Data cleaning and management document for Guppy Network Infection Experiment (GNIE)

David R. Clark

2024-07-16

```
# Load in packages Load in libraries for analysis and data wrangling
# Visualization
library(ggplot2)
library(visreg)
source("http://highstat.com/Books/BGS/GAMM/RCodeP2/HighstatLibV6.R")
# (generalized) Linear mixed modeling
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(glmmTMB)
```

```
## Warning in checkDepPackageVersion(dep_pkg = "TMB"): Package version inconsistency detected.
## glmmTMB was built with TMB version 1.9.10
## Current TMB version is 1.9.11
## Please re-install glmmTMB from source or restore original 'TMB' package (see '?reinstalling' for mor
```

```
# Statistcal analysis reporting and model validation
library(performance)
library(car)
```

```
## Loading required package: carData
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(DHARMa)
```

```
## This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')
```

```r
# Data wrangling
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## v readr     2.1.5
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x plyr::arrange()   masks dplyr::arrange()
## x purrr::compact()  masks plyr::compact()
## x plyr::count()     masks dplyr::count()
## x plyr::desc()      masks dplyr::desc()
## x tidyr::expand()   masks Matrix::expand()
## x plyr::failwith()  masks dplyr::failwith()
## x dplyr::filter()   masks stats::filter()
## x plyr::id()        masks dplyr::id()
## x dplyr::lag()      masks stats::lag()
## x plyr::mutate()    masks dplyr::mutate()
## x tidyr::pack()     masks Matrix::pack()
## x dplyr::recode()   masks car::recode()
## x plyr::rename()    masks dplyr::rename()
## x purrr::some()     masks car::some()
## x plyr::summarise() masks dplyr::summarise()
## x plyr::summarize() masks dplyr::summarize()
## x tidyr::unpack()   masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidylog)
```

```
##
## Attaching package: 'tidylog'
##
## The following objects are masked from 'package:tidyr':
##
##     drop_na, fill, gather, pivot_longer, pivot_wider, replace_na,
##     spread, uncount
##
## The following objects are masked from 'package:plyr':
##
##     count, mutate, rename, summarise, summarize
##
## The following objects are masked from 'package:dplyr':
##
##     add_count, add_tally, anti_join, count, distinct, distinct_all,
##     distinct_at, distinct_if, filter, filter_all, filter_at, filter_if,
##     full_join, group_by, group_by_all, group_by_at, group_by_if,
##     inner_join, left_join, mutate, mutate_all, mutate_at, mutate_if,
##     relocate, rename, rename_all, rename_at, rename_if, rename_with,
##     right_join, sample_frac, sample_n, select, select_all, select_at,
##     select_if, semi_join, slice, slice_head, slice_max, slice_min,
##     slice_sample, slice_tail, summarise, summarise_all, summarise_at,
##     summarise_if, summarize, summarize_all, summarize_at, summarize_if,
##     tally, top_frac, top_n, transmute, transmute_all, transmute_at,
##     transmute_if, ungroup
##
## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(readr)
```

```r
# Loading in data frame with important details
PopContacts <- read.csv("GNIE_Contacts_Worm_Counts_20240717.csv")
View(PopContacts)

# renaming some of the variables we will be working with
PopContacts <- PopContacts %>%
    rename(day = experiment_day, fishID = fishBOverallID, Sex = fishBSex,
        worms = fishBWormCount, InfectionTrt = Infection) %>%
    mutate(ContactInitR = contacts_B_init/frame_num)
```

```
## rename: renamed 5 variables (day, fishID, Sex, worms, InfectionTrt)

## mutate: new variable 'ContactInitR' (double) with 15,520 unique values and 0% NA
```

```r
# Creating some new variables to use in our analysis

# Create a new variable indicating whether each individual is an
# index or not
PopContacts <- PopContacts %>%
    mutate(Index = case_when(fishID == 10 ~ 1, fishID < 10 ~ 0))
```

```
## mutate: new variable 'Index' (double) with 3 unique values and <1% NA
```

```r
# Subsetting data frame down to days 1,2,and 3
PopContactsDay1_3 <- PopContacts %>%
    filter(day == 1 | day == 2 | day == 3)
```

```
## filter: removed 21,186 rows (71%), 8,712 rows remaining
```

```r
# Subsetting and calculating total contacts
PopContactssum <- PopContactsDay1_3 %>%
    # group by population, day, and fishID for calculation
group_by(population, day, fishID) %>%
    # calculate the contacts and include details needed for analysis
dplyr::summarise(TotalContactR = sum(total_num_contacts)/frame_num, Contactinit = sum(ContactInitR),
    Sex = unique(Sex), worms = unique(worms), IndexWorm = unique(IndexWorm),
    InfectionTrt = unique(InfectionTrt)) %>%
    distinct(fishID, .keep_all = TRUE)
```

```
## group_by: 3 grouping variables (population, day, fishID)

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'population', 'day', 'fishID'. You can
