

Cal Poly Herpetology CURE - Capture Data Analyses

Savannah Weaver

June 2021

Contents

Packages	1
Background and Goals	1
Data	1
Compute Scaled Mass Index	8
Check Data Distributions	19
Histograms & Q-Q Plots	19
Conclusion	38
Basic Figs & LMs	38
What affects osmolality & hematocrit?	38
Conclusion	74
What affects evaporative water loss?	74
Conclusion	113
LMMs	113
Hydration	113
CEWL	120
What to Present in the Paper	142

Packages

Background and Goals

This data was collected April - May 2021 during a course-based undergraduate research experience (CURE) in Herpetology, taught by Emily Taylor at Cal Poly, San Luis Obispo. This part of the study was conducted to describe the variation in hydrophysiology in *Sceloporus occidentalis* and to investigate what drives that variation. Please refer to **doi:** for full details.

In this document, we investigate differences in cutaneous evaporative water loss (CEWL) across body regions and dependent on environment, body size, condition, and hydration.

Data

Morphometrics and Blood Data

This data was collected upon capture of each lizard.

Variables in this dataframe: - date - collection/capture time for each lizard - individual ID for each lizard - sock ID used to capture each lizard (removed, not relevant to analyses) - SVL = snout-vent length - mass in grams - sex - if female, whether or not gravid (with eggs) - which eye the blood sample was taken from - percent hematocrit = percent of blood that's red blood cells - osmolality = a proxy of hydration, should be inversely related to water content of a lizard (this is the average of 1-3 replicates) - cloacal temperature at the time of CEWL measurement - processing time for each lizard, when all measurements were finished - hemolyzed = whether or not red blood cells burst and contaminated plasma

Before loading in this data, some incorrectly-measured hematocrit and osmolality were omitted: - hematocrit for individuals 1-16, due to observer error - osmolality for individual 19, due to instrumental error

```
# load and format data
morpho_blood_dat <- read.csv("./data/Herpetology_Data.csv", # filename
                             na.strings=c("", "NA") # fix empty cells
                             ) %>%

dplyr::mutate(# put date and time together
              collect_date_time = (paste(date, collect_time)),
              # replace some date-time values that have missing times
              collect_date_time = replace(collect_date_time,
                                           collect_date_time == "4/5/21 NA", NA),
              # correctly format date-time variable
              collect_date_time = as.POSIXct(collect_date_time,
                                              format = "%m/%d/%y %H:%M"),
              # correctly format date-only variable
              date = as.Date(date, format = "%m/%d/%y"),
              # correctly format collection time variable
              # format extracts just time after posix adds arbitrary date
              collect_time = (as.POSIXct(collect_time, format = "%H:%M")),
              # correctly format processing time variable
              processing_time = (as.POSIXct(processing_time, format = "%H:%M")),
              # set individual_ID variable as a factor, not numeric
              individual_ID = as.factor(individual_ID),
              # set sex variable as a factor, not character
              sex_M_F = as.factor(sex_M_F),
              # set gravidity variable as a factor, not character
              gravid_Y_N = as.factor(gravid_Y_N),
              # set blood sample eye variable as a factor, not character
              blood_sample_eye = as.factor(blood_sample_eye),
              # set hemolyzed variable as a factor, not character
              hemolyzed = as.factor(hemolyzed),
              # compute holding time as capture time - cloacal measurement time:
              hold_time = as.numeric(processing_time - collect_time)
              ) %>%

# remove two columns not relevant for statistics
dplyr::select(-sock_ID, -notes)
summary(morpho_blood_dat)
```

##	date	collect_time	individual_ID
##	Min. :2021-04-05	Min. :2021-07-02 10:17:00	1 : 1
##	1st Qu.:2021-04-19	1st Qu.:2021-07-02 12:36:00	2 : 1
##	Median :2021-04-26	Median :2021-07-02 12:48:00	3 : 1
##	Mean :2021-04-27	Mean :2021-07-02 12:51:12	4 : 1
##	3rd Qu.:2021-05-10	3rd Qu.:2021-07-02 13:03:00	5 : 1
##	Max. :2021-05-17	Max. :2021-07-02 15:57:00	6 : 1
##		NA's :3	(Other):142

```
##      SVL_mm      mass_g      sex_M_F gravid_Y_N blood_sample_eye
## Min.   :42.00   Min.    : 2.300   F: 48   N    : 22   both:  2
## 1st Qu.:63.00   1st Qu.: 9.125   M:100   Y    : 26   L    :  4
## Median :67.00   Median :11.200           NA's:100   R    :142
## Mean   :64.97   Mean    :10.586
## 3rd Qu.:69.00   3rd Qu.:12.725
## Max.   :73.00   Max.    :15.000
##
## hematocrit_percent osmolality_mmol_kg cloacal_temp_C
## Min.   :16.00      Min.    :293      Min.    :20.00
## 1st Qu.:33.00      1st Qu.:341      1st Qu.:22.00
## Median :35.00      Median :366      Median :23.00
## Mean   :35.36      Mean    :365      Mean    :23.48
## 3rd Qu.:38.00      3rd Qu.:387      3rd Qu.:25.00
## Max.   :54.00      Max.    :436      Max.    :28.00
## NA's   :27         NA's    :3         NA's    :7
## processing_time      hemolyzed collect_date_time
## Min.   :2021-07-02 12:44:00   N    :85   Min.    :2021-04-05 10:17:00
## 1st Qu.:2021-07-02 14:09:00   Y    :39   1st Qu.:2021-04-19 12:49:00
## Median :2021-07-02 15:17:30   NA's:24   Median :2021-04-26 15:34:00
## Mean   :2021-07-02 15:12:09           Mean    :2021-04-28 20:28:01
## 3rd Qu.:2021-07-02 16:15:15           3rd Qu.:2021-05-10 12:44:00
## Max.   :2021-07-02 17:38:00           Max.    :2021-05-17 13:01:00
## NA's   :8                         NA's    :3
## hold_time
## Min.   : 21.0
## 1st Qu.: 95.0
## Median :141.5
## Mean   :143.8
## 3rd Qu.:197.5
## Max.   :268.0
## NA's   :10
```

```
unique(morpho_blood_dat$date)
```

```
## [1] "2021-04-05" "2021-04-19" "2021-04-26" "2021-05-03" "2021-05-10"
## [6] "2021-05-17"
```

```
# export
write.csv(morpho_blood_dat, "exported_data/capture_hydration.csv")
```

I want to test if any IDs are missing, and which ones if so.

```
test <- c(seq(1, 150, by = 1))
lost <- test[test %nin% morpho_blood_dat$individual_ID]
lost
```

```
## [1] 23 56
```

Individuals 23 and 56 actually both do not exist because those numbers were skipped when assigning IDs, so we have all the individuals measured in the dataframe.

CEWL Data

First, load it all in and merge.

Variables in this dataframe are: - date - time - date_time combined variable - individual_ID for each lizard measured - region = where on the body CEWL was measured - TEWL_g_m2h = CEWL measurement

value in grams/sq-meter/hour - ambient_temp_C = temperature when and where measurement was taken
- ambient_RH_percent = relative humidity when and where measurement was taken - abs_humidity =
computed from RH using formula on this website: <https://carnotcycle.wordpress.com/2012/08/04/how-to-convert-relative-humidity-to-absolute-humidity/>

```
# week 1
CEWL_April_05 <- read.csv("../data/capture_CEWL/4-5-21-CEWL.csv", # filename
                          na.strings=c("", "NA")) %>% # fix empty cells

# rename and select the pertinent variables/cols
# I have to do this for each one
# so they all have the same number of columns for joining
dplyr::select(date = Date,
              Time, Status,
              ID = Comments,
              TEWL_g_m2h = TEWL..g..m2h.., # rename
              ambient_temp_C = AmbT..C., # rename
              ambient_RH_percent = AmbRH....
            )

# week 2
CEWL_April_19 <- read.csv("../data/capture_CEWL/4-19-21-CEWL.csv",
                          na.strings=c("", "NA")) %>%

dplyr::select(date = Date,
              Time, Status,
              ID = Comments,
              TEWL_g_m2h = TEWL..g..m2h.., # rename
              ambient_temp_C = AmbT..C., # rename
              ambient_RH_percent = AmbRH....
            )

# week 3
CEWL_April_26 <- read.csv("../data/capture_CEWL/4-26-21-CEWL.csv",
                          na.strings=c("", "NA")) %>%

dplyr::select(date = Date,
              Time, Status,
              ID = Comments,
              TEWL_g_m2h = TEWL..g..m2h.., # rename
              ambient_temp_C = AmbT..C., # rename
              ambient_RH_percent = AmbRH....
            )

# week 4
CEWL_May_3 <- read.csv("../data/capture_CEWL/5-3-21-CEWL.csv",
                      na.strings=c("", "NA")) %>%

dplyr::select(date = Date,
              Time, Status,
              ID = Comments,
              TEWL_g_m2h = TEWL..g..m2h.., # rename
              ambient_temp_C = AmbT..C., # rename
              ambient_RH_percent = AmbRH....
            )

# week 5
CEWL_May_10 <- read.csv("../data/capture_CEWL/5-10-21-CEWL.csv",
```

```

        na.strings=c("", "NA")) %>%
dplyr::select(date = Date,
              Time, Status,
              ID = Comments,
              TEWL_g_m2h = TEWL..g..m2h.., # rename
              ambient_temp_C = AmbT..C., # rename
              ambient_RH_percent = AmbRH....
            )

# week 6
CEWL_May_17 <- read.csv("../data/capture_CEWL/5-17-21-CEWL.csv",
                      na.strings=c("", "NA")) %>%
dplyr::select(date = Date,
              Time, Status,
              ID = Comments,
              TEWL_g_m2h = TEWL..g..m2h.., # rename
              ambient_temp_C = AmbT..C., # rename
              ambient_RH_percent = AmbRH....
            )

# merge all CEWL datafiles & reformat
CEWL <- CEWL_April_05 %>% # week 1
# join with weeks 2-6
rbind(., CEWL_April_19,
       CEWL_April_26,
       CEWL_May_3,
       CEWL_May_10,
       CEWL_May_17
      ) %>%
# remove any unsuccessful measurements
dplyr::filter(Status == "Normal") %>%
# extract individual_ID and region separately from the "ID" variable
separate(ID, c("individual_ID", "region")) %>%
# reformat data
dplyr::mutate(# paste and format date-time variable
              CEWL_date_time = as.POSIXct(paste(date, Time),
                                             format = "%m/%d/%y %I:%M:%S %p"),
              # reformat date only
              date = as.Date(date, format = "%m/%d/%y"),
              # reformat time
              # format extracts just time after posix adds arbitrary date
              # but then it's a character again...
              Time = format(as.POSIXct(Time, format = "%I:%M:%S %p"),
                            format = "%H:%M:%S"),
              # format individual ID as a factor
              individual_ID = as.factor(individual_ID),
              # set body region as a factor variable after getting only the consistent characters due
              region = as.factor(substring(region, 1, 4)),
              # convert RH to absolute humidity
              abs_humidity_g_m3 = ((6.112 * exp((17.67*ambient_temp_C)/(ambient_temp_C + 243.5)) * am
            ) %>%
# remove cols not relevant to stats
dplyr::select(-Status) %>%

```

```
# remove any rows with missing values
dplyr::filter(complete.cases())
summary(CEWL)
```

```
##      date                Time      individual_ID  region
## Min.   :2021-04-05   Length:699      01      : 5    dewl:139
## 1st Qu.:2021-04-19   Class :character  02      : 5    dors:141
## Median :2021-04-26   Mode  :character  03      : 5    head:141
## Mean   :2021-04-28                                04      : 5    mite:137
## 3rd Qu.:2021-05-10                                05      : 5    vent:141
## Max.   :2021-05-17                                06      : 5
##                                           (Other):669
##      TEWL_g_m2h    ambient_temp_C  ambient_RH_percent
## Min.   : 3.41    Min.   :22.30    Min.   :34.00
## 1st Qu.:17.09    1st Qu.:23.00    1st Qu.:41.30
## Median :22.00    Median :23.20    Median :45.20
## Mean   :25.87    Mean   :23.44    Mean   :43.56
## 3rd Qu.:32.59    3rd Qu.:23.80    3rd Qu.:46.30
## Max.   :96.16    Max.   :25.30    Max.   :53.10
##
## CEWL_date_time      abs_humidity_g_m3
## Min.   :2021-04-05 13:24:15    Min.   : 6.989
## 1st Qu.:2021-04-19 14:07:34    1st Qu.: 8.611
## Median :2021-04-26 17:10:23    Median : 9.483
## Mean   :2021-04-28 23:39:45    Mean   : 9.188
## 3rd Qu.:2021-05-10 16:03:10    3rd Qu.: 9.899
## Max.   :2021-05-17 17:22:31    Max.   :10.632
##
```

Write CEWL dataframe as a csv for use in other analyses:

```
write.csv(CEWL, "exported_data/capture_CEWL.csv")
```

Weather Data

This data was obtained from <http://www.itrc.org/databases/precip/> (Adcon Server Data) to test the effect of ambient conditions on CEWL. This is different from the ambient conditions already measured with CEWL, which are the temperature and humidity around the measurement device at the time of measurement. We think that the temperature, humidity, wind speed, and solar radiation the lizard was exposed to prior to capture may also affect CEWL.

tbd = daylight savings

```
# load in csvs and put all in one dataframe
weather <- read.csv("./data/weather/4_5Weather.csv", sep = ';') %>%
  rbind(read.csv("./data/weather/4_19Weather.csv", sep = ';')) %>%
  rbind(read.csv("./data/weather/5_3Weather.csv", sep = ';')) %>%
  rbind(read.csv("./data/weather/5_10Weather.csv", sep = ';')) %>%
  rbind(read.csv("./data/weather/5_17Weather.csv", sep = ';')) %>%
  # add a variable for combined date-time
  mutate(collect_date_time = as.POSIXct(paste(Date, Time),
                                          format = "%m/%d/%y %I:%M:%S %p")) %>%
  # remove lonely date and time
  dplyr::select(-Date, -Time)
```

The weather data is only every 15 minutes, but I want to match it to any minute measurement, so I need to

interpolate the values for each minute.

First, make a separate dataframe with every minute for each of those days.

```
all_times <- data.frame(collect_date_time = c(# April 5
      seq(from = as.POSIXct("2021-04-05 10:00"),
        to = as.POSIXct("2021-04-05 16:00"),
        by="min"),
    # April 19
      seq(from = as.POSIXct("2021-04-19 10:00"),
        to = as.POSIXct("2021-04-19 16:00"),
        by="min"),
    # April 26
      seq(from = as.POSIXct("2021-04-26 10:00"),
        to = as.POSIXct("2021-04-26 16:00"),
        by="min"),
    # May 3
      seq(from = as.POSIXct("2021-05-03 10:00"),
        to = as.POSIXct("2021-05-03 16:00"),
        by="min"),
    # May 10
      seq(from = as.POSIXct("2021-05-10 10:00"),
        to = as.POSIXct("2021-05-10 16:00"),
        by="min"),
    # May 17
      seq(from = as.POSIXct("2021-05-17 10:00"),
        to = as.POSIXct("2021-05-17 16:00"),
        by="min")
    ))
```

Next, merge the weather data into the times dataframe and interpolate the temperature and humidity between measurements.

```
all_times_weather <- all_times %>% # time only dataframe
  # add weather measurements based on matching date-time
  left_join(weather, by = 'collect_date_time') %>%
  # convert temperature units, thanks America
  mutate(temp_C = fahrenheit.to.celsius(Temperature_F, round = 2),
    # interpolate temperatures
    temp_C_interpol = na.approx(temp_C),
    # interpolate humidities
    RH_percent_interpol = na.approx(RH_percent),
    # interpolate Wind Speeds
    Wind_mph_interpol = na.approx(Wind_Speed_mph),
    # interpolate solar radiation
    Solar_rad_Wm2_interpol = na.approx(Pyranometer_W_m),
    # compute absolute humidity
    abs_humidity_g_m3_interpol = ((6.112 * exp((17.67*temp_C_interpol)/(temp_C_interpol + 243.5)))
  ) %>%
  # keep only the relevant variables
  dplyr::select(collect_date_time,
    temp_C_interpol,
    RH_percent_interpol,
    abs_humidity_g_m3_interpol,
    Wind_mph_interpol,
    Solar_rad_Wm2_interpol)
```

```
summary(all_times_weather)
```

```
## collect_date_time      temp_C_interpol RH_percent_interpol
## Min.   :2021-04-05 10:00:00   Min.   :13.28   Min.   :38.20
## 1st Qu.:2021-04-19 13:00:15   1st Qu.:16.54   1st Qu.:56.77
## Median :2021-04-30 01:00:00   Median :17.78   Median :67.65
## Mean   :2021-04-28 21:00:00   Mean   :18.78   Mean   :65.52
## 3rd Qu.:2021-05-10 12:59:45   3rd Qu.:20.48   3rd Qu.:72.30
## Max.   :2021-05-17 16:00:00   Max.   :25.78   Max.   :92.10
## abs_humidity_g_m3_interpol Wind_mph_interpol Solar_rad_Wm2_interpol
## Min.   : 8.497              Min.   :0.100   Min.   : 356.9
## 1st Qu.: 9.634              1st Qu.:4.340   1st Qu.: 743.2
## Median :10.616              Median :4.567   Median : 882.6
## Mean   :10.361              Mean   :4.574   Mean   : 860.2
## 3rd Qu.:10.912              3rd Qu.:5.020   3rd Qu.: 979.5
## Max.   :11.790              Max.   :7.100   Max.   :1037.5
```

Compute Scaled Mass Index

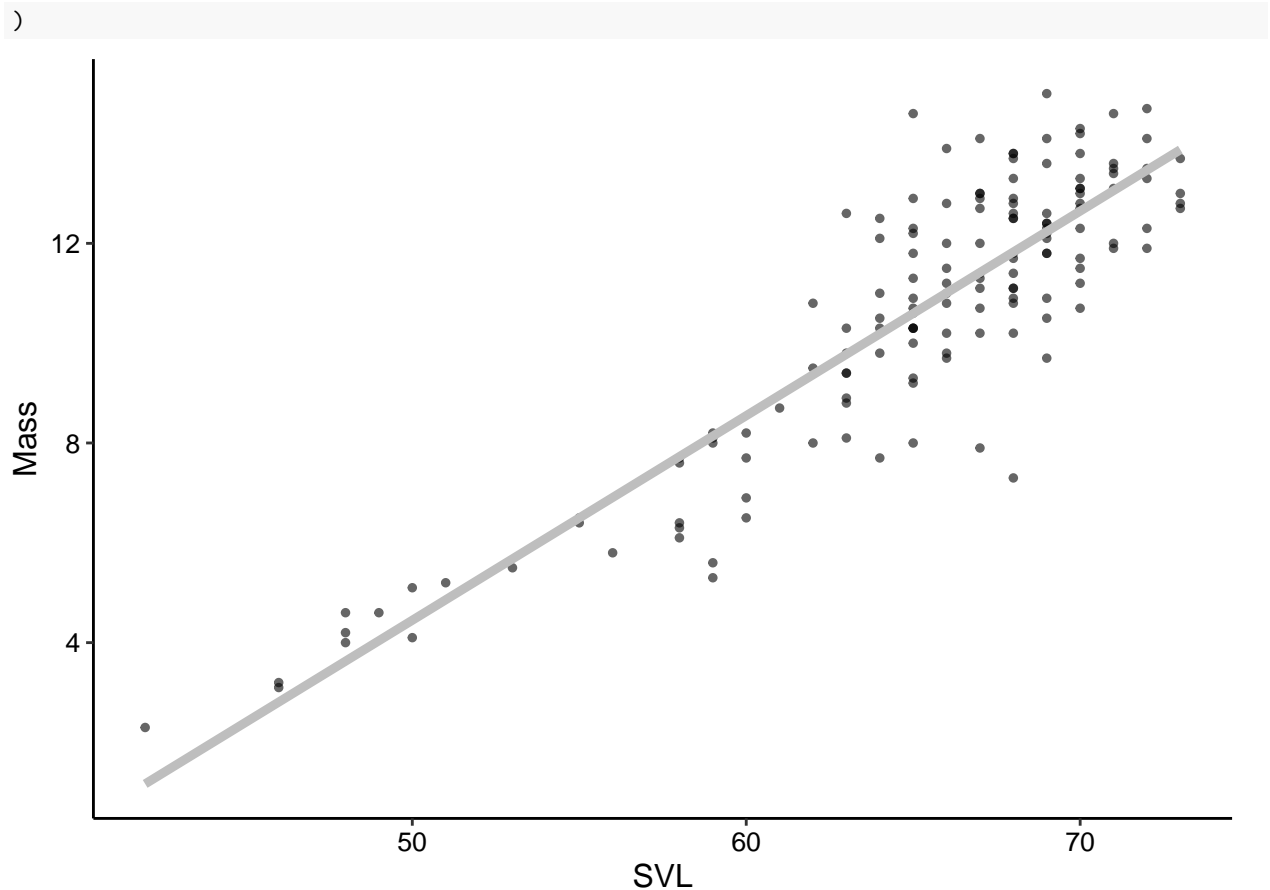
This is also known as the body condition index, or log-log residuals.

I calculate as described by: Peig, J., & Green, A. J. (2009). New perspectives for estimating body condition from mass/length data: The scaled mass index as an alternative method. *Oikos*, 118(12), 1883–1891. <https://doi.org/10.1111/j.1600-0706.2009.17643.x>

Step 1: mass ~ SVL

plot:

```
morpho_blood_dat %>%
  ggplot(data = .) +
  geom_point(aes(x = SVL_mm,
                 y = mass_g,
                 ),
             size = 1,
             alpha = 0.6) +
  stat_smooth(aes(x = SVL_mm,
                  y = mass_g,
                  ),
              formula = y ~ x,
              method = "lm",
              color = "gray",
              se = F,
              size = 1.6,
              alpha = 1 ) +
  theme_classic() +
  xlab("SVL") +
  ylab("Mass") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0
```

create a simple linear regression

```
mass_SVL_SLR <- lm(data = morpho_blood_dat, mass_g ~ SVL_mm)
summary(mass_SVL_SLR)
```

```
##
## Call:
## lm(formula = mass_g ~ SVL_mm, data = morpho_blood_dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.5265 -0.8762 -0.0024  0.6735  4.0031
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.04514    1.14303  -14.04  <2e-16 ***
## SVL_mm       0.40988    0.01751   23.40  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.316 on 146 degrees of freedom
## Multiple R-squared:  0.7895, Adjusted R-squared:  0.7881
## F-statistic: 547.7 on 1 and 146 DF, p-value: < 2.2e-16
```

look for outliers by calculating residuals

```

mass_SVL_SLR_residuals <- morpho_blood_dat %>%
  # not working with function, so just compute by hand...
  mutate(#y_hat = predict(mass_SVL_SLR),
         #e = residuals(mass_SVL_SLR),
         y_hat = 0.40988*SVL_mm - 16.04514,
         e = y_hat - mass_g
        ) %>%
  dplyr::filter(e < 3 & e > -3)

```

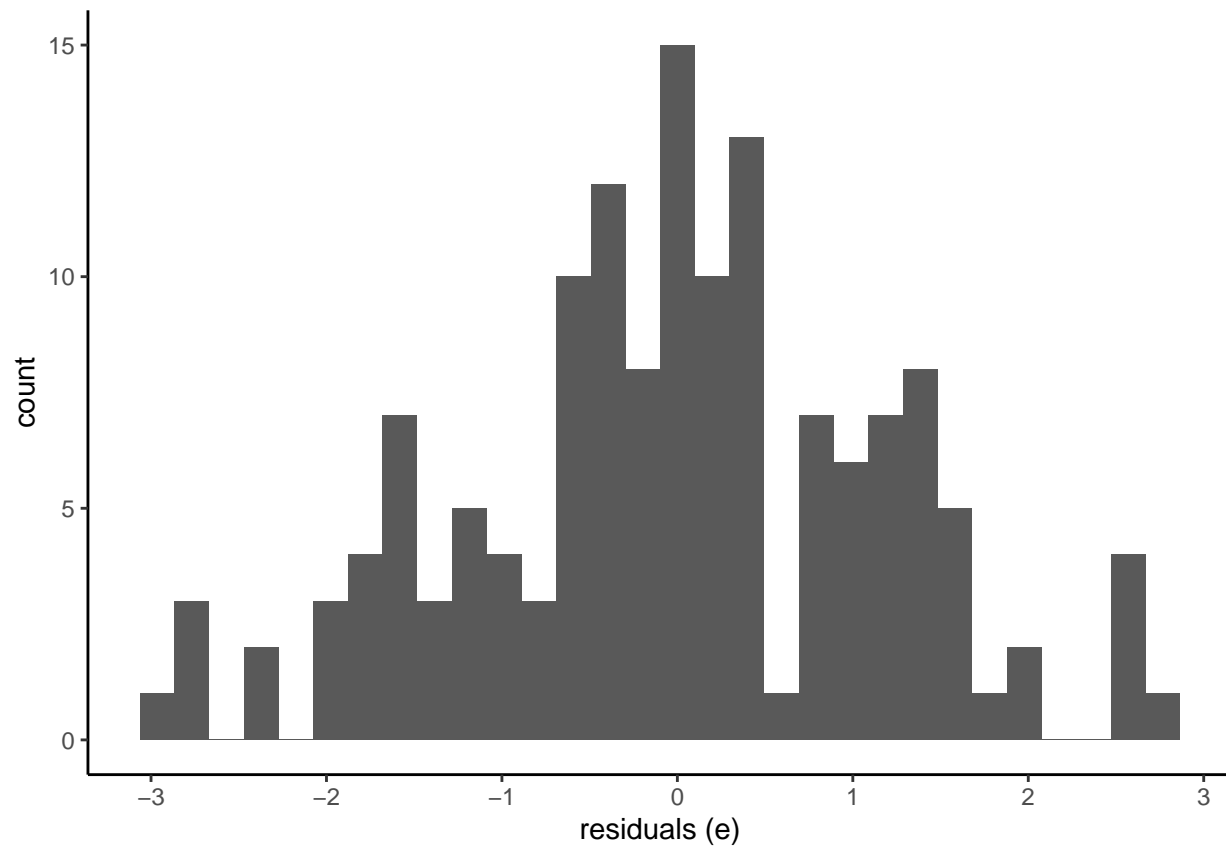
plot residuals

```

ggplot(data = mass_SVL_SLR_residuals,
       aes(x = e)) +
  geom_histogram() +
  theme_classic() +
  xlab("residuals (e)")

```

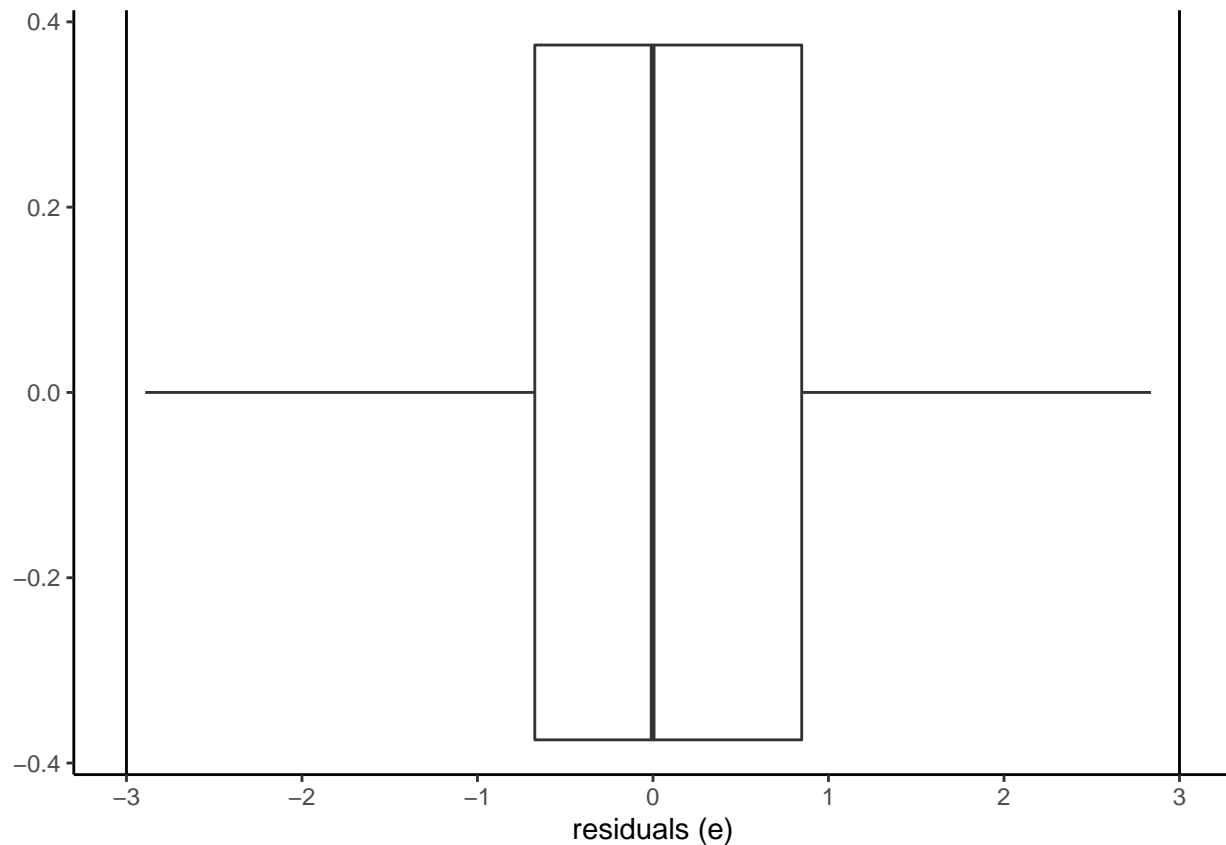
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



```

ggplot(data = mass_SVL_SLR_residuals,
       aes(x = e)) +
  geom_boxplot() +
  theme_classic() +
  scale_x_continuous(breaks = c(seq(-5, 5, 1))) +
  xlab("residuals (e)") +
  geom_vline(xintercept = -3, lty = 1) +
  geom_vline(xintercept = 3, lty = 1)

```



From the histogram, there are clearly points disconnected from the main curve. In the boxplot, we can distinguish that those outlying points have residuals <-3.5 and >3 , so I'll go back and filter those out in the code chunk before making the distribution plots. Now the boxplot has no dots outside the main distribution.

And, check mean residual value:

```
mass_SVL_SLR_residuals %>%
  summarise(mean_res = mean(e),
            med = median(e))
```

```
##      mean_res      med
## 1 -0.02769117 -0.00174
```

Not zero, which is not ideal, but pretty close.

Next, check for high leverage points:

```
# compute values for observations
high_leverage <- data.frame(H = hatvalues(mass_SVL_SLR)
                             ) %>% mutate(row = rownames(.))

# compute cutoff value
h_bar <- (3*sum(high_leverage$H))/nrow(high_leverage)

# add to original dataframe
# see which observations have extremely high leverage (if any)
high_leverage_dat <- mass_SVL_SLR_residuals %>%
  mutate(row = rownames(.)) %>%
  left_join(., high_leverage, by = "row") %>%
  dplyr::filter(H > h_bar)
```

```
high_leverage_dat
```

```
##      date      collect_time individual_ID SVL_mm mass_g sex_M_F
## 1  2021-04-05 2021-07-02 10:38:00         4    48   4.2      M
## 2  2021-04-05 2021-07-02 10:17:00         5    50   4.1      M
## 3  2021-04-05 2021-07-02 10:47:00         6    48   4.6      M
## 4  2021-04-05 2021-07-02 10:42:00         8    42   2.3      M
## 5  2021-04-05 2021-07-02 13:27:00         9    46   3.1      F
## 6  2021-04-26 2021-07-02 12:38:00        57    65  10.3      F
## 7  2021-04-26 2021-07-02 12:57:00        64    65  10.3      M
## 8  2021-04-26 2021-07-02 13:02:00        67    64  10.5      M
## 9  2021-05-03 2021-07-02 12:51:00        87    70  14.3      M
## 10 2021-05-17 2021-07-02 12:19:00       123    72  11.9      M
##      gravid_Y_N blood_sample_eye hematocrit_percent osmolality_mmol_kg
## 1          <NA>                R                NA                341
## 2          <NA>                R                NA                354
## 3          <NA>                L                NA                355
## 4          <NA>                R                NA                304
## 5           N                L                NA                NA
## 6           N                R                30                388
## 7          <NA>                R                30                378
## 8          <NA>                R                32                402
## 9          <NA>                R                35                394
## 10         <NA>                R                52                362
##      cloacal_temp_C      processing_time hemolyzed      collect_date_time hold_time
## 1          26 2021-07-02 14:02:00      Y 2021-04-05 10:38:00      204
## 2          25 2021-07-02 13:59:00      Y 2021-04-05 10:17:00      222
## 3          24 2021-07-02 14:06:00      N 2021-04-05 10:47:00      199
## 4          23 2021-07-02 14:20:00      N 2021-04-05 10:42:00      218
## 5          23 2021-07-02 14:43:00    <NA> 2021-04-05 13:27:00       76
## 6          26 2021-07-02 13:27:00      N 2021-04-26 12:38:00       49
## 7          22 2021-07-02 15:20:00      Y 2021-04-26 12:57:00      143
## 8          24 2021-07-02 15:41:00      N 2021-04-26 13:02:00      159
## 9          24 2021-07-02 14:32:00      N 2021-05-03 12:51:00      101
## 10         27 2021-07-02 12:47:00    <NA> 2021-05-17 12:19:00       28
##      y_hat      e row      H
## 1  3.62910 -0.57090   4 0.05776372
## 2  4.44886  0.34886   5 0.04645120
## 3  3.62910 -0.97090   6 0.05776372
## 4  1.16982 -1.13018   8 0.10020003
## 5  2.80934 -0.29066   9 0.07049270
## 6 10.59706  0.29706  54 0.07049270
## 7 10.59706  0.29706  60 0.04132611
## 8 10.18718 -0.31282  63 0.05776372
## 9 12.64646 -1.65354  83 0.05193040
## 10 13.46622  1.56622 118 0.04645120
```

The points for individuals 4, 5, 6, 8, 9, 57, 64, 67, 87, and 123 seem to be high-leverage, so we will try removing them.

Check for influential points based on Cook's distance:

```
# get Cook's distance
cooks <- data.frame(c = cooks.distance(mass_SVL_SLR) # specify model name
) %>% mutate(row = rownames(.))
```

```

# add to original dataframe
influential <- mass_SVL_SLR_residuals %>%
  mutate(row = rownames(.)) %>%
  left_join(., cooks, by = "row")

# see moderately influential points
cook_mod_inf <- influential %>%
  dplyr::filter(c>0.5)
cook_mod_inf

```

```

## [1] date          collect_time    individual_ID    SVL_mm
## [5] mass_g         sex_M_F        gravid_Y_N       blood_sample_eye
## [9] hematocrit_percent osmolality_mmol_kg cloacal_temp_C   processing_time
## [13] hemolyzed      collect_date_time hold_time        y_hat
## [17] e             row            c
## <0 rows> (or 0-length row.names)

```

There are no even moderately-influential points, at least based on Cook's distance, so there's nothing to potentially remove.

Create new SLR and check mean residual value after removing outliers and high leverage points:

```

# create new dataframe with filtered data
cleaned_SMI_dat <- mass_SVL_SLR_residuals %>% # already had outliers removed
  mutate(row = rownames(.)) %>%
  # add high leverage point info
  left_join(., high_leverage, by = "row") %>%
  # remove high leverage points
  dplyr::filter(H < h_bar)

# model
mass_SVL_SLR2 <- lm(data = cleaned_SMI_dat, mass_g ~ SVL_mm)
summary(mass_SVL_SLR2)

```

```

##
## Call:
## lm(formula = mass_g ~ SVL_mm, data = cleaned_SMI_dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.72134 -0.88611  0.00146  0.70819  2.87193
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.32109    1.28410  -13.49  <2e-16 ***
## SVL_mm       0.42953     0.01953   21.99  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.209 on 133 degrees of freedom
## Multiple R-squared:  0.7843, Adjusted R-squared:  0.7827
## F-statistic: 483.6 on 1 and 133 DF, p-value: < 2.2e-16

# compute residuals
mass_SVL_SLR2_residuals <- cleaned_SMI_dat %>%
  mutate(y_hat = predict(mass_SVL_SLR2),

```

```

e = residuals(mass_SVL_SLR2))

# check residuals values
mass_SVL_SLR2_residuals %>%
  summarise(mean_res = mean(e),
            med = median(e))

```

```

##           mean_res           med
## 1 4.654038e-17 0.001458016

```

The residuals are much much smaller, so I believe it is the right choice to remove the high leverage points. The “cleaned_SMI_dat” will be used to calculate the equation for SMI.

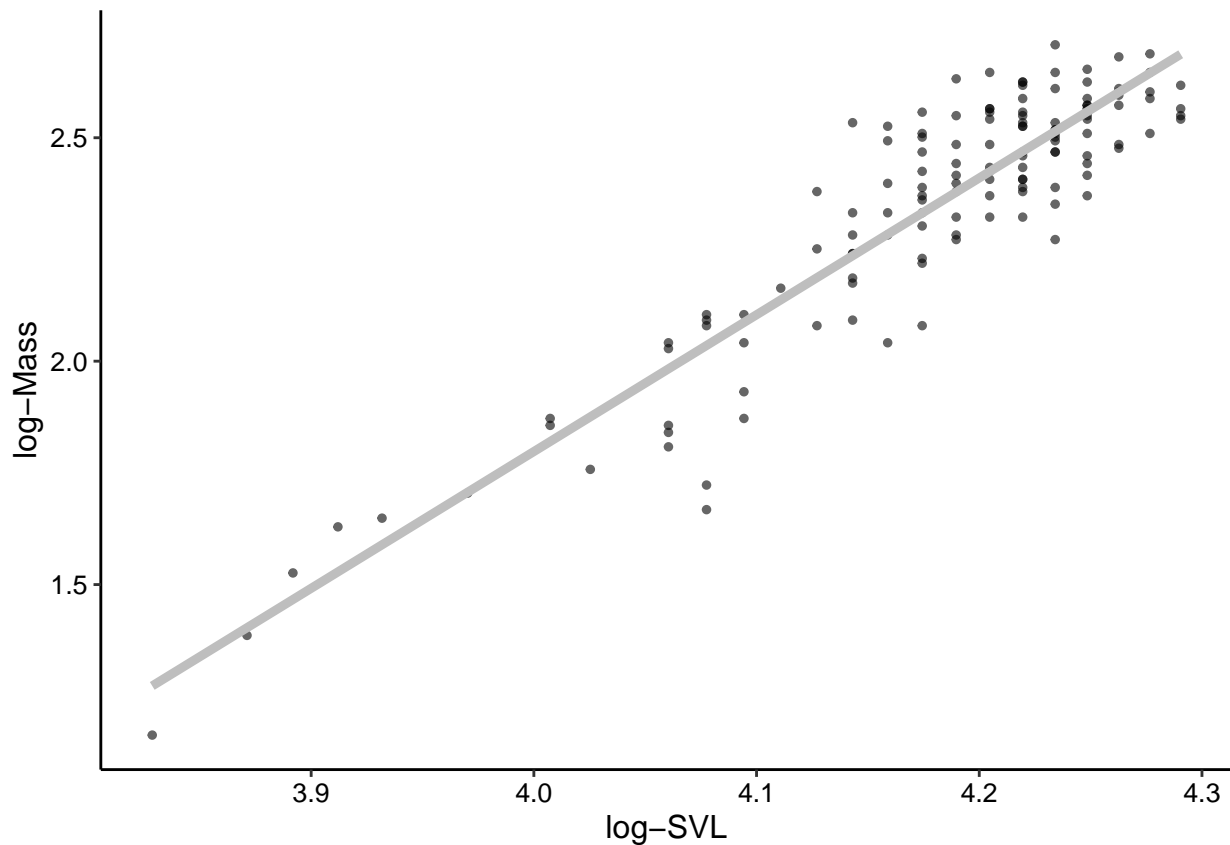
Step 2: make log-log relationship

plot and calculate SLR for filtered data

```

cleaned_SMI_dat %>%
  ggplot(data = .) +
    geom_point(aes(x = log(SVL_mm),
                  y = log(mass_g),
                  ),
              size = 1,
              alpha = 0.6) +
    stat_smooth(aes(x = log(SVL_mm),
                    y = log(mass_g),
                    ),
               formula = y ~ x,
               method = "lm",
               color = "gray",
               se = F,
               size = 1.6,
               alpha = 1 ) +
    theme_classic() +
    xlab("log-SVL") +
    ylab("log-Mass") +
    theme(text = element_text(color = "black",
                              family = "sans",
                              size = 12),
          axis.text = element_text(color = "black",
                                    family = "sans",
                                    size = 10),
          legend.text.align = 0
    )

```



SLR

```
log_mass_SVL_SLR <- lm(data = cleaned_SMI_dat,
                        log(mass_g) ~ log(SVL_mm))
summary(log_mass_SVL_SLR)
```

```
##
## Call:
## lm(formula = log(mass_g) ~ log(SVL_mm), data = cleaned_SMI_dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36738 -0.07518  0.00334  0.06423  0.29781
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.4465     0.4858  -21.50  <2e-16 ***
## log(SVL_mm)   3.0611     0.1162   26.34  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1175 on 133 degrees of freedom
## Multiple R-squared:  0.8391, Adjusted R-squared:  0.8379
## F-statistic: 693.6 on 1 and 133 DF, p-value: < 2.2e-16
```

compute standardized major axis using this regression equation:

```
r <- sqrt(0.8391) # Pearson's correlection coefficient (sqrt of R-squared)
b_OLS <- 3.0611 # regression slope
```

```
b_SMA <- b_OLS/r
```

also get a value for L0:

```
L0 <- mean(cleaned_SMI_dat$SVL_mm)
```

Step 3: calculate scaled mass index

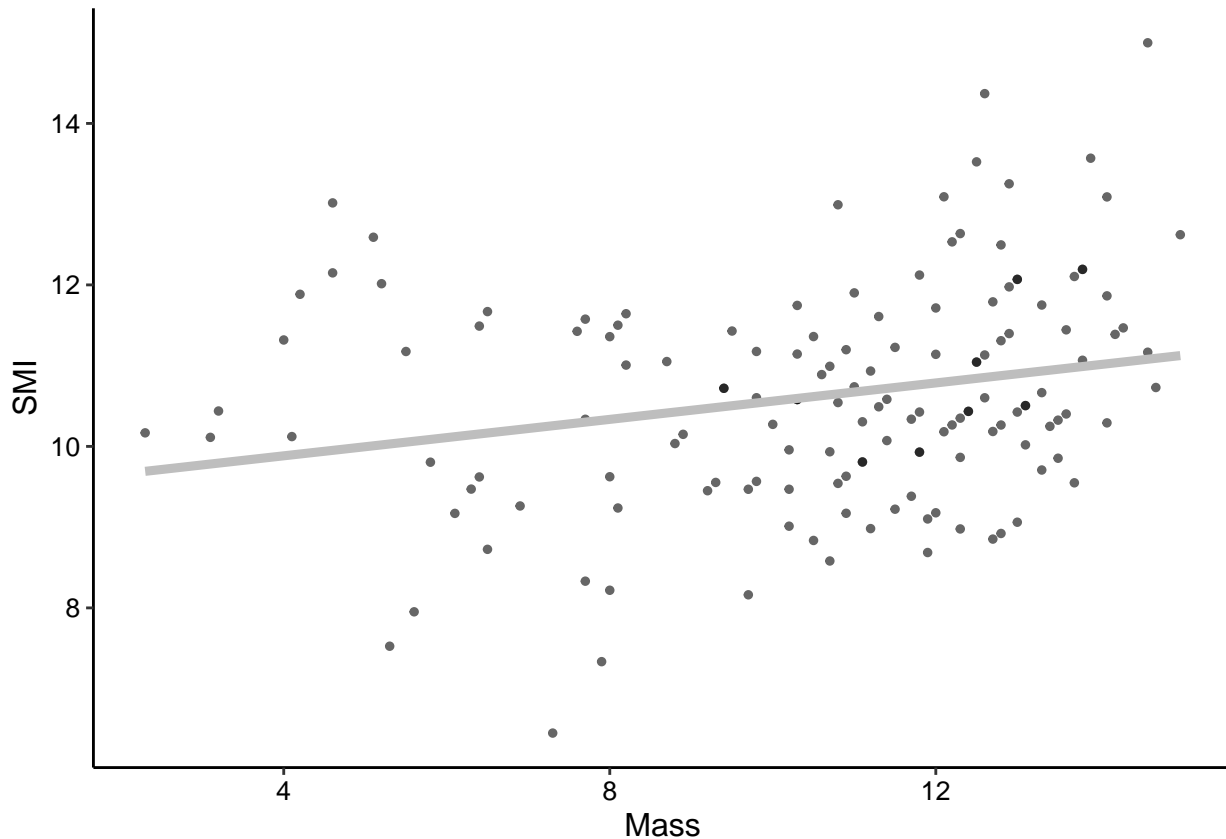
(And join weather data.)

```
morpho_blood_SMI <- morpho_blood_dat %>%  
  mutate(SMI = mass_g * ((L0/SVL_mm) ^ b_SMA)) %>%  
  left_join(all_times_weather,  
    by = c("collect_date_time")  
  )
```

Check

Look at the difference between regular mass and SMI:

```
morpho_blood_SMI %>%  
  ggplot(data = .) +  
  geom_point(aes(x = mass_g,  
    y = SMI,  
    ),  
    size = 1,  
    alpha = 0.6) +  
  stat_smooth(aes(x = mass_g,  
    y = SMI,  
    ),  
    formula = y ~ x,  
    method = "lm",  
    color = "gray",  
    se = F,  
    size = 1.6,  
    alpha = 1 ) +  
  theme_classic() +  
  xlab("Mass") +  
  ylab("SMI") +  
  theme(text = element_text(color = "black",  
    family = "sans",  
    size = 12),  
    axis.text = element_text(color = "black",  
    family = "sans",  
    size = 10),  
    legend.text.align = 0  
  )
```

Join Data

There are several CEWL measurements for each of the other measures, so I'm going to join two ways. Each way allows slightly different analyses.

