

# Fit Thermal Models

Michael Gilchrist

date: 2022-10-18

## Goal

- Fit series of thermal models found in `rTPC` to data collected in the Derryberry lab.
- Initial analysis should be with GLM and mixture models.

## Set up

### Install libraries

```
## install packages user might not have by replacing FALSE with TRUE
if(FALSE) {
  BiocManager::install("mixOmics") ## needed by RVAideMemoire
  install.packages(c("RSQLite", "nls.multstart", "lme4", "RVAideMemoire"))
  ## Install the thermal curve package from git_hub, not cran
  remotes::install_github("padpadpadpad/rTPC")
}
```

```
## load libraries
library(stats)
require(MASS) # provides negative binomial fitting: glm.nb
```

## Loading required package: MASS

```
library(RSQLite) # Don't think we need this.
library(rTPC) ##
library(nls.multstart)
library(broom)
library(tidyverse)
```

```
## -- Attaching packages
## ----- tidyverse
## 1.3.2 --
```

```
## v ggplot2 3.3.6          v purrr 0.3.4
## v tibble 3.1.8           v dplyr 1.0.99.9000
```

```
## v tidyr 1.2.0.9000 v stringr 1.4.1
## v readr 2.1.2 v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflic
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
require(ggplot2)
require(ggpubr)
```

```
## Loading required package: ggpubr
```

```
require(viridisLite)
```

```
## Loading required package: viridisLite
```

```
#options(ggplot2.continuous.colour="viridis",
#       ggplot2.discrete.colour="viridis",
#       ggplot2.scale_fill_discrete = scale_fill_viridis_d,
#       ggplot2.scale_fill_continuous = scale_fill_viridis_c)
require(GGally)
```

```
## Loading required package: GGally
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg ggplot2
```

```
require(reshape2)
```

```
## Loading required package: reshape2
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##   smiths
```

```
require(lme4)
```

```
## Loading required package: lme4
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##   expand, pack, unpack
```

```
require(RVAideMemoire) # provides overdisp.glmmer()
```

```
## Loading required package: RVAideMemoire
## *** Package RVAideMemoire v 0.9-81-2 ***
##
## Attaching package: 'RVAideMemoire'
##
## The following object is masked from 'package:lme4':
##
##     dummy
##
## The following object is masked from 'package:broom':
##
##     bootstrap
```

## Load Data

```
## Read in ZEFI Data sets
## Treat 'repeatability' as round = 0
## Add round info

## Repeatability was done between round 1 and 2, female was present, but only one temp. so treating as

git_root <- system("git rev-parse --show-toplevel", intern = TRUE)

data_raw = list()

data_raw[[1]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Round-1-Heat-Trials.csv")) %>%
  ## Note T237 and T230 are missing numbers in the song_count column
  ## so we are filtering these observations out until they are found
  filter(!is.na(song_count))

data_raw[[2]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Repeatability-Song-Count.csv"))
  mutate(round = 2) %>%
  group_by(male) %>%
  mutate(test_order = rank(date)) %>%
  ungroup()

data_raw[[3]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Round-2-Heat-Trials.csv")) %>%
  mutate(round = 3) %>%
  # group_by(male) %>% mutate(total_count_round_3 = sum(song_count) )

## Join data and discard empty columns
data_full <- full_join(data_raw[[1]], data_raw[[2]]) %>%
  full_join(data_raw[[3]]) %>%
  discard(~all(is.na(.) | . == "")) %>% ## get rid of columns of only NA
  mutate(trial_completed = !(is.na(song_count)) ) %>%
  mutate(song_count = ifelse(is.na(song_count), 0, song_count)) %>%
  mutate(song_count = song_count*1.0) %>% ## convert to a double so it's not treated as an integer
  mutate(chamber = as.factor(chamber), male = as.factor(male)) %>%
  ## create a global variable trial_order based on individual rounds
```

```

mutate(trial_index = as.integer(round*10+test_order)) %>%
mutate(song_count_plus_1 = (song_count + 1)) %>%
mutate(log_song_count_plus_1 = log(song_count + 1)) %>%
mutate(temp_target = as.numeric(temp_target)) %>%
## Add column with total song_count for a given round
group_by(male, round ) %>% mutate(count_total_round = sum(song_count) ) %>%
ungroup() %>%
mutate() ## Dummy function so we can comment out lines above it w/o any issues

```

```

## Joining, by = c("male", "chamber", "date", "song_count", "counter", "test_order", "temp_target", "round")
## Joining, by = c("male", "chamber", "date", "song_count", "counter", "test_order", "temp_target", "round")

