

Preliminary analysis of the available data for BC Butterflies and Dragonflies with a comparison of the data available online from collections and the data available from iNaturalist

Nathan G. Earley

2026-01-173

Intro

We're trying to figure out where we should do the Southern Interior of BC's Dragonfly Count and Butterfly Count.

I'm not uploading the data to GitHub because they're fairly big files... let me know if you want them and I'll share.

Candidate locations

There are 3 Christmas Bird count locations that seem like they would be the best fit for hosting the Ode and Butt Counts. Apex Mountain, Oliver-Osoyoos, and Vaseux Lake.

First we'll get those circles.

For the Oliver-Osoyoos, and Vaseux Lake CBC circles we can use Dave Bell's CBC circles which I downloaded from his "(My Google Maps)[<https://www.google.com/maps/d/viewer?mid=1D49EyLXIFoIBQYS5rE6Fi9YTEnb6H46A&ll=49.715363640775124%2C-119.40967101463359&z=8>]" and edited in OkanaganCBCcircles.R in this .RProj directory.

```
OliverOsoyoosCBC <- sf::st_read("data/NathanCBCkmls/OliverOsoyoosCBC_Circle.kml",
                                layer = "OliverOsoyoosCBC_Circle",
                                quiet = TRUE)
VaseuxLakeCBC <- sf::st_read("data/NathanCBCkmls/VaseuxLakeCBC_Circle.kml",
                              layer = "VaseuxLakeCBC_Circle",
                              quiet = TRUE)

## Reproject to EPSG:4326
OliverOsoyoosCBC_wgs84 <- sf::st_transform(OliverOsoyoosCBC, 4326)
VaseuxLakeCBC_wgs84 <- sf::st_transform(VaseuxLakeCBC, 4326)
```

Dave does not have a circle for apex so I'm approximating the centroid of the circles on the Birds Canada CBC map and rolling with that.

```
## Set the centroid (Note: this is approximate)
ApexCentroid <- sf::st_sfc(
  sf::st_point(c(-119.911581, 49.308790)), # lon, lat
  crs = 4326)

## Project to EPSG:3005 = BC Albers
ApexCentroid_proj <- sf::st_transform(ApexCentroid, 3005)
```

```
## Make the circle (12km radius)
ApexCBCcircle <- sf::st_buffer(ApexCentroid_proj, dist = 12000)

## Reproject to EPSG:4326
ApexCBCcircle_wgs84 <- sf::st_transform(ApexCBCcircle, 4326)
```

Data in Collections

Odonate data from Gibson et al (2024)

Now that we have the circles lets bring in the digitized Odonata holdings of the BC Entomology Collections from Gibson et al (2024; Can. Entomol. 156(e42): 1–16. doi:10.4039/tce.2024.38).

```
## Read in Gibson data
GibsonDF <- read.csv("data/GibsonData/Gibson2024CanEntSup002.csv")

## Remove rows with NA Lat Long data
GibsonDF <- GibsonDF %>%
  dplyr::filter(!is.na(Longitude), !is.na(Latitude))

## Restrict the dataset to just BC records
GibsonBC <- GibsonDF %>%
  dplyr::filter(Prov_State == "British Columbia")

## Finally we need to make sure that the Gibson data are in the right projection
GibsonBC_sf <- GibsonBC %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"),
    crs = 4326,
    remove = FALSE)
```

Now that we've got that sorted we can subset the data in the Gibson dataset to those records within these circles.

```
## Filter the data to the circles
ApexOdeRecords <- GibsonBC_sf %>%
  dplyr::filter(sf::st_intersects(., ApexCBCcircle_wgs84, sparse = FALSE)[, 1])
OliverOdeRecords <- GibsonBC_sf %>%
  dplyr::filter(sf::st_intersects(., OliverOsoyoosCBC_wgs84, sparse = FALSE)[, 1])
VaseuxOdeRecords <- GibsonBC_sf %>%
  dplyr::filter(sf::st_intersects(., VaseuxLakeCBC_wgs84, sparse = FALSE)[, 1])
```

Now we can do some quick summary stats to see how much data is in each dataset

```
## How many specimens in each circle
nrow(ApexOdeRecords)

## [1] 18
nrow(OliverOdeRecords)

## [1] 987
nrow(VaseuxOdeRecords)

## [1] 806

## How many species (ItemName) in each circle
length(unique(ApexOdeRecords$ItemName))
```

```
## [1] 4
length(unique(OliverOdeRecords$ItemName))

## [1] 50
length(unique(VaseuxOdeRecords$ItemName))

## [1] 49
## Now lets see how many of each species
ApexOdeRecSppCounts <- ApexOdeRecords %>%
  dplyr::group_by(ItemName) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))
OliverOdeRecSppCounts <- OliverOdeRecords %>%
  dplyr::group_by(ItemName) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))
VaseuxOdeRecSppCounts <- VaseuxOdeRecords %>%
  dplyr::group_by(ItemName) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))

## What are the top 5 spp and how many records do they have?
head(sf::st_drop_geometry(ApexOdeRecSppCounts))

## # A tibble: 4 x 2
##   ItemName          count
##   <chr>            <int>
## 1 Somatochlora semicircularis    6
## 2 Aeshna juncea                5
## 3 Leucorrhinia hudsonica        5
## 4 Somatochlora albicincta       2

head(sf::st_drop_geometry(OliverOdeRecSppCounts))

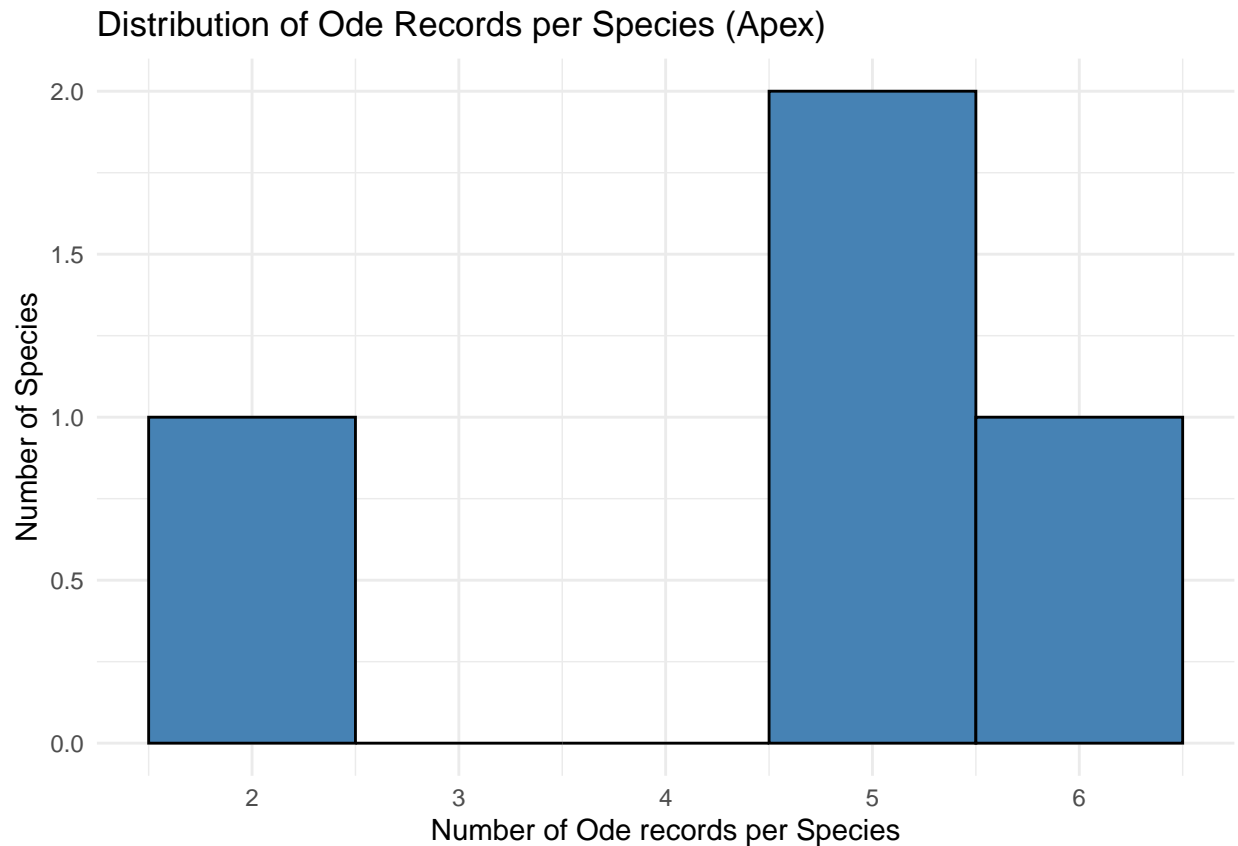
## # A tibble: 6 x 2
##   ItemName          count
##   <chr>            <int>
## 1 Enallagma annexum          101
## 2 Enallagma ebrium           65
## 3 Sympetrum occidentale       63
## 4 Enallagma carunculatum      53
## 5 Ischnura cervula            51
## 6 Sympetrum costiferum        46

head(sf::st_drop_geometry(VaseuxOdeRecSppCounts))

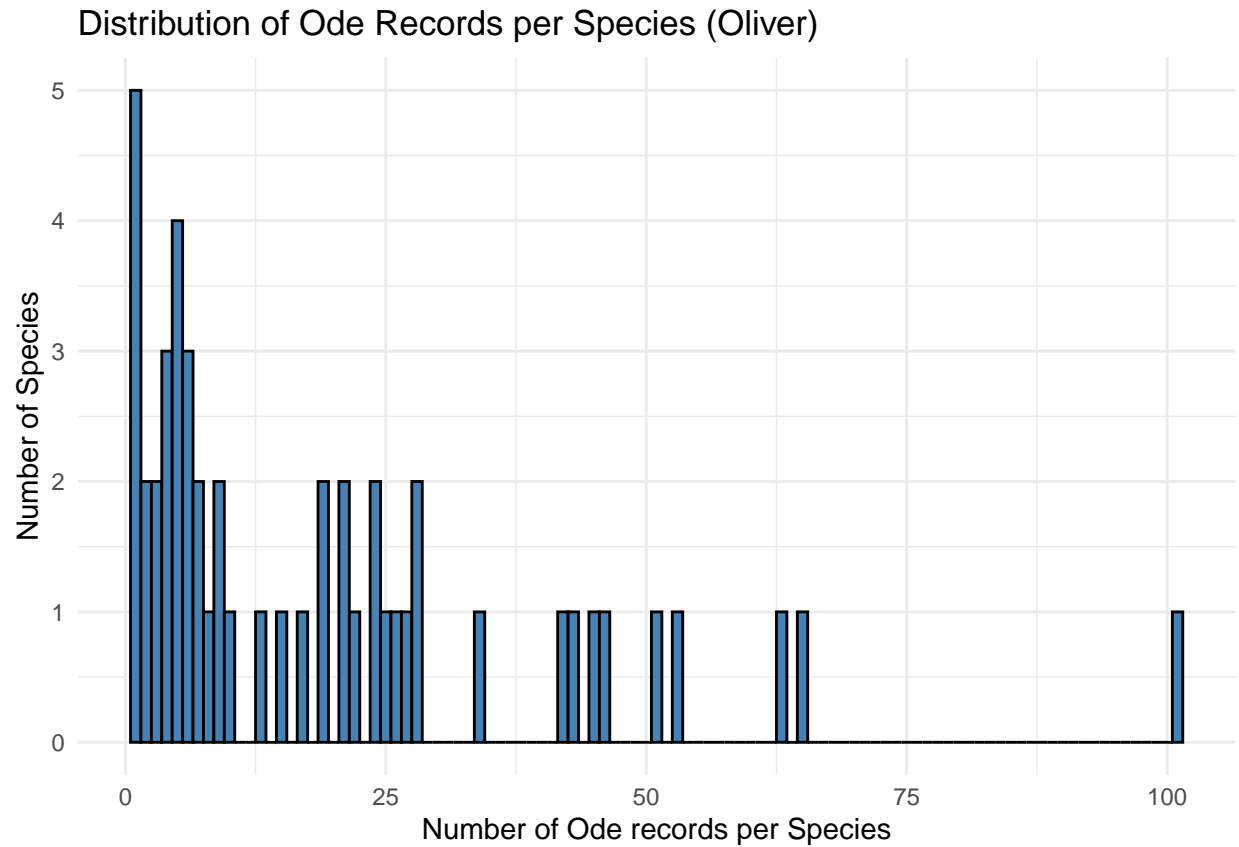
## # A tibble: 6 x 2
##   ItemName          count
##   <chr>            <int>
## 1 Ischnura cervula            73
## 2 Amphiagrion abbreviatum     61
## 3 Enallagma boreale           49
## 4 Enallagma annexum          39
## 5 Lestes disjunctus           38
## 6 Sympetrum obtrusum          37
```

