Compare target and observed temperatures.

Michael Gilchrist

date: 2023-01-13

Goal

- Evaluate variance in `temp_mean` vs. `temp_target`

Set up

Install libraries

```
# install packages user might not have by replacing FALSE with TRUE

## load libraries
library(stats)
library(ggplot2)
library(ggpubr)
library(ggpmisc)
library(grid)
library(gridExtra)
library(GGally)
library(broom)
library(tidyverse)
library(viridisLite)
```

Load Data

```
load(file.path("input", "data.processing_2022-12-15.Rda"),
     verbose = TRUE)
```

```
## Loading objects:
##   motif_data
##   motif_data_40C
##   motif_stats
##   motif_stats_40C
##   bird_bill_data
```

```
motif_data
```

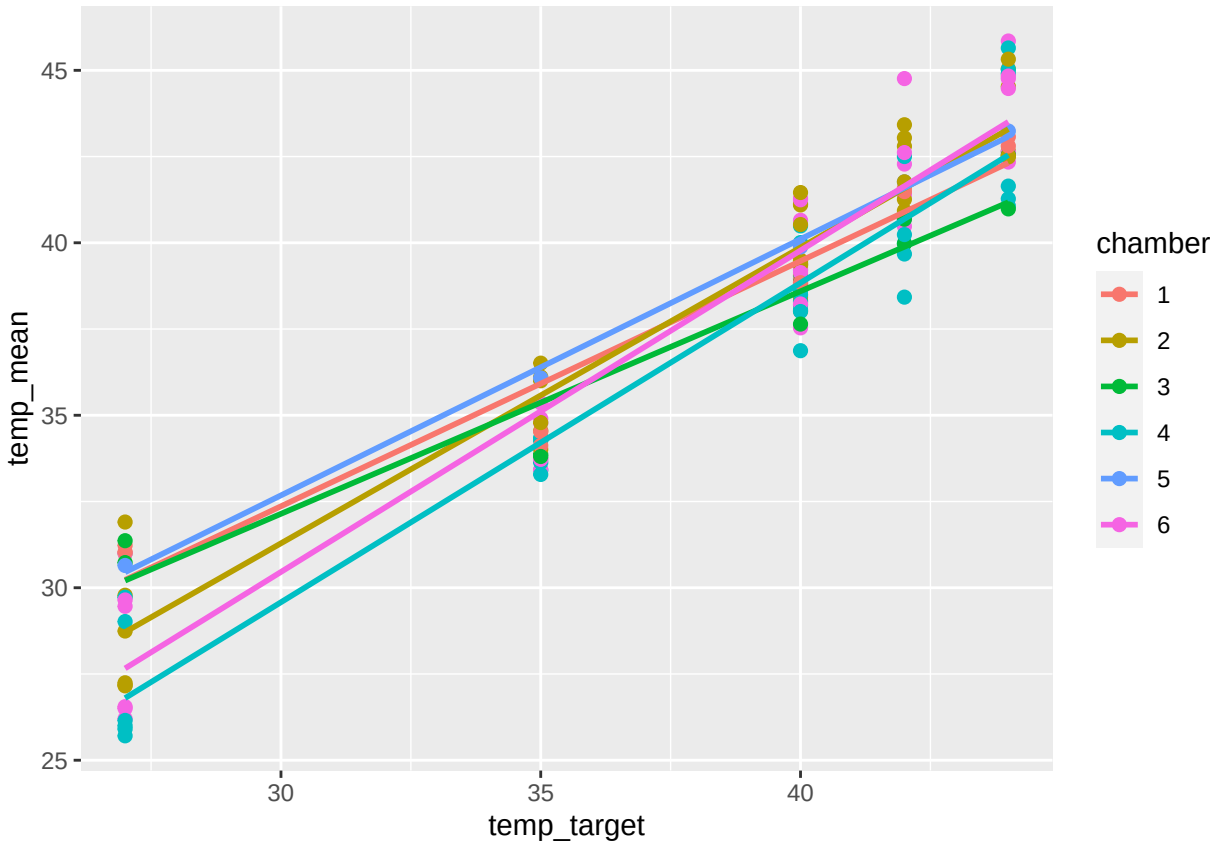
```
## # A tibble: 146 x 28
##    male  round trial_round motif~1 motif~2 temp_~3 humid~4 chamber date  counter
##    <fct> <dbl>       <dbl>   <int>   <dbl>   <dbl>   <dbl> <fct>   <chr> <chr>
##  1 T229      1           1       0 0          45.8      NA 6       02/1~ KIM
##  2 T229      1           2      24 0.0131     42.3      NA 6       02/1~ KIM
##  3 T229      1           3     114 0.0622     40.7      NA 6       02/1~ KIM
##  4 T229      1           4     198 0.108      26.2      NA 6       02/1~ KIM
##  5 T229      1           5     315 0.172      34.9      NA 6       02/2~ KIM
##  6 T231      1           1      57 0.0431     42.8      NA 2       02/1~ RAS
##  7 T231      1           2       7 0.00529    45.0      NA 2       02/1~ RAS
##  8 T231      1           3      86 0.0650     41.1      NA 2       02/1~ KIM
##  9 T231      1           4      24 0.0181     27.2      NA 2       02/1~ RAS
## 10 T231      1           5     215 0.162      36.5      NA 2       02/2~ RAS
## # ... with 136 more rows, 18 more variables: test_order <int>,
## #   temp_target <dbl>, temp_median <dbl>, humdity_mean <dbl>, motif_rate <dbl>,
## #   mass <dbl>, n_obs_completed <lgl>, motif_count_plus_1 <int>,
## #   log_motif_count_plus_1 <dbl>, temp <dbl>, n_obs_round <int>, n_obs <int>,
## #   trial <int>, motif_prop_round <dbl>, weights <dbl>, svp <dbl>, vpd <dbl>,
## #   vpd_offset <dbl>, and abbreviated variable names 1: motif_count,
## #   2: motif_prop, 3: temp_mean, 4: humidity_mean
```