```

## Second Analysis for Liz

### Examine Data

```

data_count_total <- data_full %>% group_by(round) %>%
  select(male, round, count_total_round) %>%
  distinct() %>% mutate(filtered = FALSE)

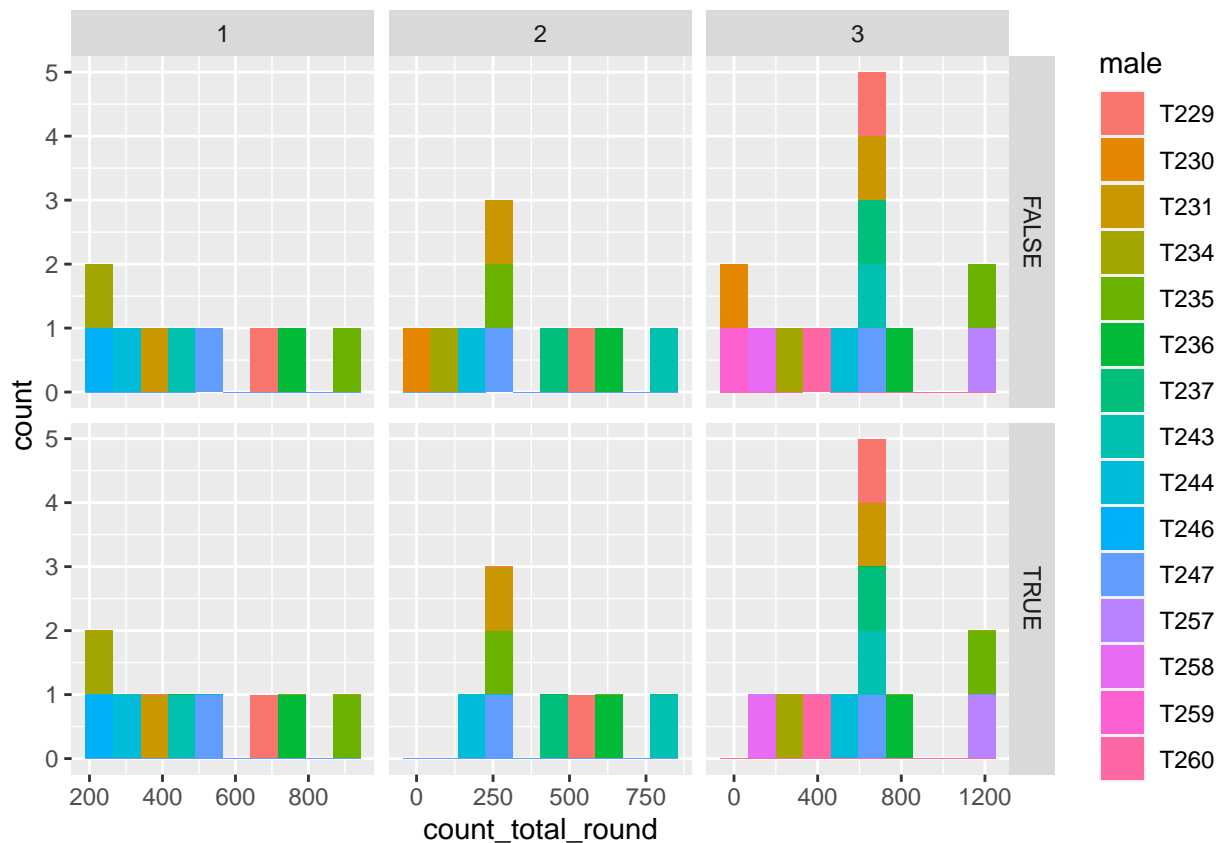
data_count_total_filter <- data_count_total %>%
  filter(count_total_round > 150) %>%
  mutate(filtered = TRUE)

data_tmp <- full_join(data_count_total, data_count_total_filter)

## Joining, by = c("male", "round", "count_total_round", "filtered")

t <- ggplot(data_tmp, aes(count_total_round, fill = male)) +
  geom_histogram(bins = 10) #+
# scale_x_log10()
hist_count_total <- t +
  facet_grid(cols = vars(round), rows = vars(filtered), scales = "free_x")
hist_count_total

```



```
data <- data_full %>% filter(round ==3)

data %>% group_by(temp_target) %>%
  summarize(ave = mean(song_count), sd = sd(song_count), n = length(song_count), n_completed = sum(tr
```

```
## # A tibble: 5 x 5
##   temp_target  ave    sd    n n_completed
##   <dbl> <dbl> <dbl> <int> <int>
## 1      27 124.  105.   24      24
## 2      35 103.   82.4   23      23
## 3      40 112.   84.6   53      53
## 4      42 112.   96.1   23      23
## 5      44  78.2  78.5   23      22
```

```
## Look at occurrence of males across rounds
```

```
tmp1 <- data_full %>%
  filter(round==1) %>%
  pull(male) %>%
  unique(); tmp1
```

```
## [1] T236 T243 T246 T247 T234 T231 T244 T235 T229
## 15 Levels: T229 T230 T231 T234 T235 T236 T237 T243 T244 T246 T247 T257 ... T260
```

```
tmp2 <- data_full %>%
  filter(round %in% c(2,3)) %>%
  pull(male) %>%
  unique() ; tmp2
```

```
## [1] T247 T236 T243 T234 T244 T231 T235 T229 T237 T230 T259 T258 T257 T260
## 15 Levels: T229 T230 T231 T234 T235 T236 T237 T243 T244 T246 T247 T257 ... T260
```

```
lapply(list(tmp1, tmp2, intersect(tmp1, tmp2)), length)
```

