

Manuscript

Savannah Hammerton

2022-11-30

```
# Load packages
library(tidyverse)
library(NatParksPalettes)
library(here)
library(gtsummary)
library(naniar)

# Set ggplot2 theme for whole script
ggplot2::theme_set(ggplot2::theme_linedraw())

# Set paths
here::here()

## [1] "/Users/savannahhammerton/Desktop/GitHub/EPID7500_Final_Project"
```

Data

Data import and basic info

```
# Load data
colony <- readr::read_rds(here::here("data/colony.rds"))
stressor <- readr::read_rds(here::here("data/stressor.rds"))

# Check out the data
dplyr::glimpse(colony)

## Rows: 1,222
## Columns: 10
## $ year          <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, ~
## $ months        <chr> "January-March", "January-March", "January-March", "Ja~
## $ state          <chr> "Alabama", "Arizona", "Arkansas", "California", "Color~
## $ colony_n       <dbl> 7000, 35000, 13000, 1440000, 3500, 3900, 305000, 10400~
## $ colony_max     <dbl> 7000, 35000, 14000, 1690000, 12500, 3900, 315000, 1050~
## $ colony_lost    <dbl> 1800, 4600, 1500, 255000, 1500, 870, 42000, 14500, 380~
## $ colony_lost_pct <dbl> 26, 13, 11, 15, 12, 22, 13, 14, 4, 4, 40, 22, 18, 23, ~
## $ colony_added   <dbl> 2800, 3400, 1200, 250000, 200, 290, 54000, 47000, 3400~
## $ colony_reno    <dbl> 250, 2100, 90, 124000, 140, NA, 25000, 9500, 760, 8000~
## $ colony_reno_pct <dbl> 4, 6, 1, 7, 1, NA, 8, 9, 7, 9, 4, 1, 2, 1, NA, 13, NA,~
dplyr::glimpse(stressor)

## Rows: 7,332
## Columns: 5
## $ year          <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, ~
```

```
## $ months      <chr> "January-March", "January-March", "January-March", "January~
## $ state       <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "Ala~
## $ stressor    <chr> "Varroa mites", "Other pests/parasites", "Disesesases", "Pest~
## $ stress_pct  <dbl> 10.0, 5.4, NA, 2.2, 9.1, 9.4, 26.9, 20.5, 0.1, NA, 1.8, 3.1~
```

Data processing

It looks like I could join the colony and stressor data sets into one, matching on `year`, `months`, and `state`. I'm going to do that now so I have one final data set to work with. I don't really want any data about stressors without any colony data, so I'm going to keep all the rows in the colony dataset, using `dplyr::left_join()`.

```
# Join data sets on year, months, and state, keeping all the rows in colony
savethebees <-
  dplyr::left_join(colony, stressor,
                   by = c("year", "months", "state"))
# Check out the new data set
dplyr::glimpse(savethebees)
```

```
## Rows: 7,332
## Columns: 12
## $ year      <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, ~
## $ months    <chr> "January-March", "January-March", "January-March", "Ja~
## $ state     <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", ~
## $ colony_n  <dbl> 7000, 7000, 7000, 7000, 7000, 7000, 35000, 35000, 3500~
## $ colony_max <dbl> 7000, 7000, 7000, 7000, 7000, 7000, 35000, 35000, 3500~
## $ colony_lost <dbl> 1800, 1800, 1800, 1800, 1800, 1800, 4600, 4600, 4600, ~
## $ colony_lost_pct <dbl> 26, 26, 26, 26, 26, 26, 13, 13, 13, 13, 13, 13, 11, 11~
## $ colony_added <dbl> 2800, 2800, 2800, 2800, 2800, 2800, 3400, 3400, 3400, ~
## $ colony_reno <dbl> 250, 250, 250, 250, 250, 250, 2100, 2100, 2100, 2100, ~
## $ colony_reno_pct <dbl> 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 1, 1, 1, 1, 1, ~
## $ stressor   <chr> "Varroa mites", "Other pests/parasites", "Disesesases", ~
## $ stress_pct <dbl> 10.0, 5.4, NA, 2.2, 9.1, 9.4, 26.9, 20.5, 0.1, NA, 1.8~
```