How many species have 1 specimen, 2 specimens, etc?

```
ggplot(ApexOdeRecSppCounts, aes(x = count)) +  
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +  
  labs(  
    x = "Number of Ode records per Species",  
    y = "Number of Species",  
    title = "Distribution of Ode Records per Species (Apex)"  
  ) +  
  theme_minimal()
```

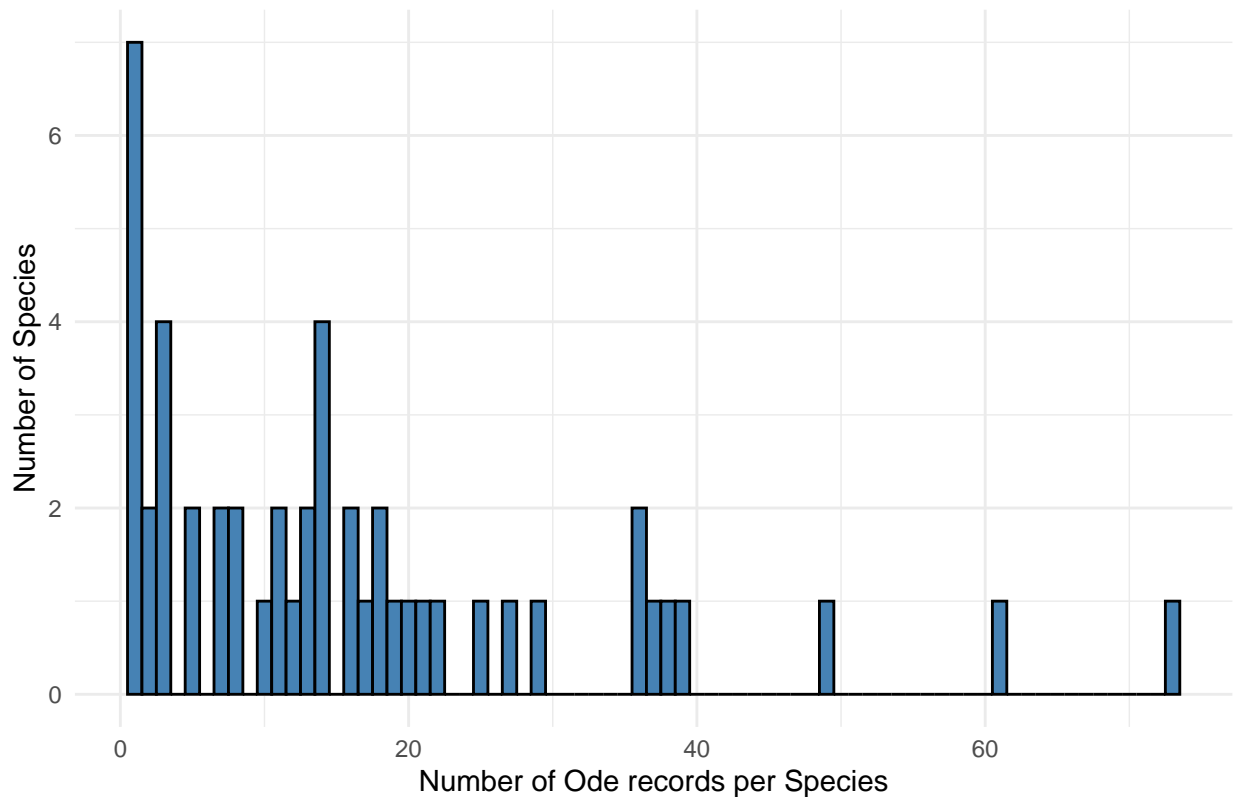


```
ggplot(OliverOdeRecSppCounts, aes(x = count)) +  
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +  
  labs(  
    x = "Number of Ode records per Species",  
    y = "Number of Species",  
    title = "Distribution of Ode Records per Species (Oliver)"  
  ) +  
  theme_minimal()
```



```
ggplot(VaseuxOdeRecSppCounts, aes(x = count)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  labs(
    x = "Number of Ode records per Species",
    y = "Number of Species",
    title = "Distribution of Ode Records per Species (Vaseux)"
  ) +
  theme_minimal()
```

Distribution of Ode Records per Species (Vaseux)



Now we know what's going on in the BC ode collections according to Gibson et al (2024) for these circles.