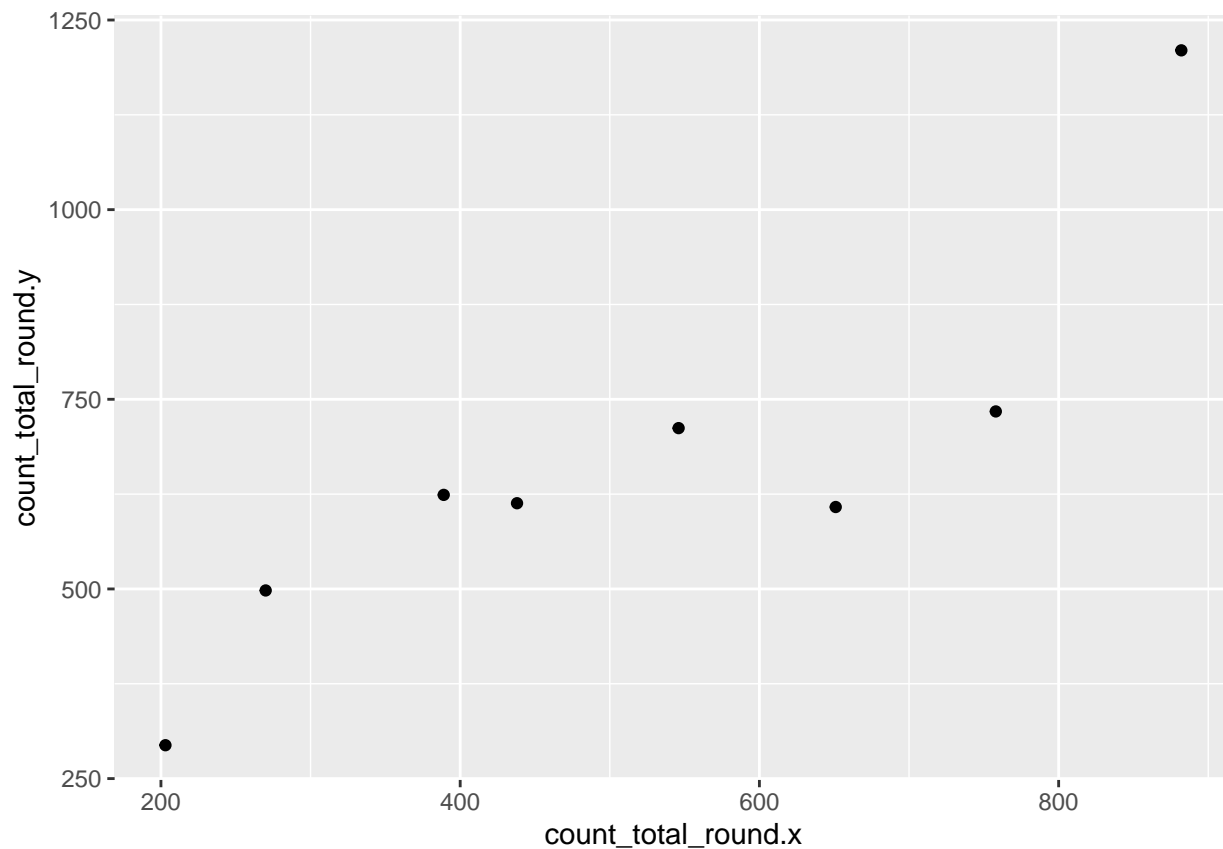
```
## [[1]]
## [1] 9
##
## [[2]]
## [1] 14
##
## [[3]]
## [1] 8
```

### Compare count\_total\_round between round 1 and 3

```
x <- data_count_total %>% filter(round ==1) %>% select(male, count_total_round, round)
y <- data_count_total %>% filter(round ==3) %>% select(male, count_total_round, round)

data_xy <- full_join(x = x, y = y, by = "male") %>% na.omit()

ggplot(data_xy) +
  aes(x = count_total_round.x, y = count_total_round.y) +
  geom_point()
```



## Result

- We see consistency between count\_total\_round which is good
- Note one male has trial\_completed = FALSE and, thus, temp\_mean = NA. In future will create new column temp\_working and use temp\_target for this instance and temp\_mean for remaining ones.