Since I now have the same number of rows in my final dataset that I did in the stressor dataset (which had exactly six times the number of rows in the colony data set), I can conclude that there are six stressors observed, and the data collectors included all of the six stressors for each year/location combination. I want to break those stressors into indicator/dummy variables so I can explore them a little more in depth. I will use `tidyr::pivot_wider()`, pulling names from the stressors themselves, and values from the percentage of colonies affected by the stressors during that time period in that location. When I do that, I will also rename the two stressors that have spaces in the names, and use `janitor::clean_names()` to make sure every variable is lower case.

```
# Pivot stressor data wider and rename variables with spaces
savethebees_wide <-
  savethebees |>
  dplyr::mutate(stressor = ifelse(stressor == "Disesesases", "Diseases", stressor)) |>
  tidyr::pivot_wider(names_from = stressor, values_from = stress_pct) |>
  dplyr::rename(varroa_mites = `Varroa mites`,
                other_pests_parasites = `Other pests/parasites`) |>
  janitor::clean_names()

# Check out new data
dplyr::glimpse(savethebees_wide)
```

```
## Rows: 1,222
## Columns: 16
```

```
## $ year          <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, ~
## $ months        <chr> "January-March", "January-March", "January-March~
## $ state         <chr> "Alabama", "Arizona", "Arkansas", "California", ~
## $ colony_n      <dbl> 7000, 35000, 13000, 1440000, 3500, 3900, 305000,~
## $ colony_max    <dbl> 7000, 35000, 14000, 1690000, 12500, 3900, 315000~
## $ colony_lost   <dbl> 1800, 4600, 1500, 255000, 1500, 870, 42000, 1450~
## $ colony_lost_pct <dbl> 26, 13, 11, 15, 12, 22, 13, 14, 4, 4, 40, 22, 18~
## $ colony_added  <dbl> 2800, 3400, 1200, 250000, 200, 290, 54000, 47000~
## $ colony_reno   <dbl> 250, 2100, 90, 124000, 140, NA, 25000, 9500, 760~
## $ colony_reno_pct <dbl> 4, 6, 1, 7, 1, NA, 8, 9, 7, 9, 4, 1, 2, 1, NA, 1~
## $ varroa_mites  <dbl> 10.0, 26.9, 17.6, 24.7, 14.6, 2.5, 22.3, 6.2, 38~
## $ other_pests_parasites <dbl> 5.4, 20.5, 11.4, 7.2, 0.9, 1.4, 13.5, 4.9, 37.7,~
## $ diseases      <dbl> NA, 0.1, 1.5, 3.0, 1.8, NA, 0.8, 3.3, 1.6, 12.5,~
## $ pesticides    <dbl> 2.2, NA, 3.4, 7.5, 0.6, NA, 8.9, 2.6, NA, 4.8, 0~
## $ other         <dbl> 9.1, 1.8, 1.0, 6.5, 2.6, 21.2, 5.1, 4.8, 2.0, 8.~
## $ unknown       <dbl> 9.4, 3.1, 1.0, 2.8, 5.9, 2.4, 4.4, 10.5, NA, 4.9~
```

The data set is now back to the number of rows in the initial colony data set, which makes sense as the stressor variables were what were making the dataset longer. I'm now going to check for missingness using `naniar::gg_miss_var()`, which will show me the number of data points missing for each variable.

```
# Create function to rename variables in long (initial) dataset
rename_long <- function(data) {
  data |>
    dplyr::rename(Year = year,
                  Quarter = months,
                  State = state,
                  `Number of colonies` = colony_n,
                  `Maximum colonies` = colony_max,
                  `Colonies lost` = colony_lost,
                  `Percentage of total colonies lost` = colony_lost_pct,
                  `Colonies added` = colony_added,
                  `Colonies renovated` = colony_reno,
                  `Percent of colonies renovated` = colony_reno_pct,
                  `Stress type` = stressor,
                  `Percent of colonies affected by stressor in quarter` =
                    stress_pct)
}

# Create function to rename variables in wide dataset
rename_wide <- function(data) {
  data |>
    dplyr::rename(Year = year,
                  Quarter = months,
                  State = state,
                  `Number of colonies` = colony_n,
                  `Maximum colonies` = colony_max,
                  `Colonies lost` = colony_lost,
                  `Percentage of total colonies lost` = colony_lost_pct,
                  `Colonies added` = colony_added,
                  `Colonies renovated` = colony_reno,
                  `Percent of colonies renovated` = colony_reno_pct,
                  `Varroa mites` = varroa_mites,
                  `Other pests and parasites` = other_pests_parasites,
                  `Diseases` = diseases,
```

```

    `Pesticides` = pesticides,
    `Other` = other,
    `Unknown` = unknown)
}

