00"
##                Mean                3rd Qu.                Max.
## "1970-01-01 03:45:32" "1970-01-01 02:20:00" "1970-01-01 23:59:00"
##                NA's
##                "21"
```

This is not a perfect way to summarize net open and close times because the “summarize” function doesn’t recognize times across midnight here. But, by looking at the median and mean, we can tell that net open and close times are usually what we would expect, with a few late/early nights thrown in.

Total mistnetting minutes per session

```
ggplot(metadata, aes(min)) +  
  geom_histogram(binwidth = 10) +  
  theme_bw()
```

```
## Warning: Removed 26 rows containing non-finite values (stat_bin).
```



```
# summary of total mistnetting minutes  
summary(metadata$min)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's  
##      56.0  212.2   290.0   266.1  315.0   530.0      26
```

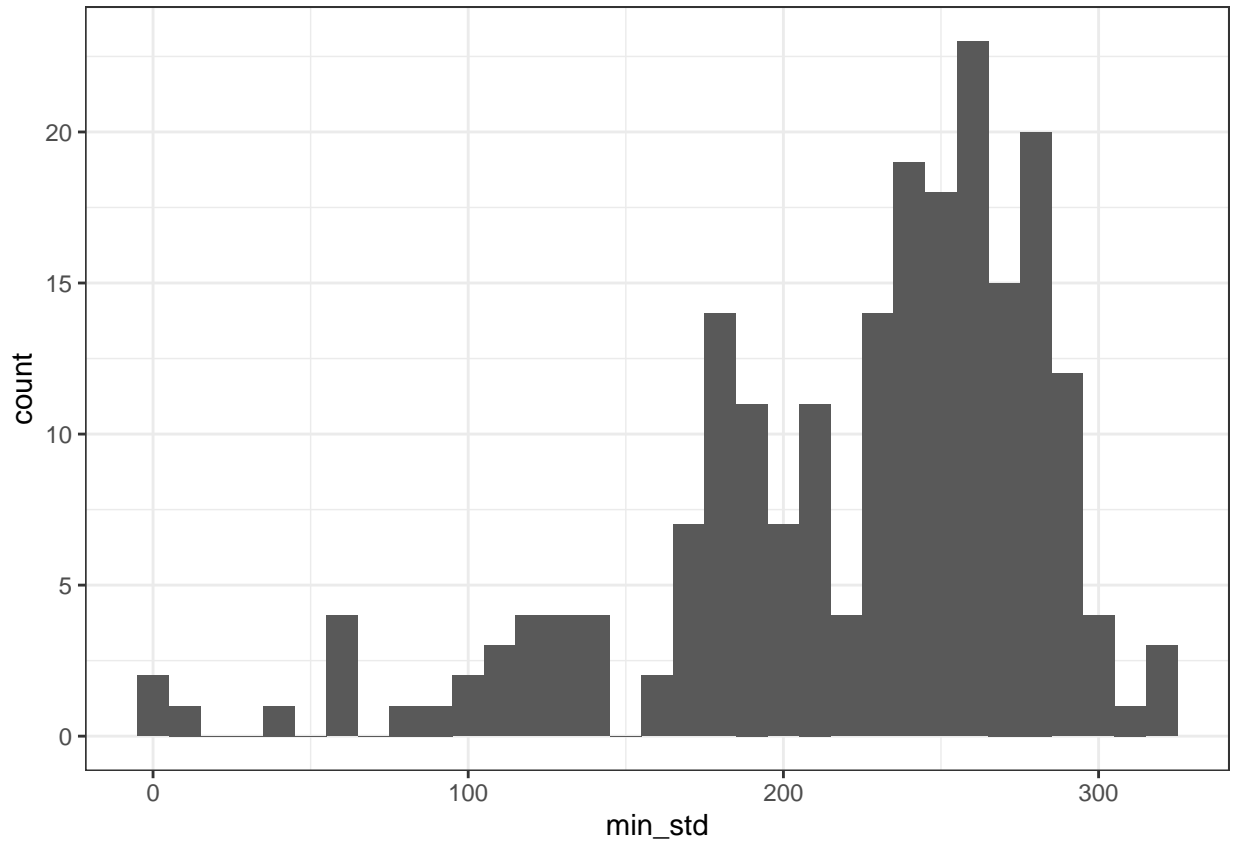
The graph and summary stats above show minutes calculated for each netting session. We want to check that minutes were added correctly across multiple open/close sessions and also that minutes were added accurately across midnight. It looks like minutes were not added accurately across midnight on four occasions (the outliers to the far right)

Total mistnetting standard minutes per session

Total minutes from start until end or standard ending, whichever came first

```
ggplot(metadata, aes(min_std)) +  
  geom_histogram(binwidth = 10) +  
  theme_bw()
```

```
## Warning: Removed 26 rows containing non-finite values (stat_bin).
```



```
# summary of mistnetting minutes cut to standard ending time
summary(metadata$min_std)
```

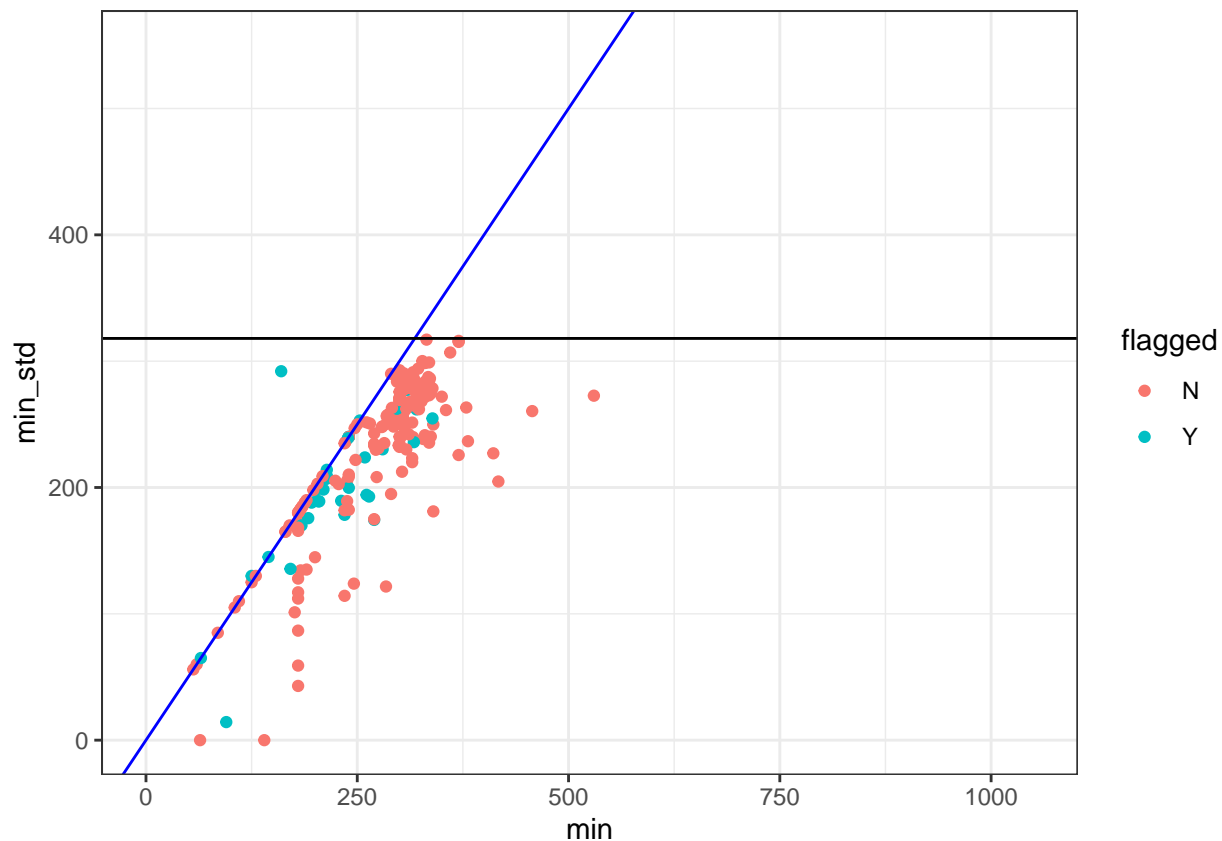
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##      0.0   188.8   239.6   222.8   267.7   317.1     26
```