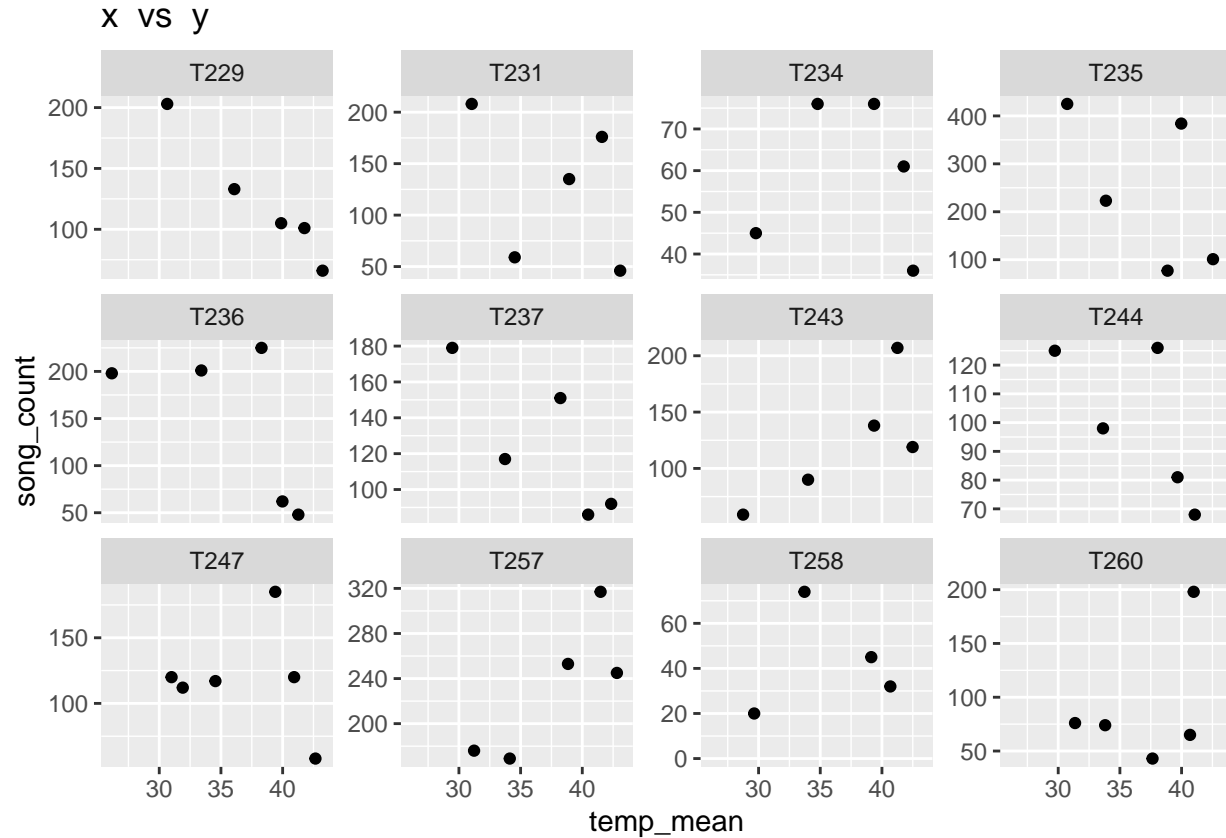
## Plot Filtered Data

```
data <- data_full %>%
  filter(round==3) %>%
  filter(count_total_round >=100)

xlab <- "Temperature"
ylab <- "song_count"

plot_temp_data <-
  ggplot(data) +
    aes(x = temp_mean,
        y = song_count) +
    facet_wrap("male", scales = "free_y") +
    geom_point() +
    labs(title = paste( substitute(x), " vs ", substitute(y)))
last_plot()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



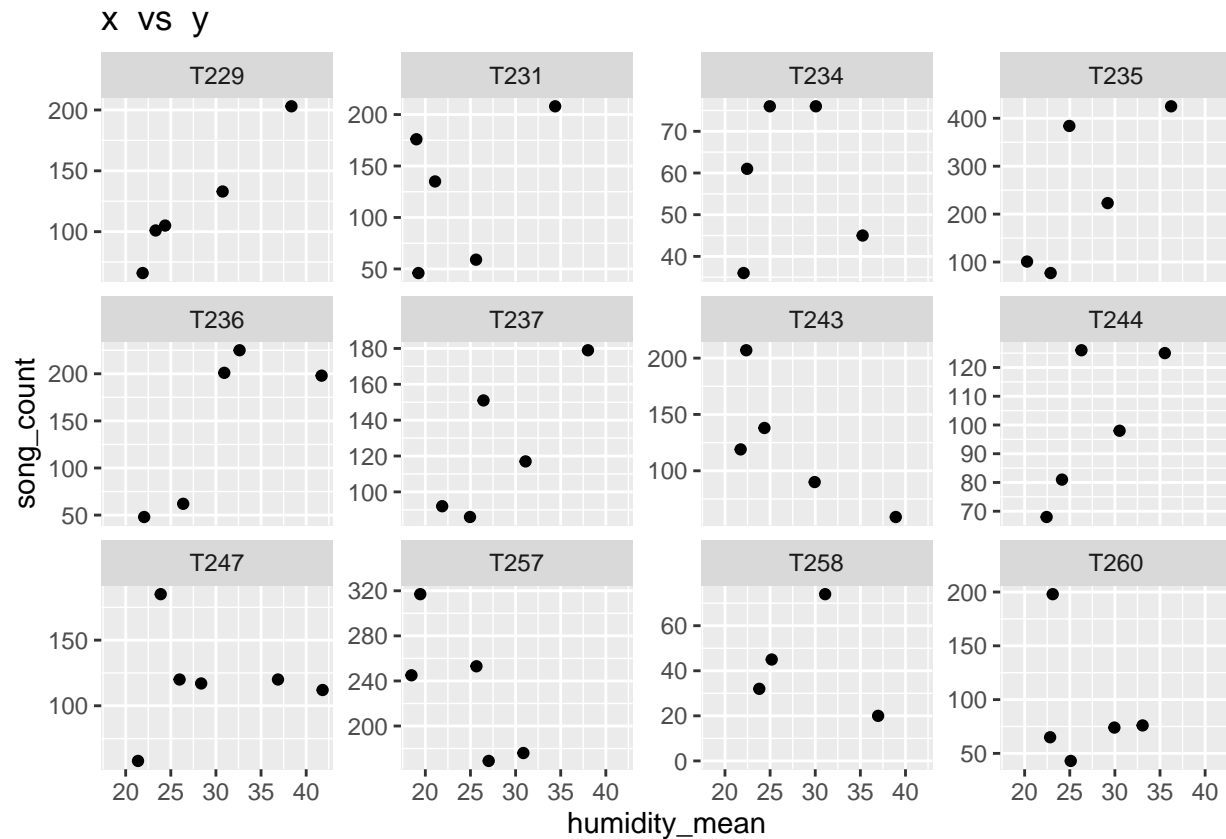
```
## Create again for humidity
```

```
xlab <- "humidity"
ylab <- "song_count"

plot_humidity_data <-
  ggplot(data) +
    aes(x = humidity_mean,
        y = song_count) +
    facet_wrap("male", scales = "free_y") +
    geom_point() +
    labs(title = paste( substitute(x), " vs ", substitute(y)))
last_plot()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```





