Data from: Balancing risk and reward: mating opportunity influences thermal refuge use in fiddler crabs

M. Zachary Darnell, Talene G. Yeghissian, Zachary M. Lane

September 23, 2020

Contents

Setup

```
library(tidyverse)
library(ggeffects)
library(car)
library(lme4)
library(lubridate)
library(hms)
library(gridExtra)
library(gtable)
```

Data preparation

Microclimate data

Microclimate data are in two files. One downloaded from H21-USB on 2019-12-07 and one downloaded on 2019-12-12. These need to be imported and merged.

Importing the ws-20191207 data file:

```
ws.20191207<-read_csv("data/ws-20191207.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_character(),
##   Time = col_time(format = ""),
##   `Wind Speed (m/s)  #20684472` = col_double(),
##   `Gust Speed (m/s)  #20684472` = col_double(),
##   `2 Temp (°C)` = col_double(),
##   `RH (%)  #20703541` = col_double(),
##   `crab level Temp (°C)` = col_double(),
##   `sediment Temp (°C)` = col_double()
## )
```

Renaming columns:

```
ws.20191207 <- ws.20191207 %>%
  rename(date = Date,
         time = Time,
         wind.speed = "Wind Speed (m/s)  #20684472",
         gust.speed = "Gust Speed (m/s)  #20684472",
         temp.193cm = "2 Temp (°C)",
         rh = "RH (%)  #20703541",
         temp.6cm = "crab level Temp (°C)",
         temp.sed = "sediment Temp (°C)"
         )
```

Importing the ws-20191212 data file:

```r
ws.20191212<-read_csv("data/ws-20191212.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_character(),
##   Time = col_time(format = ""),
##   `Solar Radiation (W/m²)  #20662116` = col_double(),
##   `Wind Speed (m/s)  #20684472` = col_double(),
##   `Gust Speed (m/s)  #20684472` = col_double(),
##   `2 m Temp (°C)` = col_double(),
##   `RH (%)  #20703541` = col_double(),
##   `crab level Temp (°C)` = col_double(),
##   `sediment Temp (°C)` = col_double()
## )
```

Renaming columns:

```r
ws.20191212 <- ws.20191212 %>%
  rename(date = Date,
         time = Time,
         sol.irrad = "Solar Radiation (W/m²)  #20662116",
         wind.speed = "Wind Speed (m/s)  #20684472",
         gust.speed = "Gust Speed (m/s)  #20684472",
         temp.193cm = "2 m Temp (°C)",
         rh = "RH (%)  #20703541",
         temp.6cm = "crab level Temp (°C)",
         temp.sed = "sediment Temp (°C)"
         )
```

We're missing sol.irrad from the 20191207 dataset because pyranometer was not yet installed. Able to combine using `bind_rows`, which will include NAs for the missing sol.irrad data:

```r
ws <- bind_rows(ws.20191207,ws.20191212)
```

Convert dates into dates (they were originally loaded as character):

```r
ws$date <- as.Date(ws$date, format = "%y-%m-%d")
```

Combine the date and time columns into a datetime:

```r
ws <- ws %>%
  mutate(dt = ymd_hms(paste(date, time), tz = "Australia/Darwin"))
```

Removing all data after 15:00 on 2019-12-11, when the weather station was taken down:

```r
ws <- ws %>%
  filter(dt < as_datetime("2019-12-11 15:00:00", tz = "Australia/Darwin"))
ws
```

```
## # A tibble: 7,452 x 10
##    date       time     wind.speed gust.speed temp.193cm    rh temp.6cm temp.sed
```

```
##    <date>     <time>        <dbl>      <dbl>      <dbl> <dbl>     <dbl>     <dbl>
##  1 2019-12-06 10:45:02       0.5        5.54       33.9  65.3      36.1      42.5
##  2 2019-12-06 10:46:02       0.5        2.01       33.8  65.4      36.1      42.5
##  3 2019-12-06 10:47:02       0.5        1.51       33.7  66.3      36.2      42.6
##  4 2019-12-06 10:48:02       1.51       2.01       33.5  65.2      36.2      42.7
##  5 2019-12-06 10:49:02       0.5        2.01       33.3  65.5      36.1      42.7
##  6 2019-12-06 10:50:02       0.5        2.01       33.3  66.7      36.1      42.7
##  7 2019-12-06 10:51:02       1.51       2.52       33.3  66.5      35.9      42.7
##  8 2019-12-06 10:52:02       1.51       2.52       33.2  66.3      35.9      42.7
##  9 2019-12-06 10:53:02       1.01       2.52       33.1  67.1      36.0      42.7
## 10 2019-12-06 10:54:02       0.5        2.01       33.3  67.9      36.1      42.8
## # ... with 7,442 more rows, and 2 more variables: sol.irrad <dbl>, dt <dttm>
```

There are some pyranometer data that need to be excluded- measurements from when we were setting it up, when it was was being leveled, etc. Only pyranometer data need to be removed- the other sensors were recording correctly. First pulling out the dt and sol.irrad column into a different data frame for editing, then removing data before 8:05 AM on 2019-12-07, removing data from 12:54-12:55 on 2019-12-08, and removing data from 10:53-10:55 on 2019-12-10:

```
ws.solrad <- ws %>%
  select(dt, sol.irrad) %>%
  filter(dt >= as_datetime("2019-12-07 08:05:00", tz = "Australia/Darwin")) %>%
  filter((dt < as_datetime("2019-12-08 12:54:00", tz = "Australia/Darwin")) |
           (dt >= as_datetime("2019-12-08 12:55:00", tz = "Australia/Darwin"))) %>%
  filter((dt < as_datetime("2019-12-10 10:53:00", tz = "Australia/Darwin")) |
           (dt >= as_datetime("2019-12-10 10:55:00", tz = "Australia/Darwin")))# %>%
ws.solrad
```

```
## # A tibble: 6,172 x 2
##    dt                  sol.irrad
##    <dttm>                  <dbl>
##  1 2019-12-07 08:05:02      343.
##  2 2019-12-07 08:06:02      347.
##  3 2019-12-07 08:07:02      352.
##  4 2019-12-07 08:08:02      357.
##  5 2019-12-07 08:09:02      363.
##  6 2019-12-07 08:10:02      368.
##  7 2019-12-07 08:11:02      373.
##  8 2019-12-07 08:12:02      377.
##  9 2019-12-07 08:13:02      382.
## 10 2019-12-07 08:14:02      386.
## # ... with 6,162 more rows
```