First, with CEWL as the primary dataframe. This means each of the other variables will be duplicated for each lizards CEWL measurements.

```
CEWL_data_full <- CEWL %>%
  left_join(morpho_blood_SMI,
            by = c("date", "individual_ID")
            )
summary(CEWL_data_full)
```

```
##      date           Time      individual_ID  region
## Min.   :2021-04-05   Length:699      01      : 5   dewl:139
## 1st Qu.:2021-04-19   Class :character 02      : 5   dors:141
## Median :2021-04-26   Mode  :character 03      : 5   head:141
## Mean   :2021-04-28                                     04      : 5   mite:137
## 3rd Qu.:2021-05-10                                     05      : 5   vent:141
## Max.   :2021-05-17                                     06      : 5
##                                                         (Other):669
##      TEWL_g_m2h    ambient_temp_C    ambient_RH_percent
## Min.   : 3.41     Min.   :22.30     Min.   :34.00
## 1st Qu.:17.09     1st Qu.:23.00     1st Qu.:41.30
## Median :22.00     Median :23.20     Median :45.20
## Mean   :25.87     Mean   :23.44     Mean   :43.56
## 3rd Qu.:32.59     3rd Qu.:23.80     3rd Qu.:46.30
```

```

## Max. :96.16 Max. :25.30 Max. :53.10
##
## CEWL_date_time abs_humidity_g_m3 collect_time
## Min. :2021-04-05 13:24:15 Min. : 6.989 Min. :2021-07-02 11:29:00
## 1st Qu.:2021-04-19 14:07:34 1st Qu.: 8.611 1st Qu.:2021-07-02 12:37:00
## Median :2021-04-26 17:10:23 Median : 9.483 Median :2021-07-02 12:48:00
## Mean :2021-04-28 23:39:45 Mean : 9.188 Mean :2021-07-02 12:55:48
## 3rd Qu.:2021-05-10 16:03:10 3rd Qu.: 9.899 3rd Qu.:2021-07-02 13:02:15
## Max. :2021-05-17 17:22:31 Max. :10.632 Max. :2021-07-02 15:44:00
## NA's :59
## SVL_mm mass_g sex_M_F gravid_Y_N blood_sample_eye
## Min. :46.00 Min. : 3.20 F :216 N : 91 both: 10
## 1st Qu.:64.00 1st Qu.: 9.70 M :439 Y :125 L : 0
## Median :67.00 Median :11.40 NA's: 44 NA's:483 R :645
## Mean :65.81 Mean :10.88 NA's: 44
## 3rd Qu.:69.00 3rd Qu.:12.80
## Max. :73.00 Max. :15.00
## NA's :44 NA's :44
## hematocrit_percent osmolality_mmol_kg cloacal_temp_C
## Min. :16.0 Min. :293.0 Min. :20.0
## 1st Qu.:33.0 1st Qu.:347.0 1st Qu.:22.0
## Median :36.0 Median :368.0 Median :23.0
## Mean :35.4 Mean :366.8 Mean :23.4
## 3rd Qu.:38.0 3rd Qu.:387.0 3rd Qu.:25.0
## Max. :54.0 Max. :436.0 Max. :28.0
## NA's :119 NA's :49 NA's :49
## processing_time hemolyzed collect_date_time
## Min. :2021-07-02 12:44:00 N :368 Min. :2021-04-05 13:25:00
## 1st Qu.:2021-07-02 14:14:00 Y :179 1st Qu.:2021-04-19 13:09:30
## Median :2021-07-02 15:25:00 NA's:152 Median :2021-05-03 12:40:00
## Mean :2021-07-02 15:16:48 Mean :2021-05-01 01:04:48
## 3rd Qu.:2021-07-02 16:18:00 3rd Qu.:2021-05-10 12:52:00
## Max. :2021-07-02 17:38:00 Max. :2021-05-17 13:01:00
## NA's :59 NA's :59
## hold_time SMI temp_C_interpol RH_percent_interpol
## Min. : 21.0 Min. : 6.450 Min. :15.67 Min. :44.29
## 1st Qu.: 91.0 1st Qu.: 9.624 1st Qu.:16.66 1st Qu.:57.51
## Median :132.0 Median :10.505 Median :18.68 Median :68.10
## Mean :140.5 Mean :10.573 Mean :18.77 Mean :66.53
## 3rd Qu.:189.0 3rd Qu.:11.444 3rd Qu.:19.96 3rd Qu.:72.54
## Max. :268.0 Max. :14.999 Max. :23.61 Max. :81.10
## NA's :69 NA's :44 NA's :59 NA's :59
## abs_humidity_g_m3_interpol Wind_mph_interpol Solar_rad_Wm2_interpol
## Min. : 8.912 Min. :3.773 Min. : 587.0
## 1st Qu.: 9.948 1st Qu.:4.577 1st Qu.: 741.9
## Median :10.743 Median :5.000 Median : 951.3
## Mean :10.568 Mean :4.945 Mean : 892.3
## 3rd Qu.:10.908 3rd Qu.:5.233 3rd Qu.:1032.9
## Max. :11.520 Max. :6.200 Max. :1037.5
## NA's :59 NA's :59 NA's :59

```

```
unique(CEWL_data_full$individual_ID)
```

```

## [1] 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18
## [19] 19 20 21 26 27 30 29 24 31 32 34 33 35 36 37 38 39 40

```

```
## [37] 42 41 43 44 45 46 47 52 53 48 50 51 54 55 57 58 59 60
## [55] 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
## [73] 79 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
## [91] 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 115 116 119
## [109] 113 114 117 118 122 120 123 124 125 126 127 128 129 130 131 132 133 134
## [127] 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
## 157 Levels: 01 02 03 04 05 06 07 08 09 10 100 101 102 103 104 105 106 ... 121
```

Export Data

I want to save the data as a csv for loading into other analyses.

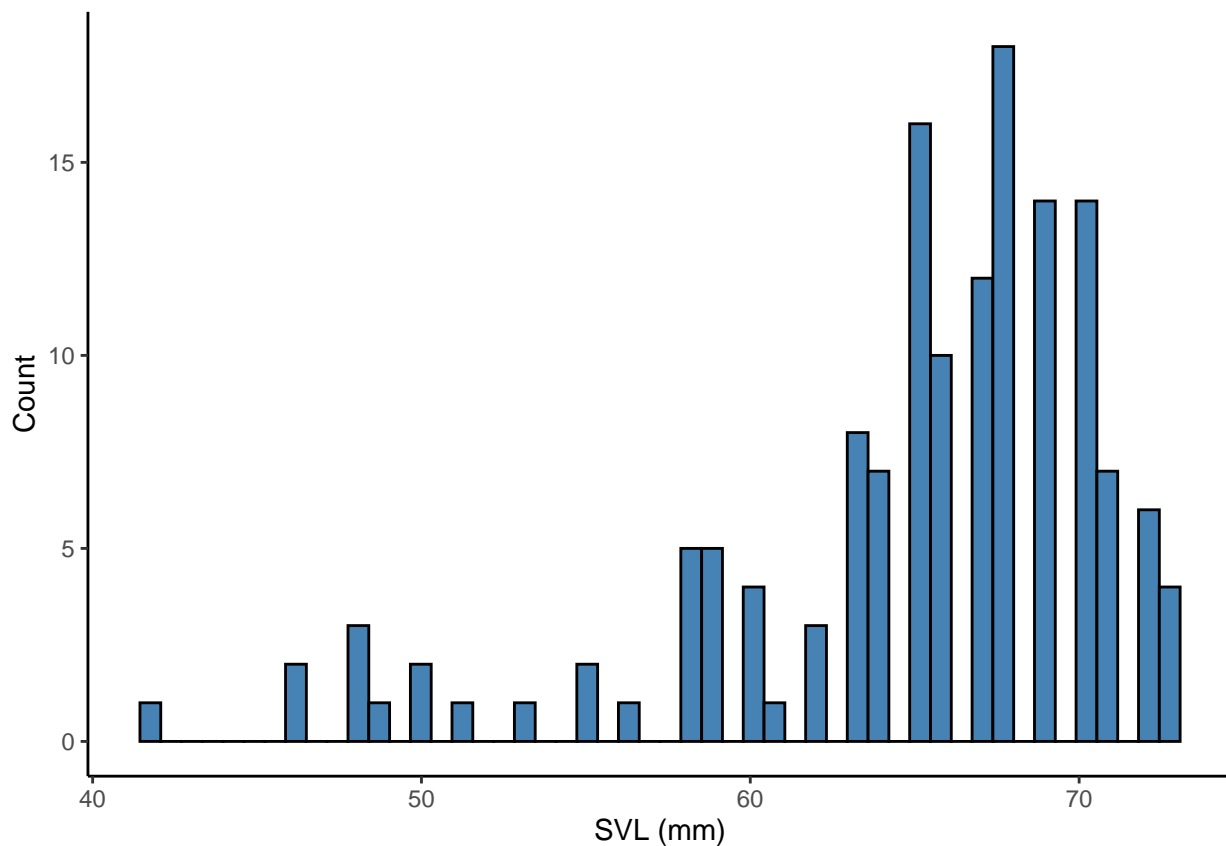
```
write.csv(CEWL_data_full, "exported_data/capture_data_long.csv")
```

Check Data Distributions

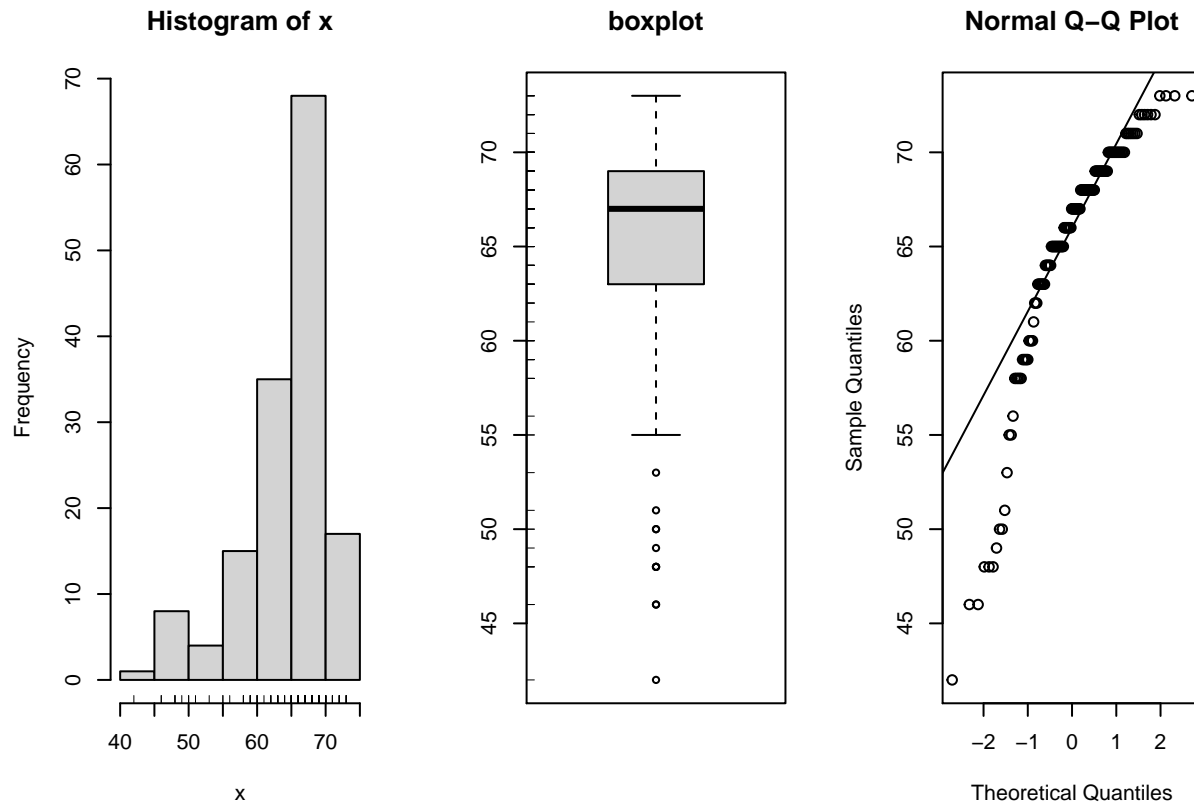
Histograms & Q-Q Plots

SVL

```
morpho_blood_SMI %>%
  ggplot(., aes(x = SVL_mm)) +
  geom_histogram(color = "black", fill="steelblue", bins=50) +
  theme_classic() +
  xlab("SVL (mm)") +
  ylab("Count")
```



```
simple.eda(morpho_blood_SMI$SVL_mm)
```



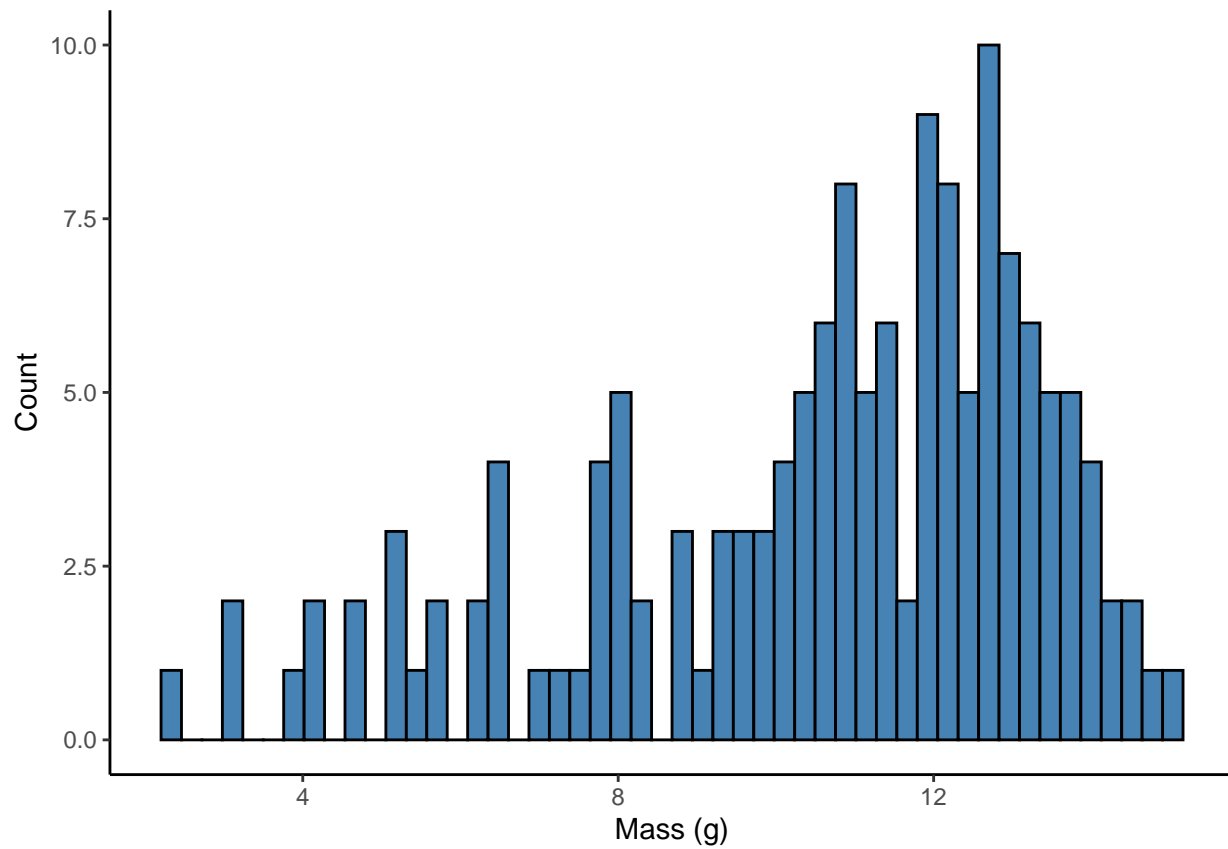
```
# Normality test if p > .05, data is normal. Data is not normal.
```

```
shapiro.test(morpho_blood_SMI$SVL_mm)
```

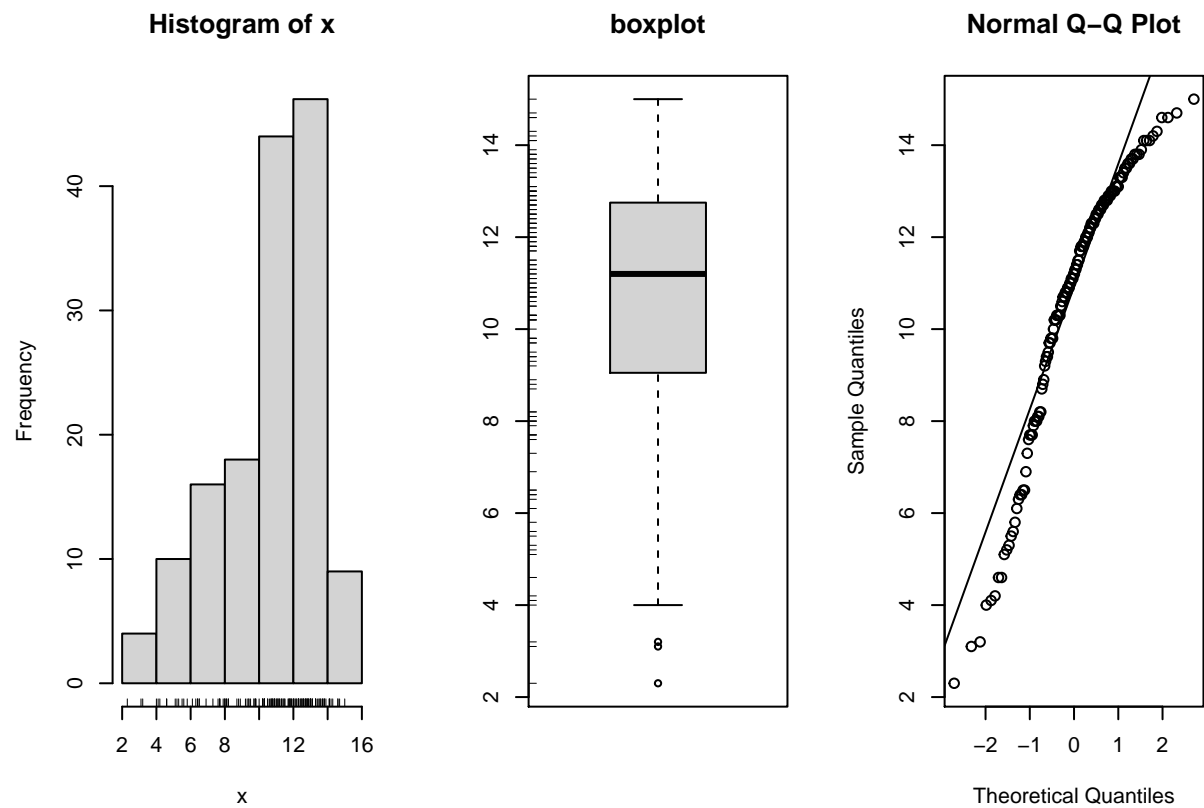
```
##
##  Shapiro-Wilk normality test
##
## data:  morpho_blood_SMI$SVL_mm
## W = 0.85469, p-value = 8.85e-11
```

Mass

```
morpho_blood_SMI %>%
  ggplot(., aes(x = mass_g)) +
  geom_histogram(color = "black", fill="steelblue", bins=50) +
  theme_classic() +
  xlab("Mass (g)") +
  ylab("Count")
```



```
simple.eda(morpho_blood_SMI$mass_g)
```

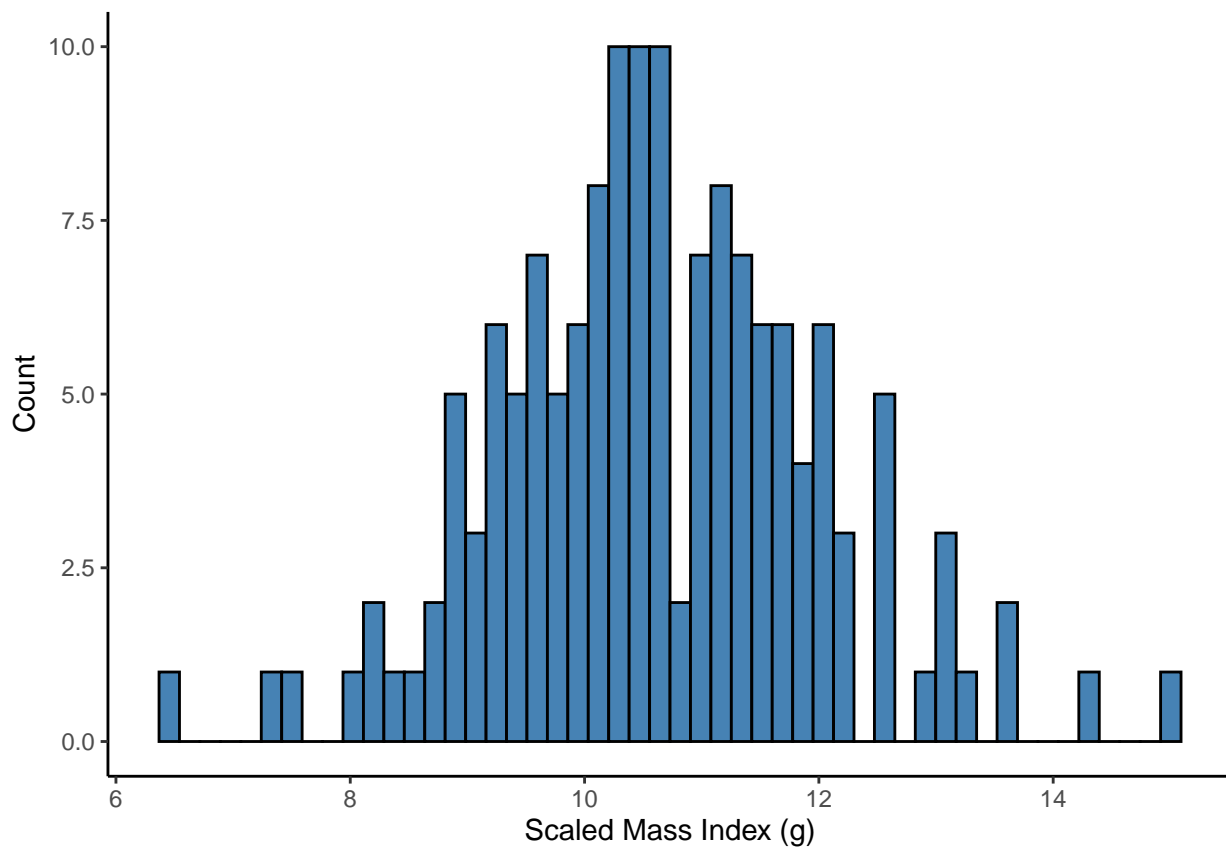


```
# Normality test if p > .05, data is normal. Data is not normal.  
shapiro.test(morpho_blood_SMI$mass_g)
```

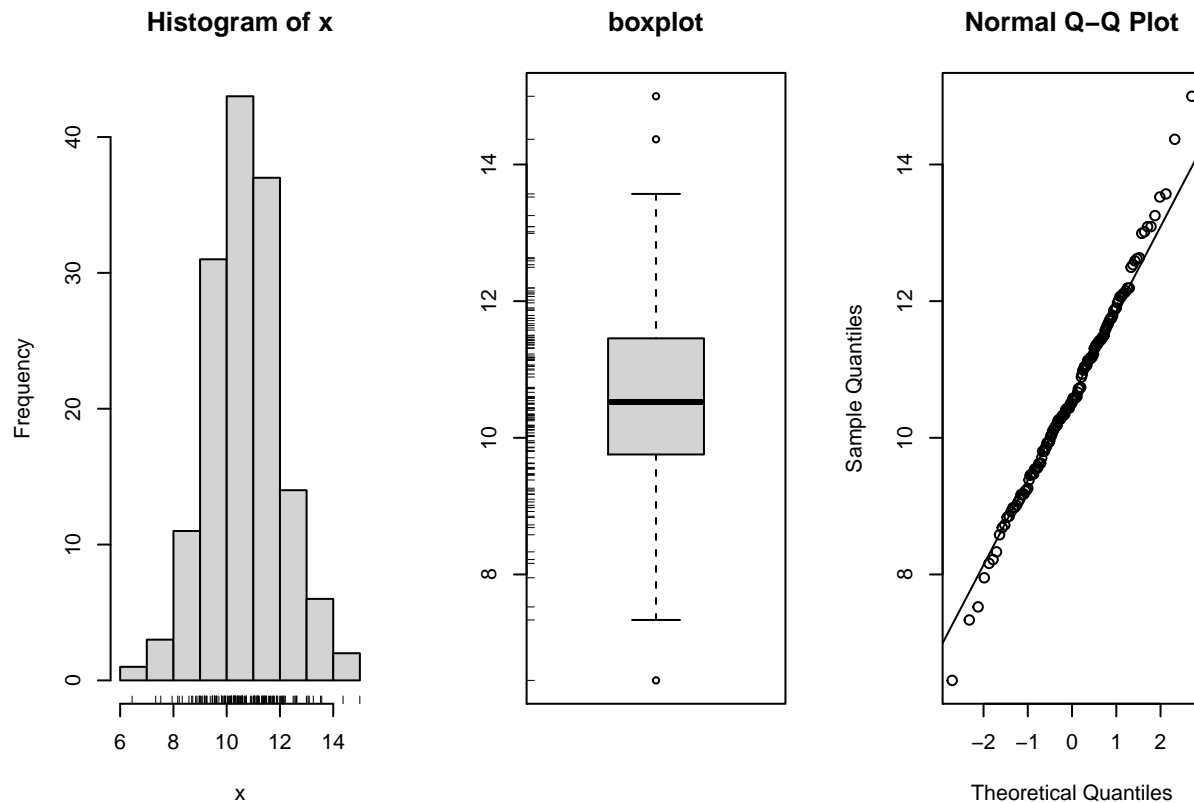
```
##  
## Shapiro-Wilk normality test  
##  
## data: morpho_blood_SMI$mass_g  
## W = 0.92517, p-value = 5.397e-07
```

SMI

```
morpho_blood_SMI %>%  
  ggplot(., aes(x = SMI)) +  
  geom_histogram(color = "black", fill="steelblue", bins=50) +  
  theme_classic() +  
  xlab("Scaled Mass Index (g)") +  
  ylab("Count")
```



```
simple.eda(morpho_blood_SMI$SMI)
```



```
# Normality test if p > .05, data is normal. Data is not normal.
shapiro.test(morpho_blood_SMI$SMI)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  morpho_blood_SMI$SMI
## W = 0.9937, p-value = 0.7667
```

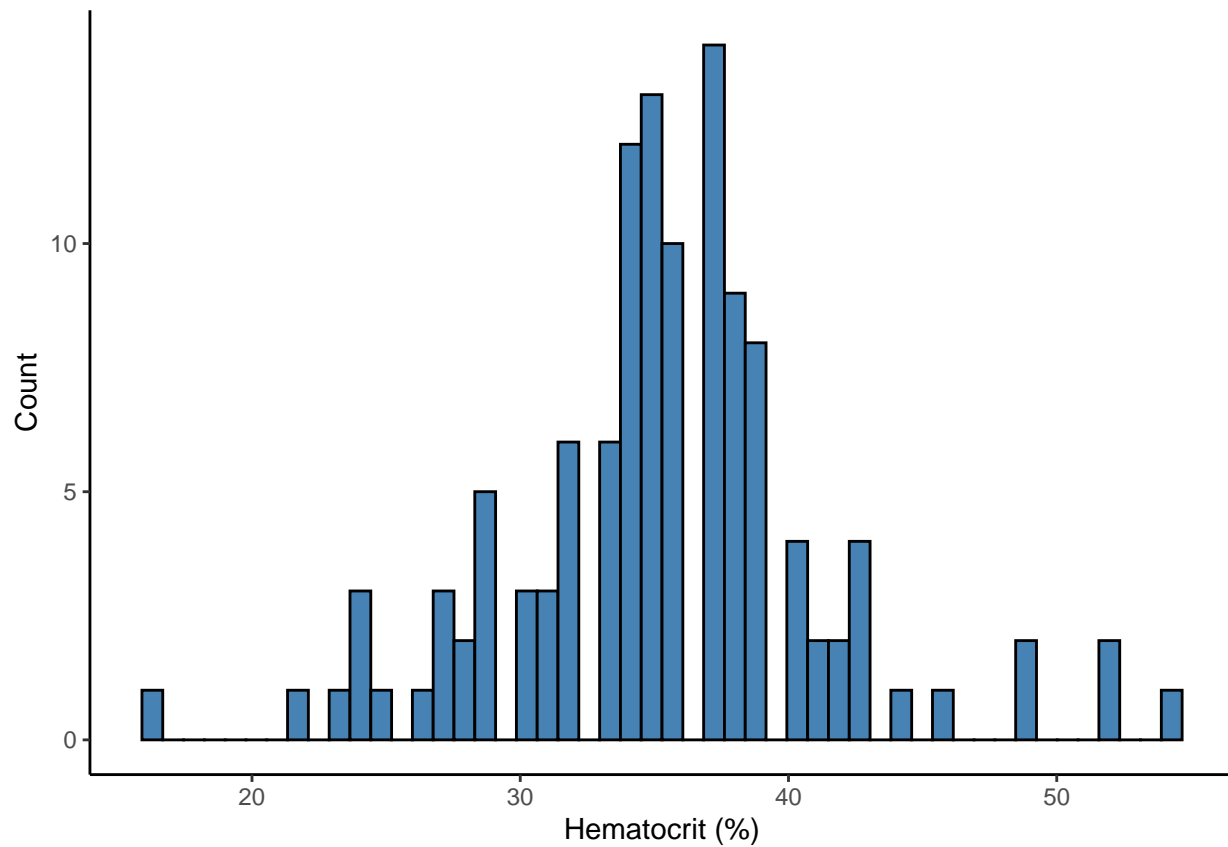
SMI is one of the few variables actually normally distributed!

Hematocrit

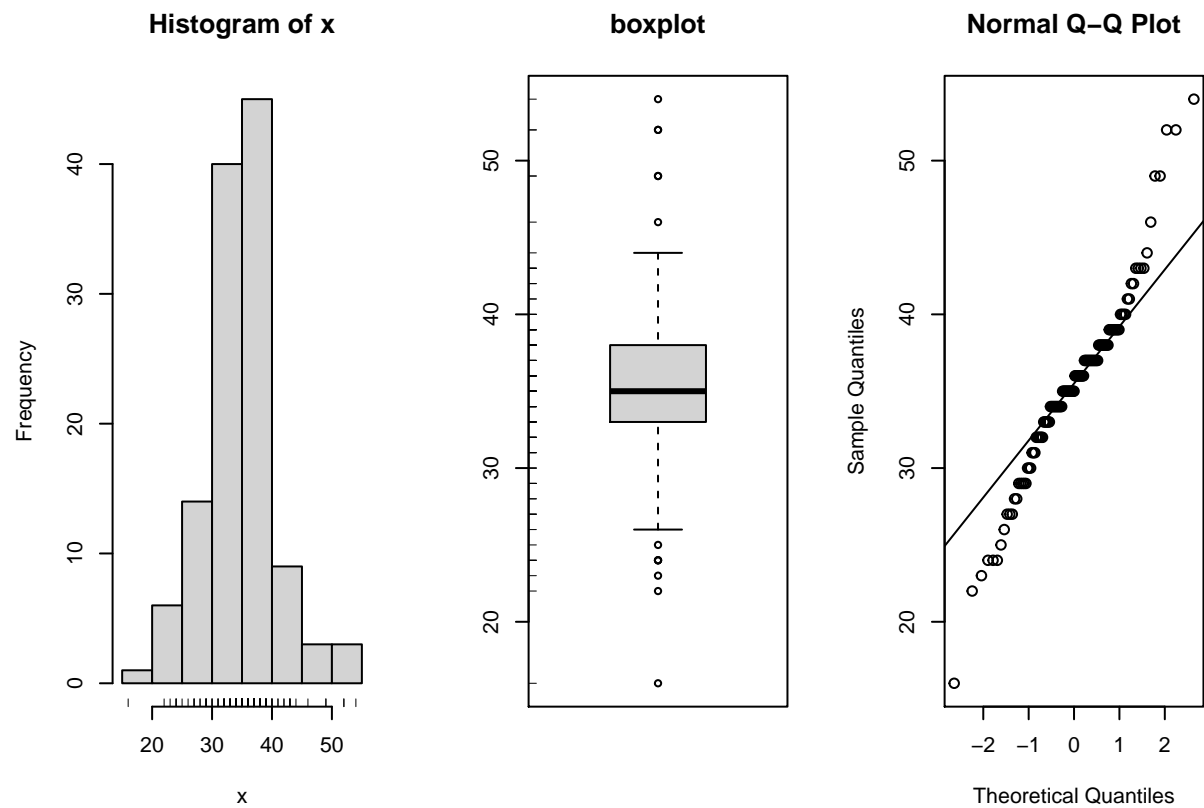
looks pretty normally distributed around ~35%, but not statistically normal

```
morpho_blood_SMI %>%
  ggplot(., aes(x = hematocrit_percent)) +
  geom_histogram(color = "black", fill="steelblue", bins=50) +
  theme_classic() +
  xlab("Hematocrit (%)") +
  ylab("Count")
```

```
## Warning: Removed 27 rows containing non-finite values (stat_bin).
```



```
simple.eda(morpho_blood_SMI$hematocrit_percent)
```




```
# Normality test if p > .05, data is normal. Data is not normal
shapiro.test(morpho_blood_SMI$hematocrit_percent)
```

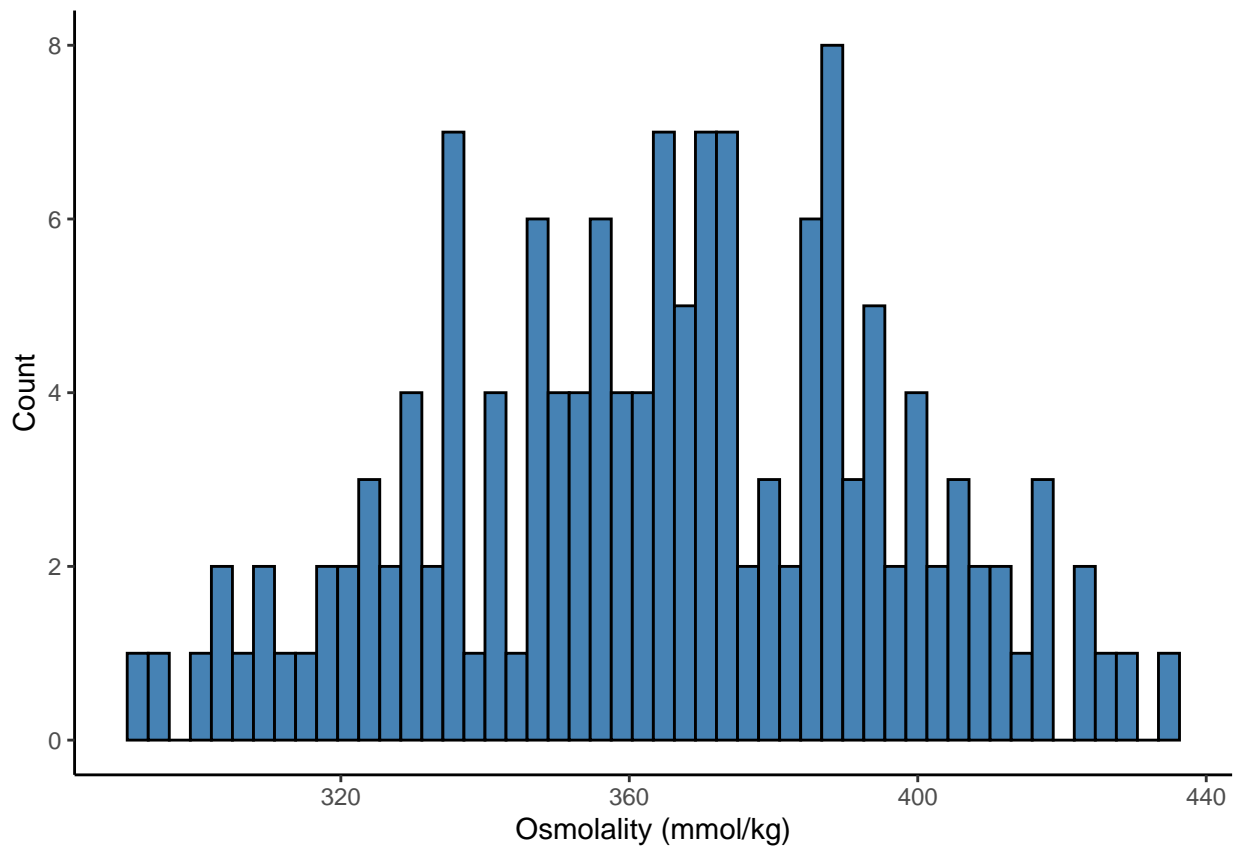
```
##
## Shapiro-Wilk normality test
##
## data: morpho_blood_SMI$hematocrit_percent
## W = 0.95933, p-value = 0.001056
```

Osmolality

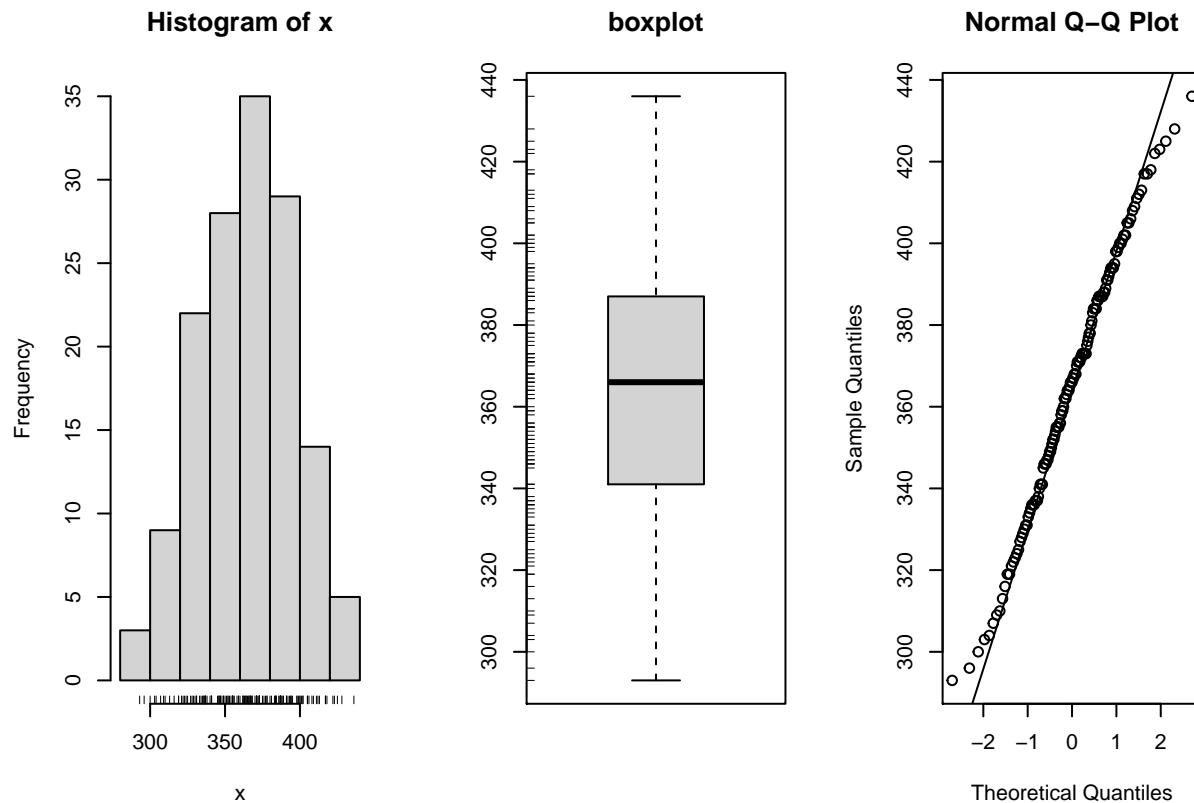
pretty normally distributed around ~370! :D

```
morpho_blood_SMI %>%
  ggplot(., aes(x = osmolality_mmol_kg)) +
  geom_histogram(color = "black", fill="steelblue", bins=50) +
  theme_classic() +
  xlab("Osmolality (mmol/kg)") +
  ylab("Count")
```

```
## Warning: Removed 3 rows containing non-finite values (stat_bin).
```



```
simple.eda(morpho_blood_SMI$osmolality_mmol_kg)
```



```
# Normality test if p > .05, data is normal. Data is normal
shapiro.test(morpho_blood_SMI$osmolality_mmol_kg)
```

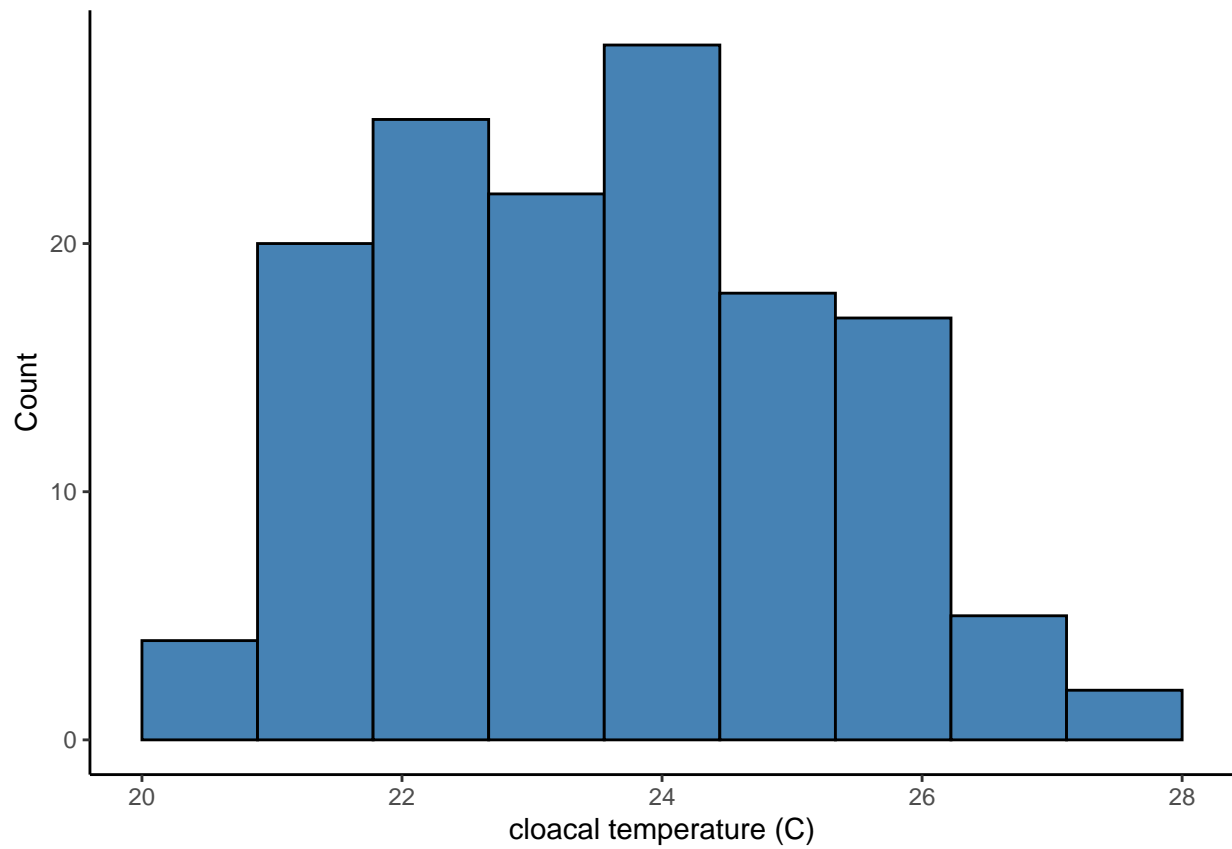
```
##
##  Shapiro-Wilk normality test
##
## data:  morpho_blood_SMI$osmolality_mmol_kg
## W = 0.99198, p-value = 0.5876
```