The above graph and summary stats show the total number of minutes for the standardized session (from net open to net close or standard ending whichever came first). The standard ending is 5.3 hours (318 minutes) after sunset. Nets opened sometime after sunset.

min vs. min_std for each session

```
ggplot(metadata, aes(min, min_std)) +
  geom_point(aes(color = flagged)) +
  geom_abline(intercept = 0, slope = 1, color = "blue") +
  geom_hline(yintercept = 318, color = "black") +
  xlim(0,1050) + ylim(0, 550) +
  theme_bw()
```

```
## Warning: Removed 26 rows containing missing values (geom_point).
```



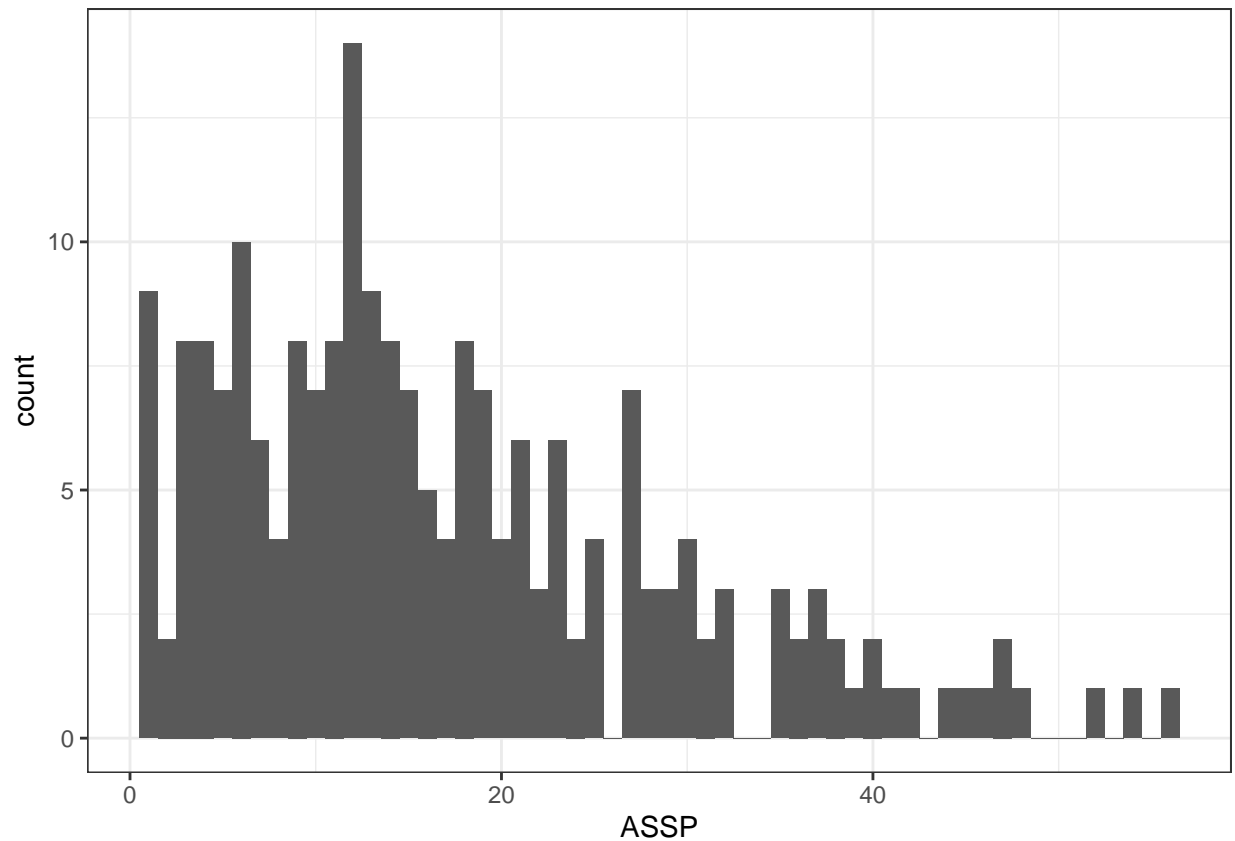
This plot of minutes vs. standardized minutes (before the 5.3 hour standardized ending). Blue line = slope of 1. Black line = standard ending. Here we can make sure standardized net open minutes is equal or less the total number of minutes and sunset to standard ending of 5.3 hours (318 minutes). Turquoise points = data that has been flagged due to inconsistencies in data entry. It does not appear that the reason these entries were flagged effects net minutes.

ASSP

ASSP caught per session

```
ggplot(metadata, aes(ASSP)) +
  geom_histogram(binwidth = 1) +
  theme_bw()
```

Warning: Removed 28 rows containing non-finite values (stat_bin).



```
# summary of ASSP catches
summary(metadata$ASSP)
```

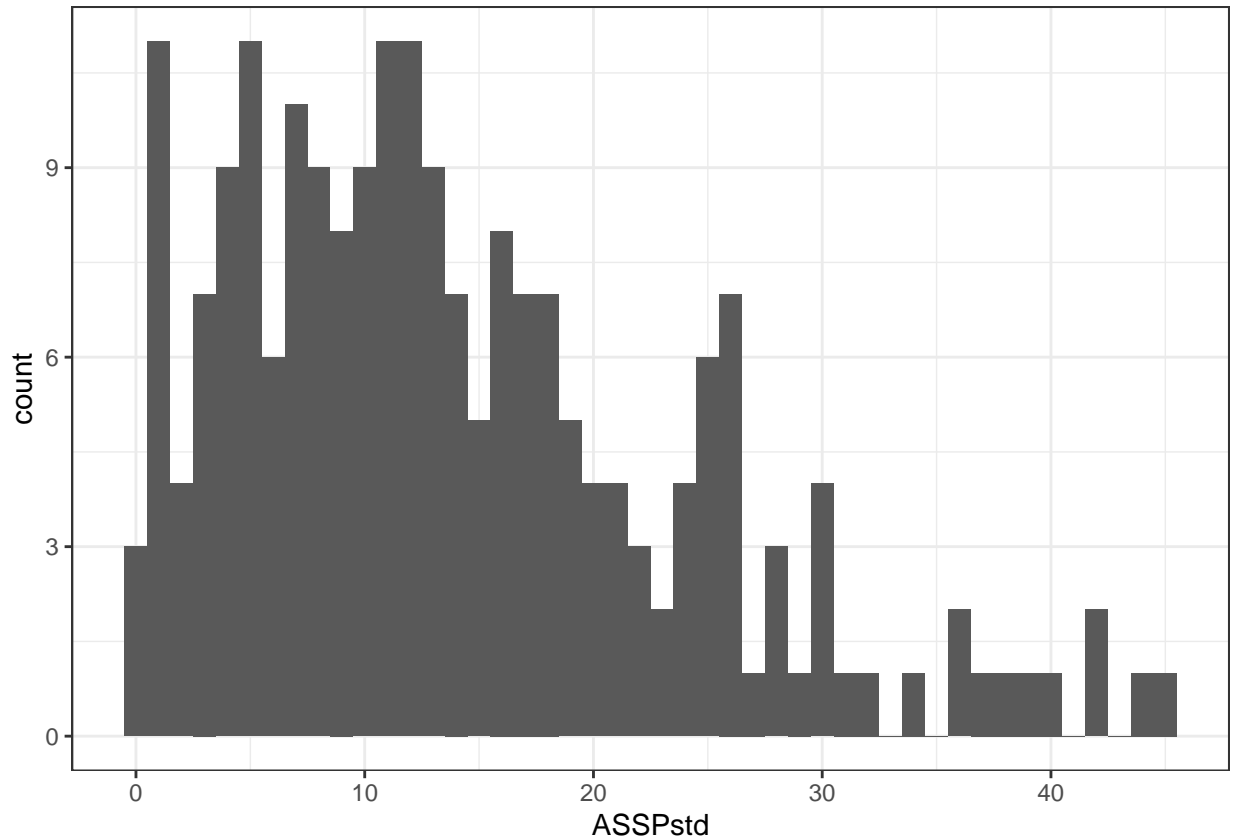
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##      1.00   8.00   14.00   17.12  23.00   56.00    28
```

The graph and summary stats above show the distribution of total numbers of ASSP caught per session.

ASSP caught per standardized session