Butterfly data from GBIF

Now that we have the circles lets bring in the digitized butterfly data available on GBIF.

```
## Make a download request for GBIF data... I'm using an occ_download
## instead of an occ_search because i) it lets me pull multiple families
## and ii) it lets me exclude iNat data explicitly.
# dl <- rgbif::occ_download(
#   pred_in("taxonKey", c(6951,      ## Hedylidae
#                         6953,      ## Hesperidae
#                         5473,      ## Lycaenidae
#                         7017,      ## Nymphalidae
#                         9417,      ## Papilionidae
#                         5481,      ## Pieridae
#                         1933999    ## Riodinidae
#   )),
#   pred("GADM_GID", "CAN.2_1"),
#   pred("hasCoordinate", TRUE),
#   pred_not(
#     pred("publishingOrg", "28eb1a3f-1c15-4a95-931a-4af90ecb574d")
#   ),
#   format = "SIMPLE_CSV",
#   user = Sys.getenv("GBIF_USER"),
#   pwd = Sys.getenv("GBIF_PWD"),
#   email = Sys.getenv("GBIF_EMAIL"))
```

```

#
# ## Check the status of that download with
# occ_download_wait(dl)
#
# ## get some metadata
# meta <- occ_download_meta(dl)
#
# ## get that download
# d <- occ_download_get(dl) %>%
#   occ_download_import()
#
# ## Finally we need to make sure that the Gibson data are in the right projection
# GbifButtData_sf <- d %>%
#   dplyr::rename(Longitude = decimalLongitude,
#                 Latitude = decimalLatitude) %>%
#   sf::st_as_sf(coords = c("Longitude", "Latitude"),
#                crs = 4326,
#                remove = FALSE)
#
# ## Write this into the data
# saveRDS(GbifButtData_sf, "data/GbifData/GbifButtData_sf.rds")

## Read the data back in
GbifButtData <- readRDS("data/GbifData/GbifButtData_sf.rds")

## Remove obs that aren't IDed to spp
GbifButtData <- GbifButtData %>%
  dplyr::filter(!is.na(species) & species != "")

```

The GBIF download DOI is meta\$doi.

Now that we've got that sorted we can subset the data in the GBIF dataset to those records within these circles.

```

## Filter the data to the circles
ApexButtRecords <- GbifButtData %>%
  dplyr::filter(sf::st_intersects(., ApexCBCcircle_wgs84, sparse = FALSE)[, 1])
OliverButtRecords <- GbifButtData %>%
  dplyr::filter(sf::st_intersects(., OliverOsoyoosCBC_wgs84, sparse = FALSE)[, 1])
VaseuxButtRecords <- GbifButtData %>%
  dplyr::filter(sf::st_intersects(., VaseuxLakeCBC_wgs84, sparse = FALSE)[, 1])

```

Now we can do some quick summary stats to see how much data is in each dataset

```

## How many specimens in each circle
nrow(ApexButtRecords)

```

```
## [1] 1810
```

```
nrow(OliverButtRecords)
```

```
## [1] 5258
```

```
nrow(VaseuxButtRecords)
```

```
## [1] 1605
```

```

## How many species in each circle
length(unique(ApexButtRecords$species))

## [1] 104

length(unique(OliverButtRecords$species))

## [1] 132

length(unique(VaseuxButtRecords$species))

## [1] 101

## Now lets see how many of each species
ApexButtRecSppCounts <- ApexButtRecords %>%
  dplyr::group_by(species) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))
OliverButtRecSppCounts <- OliverButtRecords %>%
  dplyr::group_by(species) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))
VaseuxButtRecSppCounts <- VaseuxButtRecords %>%
  dplyr::group_by(species) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))

## What are the top 5 spp and how many records do they have?
head(sf::st_drop_geometry(ApexButtRecSppCounts))

## # A tibble: 6 x 2
##   species      count
##   <chr>      <int>
## 1 Parnassius smintheus    313
## 2 Erebia vidleri        151
## 3 Parnassius clodius     148
## 4 Occidryas anicia      136
## 5 Erebia epipsodea      115
## 6 Hesperia comma         63

head(sf::st_drop_geometry(OliverButtRecSppCounts))

## # A tibble: 6 x 2
##   species      count
##   <chr>      <int>
## 1 Pieris rapae          519
## 2 Coenonympha tullia    417
## 3 Colias philodice      314
## 4 Ochloides sylvanoides  265
## 5 Strymon melinus       213
## 6 Icaricia icarioides   168

head(sf::st_drop_geometry(VaseuxButtRecSppCounts))

## # A tibble: 6 x 2
##   species      count
##   <chr>      <int>

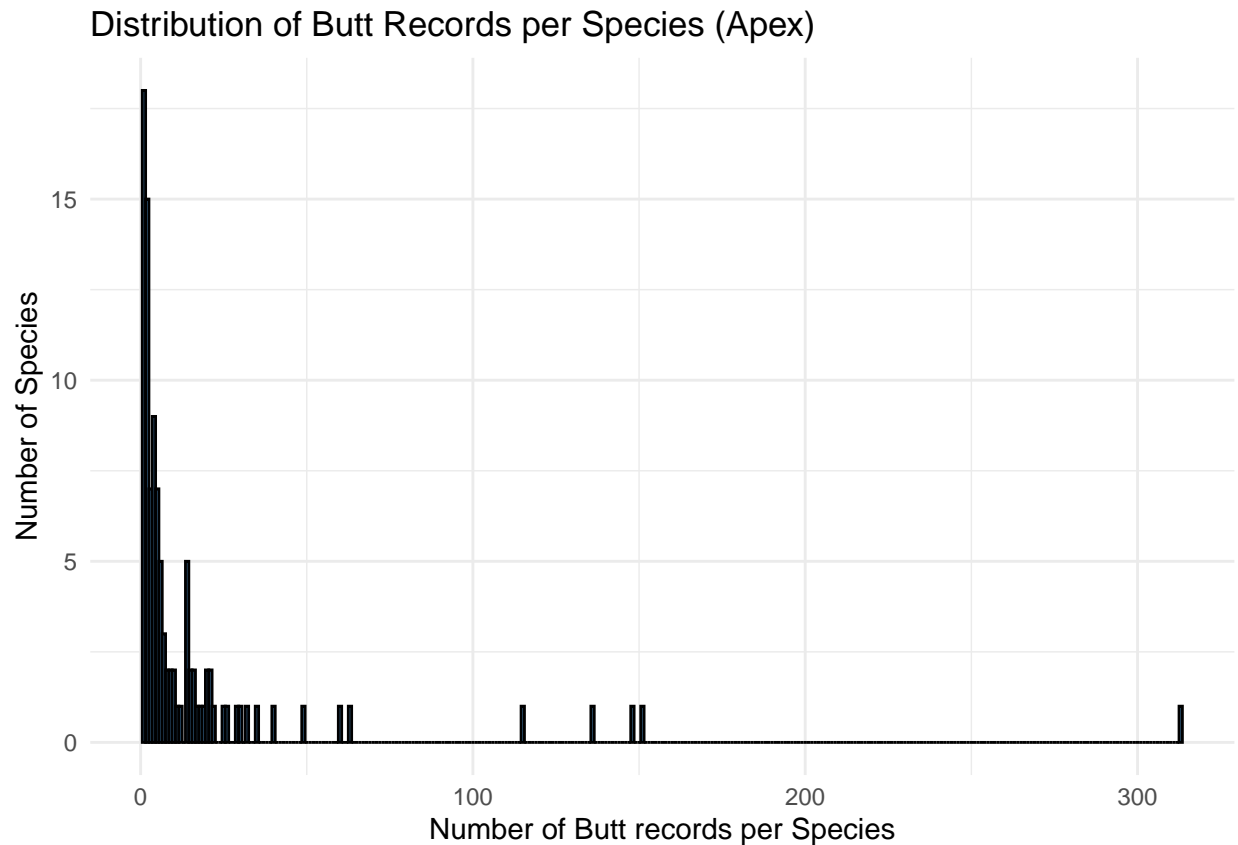
```