Cloacal Temperature

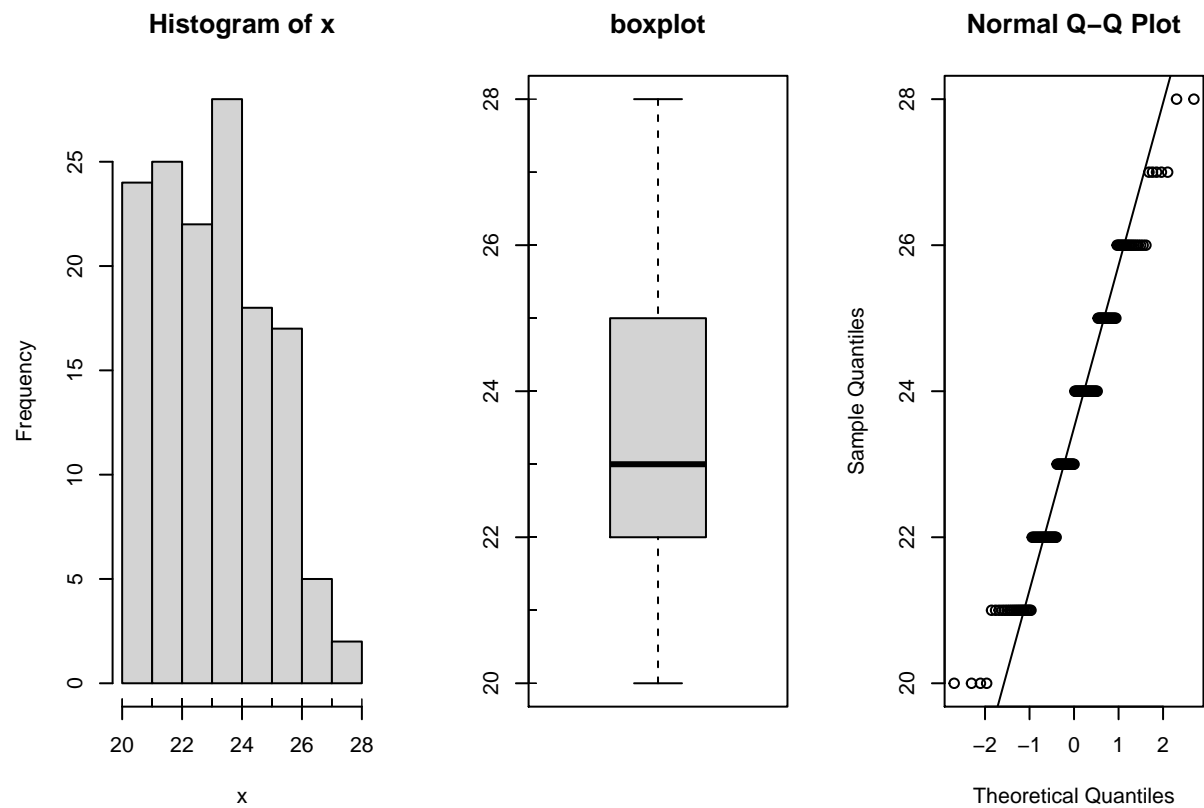
seems normally distributed, but not normal

```
morpho_blood_SMI %>%
  ggplot(., aes(x = cloacal_temp_C)) +
  geom_histogram(color = "black", fill="steelblue", bins=10) +
  theme_classic() +
  xlab("cloacal temperature (C)") +
  ylab("Count")
```

```
## Warning: Removed 7 rows containing non-finite values (stat_bin).
```



```
simple.eda(morpho_blood_SMI$cloacal_temp_C)
```

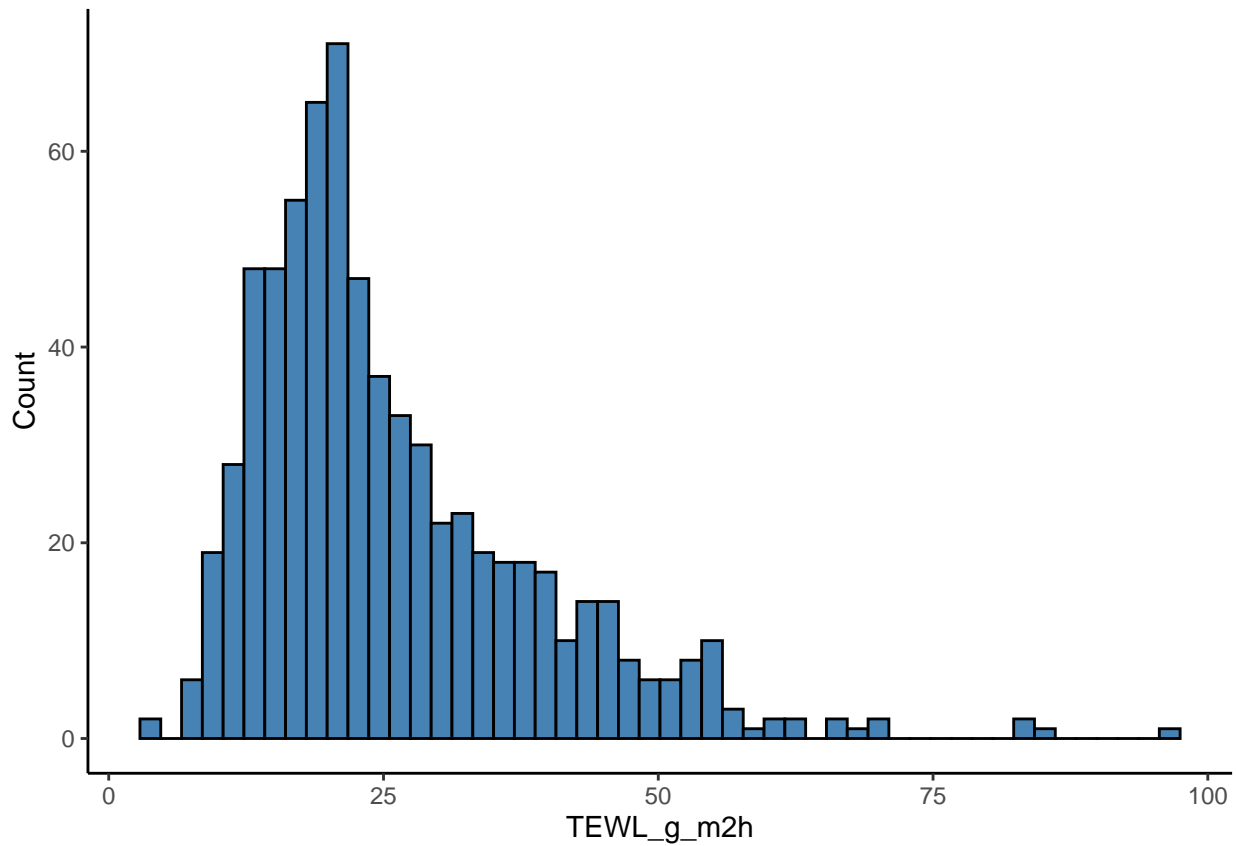


```
# Normality test if p > .05, data is normal. Data is not normal
shapiro.test(morpho_blood_SMI$cloacal_temp_C)
```

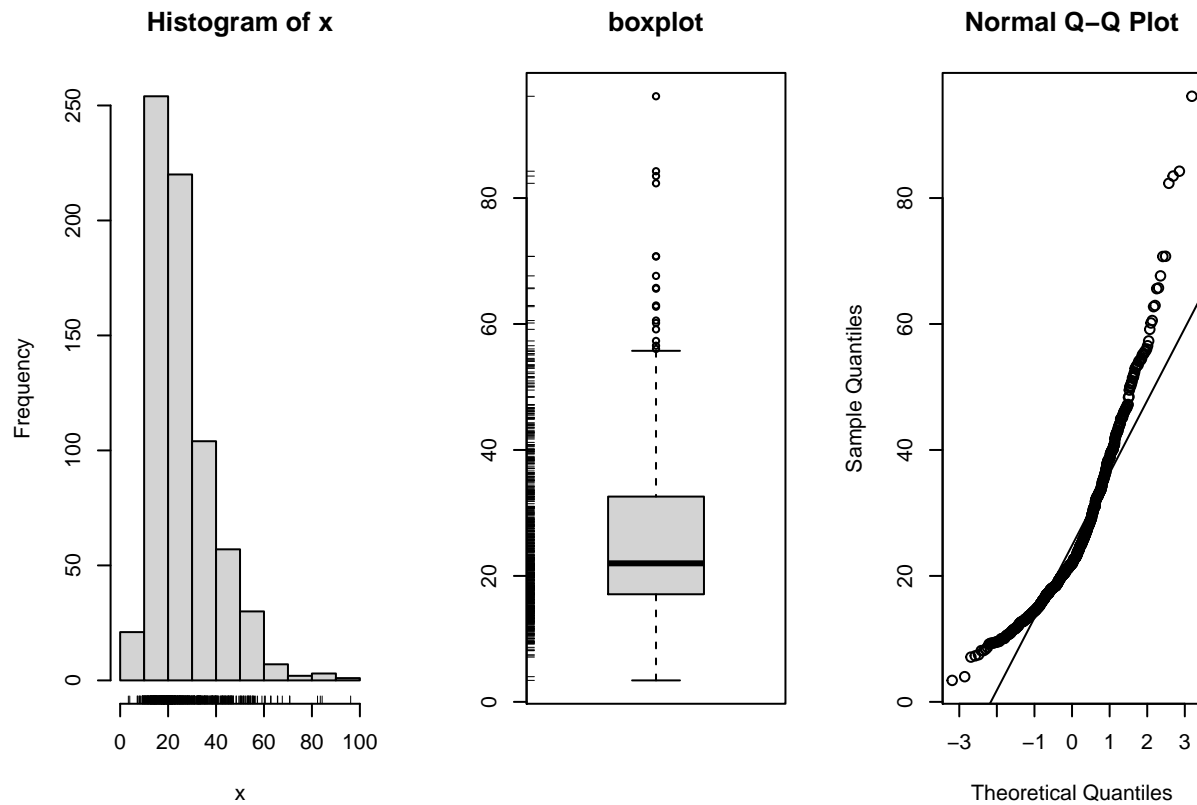
```
##
## Shapiro-Wilk normality test
##
## data: morpho_blood_SMI$cloacal_temp_C
## W = 0.95661, p-value = 0.0002007
```

CEWL

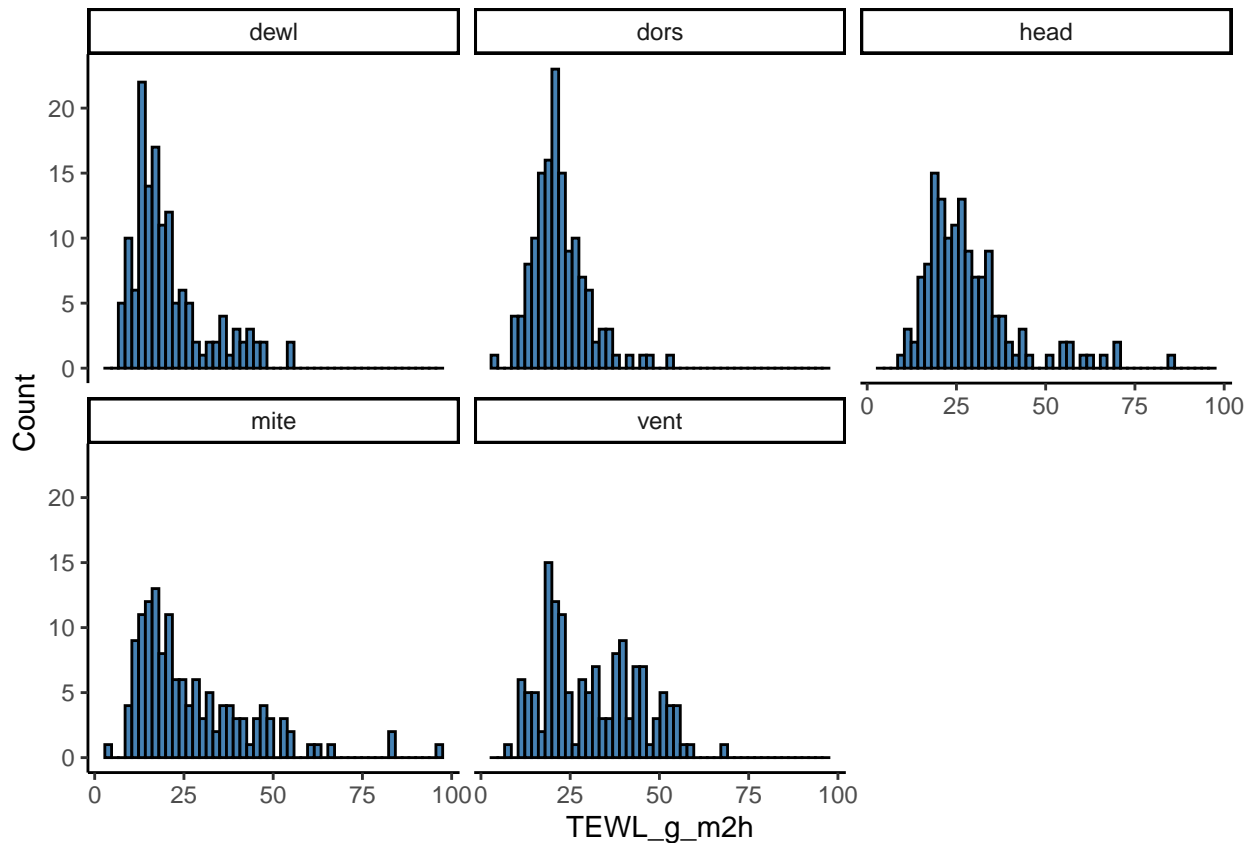
```
CEWL_data_full %>%
  ggplot(., aes(x = TEWL_g_m2h)) +
  geom_histogram(color = "black", fill="steelblue", bins=50) +
  theme_classic() +
  xlab("TEWL_g_m2h") +
  ylab("Count")
```



```
simple.eda(CEWL_data_full$TEWL_g_m2h)
```



```
CEWL_data_full %>%
  ggplot(., aes(x = TEWL_g_m2h)) +
  geom_histogram(color = "black", fill="steelblue", bins=50) +
  theme_classic() +
  xlab("TEWL_g_m2h") +
  ylab("Count") +
  facet_wrap(~region)
```



```
# Normality test if p > .05, data is normal. Data is not normal
shapiro.test(CEWL_data_full$TEWL_g_m2h)
```

```
##
## Shapiro-Wilk normality test
##
## data: CEWL_data_full$TEWL_g_m2h
## W = 0.89548, p-value < 2.2e-16
```

```
# Log transformation
shapiro.test(log(CEWL_data_full$TEWL_g_m2h))
```

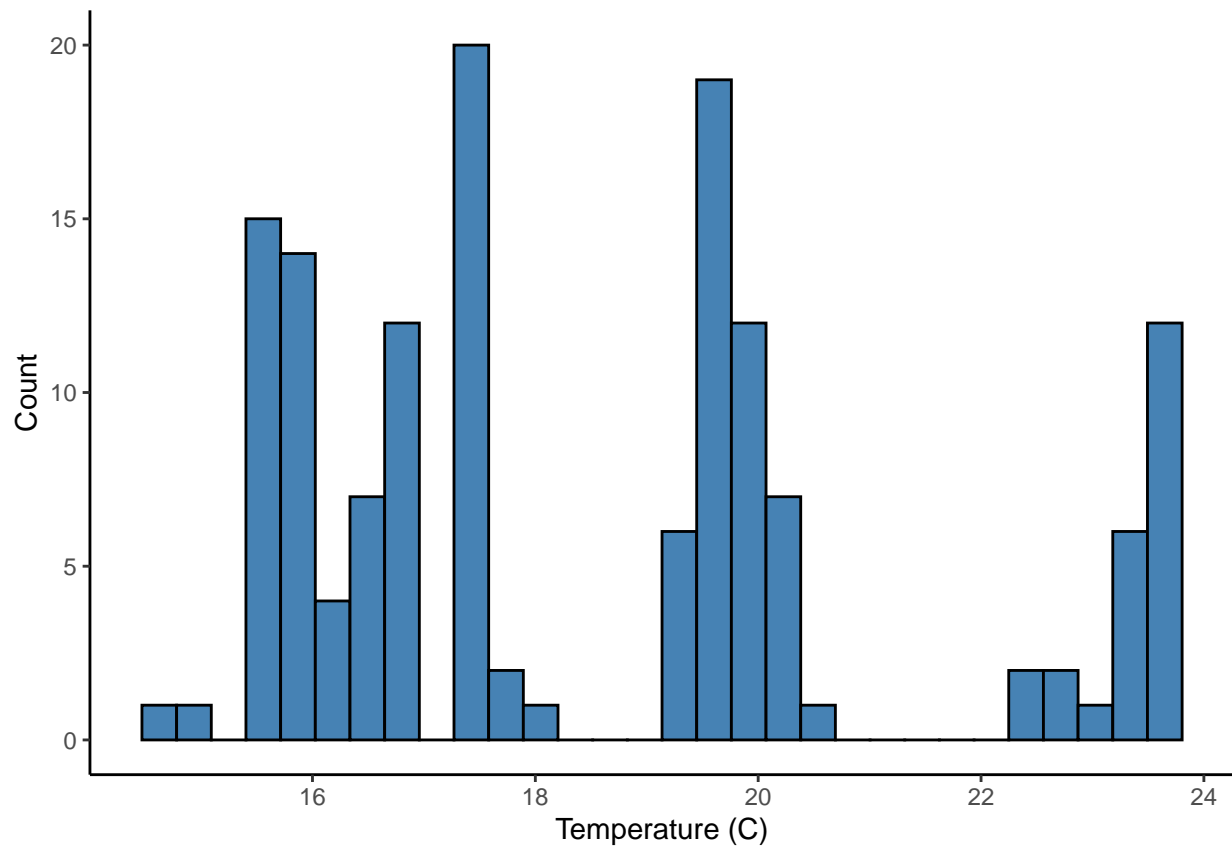
```
##
## Shapiro-Wilk normality test
##
## data: log(CEWL_data_full$TEWL_g_m2h)
## W = 0.99368, p-value = 0.004976
```

```
# p-value improves to 0.00548, but is still significant
```

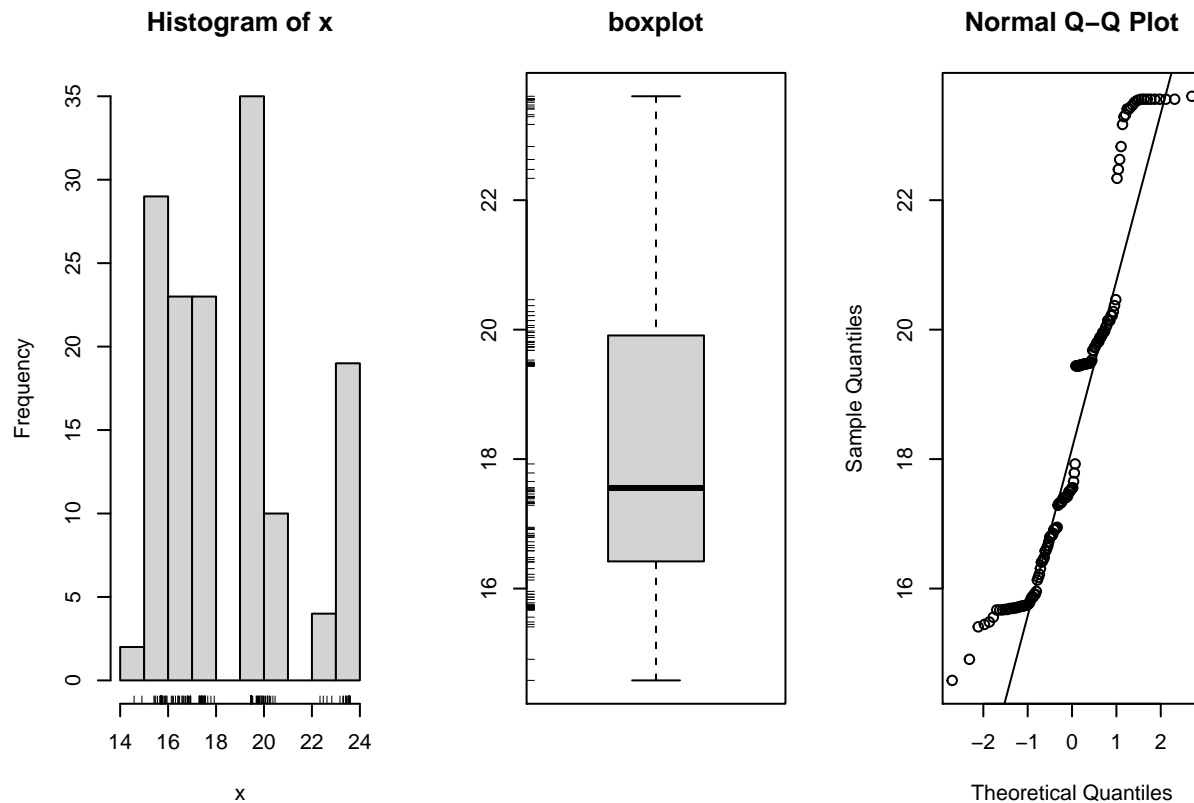
Capture Temperature

```
morpho_blood_SMI %>%
  ggplot(., aes(x = temp_C_interpol)) +
  geom_histogram(color = "black", fill="steelblue", bins=30) +
  theme_classic() +
  xlab("Temperature (C)") +
  ylab("Count")
```

```
## Warning: Removed 3 rows containing non-finite values (stat_bin).
```



```
simple.eda(morpho_blood_SMI$temp_C_interpol)
```



```
# Normality test if  $p > .05$ , data is normal. Data is not normal
shapiro.test(morpho_blood_SMI$temp_C_interpol)
```

```
##
## Shapiro-Wilk normality test
##
## data: morpho_blood_SMI$temp_C_interpol
## W = 0.89272, p-value = 8.162e-09
```

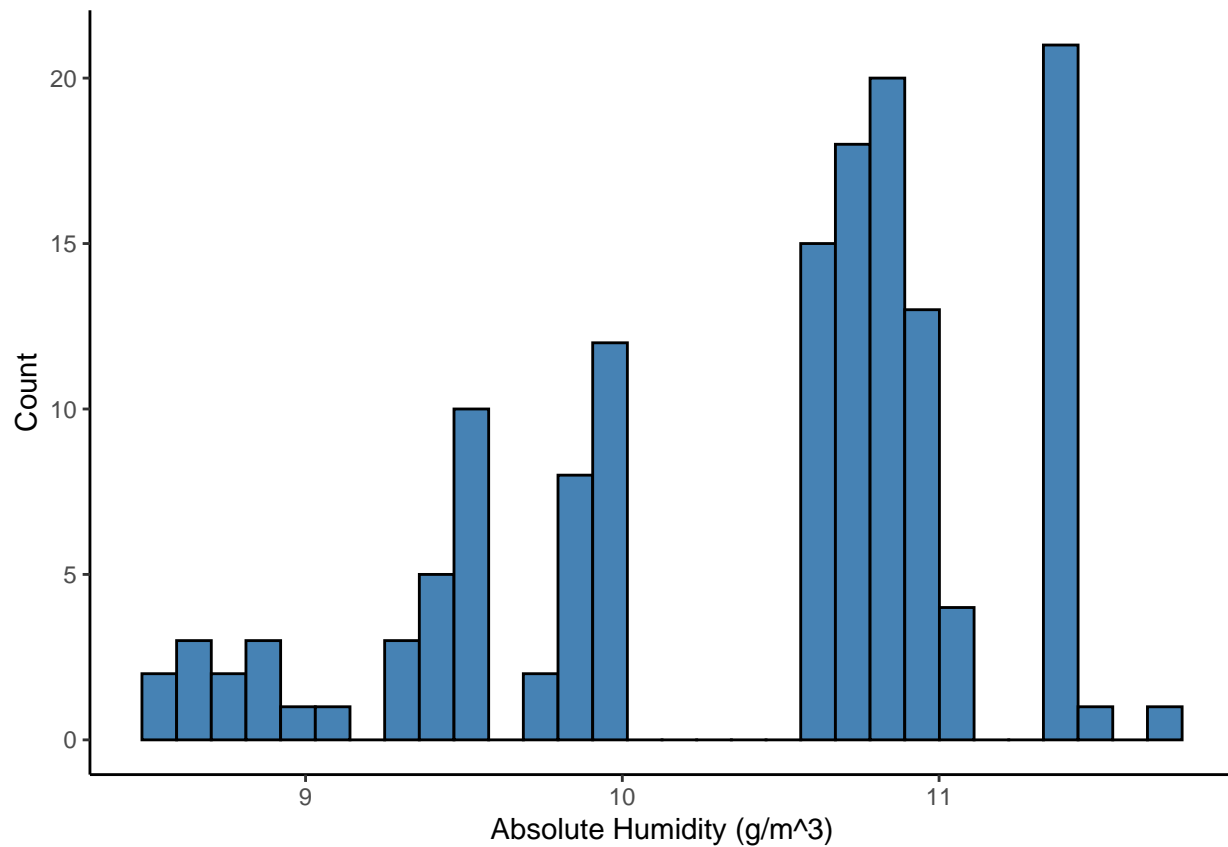
```
# log transform
shapiro.test(log(morpho_blood_SMI$temp_C_interpol))
```

```
##
## Shapiro-Wilk normality test
##
## data: log(morpho_blood_SMI$temp_C_interpol)
## W = 0.91155, p-value = 9.439e-08
```

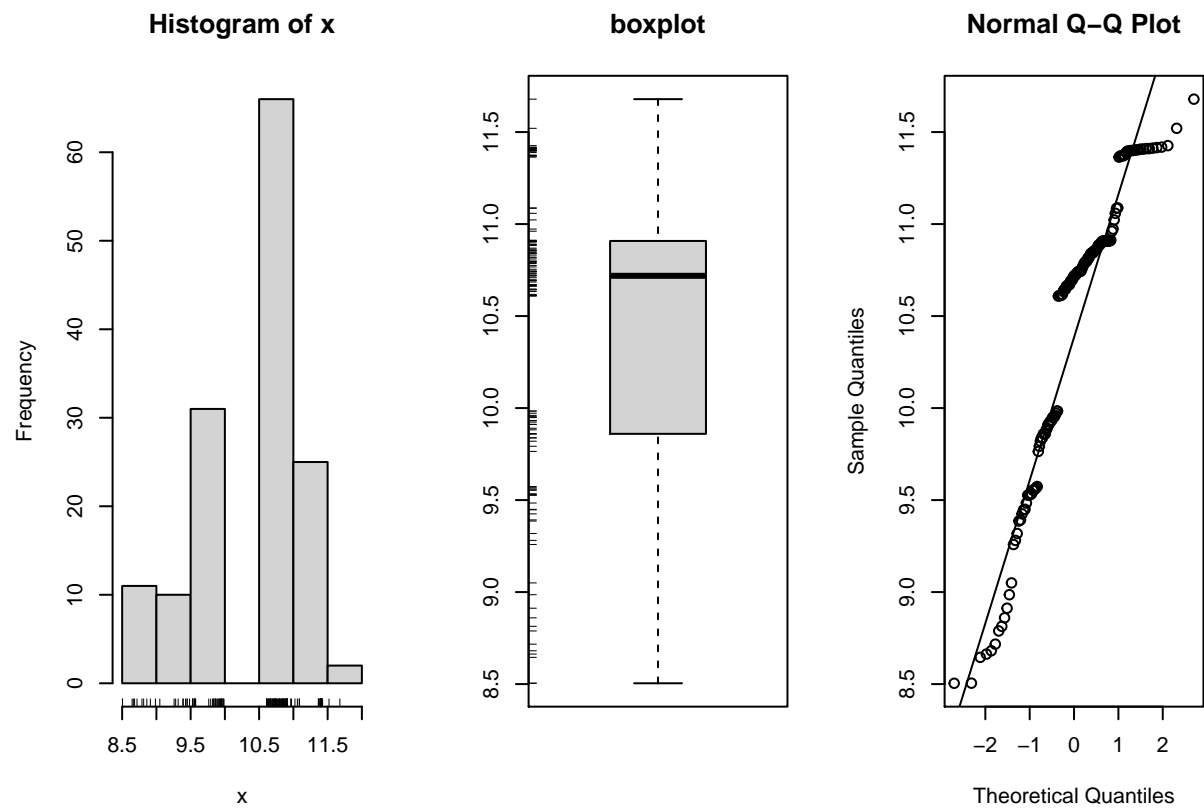
Capture Absolute Humidity

```
morpho_blood_SMI %>%
  ggplot(., aes(x = abs_humidity_g_m3_interpol)) +
  geom_histogram(color = "black", fill="steelblue", bins=30) +
  theme_classic() +
  xlab("Absolute Humidity (g/m^3)") +
  ylab("Count")
```

```
## Warning: Removed 3 rows containing non-finite values (stat_bin).
```

```
simple.eda(morpho_blood_SMI$abs_humidity_g_m3_interpol)
```



```
# Normality test if p > .05, data is normal. Data is not normal  
shapiro.test(morpho_blood_SMI$abs_humidity_g_m3_interpol)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: morpho_blood_SMI$abs_humidity_g_m3_interpol  
## W = 0.90867, p-value = 6.365e-08
```

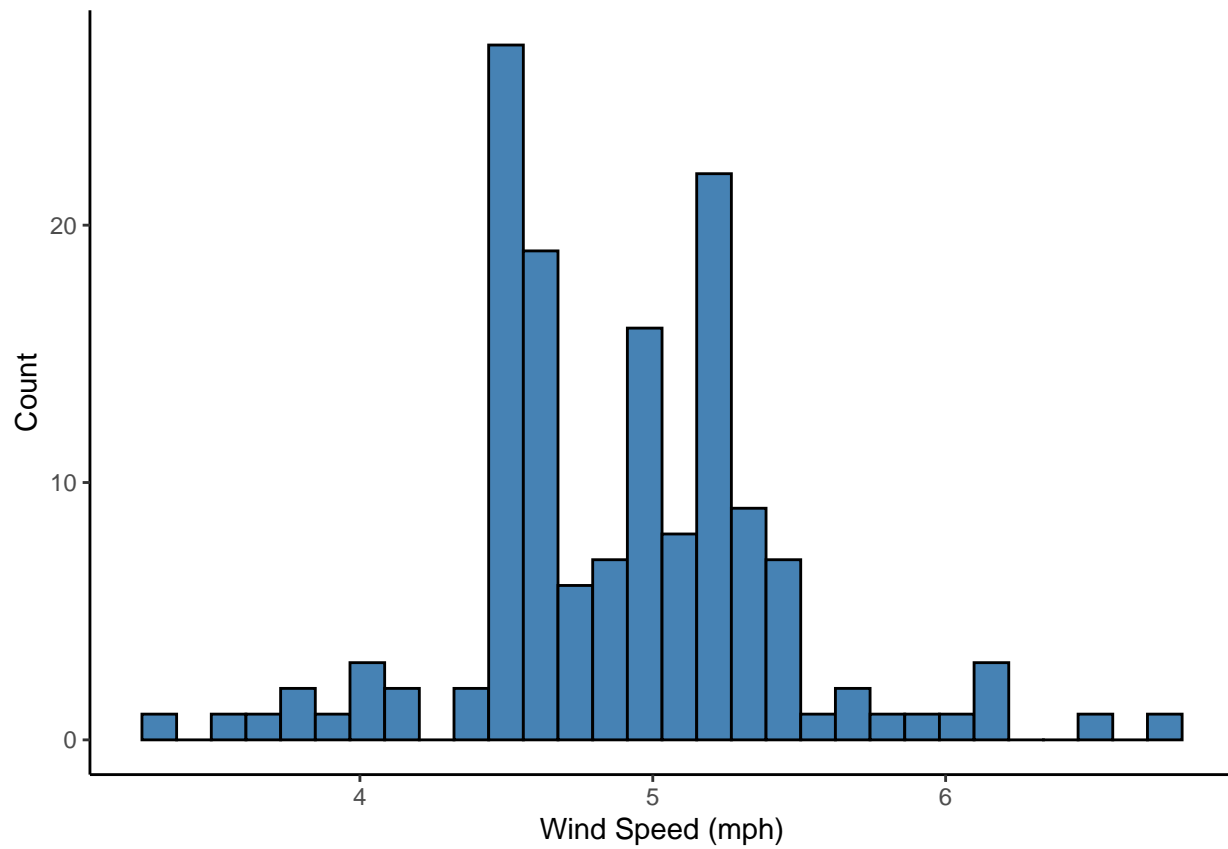
```
# Doesn't fix non normality  
shapiro.test(log(morpho_blood_SMI$abs_humidity_g_m3_interpol))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: log(morpho_blood_SMI$abs_humidity_g_m3_interpol)  
## W = 0.89617, p-value = 1.252e-08
```

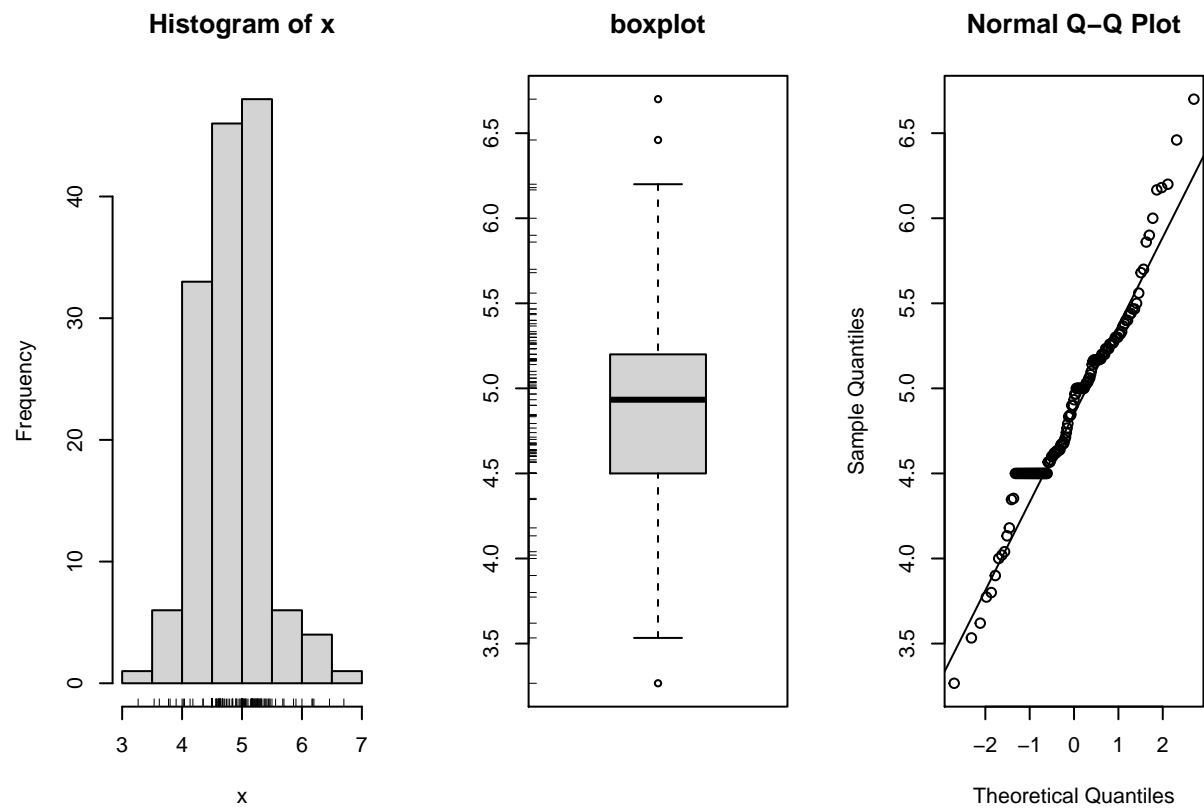
Wind Speed at Capture

```
morpho_blood_SMI %>%  
  ggplot(., aes(x = Wind_mph_interpol)) +  
  geom_histogram(color = "black", fill="steelblue", bins=30) +  
  theme_classic() +  
  xlab("Wind Speed (mph)") +  
  ylab("Count")
```

```
## Warning: Removed 3 rows containing non-finite values (stat_bin).
```



```
simple.eda(morpho_blood_SMI$Wind_mph_interpol)
```



```
# Normality test if p > .05, data is normal. Data is not normal  
shapiro.test(morpho_blood_SMI$Wind_mph_interpol)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: morpho_blood_SMI$Wind_mph_interpol  
## W = 0.96149, p-value = 0.0004364
```

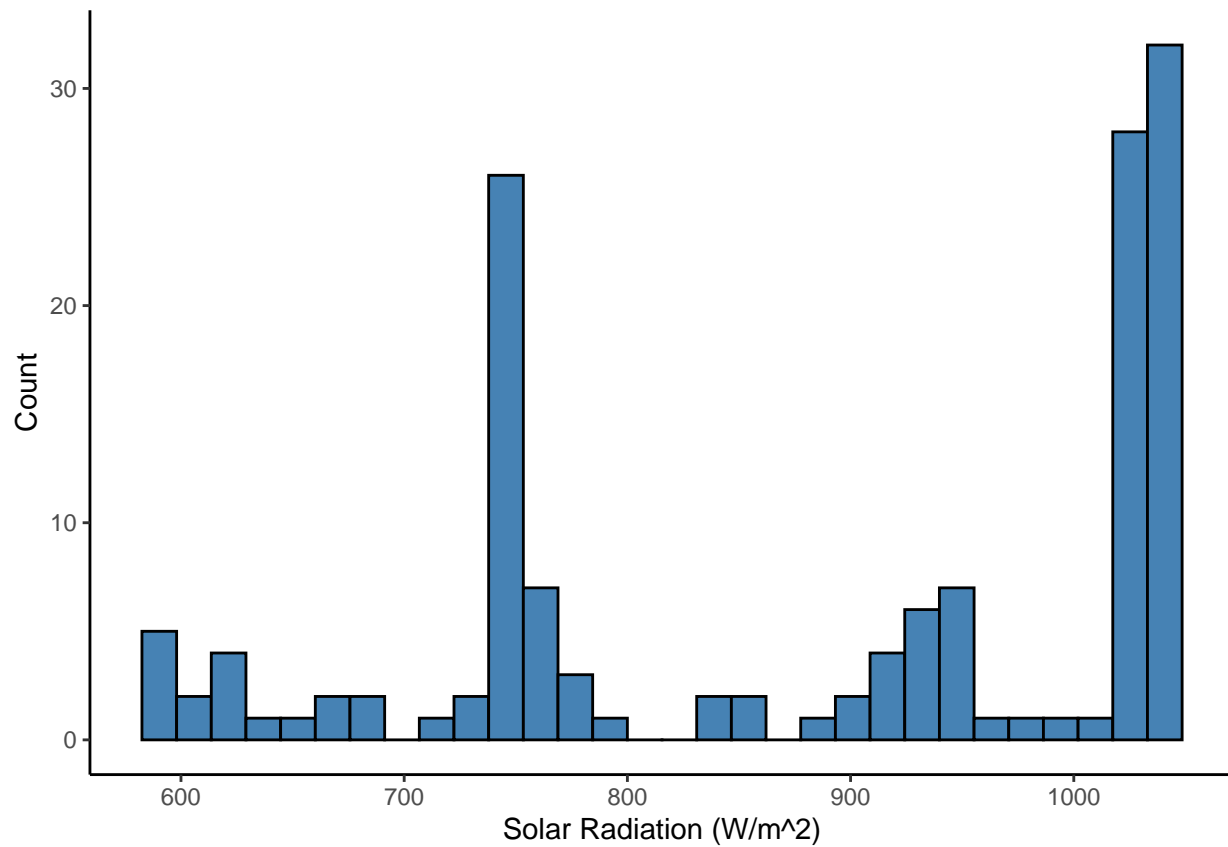
```
# ln transformation doesn't fix non normality  
shapiro.test(log(morpho_blood_SMI$Wind_mph_interpol))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: log(morpho_blood_SMI$Wind_mph_interpol)  
## W = 0.95827, p-value = 0.0002241
```

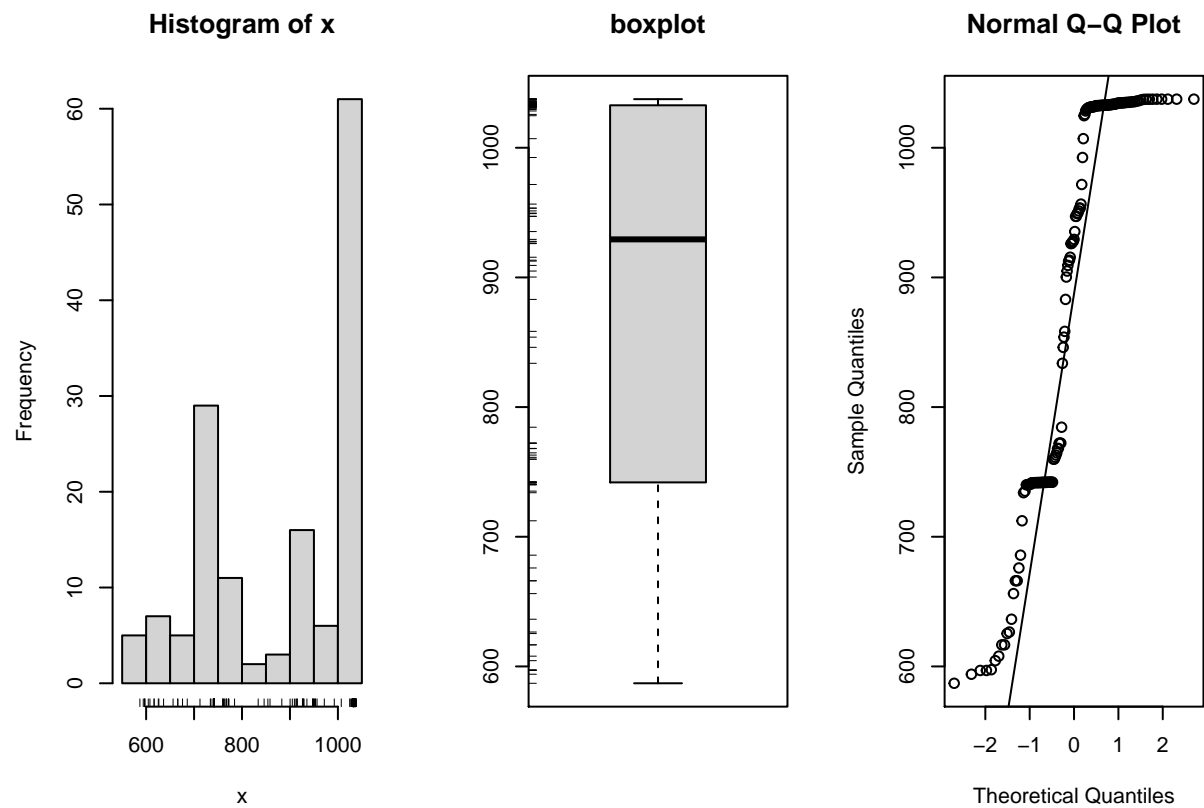
Solar Radiation at Capture

```
morpho_blood_SMI %>%  
  ggplot(., aes(x = Solar_rad_Wm2_interpol)) +  
  geom_histogram(color = "black", fill="steelblue", bins=30) +  
  theme_classic() +  
  xlab("Solar Radiation (W/m^2)") +  
  ylab("Count")
```

```
## Warning: Removed 3 rows containing non-finite values (stat_bin).
```



```
simple.eda(morpho_blood_SMI$Solar_rad_Wm2_interpol)
```



```
# Normality test if p > .05, data is normal. Data is not normal
shapiro.test(morpho_blood_SMI$Solar_rad_Wm2_interpol)
```

```
##
## Shapiro-Wilk normality test
##
## data: morpho_blood_SMI$Solar_rad_Wm2_interpol
## W = 0.82946, p-value = 1.072e-11
```

```
# Doesn't fix non normality
shapiro.test(log(morpho_blood_SMI$Solar_rad_Wm2_interpol))
```

```
##
## Shapiro-Wilk normality test
##
## data: log(morpho_blood_SMI$Solar_rad_Wm2_interpol)
## W = 0.829, p-value = 1.028e-11
```

Conclusion

Osmolality and SMI were the only normally distributed variables.