```
ggplot(metadata, aes(ASSPstd)) +
  geom_histogram(binwidth = 1) +
  theme_bw()
```

```
## Warning: Removed 29 rows containing non-finite values (stat_bin).
```



```
# summary of standardized ASSP catches
summary(metadata$ASSPstd)
```

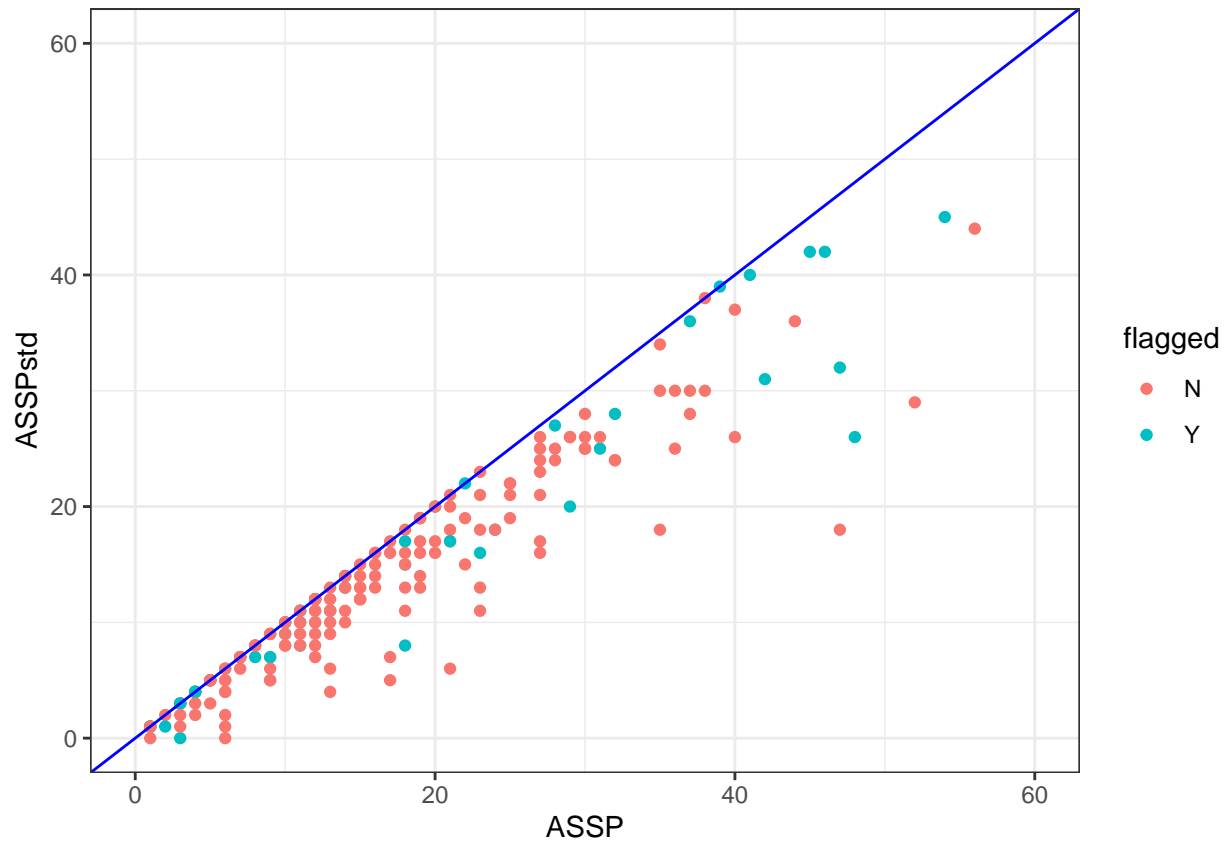
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##      0.00   7.00   12.00   14.08   19.00   45.00    29
```

The graph and summary stats above show the distribution of total numbers of ASSP caught before standard ending or net close, whichever came first. This distribution is more constrained than the one above, which is what we would expect with the standard ending cutoff.

ASSP vs ASSPstd

```
ggplot(metadata, aes(ASSP, ASSPstd)) +
  geom_point(aes(color = flagged)) +
  geom_abline(intercept = 0, slope = 1, color = "blue") +
  xlim(0,60) + ylim(0, 60) +
  theme_bw()
```

```
## Warning: Removed 29 rows containing missing values (geom_point).
```

The plot above shows the total number of ASSP vs. total number of ASSP before the standard ending. Blue line = slope of 1. Here we double check that the standardized number of ASSP is always equal to or less than the total number. Turquoise points = data that has been flagged due to inconsistencies in data entry. It does not appear that the reason these entries were flagged effects the number of birds caught.

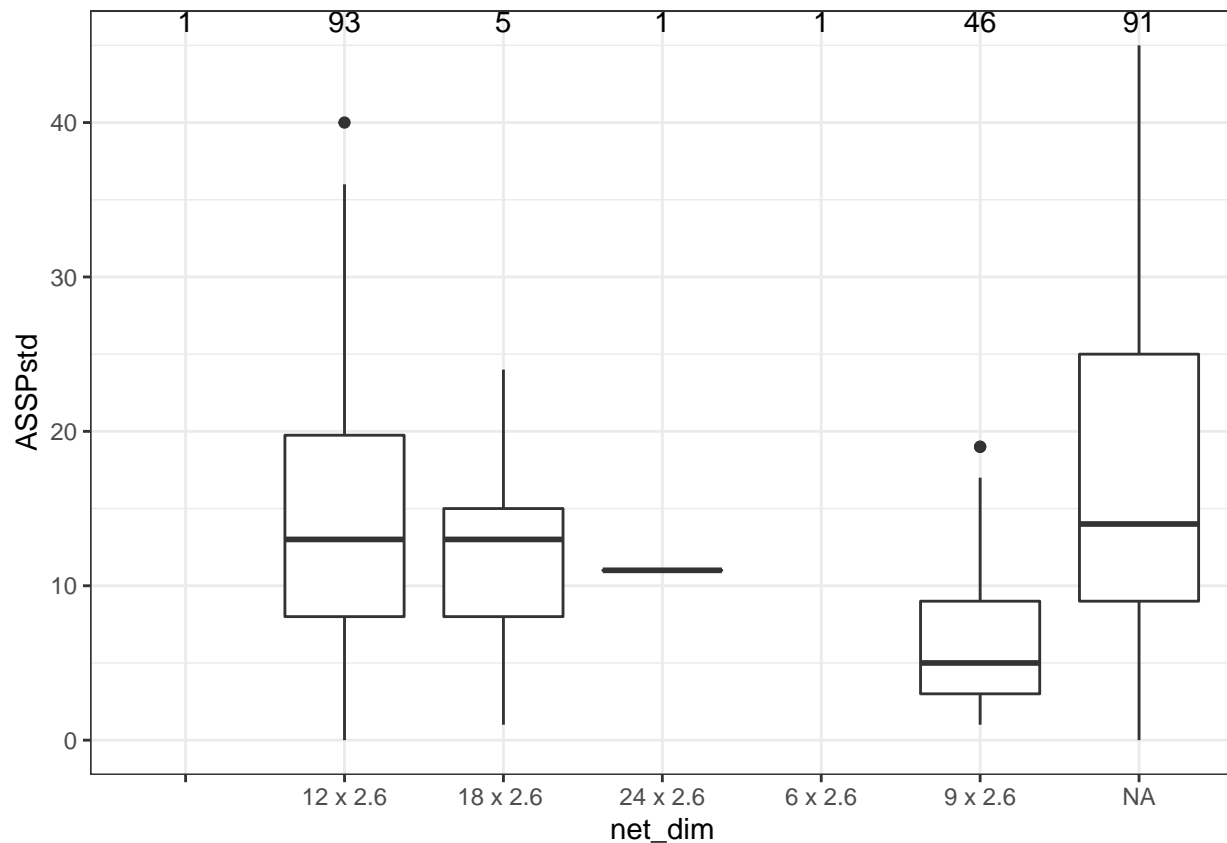
Next we will look into what else could effect the number of birds caught.