## Work with mean

- This approach won't work because `temp_target` varies greatly from `temp_mean`.

```
plot_combine <- ggplot(data) +
  aes(x = temp_target, y = song_count, color = male) +
  geom_point()

# last_plot()

data_summarize <-
  data %>% group_by(temp_target) %>%
  summarize(song_mean = mean(song_count),
            song_sd = sd(song_count),
            song_n = dplyr::n(),
            song_se = song_sd/sqrt(song_n),
            song_ci = song_se*1.96
  )

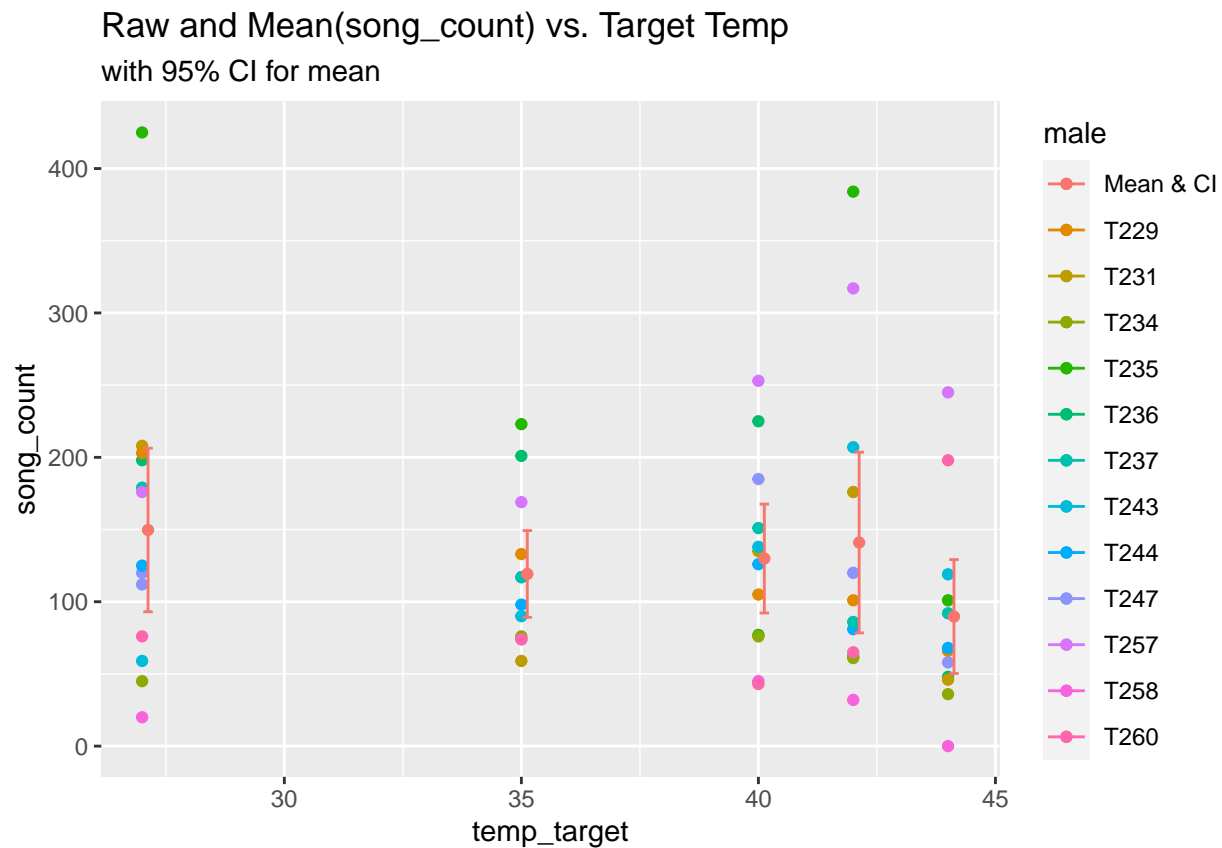
plot_summarize <- ggplot(data_summarize) +
  aes(x = temp_target, y = song_mean) +
  geom_point()+
  geom_errorbar(aes(ymin=song_mean - song_ci, ymax=song_mean+song_ci), width=.2)

# last_plot()

plot_combine +
```

```
geom_point(data = data_summarize, aes(x = temp_target+0.125, y = song_mean, colour = "Mean & CI"))+
geom_errorbar(data = data_summarize, aes(x = temp_target+0.125,
    y = song_mean,
    ymin=song_mean - song_ci,
    ymax=song_mean+song_ci,
    colour= "Mean & CI"), width=.2) +

labs(title = "Raw and Mean(song_count) vs. Target Temp",
    subtitle = "with 95% CI for mean")
```



## Result

- Data is noisy.
- Note temp\_target is not necessarily accurate.

## Humidity vs. Temp

```
data <- data_full %>% filter(round == 3)

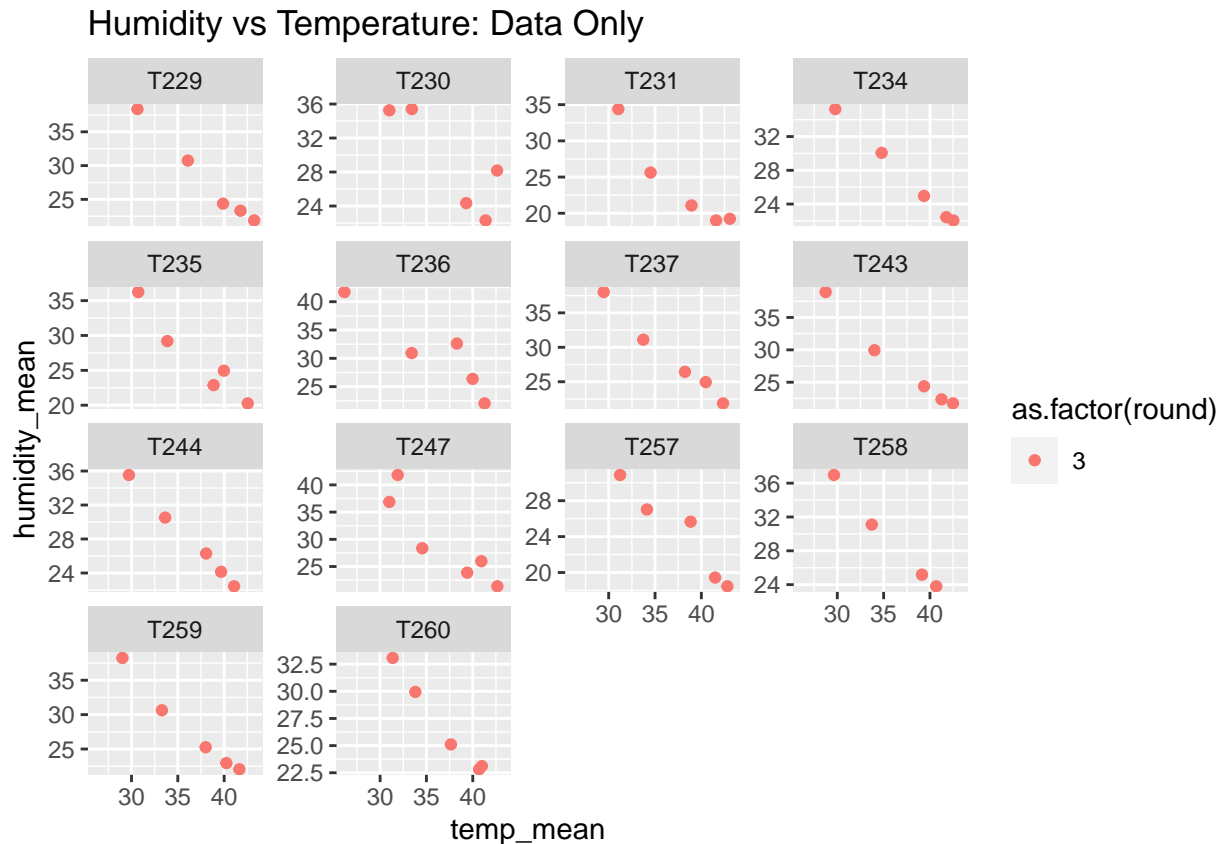
## As one might expect, Humidity is negatively correlated with temperature
ylab = "temp_mean"
plot_data_humidity_vs_temp <-
  ggplot(data = data) +
  aes(x = temp_mean,
```

```

    y = humidity_mean,
    color = as.factor(round)) +
  facet_wrap("male", scales = "free_y") +
  geom_jitter() +
  labs( title = "Humidity vs Temperature: Data Only")
last_plot()