The following variables each had non-normal distributions: - SVL (skewed left) - mass (skewed left) - hct (looks like a bell curve, but not very pretty) - cloacal temp (skewed right) - CEWL (skewed right, overall and for each region individually) - capture temp (multimodal) - capture humidity (multimodal and skewed left) - wind speed (multimodal but sorta bell curve) - solar radiation (multimodal)

All of the non-normal variables are likely to need transformation if they're included in the final model.

Basic Figs & LMs

What affects osmolality & hematocrit?

Hct ~ SVL

```
# plot
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = SVL_mm,
                 y = hematocrit_percent,
                 ),
            size = 1,
            alpha = 0.6) +
  stat_smooth(aes(x = SVL_mm,
                 y = hematocrit_percent,
                 ),
            formula = y ~ x,
            method = "lm",
            color = "gray",
            se = F,
            size = 1.6,
            alpha = 1 ) +
  theme_classic() +
  xlab("SVL") +
  ylab("Hct") +
```

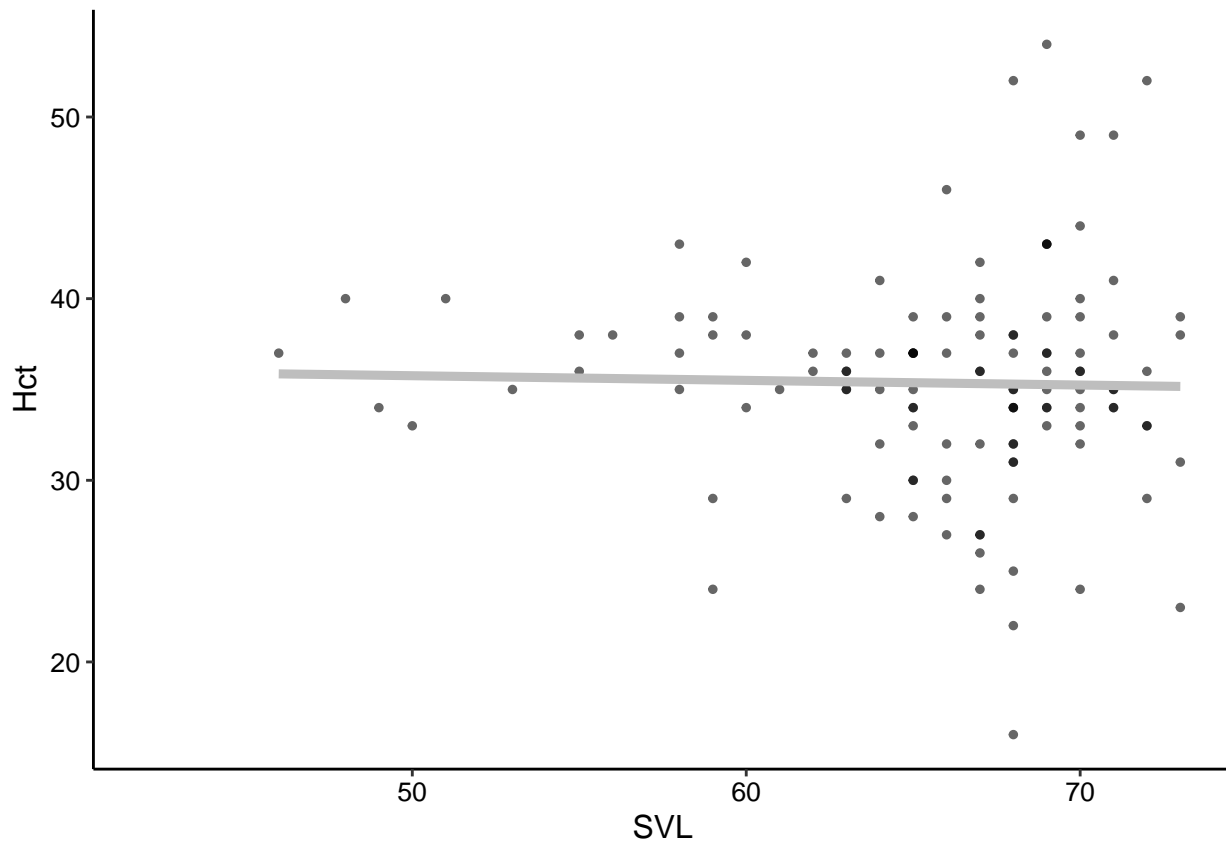
```

theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
      axis.text = element_text(color = "black",
                                family = "sans",
                                size = 10),
      legend.text.align = 0,
)

```

```
## Warning: Removed 27 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 27 rows containing missing values (geom_point).
```



```

# lm
lm_hct_SVL <- lm(hematocrit_percent ~ SVL_mm,
                  data = morpho_blood_SMI)
summary(lm_hct_SVL)

```

```

##
## Call:
## lm(formula = hematocrit_percent ~ SVL_mm, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.2957  -2.7565  -0.2189   2.7043  18.7299
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)

```

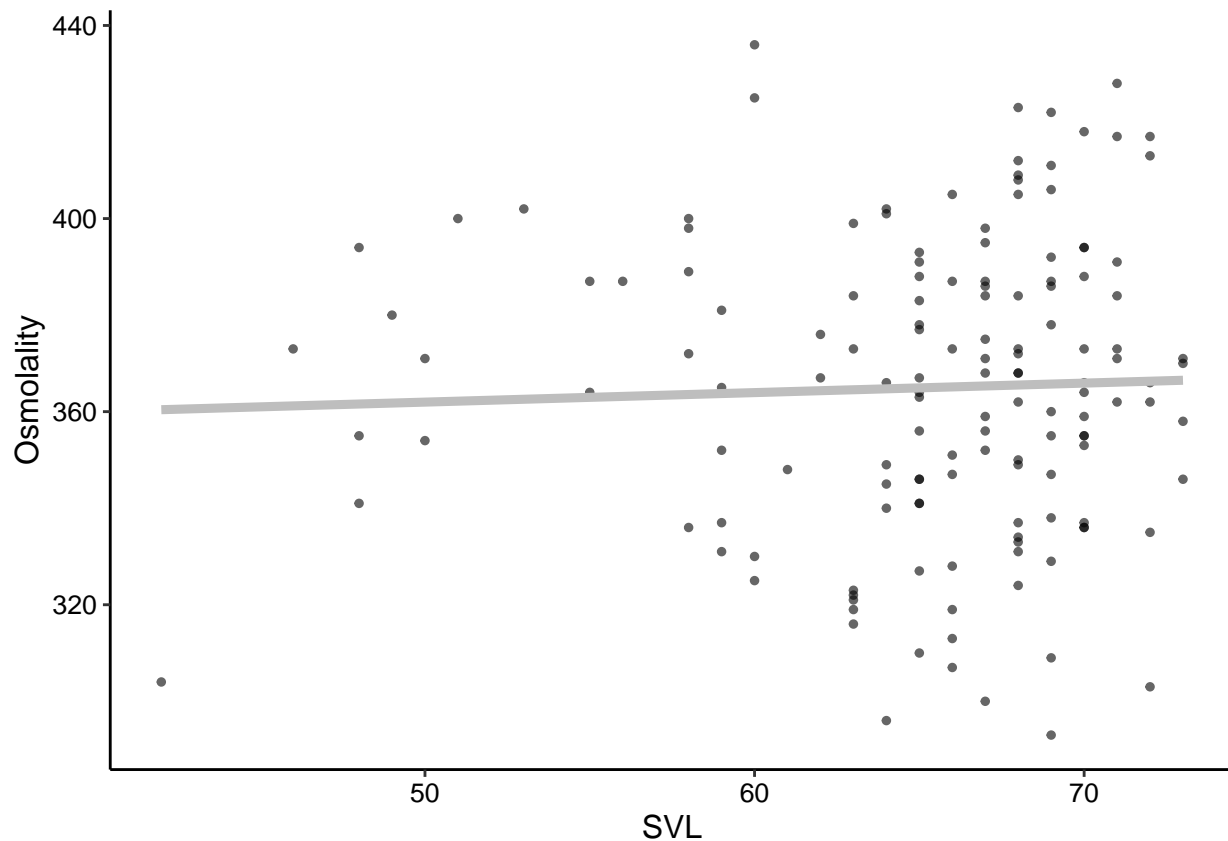
```
## (Intercept) 37.03634    6.51586    5.684 9.54e-08 ***
## SVL_mm      -0.02560    0.09888   -0.259    0.796
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.953 on 119 degrees of freedom
## (27 observations deleted due to missingness)
## Multiple R-squared:  0.0005628, Adjusted R-squared:  -0.007836
## F-statistic: 0.06702 on 1 and 119 DF, p-value: 0.7962
```

Osml ~ SVL

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = SVL_mm,
                 y = osmolality_mmol_kg,
                 ),
             size = 1,
             alpha = 0.6) +
  stat_smooth(aes(x = SVL_mm,
                  y = osmolality_mmol_kg,
                  ),
              formula = y ~ x,
              method = "lm",
              color = "gray",
              se = F,
              size = 1.6,
              alpha = 1 ) +
  theme_classic() +
  xlab("SVL") +
  ylab("Osmolality") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

```
# lm
lm_osml_SVL <- lm(osmolality_mmol_kg ~ SVL_mm,
                  data = morpho_blood_SMI)
summary(lm_osml_SVL)
```

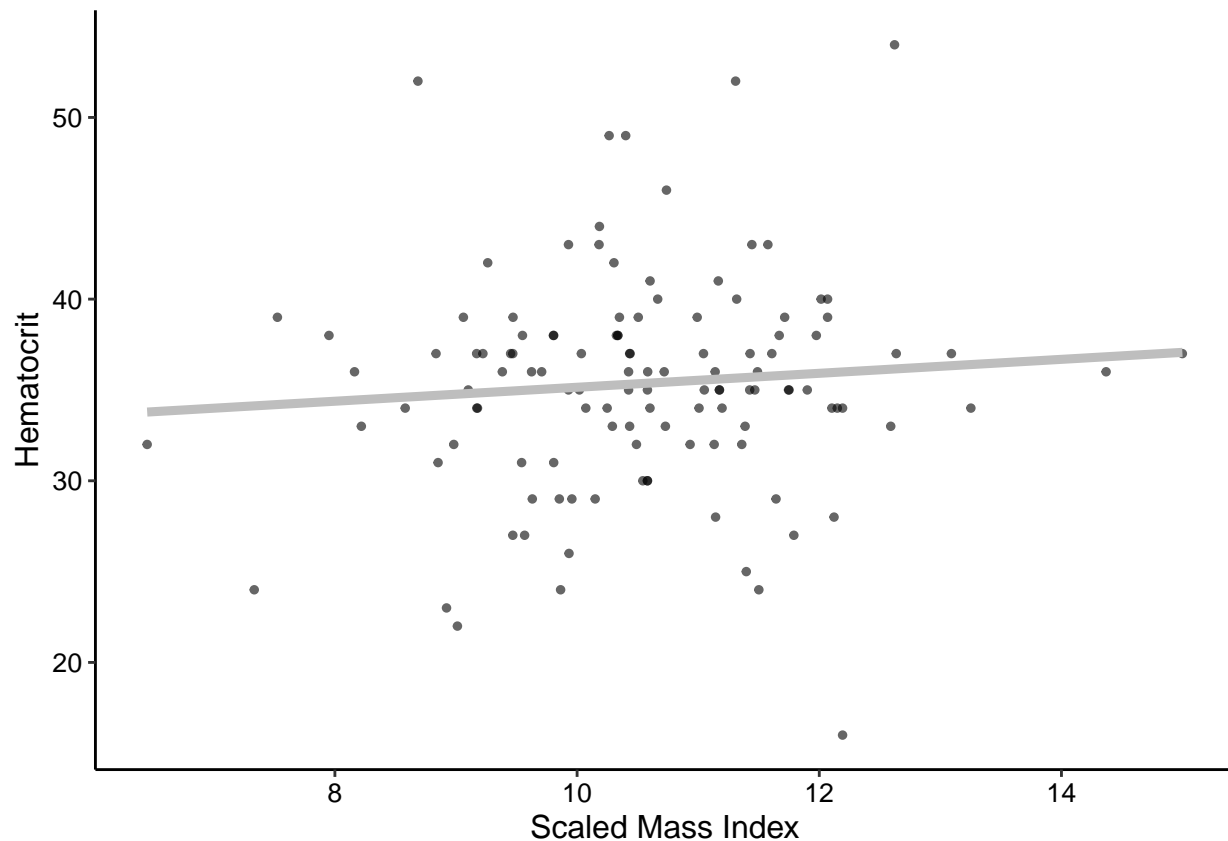
```
##
## Call:
## lm(formula = osmolality_mmol_kg ~ SVL_mm, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -72.716 -20.584   1.268  22.088  72.055
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  352.1404    28.1557  12.507  <2e-16 ***
## SVL_mm        0.1967     0.4305   0.457   0.648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.27 on 143 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.001458,    Adjusted R-squared:  -0.005525
## F-statistic: 0.2088 on 1 and 143 DF,  p-value: 0.6484
```

Hct ~ SMI

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = SMI,
                 y = hematocrit_percent,
                 ),
            size = 1,
            alpha = 0.6) +
  stat_smooth(aes(x = SMI,
                  y = hematocrit_percent,
                  ),
             formula = y ~ x,
             method = "lm",
             color = "gray",
             se = F,
             size = 1.6,
             alpha = 1 ) +
  theme_classic() +
  xlab("Scaled Mass Index") +
  ylab("Hematocrit") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```

```
## Warning: Removed 27 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 27 rows containing missing values (geom_point).
```



```
# lm
lm_hct_mass <- lm(hematocrit_percent ~ SMI,
                  data = morpho_blood_SMI)
summary(lm_hct_mass)

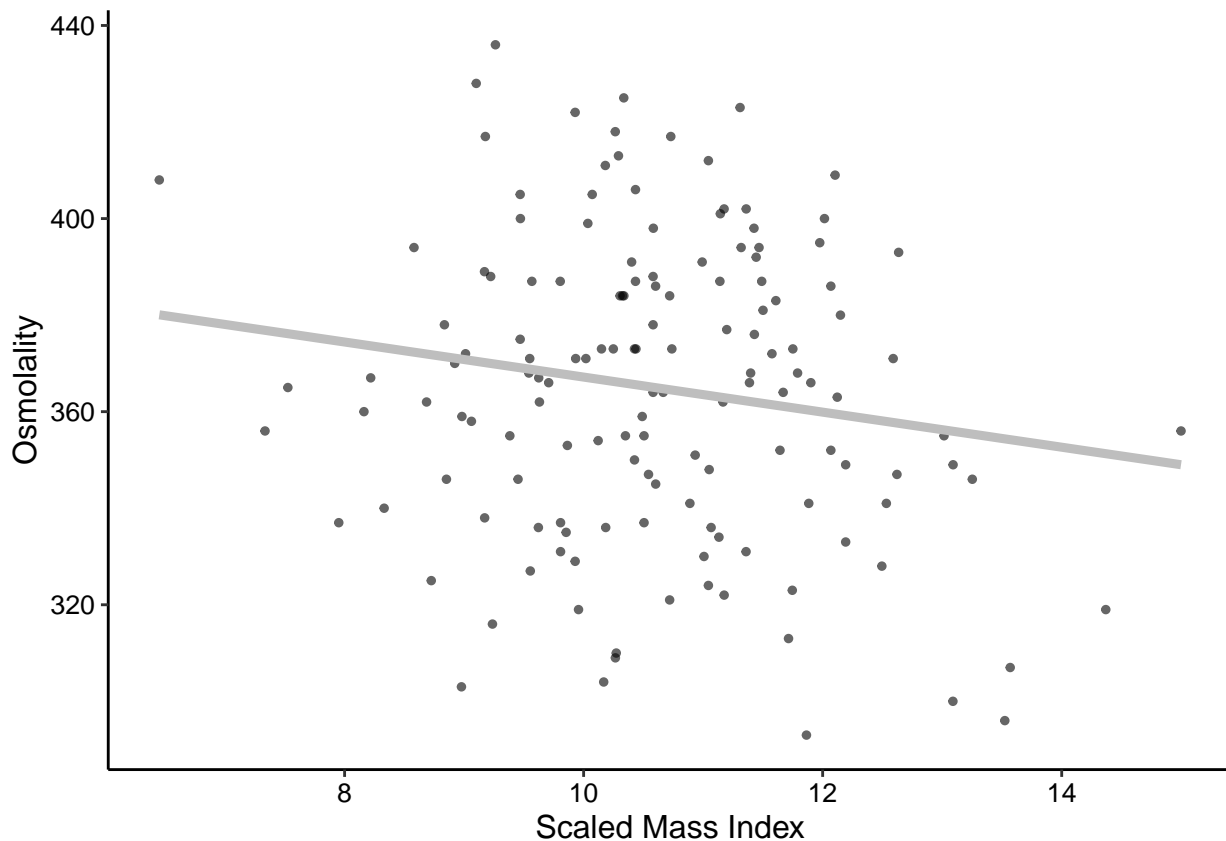
##
## Call:
## lm(formula = hematocrit_percent ~ SMI, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.990  -2.680  -0.120   2.927  17.845
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  31.3051     4.3249   7.238 4.84e-11 ***
## SMI           0.3842     0.4071   0.944   0.347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.933 on 119 degrees of freedom
## (27 observations deleted due to missingness)
## Multiple R-squared:  0.007431, Adjusted R-squared: -0.0009099
## F-statistic: 0.8909 on 1 and 119 DF, p-value: 0.3471
```

Osml ~ SMI

```
morpho_blood_SMI %>%  
  ggplot(data = .) +  
  geom_point(aes(x = SMI,  
                 y = osmolality_mmol_kg,  
                 ),  
            size = 1,  
            alpha = 0.6) +  
  stat_smooth(aes(x = SMI,  
                  y = osmolality_mmol_kg),  
              formula = y ~ x,  
              method = "lm",  
              se = F,  
              color = "gray",  
              size = 1.6,  
              alpha = 1 ) +  
  theme_classic() +  
  xlab("Scaled Mass Index") +  
  ylab("Osmolality") +  
  theme(text = element_text(color = "black",  
                             family = "sans",  
                             size = 12),  
        axis.text = element_text(color = "black",  
                                   family = "sans",  
                                   size = 10),  
        legend.text.align = 0,  
  )
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



```
# lm
lm_osml_mass <- lm(osmolality_mmol_kg ~ SMI,
                   data = morpho_blood_SMI)
summary(lm_osml_mass)

##
## Call:
## lm(formula = osmolality_mmol_kg ~ SMI, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -67.879  -22.935   1.245   21.677   66.154
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  403.490     20.121  20.053  <2e-16 ***
## SMI          -3.632       1.881  -1.931  0.0555 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.89 on 143 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.02541,    Adjusted R-squared:  0.0186
## F-statistic: 3.729 on 1 and 143 DF,  p-value: 0.05545
```

Hct ~ Sex

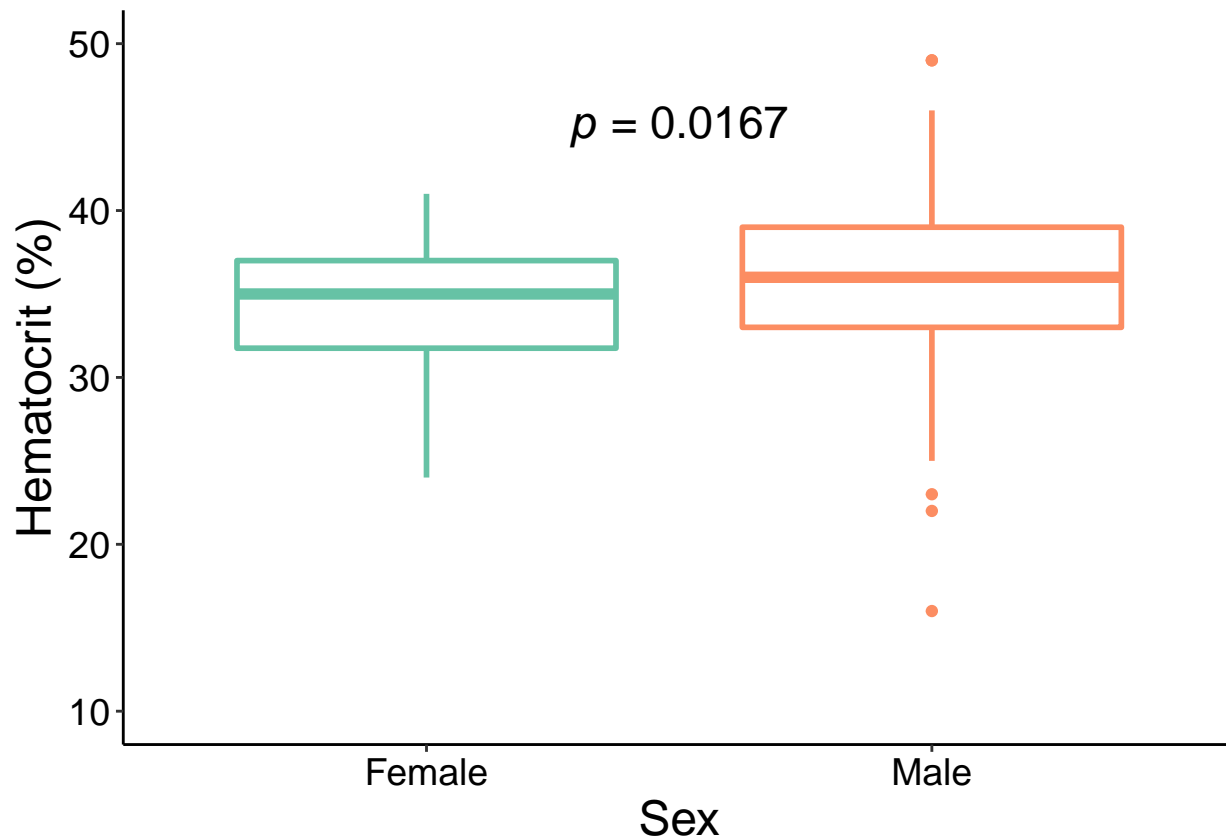
males have significantly higher hematocrit %

```

morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = sex_M_F,
                  y = hematocrit_percent,
                  color = sex_M_F
                  ),
              size = 1,
              alpha = 1) +
  theme_classic() +
  xlab("Sex") +
  ylab("Hematocrit (%)") +
  annotate("text", x = 1.5, y = 45,
          label = "paste(italic(p), \" = 0.0167\\")",
          parse = TRUE,
          size = 6) +
  ylim(10, 50) +
  scale_x_discrete(labels = c("F" = "Female",
                             "M" = "Male")) +
  scale_color_brewer(palette = "Set2") +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 18),
        axis.text = element_text(color = "black",
                                  family = "sans",
                                  size = 14),
        legend.text.align = 0,
        legend.position = "none"
  ) -> hct_sex_fig
hct_sex_fig

```

```
## Warning: Removed 30 rows containing non-finite values (stat_boxplot).
```



```
# export figure
#ggsave(filename = "hct_sex_fig.tiff",
#       plot = hct_sex_fig,
#       path = "./final_figures",
#       device = "tiff",
#       dpi = 1200,
#       width = 6, height = 4)
```

```
# lms
aov_hct_sex <- aov(hematocrit_percent ~ sex_M_F,
                  data = morpho_blood_SMI)
TukeyHSD(aov_hct_sex) # this is the stat to present
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = hematocrit_percent ~ sex_M_F, data = morpho_blood_SMI)
##
## $sex_M_F
##      diff      lwr      upr      p adj
## M-F 2.665584 0.4904345 4.840734 0.0167428
```

```
lm_hct_sex_mass <- glm(hematocrit_percent ~ sex_M_F*mass_g - mass_g,
                      data = morpho_blood_SMI)
summary(lm_hct_sex_mass)
```

```
##
## Call:
```

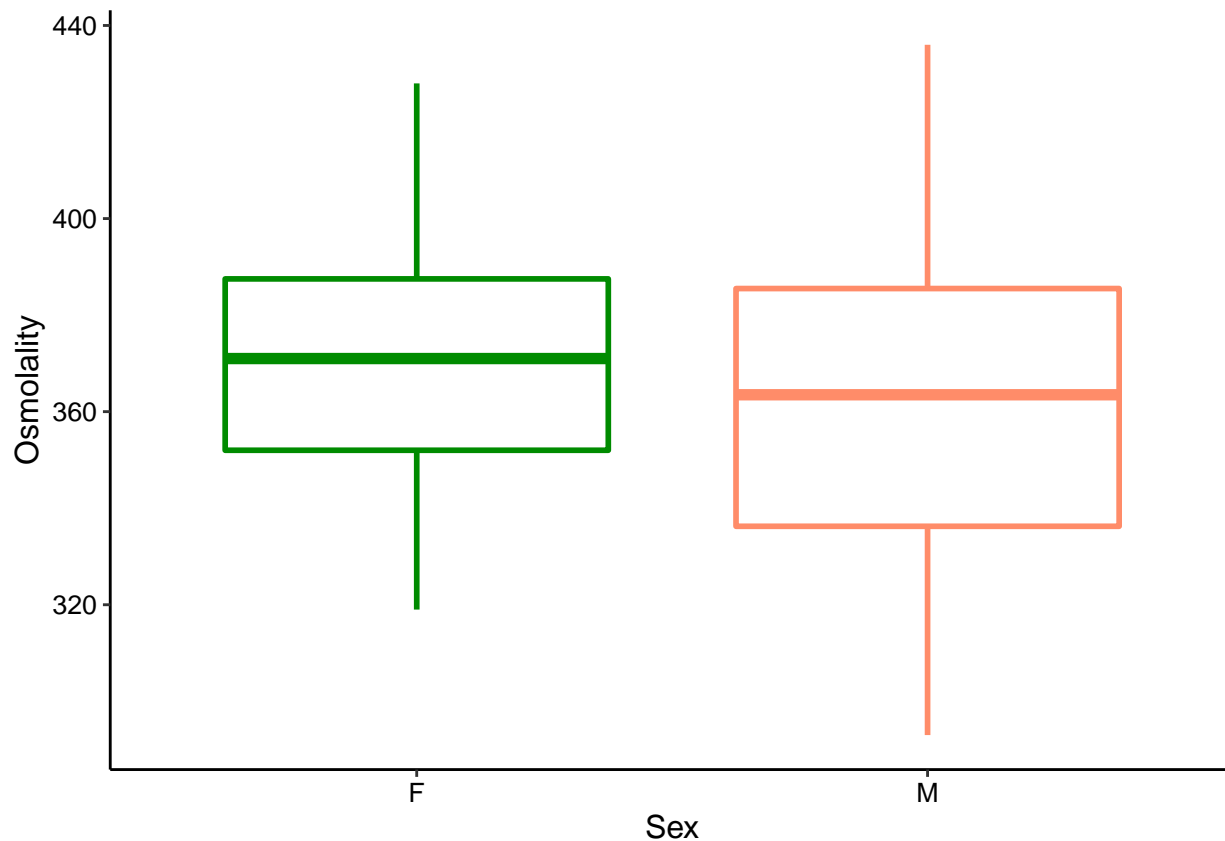
```
## glm(formula = hematocrit_percent ~ sex_M_F * mass_g - mass_g,
##      data = morpho_blood_SMI)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6306  -2.6187   0.6422   3.1481  17.2271
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    35.9163     3.7236   9.646  <2e-16 ***
## sex_M_FM       -0.9224     4.7553  -0.194   0.847
## sex_M_FF:mass_g -0.2244     0.3596  -0.624   0.534
## sex_M_FM:mass_g  0.1186     0.2568   0.462   0.645
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 34.18947)
##
##      Null deviance: 4219.7  on 120  degrees of freedom
## Residual deviance: 4000.2  on 117  degrees of freedom
## (27 observations deleted due to missingness)
## AIC: 776.68
##
## Number of Fisher Scoring iterations: 2
```

Hematocrit is significantly predicted by sex, but the interaction between sex and mass is ~nonexistent.

Osml ~ Sex

```
morpho_blood_SMI %>%
  ggplot(data = .) +
    geom_boxplot(aes(x = sex_M_F,
                     y = osmolality_mmol_kg,
                     color = sex_M_F
                    ),
                size = 1,
                alpha = 0.6) +
    scale_colour_manual(name = "Sex",
                       values = c("green4", "salmon1")) +
    theme_classic() +
    xlab("Sex") +
    ylab("Osmolality") +
    theme(text = element_text(color = "black", family = "sans", size = 12),
          axis.text = element_text(color = "black", family = "sans", size = 10),
          legend.text.align = 0,
          legend.position = "none"
    )
```

```
## Warning: Removed 3 rows containing non-finite values (stat_boxplot).
```

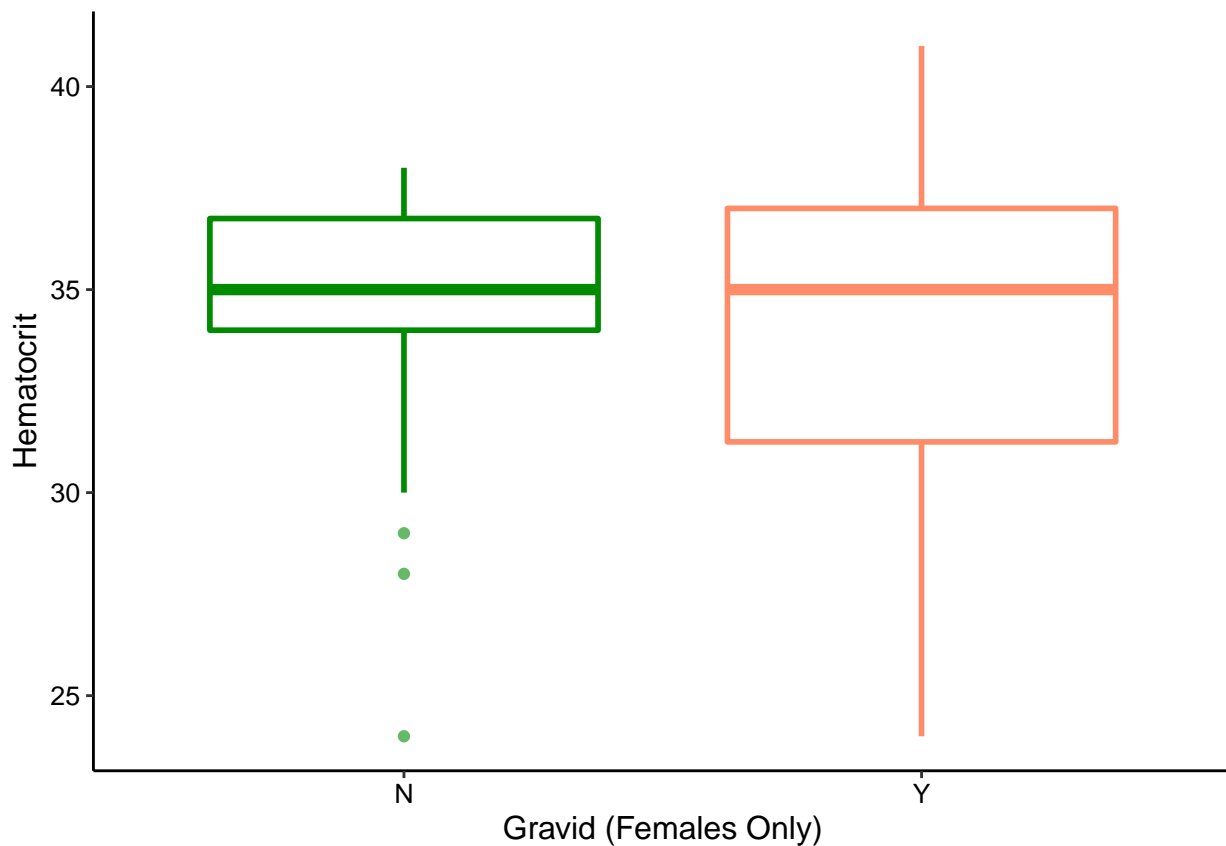
```
# lm
lm_osml_sex <- lm(osmolality_mmol_kg ~ sex_M_F,
                  data = morpho_blood_SMI)
summary(lm_osml_sex)

##
## Call:
## lm(formula = osmolality_mmol_kg ~ sex_M_F, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -68.612 -22.915   1.085  22.388  74.388
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   371.915     4.509   82.479  <2e-16 ***
## sex_M_FM      -10.303     5.485   -1.878   0.0624 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.91 on 143 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.02408,    Adjusted R-squared:  0.01725
## F-statistic: 3.528 on 1 and 143 DF,  p-value: 0.06237
```

Hct ~ Gravidity

```
morpho_blood_SMI %>%
  dplyr::filter(sex_M_F == 'F') %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = gravid_Y_N,
                    y = hematocrit_percent,
                    color = gravid_Y_N
                  ),
              size = 1,
              alpha = 0.6) +
  scale_colour_manual(name = "Gravid",
                      values = c("green4", "salmon1") ) +
  theme_classic() +
  xlab("Gravid (Females Only)") +
  ylab("Hematocrit") +
  theme(text = element_text(color = "black", family = "sans", size = 12),
        axis.text = element_text(color = "black", family = "sans", size = 10),
        legend.text.align = 0,
        legend.position = "none"
  )
)
```

Warning: Removed 4 rows containing non-finite values (stat_boxplot).



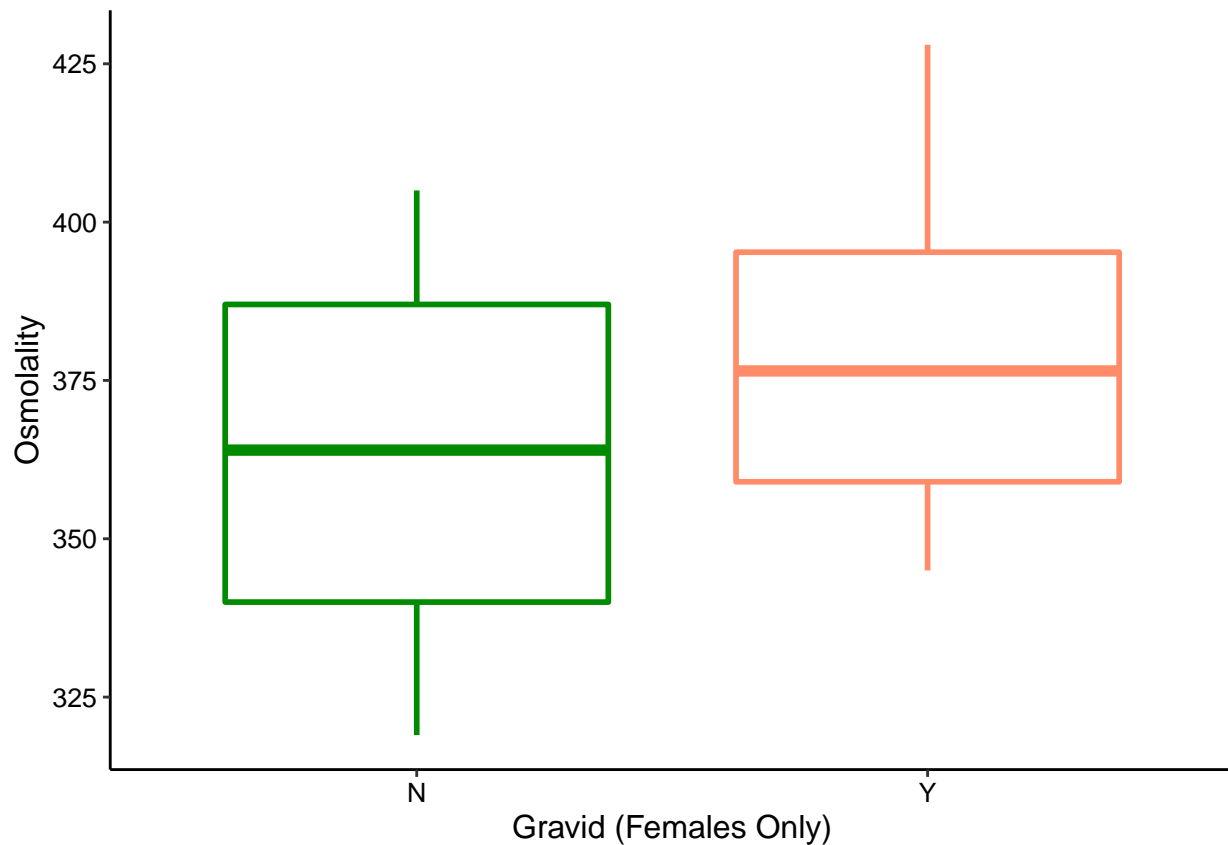
```
# lm
lm_hct_gravid <- lm(hematocrit_percent ~ gravid_Y_N,
                    data = morpho_blood_SMI)
summary(lm_hct_gravid)
```

```
##
## Call:
## lm(formula = hematocrit_percent ~ gravid_Y_N, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.111  -1.596   1.271   3.080   7.654
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    34.111      1.017  33.546  <2e-16 ***
## gravid_Y_NY     -0.765      1.323  -0.578    0.566
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.314 on 42 degrees of freedom
## (104 observations deleted due to missingness)
## Multiple R-squared:  0.0079, Adjusted R-squared: -0.01572
## F-statistic: 0.3344 on 1 and 42 DF, p-value: 0.5662
```

Osml ~ Gravidity

```
morpho_blood_SMI %>%
  dplyr::filter(sex_M_F == 'F') %>%
  ggplot(data = .) +
    geom_boxplot(aes(x = gravid_Y_N,
                     y = osmolality_mmol_kg,
                     color = gravid_Y_N
                    ),
                size = 1,
                alpha = 0.6) +
    scale_colour_manual(name = "Gravid",
                       values = c("green4", "salmon1") ) +
    theme_classic() +
    xlab("Gravid (Females Only)") +
    ylab("Osmolality") +
    theme(text = element_text(color = "black", family = "sans", size = 12),
          axis.text = element_text(color = "black", family = "sans", size = 10),
          legend.text.align = 0,
          legend.position = "none"
    )
```

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



```
# lm
lm_osml_gravid <- lm(osmolality_mmol_kg ~ gravid_Y_N,
  data = morpho_blood_SMI)
summary(lm_osml_gravid)

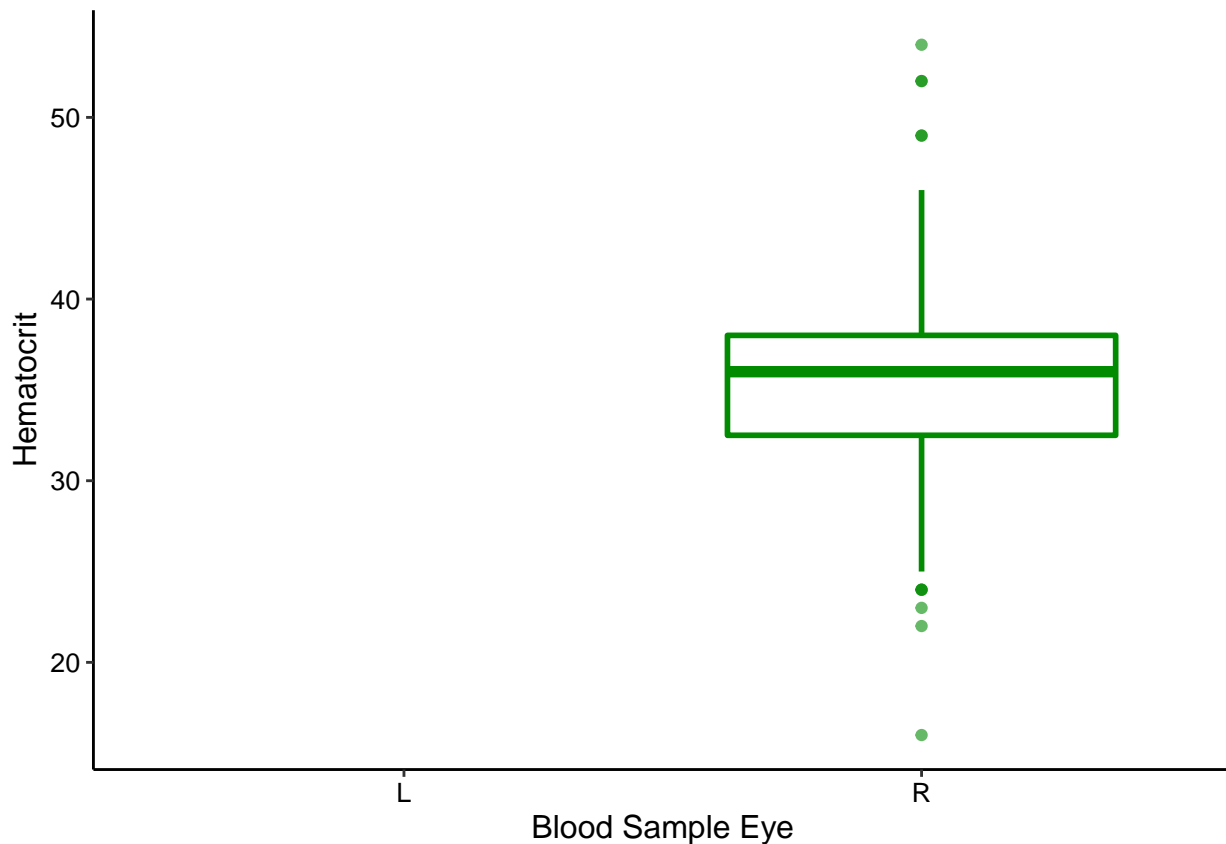
##
## Call:
## lm(formula = osmolality_mmol_kg ~ gravid_Y_N, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -45.333 -22.186  -0.333   22.667   49.962
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   364.333      5.604   65.011  <2e-16 ***
## gravid_Y_N     13.705      7.535    1.819   0.0756 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.68 on 45 degrees of freedom
## (101 observations deleted due to missingness)
## Multiple R-squared:  0.06848,    Adjusted R-squared:  0.04778
## F-statistic: 3.308 on 1 and 45 DF,  p-value: 0.07559
```

Hct ~ Sample Eye

Actually, any blood samples not taken from the right eye ended up being excluded or didn't have hematocrit values, so we can't test this difference.

```
morpho_blood_SMI %>%
  dplyr::filter(blood_sample_eye %in% c("R", "L")) %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = blood_sample_eye,
                    y = hematocrit_percent,
                    color = blood_sample_eye
                  ),
               size = 1,
               alpha = 0.6) +
  scale_colour_manual(name = "Blood Sample Eye",
                     values = c("green4", "salmon1")) +
  theme_classic() +
  xlab("Blood Sample Eye") +
  ylab("Hematocrit") +
  theme(text = element_text(color = "black", family = "sans", size = 12),
        axis.text = element_text(color = "black", family = "sans", size = 10),
        legend.text.align = 0,
        legend.position = "none"
  )
```

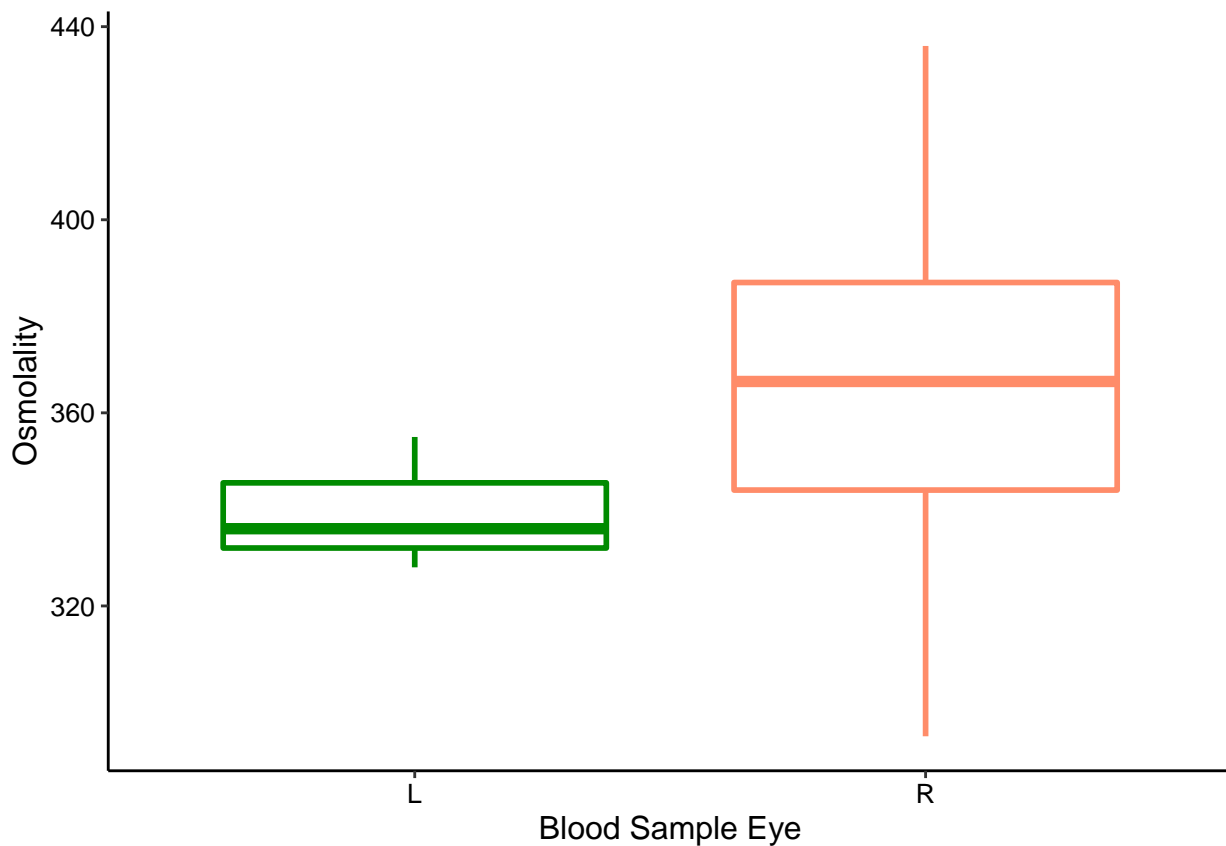
Warning: Removed 27 rows containing non-finite values (stat_boxplot).



Osml ~ Sample Eye

```
morpho_blood_SMI %>%  
  dplyr::filter(blood_sample_eye %in% c("R", "L")) %>%  
  ggplot(data = .) +  
    geom_boxplot(aes(x = blood_sample_eye,  
                     y = osmolality_mmol_kg,  
                     color = blood_sample_eye  
                    ),  
                size = 1,  
                alpha = 0.6) +  
    scale_colour_manual(name = "Blood Sample Eye",  
                        values = c("green4", "salmon1") ) +  
    theme_classic() +  
    xlab("Blood Sample Eye") +  
    ylab("Osmolality") +  
    theme(text = element_text(color = "black", family = "sans", size = 12),  
          axis.text = element_text(color = "black", family = "sans", size = 10),  
          legend.text.align = 0,  
          legend.position = "none"  
    )
```

Warning: Removed 3 rows containing non-finite values (stat_boxplot).