Now need to merge this back with the main data frame, after removing the old sol.irrad column:

```
ws <- ws %>%
  select(-sol.irrad) %>%
  left_join (ws.solrad, by = "dt")
```

Compute correlation matrix for solar irradiance and temperature measurements:

```
ws %>%
  select(temp.193cm, temp.6cm, temp.sed, sol.irrad) %>%
  cor(use = "complete.obs")
```

```
##            temp.193cm  temp.6cm  temp.sed sol.irrad
## temp.193cm  1.0000000 0.9725298 0.9006268 0.8293532
## temp.6cm    0.9725298 1.0000000 0.9360600 0.8860478
## temp.sed    0.9006268 0.9360600 1.0000000 0.8578254
## sol.irrad   0.8293532 0.8860478 0.8578254 1.0000000
```

Experimental data

Loading the surface time experimental data:

```
surf.time <- read_csv("data/mjoebergi-surf-time-2019.csv")
```

```
## Parsed with column specification:
## cols(
##   num = col_double(),
##   date = col_character(),
##   trt = col_character(),
##   fem.id = col_double(),
##   fem.cw = col_double(),
##   waves = col_double(),
##   sur.time.min = col_double(),
##   sur.time.sec = col_double(),
##   end.time.acst = col_time(format = ""),
##   burrow.diam = col_double(),
##   notes = col_character(),
##   observer = col_character(),
##   exclude = col_character(),
##   qaqcd = col_character()
## )
```

Set trt as factor:

```
surf.time <- surf.time %>%
  mutate(trt = as.factor(trt))
```

Convert end time to datetime format:

```
surf.time <- surf.time %>%
  mutate(end.dt = mdy_hms(paste(date, end.time.acst), tz = "Australia/Darwin")) %>%
  mutate(date = as.Date(date, format = "%m/%d/%Y"))
surf.time
```

```
## # A tibble: 61 x 15
##      num date       trt   fem.id fem.cw waves sur.time.min sur.time.sec
##    <dbl> <date>     <fct>  <dbl>  <dbl> <dbl>        <dbl>        <dbl>
##  1     1 2019-12-06 fema~      1    9.2    33            5           40
```

```
##  2      2 2019-12-06 cont~     NA    NA       0           0            5
##  3      3 2019-12-06 fema~      4    12      45           4           23
##  4      4 2019-12-06 cont~     NA    NA       7           0           39
##  5      5 2019-12-06 fema~      2     9.3    13           1           39
##  6      6 2019-12-06 cont~     NA    NA       4           0           23
##  7      7 2019-12-06 fema~      5    10.7     0           1           16
##  8      8 2019-12-07 fema~     10     9.1   217          17           22
##  9      9 2019-12-07 cont~     NA    NA      23           1           57
## 10     10 2019-12-07 fema~     11    10.9     9           0           46
## # ... with 51 more rows, and 7 more variables: end.time.acst <time>,
## #   burrow.diam <dbl>, notes <chr>, observer <chr>, exclude <chr>, qaqcd <chr>,
## #   end.dt <dttm>
```

Calculating surface duration (surf.dur, s) and calculating start date/time (start.dt):

```
surf.time <- surf.time %>%
  mutate(surf.dur = (sur.time.min*60 + sur.time.sec)) %>%
  mutate(start.dt = end.dt - surf.dur)
```

Summary statistics for surf.dur:

```
summary(surf.time$surf.dur)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     5.0    46.0    94.0   173.8   199.0  1245.0
```

Calculating average sediment temperature and 10-cm air temperature for each surface interval. Experimental and microclimate datasets first need to be joined, using `inner_join` by date, which will generate many rows per crab observation (1 row per ws observation within that date, so multiple rows for each crab). Then will average between start.dt and end.dt to get the avg. temps:

```
avg.temps1 <- inner_join(surf.time,ws,by = "date") %>%
  filter(dt > start.dt & dt <= end.dt) %>%
  group_by(num) %>%
  summarise(sed.temp = mean(temp.sed),
            air.temp = mean(temp.6cm))
avg.temps1
```

```
## # A tibble: 42 x 3
##      num sed.temp air.temp
##    <dbl>    <dbl>    <dbl>
##  1     1     43.6     36.2
##  2     3     48.1     36.9
##  3     5     50.6     37.3
##  4     7     51.9     37.3
##  5     8     36.8     33.7
##  6     9     39.0     34.7
##  7    12     43.5     36.5
##  8    14     47.8     37.6
##  9    16     52.2     37.4
## 10    17     52.9     37.6
## # ... with 32 more rows
```

Only 42 rows instead of the 61 that should exist, because some crabs had short surface durations and there was no weather station reading during that interval. Will need to pull those out another way, by selecting the sed temp and air temp value closest to end.dt:

```
avg.temps2 <- inner_join(surf.time,ws,by = "date") %>%
  filter(surf.dur < 60) %>%
  group_by(num) %>%
  slice(which.min(abs(end.dt - dt))) %>%
  summarise(sed.temp = mean(temp.sed),
            air.temp = mean(temp.6cm))
avg.temps2
```

```
## # A tibble: 19 x 3
##      num sed.temp air.temp
##    <dbl>    <dbl>    <dbl>
##  1     2     46.0     37.2
##  2     4     49.1     37.0
##  3     6     51.4     37.6
##  4    10     40.3     34.7
##  5    11     42.3     35.8
##  6    13     44.2     36.2
##  7    15     49.4     37.6
##  8    24     40.7     35.3
##  9    26     45.1     36.2
## 10    28     51.3     37.6
## 11    31     39.7     35.0
## 12    36     37.9     34.5
## 13    37     41.0     35.6
## 14    39     43.4     36.5
## 15    40     46.1     36.1
## 16    49     46.7     35.6
## 17    51     37.1     34.0
## 18    55     39.1     35.0
## 19    58     40.3     34.9
```