ASSP caught in relation to net size

```
netUse <- metadata %>%
  group_by(net_dim) %>%
  tally()

ggplot(metadata, aes(net_dim, ASSPstd)) +
  geom_boxplot() +
  geom_text(data = netUse,
            aes(net_dim, Inf, label = n), vjust = 1) +
  theme_bw()
```

```
## Warning: Removed 29 rows containing non-finite values (stat_boxplot).
```



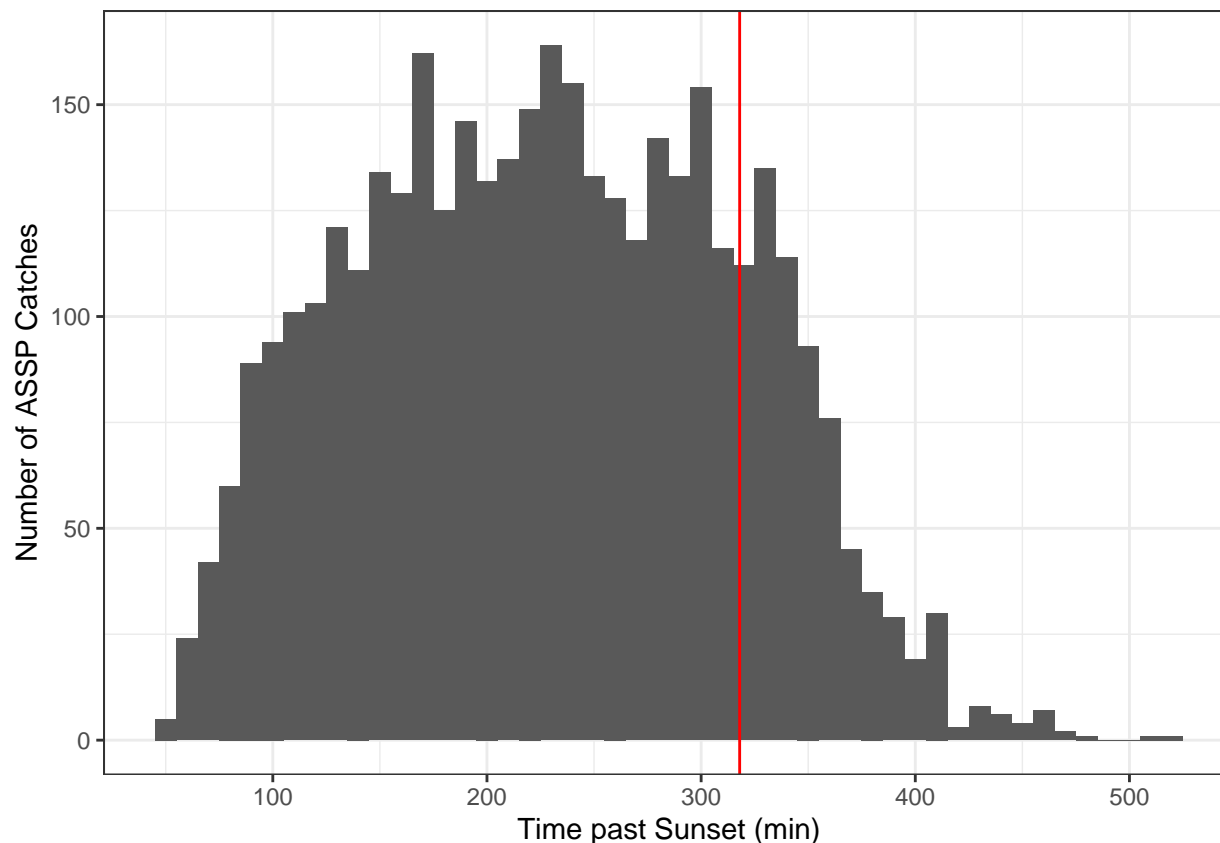
The graph above shows the number of ASSP caught within the standardized period in relation to net dimensions. Number at top of each box plot = sample size. The size of the net doesn't seem to effect the number of birds caught in a specific night. Below we will explore this effect with number of catches standardized to effort.

Timing of ASSP catches

Next lets explore the frequency of catches in relation to the standard ending. Do catches start dropping off before 5.3 hours after sunset? After? NOTE - the timing of net closures will effect the number of late night captures

```
ggplot(catches, aes(catchPastSS)) +
  geom_histogram(binwidth = 10) +
  geom_vline(xintercept = 318, color = "red") +
  xlab("Time past Sunset (min)") + ylab("Number of ASSP Catches") +
  theme_bw()
```

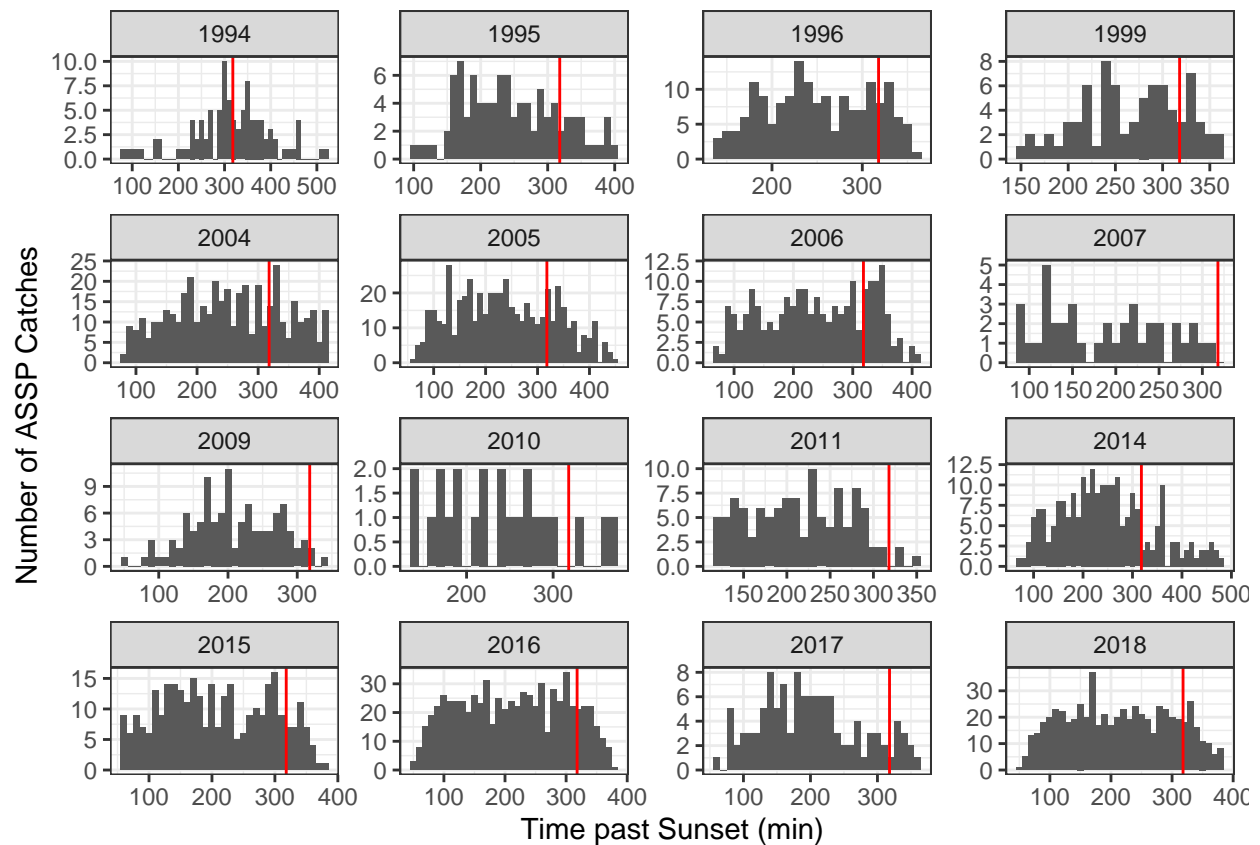
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
## Warning: Removed 44 rows containing non-finite values (stat_bin).
```



The 5.3 hour cutoff occurs when number of catches are still relatively high. How does that differ across years? In recent years (2014 - 2018) nets were usually closed at 2am. In earlier years nets were sometimes left open past 2am.