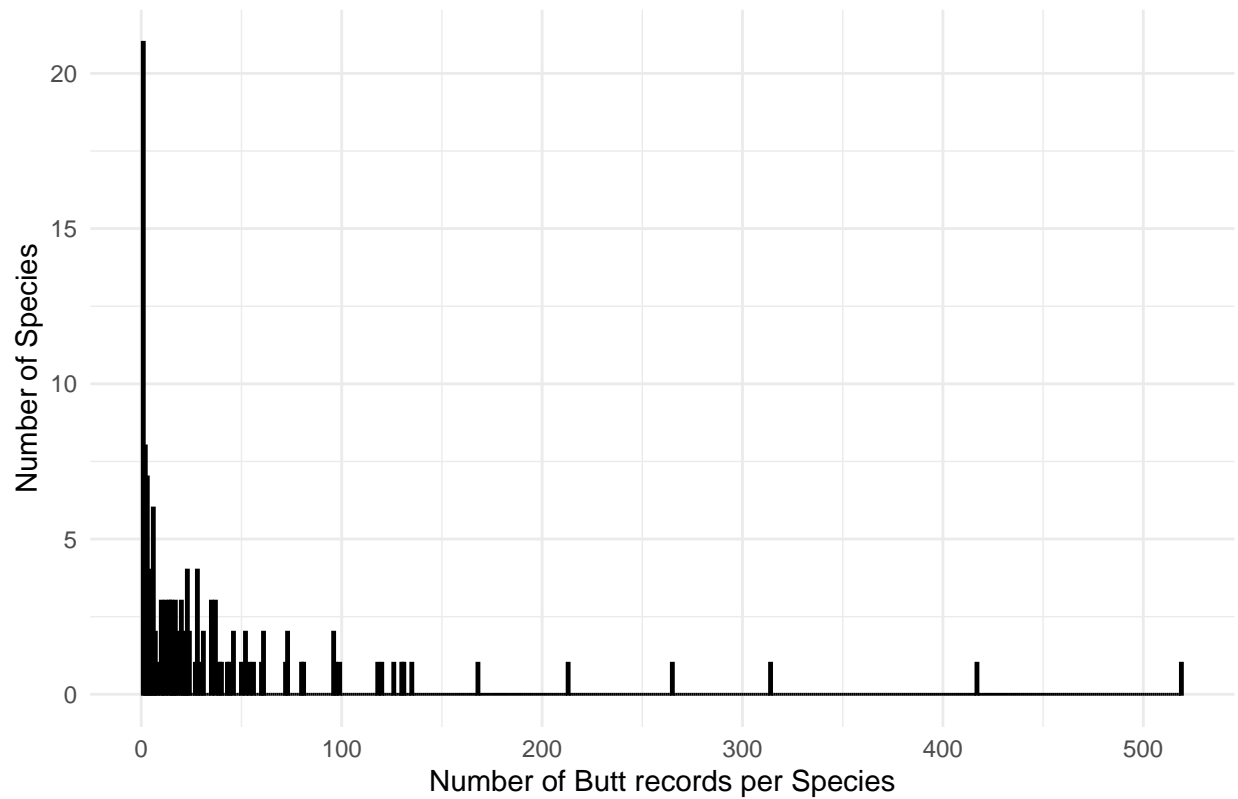
```
## 1 Pieris rapae      205
## 2 Coenonympha tullia 170
## 3 Ochlodes sylvanoides 152
## 4 Colias philodice  103
## 5 Nymphalis antiopa  95
## 6 Erebia epipsodea   64
```

How many species have 1 specimen, 2 specimens, etc?

```
ggplot(ApexButtRecSppCounts, aes(x = count)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  labs(
    x = "Number of Butt records per Species",
    y = "Number of Species",
    title = "Distribution of Butt Records per Species (Apex)"
  ) +
  theme_minimal()
```

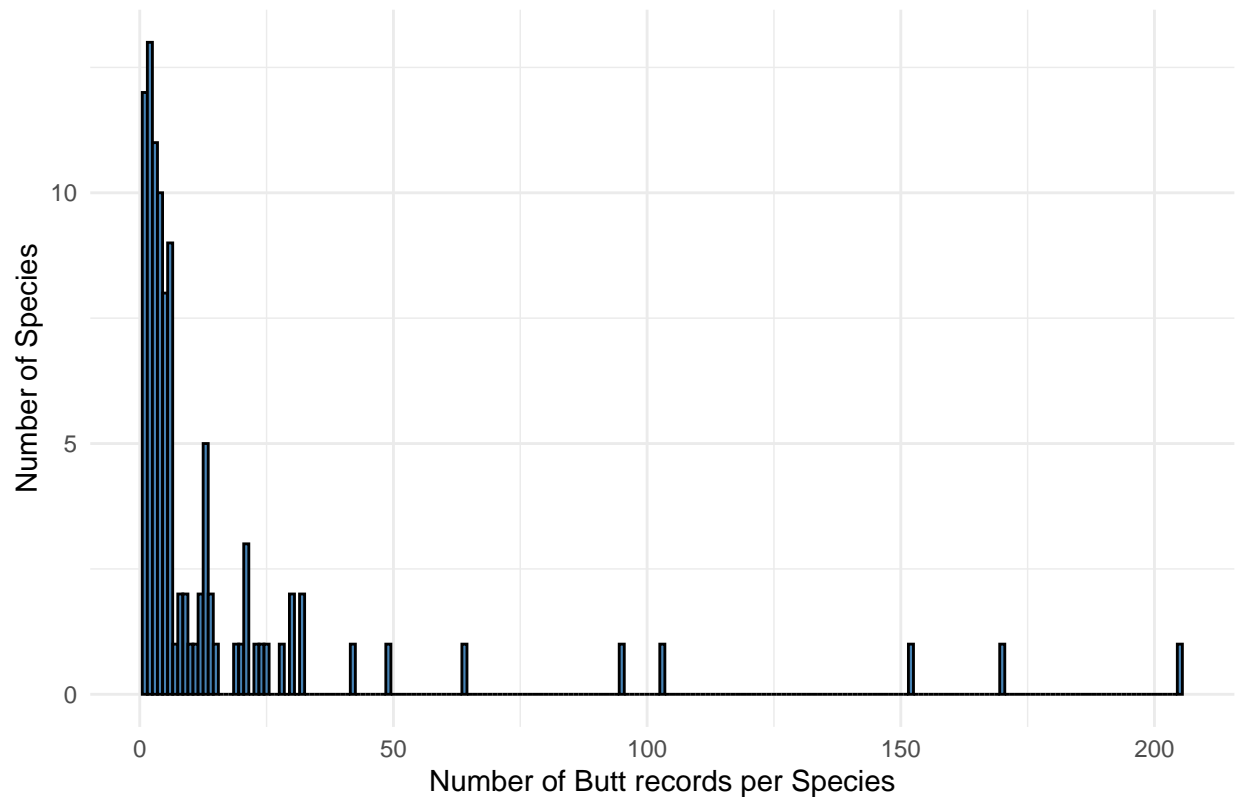


Distribution of Butt Records per Species (Oliver)



```
ggplot(VaseuxButtRecSppCounts, aes(x = count)) +  
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +  
  labs(  
    x = "Number of Butt records per Species",  
    y = "Number of Species",  
    title = "Distribution of Butt Records per Species (Vaseux)"  
  ) +  
  theme_minimal()
```

Distribution of Butt Records per Species (Vaseux)



Now we know what's going on in the BC Butt collections according to GBIF for these circles.

Data on iNaturalist

Lets look at iNat to see how many records there are in the iNat dataset for Odonata and Papilionoidea. This dataset is much younger but will likely have more recent data. The data we'll use were downloaded from iNaturalist through the Export function on 2026-01-13. Note, some of these records are obscured... we'll start by not removing them but keep that in mind.

Odonate iNat data

Now we'll read in the iNat data.

```
## Read in the data
iNatOdeDF <- read.csv("data/iNatData/iNatOdesBC2026JAN13.csv")

## Keep only research grade obs
iNatOdeRG <- iNatOdeDF %>%
  dplyr::filter(quality_grade == "research")

## Keep only CC0, CC-BY, CC-BY-NC records so you don't go to jail for 1B years
iNatOdeRG_CC <- iNatOdeRG %>%
  dplyr::filter(license %in% c("CC0", "CC-BY", "CC-BY-NC"))

## Remove read.csv2()## Remove all data with big (>5km) or NA accuracy
iNatOdeRG_CC_acc <- iNatOdeRG_CC %>%
  dplyr::filter(positional_accuracy < 5000)
```

```
## Change the Lat and long columns to Lat and Long
iNatOdeRG_CC_acc <- iNatOdeRG_CC_acc %>%
  dplyr::rename("Longitude" = "longitude") %>%
  dplyr::rename("Latitude" = "latitude")

## Finally we need to make sure that the iNat data are in the right projection
iNatOde_sf <- iNatOdeRG_CC_acc %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"),
    crs = 4326,
    remove = FALSE)
```