Process Data

Create Working Dataset

```
data <- motif_data %>% select(temp_target, temp_mean, chamber, date)
data %>% ggplot() +
    aes(x = temp_target, y = temp_mean, color = chamber) +
    geom_point(size = 2) +
    stat_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
temp_var_by_temp <- data %>%
    group_by(temp_target) %>%
    summarize(var = var(temp_mean, na.rm = TRUE),
              se_var = var * sqrt(2/n()))

print(temp_var_by_temp)
```

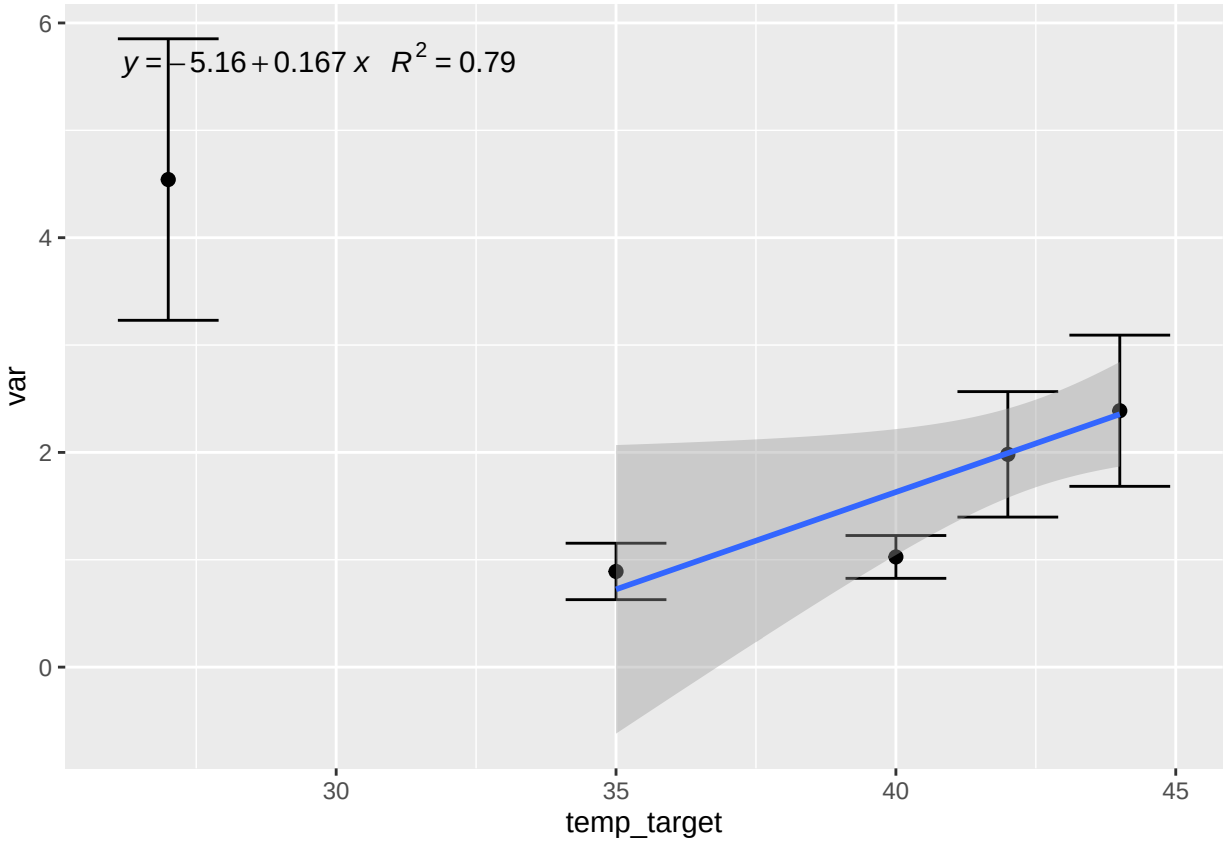
```
## # A tibble: 5 x 3
##   temp_target   var se_var
##         <dbl> <dbl>  <dbl>
## 1          27 4.54   1.31
## 2          35 0.891  0.263
## 3          40 1.03   0.199
## 4          42 1.98   0.584
## 5          44 2.39   0.704
```

```
plt <- ggplot(temp_var_by_temp) +
    aes(x = temp_target,
        y = var,
        ymin = var - se_var,
        ymax = var + se_var) +
    geom_point( size = 2) +
    geom_errorbar() +
    stat_smooth(data = ~ .x %>% filter( temp_target > 27),
                method = "lm",
                aes(weight = (se_var)^2)
```

```
                )

plt + stat_poly_eq(data = ~ .x %>% filter( temp_target > 27),
                   formula = y ~ x,
               aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
               parse = TRUE)
```