```

## Warning: Removed 1 rows containing missing values (geom\_point).

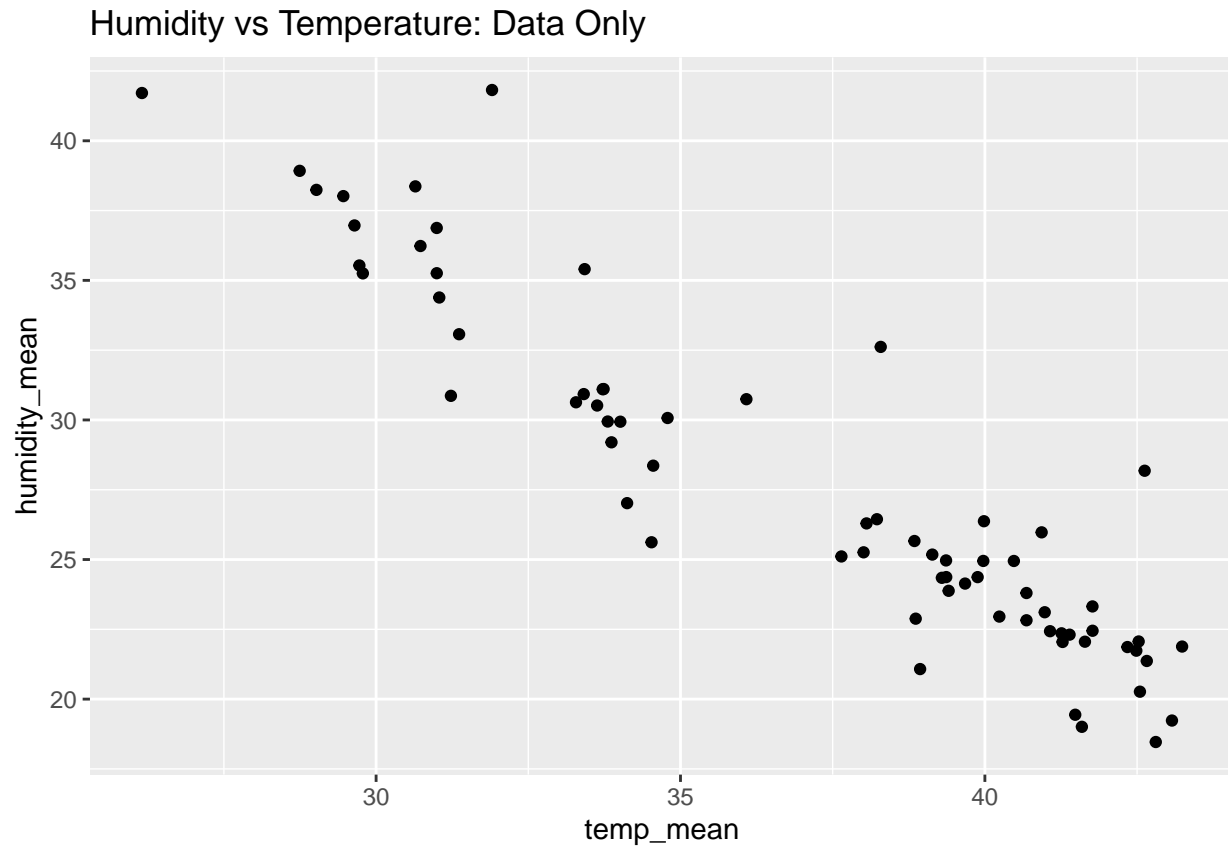


```

plot_data_humidity_vs_temp_combined <-
  ggplot(data = data) +
  aes(x = temp_mean,
      y = humidity_mean
    ) +
  geom_point() +
  labs( title = "Humidity vs Temperature: Data Only")
last_plot()