Now that we have the iNat data read in and cleaned lets filter to the same circles as before.

```
## Filter the data to the circles
ApexOdeObs <- iNatOde_sf %>%
  dplyr::filter(sf::st_intersects(., ApexCBCcircle_wgs84, sparse = FALSE)[, 1])
OliverOdeObs <- iNatOde_sf %>%
  dplyr::filter(sf::st_intersects(., OliverOsoyoosCBC_wgs84, sparse = FALSE)[, 1])
VaseuxOdeObs <- iNatOde_sf %>%
  dplyr::filter(sf::st_intersects(., VaseuxLakeCBC_wgs84, sparse = FALSE)[, 1])
```

Now we can do some quick summary stats to see how much data is in each dataset

```
## How many specimens in each circle
nrow(ApexOdeObs)

## [1] 2
nrow(OliverOdeObs)

## [1] 358
nrow(VaseuxOdeObs)

## [1] 461

## How many species (scientific_name) in each circle
length(unique(ApexOdeObs$scientific_name))

## [1] 1
length(unique(OliverOdeObs$scientific_name))

## [1] 40
length(unique(VaseuxOdeObs$scientific_name))

## [1] 44
```

```
## Now lets see how many of each species
ApexOdeObsSppCounts <- ApexOdeObs %>%
  dplyr::group_by(scientific_name) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))
OliverOdeObsSppCounts <- OliverOdeObs %>%
  dplyr::group_by(scientific_name) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))
VaseuxOdeObsSppCounts <- VaseuxOdeObs %>%
```

```
dplyr::group_by(scientific_name) %>%
dplyr::summarise(count = n(), .groups = "drop") %>%
dplyr::arrange(dplyr::desc(count))

## What are the top 5 spp and how many records do they have?
head(sf::st_drop_geometry(ApexOdeObsSppCounts))
```

```
## # A tibble: 1 x 2
##   scientific_name    count
##   <chr>             <int>
## 1 Libellula forensis      2

head(sf::st_drop_geometry(OliverOdeObsSppCounts))
```

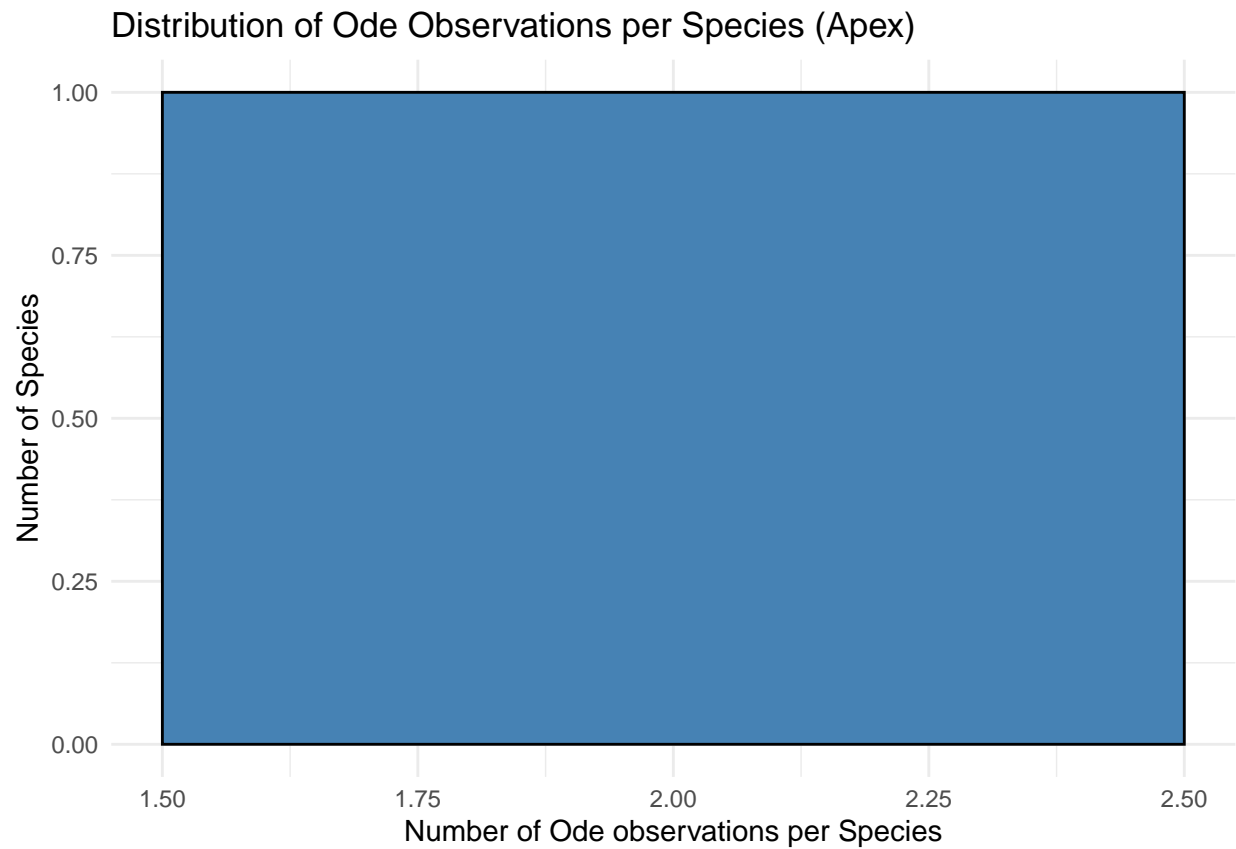
```
## # A tibble: 6 x 2
##   scientific_name    count
##   <chr>             <int>
## 1 Argia emma         56
## 2 Libellula forensis  41
## 3 Libellula pulchella 30
## 4 Sympetrum corruptum 23
## 5 Ischnura cervula   18
## 6 Pachydiplax longipennis 16

head(sf::st_drop_geometry(VaseuxOdeObsSppCounts))
```

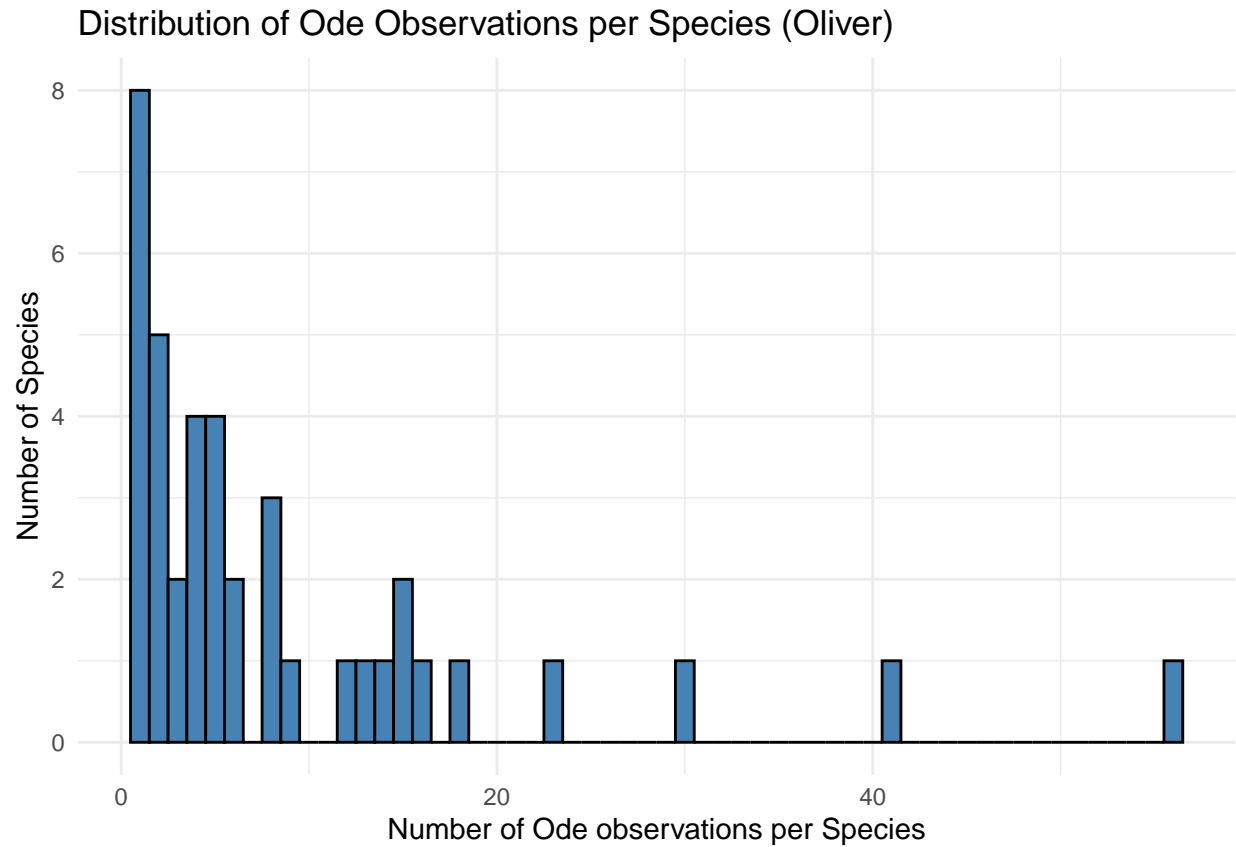
```
## # A tibble: 6 x 2
##   scientific_name    count
##   <chr>             <int>
## 1 Libellula forensis    71
## 2 Rhionaeschna californica 42
## 3 Ischnura cervula     36
## 4 Libellula quadrimaculata 33
## 5 Argia emma           25
## 6 Aeshna palmata       24
```