Combining those two data frames:

```
avg.temps <- bind_rows(avg.temps1, avg.temps2)
avg.temps
```

```
## # A tibble: 61 x 3
##      num sed.temp air.temp
##    <dbl>    <dbl>    <dbl>
##  1     1     43.6     36.2
##  2     3     48.1     36.9
##  3     5     50.6     37.3
##  4     7     51.9     37.3
##  5     8     36.8     33.7
##  6     9     39.0     34.7
##  7    12     43.5     36.5
##  8    14     47.8     37.6
##  9    16     52.2     37.4
## 10    17     52.9     37.6
## # ... with 51 more rows
```

Now need to join avg.temps with surf.time, by num (crab ID column):

```
surf.time <- left_join(surf.time, avg.temps, by = "num")
```

Data analysis

Microclimate data summary statistics

```
summary(ws$temp.193cm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.57   29.46   30.87   31.15   33.57   35.21
```

```
summary(ws$temp.6cm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.67   29.66   31.69   32.64   36.47   39.63
```

```
summary(ws$temp.sed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27.58   31.23   34.81   38.74   46.80   57.02
```

```
summary(ws$wind.speed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.5000  0.5819  1.0100  3.0200
```

```
summary(ws$gust.speed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.500   1.510   1.742   2.520   6.040
```

```
summary(ws$rh)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   52.60   64.80   71.90   71.74   78.33   90.40
```

```
summary(ws$sol.irrad)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.6     0.6    81.9   334.6   755.6  1024.4    1280
```

Hypothesis testing

Checking the number of crabs in each treatment:

```r
summary(surf.time$trt)
```

```
## control  female
##      30      31
```

Filtering out rows where `exclude` = y:

```r
surf.time <- filter(surf.time, exclude == "n")
```

Checking the final number of crabs in each treatment:

```r
summary(surf.time$trt)
```

```
## control  female
##      30      29
```

Hypothesis 1: Although the courtship display (i.e., claw waving) occurs regardless of female presence, display behaviour increases in frequency and intensity in the presence of a female.

Need to classify each male as waving or not waving. If `waves` == 0, classify as "no", if `waves` > 0, classify as "yes":

```r
surf.time$waved <- if_else(condition = surf.time$waves == 0,
                           true = "no",
                           false = "yes")
```

Generate 2x2 table showing number of males who did not wave and number who did wave in each treatment:

```r
surf.time %>%
  group_by(trt, waved) %>%
  summarize(n = n()) %>%
  spread(waved, n)
```

```
## # A tibble: 2 x 3
## # Groups:   trt [2]
##   trt        no   yes
##   <fct>   <int> <int>
## 1 control    11    19
## 2 female      1    28
```

Fisher's exact test, comparing proportion waving between treatments:

```r
fisher.test(surf.time$trt, surf.time$waved)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  surf.time$trt and surf.time$waved
## p-value = 0.002466
```

```
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    1.97001 719.22124
## sample estimates:
## odds ratio
##   15.55601
```

Calculating waves per second by dividing `waves` by `surf.dur`:

```
surf.time <- surf.time %>%
  mutate(waves.per.sec = waves/surf.dur )
```

Split out only those who waved:

```
waved.only <- filter(surf.time,waved == "yes")
```

Mann-Whitney U test, comparing waving rate between treatments (for only those crabs that waved). Note that `wilcox.test` performs a Mann-Whitney U test when used in a two-sample context:

```
wilcox.test(waves.per.sec ~ trt, data = waved.only, alternative = "two.sided", paired = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  waves.per.sec by trt
## W = 134, p-value = 0.00436
## alternative hypothesis: true location shift is not equal to 0
```

Summary statistics:

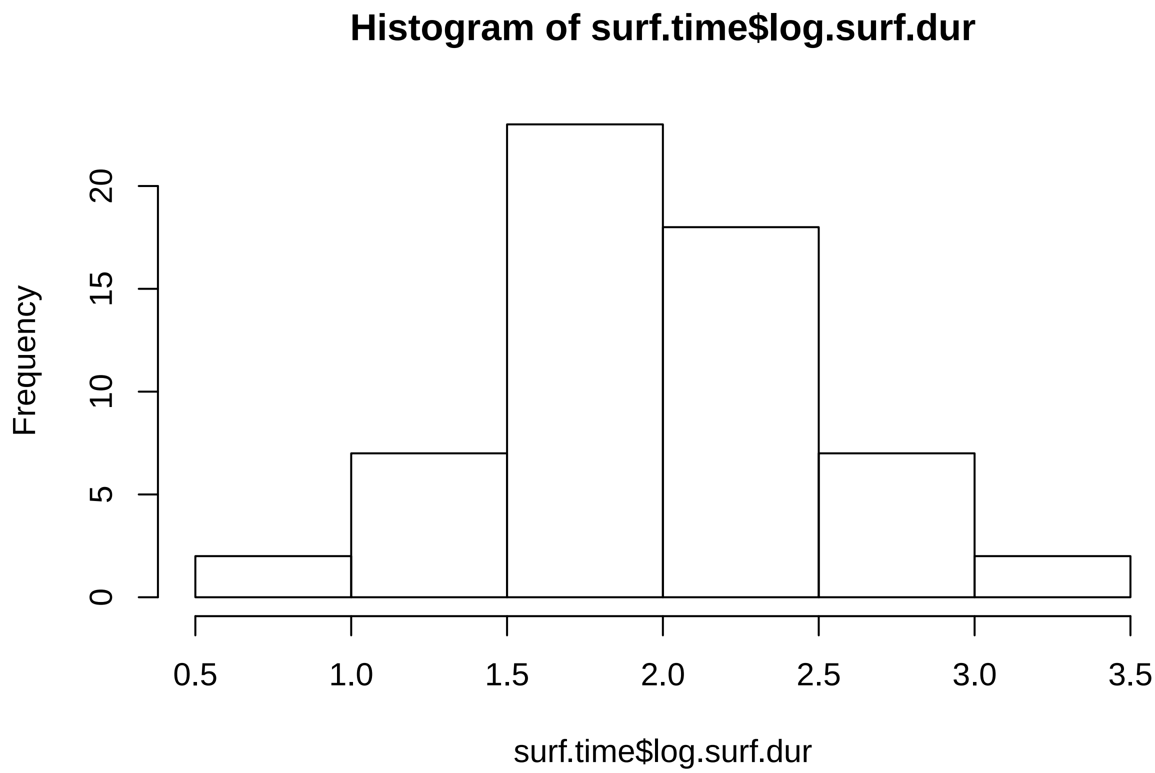
```
waved.only %>%
  group_by(trt) %>%
  summarize(n(), mean(waves.per.sec), 'se(waves.per.sec)' = sd(waves.per.sec)/sqrt(n()))
```

```
## # A tibble: 2 x 4
##   trt     `n()` `mean(waves.per.sec)` `se(waves.per.sec)`
##   <fct>   <int>                 <dbl>               <dbl>
## 1 control    19                0.0778              0.0159
## 2 female     28                0.147               0.0138
```