```
ggplot(catches, aes(catchPastSS)) +
  geom_histogram(binwidth = 10) +
  geom_vline(xintercept = 318, color = "red") +
  xlab("Time past Sunset (min)") + ylab("Number of ASPP Catches") +
  facet_wrap(~ session_year, scales = "free") +
  theme_bw()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
## Warning: Removed 44 rows containing non-finite values (stat_bin).
```



NOTE: scales on x- and y- axis differ between panes

No clear pattern of catch timing can be seen across years.

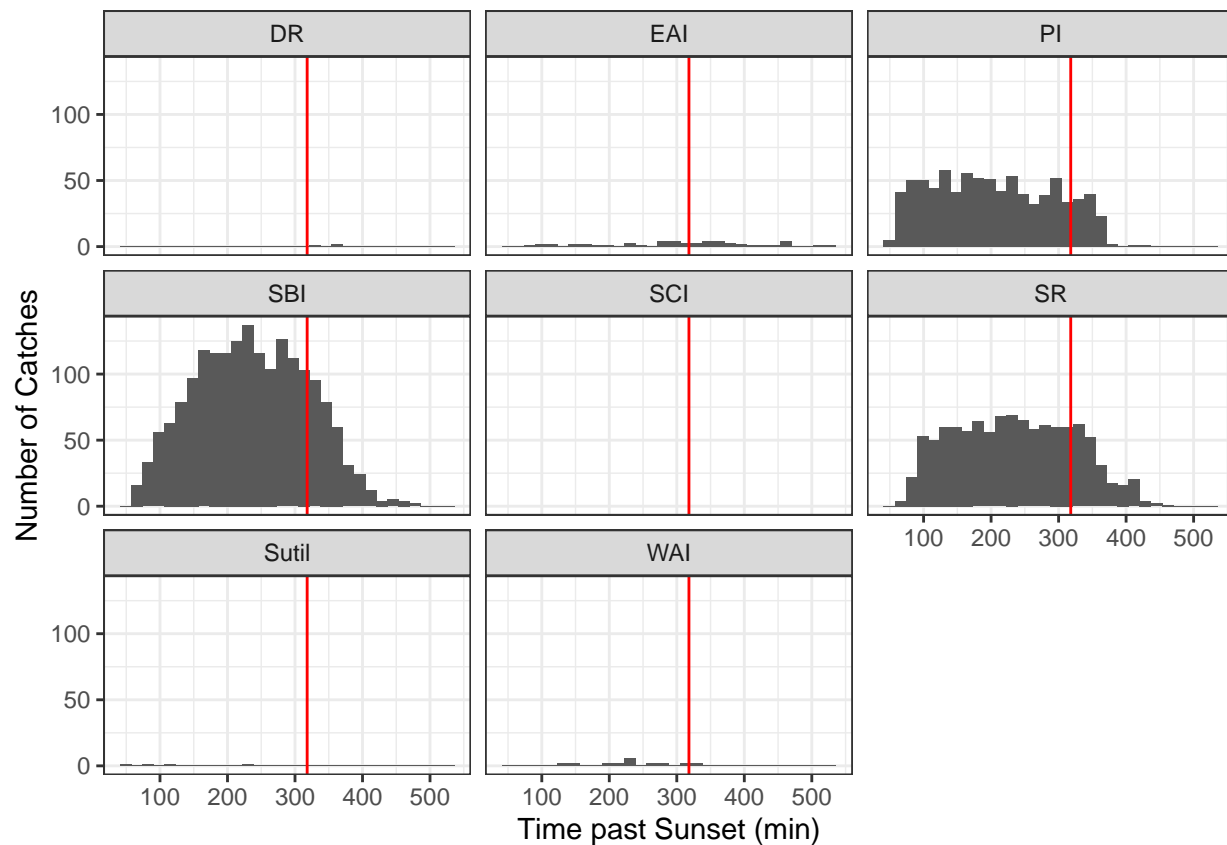
How does catch time post sunset differ between netting sub-islands?

```
ggplot(catches, aes(catchPastSS)) +
  geom_histogram() +
  geom_vline(xintercept = 318, color = "red") +
  xlab("Time past Sunset (min)") + ylab("Number of Catches") +
  facet_wrap(~ subisland_code) +
  theme_bw()
```

Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.

Warning: Removed 44 rows containing non-finite values (stat_bin).



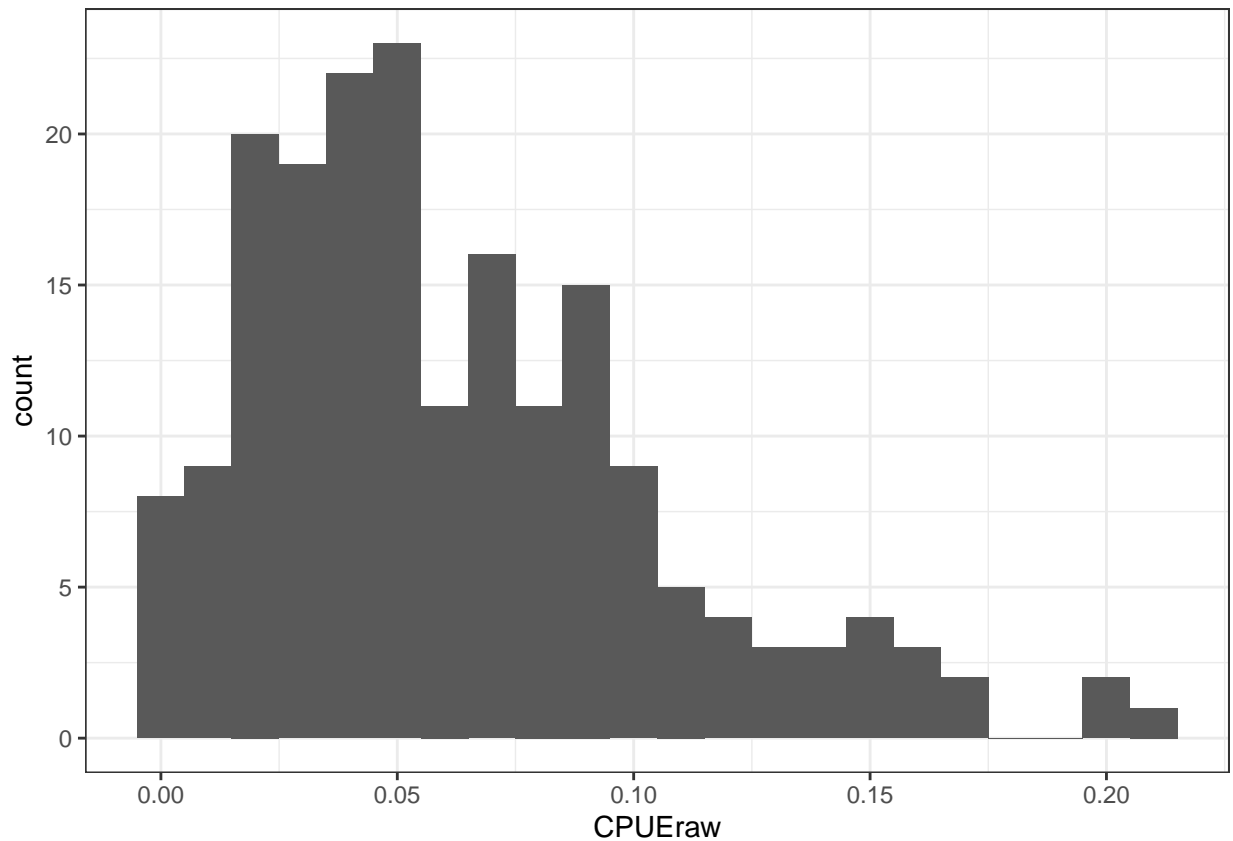
There doesn't appear to be any distinct pattern in timing of catches between locations, especially given the variation in sample sizes between locations.

CPUE

CPUE per session