How many species have 1 specimen, 2 specimens, etc?

```
ggplot(ApexOdeObsSppCounts, aes(x = count)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  labs(
    x = "Number of Ode observations per Species",
    y = "Number of Species",
    title = "Distribution of Ode Observations per Species (Apex)"
  ) +
  theme_minimal()
```

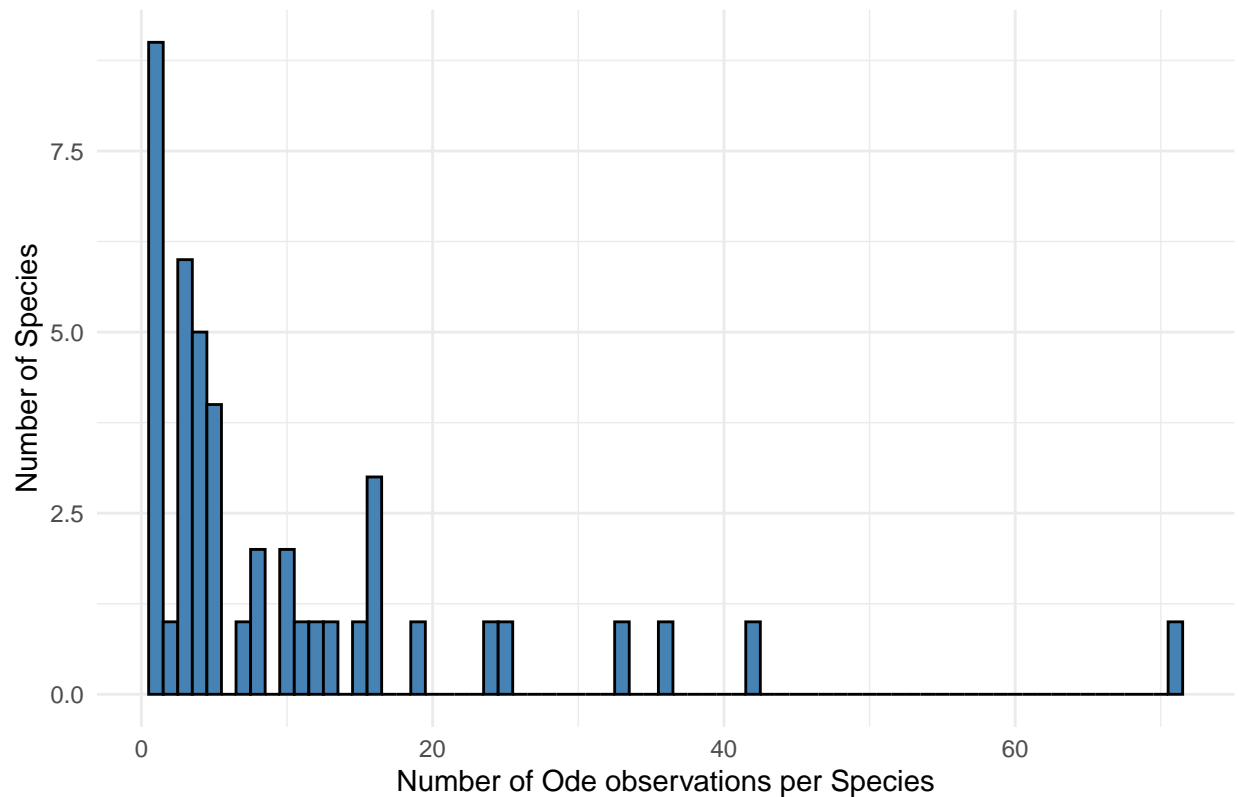


```
ggplot(OliverOdeObsSppCounts, aes(x = count)) +  
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +  
  labs(  
    x = "Number of Ode observations per Species",  
    y = "Number of Species",  
    title = "Distribution of Ode Observations per Species (Oliver)"  
  ) +  
  theme_minimal()
```



```
ggplot(VaseuxOdeObsSppCounts, aes(x = count)) +  
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +  
  labs(  
    x = "Number of Ode observations per Species",  
    y = "Number of Species",  
    title = "Distribution of Ode Observations per Species (Vaseux)"  
  ) +  
  theme_minimal()
```

Distribution of Ode Observations per Species (Vaseux)



Now that we've done that we can see that the two datasets are (at least) superficially similar for Odonates.

Butterfly iNat data

Now we'll read in the iNat data.

```
## Read in the data
iNatButtDF <- read.csv("data/iNatData/iNatButtsBC2026JAN13.csv")

## Keep only research grade obs
iNatButtRG <- iNatButtDF %>%
  dplyr::filter(quality_grade == "research")

## Keep only CC0, CC-BY, CC-BY-NC records so you don't go to jail for 1B years
iNatButtRG_CC <- iNatButtRG %>%
  dplyr::filter(license %in% c("CC0", "CC-BY", "CC-BY-NC"))

## Remove read.csv2()## Remove all data with big (>5km) or NA accuracy
iNatButtRG_CC_acc <- iNatButtRG_CC %>%
  dplyr::filter(positional_accuracy < 5000)

## Change the Lat and long columns to Lat and Long
iNatButtRG_CC_acc <- iNatButtRG_CC_acc %>%
  dplyr::rename("Longitude" = "longitude") %>%
  dplyr::rename("Latitude" = "latitude")

## Finally we need to make sure that the iNat data are in the right projection
```



```
iNatButt_sf <- iNatButtRG_CC_acc %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"),
    crs = 4326,
    remove = FALSE)
```

Now that we have the iNat data read in and cleaned lets filter to the same circles as before.

```
## Filter the data to the circles
ApexButtObs <- iNatButt_sf %>%
  dplyr::filter(sf::st_intersects(., ApexCBCcircle_wgs84, sparse = FALSE)[, 1])
OliverButtObs <- iNatButt_sf %>%
  dplyr::filter(sf::st_intersects(., OliverOsoyoosCBC_wgs84, sparse = FALSE)[, 1])
VaseuxButtObs <- iNatButt_sf %>%
  dplyr::filter(sf::st_intersects(., VaseuxLakeCBC_wgs84, sparse = FALSE)[, 1])
```

Now we can do some quick summary stats to see how much data is in each dataset

```
## How many specimens in each circle
nrow(ApexButtObs)

## [1] 433
nrow(OliverButtObs)

## [1] 2864
nrow(VaseuxButtObs)

## [1] 2642
## How many species (scientific_name) in each circle
length(unique(ApexButtObs$scientific_name))

## [1] 83
length(unique(OliverButtObs$scientific_name))

## [1] 102
length(unique(VaseuxButtObs$scientific_name))

## [1] 90
## Now lets see how many of each species
ApexButtObsSppCounts <- ApexButtObs %>%
  dplyr::group_by(scientific_name) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))
OliverButtObsSppCounts <- OliverButtObs %>%
  dplyr::group_by(scientific_name) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))
VaseuxButtObsSppCounts <- VaseuxButtObs %>%
  dplyr::group_by(scientific_name) %>%
  dplyr::summarise(count = n(), .groups = "drop") %>%
  dplyr::arrange(dplyr::desc(count))

## What are the top 5 spp and how many records do they have?
head(sf::st_drop_geometry(ApexButtObsSppCounts))
```

```
## # A tibble: 6 x 2
##   scientific_name      count
##   <chr>              <int>
## 1 Icaricia icarioides    34
## 2 Ochloides sylvanoides  24
## 3 Tharsalea helloides   21
## 4 Polygonia gracilis    20
## 5 Limenitis lorquini    13
## 6 Papilio rutulus       13
```

```
head(sf::st_drop_geometry(OliverButtObsSppCounts))
```

```
## # A tibble: 6 x 2
##   scientific_name      count
##   <chr>              <int>
## 1 Icaricia icarioides   381
## 2 Euphilotes heracleoides 177
## 3 Glaucopsyche lygdamus  176
## 4 Coenonympha californica 116
## 5 Satyrium behrii       111
## 6 Strymon melinus        94
```