## override using the `.groups` argument.

## distinct (grouped): removed 8,327 rows (96%), 385 rows remaining
```

```
# Create the contacts without the index included
PopContactswoI <- PopContactsDay1_3 %>%
    filter(day == 3 & Index == 0 & fishAOverallID != 10) %>%
    group_by(population, day, fishID) %>%
    mutate(ContactwoI = sum(ContactInitR)) %>%
    select(population, fishID, ContactwoI) %>%
    distinct(fishID, .keep_all = TRUE) %>%
    ungroup() %>%
    select(-c(day))
```

```
## filter: removed 6,136 rows (70%), 2,576 rows remaining

## group_by: 3 grouping variables (population, day, fishID)

## mutate (grouped): new variable 'ContactwoI' (double) with 126 unique values and 0% NA

## Adding missing grouping variables: `day`
## select: dropped 20 variables (fishAOverallID, vid, fishAID, fishBID,
## total_num_contacts, ...)
## distinct (grouped): removed 2,450 rows (95%), 126 rows remaining
## ungroup: no grouping variables
## select: dropped one variable (day)
```

```
# Merge this column back into the main dataset
PopContactssum2 <- merge(PopContactssum, PopContactswoI, by = c("population",
    "fishID"), .keep = all)

PopContactssumCh <- PopContactssum2 %>%
    filter(day == 2 | day == 3) %>%
    select(-c(worms, InfectionTrt, ContactwoI)) %>%
    group_by(population, fishID) %>%
    pivot_wider(names_from = day, values_from = c(TotalContactR, Contactinit)) %>%
    mutate(TotalCRCh = TotalContactR_3 - TotalContactR_2, CRinitCh = Contactinit_3 -
        Contactinit_2) %>%
    select(-c(TotalContactR_3, TotalContactR_2, Contactinit_3, Contactinit_2,
        Sex, IndexWorm))
```

```
## filter: removed 117 rows (32%), 252 rows remaining
## select: dropped 3 variables (worms, InfectionTrt, ContactwoI)
## group_by: 2 grouping variables (population, fishID)
## pivot_wider: reorganized (day, TotalContactR, Contactinit) into (TotalContactR_3, TotalContactR_2, C
## mutate (grouped): new variable 'TotalCRCh' (double) with 126 unique values and 0% NA
##                   new variable 'CRinitCh' (double) with 126 unique values and 0% NA
## select: dropped 6 variables (Sex, IndexWorm, TotalContactR_3, TotalContactR_2, Contactinit_3, ...)
```

```r
PopContactssum2 <- merge(PopContactssum2, PopContactssumCh, by = c("population",
    "fishID"), .keep = all)


# Quickly view to make sure everything looks good
# View(PopContactssum2)
```

```r
# Calculating index contacts
PopContactsDay3 <- PopContacts %>%
    # first filter down to day 3 and only contacts with index
filter(day == 3 & fishAOverallID == 10) %>%
    # group for calculation of index contacts
group_by(population, fishID) %>%
    # calculate contacts with index during each video during day 3
    # and sum them together
mutate(IndexContact_init = sum(ContactInitR))
```

```
## filter: removed 29,575 rows (99%), 323 rows remaining

## group_by: 2 grouping variables (population, fishID)

## mutate (grouped): new variable 'IndexContact_init' (double) with 126 unique values and 0% NA
```

```r
# Create a new variable for infection status of each individaul
PopContactsDay3 <- PopContactsDay3 %>%
    mutate(Infectionstat = case_when(worms >= 1 ~ 1, worms < 1 ~ 0, is.na(worms) ~
        0))
```

```
## mutate (grouped): new variable 'Infectionstat' (double) with 2 unique values and 0% NA
```

```r
PopContactsDay3 <- PopContactsDay3 %>%
    select(population, fishID, IndexContact_init, Infectionstat) %>%
    # subsetting down to one row per individual
distinct(fishID, .keep_all = TRUE)
```