Hypothesis 2: Thermal stress limits the duration of surface activity in courting male fiddler crabs, yet males take greater thermal risks by remaining on the surface longer when the probability of attracting a mate is high.

Log10-transform surface durations:

```
surf.time$log.surf.dur<-log10(surf.time$surf.dur)
hist(surf.time$log.surf.dur)
```



Test for normality of log10-transformed surface durations:

```
shapiro.test(surf.time$log.surf.dur)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  surf.time$log.surf.dur
## W = 0.98978, p-value = 0.9023
```

Fit linear model testing for effects of treatment and sediment surface temperature on surface duration, generate ANOVA table wih type II F-tests:

```
surface.lm.sed<-lm(log.surf.dur ~ trt*sed.temp, data = surf.time)
Anova(surface.lm.sed, test.statistic = "F", type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: log.surf.dur
##              Sum Sq Df F value    Pr(>F)
## trt          5.1860  1 35.1513 2.096e-07 ***
## sed.temp     1.3579  1  9.2040  0.003682 **
## trt:sed.temp 0.0001  1  0.0006  0.980582
## Residuals    8.1143 55
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Generate summary statistics for surface durations, by treatment:

```
surf.time %>%
  group_by(trt) %>%
  summarize(n(), mean(surf.dur), sd(surf.dur), 'se(surf.dur)' = sd(surf.dur)/sqrt(n()),
            median(surf.dur))
```

```
## # A tibble: 2 x 6
##   trt     `n()` `mean(surf.dur)` `sd(surf.dur)` `se(surf.dur)` `median(surf.dur~
##   <fct>   <int>            <dbl>          <dbl>          <dbl>             <dbl>
## 1 control    30             69.9           56.3           10.3                57
## 2 female     29            288.           291.            54.1               225
```

What is the decrease in surface duration in each treatment as sediment temperature increases from 35-50°C?

```
newdata <- expand.grid(
  sed.temp = c(35, 50),
  trt = c("female", "control"))

pred3550 <- predict(surface.lm.sed, newdata, interval = "confidence")

cbind(newdata, pred3550) %>%
  as.data.frame() %>%
  mutate(bt_surfdur = 10^fit, # back-transforming the predicted value and confidence int.
         bt_lwr = 10^lwr,
         bt_upr = 10^upr)
```

```
##   sed.temp     trt      fit      lwr      upr bt_surfdur    bt_lwr    bt_upr
## 1       35  female 2.511743 2.262164 2.761322  324.89492 182.87904 577.19413
## 2       50  female 2.065030 1.819349 2.310710  116.15283  65.97041 204.50807
## 3       35 control 1.922137 1.650994 2.193280   83.58665  44.77069 156.05587
## 4       50 control 1.468146 1.218228 1.718063   29.38637  16.52831  52.24724
```

Hypothesis 3: These behavioural decisions also take into account female quality such that males are willing to take greater thermal risks when larger and thus potentially more fecund females are nearby.

Subset to only the female-present treatment:

```
surf.time.fem<-filter(surf.time,trt == "female")
```

Summary statistics for female carapace width:

```
summary(surf.time.fem$fem.cw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.300   9.000  10.000   9.721  10.900  12.000
```

```
surf.time.fem %>%
  summarize(mean(fem.cw),
            'se(fem.cw)' = sd(fem.cw)/sqrt(n()))
```

```
## # A tibble: 1 x 2
##   `mean(fem.cw)` `se(fem.cw)`
##            <dbl>        <dbl>
## 1           9.72        0.210
```

Fit linear model, testing for effects of female carapace width and sediment surface temperature on surface duration, generate ANOVA table with type II F-tests:

```
fem.lm.sed<-lm(log.surf.dur ~ fem.cw*sed.temp, data = surf.time.fem)
Anova(fem.lm.sed, test.statistic = "F", type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: log.surf.dur
##                 Sum Sq Df F value  Pr(>F)
## fem.cw          0.0626  1  0.4553 0.50604
## sed.temp        0.7071  1  5.1414 0.03226 *
## fem.cw:sed.temp 0.0844  1  0.6136 0.44079
## Residuals       3.4382 25
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure preparation

Figure 2

Left half

Create data frames for each measurement type:

```
ws.temp <- ws %>%
  select(c(dt, temp.sed, temp.6cm, temp.193cm)) %>%
  pivot_longer(-dt, names_to = "type")
ws.temp$type = factor(ws.temp$type, levels = c("temp.193cm", "temp.6cm", "temp.sed"))

ws.rh <- ws %>%
  select(c(dt, rh)) %>%
  pivot_longer(-dt, names_to = "type")

ws.wind <- ws %>%
  select(c(dt, wind.speed, gust.speed)) %>%
  pivot_longer(-dt, names_to = "type")
ws.wind$type = factor(ws.wind$type, levels = c("wind.speed", "gust.speed"))

ws.solrad <- ws %>%
  select(c(dt, sol.irrad)) %>%
  pivot_longer(-dt, names_to = "type")
```