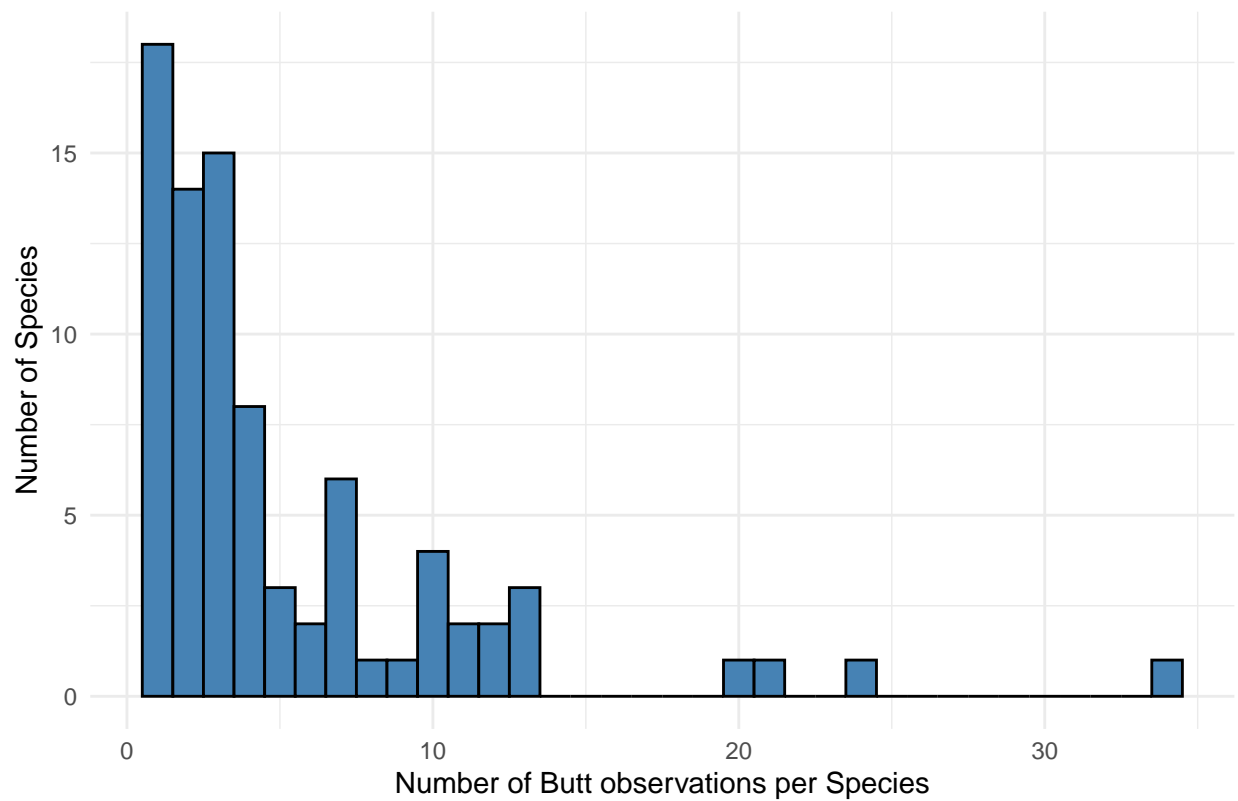
```
head(sf::st_drop_geometry(VaseuxButtObsSppCounts))
```

```
## # A tibble: 6 x 2
##   scientific_name      count
##   <chr>              <int>
## 1 Strymon melinus       418
## 2 Satyrium behrii      295
## 3 Icaricia icarioides  162
## 4 Ochloides sylvanoides 141
## 5 Papilio rutulus      127
## 6 Pieris rapae         111
```

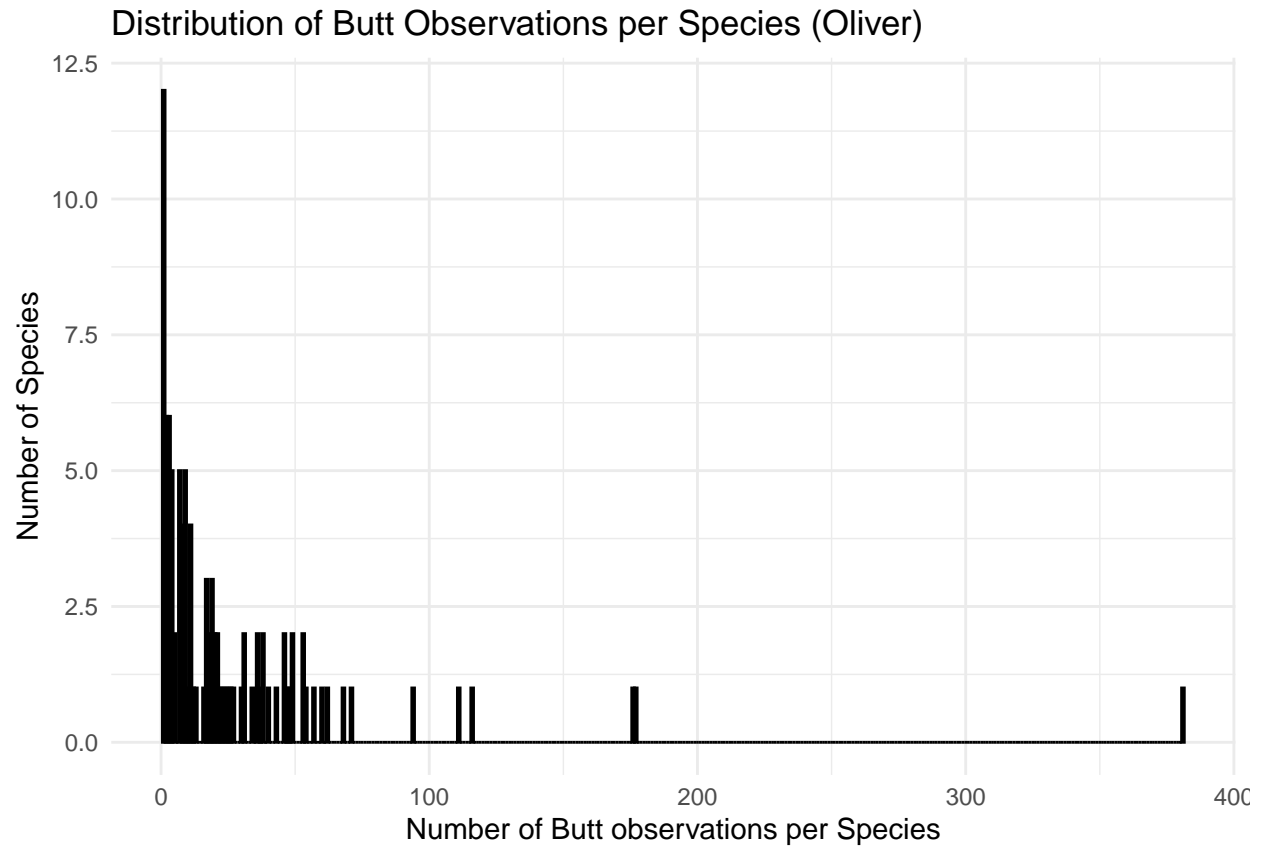
How many species have 1 specimen, 2 specimens, etc?

```
ggplot(ApexButtObsSppCounts, aes(x = count)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  labs(
    x = "Number of Butt observations per Species",
    y = "Number of Species",
    title = "Distribution of Butt Observations per Species (Apex)"
  ) +
  theme_minimal()
```

Distribution of Butt Observations per Species (Apex)