```
# lm  
lm_osml_eye <- lm(osmolality_mmol_kg ~ blood_sample_eye,  
                  data = morpho_blood_SMI)  
summary(lm_osml_eye)
```

```
##
## Call:
## lm(formula = osmolality_mmol_kg ~ blood_sample_eye, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -71.986 -19.986   1.014  22.014  71.014
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      400.50      21.85  18.330 <2e-16 ***
## blood_sample_eyeL    -60.83      28.21   -2.157  0.0327 *
## blood_sample_eyeR   -35.51      22.01   -1.614  0.1088
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.9 on 142 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.03175,    Adjusted R-squared:  0.01811
## F-statistic: 2.328 on 2 and 142 DF,  p-value: 0.1012
summary(morpho_blood_SMI$blood_sample_eye)
```

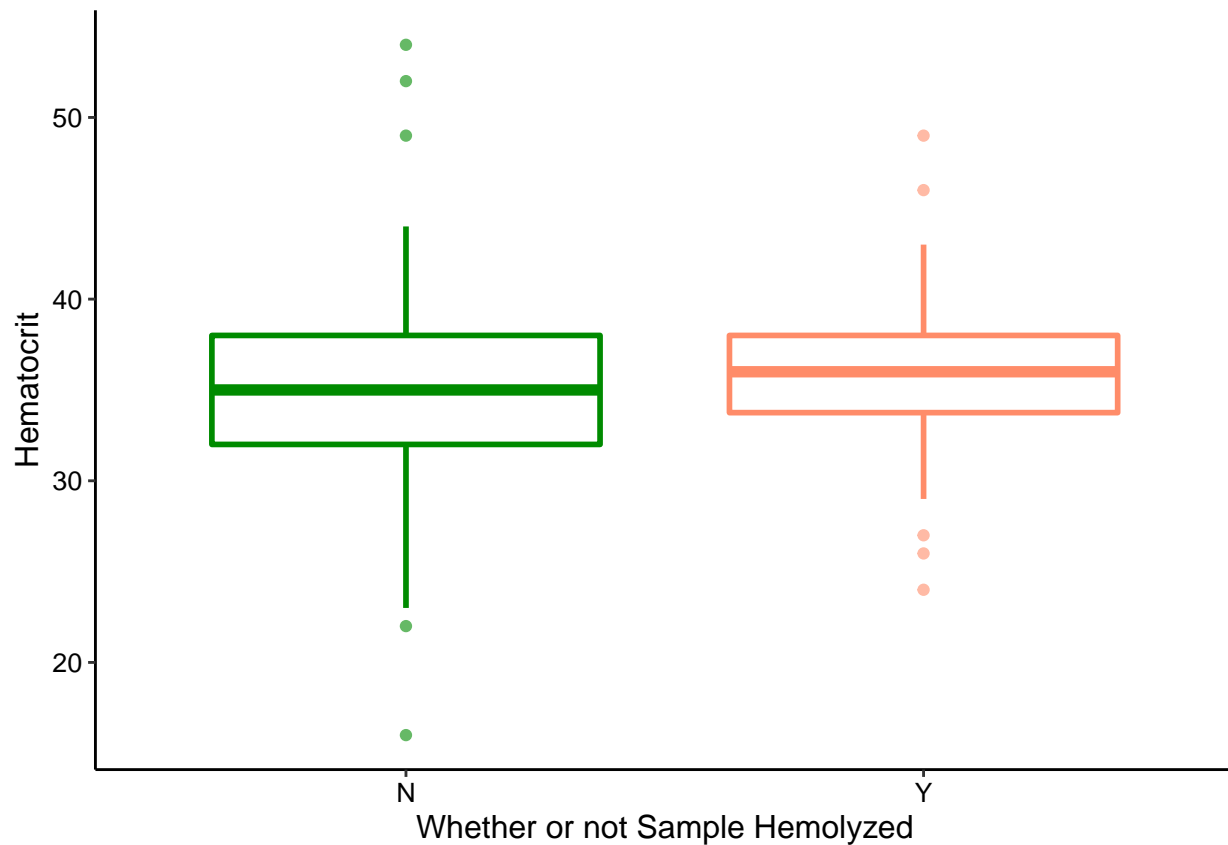
```
## both    L    R
##      2    4 142
```

There IS a significant difference, but upon further inspection, the number of blood draws taken from the right postorbital sinus vs left or both is so uneven, this would be unhelpful in the actual model.

Hct ~ Hemolyzed/Not

```
morpho_blood_SMI %>%
  dplyr::filter(hemolyzed %in% c("Y", "N")) %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = hemolyzed,
                   y = hematocrit_percent,
                   color = hemolyzed
                  ),
               size = 1,
               alpha = 0.6) +
  scale_colour_manual(name = "Blood Sample Eye",
                     values = c("green4", "salmon1", "green4", "salmon1") ) +
  theme_classic() +
  xlab("Whether or not Sample Hemolyzed") +
  ylab("Hematocrit") +
  theme(text = element_text(color = "black", family = "sans", size = 12),
        axis.text = element_text(color = "black", family = "sans", size = 10),
        legend.text.align = 0,
        legend.position = "none"
  )
```

```
## Warning: Removed 25 rows containing non-finite values (stat_boxplot).
```



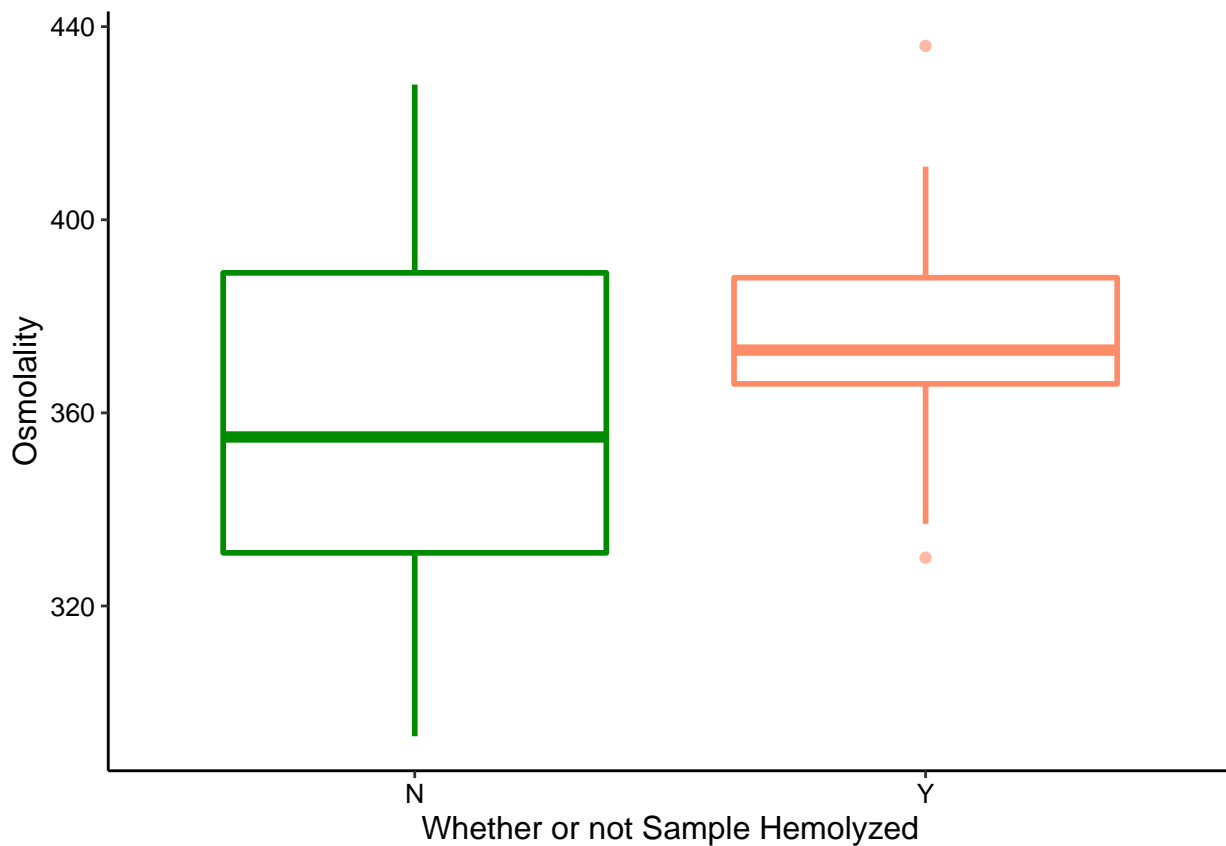
```
# lm
lm_hct_hem <- lm(hematocrit_percent ~ hemolyzed,
                 data = morpho_blood_SMI)
summary(lm_hct_hem)

##
## Call:
## lm(formula = hematocrit_percent ~ hemolyzed, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.0952  -2.7778  -0.0952   2.2222  18.9048
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   35.0952     0.7636  45.958  <2e-16 ***
## hemolyzedY     0.6825     1.2664   0.539    0.591
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.061 on 97 degrees of freedom
## (49 observations deleted due to missingness)
## Multiple R-squared:  0.002986,    Adjusted R-squared:  -0.007293
## F-statistic: 0.2905 on 1 and 97 DF,  p-value: 0.5911
```


Osml ~ Hemolyzed/Not

```
morpho_blood_SMI %>%
  dplyr::filter(hemolyzed %in% c("Y", "N")) %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = hemolyzed,
                    y = osmolality_mmol_kg,
                    color = hemolyzed
                  ),
               size = 1,
               alpha = 0.6) +
  scale_colour_manual(name = "Blood Sample Eye",
                      values = c("green4", "salmon1", "green4", "salmon1") ) +
  theme_classic() +
  xlab("Whether or not Sample Hemolyzed") +
  ylab("Osmolality") +
  theme(text = element_text(color = "black", family = "sans", size = 12),
        axis.text = element_text(color = "black", family = "sans", size = 10),
        legend.text.align = 0,
        legend.position = "none"
  )
```

Warning: Removed 1 rows containing non-finite values (stat_boxplot).



```
# lm
lm_osml_hem <- lm(osmolality_mmol_kg ~ hemolyzed,
                  data = morpho_blood_SMI)
summary(lm_osml_hem)
```

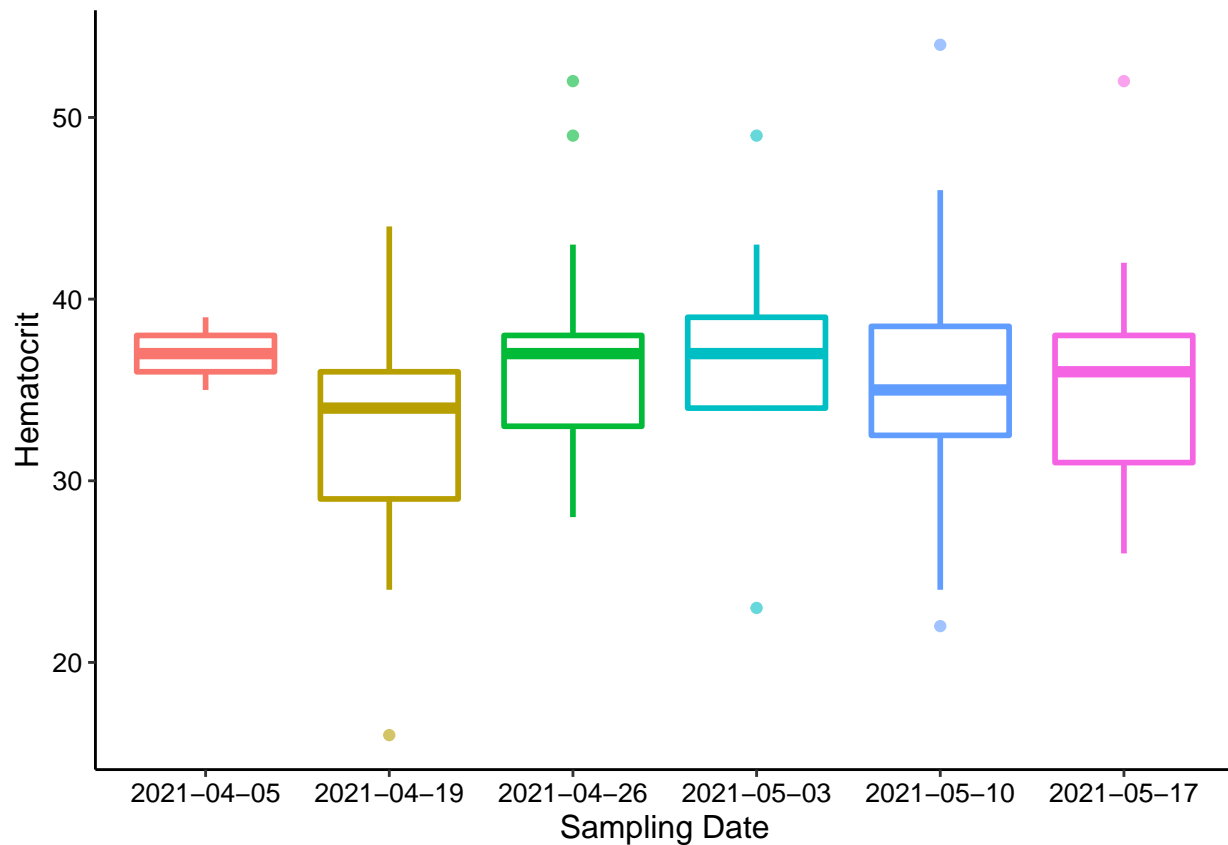
```
##
## Call:
## lm(formula = osmolality_mmol_kg ~ hemolyzed, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -66.690 -23.858  -2.026   24.310   68.310
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   359.690      3.523  102.109  <2e-16 ***
## hemolyzedY     15.335       6.256   2.451   0.0157 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.29 on 121 degrees of freedom
## (25 observations deleted due to missingness)
## Multiple R-squared:  0.04731,    Adjusted R-squared:  0.03944
## F-statistic: 6.009 on 1 and 121 DF,  p-value: 0.01566
```

Osmolality in hemolyzed samples versus not are significantly different! This is an unfortunate trait of the dataset, and not something we are specifically interested in testing, so we will include it as a random factor in the LMM.

Hct ~ Week

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = as.factor(date),
                  y = hematocrit_percent,
                  color = as.factor(date)
                ),
              size = 1,
              alpha = 0.6) +
  theme_classic() +
  xlab("Sampling Date") +
  ylab("Hematocrit") +
  theme(text = element_text(color = "black", family = "sans", size = 12),
        axis.text = element_text(color = "black", family = "sans", size = 10),
        legend.text.align = 0,
        legend.position = "none"
  )
```

```
## Warning: Removed 27 rows containing non-finite values (stat_boxplot).
```



```
# lm
lm_hct_wk <- lm(hematocrit_percent ~ date,
  data = morpho_blood_SMI)
summary(lm_hct_wk)

##
## Call:
## lm(formula = hematocrit_percent ~ date, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.8497  -2.8497   0.1348   2.6425  18.3886
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -644.69394   942.77212  -0.684   0.495
## date           0.03627    0.05028   0.721   0.472
##
## Residual standard error: 5.942 on 119 degrees of freedom
## (27 observations deleted due to missingness)
## Multiple R-squared:  0.004353,    Adjusted R-squared:  -0.004013
## F-statistic: 0.5203 on 1 and 119 DF,  p-value: 0.4721
```

Osml ~ Week

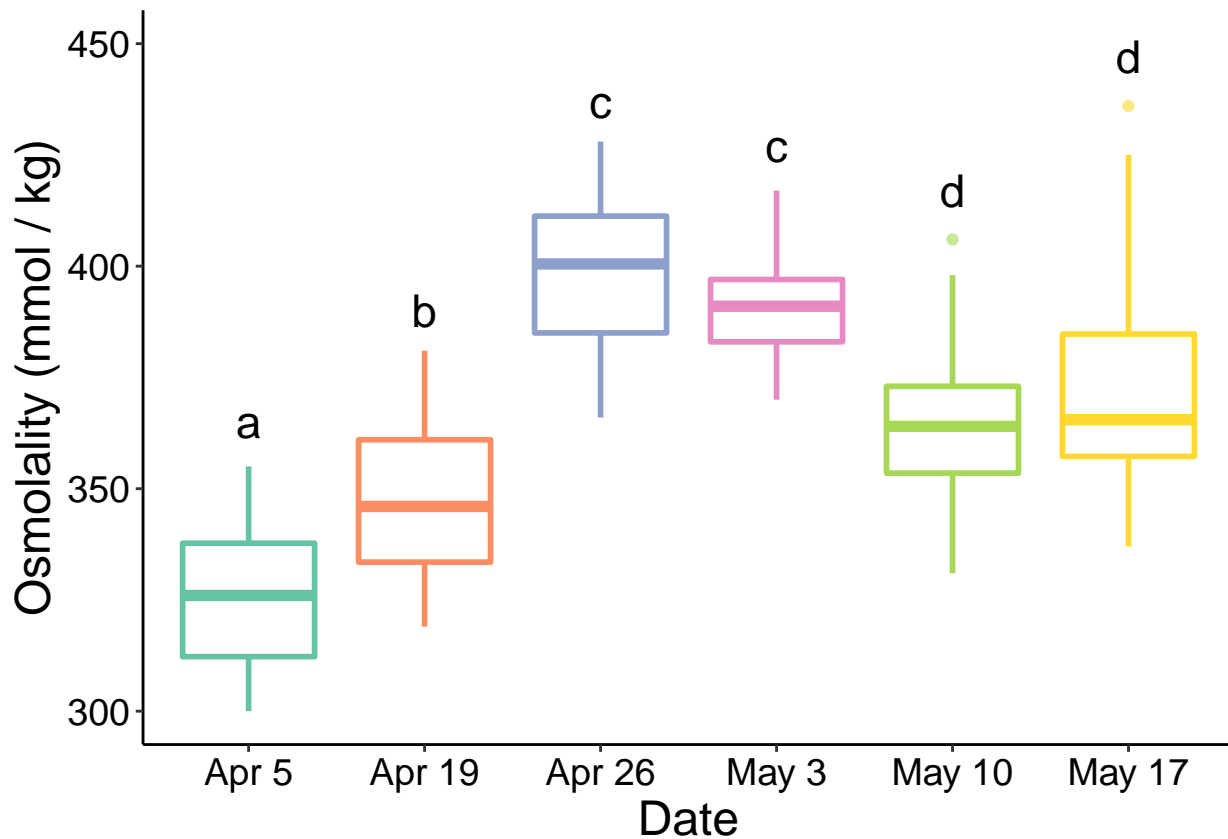
```
morpho_blood_SMI %>%
  ggplot(data = .) +
```

```

geom_boxplot(aes(x = as.factor(date),
                 y = osmolality_mmol_kg,
                 color = as.factor(date)
                ),
             size = 1,
             alpha = 0.6) +
theme_classic() +
xlab("Date") +
ylab("Osmolality (mmol / kg)") +
annotate("text", x = 1, y = 365, label = "a", size = 6) +
annotate("text", x = 2, y = 390, label = "b", size = 6) +
annotate("text", x = 3, y = 437, label = "c", size = 6) +
annotate("text", x = 4, y = 427, label = "c", size = 6) +
annotate("text", x = 5, y = 417, label = "d", size = 6) +
annotate("text", x = 6, y = 447, label = "d", size = 6) +
scale_x_discrete(labels = c("2021-04-05" = "Apr 5",
                           "2021-04-19" = "Apr 19",
                           "2021-04-26" = "Apr 26",
                           "2021-05-03" = "May 3",
                           "2021-05-10" = "May 10",
                           "2021-05-17" = "May 17")) +
scale_color_brewer(palette = "Set2") +
ylim(300, 450) +
theme(text = element_text(color = "black",
                           family = "sans",
                           size = 18),
      axis.text = element_text(color = "black",
                                family = "sans",
                                size = 14),
      legend.text.align = 0,
      legend.position = "none"
) -> osml_date_fig
osml_date_fig

```

```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```



```
# export figure
#ggsave(filename = "osml_date_fig.tiff",
#       plot = osml_date_fig,
#       path = "./final_figures",
#       device = "tiff",
#       dpi = 1200,
#       width = 6, height = 4)

# lm
lm_osml_wk <- glm(osmolality_mmol_kg ~ date,
                  data = morpho_blood_SMI)
summary(lm_osml_wk)

##
## Call:
## glm(formula = osmolality_mmol_kg ~ date, data = morpho_blood_SMI)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -48.236  -21.636   -2.635   20.364   65.564
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.999e+04  2.954e+03  -6.766 3.15e-10 ***
## date         1.086e+00  1.576e-01   6.889 1.65e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for gaussian family taken to be 735.2172)
##
## Null deviance: 140031 on 144 degrees of freedom
## Residual deviance: 105136 on 143 degrees of freedom
## (3 observations deleted due to missingness)
## AIC: 1372.5
##
## Number of Fisher Scoring iterations: 2
```

Osmolality was VERY different by week, and that was not easily attributable to climate factors (see below). We also experienced technical difficulties with the osmometer between some weeks, so some of the variation may be attributable to technical error. Thus date/week will be a good random factor to include in the LMM.

distinguish pairwise differences using an ANOVA:

```
osml_date_aov <- aov(osmolality_mmol_kg ~ as.factor(date),
                     data = morpho_blood_SMI)
TukeyHSD(osml_date_aov)
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = osmolality_mmol_kg ~ as.factor(date), data = morpho_blood_SMI)
##
## $`as.factor(date)`
##
```

		diff	lwr	upr	p adj
##	2021-04-19-2021-04-05	22.891304	7.979417	37.803191	0.0002634
##	2021-04-26-2021-04-05	75.076923	60.628732	89.525114	0.0000000
##	2021-05-03-2021-04-05	66.657895	50.935165	82.380624	0.0000000
##	2021-05-10-2021-04-05	41.543478	26.631591	56.455365	0.0000000
##	2021-05-17-2021-04-05	46.785714	32.597872	60.973557	0.0000000
##	2021-04-26-2021-04-19	52.185619	37.273732	67.097506	0.0000000
##	2021-05-03-2021-04-19	43.766590	27.616718	59.916462	0.0000000
##	2021-05-10-2021-04-19	18.652174	3.290581	34.013766	0.0078361
##	2021-05-17-2021-04-19	23.894410	9.234635	38.554185	0.0000857
##	2021-05-03-2021-04-26	-8.419028	-24.141758	7.303701	0.6342961
##	2021-05-10-2021-04-26	-33.533445	-48.445332	-18.621558	0.0000000
##	2021-05-17-2021-04-26	-28.291209	-42.479051	-14.103367	0.0000008
##	2021-05-10-2021-05-03	-25.114416	-41.264289	-8.964544	0.0002088
##	2021-05-17-2021-05-03	-19.872180	-35.356006	-4.388355	0.0039942
##	2021-05-17-2021-05-10	5.242236	-9.417539	19.902011	0.9059063

Osml, Humidity, Week

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = as.factor(date),
                  y = osmolality_mmol_kg,
                  color = as.factor(date)
                ),
              size = 1,
              alpha = 0.6) +
  geom_jitter(aes(x = as.factor(date),
                  # added multiplier to put on same axis range
                  y = 35*abs_humidity_g_m3_interpol)) +
```

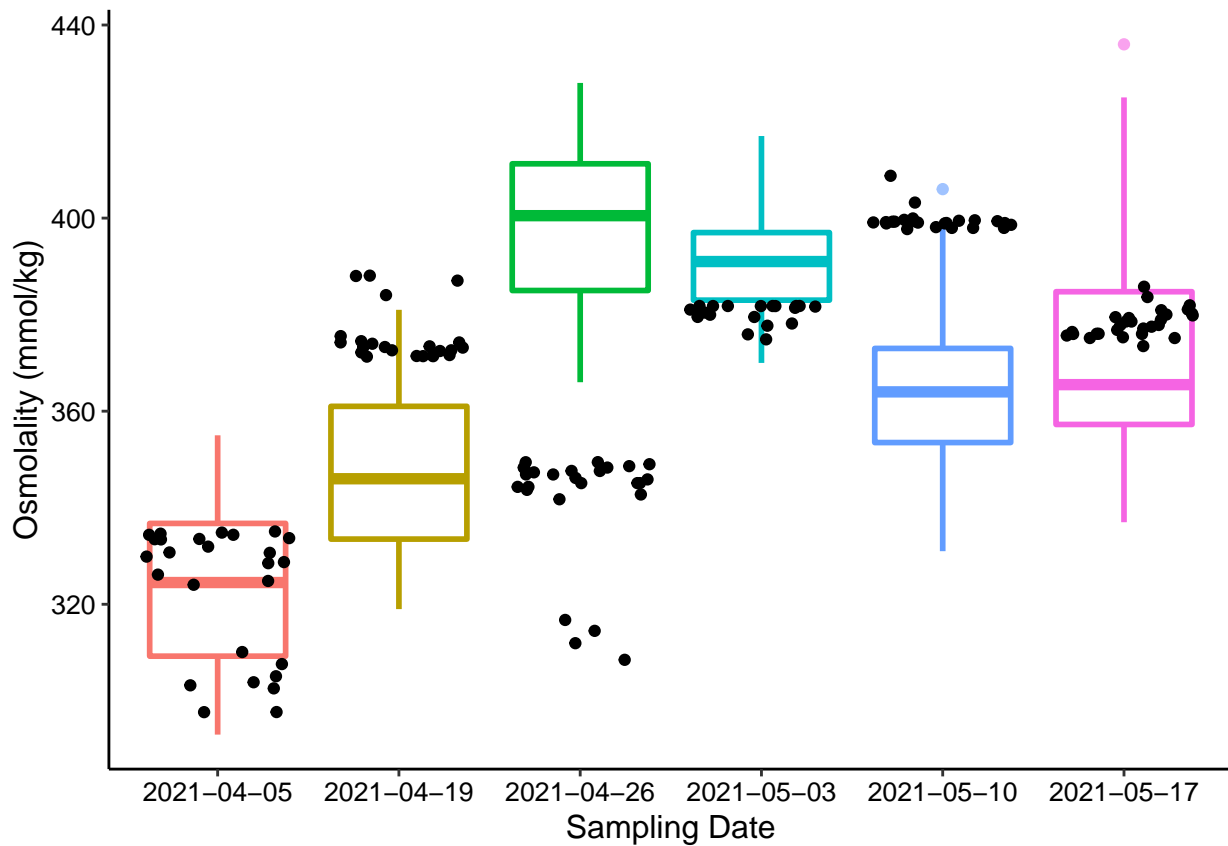
```

theme_classic() +
xlab("Sampling Date") +
ylab("Osmolality (mmol/kg)") +
theme(text = element_text(color = "black", family = "sans", size = 12),
      axis.text = element_text(color = "black", family = "sans", size = 10),
      legend.text.align = 0,
      legend.position = "none"
)

```

Warning: Removed 3 rows containing non-finite values (stat_boxplot).

Warning: Removed 3 rows containing missing values (geom_point).



Osml ~ Abs. Humidity

```

morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = abs_humidity_g_m3_interpol,
                 y = osmolality_mmol_kg,
                 ),
            size = 1,
            alpha = 0.4) +
  stat_smooth(aes(x = abs_humidity_g_m3_interpol,
                  y = osmolality_mmol_kg),
             formula = y ~ x,
             method = "lm",
             se = F,

```

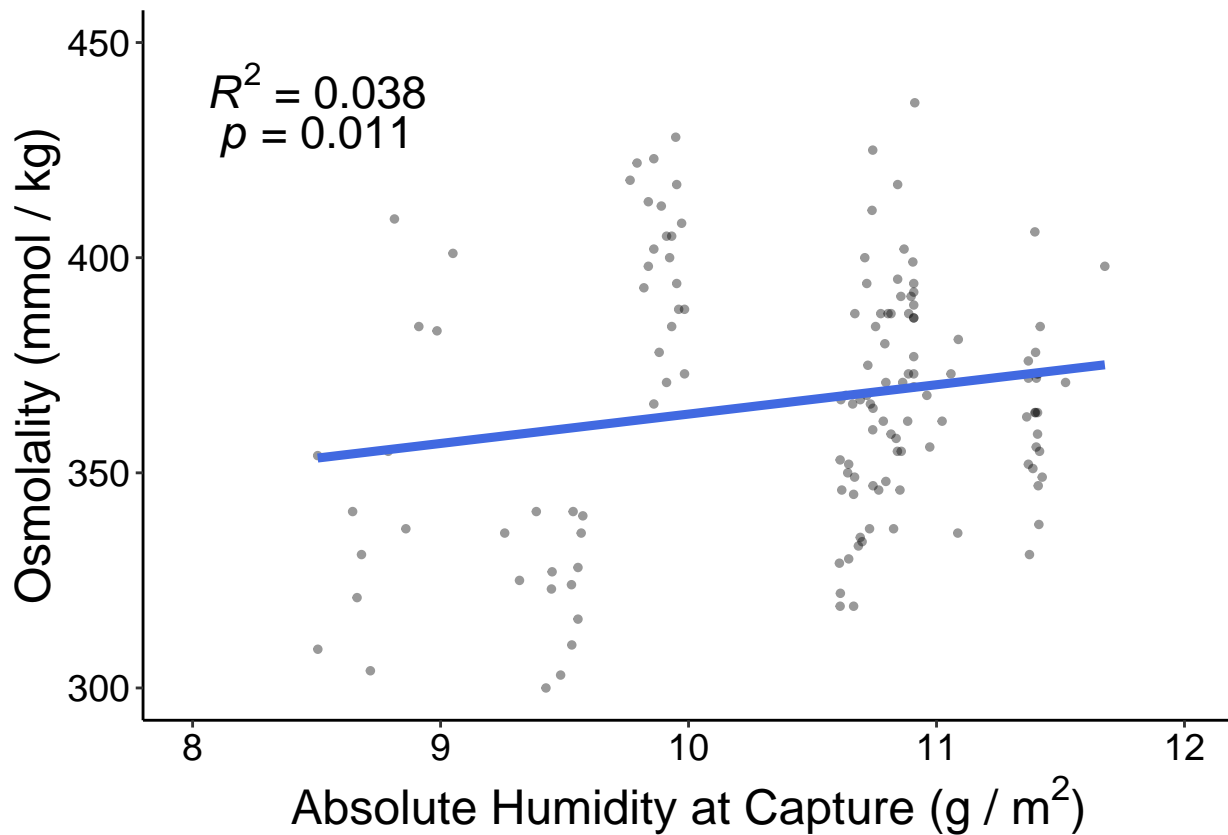
```

        color = "royalblue",
        size = 1.6,
        alpha = 1 ) +
theme_classic() +
xlab(bquote('Absolute Humidity at Capture (g / '*m^2*')')) +
ylab("Osmolality (mmol / kg)") +
xlim(8, 12) +
ylim(300, 450) +
annotate("text", x = 8.5, y = 440,
          label = "paste(italic(R) ^ 2, \" = 0.038\\")",
          parse = TRUE,
          size = 6) +
annotate("text", x = 8.5, y = 428,
          label = "paste(italic(p), \" = 0.011\\")",
          parse = TRUE,
          size = 6) +
theme(text = element_text(color = "black",
                           family = "sans",
                           size = 18),
      axis.text = element_text(color = "black",
                                family = "sans",
                                size = 14),
      legend.text.align = 0,
) -> osml_abhum_fig
osml_abhum_fig

```

Warning: Removed 7 rows containing non-finite values (stat_smooth).

Warning: Removed 7 rows containing missing values (geom_point).



```
# export figure
#ggsave(filename = "osml_abhum_fig.tiff",
#       plot = osml_abhum_fig,
#       path = "./final_figures",
#       device = "tiff",
#       dpi = 1200,
#       width = 6, height = 4)

# lm
lm_osml_abshum <- lm(osmolality_mmol_kg ~ abs_humidity_g_m3_interpol,
                     data = morpho_blood_SMI)
summary(lm_osml_abshum)
```

```
##
## Call:
## lm(formula = osmolality_mmol_kg ~ abs_humidity_g_m3_interpol,
##     data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -64.06  -21.24   -1.45   21.64   66.35
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    279.229     33.666   8.294 7.88e-14 ***
## abs_humidity_g_m3_interpol  8.286      3.216   2.576  0.011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 30.11 on 141 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared: 0.04496, Adjusted R-squared: 0.03818
## F-statistic: 6.637 on 1 and 141 DF, p-value: 0.01101
```

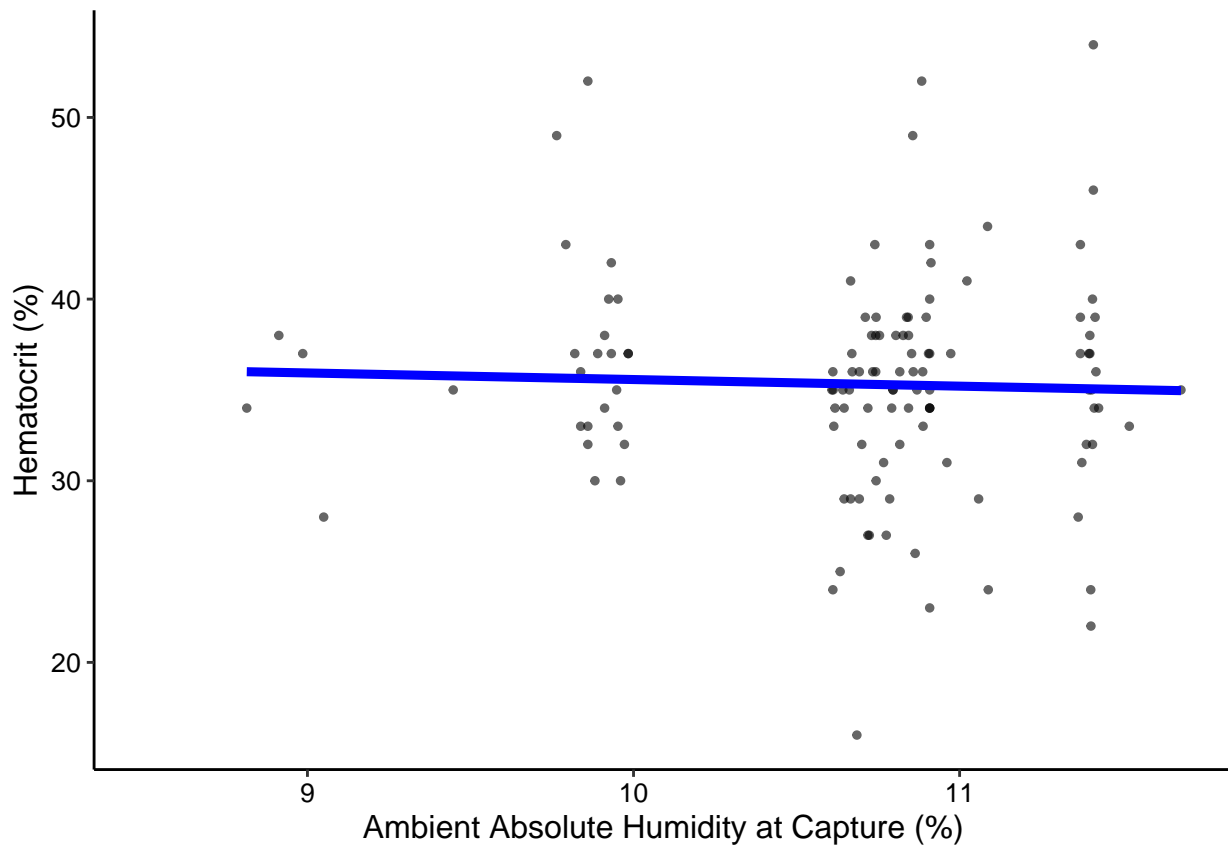
positive correlation...

Hct ~ Humidity

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = abs_humidity_g_m3_interpol,
                 y = hematocrit_percent,
                 ),
             size = 1,
             alpha = 0.6) +
  stat_smooth(aes(x = abs_humidity_g_m3_interpol,
                  y = hematocrit_percent),
              formula = y ~ x,
              method = "lm",
              se = F,
              color = "blue",
              size = 1.6,
              alpha = 1 ) +
  theme_classic() +
  xlab("Ambient Absolute Humidity at Capture (%)") +
  ylab("Hematocrit (%)") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```

```
## Warning: Removed 28 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 28 rows containing missing values (geom_point).
```



```
# lm
lm_hct_RH <- lm(hematocrit_percent ~ abs_humidity_g_m3_interpol,
                 data = morpho_blood_SMI)
summary(lm_hct_RH)

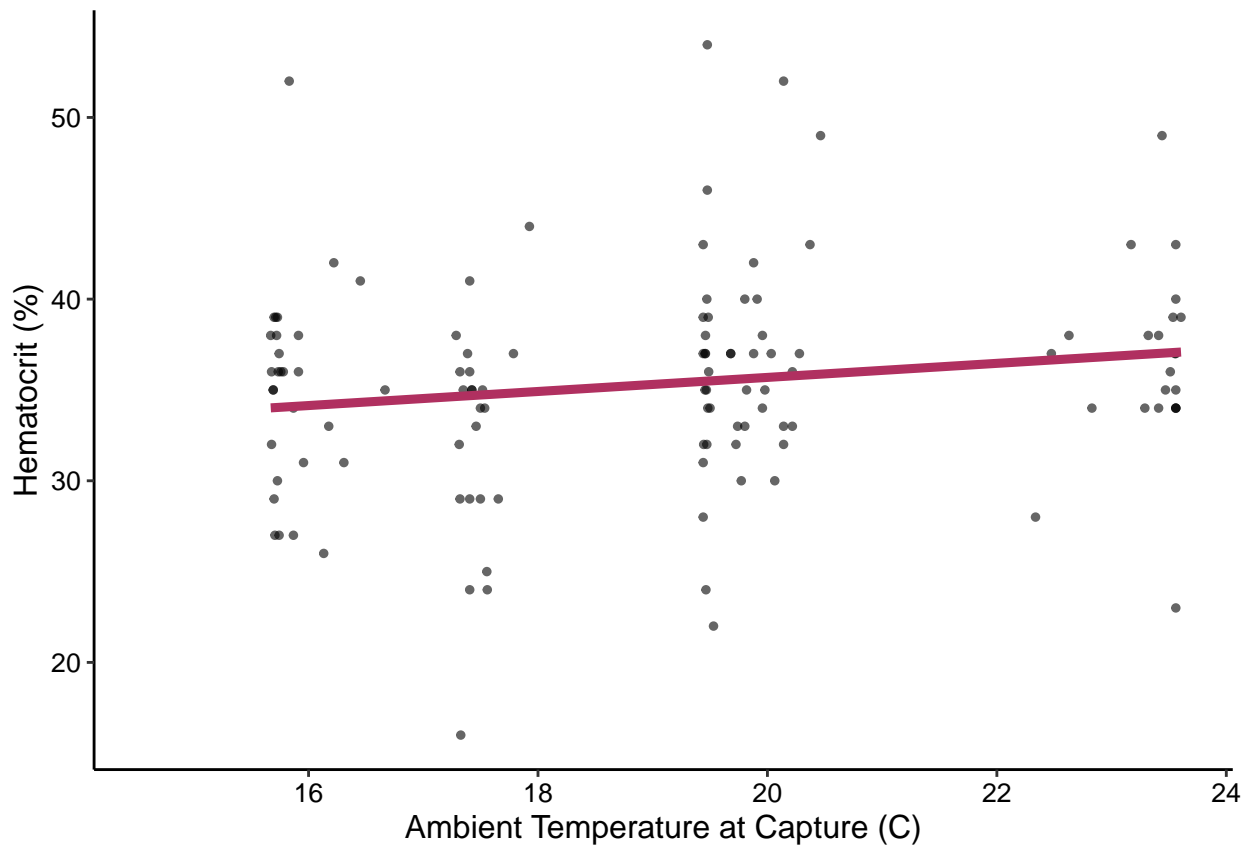
##
## Call:
## lm(formula = hematocrit_percent ~ abs_humidity_g_m3_interpol,
##     data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.3218  -2.7371  -0.0102   2.7080  18.9419
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      39.2073     9.8882   3.965 0.000126 ***
## abs_humidity_g_m3_interpol -0.3636     0.9248  -0.393 0.694867
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.967 on 118 degrees of freedom
## (28 observations deleted due to missingness)
## Multiple R-squared:  0.001309, Adjusted R-squared: -0.007155
## F-statistic: 0.1546 on 1 and 118 DF, p-value: 0.6949
```

Hct ~ Temperature

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = temp_C_interpol,
                 y = hematocrit_percent,
                 ),
            size = 1,
            alpha = 0.6) +
  stat_smooth(aes(x = temp_C_interpol,
                  y = hematocrit_percent),
              formula = y ~ x,
              method = "lm",
              se = F,
              color = "maroon",
              size = 1.6,
              alpha = 1 ) +
  theme_classic() +
  xlab("Ambient Temperature at Capture (C)") +
  ylab("Hematocrit (%)") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```

```
## Warning: Removed 28 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 28 rows containing missing values (geom_point).
```



```
# lm
lm_hct_temp <- lm(hematocrit_percent ~ temp_C_interpol,
                  data = morpho_blood_SMI)
summary(lm_hct_temp)
```

```
##
## Call:
## lm(formula = hematocrit_percent ~ temp_C_interpol, data = morpho_blood_SMI)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-18.6541	-3.0678	0.3085	2.3757	18.5137