Create individual panels for each:

```
temp.fig<- ggplot(ws.temp) +
  geom_line(aes(x = dt, y = value, linetype = type)) +
  scale_linetype_manual(values = c("dotted","solid", "dashed"),
                        labels = c("Air (193 cm)", "Air (6 cm)","Sed. surface")) +
  labs(x="Date",
       y=expression(atop("Temperature", "(°C)"))) +
  scale_y_continuous(limits = c(20, 60),
                     breaks = seq (20,60,10)) +
  scale_x_datetime(date_breaks = "day",
                   limits = c(as.POSIXct("2019-12-06 09:00:00", tz = "Australia/Darwin"),
                              as.POSIXct("2019-12-11 15:00:00", tz = "Australia/Darwin")),
                   date_labels = "%b %d",
                   timezone = "Australia/Darwin") +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.3,
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=18),
    axis.text.x = element_blank(),
    axis.title.y = element_text(margin = margin(t = 0, r = 0, b = 0, l = 0),
                                colour="black", size=18),
    axis.title.x = element_blank(),
    legend.position = "none",
    legend.text=element_text(size=18),
    legend.title=element_blank())

rh.fig<- ggplot(data = ws.rh) +
  geom_line(aes(x = dt, y = value, linetype = type)) +
  labs(x="Date",
       y=expression(atop("Relative humidity", "(%)"))) +
  scale_linetype_manual(values = c("solid"),
                        labels = c("Relative humidity")) +
  scale_y_continuous(limits = c(0, 100),
                     breaks = seq (0,100,25)) +
  scale_x_datetime(date_breaks = "day",
                   limits = c(as.POSIXct("2019-12-06 09:00:00", tz = "Australia/Darwin"),
                              as.POSIXct("2019-12-11 15:00:00", tz = "Australia/Darwin")),
                   date_labels = "%b %d",
                   timezone = "Australia/Darwin") +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.3,
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=18),
    axis.text.x = element_blank(),
    axis.title.y = element_text(margin = margin(t = 0, r = 0, b = 0, l = 0),
                                colour="black", size=18),
    axis.title.x = element_blank(),
    legend.position = "none",
```

```
    legend.text=element_text(size=18),
    legend.title=element_blank())

wind.fig<- ggplot(ws.wind) +
  geom_line(aes(x = dt, y = value, color = type), size = 0.25) +
  geom_line(data = subset(ws.wind, type == "wind.speed"), aes(x = dt, y = value),
            size = 0.25) +
  scale_shape_manual(values = c(19,3),
                     labels = c("Wind speed", "Gust speed")) +
  scale_color_manual(values = c("black","gray"),
                     labels = c("Wind speed", "Gust speed")) +
  labs(x="Date",
       y=expression(atop("Wind speed", paste((m~s^-1))))) +
  scale_y_continuous(limits = c(-0.8, 7),
                     breaks = seq (0,8,2)) +
  scale_x_datetime(date_breaks = "day",
                   limits = c(as.POSIXct("2019-12-06 09:00:00", tz = "Australia/Darwin"),
                              as.POSIXct("2019-12-11 15:00:00", tz = "Australia/Darwin")),
                   date_labels = "%b %d",
                   timezone = "Australia/Darwin") +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.3,
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=18),
    axis.text.x = element_blank(),
    axis.title.y = element_text(margin = margin(t = 0, r = 0, b = 0, l = 0),
                                colour="black", size=18),
    axis.title.x = element_blank(),
    legend.position = "none",
    legend.text=element_text(size=18),
    legend.title=element_blank())

solrad.fig<- ggplot(data = ws.solrad) +
  geom_line(aes(x = dt, y = value, linetype = type)) +
  labs(x="Date",
       y=expression(atop("Solar irradiance", paste((W~m^-2))))) +
   scale_linetype_manual(values = c("solid"),
                        labels = c("Solar irradiance")) +
  scale_y_continuous(limits = c(0, 1200),
                     breaks = seq (0,1200,400)) +
  scale_x_datetime(date_breaks = "day",
                   limits = c(as.POSIXct("2019-12-06 09:00:00", tz = "Australia/Darwin"),
                              as.POSIXct("2019-12-11 15:00:00", tz = "Australia/Darwin")),
                   date_labels = "%b %d",
                   timezone = "Australia/Darwin") +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.3,
```

```
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                              colour="black", size=18),
    axis.text.x = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                              colour="black", size=18),
    axis.title.y = element_text(margin = margin(t = 0, r = 0, b = 0, l = 0),
                               colour="black", size=18),
    axis.title.x = element_text(margin = margin(t = 0, r = 0, b = 0, l = 0),
                               colour="black", size=18),
    legend.position = "none",
    legend.text=element_text(size=18),
    legend.key = element_blank(),
    legend.title=element_blank())
```

Right half

Average every 5-min, across all dates.

```
ws.avg <- ws %>%
  mutate(today = rep("2019-03-12", length(time))) %>%
  mutate(fake.dt = ymd_hms(paste(today, time), tz = "Australia/Darwin")) %>%
  mutate(fake.dt.rnd = floor_date(fake.dt, unit = "5 min")) %>%
  group_by(fake.dt.rnd) %>%
  summarise(mean.wind.speed = mean(wind.speed, na.rm = TRUE),
            mean.gust.speed = mean(gust.speed, na.rm = TRUE),
            mean.temp.193cm = mean(temp.193cm, na.rm = TRUE),
            mean.temp.6cm = mean(temp.6cm, na.rm = TRUE),
            mean.temp.sed = mean(temp.sed, na.rm = TRUE),
            mean.rh = mean(rh, na.rm = TRUE),
            mean.sol.irrad = mean(sol.irrad, na.rm = TRUE),
            sd.wind = sd(wind.speed),
            sd.gust = sd(gust.speed),
            sd.193cm = sd(temp.193cm),
            sd.6cm = sd(temp.6cm),
            sd.sed = sd(temp.sed),
            sd.rh = sd(rh),
            sd.irrad = sd(sol.irrad, na.rm = TRUE))
```

