Project 1 Exploratory Analysis

Jack Cunningham

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

Loading Data

```r
data <- readxl::read_excel("Project2_S25_Data.xlsx")[,1:12]
```

```
New names:
* `` -> `...13`
* `` -> `...14`
* `` -> `...15`
* `` -> `...16`
* `` -> `...17`
* `` -> `...18`
* `` -> `...19`
* `` -> `...20`
* `` -> `...21`
```

```r
data
```

```
# A tibble: 1,508 x 12
   Tick_ID Deer_ID Vaccine_Group Infestation Mortality Weight_Replete_Female(g~1
   <chr>     <dbl> <chr>         <chr>           <dbl>                     <dbl>
 1 792_01      792 P             PRE                 0                    0.0682
 2 792_02      792 P             PRE                 0                    0.178
 3 792_03      792 P             PRE                 0                    0.035
 4 792_04      792 P             PRE                 0                    0.0368
 5 792_05      792 P             PRE                 1                    0.0263
 6 792_06      792 P             PRE                 0                    0.195
 7 792_07      792 P             PRE                 1                    0.0648
 8 792_08      792 P             PRE                 0                    0.0607
 9 792_09      792 P             PRE                 1                    0.0689
10 792_10      792 P             PRE                 0                    0.181
# i 1,498 more rows
# i abbreviated name: 1: `Weight_Replete_Female(g)`
# i 6 more variables: `Weight_ Egg_Mass(g)` <chr>,
#   `Bloodmeal_Conversion(%)` <chr>, `%_Hatched` <chr>, Num_Eggs_laid <chr>,
#   Num_Larvae <chr>, Comments <chr>
```

Missing Data

```r
na_count <- sapply(data, function(y) sum(length(which(is.na(y)))))
na_count
```

```
                 Tick_ID                  Deer_ID            Vaccine_Group
                       0                        0                        0
             Infestation                Mortality Weight_Replete_Female(g)
                       0                        0                        0
     Weight_ Egg_Mass(g)  Bloodmeal_Conversion(%)                %_Hatched
                       0                        0                        4
           Num_Eggs_laid               Num_Larvae                 Comments
                       5                        6                      878
```

We can see that there does not appear to be a lot of missing data on first glance, but after looking at the excel we can see that there does appear to be many "N/A" values. I also filter out the Comments column, because after reviewing the excel it seems it was only populated when there were extreme circumstances.

```
data_no_coms <- data[,1:11]
na_count_manual <- sapply(data_no_coms, function(y) sum(length(which(y == "N/A"))))
na_count_manual
```

```
                  Tick_ID                   Deer_ID             Vaccine_Group
                        0                         0                         0
              Infestation                 Mortality Weight_Replete_Female(g)
                        0                         0                         0
       Weight_ Egg_Mass(g)  Bloodmeal_Conversion(%)                  %_Hatched
                      552                       552                       560
             Num_Eggs_laid                Num_Larvae
                      557                       560
```

There appears to be a significant amount of data that is manually classified with "N/A". Lets see if they are often in the same row.

```
data$na_count <- apply(data_no_coms, 1, function(x) sum(x == "N/A"|is.na(x)))
table(data$na_count)
```

```
  0   1   2   3   5 
942   1   4   9 552 
```

We can see that almost always missing values are grouped in the same row.

Deer groups:

Client said there are three deer groups.

```
table(data$Deer_ID)
```

```
790 791 792 793 796 798 799 800 801 803 804 805 
 47  99 141 165 113  74 109 180 141 166 150 123 
```

We see 12 different Deer IDs, mostly evenly distributed data points. Will need to figure out how these are grouped. These deer have different vaccination statuses:

```
data |> group_by(Deer_ID, Vaccine_Group) |>
  summarise(n = n()) |>
  View()
```

```
`summarise()` has grouped output by 'Deer_ID'. You can override using the
`.groups` argument.
```

In some instances we have deer with the same ID but different vaccination status. This makes me question what the Deer ID variable means.

Vaccination Status

```
table(data$Vaccine_Group)
```

```

  C   H   L   P 
426 393 439 250 
```

There are four vaccine statuses. P - Pre-infested, C - Control, L - low, H - high.

Lets take another look at the Deer_ID/Vaccinate status combinations with that in mind:

```
data |> group_by(Deer_ID, Vaccine_Group) |>
  summarise(n = n()) |>
  View()
```

```
`summarise()` has grouped output by 'Deer_ID'. You can override using the
`.groups` argument.
```

We can see that the duplicates in Deer_ID/Vaccination Status are due to that pre-infested group.

Infestation

Next we look at the Infestation feature.

```r
table(data$Infestation)
```

```
 FIRST    PRE SECOND  THIRD
   478    250    476    304
```

We can see that the count decreases as the stage of infestation grows further, from 478 to 304.