```
##
## Coefficients:
```

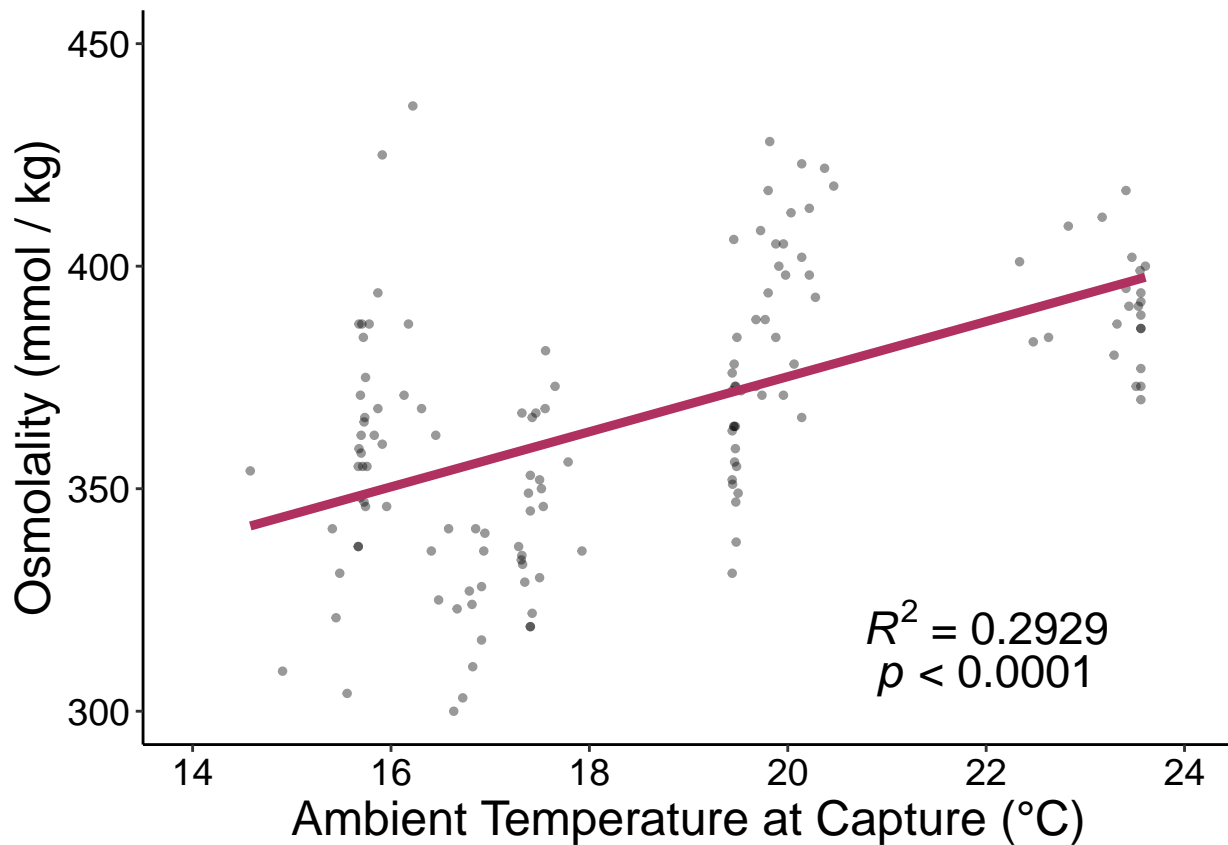
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	27.9447	3.9993	6.987	1.78e-10 ***
temp_C_interpol	0.3872	0.2079	1.862	0.0651 .

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.885 on 118 degrees of freedom
## (28 observations deleted due to missingness)
## Multiple R-squared:  0.02855,    Adjusted R-squared:  0.02032
## F-statistic: 3.468 on 1 and 118 DF,  p-value: 0.06505
```

Osml ~ Temperature

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = temp_C_interpol,
                 y = osmolality_mmol_kg,
                 ),
            size = 1,
            alpha = 0.4) +
  stat_smooth(aes(x = temp_C_interpol,
                  y = osmolality_mmol_kg),
              formula = y ~ x,
              method = "lm",
              se = F,
              color = "maroon",
              size = 1.6,
              alpha = 1 ) +
  theme_classic() +
  xlab("Ambient Temperature at Capture (°C)") +
  ylab("Osmolality (mmol / kg)") +
  annotate("text", x = 22, y = 320,
          label = "paste(italic(R) ^ 2, \" = 0.2929\\\")",
          parse = TRUE,
          size = 6) +
  annotate("text", x = 22, y = 308,
          label = "paste(italic(p), \" < 0.0001\\\")",
          parse = TRUE,
          size = 6) +
  xlim(14, 24) +
  ylim(300,450) +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 18),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 14),
        legend.text.align = 0,
  ) -> osml_temp_fig
osml_temp_fig

## Warning: Removed 7 rows containing non-finite values (stat_smooth).
## Warning: Removed 7 rows containing missing values (geom_point).
```



```
# export figure
#ggsave(filename = "osml_temp_fig.tiff",
#       plot = osml_temp_fig,
#       path = "./final_figures",
#       device = "tiff",
#       dpi = 1200,
#       width = 6, height = 4)

# lm
lm_osml_temp <- lm(osmolality_mmol_kg ~ temp_C_interpol,
                  data = morpho_blood_SMI)
summary(lm_osml_temp)

##
## Call:
## lm(formula = osmolality_mmol_kg ~ temp_C_interpol, data = morpho_blood_SMI)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -59.723 -16.089  -1.996  15.667   85.643
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   246.0362    15.6239   15.747 < 2e-16 ***
## temp_C_interpol  6.4316     0.8315    7.735 1.8e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

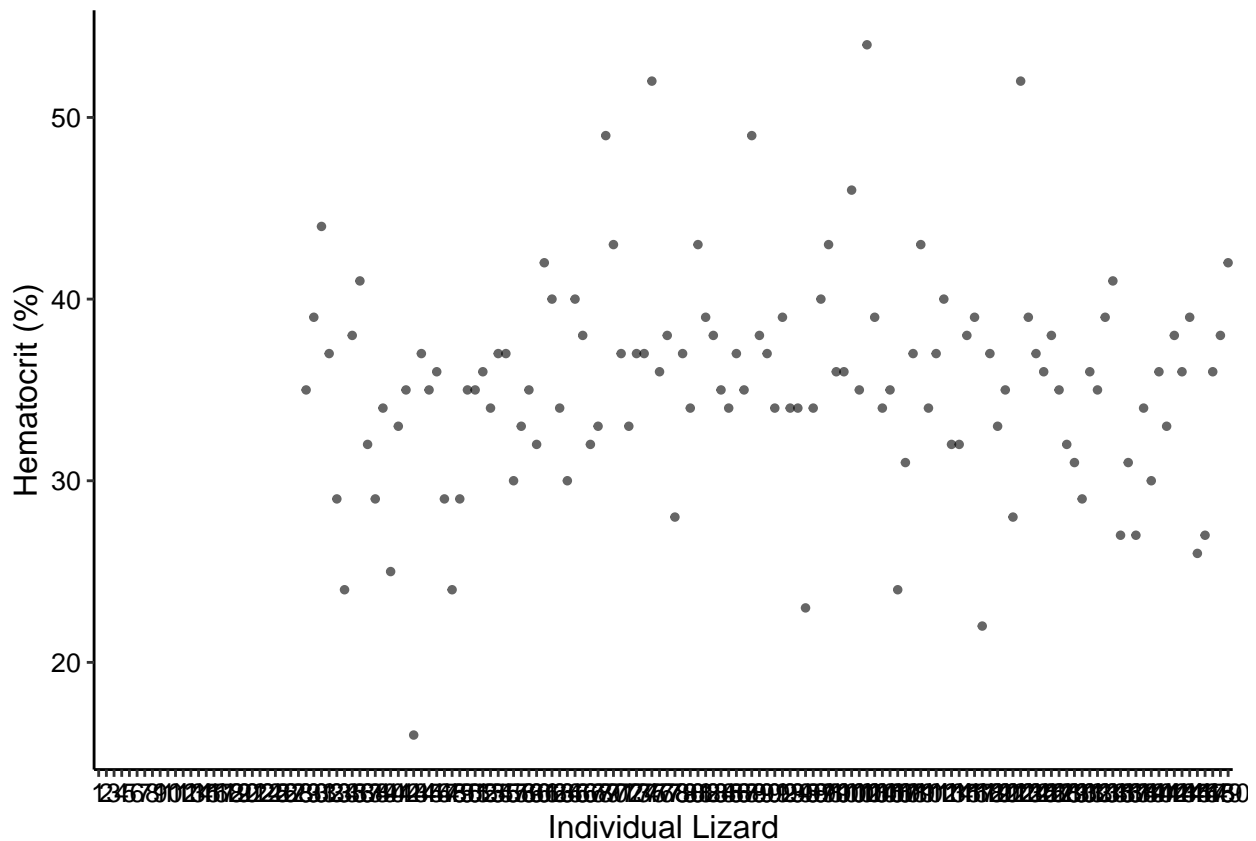
```
##
## Residual standard error: 25.82 on 141 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared: 0.2979, Adjusted R-squared: 0.2929
## F-statistic: 59.82 on 1 and 141 DF, p-value: 1.796e-12
```

Hct ~ Individual

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = individual_ID,
                 y = hematocrit_percent,
                 ),
            size = 1,
            alpha = 0.6) +
  stat_smooth(aes(x = individual_ID,
                 y = hematocrit_percent,
                 ),
            formula = y ~ x,
            method = "lm",
            color = "gray",
            se = F,
            size = 1.6,
            alpha = 1 ) +
  theme_classic() +
  xlab("Individual Lizard") +
  ylab("Hematocrit (%)") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```

```
## Warning: Removed 27 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 27 rows containing missing values (geom_point).
```

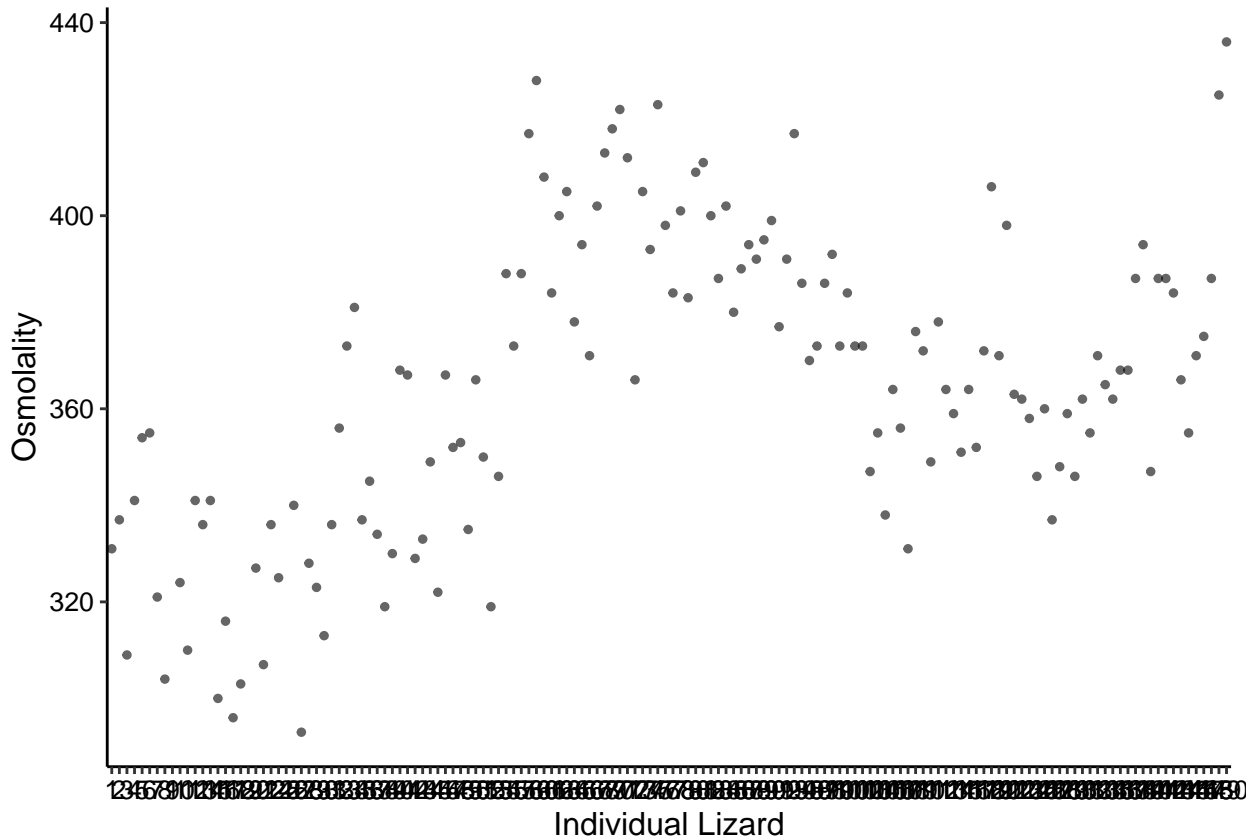
Osml ~ Individual

```
morpho_blood_SMI %>%
  ggplot(data = .) +
  geom_point(aes(x = individual_ID,
                 y = osmolality_mmol_kg,
                 ),
            size = 1,
            alpha = 0.6) +
  stat_smooth(aes(x = individual_ID,
                 y = osmolality_mmol_kg,
                 ),
            formula = y ~ x,
            method = "lm",
            color = "gray",
            se = F,
            size = 1.6,
            alpha = 1 ) +
  theme_classic() +
  xlab("Individual Lizard") +
  ylab("Osmolality") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                  family = "sans",
```

```
        size = 10),
    legend.text.align = 0,
)
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



Conclusion

Osmolality seems to be affected by: - SMI (large effect, although not statistically significant) - sex (NS) - whether or not the sample was hemolyzed () ! - *week/date of sampling!!* () - **capture temp & humidity (both *)**

So, for the LMM to predict osmolality, we will start with SMI, hemolyzed/not, week/date, and capture temp and absolute humidity as our predictor variables in the model.

Hematocrit was only meaningfully different based on sex, so that's the only statistic we'll include in the paper for what variables affect hematocrit.

What affects evaporative water loss?

Potential relationships: - CEWL ~ date/week, individual, SVL, SMI, gravidity, hct, osml, cloacal temp, ambient temp, ambient RH, measurement temp, measurement RH, **body region**

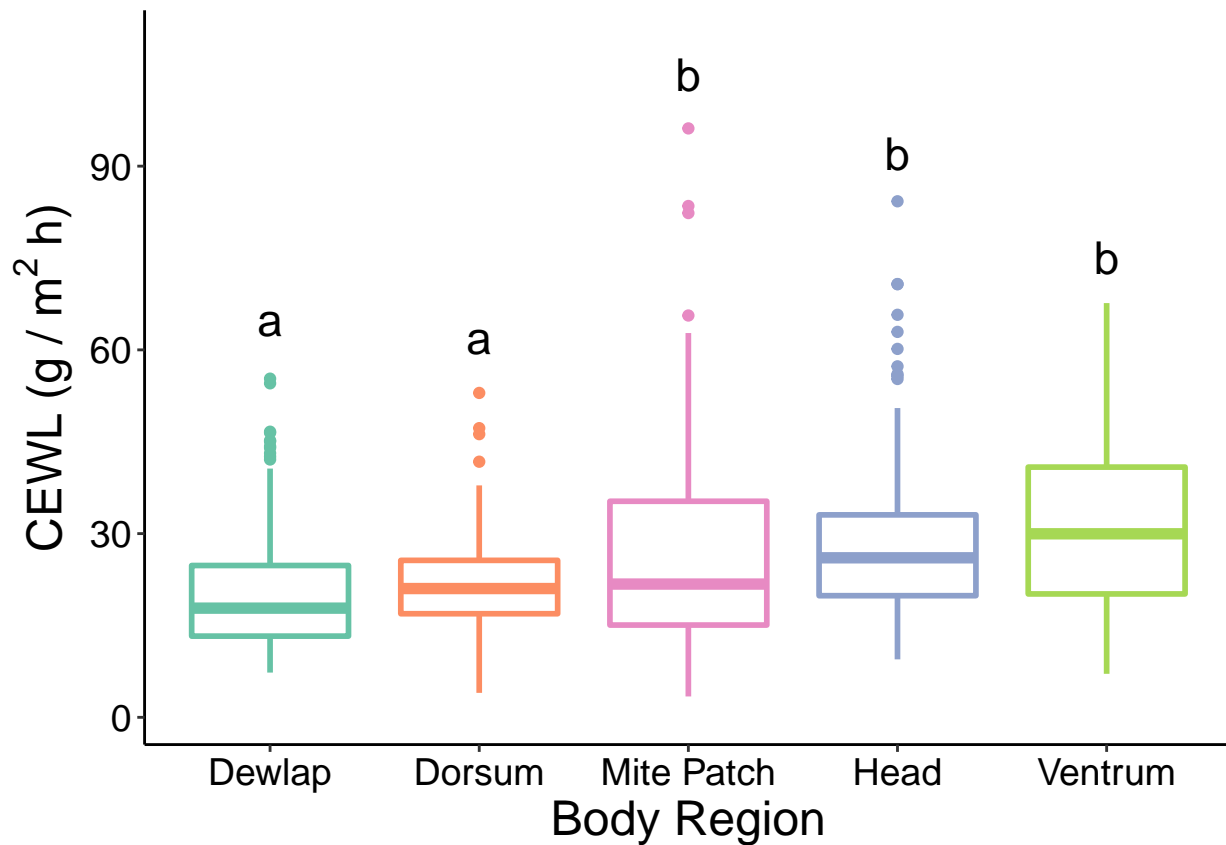
CEWL ~ Body Region

```

CEWL_data_full %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = reorder(region, TEWL_g_m2h),
                    y = TEWL_g_m2h,
                    color = region
                  ),
              size = 1,
              alpha = 1) +
  scale_x_discrete(labels = c("Dewlap", "Dorsum",
                              "Mite Patch", "Head", "Ventrums")) +

  theme_classic() +
  xlab("Body Region") +
  ylab(bquote('CEWL (g / '*m^2~h*')')) +
  annotate("text", x = 1, y = 65, label = "a", size = 6) +
  annotate("text", x = 2, y = 62, label = "a", size = 6) +
  annotate("text", x = 3, y = 105, label = "b", size = 6) +
  annotate("text", x = 4, y = 92, label = "b", size = 6) +
  annotate("text", x = 5, y = 75, label = "b", size = 6) +
  scale_color_brewer(palette = "Set2") +
  ylim(1, 110) +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 18),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 14),
        legend.text.align = 0,
        legend.position = "none"
  ) -> CEWL_region_fig
CEWL_region_fig

```



```
# export figure
#ggsave(filename = "CEWL_region_fig.tiff",
#       plot = CEWL_region_fig,
#       path = "./final_figures",
#       device = "tiff",
#       dpi = 1200,
#       width = 6, height = 4)
```

```
CEWL_lm1 <- lm(TEWL_g_m2h ~ region, data = CEWL_data_full)
summary(CEWL_lm1)
```

```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region, data = CEWL_data_full)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-24.153	-8.364	-2.366	5.668	68.934

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	20.648	1.051	19.642	< 2e-16 ***
regiondors	1.255	1.481	0.847	0.397
regionhead	7.646	1.481	5.161	3.20e-07 ***
regionmite	6.578	1.492	4.409	1.20e-05 ***
regionvent	10.615	1.481	7.166	1.98e-12 ***

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.39 on 694 degrees of freedom
## Multiple R-squared:  0.09502,    Adjusted R-squared:  0.0898
## F-statistic: 18.22 on 4 and 694 DF,  p-value: 3.059e-14
```

Also get pairwise statistics:

```
# one-way ANOVA
CEWL_region_aov <- aov(data = CEWL_data_full,
                      TEWL_g_m2h ~ region)
# post-hoc pairwise analysis
TukeyHSD(CEWL_region_aov)
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = TEWL_g_m2h ~ region, data = CEWL_data_full)
##
## $region
##          diff          lwr          upr      p adj
## dors-dewl  1.254852 -2.79651839  5.306223 0.9156911
## head-dewl  7.645739  3.59436813 11.697109 0.0000032
## mite-dewl   6.578364  2.49773783 10.658990 0.0001172
## vent-dewl 10.615491  6.56411991 14.666861 0.0000000
## head-dors   6.390887  2.35401096 10.427762 0.0001662
## mite-dors   5.323511  1.25727637  9.389746 0.0033630
## vent-dors   9.360638  5.32376273 13.397514 0.0000000
## mite-head  -1.067375 -5.13361016  2.998860 0.9523881
## vent-head   2.969752 -1.06712379  7.006627 0.2612293
## vent-mite   4.037127 -0.02910817  8.103362 0.0527256
```

CEWL ~ Osmolality

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = osmolality_mmol_kg,
                y = TEWL_g_m2h,
                color = region
                ),
            size = 1,
            alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = osmolality_mmol_kg,
                  y = TEWL_g_m2h,
                  color = region
                  ),
            formula = y ~ x,
            method = "lm",
            se = F,
            size = 1.6,
            alpha = 1 ) +
  theme_classic() +
  xlab("Osmolality") +
```

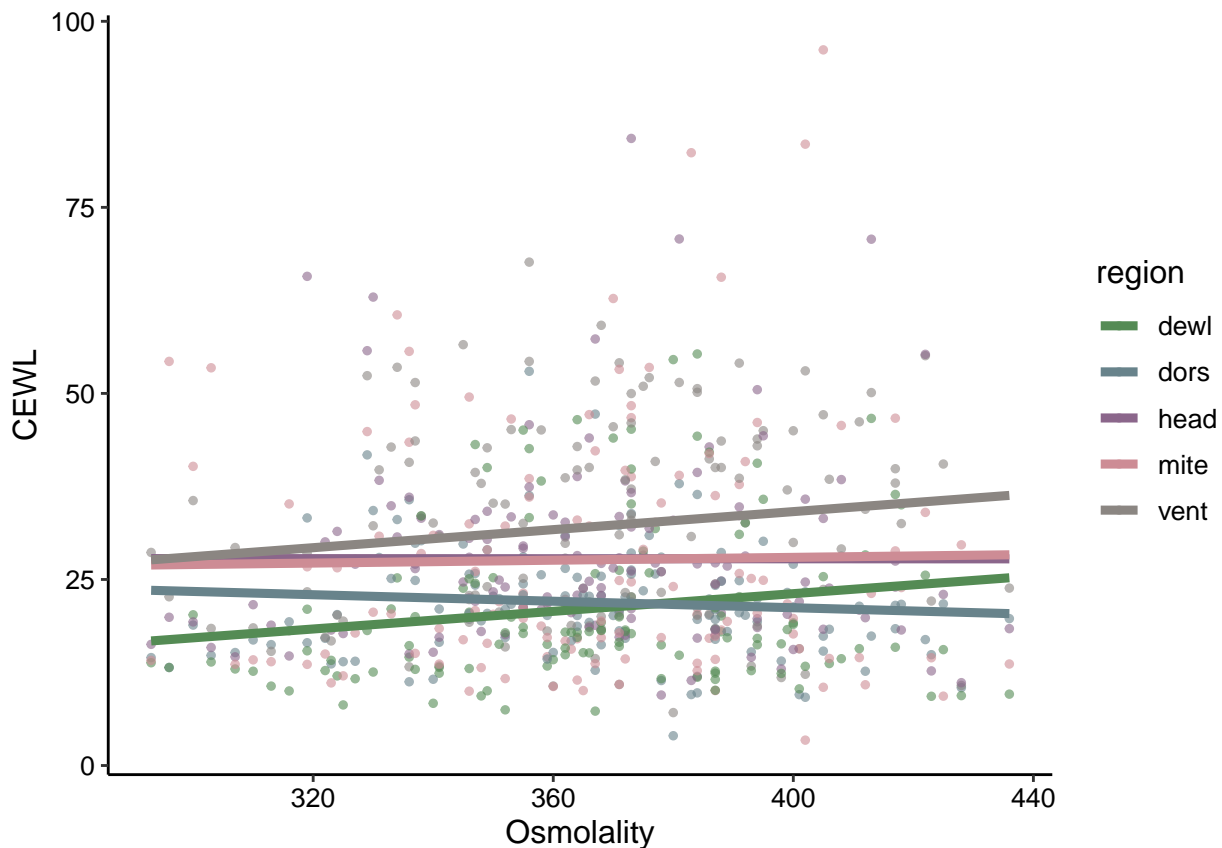
```

ylab("CEWL") +
theme(text = element_text(color = "black",
                           family = "sans",
                           size = 12),
      axis.text = element_text(color = "black",
                                family = "sans",
                                size = 10),
      legend.text.align = 0,
)

```

```
## Warning: Removed 49 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 49 rows containing missing values (geom_point).
```



```

# lm
CEWL_lm2 <- lm(TEWL_g_m2h ~ region + osmolality_mmol_kg,
               data = CEWL_data_full)
summary(CEWL_lm2)

```

```

##
## Call:
## lm(formula = TEWL_g_m2h ~ region + osmolality_mmol_kg, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.283  -8.587  -2.447   5.805  67.689
##
## Coefficients:

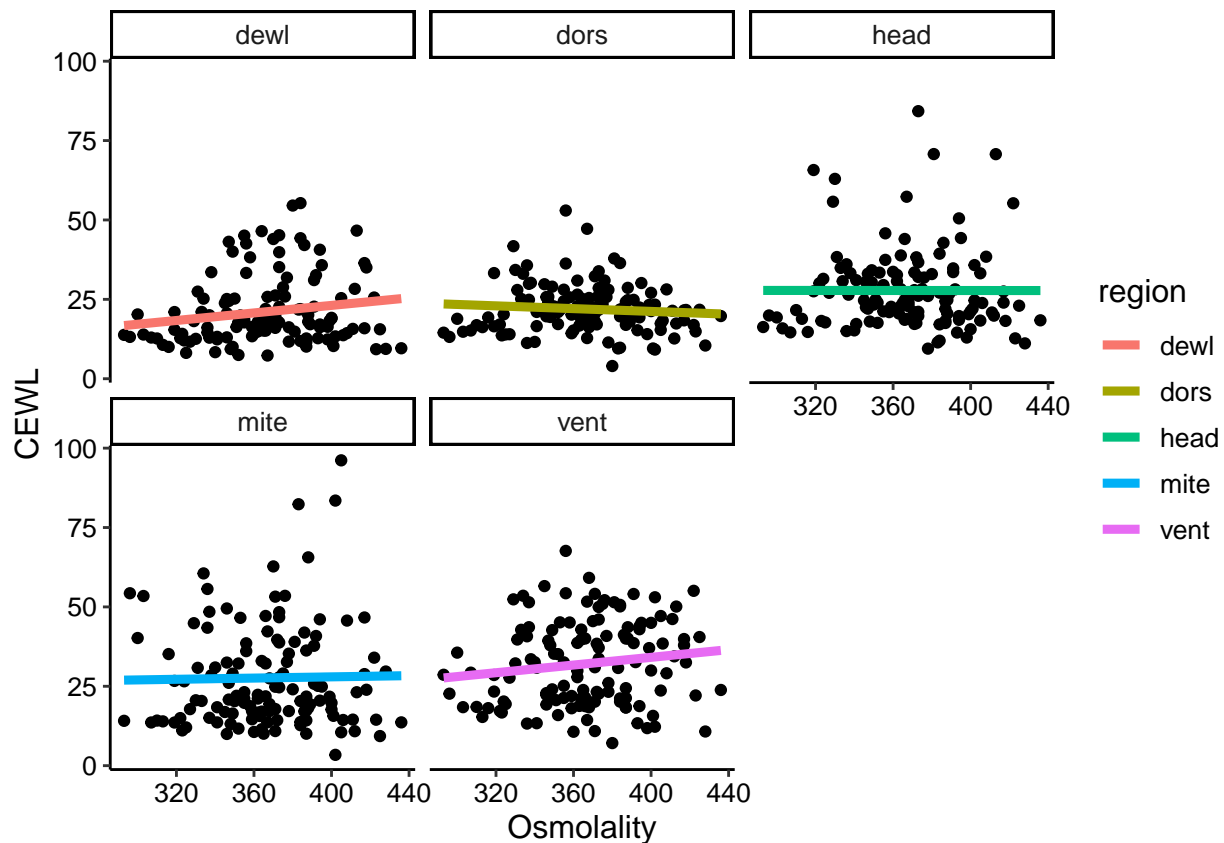
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.25915    5.96986   2.221  0.0267 *
## regiondors      0.80956    1.54206   0.525  0.5998
## regionhead      6.67027    1.54206   4.326 1.76e-05 ***
## regionmite      6.54184    1.55099   4.218 2.82e-05 ***
## regionvent     10.99851    1.54206   7.132 2.66e-12 ***
## osmolality_mmol_kg 0.02141    0.01601   1.337  0.1816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.43 on 644 degrees of freedom
## (49 observations deleted due to missingness)
## Multiple R-squared:  0.1011, Adjusted R-squared:  0.0941
## F-statistic: 14.48 on 5 and 644 DF,  p-value: 1.836e-13
```

```
# Facet ggplot
ggplot(aes(osmolality_mmol_kg, TEWL_g_m2h), data = CEWL_data_full) +
  geom_point() +
  stat_smooth(aes(x = osmolality_mmol_kg,
                  y = TEWL_g_m2h,
                  color = region
                  ),
              formula = y ~ x,
              method = "lm",
              se = F,
              size = 1.6,
              alpha = 1 )+
  theme_classic() +
  xlab("Osmolality") +
  ylab("CEWL") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )+
  facet_wrap(~ region) # create a facet for each body region
```

```
## Warning: Removed 49 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 49 rows containing missing values (geom_point).
```



CEWL ~ Hematocrit

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = hematocrit_percent,
                 y = TEWL_g_m2h,
                 color = region
                 ),
            size = 1,
            alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = hematocrit_percent,
                  y = TEWL_g_m2h,
                  color = region
                  ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1) +
  theme_classic() +
  xlab("Hematocrit") +
  ylab("CEWL") +
  theme(text = element_text(color = "black",
                             family = "sans",
```



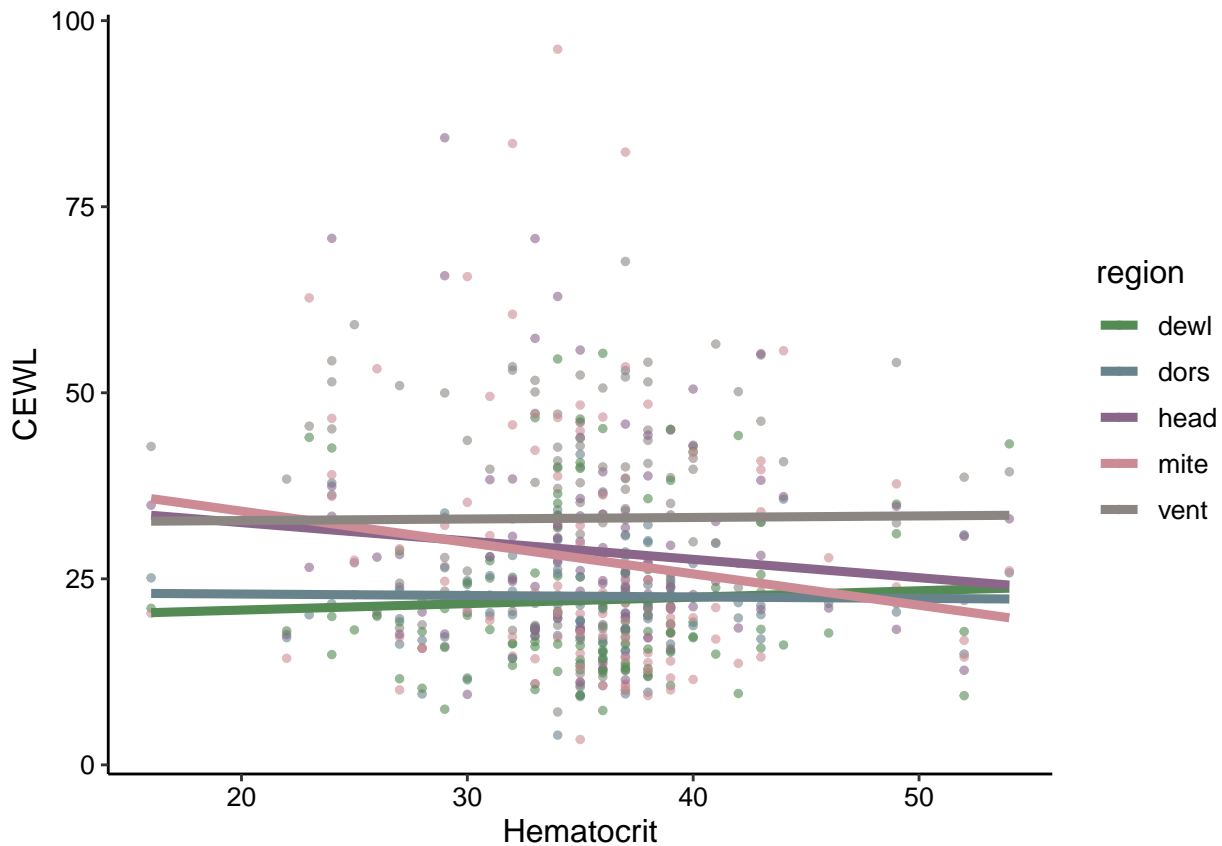
```

        size = 12),
    axis.text = element_text(color = "black",
                             family = "sans",
                             size = 10),
    legend.text.align = 0,
)

```

```
## Warning: Removed 119 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 119 rows containing missing values (geom_point).
```



```

# lm
CEWL_lm3 <- lm(TEWL_g_m2h ~ region * hematocrit_percent,
               data = CEWL_data_full)
summary(CEWL_lm3)

```

```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region * hematocrit_percent, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.008  -8.565  -2.311   5.873  67.969
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    19.06834     7.02497   2.714  0.00684 **
## regiondors      4.28061     9.92716   0.431  0.66648
```

```
## regionhead          18.41959    9.92310    1.856    0.06394 .
## regionmite          23.45205    9.94608    2.358    0.01871 *
## regionvent          13.34360    9.91989    1.345    0.17912
## hematocrit_percent    0.08656    0.19541    0.443    0.65795
## regiondors:hematocrit_percent -0.10618    0.27640   -0.384    0.70100
## regionhead:hematocrit_percent -0.33336    0.27608   -1.207    0.22775
## regionmite:hematocrit_percent -0.50803    0.27719   -1.833    0.06736 .
## regionvent:hematocrit_percent -0.06579    0.27609   -0.238    0.81173
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.6 on 570 degrees of freedom
## (119 observations deleted due to missingness)
## Multiple R-squared:  0.1063, Adjusted R-squared:  0.09216
## F-statistic: 7.531 on 9 and 570 DF,  p-value: 1.86e-10
```

CEWL ~ Cloacal Temperature

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = cloacal_temp_C,
                 y = TEWL_g_m2h,
                 color = region
                 ),
            size = 1,
            alpha = 0.4) +
  stat_smooth(aes(x = cloacal_temp_C,
                  y = TEWL_g_m2h,
                  color = region
                  ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1) +
  theme_classic() +
  xlab("Cloacal Temperature (°C)") +
  ylab(bquote('CEWL (g / *m^2~h*)')) +
  #annotate("text", x = 1, y = 65, label = "a", size = 6) +
  scale_color_brewer(palette = "Set2",
                    labels = c("Dewlap", "Dorsum", "Head",
                               "Mite Patch", "Ventrums"),
                    name = "") +
  ylim(1, 100) +
  xlim(20, 28) +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 18),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 14),
        legend.text = element_text(color = "black",
                                    family = "sans",
                                    size = 12),
```

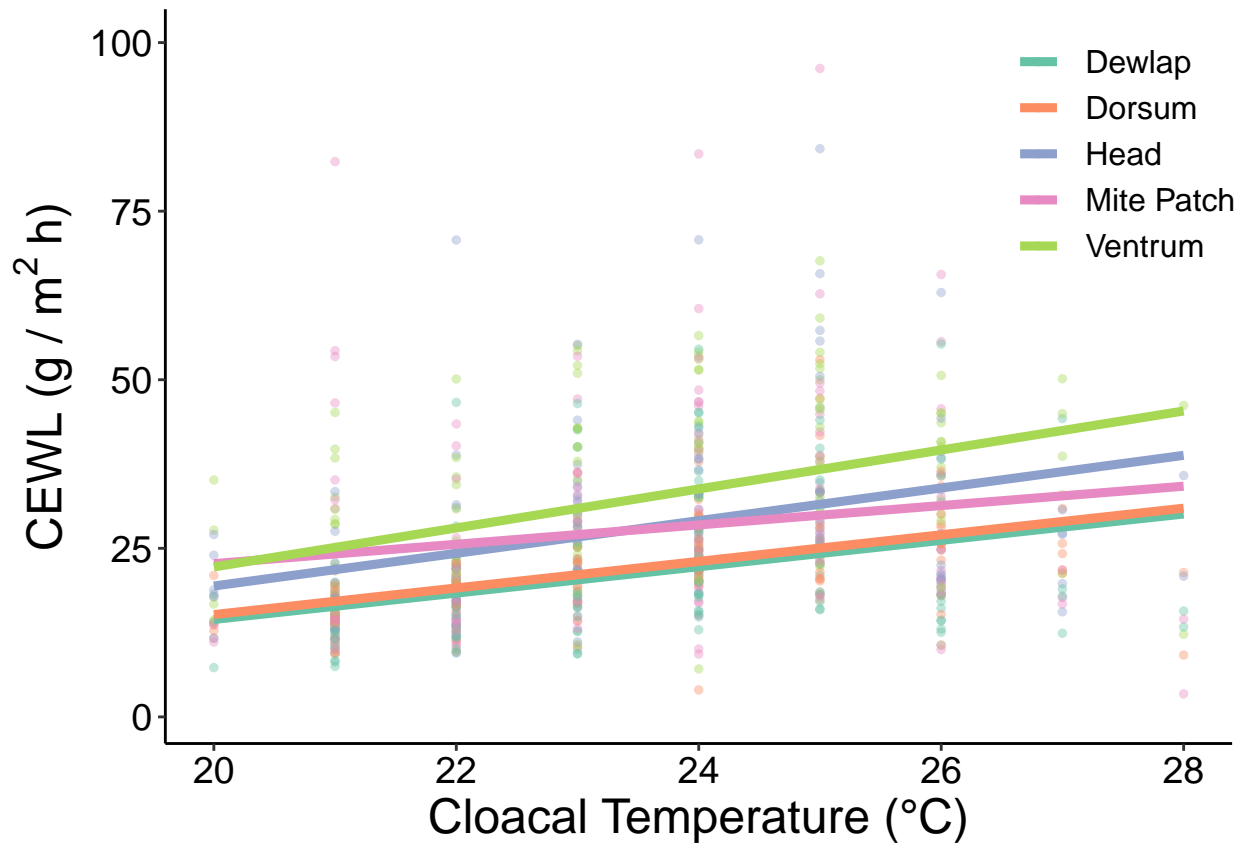
```

    legend.text.align = 0,
    legend.position = c(0.9, 0.85)
  #) +
  #guides(color = guide_legend(nrow = 2, byrow = TRUE)
    ) -> CEWL_ctype_fig
CEWL_ctype_fig

```

```
## Warning: Removed 49 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 49 rows containing missing values (geom_point).
```



```

# export figure
#ggsave(filename = "CEWL_ctype_fig.tiff",
  #      plot = CEWL_ctype_fig,
  #      path = "./final_figures",
  #      device = "tiff",
  #      dpi = 1200,
  #      width = 6, height = 4)

```

```

CEWL_lm4 <- lm(TEWL_g_m2h ~ region + cloacal_temp_C,
  data = CEWL_data_full)
summary(CEWL_lm4)

```

```

##
## Call:
## lm(formula = TEWL_g_m2h ~ region + cloacal_temp_C, data = CEWL_data_full)
##
## Residuals:

```

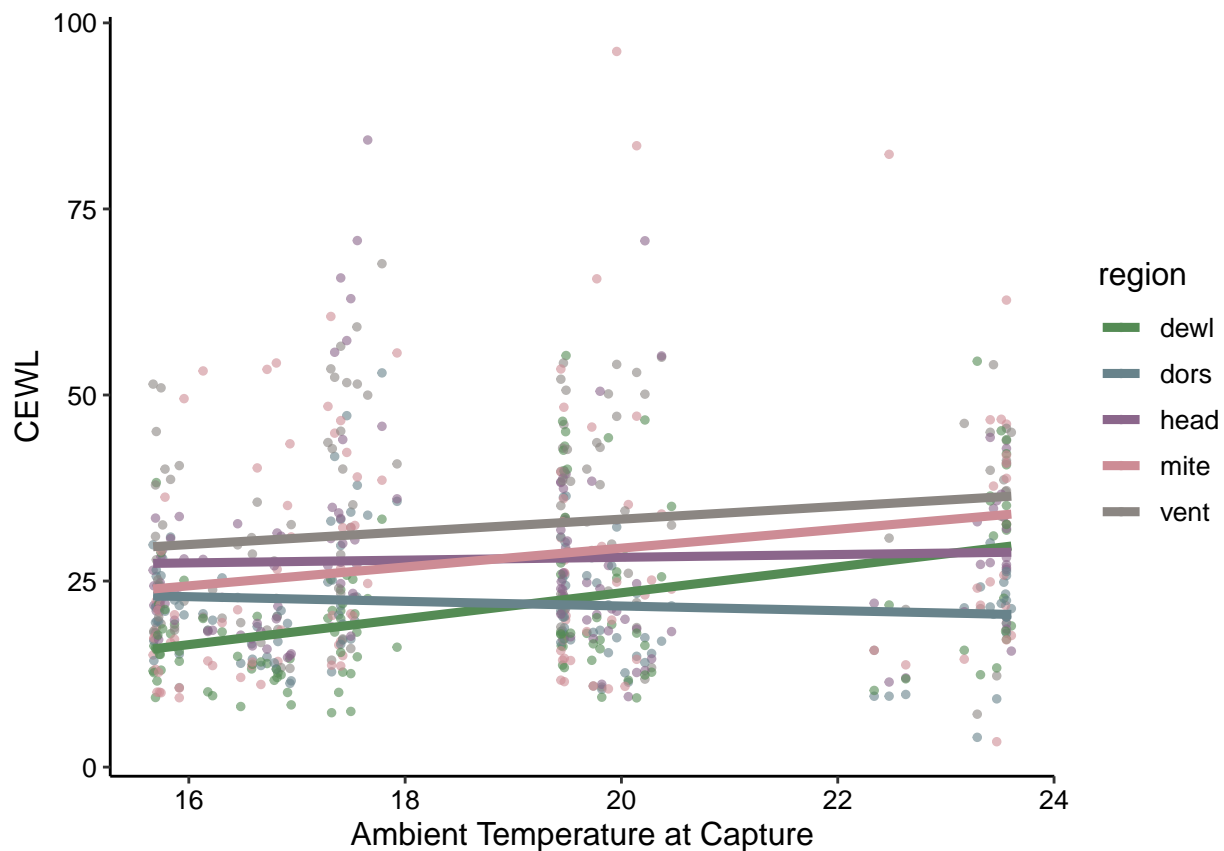
```
##      Min      1Q  Median      3Q      Max
## -34.079 -7.557 -1.903   5.192  65.103
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -29.0691     5.8706  -4.952 9.41e-07 ***
## regiondors     0.7740     1.4630   0.529  0.597
## regionhead     6.5538     1.4630   4.480 8.84e-06 ***
## regionmite     6.5272     1.4714   4.436 1.08e-05 ***
## regionvent    10.9708     1.4630   7.499 2.14e-13 ***
## cloacal_temp_C  2.1439     0.2471   8.675 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.79 on 644 degrees of freedom
## (49 observations deleted due to missingness)
## Multiple R-squared:  0.1925, Adjusted R-squared:  0.1862
## F-statistic: 30.71 on 5 and 644 DF,  p-value: < 2.2e-16
```

CEWL ~ Capture Temperature

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = temp_C_interpol,
                 y = TEWL_g_m2h,
                 color = region
                 ),
            size = 1,
            alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = temp_C_interpol,
                  y = TEWL_g_m2h,
                  color = region
                  ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1 ) +
  theme_classic() +
  xlab("Ambient Temperature at Capture") +
  ylab("CEWL") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                  family = "sans",
                                  size = 10),
        legend.text.align = 0,
  )
```

```
## Warning: Removed 59 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 59 rows containing missing values (geom_point).
```



```
# lm: CEWL ~ region + capture temp
CEWL_lm5 <- lm(TEWL_g_m2h ~ region * temp_C_interpol,
               data = CEWL_data_full)
summary(CEWL_lm5)
```

```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region * temp_C_interpol, data = CEWL_data_full)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-30.410	-8.223	-1.995	5.634	66.793

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-11.4129	8.0818	-1.412	0.158394
regiondors	39.3213	11.3626	3.461	0.000575 ***
regionhead	35.8369	11.4268	3.136	0.001791 **
regionmite	15.4923	11.3692	1.363	0.173478
regionvent	27.6451	11.4268	2.419	0.015831 *
temp_C_interpol	1.7418	0.4270	4.079	5.11e-05 ***
regiondors:temp_C_interpol	-2.0550	0.5997	-3.427	0.000650 ***
regionhead:temp_C_interpol	-1.5535	0.6036	-2.574	0.010281 *
regionmite:temp_C_interpol	-0.4746	0.6004	-0.791	0.429509
regionvent:temp_C_interpol	-0.8876	0.6036	-1.471	0.141900