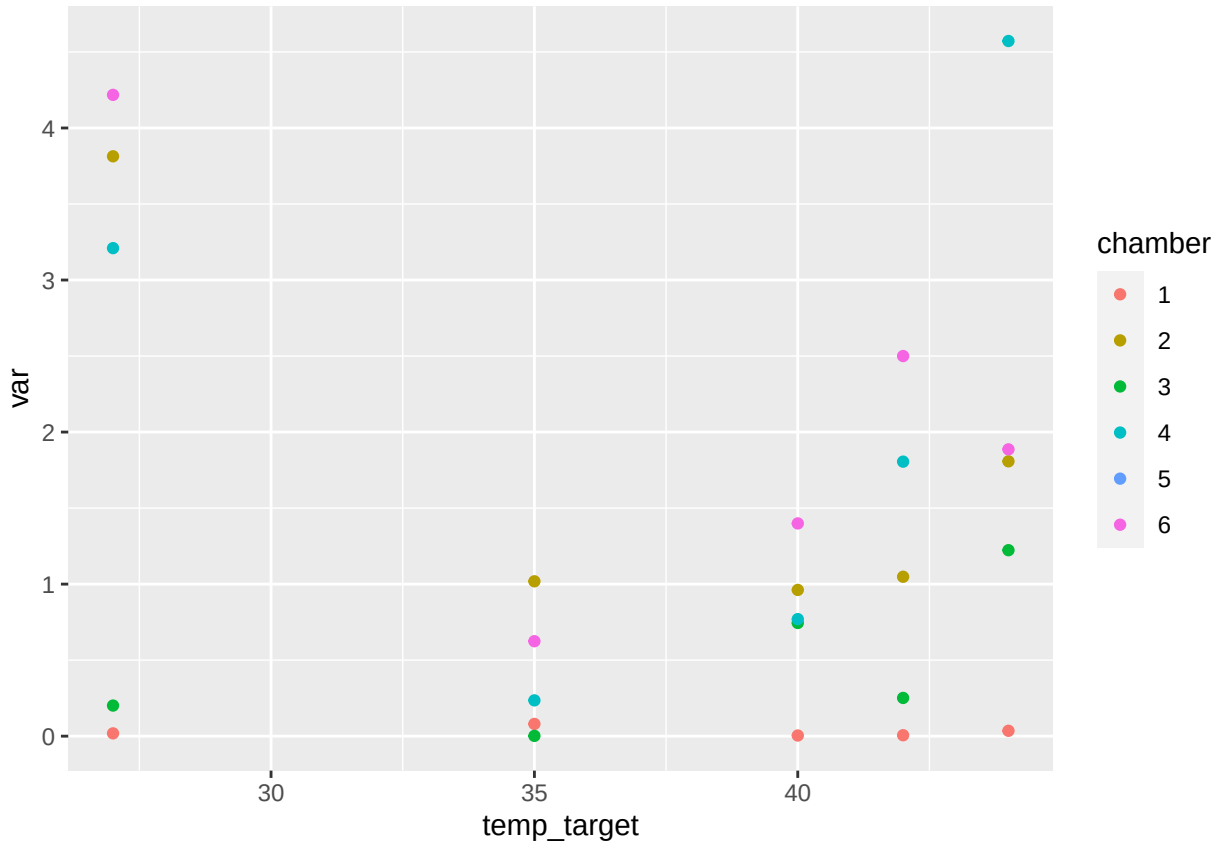
```
## `geom_smooth()` using formula = 'y ~ x'
```