```

## Warning: Removed 1 rows containing missing values (geom\_point).



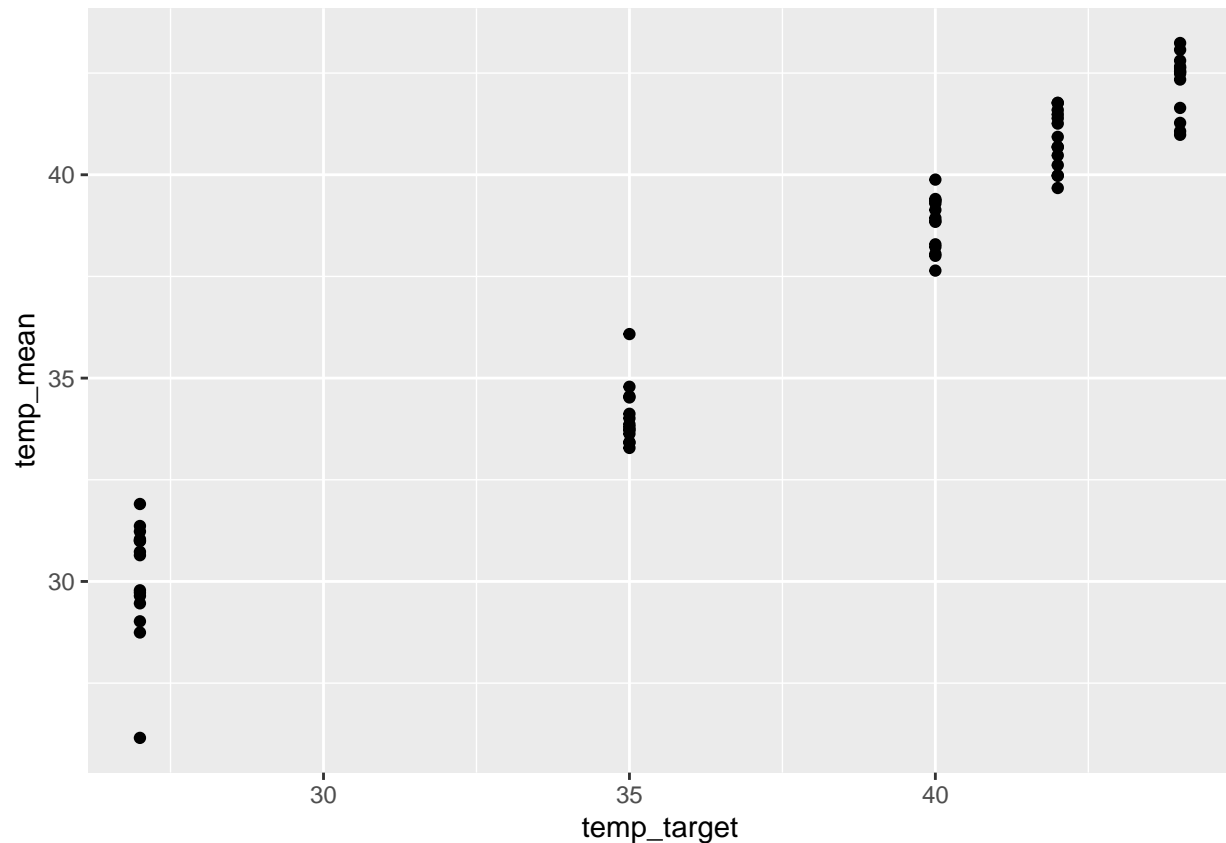
## Result

- These values clearly co-vary.
- Should consider using
  - Vapor Pressure Deficit
    - \* Uses temp to calculate max humidity and then looks at difference with relative humidity.
  - Heat Index
    - \* Adjusted for birds if available
    - \* Use just first terms ( $c_1 + c_2 T + c_3 R + c_4 T R + \dots$ )

## Compare temp\_target and temp\_mean

```
p <- ggplot(data, aes(x = temp_target, y = temp_mean)) +  
  geom_point()  
p
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



## Result

- Lots of variation thatn sometimes spans between `temp_target` categories.
- Clearly we want to work with `temp_mean`, rather than `temp_target`. As a result, only round = 3 since we don't have this data for the other rounds.

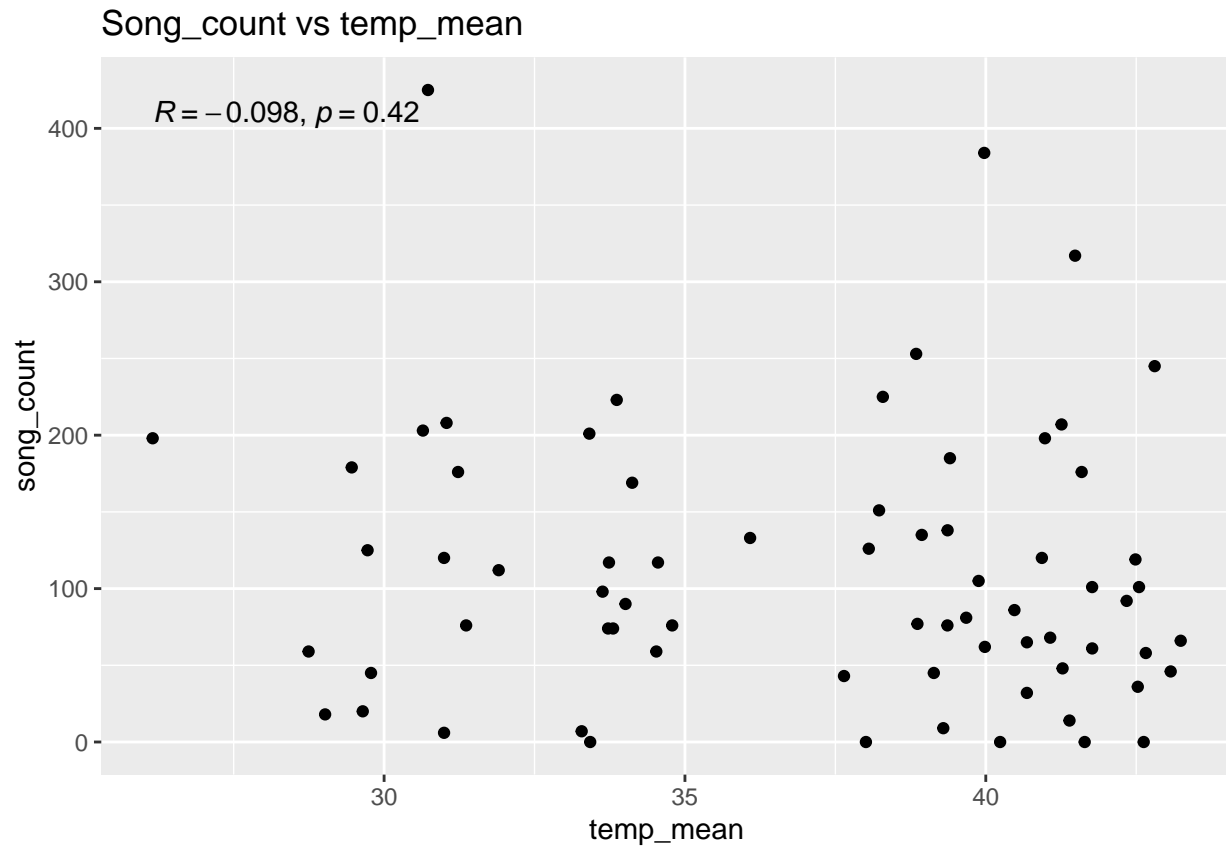
## Compare song\_count to temp\_mean and temp\_humidity

### Plot