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

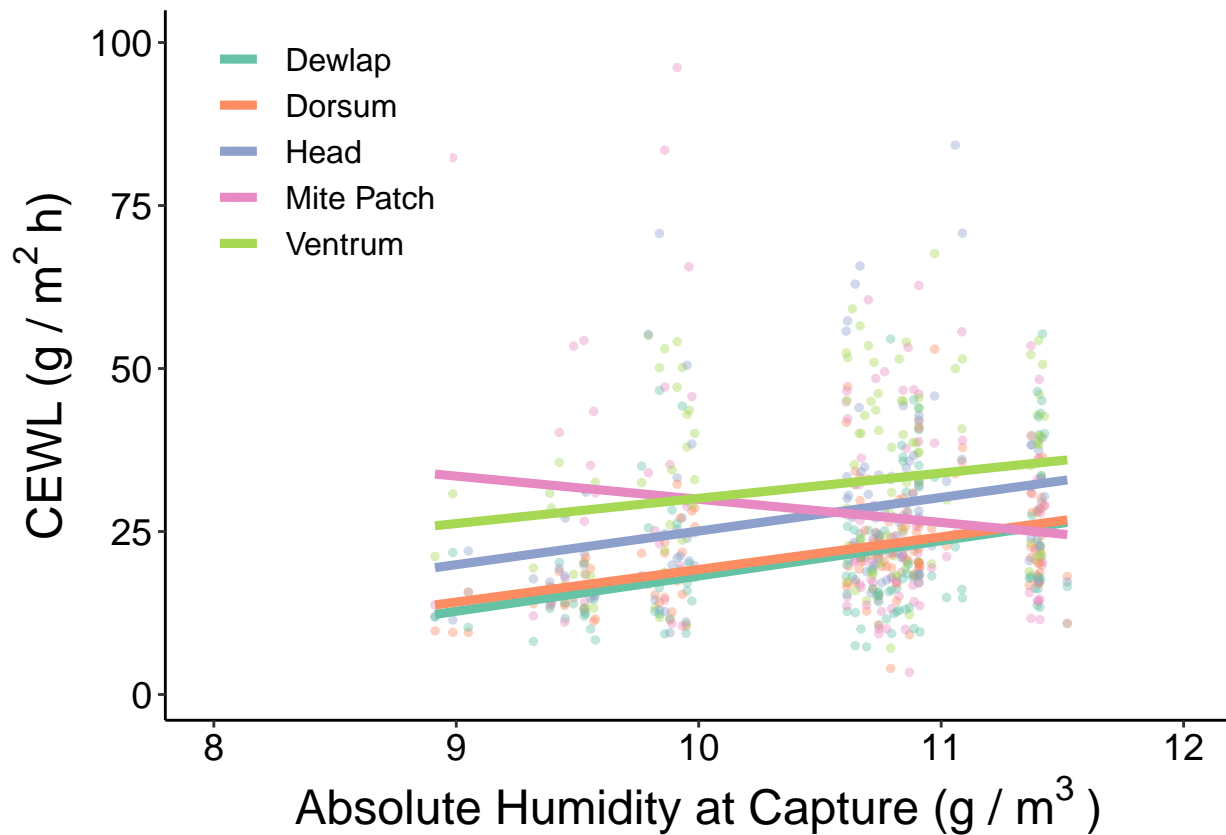
```
##
## Residual standard error: 12.22 on 630 degrees of freedom
## (59 observations deleted due to missingness)
## Multiple R-squared: 0.1406, Adjusted R-squared: 0.1283
## F-statistic: 11.45 on 9 and 630 DF, p-value: < 2.2e-16
```

CEWL ~ Capture Humidity

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_jitter(aes(x = abs_humidity_g_m3_interpol,
                  y = TEWL_g_m2h,
                  color = region
                  ),
              size = 1,
              alpha = 0.4) +
  stat_smooth(aes(x = abs_humidity_g_m3_interpol,
                  y = TEWL_g_m2h,
                  color = region
                  ),
              formula = y ~ x,
              method = "lm",
              se = F,
              size = 1.6,
              alpha = 1 ) +
  theme_classic() +
  xlab('Absolute Humidity at Capture (g / '*m^3~')') +
  ylab(bquote('CEWL (g / '*m^2~h*')')) +
  #annotate("text", x = 1, y = 65, label = "a", size = 6) +
  scale_color_brewer(palette = "Set2",
                     labels = c("Dewlap", "Dorsum",
                                "Head", "Mite Patch", "Ventrums"),
                     name = "") +
  ylim(1, 100) +
  xlim(8, 12) +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 18),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 14),
        legend.text = element_text(color = "black",
                                    family = "sans",
                                    size = 12),
        legend.text.align = 0,
        legend.position = c(0.15, 0.85)
        ) -> CEWL_abshum_fig
CEWL_abshum_fig
```

```
## Warning: Removed 59 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 59 rows containing missing values (geom_point).
```



```
# export figure
#ggsave(filename = "CEWL_abshum_fig.tiff",
#       plot = CEWL_abshum_fig,
#       path = "./final_figures",
#       device = "tiff",
#       dpi = 1200,
#       width = 6, height = 4)
```

```
CEWL_lm6 <- lm(TEWL_g_m2h ~ region * abs_humidity_g_m3_interpol,
               data = CEWL_data_full)
summary(CEWL_lm6)
```

```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region * abs_humidity_g_m3_interpol,
##     data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.050  -7.874  -2.004   5.155  65.902
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -35.9192    17.6081  -2.040  0.04177 *
## regiondors      5.0996    24.8468   0.205  0.83745
## regionhead     9.4282    24.8090   0.380  0.70405
## regionmite    101.4306    25.0192   4.054 5.66e-05 ***
## regionvent     27.2874    24.8090   1.100  0.27179
```

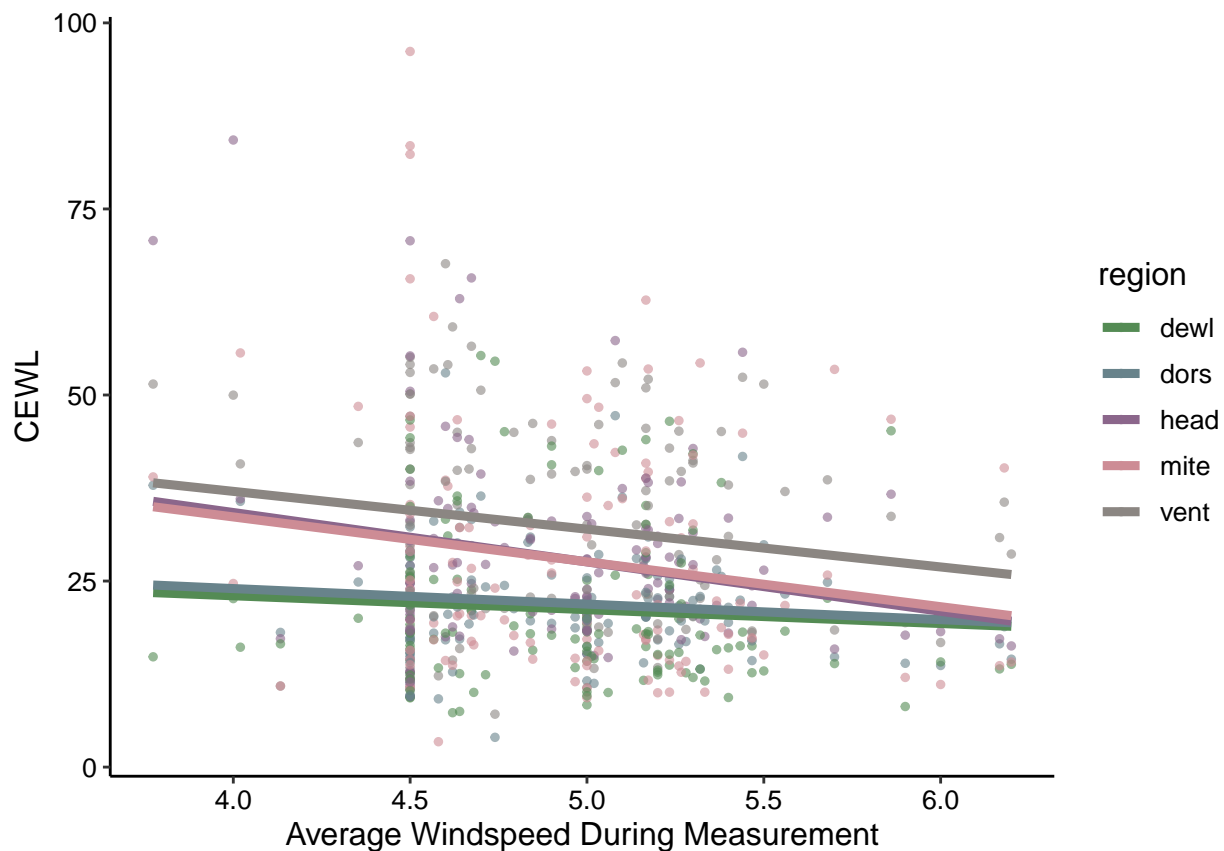
```
## abs_humidity_g_m3_interpol      5.4090      1.6627      3.253      0.00120 **
## regiondors:abs_humidity_g_m3_interpol -0.4091      2.3465     -0.174      0.86167
## regionhead:abs_humidity_g_m3_interpol -0.2532      2.3438     -0.108      0.91399
## regionmite:abs_humidity_g_m3_interpol -8.9659      2.3611     -3.797      0.00016 ***
## regionvent:abs_humidity_g_m3_interpol -1.5364      2.3438     -0.656      0.51237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.14 on 630 degrees of freedom
## (59 observations deleted due to missingness)
## Multiple R-squared:  0.1522, Adjusted R-squared:  0.1401
## F-statistic: 12.57 on 9 and 630 DF,  p-value: < 2.2e-16
```

CEWL ~ Wind Speed

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = Wind_mph_interpol,
                 y = TEWL_g_m2h,
                 color = region
                 ),
             size = 1,
             alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = Wind_mph_interpol,
                  y = TEWL_g_m2h,
                  color = region
                  ),
              formula = y ~ x,
              method = "lm",
              se = F,
              size = 1.6,
              alpha = 1 ) +
  theme_classic() +
  xlab("Average Windspeed During Measurement") +
  ylab("CEWL") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```

```
## Warning: Removed 59 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 59 rows containing missing values (geom_point).
```

```
# lm: CEWL ~ region + aquaflux measurement RH
CEWL_lm7 <- lm(TEWL_g_m2h ~ region + Wind_mph_interpol,
               data = CEWL_data_full)
summary(CEWL_lm7)
```

```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region + Wind_mph_interpol, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.075  -8.470  -2.166   5.884  66.328
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    42.7246     5.5870   7.647 7.65e-14 ***
## regiondors       0.7384     1.5405   0.479  0.632
## regionhead      6.6732     1.5405   4.332 1.72e-05 ***
## regionmite      6.6311     1.5495   4.279 2.16e-05 ***
## regionvent     10.9815     1.5405   7.129 2.77e-12 ***
## Wind_mph_interpol -4.3387     1.1071  -3.919 9.87e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.32 on 634 degrees of freedom
## (59 observations deleted due to missingness)
## Multiple R-squared:  0.1204, Adjusted R-squared:  0.1135
```

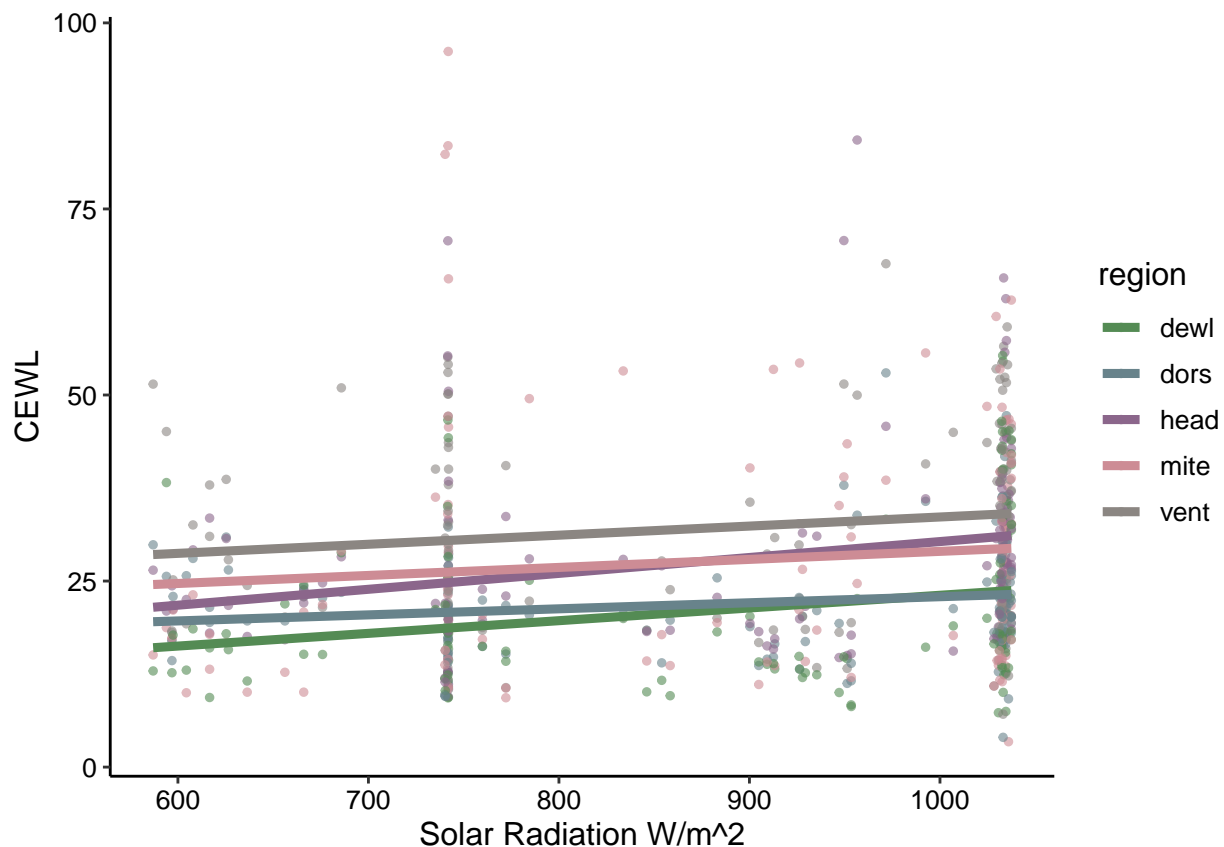
```
## F-statistic: 17.36 on 5 and 634 DF, p-value: 3.977e-16
```

CEWL ~ Solar Rad

```
CEWL_data_full %>%  
  ggplot(data = .) +  
    geom_point(aes(x = Solar_rad_Wm2_interpol,  
                  y = TEWL_g_m2h,  
                  color = region  
                  ),  
              size = 1,  
              alpha = 0.6) +  
    scale_colour_manual(values = c("palegreen4", "lightblue4",  
                                   "plum4", "lightpink3", "seashell4")) +  
    stat_smooth(aes(x = Solar_rad_Wm2_interpol,  
                   y = TEWL_g_m2h,  
                   color = region  
                   ),  
               formula = y ~ x,  
               method = "lm",  
               se = F,  
               size = 1.6,  
               alpha = 1 ) +  
    theme_classic() +  
    xlab("Solar Radiation W/m^2") +  
    ylab("CEWL") +  
    theme(text = element_text(color = "black",  
                              family = "sans",  
                              size = 12),  
          axis.text = element_text(color = "black",  
                                    family = "sans",  
                                    size = 10),  
          legend.text.align = 0,  
    )
```

```
## Warning: Removed 59 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 59 rows containing missing values (geom_point).
```



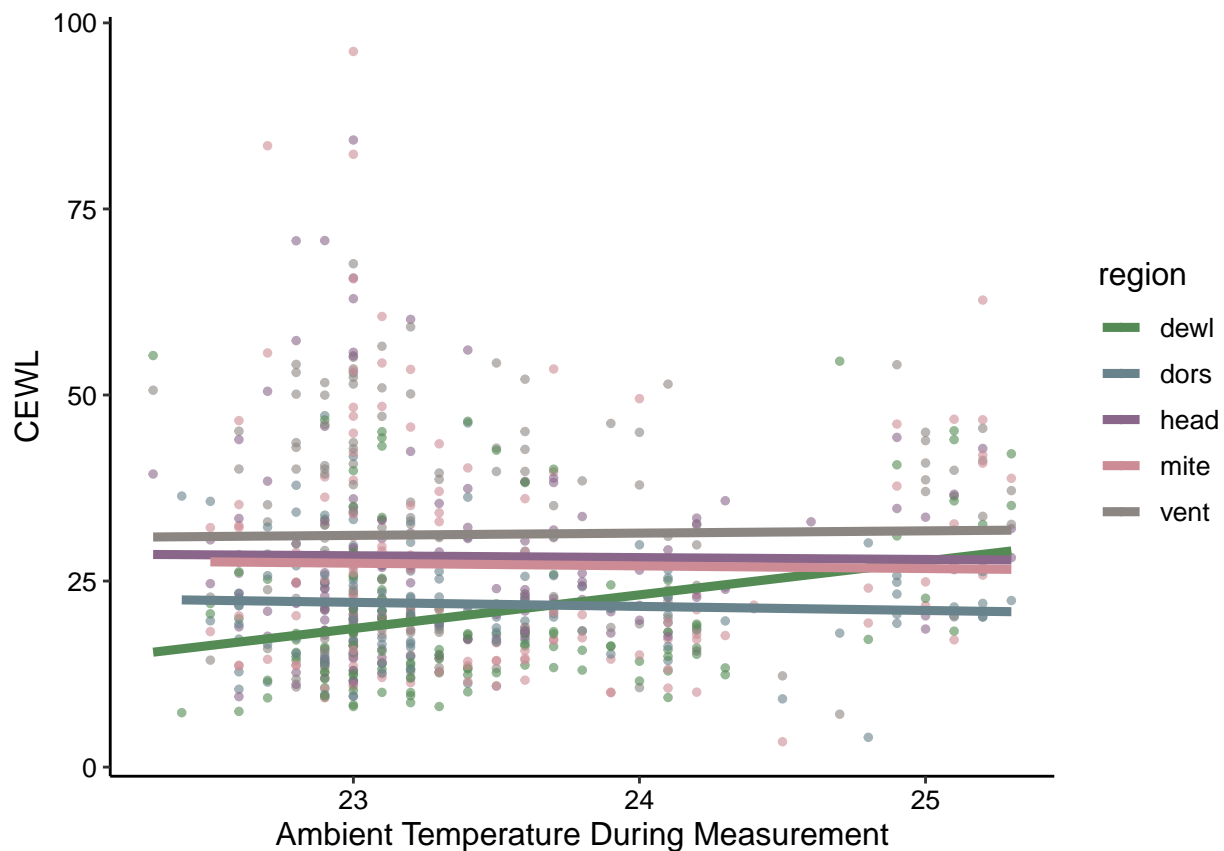
```
# lm: CEWL ~ region + aquaflux measurement RH
CEWL_lm8 <- lm(TEWL_g_m2h ~ region + Solar_rad_Wm2_interpol,
               data = CEWL_data_full)
summary(CEWL_lm8)
```

```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region + Solar_rad_Wm2_interpol, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.144  -8.072  -2.283   5.790  70.440
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.829372   3.005574   2.938  0.00343 **
## regiondors      0.766911   1.535357   0.500  0.61760
## regionhead     6.735880   1.535375   4.387 1.35e-05 ***
## regionmite     6.564058   1.544438   4.250 2.46e-05 ***
## regionvent    11.044136   1.535375   7.193 1.79e-12 ***
## Solar_rad_Wm2_interpol 0.013920   0.003138   4.436 1.08e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.28 on 634 degrees of freedom
## (59 observations deleted due to missingness)
## Multiple R-squared:  0.1262, Adjusted R-squared:  0.1194
```

F-statistic: 18.32 on 5 and 634 DF, p-value: < 2.2e-16

CEWL ~ Measurement Temperature

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = ambient_temp_C,
                 y = TEWL_g_m2h,
                 color = region
                 ),
            size = 1,
            alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = ambient_temp_C,
                  y = TEWL_g_m2h,
                  color = region
                  ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1 ) +
  theme_classic() +
  xlab("Ambient Temperature During Measurement") +
  ylab("CEWL") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```



```
# lm: CEWL ~ region + aquaflux measurement temp
CEWL_lm9 <- lm(TEWL_g_m2h ~ region * ambient_temp_C,
               data = CEWL_data_full)
summary(CEWL_lm9)
```

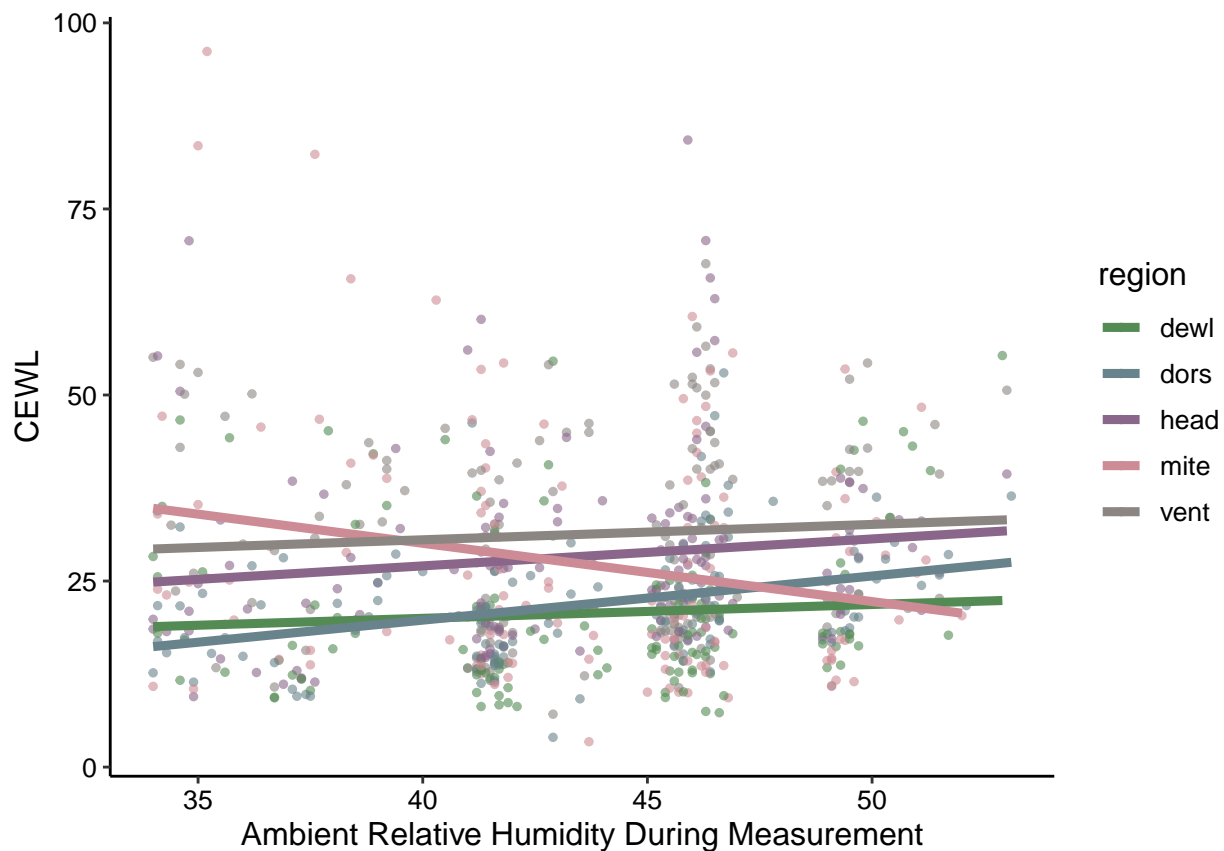
```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region * ambient_temp_C, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.542  -8.438  -2.262   5.727  68.764
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -85.690     35.629  -2.405  0.01643 *
## regiondors      120.409     49.906   2.413  0.01609 *
## regionhead      119.786     49.830   2.404  0.01648 *
## regionmite      121.357     49.735   2.440  0.01493 *
## regionvent      109.812     49.651   2.212  0.02732 *
## ambient_temp_C    4.535      1.519   2.986  0.00293 **
## regiondors:ambient_temp_C  -5.082      2.128  -2.388  0.01721 *
## regionhead:ambient_temp_C  -4.783      2.125  -2.250  0.02474 *
## regionmite:ambient_temp_C  -4.894      2.119  -2.310  0.02120 *
## regionvent:ambient_temp_C  -4.230      2.117  -1.998  0.04614 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 12.36 on 689 degrees of freedom
## Multiple R-squared:  0.1069, Adjusted R-squared:  0.09525
## F-statistic: 9.165 on 9 and 689 DF,  p-value: 3.517e-13
```

CEWL ~ Measurement Humidity

Very interesting relationship! Mite patch CEWL decreases as ambient humidity increases, but every other location appears to increase. In this case, an interaction term is warranted.

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = ambient_RH_percent,
                 y = TEWL_g_m2h,
                 color = region
                 ),
            size = 1,
            alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = ambient_RH_percent,
                  y = TEWL_g_m2h,
                  color = region
                  ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1 ) +
  theme_classic() +
  xlab("Ambient Relative Humidity During Measurement") +
  ylab("CEWL") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```



```
# lm: CEWL ~ region + aquaflox measurement RH
CEWL_lm10 <- lm(TEWL_g_m2h ~ region * ambient_RH_percent,
  data = CEWL_data_full)
summary(CEWL_lm10)
```

```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region * ambient_RH_percent, data = CEWL_data_full)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-24.032	-8.244	-2.423	6.029	62.330

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	12.558174	10.167860	1.235	0.217220
regiondors	-16.537848	14.249455	-1.161	0.246209
regionhead	-0.006792	14.232189	0.000	0.999619
regionmite	48.838420	14.534963	3.360	0.000822 ***
regionvent	9.732277	14.199427	0.685	0.493323
ambient_RH_percent	0.185765	0.232264	0.800	0.424100
regiondors:ambient_RH_percent	0.407357	0.325153	1.253	0.210697
regionhead:ambient_RH_percent	0.176350	0.325357	0.542	0.587979
regionmite:ambient_RH_percent	-0.968911	0.331720	-2.921	0.003605 **
regionvent:ambient_RH_percent	0.020570	0.324547	0.063	0.949481

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

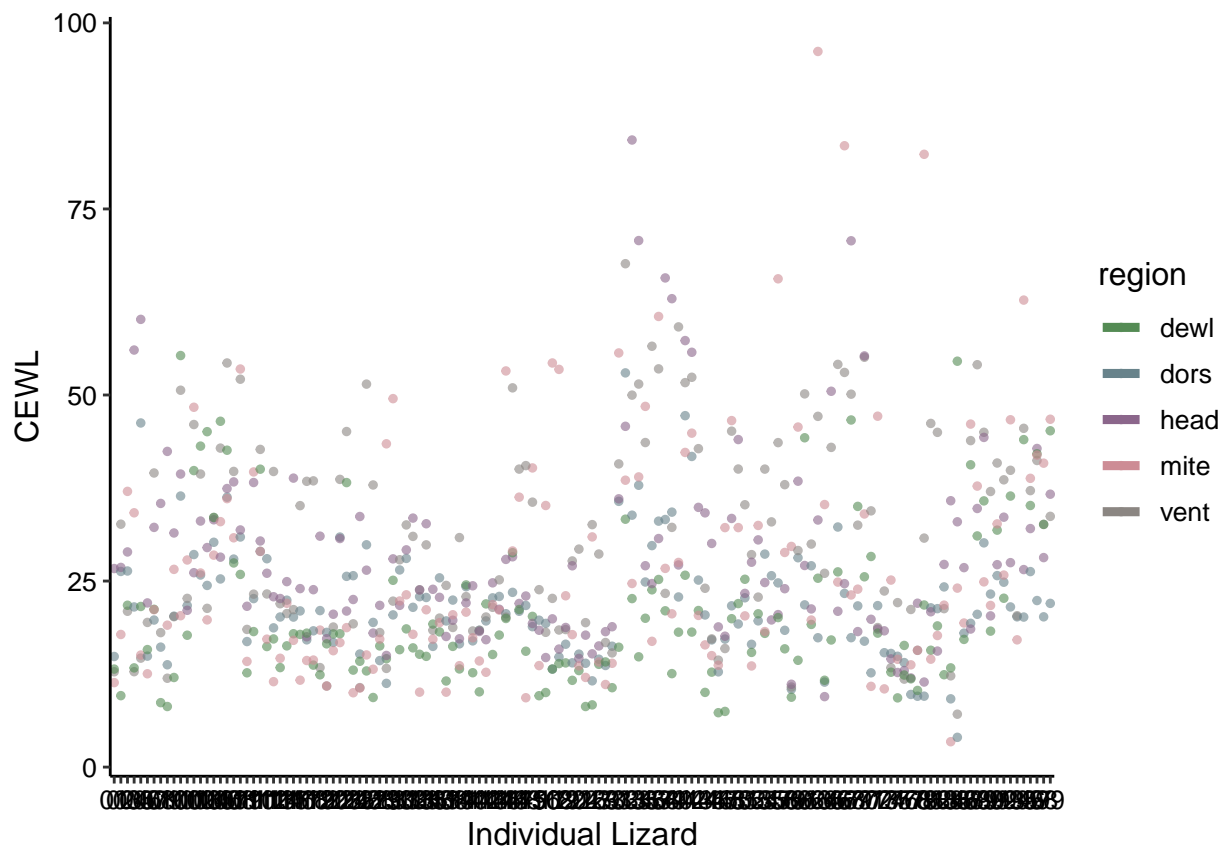
```
##
## Residual standard error: 12.25 on 689 degrees of freedom
## Multiple R-squared:  0.1227, Adjusted R-squared:  0.1112
## F-statistic: 10.7 on 9 and 689 DF,  p-value: 1.218e-15
```

CEWL ~ Individual

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = individual_ID,
                 y = TEWL_g_m2h,
                 color = region
                 ),
            size = 1,
            alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = individual_ID,
                  y = TEWL_g_m2h,
                  color = region
                  ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1 ) +
  theme_classic() +
  xlab("Individual Lizard") +
  ylab("CEWL") +

  # just to get a better look
  # ylim(5, 40) +

  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),
        axis.text = element_text(color = "black",
                                   family = "sans",
                                   size = 10),
        legend.text.align = 0,
  )
```

```
# lm
CEWL_lm11 <- lm(TEWL_g_m2h ~ region + individual_ID,
  data = CEWL_data_full)
summary(CEWL_lm11)
```

```
##
## Call:
## lm(formula = TEWL_g_m2h ~ region + individual_ID, data = CEWL_data_full)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-22.805	-5.066	-0.605	3.882	50.951

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	10.6186	4.6593	2.279	0.023047	*
regiondors	1.2460	1.2314	1.012	0.312063	
regionhead	7.5112	1.2299	6.107	1.91e-09	***
regionmite	6.5149	1.2421	5.245	2.23e-07	***
regionvent	10.5050	1.2312	8.532	< 2e-16	***
individual_ID02	6.8620	6.4959	1.056	0.291264	
individual_ID03	11.2260	6.4959	1.728	0.084516	.
individual_ID04	11.8020	6.4959	1.817	0.069782	.
individual_ID05	15.7600	6.4959	2.426	0.015579	*
individual_ID06	1.1740	6.4959	0.181	0.856646	
individual_ID07	10.9940	6.4959	1.692	0.091122	.
individual_ID08	4.1359	6.8927	0.600	0.548729	

## individual_ID09	3.2900	6.4959	0.506	0.612725	
## individual_ID10	6.3380	6.4959	0.976	0.329642	
## individual_ID100	24.6420	6.4959	3.793	0.000165	***
## individual_ID101	6.4260	6.4959	0.989	0.322977	
## individual_ID102	22.0060	6.4959	3.388	0.000755	***
## individual_ID103	17.7180	6.4959	2.728	0.006583	**
## individual_ID104	12.2440	6.4959	1.885	0.059969	.
## individual_ID105	16.0240	6.4959	2.467	0.013935	*
## individual_ID106	19.3860	6.4959	2.984	0.002967	**
## individual_ID107	25.5560	6.4959	3.934	9.41e-05	***
## individual_ID108	17.0740	6.4959	2.628	0.008816	**
## individual_ID109	23.0830	6.4959	3.553	0.000413	***
## individual_ID11	0.9960	6.4959	0.153	0.878196	
## individual_ID110	12.6240	6.4959	1.943	0.052477	.
## individual_ID111	18.4400	6.4959	2.839	0.004696	**
## individual_ID112	6.3660	6.4959	0.980	0.327512	
## individual_ID113	6.2300	6.4959	0.959	0.337943	
## individual_ID114	2.7780	6.4959	0.428	0.669068	
## individual_ID115	5.4820	6.4959	0.844	0.399079	
## individual_ID116	7.2460	6.4959	1.115	0.265131	
## individual_ID117	6.1460	6.4959	0.946	0.344491	
## individual_ID118	5.2920	6.4959	0.815	0.415612	
## individual_ID119	5.9640	6.4959	0.918	0.358956	
## individual_ID12	3.4740	6.4959	0.535	0.593003	
## individual_ID120	-1.0340	6.4959	-0.159	0.873587	
## individual_ID122	2.1560	6.4959	0.332	0.740088	
## individual_ID123	11.2120	6.4959	1.726	0.084903	.
## individual_ID124	13.9580	6.4959	2.149	0.032088	*
## individual_ID125	2.3280	6.4959	0.358	0.720195	
## individual_ID126	1.1020	6.4959	0.170	0.865351	
## individual_ID127	11.3820	6.4959	1.752	0.080296	.
## individual_ID128	3.7980	6.4959	0.585	0.559002	
## individual_ID129	1.7460	6.4959	0.269	0.788195	
## individual_ID13	3.7280	6.4959	0.574	0.566267	
## individual_ID130	13.2940	6.4959	2.047	0.041177	*
## individual_ID131	7.0620	6.4959	1.087	0.277444	
## individual_ID132	10.5240	6.4959	1.620	0.105780	
## individual_ID133	8.1840	6.4959	1.260	0.208246	
## individual_ID134	3.3780	6.4959	0.520	0.603256	
## individual_ID135	8.5000	6.4959	1.309	0.191240	
## individual_ID136	3.2040	6.4959	0.493	0.622042	
## individual_ID137	5.3940	6.4959	0.830	0.406687	
## individual_ID138	0.8740	6.4959	0.135	0.893019	
## individual_ID139	3.7500	6.4959	0.577	0.563979	
## individual_ID14	2.5180	6.4959	0.388	0.698439	
## individual_ID140	7.1280	6.4959	1.097	0.272983	
## individual_ID141	2.1580	6.4959	0.332	0.739856	
## individual_ID142	0.0960	6.4959	0.015	0.988214	
## individual_ID143	2.5460	6.4959	0.392	0.695253	
## individual_ID144	5.4120	6.4959	0.833	0.405123	
## individual_ID145	5.8540	6.4959	0.901	0.367881	
## individual_ID146	12.6000	6.4959	1.940	0.052925	.
## individual_ID147	16.3360	6.4959	2.515	0.012192	*
## individual_ID148	12.3080	6.4959	1.895	0.058648	.

## individual_ID149	6.2480	6.4959	0.962	0.336550	
## individual_ID15	11.0700	6.4959	1.704	0.088913	.
## individual_ID150	1.2660	6.4959	0.195	0.845548	
## individual_ID16	3.6880	6.4959	0.568	0.570439	
## individual_ID17	8.8720	6.4959	1.366	0.172560	
## individual_ID18	7.5120	6.4959	1.156	0.248007	
## individual_ID19	2.3800	6.4959	0.366	0.714217	
## individual_ID20	3.8700	6.4959	0.596	0.551578	
## individual_ID21	1.3540	6.4959	0.208	0.834962	
## individual_ID24	-1.5180	6.4959	-0.234	0.815315	
## individual_ID26	3.9620	6.4959	0.610	0.542162	
## individual_ID27	1.6940	6.4959	0.261	0.794359	
## individual_ID29	-1.0100	6.4959	-0.155	0.876497	
## individual_ID30	-0.7620	6.4959	-0.117	0.906661	
## individual_ID31	21.0780	6.4959	3.245	0.001246	**
## individual_ID32	31.8780	6.4959	4.907	1.22e-06	***
## individual_ID33	27.3060	6.4959	4.204	3.06e-05	***
## individual_ID34	27.0060	6.4959	4.157	3.73e-05	***
## individual_ID35	17.0280	6.4959	2.621	0.008999	**
## individual_ID36	14.5640	6.4959	2.242	0.025355	*
## individual_ID37	24.8320	6.4959	3.823	0.000147	***
## individual_ID38	18.2340	6.4959	2.807	0.005177	**
## individual_ID39	16.7400	6.4959	2.577	0.010224	*
## individual_ID40	15.1860	6.4959	2.338	0.019754	*
## individual_ID41	29.0760	6.4959	4.476	9.24e-06	***
## individual_ID42	26.8000	6.4959	4.126	4.27e-05	***
## individual_ID43	13.0740	6.4959	2.013	0.044635	*
## individual_ID44	5.2440	6.4959	0.807	0.419852	
## individual_ID45	2.6480	6.4959	0.408	0.683694	
## individual_ID46	-2.3720	6.4959	-0.365	0.715136	
## individual_ID47	2.3020	6.4959	0.354	0.723190	
## individual_ID48	17.5700	6.4959	2.705	0.007045	**
## individual_ID50	15.7140	6.4959	2.419	0.015882	*
## individual_ID51	9.6060	6.4959	1.479	0.139768	
## individual_ID52	4.5420	6.4959	0.699	0.484713	
## individual_ID53	9.4440	6.4959	1.454	0.146557	
## individual_ID54	10.1540	6.4959	1.563	0.118591	
## individual_ID55	9.1074	7.5076	1.213	0.225613	
## individual_ID57	19.1340	6.4959	2.946	0.003359	**
## individual_ID58	9.2320	6.4959	1.421	0.155820	
## individual_ID59	-1.4960	6.4959	-0.230	0.817944	
## individual_ID60	15.3640	6.4959	2.365	0.018365	*
## individual_ID61	20.3884	6.8927	2.958	0.003229	**
## individual_ID62	7.6660	6.4959	1.180	0.238455	
## individual_ID63	28.0760	6.4959	4.322	1.83e-05	***
## individual_ID64	3.0020	6.4959	0.462	0.644163	
## individual_ID65	20.2227	7.5076	2.694	0.007282	**
## individual_ID66	15.9060	6.4959	2.449	0.014650	*
## individual_ID67	29.0647	6.8927	4.217	2.90e-05	***
## individual_ID68	25.8240	6.4959	3.975	7.96e-05	***
## individual_ID69	10.4900	6.4959	1.615	0.106909	
## individual_ID70	21.5900	6.4959	3.324	0.000947	***
## individual_ID71	5.4480	6.4959	0.839	0.402008	
## individual_ID72	9.0560	6.4959	1.394	0.163843	

```

## individual_ID73    -0.1003      6.8927   -0.015  0.988390
## individual_ID74      0.4360      6.4959    0.067  0.946511
## individual_ID75    -1.0780      6.4959   -0.166  0.868256
## individual_ID76    -0.9610      6.5006   -0.148  0.882526
## individual_ID77    -2.0700      6.4959   -0.319  0.750102
## individual_ID78    -1.1300      6.4959   -0.174  0.861963
## individual_ID79    15.3960      6.4959    2.370  0.018125 *
## individual_ID81      7.9560      6.4959    1.225  0.221181
## individual_ID82      7.9280      6.4959    1.220  0.222808
## individual_ID83      5.6060      6.4959    0.863  0.388507
## individual_ID84    -0.9820      6.4959   -0.151  0.879895
## individual_ID85      8.7580      6.4959    1.348  0.178133
## individual_ID86      5.8934      6.8926    0.855  0.392907
## individual_ID87    17.9160      6.4959    2.758  0.006007 **
## individual_ID88    19.8600      6.4959    3.057  0.002341 **
## individual_ID89    20.2420      6.4959    3.116  0.001928 **
## individual_ID90      8.3440      6.4959    1.285  0.199503
## individual_ID91    15.8880      6.4959    2.446  0.014761 *
## individual_ID92    13.3280      6.4959    2.052  0.040663 *
## individual_ID93    18.6180      6.4959    2.866  0.004313 **
## individual_ID94      2.5028      7.5076    0.333  0.738981
## individual_ID95    24.0200      6.4959    3.698  0.000239 ***
## individual_ID96    18.1100      6.4959    2.788  0.005487 **
## individual_ID97    22.3080      6.4959    3.434  0.000639 ***
## individual_ID98    15.1060      6.4959    2.325  0.020407 *
## individual_ID99    21.0840      6.4959    3.246  0.001242 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.27 on 553 degrees of freedom
## Multiple R-squared:  0.5047, Adjusted R-squared:  0.3749
## F-statistic: 3.887 on 145 and 553 DF,  p-value: < 2.2e-16

```

CEWL ~ SVL

```

CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = SVL_mm,
                 y = TEWL_g_m2h,
                 color = region
                ),
            size = 1,
            alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = SVL_mm,
                  y = TEWL_g_m2h,
                  color = region
                 ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1 ) +

```

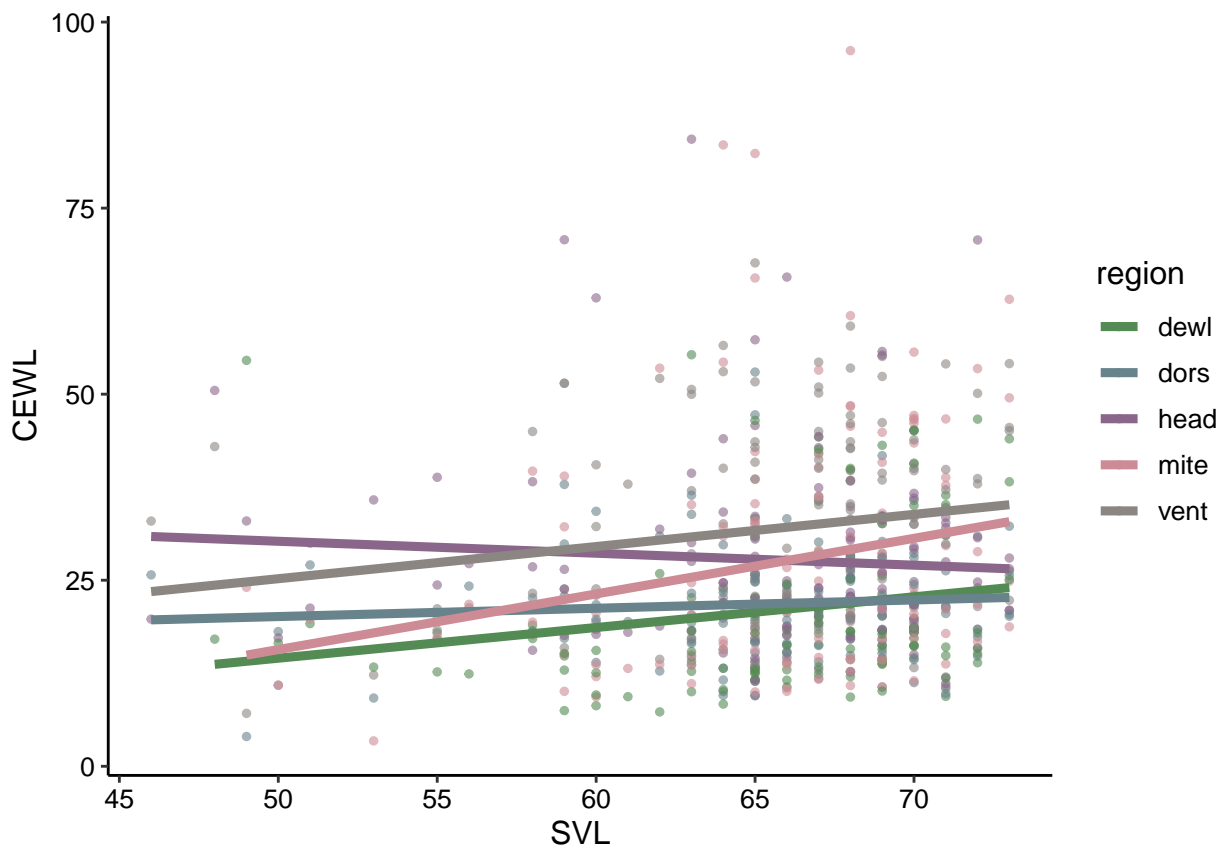
```

theme_classic() +
xlab("SVL") +
ylab("CEWL") +
theme(text = element_text(color = "black",
                           family = "sans",
                           size = 12),
      axis.text = element_text(color = "black",
                                family = "sans",
                                size = 10),
      legend.text.align = 0,
)

```

```
## Warning: Removed 44 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 44 rows containing missing values (geom_point).
```



```

# lm
CEWL_lm12 <- lm(TEWL_g_m2h ~ region + SVL_mm,
               data = CEWL_data_full)
summary(CEWL_lm12)

```

```
##
```

```
## Call:
```

```
## lm(formula = TEWL_g_m2h ~ region + SVL_mm, data = CEWL_data_full)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -22.778  -8.230  -2.224   5.569  67.956
```

```
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.8945     6.3736   0.297  0.76638
## regiondors   0.8336     1.5243   0.547  0.58465
## regionhead   6.7145     1.5244   4.405 1.24e-05 ***
## regionmite   6.5225     1.5331   4.255 2.40e-05 ***
## regionvent  10.9831     1.5243   7.205 1.61e-12 ***
## SVL_mm       0.2910     0.0954   3.050  0.00238 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.34 on 649 degrees of freedom
## (44 observations deleted due to missingness)
## Multiple R-squared:  0.1109, Adjusted R-squared:  0.1041
## F-statistic: 16.2 on 5 and 649 DF, p-value: 4.529e-15
```

CEWL ~ SMI

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = SMI,
                 y = TEWL_g_m2h,
                 color = region
                 ),
            size = 1,
            alpha = 0.4) +
  #scale_colour_manual(values = c("palegreen4", "lightblue4",
  #                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = SMI,
                  y = TEWL_g_m2h,
                  color = region
                  ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1) +
  theme_classic() +
  xlab("Scaled Mass Index (g)") +
  ylab(bquote('CEWL (g / '*m^2~h*')')) +
  #annotate("text", x = 1, y = 65, label = "a", size = 6) +
  scale_color_brewer(palette = "Set2",
                    labels = c("Dewlap", "Dorsum",
                              "Head", "Mite Patch", "Ventrum"),
                    name = "") +
  ylim(1, 100) +
  xlim(2, 16) +
  scale_x_continuous(breaks = c(seq(2, 16, by = 2))) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 18),
        axis.text = element_text(color = "black",
                                  family = "sans",
```

```

                                size = 14),
  legend.text = element_text(color = "black",
                              family = "sans",
                              size = 12),
  legend.text.align = 0,
  legend.position = c(0.15, 0.85)
) -> CEWL_SMI_fig

```

```

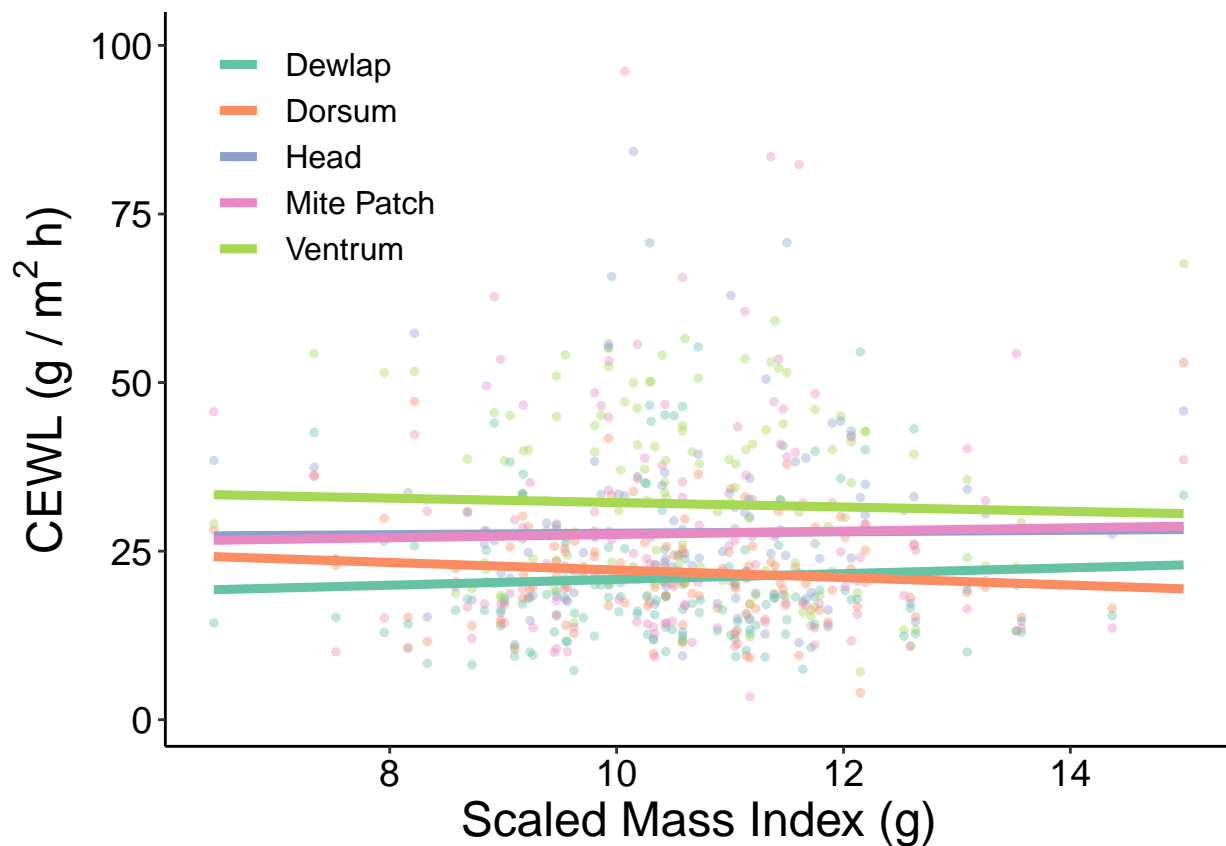
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.

```

```
CEWL_SMI_fig
```

```
## Warning: Removed 44 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 44 rows containing missing values (geom_point).
```



```

# export figure
#ggsave(filename = "CEWL_mass_fig.tiff",
#        plot = CEWL_mass_fig,
#        path = "./final_figures",
#        device = "tiff",
#        dpi = 1200,
#        width = 6, height = 4)

# lm
CEWL_lm13 <- lm(TEWL_g_m2h ~ region + SMI,

```

```

      data = CEWL_data_full)
summary(CEWL_lm13)

##
## Call:
## lm(formula = TEWL_g_m2h ~ region + SMI, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.867  -8.436  -2.543   5.906  68.539
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.27550    3.85593   5.518 4.97e-08 ***
## regiondors    0.82970    1.53515   0.540  0.589
## regionhead    6.66447    1.53515   4.341 1.64e-05 ***
## regionmite    6.55811    1.54398   4.248 2.48e-05 ***
## regionvent   10.95754    1.53515   7.138 2.55e-12 ***
## SMI          -0.02108    0.34976  -0.060  0.952
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.42 on 649 degrees of freedom
## (44 observations deleted due to missingness)
## Multiple R-squared:  0.09821,    Adjusted R-squared:  0.09126
## F-statistic: 14.14 on 5 and 649 DF,  p-value: 3.836e-13

```

CEWL ~ Mass

Head has an opposite trend from all the other body regions, so we need an interaction term.