```
ggplot(metadata, aes(CPUEraw)) +  
  geom_histogram(binwidth = 0.01) +  
  theme_bw()
```

Warning: Removed 48 rows containing non-finite values (stat_bin).



```
# summary of CPUE per session
summary(metadata$CPUEraw)
```

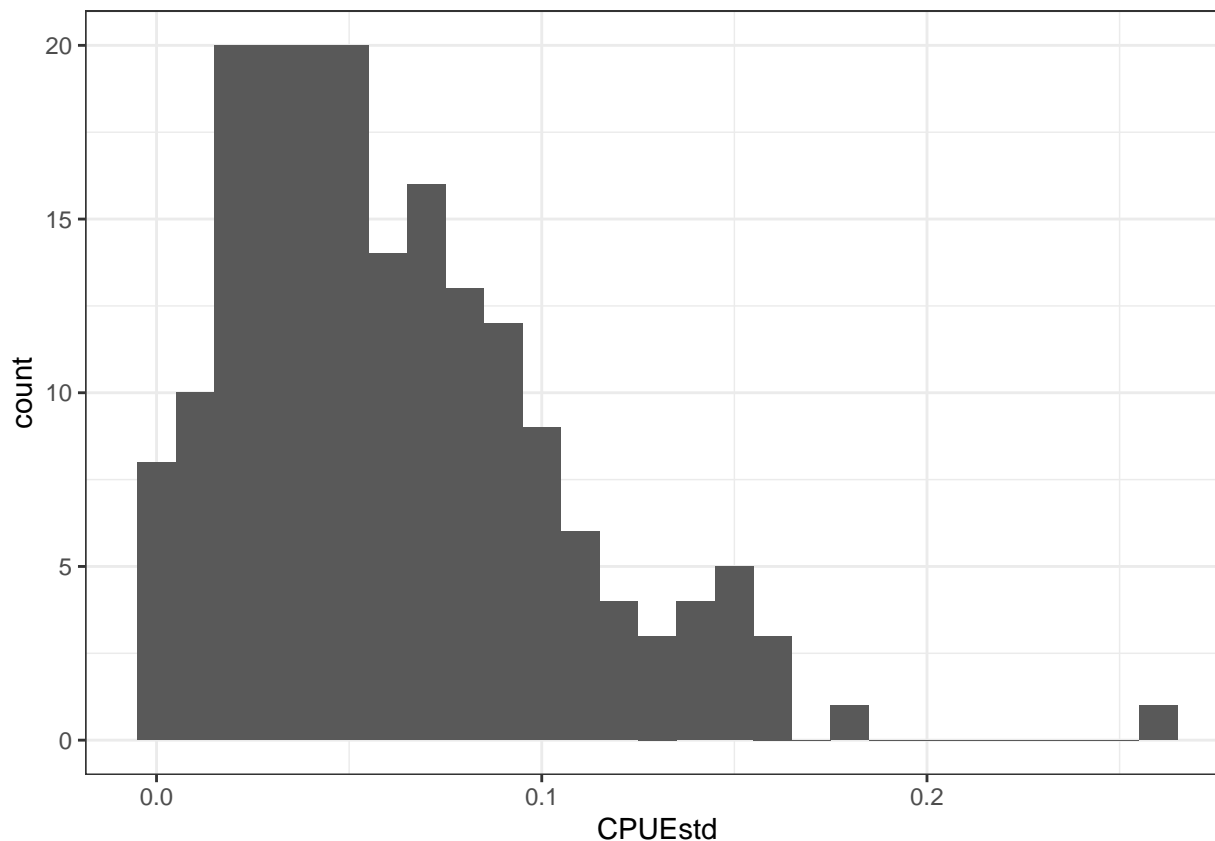
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
## 0.00282 0.03245 0.05298 0.06267 0.08795 0.20779      48
```

The graph and summary stats above show the distribution of catch-per-unit-effort (ASSP/min).

CPUE per standardized session

```
ggplot(metadata, aes(CPUEstd)) +
  geom_histogram(binwidth = 0.01) +
  theme_bw()
```

```
## Warning: Removed 49 rows containing non-finite values (stat_bin).
```



```
# summary of CPUE per standardized session
summary(metadata$CPUEstd)
```

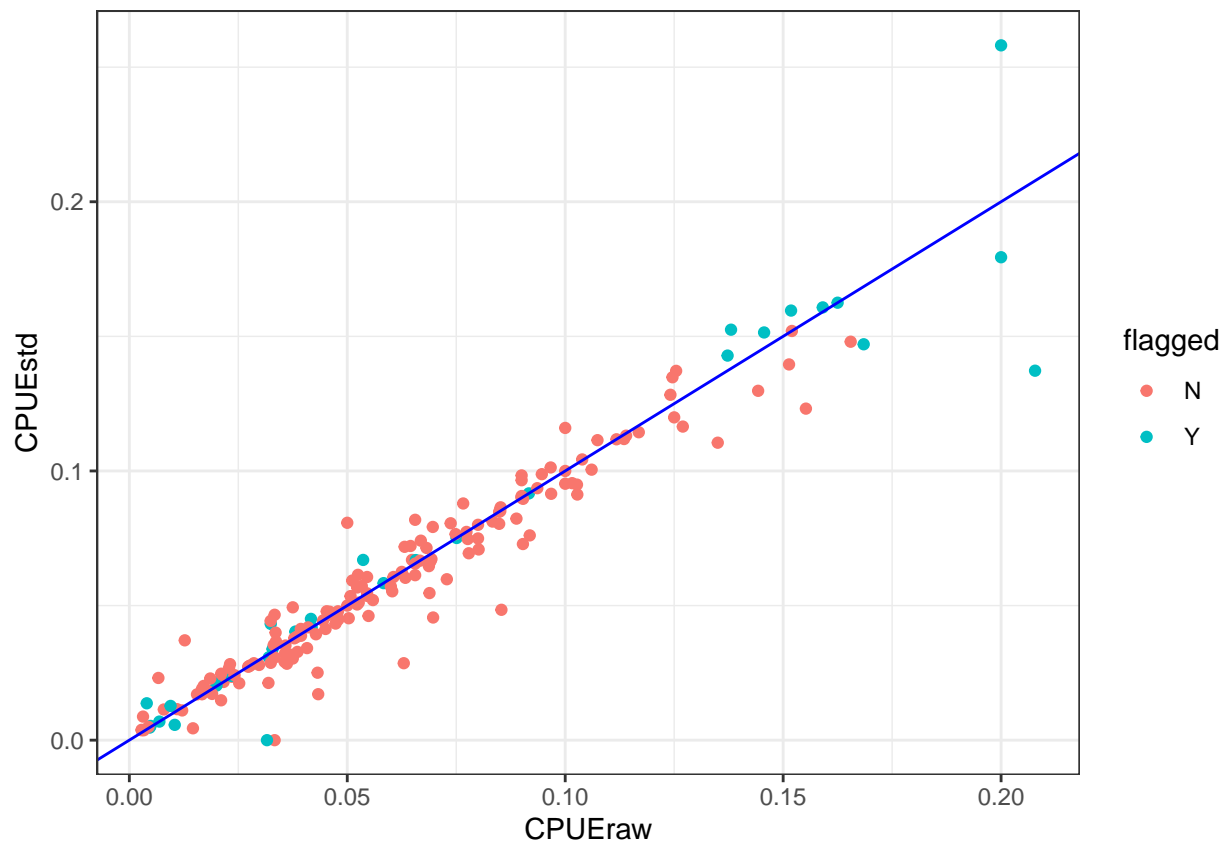
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
## 0.00000 0.02858 0.05351 0.06162 0.08511 0.25810     49
```

The graph and summary stats above show the distribution of standardized catch-per-unit-effort (ASSP-std/min_std). This distribution is more constrained than the previous one, which is what we would expect with the standard ending cutoff.

CPUE vs CPUEstd