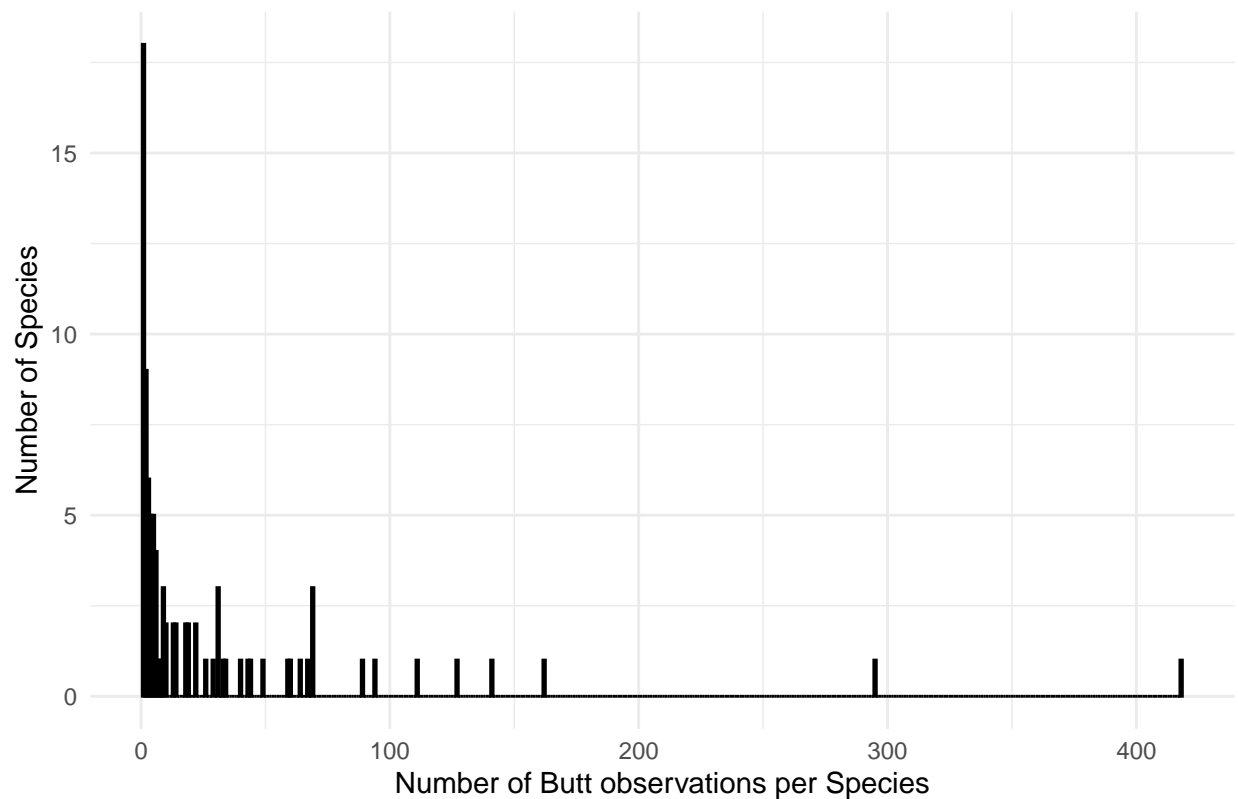


```
ggplot(OliverButtObsSppCounts, aes(x = count)) +  
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +  
  labs(  
    x = "Number of Butt observations per Species",  
    y = "Number of Species",  
    title = "Distribution of Butt Observations per Species (Oliver)"  
  ) +  
  theme_minimal()
```



```
ggplot(VaseuxButtObsSppCounts, aes(x = count)) +  
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +  
  labs(  
    x = "Number of Butt observations per Species",  
    y = "Number of Species",  
    title = "Distribution of Butt Observations per Species (Vaseux)"  
  ) +  
  theme_minimal()
```

Distribution of Butt Observations per Species (Vaseux)



```

## Odonata iNat N Observations
nrow(ApexOdeObs)

## [1] 2
nrow(OliverOdeObs)

## [1] 358
nrow(VaseuxOdeObs)

## [1] 461
## Papilionoidea iNat N Observations
nrow(ApexButtObs)

## [1] 433
nrow(OliverButtObs)

## [1] 2864
nrow(VaseuxButtObs)

## [1] 2642
## Odonata iNat N Species
length(unique(ApexOdeObs$scientific_name))

## [1] 1
length(unique(OliverOdeObs$scientific_name))

## [1] 40
length(unique(VaseuxOdeObs$scientific_name))

## [1] 44
## Papilionoidea iNat N Species
length(unique(ApexButtObs$scientific_name))

## [1] 83
length(unique(OliverButtObs$scientific_name))

## [1] 102
length(unique(VaseuxButtObs$scientific_name))

## [1] 90
## Odonata Museum N Species
length(unique(ApexOdeRecords$ItemName))

## [1] 4
length(unique(OliverOdeRecords$ItemName))

## [1] 50
length(unique(VaseuxOdeRecords$ItemName))

## [1] 49

```

```

## Papilionoidea Museum N Species
length(unique(ApexButtRecords$species))

## [1] 104
length(unique(OliverButtRecords$species))

## [1] 132
length(unique(VaseuxButtRecords$species))

## [1] 101
summary_df <- data.frame(
  Site = c("Apex", "Oliver", "Vaseux"),

  ## Odonata - Museum
  Odonata_Museum_Specimens = c(
    nrow(ApexOdeRecords),
    nrow(OliverOdeRecords),
    nrow(VaseuxOdeRecords)
  ),
  Odonata_Museum_Species = c(
    length(unique(ApexOdeRecords$ItemName)),
    length(unique(OliverOdeRecords$ItemName)),
    length(unique(VaseuxOdeRecords$ItemName))
  ),

  ## Odonata - iNaturalist
  Odonata_iNat_Observations = c(
    nrow(ApexOdeObs),
    nrow(OliverOdeObs),
    nrow(VaseuxOdeObs)
  ),
  Odonata_iNat_Species = c(
    length(unique(ApexOdeObs$scientific_name)),
    length(unique(OliverOdeObs$scientific_name)),
    length(unique(VaseuxOdeObs$scientific_name))
  ),

  ## Papilionoidea - Museum
  Papilionoidea_Museum_Specimens = c(
    nrow(ApexButtRecords),
    nrow(OliverButtRecords),
    nrow(VaseuxButtRecords)
  ),
  Papilionoidea_Museum_Species = c(
    length(unique(ApexButtRecords$species)),
    length(unique(OliverButtRecords$species)),
    length(unique(VaseuxButtRecords$species))
  ),

  ## Papilionoidea - iNaturalist
  Papilionoidea_iNat_Observations = c(
    nrow(ApexButtObs),
    nrow(OliverButtObs),

```

```

    nrow(VaseuxButtObs)
  ),
  Papilionoidea_iNat_Species = c(
    length(unique(ApexButtObs$scientific_name)),
    length(unique(OliverButtObs$scientific_name)),
    length(unique(VaseuxButtObs$scientific_name))
  ),

  stringsAsFactors = FALSE
)

## Flip the table so it's more readable
summary_df_all <- summary_df |>
  tibble::column_to_rownames("Site") |>
  t() |>
  as.data.frame()

summary_df_spp <- summary_df_all %>%
  tibble::rownames_to_column("metric") %>%
  dplyr::filter(grepl("Species", metric)) %>%
  tibble::column_to_rownames("metric")

summary_df_obs <- summary_df_all %>%
  tibble::rownames_to_column("metric") %>%
  dplyr::filter(!grepl("Species", metric)) %>%
  tibble::column_to_rownames("metric")

knitr::kable(
  summary_df_all,
  caption = "Summary of Odonata and Papilionoidea records and observations by site",
  booktabs = TRUE,
  longtable = FALSE
)

```

Table 1: Summary of Odonata and Papilionoidea records and observations by site

	Apex	Oliver	Vaseux
Odonata_Museum_Specimens	18	987	806
Odonata_Museum_Species	4	50	49
Odonata_iNat_Observations	2	358	461
Odonata_iNat_Species	1	40	44
Papilionoidea_Museum_Specimens	1810	5258	1605
Papilionoidea_Museum_Species	104	132	101
Papilionoidea_iNat_Observations	433	2864	2642
Papilionoidea_iNat_Species	83	102	90

```

knitr::kable(
  summary_df_spp,
  caption = "Summary of Odonata and Papilionoidea species by site",
  booktabs = TRUE,
  longtable = FALSE
)

```


Table 2: Summary of Odonata and Papilionoidea species by site

	Apex	Oliver	Vaseux
Odonata_Museum_Species	4	50	49
Odonata_iNat_Species	1	40	44
Papilionoidea_Museum_Species	104	132	101
Papilionoidea_iNat_Species	83	102	90

```
knitr::kable(
  summary_df_obs,
  caption = "Summary of Odonata and Papilionoidea records by site",
  booktabs = TRUE,
  longtable = FALSE
)
```

Table 3: Summary of Odonata and Papilionoidea records by site

	Apex	Oliver	Vaseux
Odonata_Museum_Specimens	18	987	806
Odonata_iNat_Observations	2	358	461
Papilionoidea_Museum_Specimens	1810	5258	1605
Papilionoidea_iNat_Observations	433	2864	2642