Generate data frames for each subplot:

```
detach("package:car", unload=TRUE)

avgtemp <- ws.avg %>%
  select(fake.dt.rnd, mean.temp.sed, mean.temp.6cm, mean.temp.193cm, sd.sed, sd.6cm, sd.193cm) %>%
  pivot_longer(c(-fake.dt.rnd,-sd.sed, -sd.6cm, -sd.193cm), names_to = "location") %>%
  pivot_longer(c(-fake.dt.rnd,-location), names_to = "stdev") %>%
  mutate(location = recode(location,
         `mean.temp.sed` = "sed",
         `mean.temp.6cm` = "6cm",
         `mean.temp.193cm` = "193cm")) %>%
  mutate(stdev = recode(stdev,
         `sd.sed` = "sed",
         `sd.6cm` = "6cm",
         `sd.193cm` = "193cm")) %>%
```

```
  filter(location == stdev | stdev == "value") %>%
  mutate(type = recode(stdev,
         `value` = "mean",
         .default = "sd")) %>%
  select(c(-stdev)) %>%
  pivot_wider(names_from="type", values_from = value) %>%
  mutate(location = factor(location, levels = c("193cm", "6cm", "sed")))

avgrh <- ws.avg %>%
  select(c(fake.dt.rnd, mean.rh, sd.rh)) %>%
  pivot_longer(c(-fake.dt.rnd, -sd.rh), names_to = "type")

avgwind <- ws.avg %>%
  select(c(fake.dt.rnd, mean.wind.speed, mean.gust.speed, sd.wind, sd.gust)) %>%
  pivot_longer(c(-fake.dt.rnd, -sd.wind, -sd.gust), names_to = "location") %>%
  pivot_longer(c(-fake.dt.rnd,-location), names_to = "stdev") %>%
  mutate(location = recode(location,
         `mean.wind.speed` = "wind",
         `mean.gust.speed` = "gust")) %>%
  mutate(stdev = recode(stdev,
         `sd.wind` = "wind",
         `sd.gust` = "gust")) %>%
  filter(location == stdev | stdev == "value") %>%
  mutate(type = recode(stdev,
         `value` = "mean",
         .default = "sd")) %>%
  select(c(-stdev)) %>%
  pivot_wider(names_from="type", values_from=value) %>%
  mutate(location = factor(location, levels = c("wind", "gust")))

avgirrad <- ws.avg %>%
  select(c(fake.dt.rnd, mean.sol.irrad, sd.irrad)) %>%
  pivot_longer(c(-fake.dt.rnd, -sd.irrad), names_to = "type")
```

Create each panel:

```
temp.fig2<- ggplot(avgtemp) +
  geom_line(aes(x = fake.dt.rnd, y = mean, linetype = location)) +
  geom_ribbon(aes(x = fake.dt.rnd, ymin = (mean-sd), ymax = (mean+sd), group = location,
                  color = NULL),
              alpha = .2) +
  labs(x="Date", y="Temperature (°C)") +
  scale_linetype_manual(values = c("dotted","solid", "dashed"),
                        labels = c("Air (193 cm)", "Air (6 cm)","Sed. surface")) +
  scale_y_continuous(limits = c(20, 60),
                     breaks = seq (20,60,10)) +
  scale_x_datetime(date_breaks = "2 hour",
                   limits = c(as.POSIXct("2019-03-12 00:00:00", tz = "Australia/Darwin"),
                              as.POSIXct("2019-03-13 00:00:00", tz = "Australia/Darwin")),
                   date_labels = "%H",
                   timezone = "Australia/Darwin") +
  theme_bw() +
  theme(
```