```
ggplot(metadata, aes(CPUERaw, CPUEstd)) +
  geom_point(aes(color = flagged)) +
  geom_abline(intercept = 0, slope = 1, color = "blue") +
  theme_bw()
```

```
## Warning: Removed 49 rows containing missing values (geom_point).
```

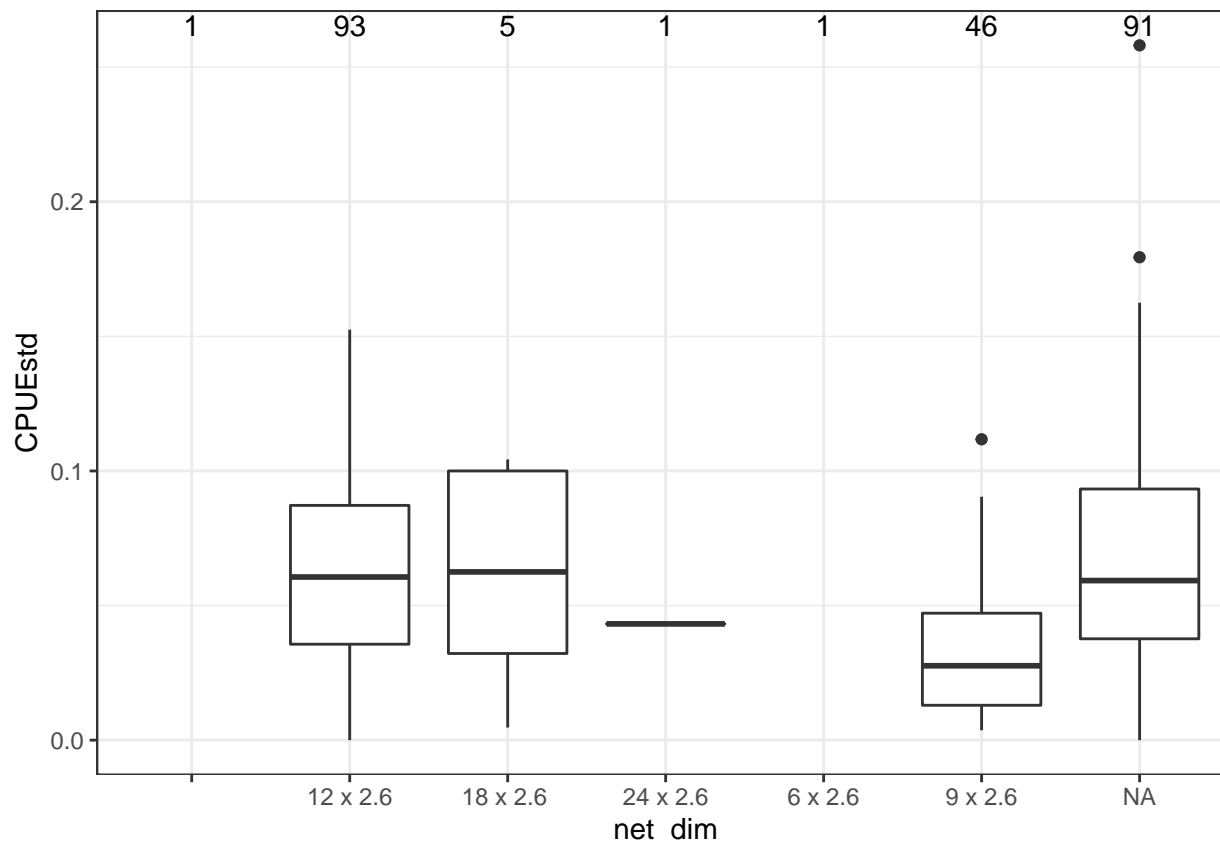


The above graph explores the correlation between CPUE and CPUE std. Blue line = slope of 1. As expected, the correlation is often 1:1, but with variation as the number of ASSP caught and number of mistnetting minutes were both affected by the standard ending cutoff but not always in a proportional way. Turquoise points = data that has been flagged due to inconsistencies in data entry. Sessions with high CPUE were flagged. Nothing in the data entry suggests these entries were erroneous.

CPUE in relation to net size

```
ggplot(metadata, aes(net_dim, CPUEstd)) +
  geom_boxplot() +
  geom_text(data = netUse,
            aes(net_dim, Inf, label = n), vjust = 1.2) +
  theme_bw()
```

Warning: Removed 49 rows containing non-finite values (stat_boxplot).



Here is the frequency of standardized CPUE values broken up by the dimensions of the net. The number above each box plot = the sample size. Unknown net sizes make it hard to determine if the size of the net influenced catch rates.

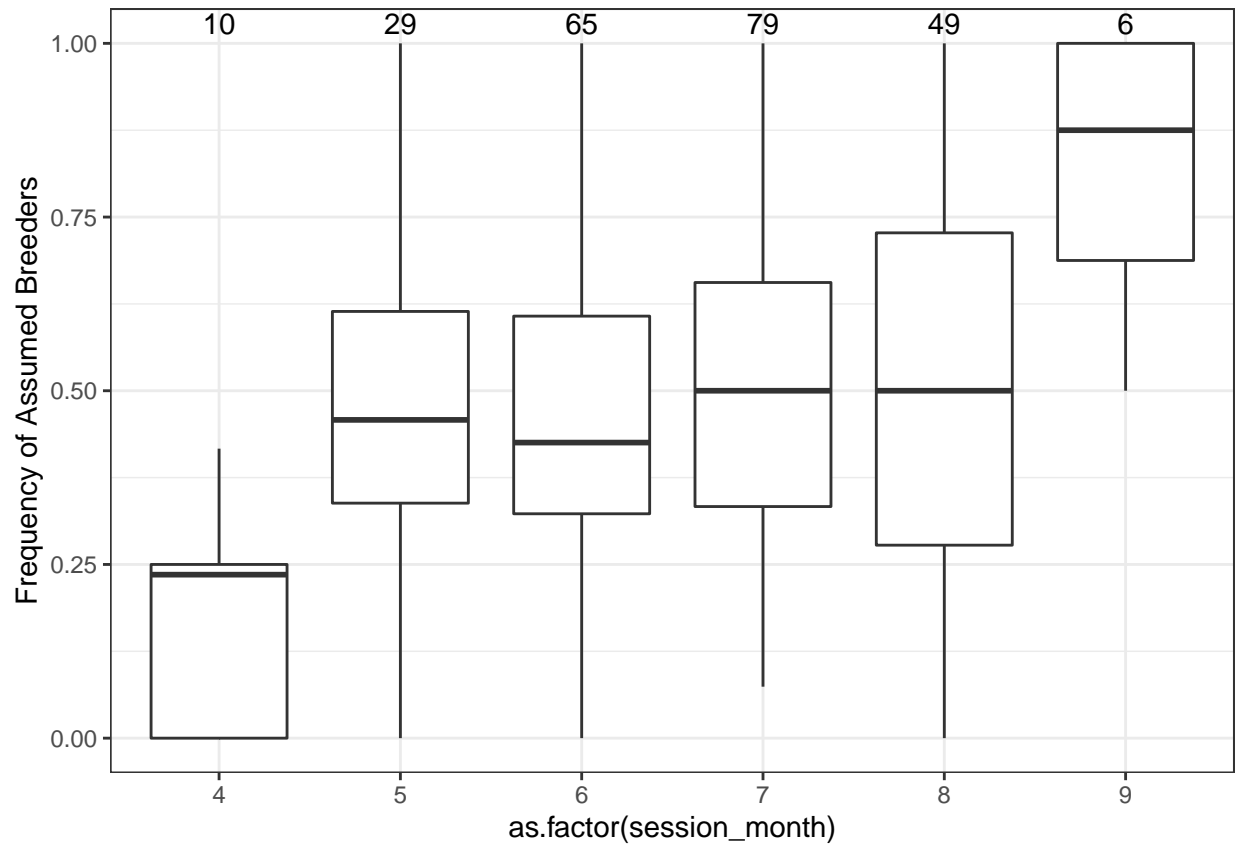
Brood Patch and Assumed Breeders

Below we explore the relationship between ASSP assumed to be breeding and the time of year. ASSP were assumed to be breeding if they had a bare broodpatch (brood patch score of 2-4; Ainley et al. 1990)