```
## select: dropped 21 variables (day, fishAOverallID, vid, fishAID, fishBID, ...)

## distinct (grouped): removed 197 rows (61%), 126 rows remaining
```

```r
# Adding index individuals back in since the step before will remove
# them during merge
PopContactssum10 <- PopContactssum %>%
    # filtering down to index only
filter(fishID == 10) %>%
    # create the index column with NAs
mutate(IndexContact_init = NA, ContactwoI = NA, TotalCRCh = NA, CRinitCh = NA)
```

```
## filter (grouped): removed 370 rows (96%), 15 rows remaining

## mutate (grouped): new variable 'IndexContact_init' (logical) with one unique value and 100% NA
```

```
##                   new variable 'ContactwoI' (logical) with one unique value and 100% NA

##                   new variable 'TotalCRCh' (logical) with one unique value and 100% NA

##                   new variable 'CRinitCh' (logical) with one unique value and 100% NA
```

```
# Creating the same metric for index fish
PopContactssum10 <- PopContactssum10 %>%
    mutate(Infectionstat = case_when(worms >= 1 ~ 1, worms < 1 ~ 0, is.na(worms) ~
        0))
```

```
## mutate (grouped): new variable 'Infectionstat' (double) with 2 unique values and 0% NA
```

```
# Merge the index contact data frame back to main dataframe for
# analysis
PopContactssum3 <- merge(PopContactssum2, PopContactsDay3, by = c("population",
    "fishID"), .keep = all)



# Bring index individuals back into the main dataframe
PopContactssum4 <- rbind(PopContactssum3, PopContactssum10)


# Create a new variable for worm contacts (the number of contacts
# multiplied by the number of worms)
PopContactssum4 <- PopContactssum4 %>%
    mutate(wormcontact = IndexWorm * IndexContact_init)
```

```
## mutate: new variable 'wormcontact' (double) with 91 unique values and 32% NA
```

```
# Create a new variable indicating whether each individual is an
# index or not
PopContactssum4 <- PopContactssum4 %>%
    mutate(Index = case_when(fishID == 10 ~ 1, fishID != 10 ~ 0))
```

```
## mutate: new variable 'Index' (double) with 2 unique values and 0% NA
```

```
# Setting data as factors
PopContactssum4$population <- as.factor(PopContactssum4$population)
PopContactssum4$fishID <- as.factor(PopContactssum4$fishID)
PopContactssum4$day <- as.factor(PopContactssum4$day)
PopContactssum4$Sex <- as.factor(PopContactssum4$Sex)
PopContactssum4$Index <- as.factor(PopContactssum4$Index)
PopContactssum4$InfectionTrt <- as.factor(PopContactssum4$InfectionTrt)
PopContactssum4$Infectionstat <- as.factor(PopContactssum4$Infectionstat)

# summary checking the data
summary(PopContactssum2)
```

```
##   population            fishID       day        TotalContactR
##  Length:369         Min.   :1   Min.   :1.000   Min.   :0.09345
##  Class :character   1st Qu.:3   1st Qu.:1.000   1st Qu.:0.81978
##  Mode  :character   Median :5   Median :2.000   Median :1.73817
##                     Mean   :5   Mean   :2.024   Mean   :2.07627
##                     3rd Qu.:7   3rd Qu.:3.000   3rd Qu.:2.83747
##                     Max.   :9   Max.   :3.000   Max.   :7.54742
##
##   Contactinit          Sex                worms           IndexWorm
##  Min.   :0.01609   Length:369         Min.   : 0.0000   Min.   : 32.00
##  1st Qu.:0.19217   Class :character   1st Qu.: 0.0000   1st Qu.: 43.00
##  Median :0.39969   Mode  :character   Median : 0.0000   Median : 98.00
##  Mean   :0.48575                      Mean   : 0.9889   Mean   : 96.38
##  3rd Qu.:0.66836                      3rd Qu.: 1.0000   3rd Qu.:114.00
##  Max.   :1.84355                      Max.   :18.0000   Max.   :214.00
##                                       NA's   :279       NA's   :108
##   InfectionTrt      ContactwoI       TotalCRCh          CRinitCh
##  Min.   :0.0000   Min.   :0.0375   Min.   :-4.6390   Min.   :-0.97576
##  1st Qu.:0.0000   1st Qu.:0.1193   1st Qu.:-0.9972   1st Qu.:-0.22239
##  Median :0.0000   Median :0.2461   Median :-0.2696   Median :-0.01691
##  Mean   :0.2439   Mean   :0.2981   Mean   :-0.4181   Mean   :-0.08435
##  3rd Qu.:0.0000   3rd Qu.:0.3953   3rd Qu.: 0.2187   3rd Qu.: 0.07852
##  Max.   :1.0000   Max.   :1.1861   Max.   : 4.7608   Max.   : 0.41856
##
```

```
write.csv(PopContactssum4, "IndividualContacts_20240719.csv")
```