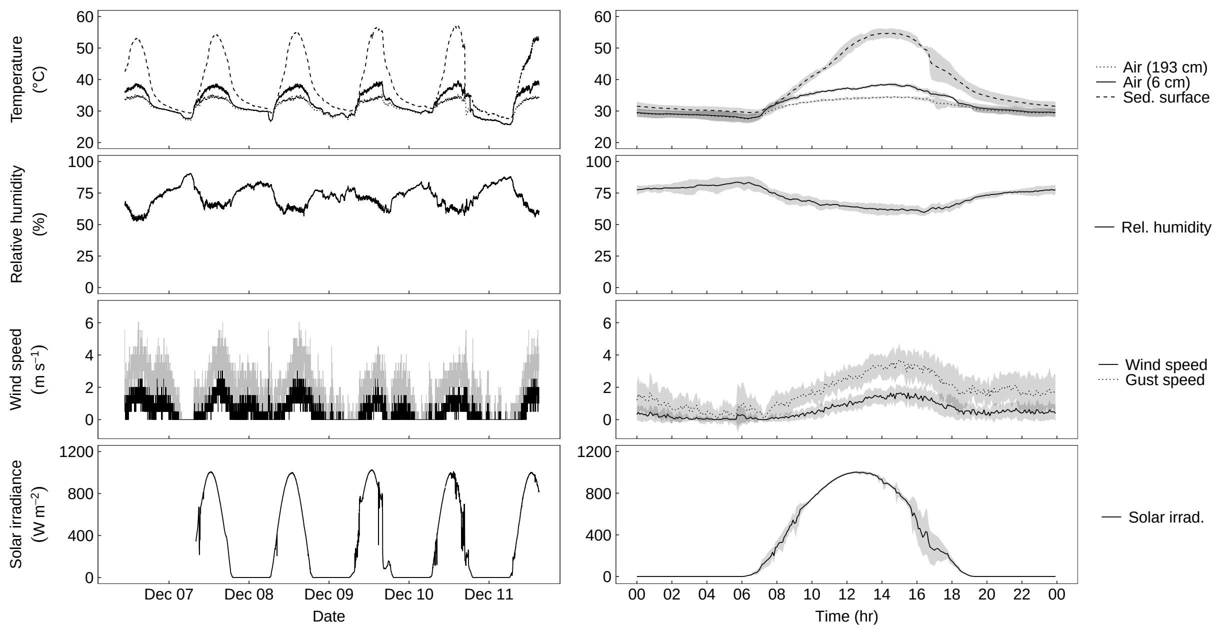
```
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.3,
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=18),
    axis.text.x = element_blank(),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    legend.text=element_text(size=18),
    legend.title=element_blank(),
    legend.key.width = unit(1, "cm"))

rh.fig2<- ggplot(data = avgrh) +
  geom_line(aes(x = fake.dt.rnd, y = value, linetype = type)) +
  geom_ribbon(aes(x = fake.dt.rnd, ymin = value -sd.rh, ymax = value + sd.rh, color = NULL),
            alpha = .2)+
  labs(x="Date", y="Relative humidity (%)") +
  scale_linetype_manual(values = c("solid"),
                      labels = c("Rel. humidity")) +
  scale_y_continuous(limits = c(0, 100),
                   breaks = seq (0,100,25)) +
  scale_x_datetime(date_breaks = "2 hour",
                  limits = c(as.POSIXct("2019-03-12 00:00:00", tz = "Australia/Darwin"),
                             as.POSIXct("2019-03-13 00:00:00", tz = "Australia/Darwin")),
                  date_labels = "%H",
                  timezone = "Australia/Darwin") +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.3,
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=18),
    axis.text.x = element_blank(),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    legend.text=element_text(size=18),
    legend.title=element_blank(),
    legend.key.width = unit(1, "cm"))

wind.fig2<- ggplot(avgwind) +
  geom_line(aes(x = fake.dt.rnd, y = mean, linetype = location)) +
  geom_ribbon(aes(x = fake.dt.rnd, ymin = (mean-sd), ymax = (mean+sd),
                group = location, color = NULL), alpha = .2) +
  scale_linetype_manual(values = c("solid","dotted"),
                      labels = c("Wind speed", "Gust speed")) +
  labs(x="Date", y=expression(Wind~speed~(m~s^-1))) +
  scale_y_continuous(limits = c(-0.8, 7),
                   breaks = seq (0,8,2)) +
  scale_x_datetime(date_breaks = "2 hour",
                  limits = c(as.POSIXct("2019-03-12 00:00:00", tz = "Australia/Darwin"),
                             as.POSIXct("2019-03-13 00:00:00", tz = "Australia/Darwin")),
                  date_labels = "%H",
```

```
                   timezone = "Australia/Darwin") +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.3,
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=18),
    axis.text.x = element_blank(),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    legend.text=element_text(size=18),
    legend.title=element_blank(),
    legend.key.width = unit(1, "cm"))

solrad.fig2<- ggplot(avgirrad) +
  geom_line(aes(x = fake.dt.rnd, y = value, linetype = type)) +
  geom_ribbon(aes(x = fake.dt.rnd, ymin = (value-sd.irrad), ymax = (value+sd.irrad),
                  color = NULL), alpha = .2)+
  labs(x="Time (hr)", y=expression(Solar~irradiance~(W~m^-2))) +
  scale_linetype_manual(values = c("solid"),
                        labels = c("Solar irrad.")) +
  scale_y_continuous(limits = c(-10, 1200),
                     breaks = seq (0,1200,400)) +
  scale_x_datetime(date_breaks = "2 hour",
                   limits = c(as.POSIXct("2019-03-12 00:00:00", tz = "Australia/Darwin"),
                              as.POSIXct("2019-03-13 00:00:00", tz = "Australia/Darwin")),
                   date_labels = "%H",
                   timezone = "Australia/Darwin") +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.3,
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=18),
    axis.text.x = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=18),
    axis.title.y = element_blank(),
    axis.title.x = element_text(margin = margin(t = 0, r = 0, b = 0, l = 0),
                                colour="black", size=18),
    legend.text=element_text(size=18),
    legend.key = element_blank(),
    legend.title=element_blank(),
    legend.key.width = unit(1, "cm"))
```

Combine all panels

```
tempg1 <- ggplotGrob(temp.fig)
rhg1 <- ggplotGrob(rh.fig)
windg1 <- ggplotGrob(wind.fig)
solg1 <- ggplotGrob(solrad.fig)
tempg2 <- ggplotGrob(temp.fig2)
```

```
rhg2 <- ggplotGrob(rh.fig2)
windg2 <- ggplotGrob(wind.fig2)
solg2 <- ggplotGrob(solrad.fig2)

grid.arrange(rbind(
  cbind(tempg1,tempg2),
  cbind(rhg1,rhg2),
  cbind(windg1,windg2),
  cbind(solg1,solg2)
  ))
```