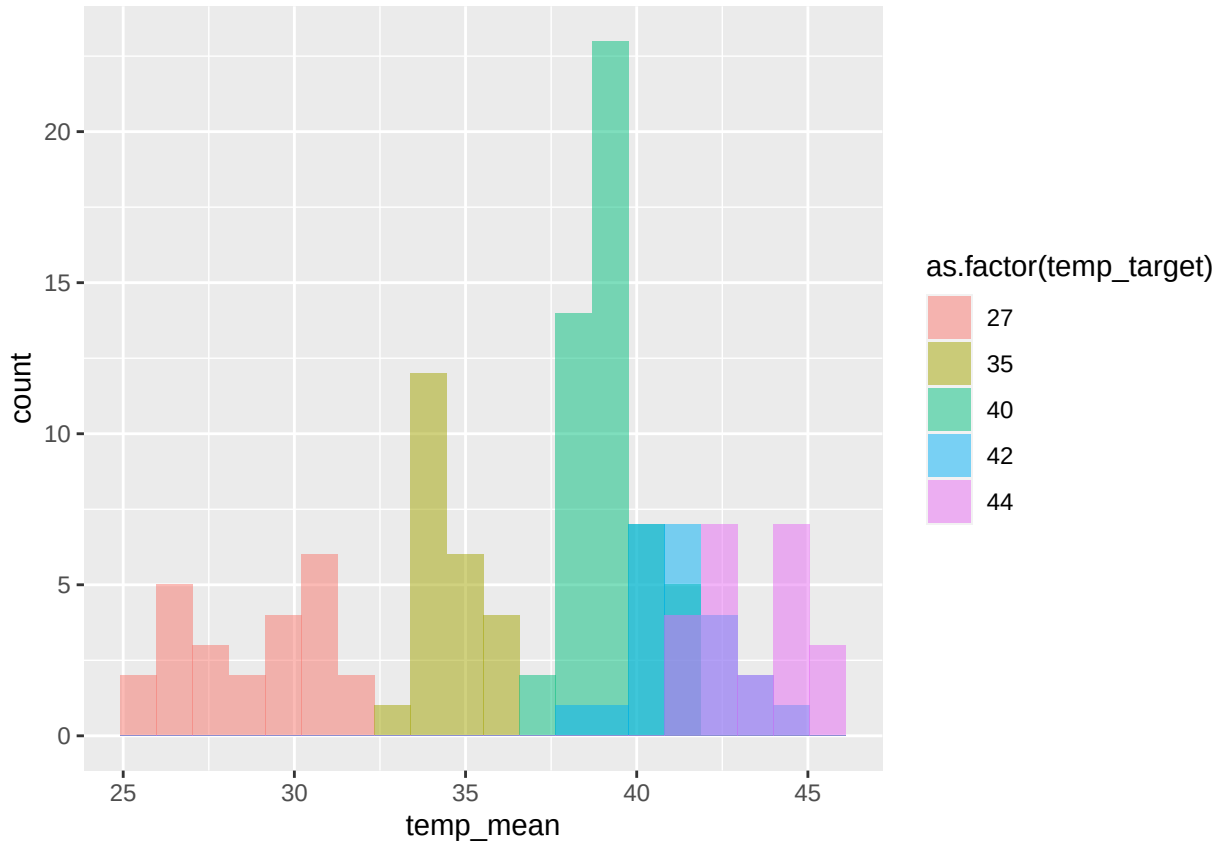

```
temp_var_by_temp_chamber <- data %>%  group_by(temp_target, chamber) %>% summarize(var = var(temp_mean,
```

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

```
ggplot(temp_var_by_temp_chamber) +
    aes(temp_target, var, color = chamber) +
    geom_point()
```




```
ggplot(data, aes(x = temp_mean, fill = as.factor(temp_target))) +                    # Draw overlayi
    geom_histogram(position = "identity",  alpha = 0.5, bins = 20)
```




```
#ggplot(data, aes(x = temp_mean, fill = as.factor(chamber))) +                    # Draw overlaying
#    geom_histogram(position = "identity",  alpha = 0.5, bins = 20)
```