```
p <- ggplot(data, aes(x = temp_mean, y = song_count), color = male) +
  geom_point() +
  stat_cor() +
  labs(title = "Song_count vs temp_mean")
last_plot()
```

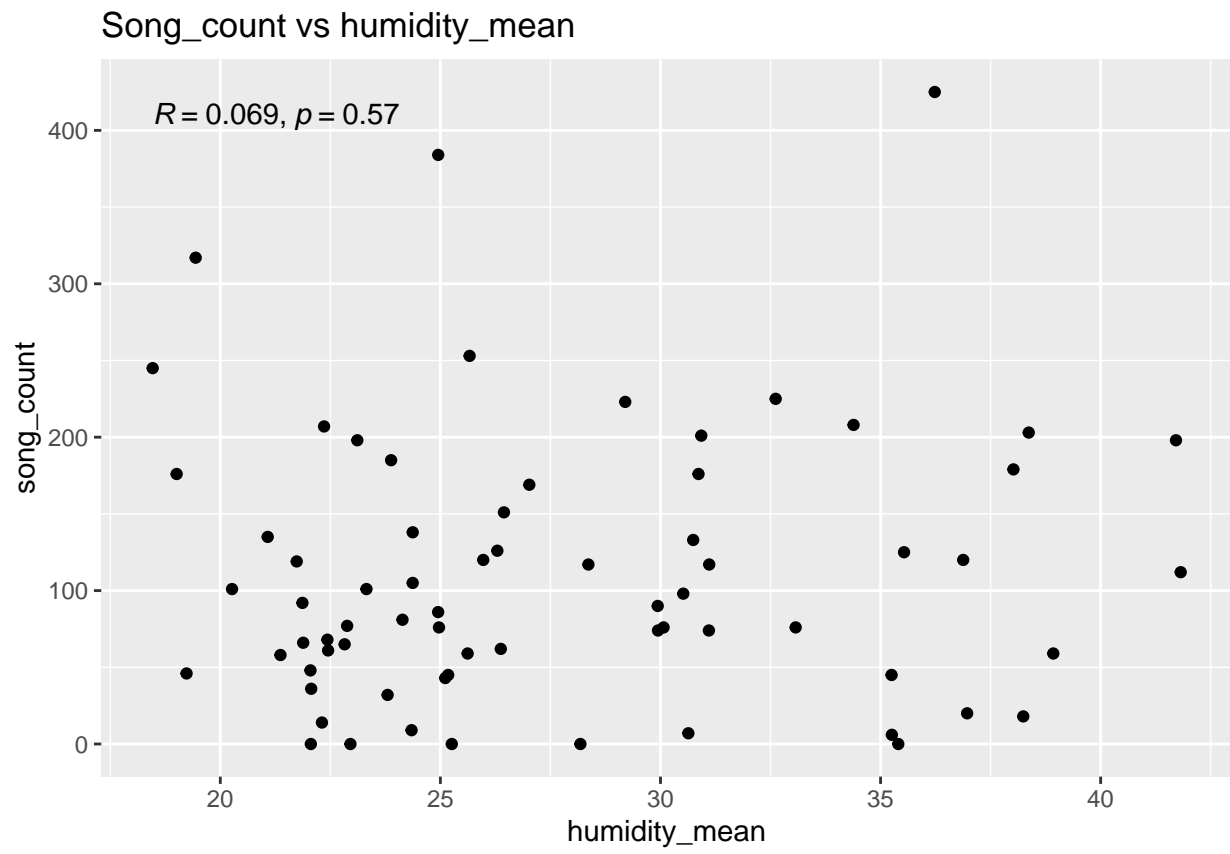
```
## Warning: Removed 1 rows containing non-finite values (stat_cor).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



```
p <- ggplot(data, aes(x = humidity_mean, y = song_count), color = male) +  
  geom_point() +  
  stat_cor() +  
  labs(title = "Song_count vs humidity_mean")  
last_plot()
```

```
## Warning: Removed 1 rows containing non-finite values (stat_cor).  
## Removed 1 rows containing missing values (geom_point).
```



```
knitr::knit_exit()
```