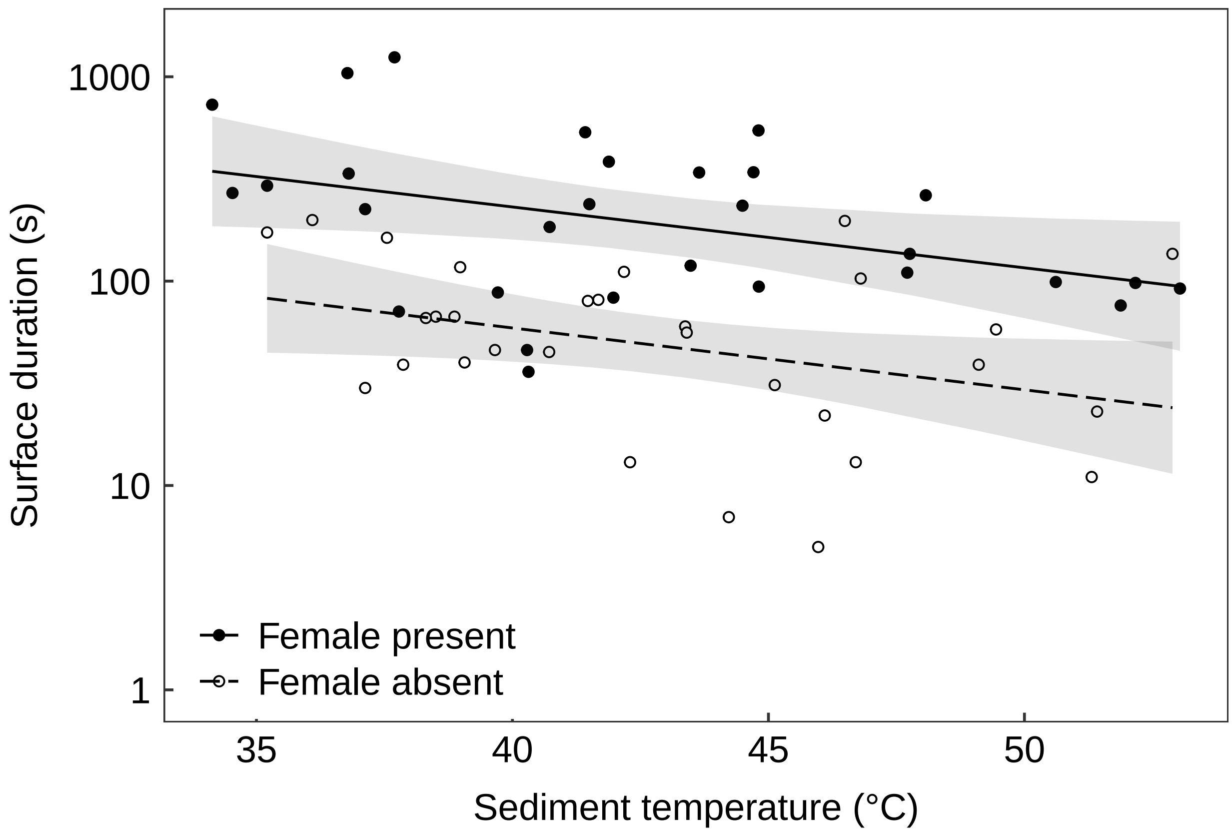



```
figure2 <- arrangeGrob(rbind(
  cbind(tempg1,tempg2),
  cbind(rhg1,rhg2),
  cbind(windg1,windg2),
  cbind(solg1,solg2)
   ))

ggsave("./figures/Figure2.tiff", figure2)
```

Figure 3

```
log.surf.pred.sed<-predict(surface.lm.sed, interval = "confidence")
surf.timeplot = cbind(surf.time, log.surf.pred.sed)

ggplot(surf.timeplot,aes(x = sed.temp, y = log.surf.dur)) +
  geom_point(shape=21, color = "black", aes(fill = trt))+
  scale_fill_manual(limits = c("female","control"),
                    values = c("black","white"),
                    labels = c("Female present","Female absent")) +
```

```r
  scale_y_continuous(limits = c(0, 3.176),
                     breaks = seq (0,3,1),
                     labels = c("1","10","100","1000")) +
  geom_ribbon(aes(ymin = lwr, ymax = upr, group = trt, color = NULL), alpha = .15) +
  geom_line(aes(y = fit, group = trt, linetype = trt)) +
  scale_linetype_manual(limits = c("female","control"),
                        values = c("solid", "longdash"),
                        labels = c("Female present","Female absent")) +
  labs(x="Sediment temperature (°C)", y="Surface duration (s)") +
  theme_bw() +
  theme(
    panel.grid = element_blank(),
    axis.ticks.length = unit(-0.05, "in"),
    aspect.ratio = 0.67,
    axis.text.y = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=14),
    axis.text.x = element_text(margin = unit(c(0.3,0.3,0.3,0.3), "cm"),
                               colour="black", size=14),
    axis.title.y = element_text(margin = margin(t = 0, r = 0, b = 0, l = 0),
                                colour="black", size=14),
    axis.title.x = element_text(margin = margin(t = 0, r = 0, b = 0, l = 0),
                                colour="black", size=14),
    legend.text=element_text(size=14),
    legend.title=element_blank(),
    legend.position = c(0.18, 0.1),
    legend.key.width = grid::unit(1.25, "lines"))
```




```r
ggsave("./figures/Figure3.tiff")
```

```
## Saving 6.5 x 4.5 in image
```

Version and package info

```r
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] gtable_0.3.0     gridExtra_2.3    hms_0.5.3        lubridate_1.7.4
##  [5] lme4_1.1-21      Matrix_1.2-18    carData_3.0-3    ggeffects_0.14.0
##  [9] forcats_0.5.0    stringr_1.4.0    dplyr_0.8.3      purrr_0.3.3
## [13] readr_1.3.1      tidyr_1.0.0      tibble_2.1.3     ggplot2_3.2.1
## [17] tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.1        jsonlite_1.6.1    splines_3.6.2     modelr_0.1.5
##  [5] assertthat_0.2.1  cellranger_1.1.0  yaml_2.2.0        pillar_1.4.3
##  [9] backports_1.1.5   lattice_0.20-38   glue_1.4.1        digest_0.6.25
## [13] rvest_0.3.5       minqa_1.2.4       colorspace_1.4-1  htmltools_0.4.0
## [17] pkgconfig_2.0.3   broom_0.5.3       haven_2.2.0       scales_1.1.0
## [21] openxlsx_4.1.4    rio_0.5.16        farver_2.0.3      generics_0.0.2
## [25] sjlabelled_1.1.2  withr_2.1.2       lazyeval_0.2.2    cli_2.0.2
## [29] magrittr_1.5      crayon_1.3.4      readxl_1.3.1      evaluate_0.14
## [33] fs_1.3.1          fansi_0.4.1       nlme_3.1-143      MASS_7.3-51.5
## [37] xml2_1.2.2        foreign_0.8-74    tools_3.6.2       data.table_1.12.8
## [41] lifecycle_0.1.0   munsell_0.5.0     reprex_0.3.0      zip_2.0.4
## [45] compiler_3.6.2    rlang_0.4.6       grid_3.6.2        nloptr_1.2.1
## [49] rstudioapi_0.11   labeling_0.3      rmarkdown_2.0     boot_1.3-24
## [53] abind_1.4-5       DBI_1.1.0         curl_4.3          sjmisc_2.8.3
## [57] R6_2.4.1          knitr_1.27        zeallot_0.1.0     utf8_1.1.4
## [61] insight_0.8.0     stringi_1.4.5     Rcpp_1.0.3        vctrs_0.2.1
## [65] dbplyr_1.4.2      tidyselect_0.2.5  xfun_0.12
```