```
monthCatches <- metadata %>%
  group_by(session_month) %>%
  tally()

ggplot(metadata_BP, aes(x = as.factor(session_month), y = BPfreq_Y)) +
  geom_boxplot() +
  ylab("Frequency of Assumed Breeders") +
  geom_text(data = monthCatches,
            aes(as.factor(session_month), Inf, label = n), vjust = 1.2) +
  theme_bw()
```

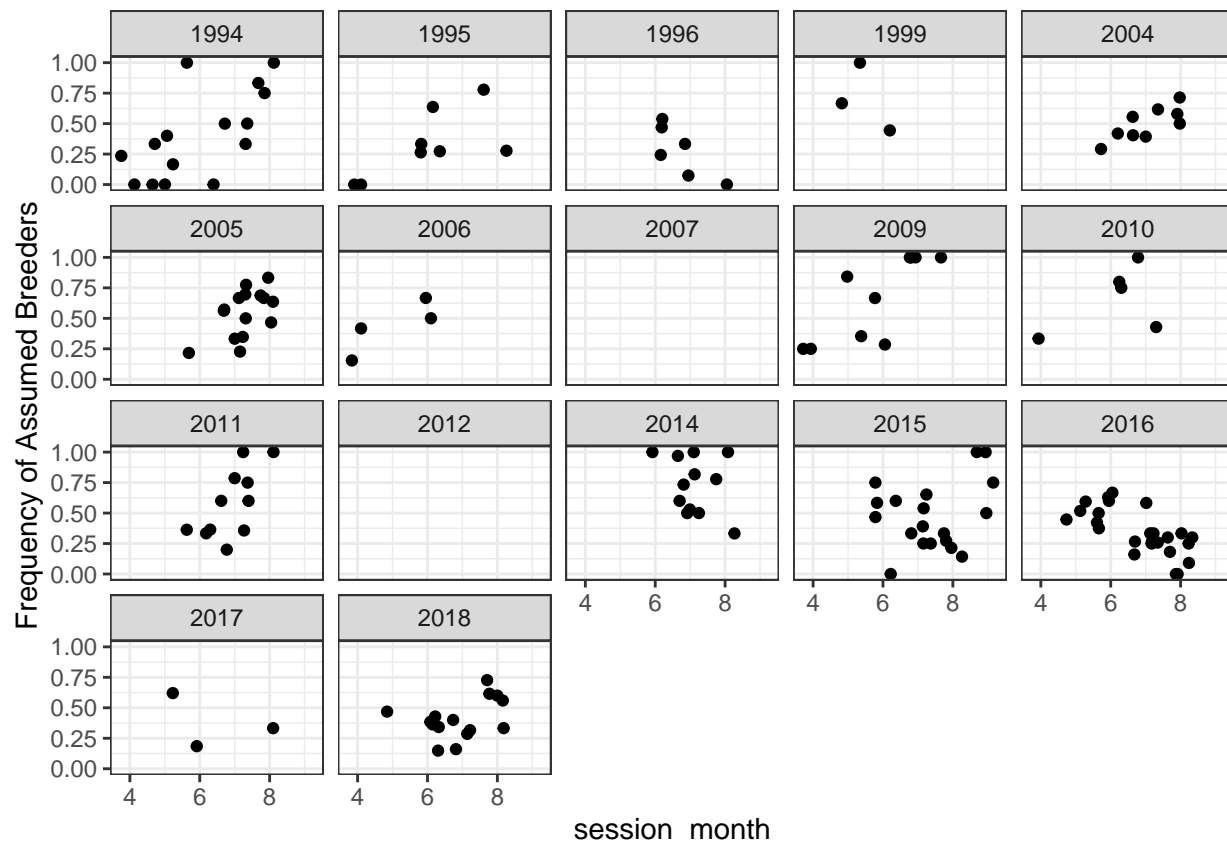
```
## Warning: Removed 78 rows containing non-finite values (stat_boxplot).
```



The above graph shows frequency of assumed breeders (brood patch 2-4) in relation to the month of netting effort. The number above each box plot = sample size. The number of assumed breeders appears to be higher later in the breeding season.

```
ggplot(metadata_BP, aes(session_month, BPfreq_Y)) +
  geom_point(position = "jitter") +
  ylab("Frequency of Assumed Breeders") +
  facet_wrap(~session_year) +
  theme_bw()
```

```
## Warning: Removed 78 rows containing missing values (geom_point).
```



The above graph shows the frequency of assumed breeders caught in mistnetting sessions across months, broken up by years. No distinct patterns appear here.

Captures and CPUE

Cross-reference session_ID in captures with session_ID in cpue to confirm that all mistnetting sessions are listed in CPUE sheet.

```
# Anti join returns records in captures are not matched by "session_ID" in cpue
chkIDs.captures <- anti_join(captures, cpue, by = "session_ID") # 0 obs

# Check for session_IDs in cpue that are not in captures
chkIDs.cpue <- anti_join(cpue, captures, by = "session_ID") # 14 obs

# Verify that these 14 obs had 0 captures
IDs <- unique(chkIDs.cpue$session_ID)
IDs.cpue <- cpue %>%
  filter(session_ID %in% IDs) # No captures

# Compare session day/month/year with capture day/month/year
captures_ydm <- captures %>%
  mutate(years = as.numeric(capture_year == session_year)) %>%
  mutate(days = as.numeric(capture_day == session_day)) %>% # there will be discrepancies due to midnights
  mutate(months = as.numeric(capture_month == session_month)) %>%
  mutate(handling_time = (as.duration(release_date - capture_date))) # this will only flag issues if relevant

## Isolate flagged values (e.g., 0s)
```

```

years <- captures_ydm %>%
  filter(years == 0) # 0 obs

months <- captures_ydm %>%
  filter(months == 0) # 0 obs

days <- captures_ydm %>%
  filter(days == 0) %>% # these are likely flagged due to after midnight captures
  mutate(capture_day = as.numeric(capture_day)) %>%
  mutate(session_day = as.numeric(session_day)) %>%
  mutate(diff = capture_day-session_day)
unique(days$diff) # all checks out (either 1, -30, or -29)

## [1] 1 -30 -29

long_handling <- captures_ydm %>%
  filter(handling_time > 1800 | handling_time < 0) # greater than 30 min

neg_handling <- captures_ydm %>%
  filter(handling_time < 0) # typos?

```