```

```

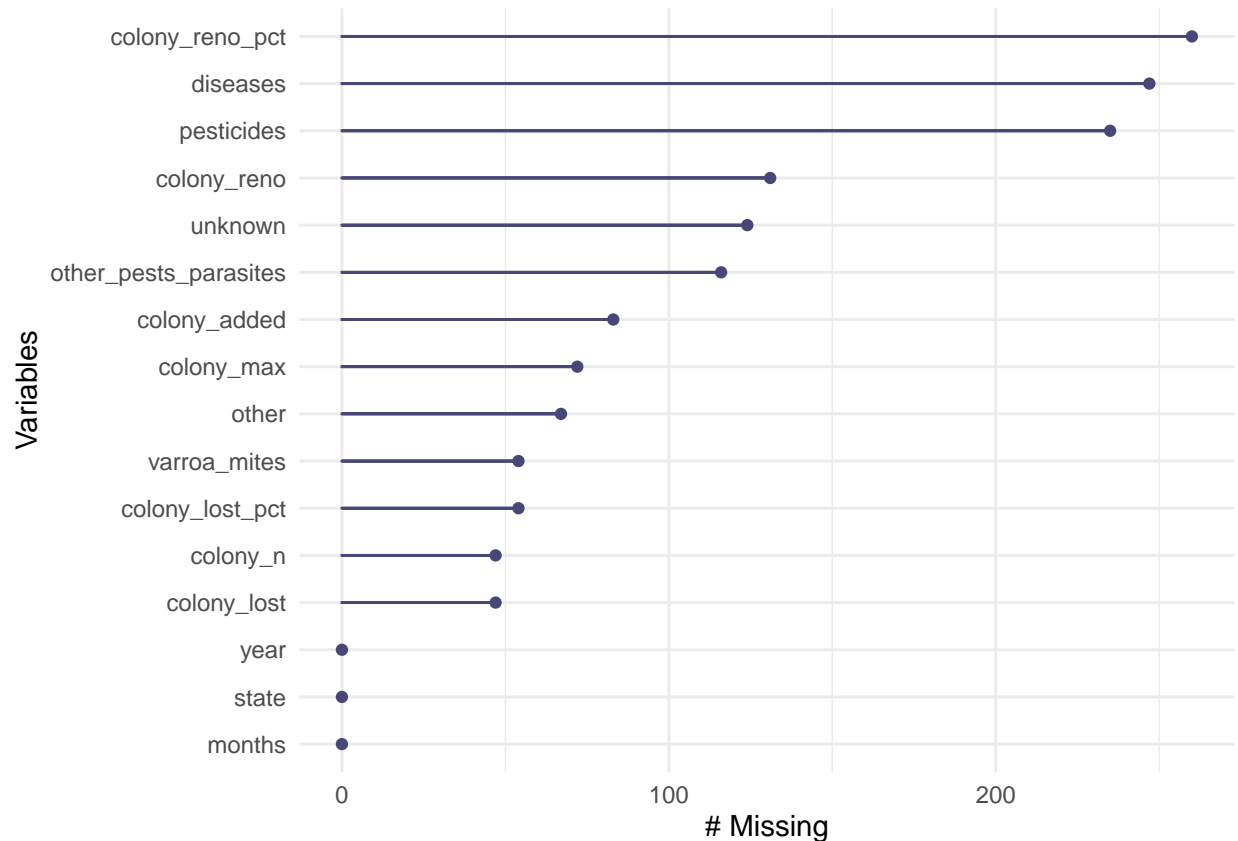
# Check for missingness
naniar::gg_miss_var(savethebees_wide)

```

```

## Warning: It is deprecated to specify 'guide = FALSE' to remove a guide. Please
## use 'guide = "none"' instead.

```



The variable with the most missing data points is `colony_reno_pct`, which is the percentage of colonies renovated. I'm going to check what the range is of that variable, and see if it's possible that the NA's actually just mean it should be zero. While I'm at it, I'm going to do this for all the variables since all but the identifier variables have some missing values.

```

# savethebees_wide |>
#   rename_wide() |>
#   dplyr::select(!c(Year, Quarter, State)) |>
#   dplyr::summarise(dplyr::across(.cols = everything(),
#                                   .fns = min, na.rm = TRUE)) |>
#   t() |>
#   as.data.frame() |>
#   gt::gt()

```