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
```

```
library(DHARMa)
```

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

```
# 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
```

```
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
```

```
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 error
```

```r
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
```

```
library(readr)
```

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

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

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

```
# Subsetting and calculating total contacts
PopContactssum <- PopContactsDay1_3 %>%
    # group by population, day, and fishID for calculation
group_by(population, experiment_day, fishBOverallID) %>%
    # calculate the contacts and include details needed for analysis
dplyr::summarise(TotalContacts = mean(total_num_contacts), Contactinit = sum(contacts_B_init),
    Sex = unique(fishBSex), worms = unique(fishBWormCount), IndexWorm = unique(IndexWorm),
    InfectionTrt = unique(Infection)) %>%
    # ungroup the factors
ungroup()
```

```
## group_by: 3 grouping variables (population, experiment_day, fishBOverallID)
```

```
## `summarise()` has grouped output by 'population', 'experiment_day'. You can
## override using the `.groups` argument.
## ungroup: no grouping variables
```

```
# rename some variables for ease during analysis step.
PopContactssum <- PopContactssum %>%
    rename(day = experiment_day, fishID = fishBOverallID)
```

```
## rename: renamed 2 variables (day, fishID)
```

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

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

```r
    # select relevant columns for merging
select(population, fishID, IndexContact_init) %>%
    # subsetting down to one row per individual
distinct(fishID, .keep_all = TRUE)
```

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

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

## mutate (grouped): new variable 'IndexContact_init' (integer) with 124 unique values and 0% NA

## rename: renamed one variable (fishID)

## select: dropped 19 variables (experiment_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)
```

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

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

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

# Bring index individuals back into the main dataframe
PopContactssum2 <- rbind(PopContactssum2, PopContactssum10)


# Create a new variable for infection status of each individaul
PopContactssum2 <- PopContactssum2 %>%
    mutate(Infectionstat = case_when(worms >= 1 ~ 1, worms < 1 ~ 0, is.na(worms) ~
        0))
```

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

```r
# Create a new variable for worm contacts (the number of contacts
# multiplied by the number of worms)
PopContactssum2 <- PopContactssum2 %>%
    mutate(wormcontact = IndexWorm * IndexContact_init)
```

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

```r
# Create a new variable indicating whether each individual is an
# index or not
PopContactssum2 <- PopContactssum2 %>%
    mutate(Index = case_when(fishID == 10 ~ 1, fishID < 10 ~ 0))
```

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

```r
# Setting data as factors
PopContactssum2$population <- as.factor(PopContactssum2$population)
PopContactssum2$fishID <- as.factor(PopContactssum2$fishID)
PopContactssum2$day <- as.factor(PopContactssum2$day)
PopContactssum2$Sex <- as.factor(PopContactssum2$Sex)
PopContactssum2$Index <- as.factor(PopContactssum2$Index)
PopContactssum2$InfectionTrt <- as.factor(PopContactssum2$InfectionTrt)
PopContactssum2$Infectionstat <- as.factor(PopContactssum2$Infectionstat)

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

```
##    population      fishID    day     TotalContacts      Contactinit    Sex
##  SBinf1 : 29   1      : 41   1:118   Min.   :  140.2   Min.   :  564   F:261
##  ADinf1 : 28   2      : 41   2:126   1st Qu.: 1345.4   1st Qu.: 6498   M:123
##  ADunf1 : 28   3      : 41   3:140   Median : 2504.1   Median :14092
##  GDinf1 : 28   4      : 41           Mean   : 3224.3   Mean   :16290
##  GZinf1 : 28   5      : 41           3rd Qu.: 4121.4   3rd Qu.:22714
##  GZunf1 : 28   6      : 41           Max.   :18095.5   Max.   :59441
##  (Other):215   (Other):138
##      worms           IndexWorm      InfectionTrt IndexContact_init Infectionstat
##  Min.   :  0.00   Min.   : 32.00   0:284        Min.   :   0      0:349
##  1st Qu.:  0.00   1st Qu.: 43.00   1:100        1st Qu.: 460      1: 35
##  Median :  0.00   Median : 98.00                Median : 926
##  Mean   : 10.16   Mean   : 96.09                Mean   :1597
##  3rd Qu.:  1.00   3rd Qu.:114.00                3rd Qu.:2241
##  Max.   :214.00   Max.   :214.00                Max.   :9444
##  NA’s   :284      NA’s   :112                   NA’s   :15
##   wormcontact       Index
##  Min.   :      0   0:369
##  1st Qu.:  23892   1: 15
##  Median :  89648
##  Mean   : 204171
##  3rd Qu.: 306384
##  Max.   :1074922
##  NA’s   :123
```

```r
write.csv(PopContactssum2, "IndividualContacts_20240729.csv")
```