```

CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = mass_g,
                 y = TEWL_g_m2h,
                 color = region
                 ),
            size = 1,
            alpha = 0.4) +
  #scale_colour_manual(values = c("palegreen4", "lightblue4",
  #                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = mass_g,
                  y = TEWL_g_m2h,
                  color = region
                  ),
             formula = y ~ x,
             method = "lm",
             se = F,
             size = 1.6,
             alpha = 1 ) +
  theme_classic() +
  xlab("Mass (g)") +
  ylab(bquote('CEWL (g / '*m^2~h*')')) +
  #annotate("text", x = 1, y = 65, label = "a", size = 6) +

```



```

scale_color_brewer(palette = "Set2",
  labels = c("Dewlap", "Dorsum",
    "Head", "Mite Patch", "Ventrums"),
  name = "") +
ylim(1, 100) +
xlim(2, 16) +
scale_x_continuous(breaks = c(seq(2, 16, by = 2))) +
theme(text = element_text(color = "black",
  family = "sans",
  size = 18),
  axis.text = element_text(color = "black",
    family = "sans",
    size = 14),
  legend.text = element_text(color = "black",
    family = "sans",
    size = 12),
  legend.text.align = 0,
  legend.position = c(0.15, 0.85)
) -> CEWL_mass_fig

```

```

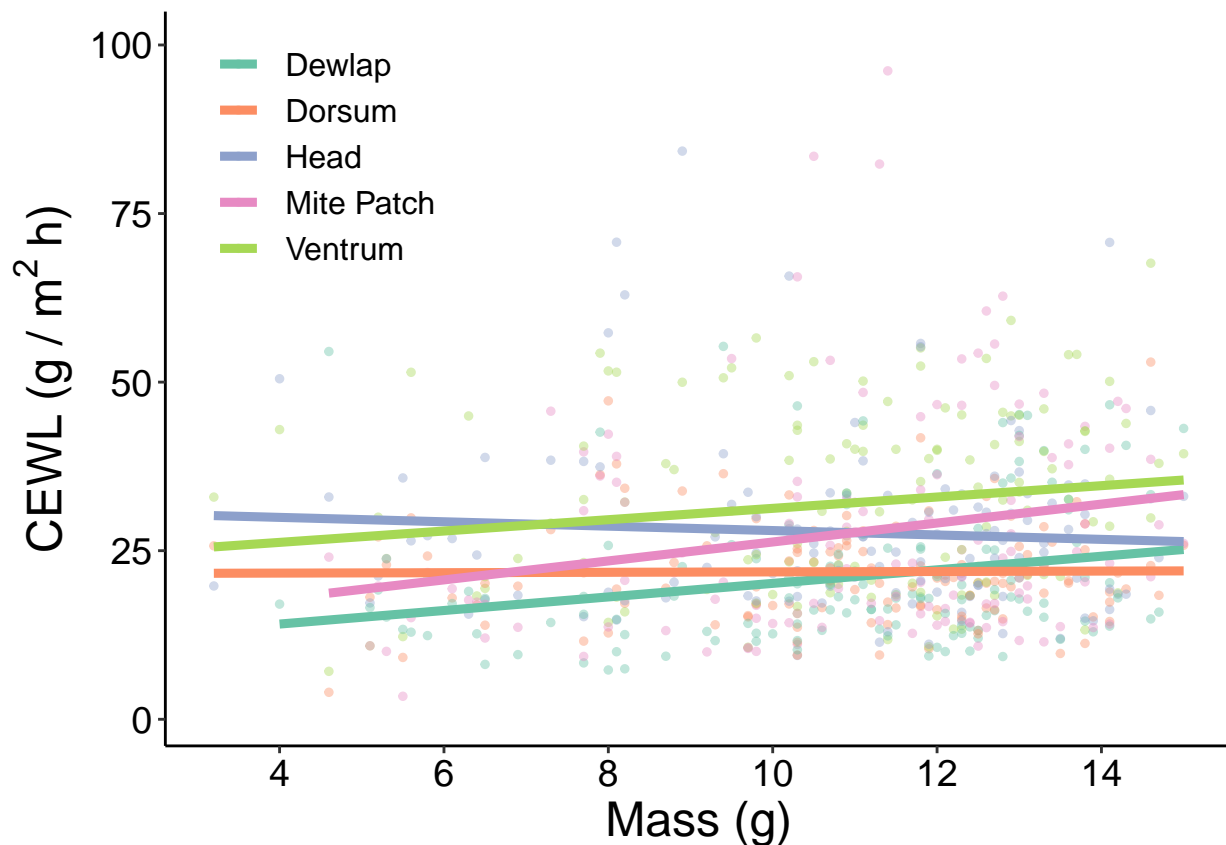
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.

```

```
CEWL_mass_fig
```

```
## Warning: Removed 44 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 44 rows containing missing values (geom_point).
```



```

# export figure
#ggsave(filename = "CEWL_mass_fig.tiff",
#       plot = CEWL_mass_fig,
#       path = "./final_figures",
#       device = "tiff",
#       dpi = 1200,
#       width = 6, height = 4)

# lm
CEWL_lm13.1 <- lm(TEWL_g_m2h ~ region*mass_g,
                 data = CEWL_data_full)
summary(CEWL_lm13.1)

##
## Call:
## lm(formula = TEWL_g_m2h ~ region * mass_g, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.116  -8.402  -2.377   6.152  67.902
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10.0978     4.7889   2.109  0.03536 *
## regiondors     11.4597     6.7379   1.701  0.08947 .
## regionhead     21.1479     6.6404   3.185  0.00152 **
## regionmite      2.1432     6.8771   0.312  0.75541
## regionvent     12.7721     6.6883   1.910  0.05662 .
## mass_g          1.0055     0.4283   2.348  0.01920 *
## regiondors:mass_g -0.9757     0.6029  -1.618  0.10606
## regionhead:mass_g -1.3316     0.5954  -2.237  0.02565 *
## regionmite:mass_g  0.3995     0.6142   0.650  0.51571
## regionvent:mass_g -0.1646     0.5986  -0.275  0.78336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.27 on 645 degrees of freedom
## (44 observations deleted due to missingness)
## Multiple R-squared:  0.1258, Adjusted R-squared:  0.1136
## F-statistic: 10.31 on 9 and 645 DF, p-value: 5.958e-15

```

CEWL ~ Sex

```

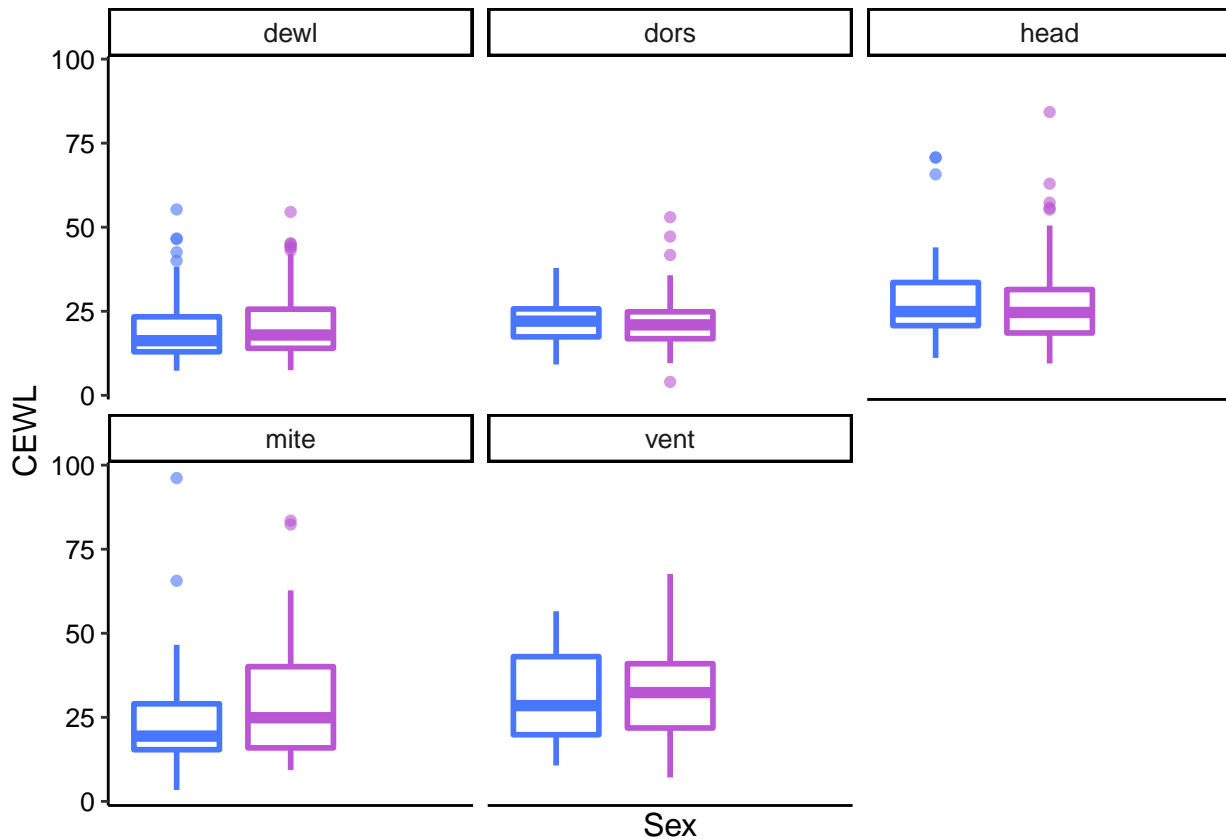
CEWL_data_full %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = sex_M_F,
                  y = TEWL_g_m2h,
                  color = sex_M_F
                  ),
              size = 1,
              alpha = 0.6) +
  facet_wrap(~region) +

```

```

scale_color_manual(values = c("royalblue1", "mediumorchid")) +
scale_x_discrete(breaks = c(1,2,3)) +
theme_classic() +
xlab("Sex") +
ylab("CEWL") +
theme(text = element_text(color = "black", family = "sans", size = 12),
      axis.text = element_text(color = "black", family = "sans", size = 10),
      legend.text.align = 0,
      legend.position = "none"
)

```



```

# lm
CEWL_lm14 <- lm(TEWL_g_m2h ~ region + sex_M_F,
  data = CEWL_data_full)
summary(CEWL_lm14)

```

```

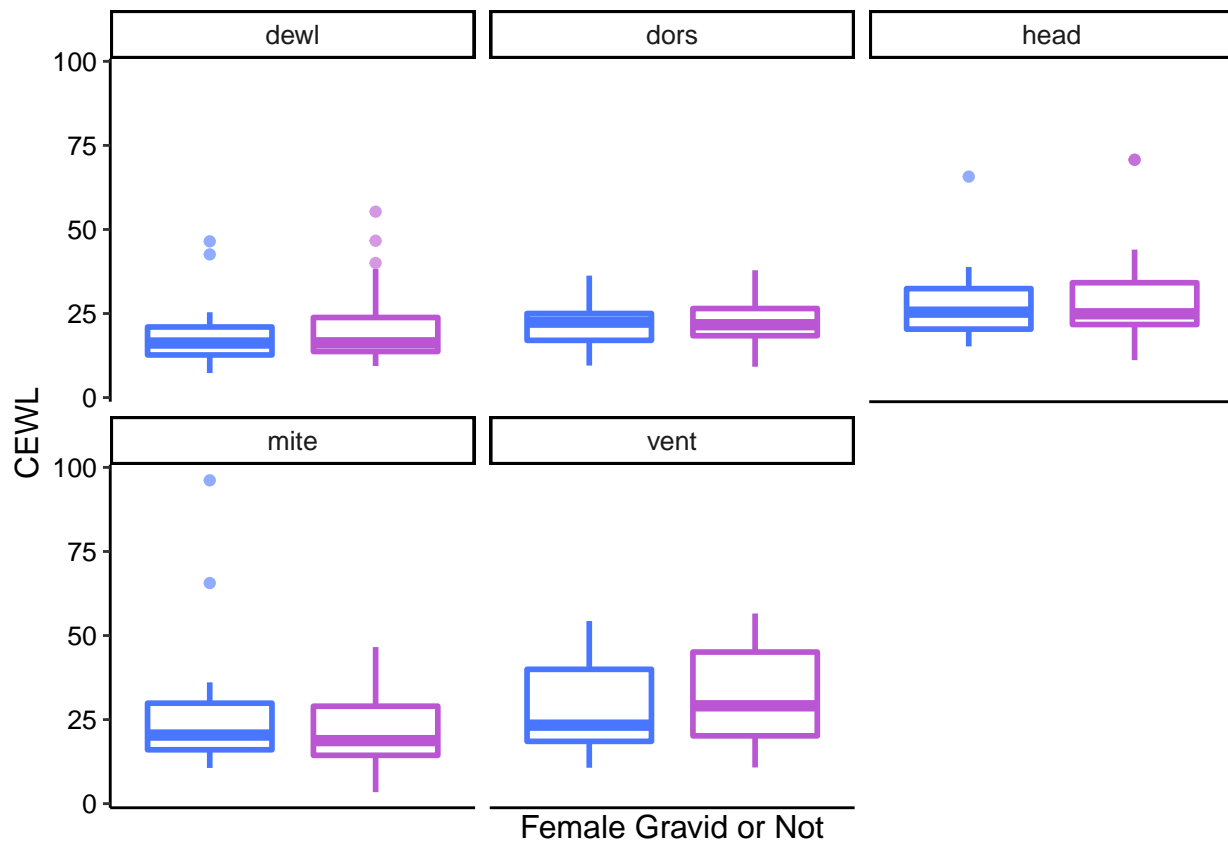
##
## Call:
## lm(formula = TEWL_g_m2h ~ region + sex_M_F, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.238  -8.527  -2.446   6.048  69.226
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.3658     1.2936  15.743  < 2e-16 ***

```

```
## regiondors      0.8403      1.5340      0.548      0.584
## regionhead      6.6673      1.5340      4.346 1.61e-05 ***
## regionmite      6.5686      1.5429      4.257 2.37e-05 ***
## regionvent     10.9678      1.5340      7.150 2.35e-12 ***
## sex_M_FM        1.0146      1.0319      0.983      0.326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.41 on 649 degrees of freedom
## (44 observations deleted due to missingness)
## Multiple R-squared:  0.09955,    Adjusted R-squared:  0.09261
## F-statistic: 14.35 on 5 and 649 DF,  p-value: 2.418e-13
```

CEWL ~ Gravidity

```
CEWL_data_full %>%
  dplyr::filter(sex_M_F == "F") %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = gravid_Y_N,
                   y = TEWL_g_m2h,
                   color = gravid_Y_N
                   ),
              size = 1,
              alpha = 0.6) +
  facet_wrap(~region) +
  scale_color_manual(values = c("royalblue1", "mediumorchid")) +
  scale_x_discrete(breaks = c(1,2,3)) +
  theme_classic() +
  xlab("Female Gravid or Not") +
  ylab("CEWL") +
  theme(text = element_text(color = "black", family = "sans", size = 12),
        axis.text = element_text(color = "black", family = "sans", size = 10),
        legend.text.align = 0,
        legend.position = "none"
  )
)
```



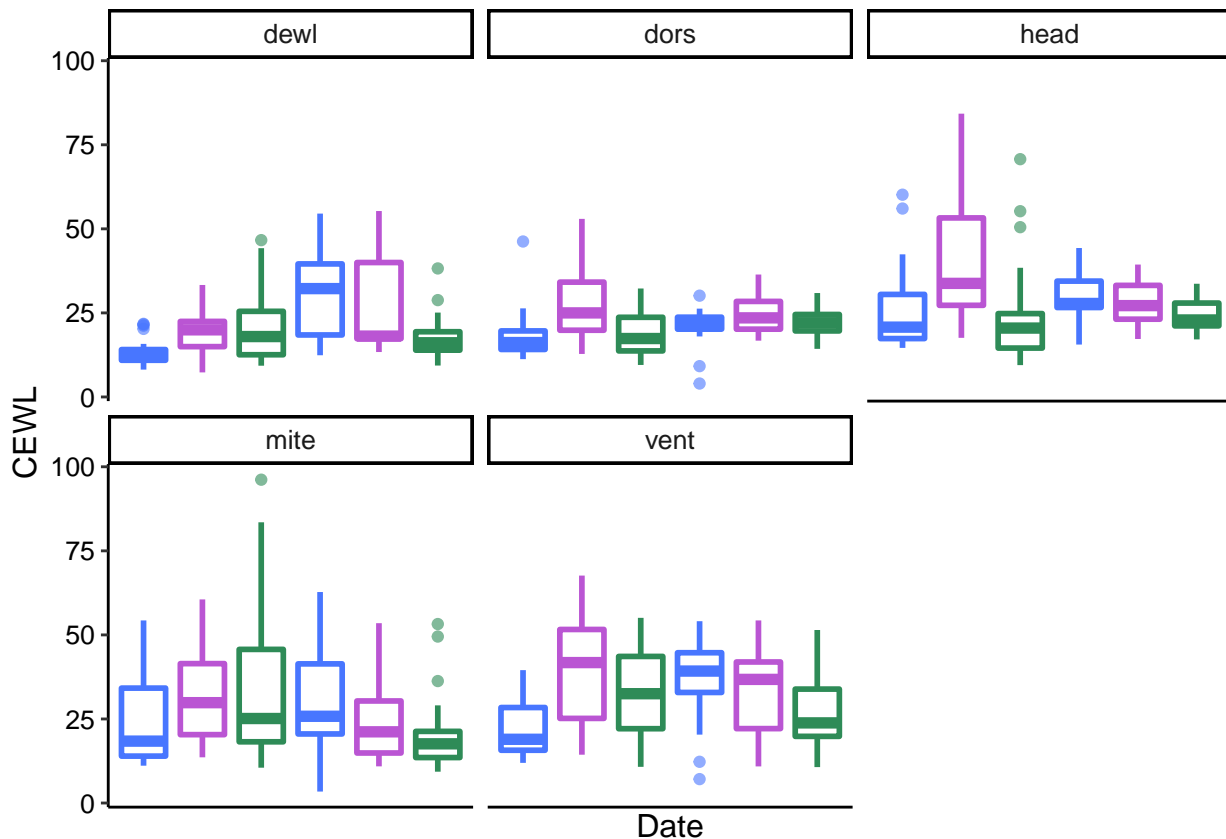
```
# lm
CEWL_lm15 <- lm(TEWL_g_m2h ~ region + gravid_Y_N,
  data = CEWL_data_full)
summary(CEWL_lm15)

##
## Call:
## lm(formula = TEWL_g_m2h ~ region + gravid_Y_N, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.808  -7.963  -2.657   4.874  72.433
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.2109     2.2158   9.121  < 2e-16 ***
## regiondors    1.5956     2.7356   0.583  0.560324
## regionhead    8.7809     2.7508   3.192  0.001629 **
## regionmite    3.5165     2.7508   1.278  0.202532
## regionvent   10.4965     2.7356   3.837  0.000165 ***
## gravid_Y_NY    0.4904     1.7475   0.281  0.779265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.68 on 210 degrees of freedom
## (483 observations deleted due to missingness)
## Multiple R-squared:  0.09649,    Adjusted R-squared:  0.07498
```

```
## F-statistic: 4.486 on 5 and 210 DF, p-value: 0.000663
```

CEWL ~ Week

```
CEWL_data_full %>%
  ggplot(data = .) +
  geom_boxplot(aes(x = as.factor(date),
                  y = TEWL_g_m2h,
                  color = as.factor(date)
                  ),
              size = 1,
              alpha = 0.6) +
  facet_wrap(~region) + # could not figure out how to change facet labels without changing underlying d
  scale_color_manual(values = c("royalblue1", "mediumorchid", "seagreen4",
                                "royalblue1", "mediumorchid", "seagreen4")) +
  scale_x_discrete(breaks = c(1,2,3)) +
  theme_classic() +
  xlab("Date") +
  ylab("CEWL") +
  theme(text = element_text(color = "black", family = "sans", size = 12),
        axis.text = element_text(color = "black", family = "sans", size = 10),
        legend.text.align = 0,
        legend.position = "none"
  )
```



```
# lm
CEWL_lm16 <- lm(TEWL_g_m2h ~ region + date,
```

```

      data = CEWL_data_full)
summary(CEWL_lm16)

##
## Call:
## lm(formula = TEWL_g_m2h ~ region + date, data = CEWL_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.299  -8.269  -2.413   5.549  69.013
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -559.70526   607.60731  -0.921   0.357
## regiondors    1.25433     1.48142   0.847   0.397
## regionhead    7.64676     1.48142   5.162 3.20e-07 ***
## regionmite    6.57099     1.49214   4.404 1.23e-05 ***
## regionvent   10.61651     1.48142   7.166 1.98e-12 ***
## date          0.03096     0.03241   0.955   0.340
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.39 on 693 degrees of freedom
## Multiple R-squared:  0.09621,    Adjusted R-squared:  0.08968
## F-statistic: 14.75 on 5 and 693 DF,  p-value: 9.094e-14

```

CEWL ~ holding time

```

CEWL_data_full %>%
  ggplot(data = .) +
  geom_point(aes(x = hold_time,
                 y = TEWL_g_m2h,
                 color = region
                 ),
             size = 1,
             alpha = 0.6) +
  scale_colour_manual(values = c("palegreen4", "lightblue4",
                                "plum4", "lightpink3", "seashell4")) +
  stat_smooth(aes(x = hold_time,
                  y = TEWL_g_m2h,
                  color = region
                  ),
              formula = y ~ x,
              method = "lm",
              se = F,
              size = 1.6,
              alpha = 1 ) +
  theme_classic() +
  xlab("Holding Time (minutes)") +
  ylab("CEWL (g/m^2/hr)") +
  theme(text = element_text(color = "black",
                             family = "sans",
                             size = 12),

```

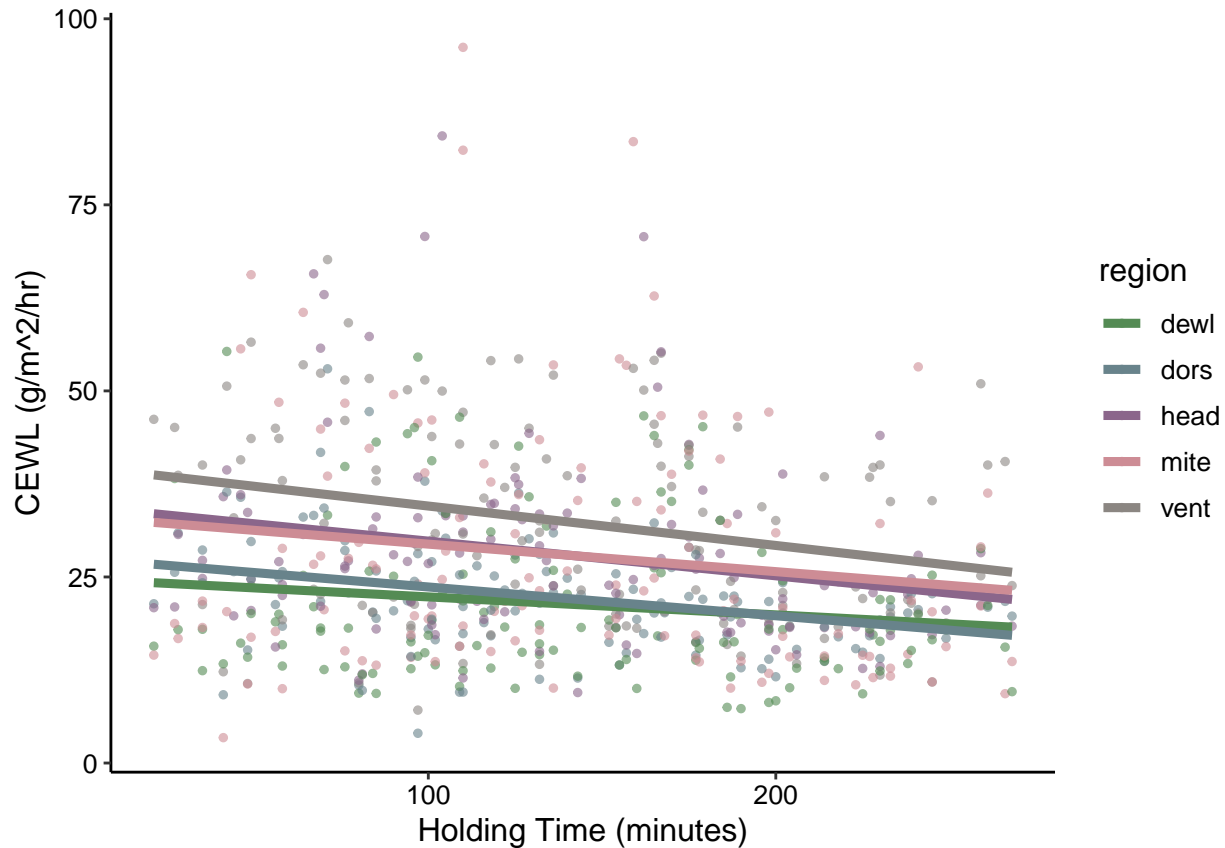
```

axis.text = element_text(color = "black",
                           family = "sans",
                           size = 10),
legend.text.align = 0,
)

```

```
## Warning: Removed 69 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 69 rows containing missing values (geom_point).
```



```

CEWL_lm17 <- glm(data = CEWL_data_full,
                  TEWL_g_m2h ~ hold_time + region)
summary(CEWL_lm17)

```

```

##
## Call:
## glm(formula = TEWL_g_m2h ~ hold_time + region, data = CEWL_data_full)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -28.477   -8.346   -2.076    5.613   67.028
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.958624   1.534599  17.567  < 2e-16 ***
## hold_time    -0.039935   0.007613  -5.246 2.14e-07 ***
## regiondors    0.739762   1.546876   0.478  0.633
## regionhead    6.581377   1.546876   4.255 2.42e-05 ***

```



```
## regionmite    6.566046    1.556123    4.219 2.81e-05 ***
## regionvent   11.034295    1.546868    7.133 2.73e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 150.7369)
##
##      Null deviance: 108878  on 629  degrees of freedom
## Residual deviance:  94060  on 624  degrees of freedom
## (69 observations deleted due to missingness)
## AIC: 4955.6
##
## Number of Fisher Scoring iterations: 2
```

Conclusion

The following variables should be included in the full LMM: - *body region* - *cloacal temperature* at measurement - *capture temp* (and try an interaction with region) - *capture wind speed* and *solar radiation* - *absolute humidity* (and try an interaction with region) - *measurement temperature* (and try an interaction with region) - *individual_ID* (as a random effect) - *SVL* and *mass* - *hold time*

LMMs

Hydration

Based on the simple linear models and figures above, osmolality should be predicted by sample eye, hemolysis, date/week, individual, capture temperature, and capture absolute humidity.

Prep dataframe for computing models:

```
hydrat_mod_dat <- morpho_blood_SMI %>%
  dplyr::select(date,
                individual_ID,
                osmolality_mmol_kg,
                temp_C_interpol,
                abs_humidity_g_m3_interpol,
                SMI,
                hemolyzed
                ) %>%
  dplyr::filter(complete.cases(.))
summary(hydrat_mod_dat)
```

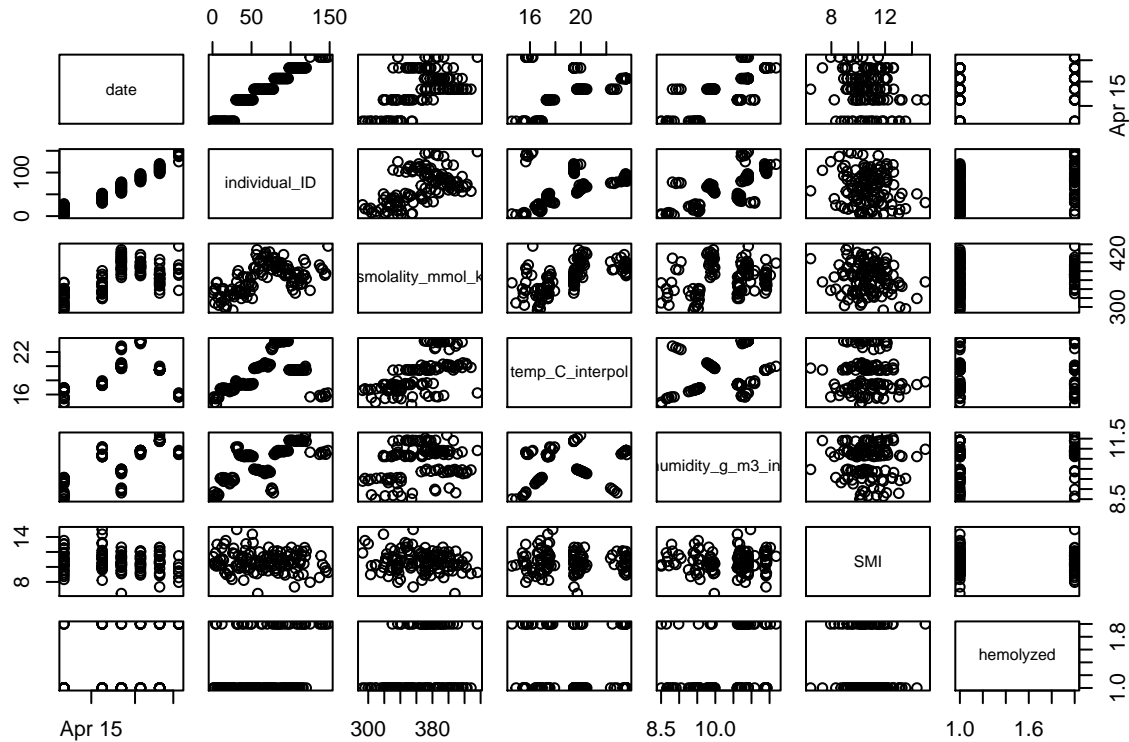
```
##      date            individual_ID osmolality_mmol_kg temp_C_interpol
## Min.   :2021-04-05    1           : 1  Min.   :293.0      Min.   :14.58
## 1st Qu.:2021-04-19    2           : 1  1st Qu.:338.0      1st Qu.:16.95
## Median :2021-04-26    3           : 1  Median :370.0      Median :19.46
## Mean   :2021-04-25    4           : 1  Mean   :365.5      Mean   :19.08
## 3rd Qu.:2021-05-03    5           : 1  3rd Qu.:389.0      3rd Qu.:20.14
## Max.   :2021-05-17    6           : 1  Max.   :436.0      Max.   :23.61
##
##      (Other):115
## abs_humidity_g_m3_interpol      SMI      hemolyzed
## Min.   : 8.504                Min.   : 6.450      N:82
## 1st Qu.: 9.820                1st Qu.: 9.929      Y:39
## Median :10.666                Median :10.583
## Mean   :10.377                Mean   :10.714
```

```
## 3rd Qu.:10.908      3rd Qu.:11.501
## Max.    :11.679      Max.    :14.999
##
```

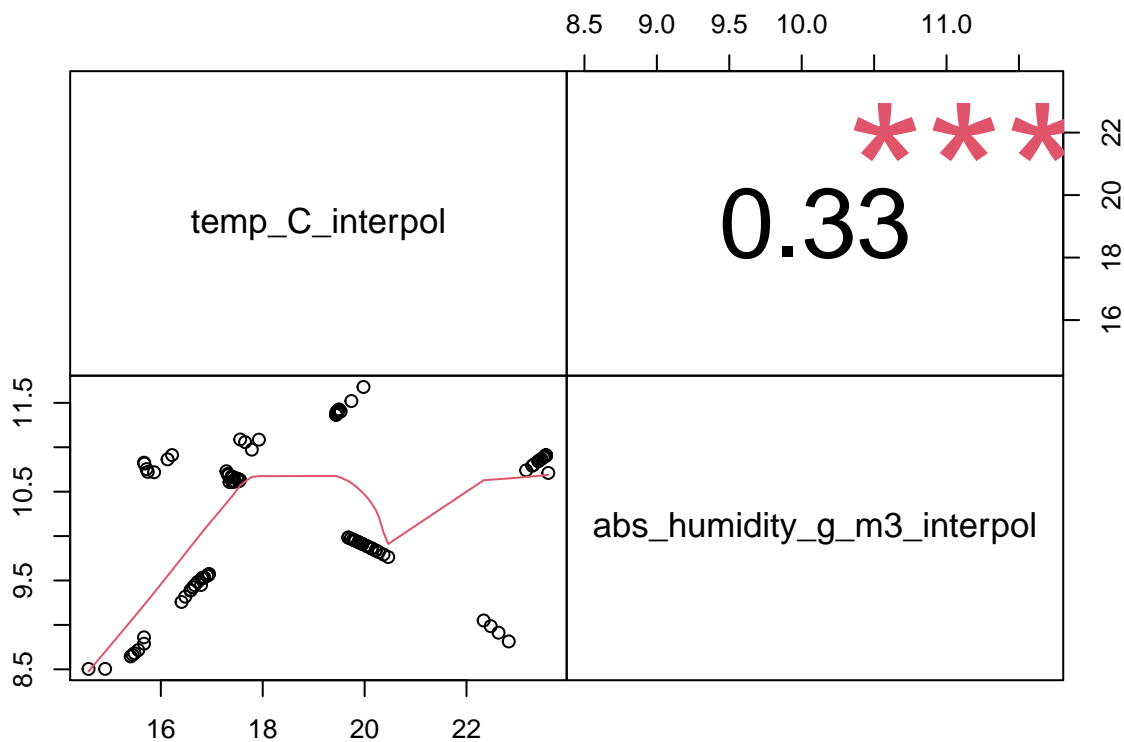
Multicollinearity

First, check for multicollinearity among independent variables:

```
pairs(hydrat_mod_dat)
```



```
# also make another plot with r-sq values
# only continuous numeric variables work for this one
hydrat_mod_dat %>%
  # select cont variables
  dplyr::select(temp_C_interpol,
                abs_humidity_g_m3_interpol) %>%
  # multicollinearity plot
  chart.Correlation(., histogram = F, pch = 19)
```



Date and individual_ID are collinear and should not both be used as fixed effects. But, individual_ID is a factor and there is not repetition in the dataset, so that's not feasible as a variable. Temperature and absolute humidity are the only continuous variables, and they are not badly collinear.

Models & Selection

Start with all the variables that were significant individually in SLRs.

```
# model 1
hydrat_mod1 <- lme4::lmer(data = hydrat_mod_dat,
  # response variable
  osmolality_mmol_kg ~
  # start with interaction + singular effect
  abs_humidity_g_m3_interpol*temp_C_interpol +
  # other potentially important factor
  SMI +
  # random effects
  (1|date) + (1|hemolyzed))

summary(hydrat_mod1)

## Linear mixed model fit by REML ['lmerMod']
## Formula: osmolality_mmol_kg ~ abs_humidity_g_m3_interpol * temp_C_interpol +
##       SMI + (1 | date) + (1 | hemolyzed)
## Data: hydrat_mod_dat
##
## REML criterion at convergence: 1035.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.54027 -0.58267  0.07275  0.63884  2.96793
##
## Random effects:
```

```
## Groups      Name      Variance Std.Dev.
## date      (Intercept) 823.13   28.690
## hemolyzed (Intercept)  26.43    5.141
## Residual                307.36   17.532
## Number of obs: 121, groups:  date, 6; hemolyzed, 2
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)      865.3682   329.7647   2.624
## abs_humidity_g_m3_interpol -57.7664   36.5767  -1.579
## temp_C_interpol -27.6566   17.7098  -1.562
## SMI              -0.8364    1.2377  -0.676
## abs_humidity_g_m3_interpol:temp_C_interpol  3.2251    1.9630   1.643
##
## Correlation of Fixed Effects:
##              (Intr) ab___3_ tmp_C_ SMI
## abs_hmd___3_ -0.989
## tmp_C_ntrpl -0.980  0.977
## SMI          -0.043  0.019  -0.010
## ab___3_:_C_  0.971 -0.987  -0.990 -0.008
```

```
drop1(hydrat_mod1)
```

```
## Single term deletions
##
## Model:
## osmolality_mmol_kg ~ abs_humidity_g_m3_interpol * temp_C_interpol +
##      SMI + (1 | date) + (1 | hemolyzed)
##
##              npar      AIC
## <none>              1072.2
## SMI                  1 1070.7
## abs_humidity_g_m3_interpol:temp_C_interpol  1 1072.8
```

The model would improve (based on lower AIC) if we drop SMI as a predictor.

```
# model 2
hydrat_mod2 <- lme4::lmer(data = hydrat_mod_dat,
                        osmolality_mmol_kg ~
                        abs_humidity_g_m3_interpol*temp_C_interpol +
                        (1|date) + (1|hemolyzed))
```

```
summary(hydrat_mod2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: osmolality_mmol_kg ~ abs_humidity_g_m3_interpol * temp_C_interpol +
##      (1 | date) + (1 | hemolyzed)
##      Data: hydrat_mod_dat
##
## REML criterion at convergence: 1038.1
##
## Scaled residuals:
##      Min      1Q   Median      3Q      Max
## -2.47326 -0.62565  0.07308  0.64959  2.99129
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## date      (Intercept) 844.50   29.060
```

```

## hemolyzed (Intercept) 24.79 4.978
## Residual 305.73 17.485
## Number of obs: 121, groups: date, 6; hemolyzed, 2
##
## Fixed effects:
##
## Estimate Std. Error t value
## (Intercept) 861.250 330.658 2.605
## abs_humidity_g_m3_interpol -57.895 36.763 -1.575
## temp_C_interpol -28.063 17.782 -1.578
## abs_humidity_g_m3_interpol:temp_C_interpol 3.247 1.974 1.645
##
## Correlation of Fixed Effects:
## (Intr) ab___3_ tmp_C_
## abs_hmd___3_ -0.989
## tmp_C_ntrpl -0.982 0.978
## ab___3_:_C_ 0.972 -0.987 -0.991
# compare to full model
anova(hydrat_mod1, hydrat_mod2)

## refitting model(s) with ML (instead of REML)

## Data: hydrat_mod_dat
## Models:
## hydrat_mod2: osmolality_mmol_kg ~ abs_humidity_g_m3_interpol * temp_C_interpol +
## hydrat_mod2: (1 | date) + (1 | hemolyzed)
## hydrat_mod1: osmolality_mmol_kg ~ abs_humidity_g_m3_interpol * temp_C_interpol +
## hydrat_mod1: SMI + (1 | date) + (1 | hemolyzed)
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)
## hydrat_mod2 7 1070.7 1090.3 -528.37 1056.7
## hydrat_mod1 8 1072.2 1094.6 -528.11 1056.2 0.5164 1 0.4724

We do not decrease our predictive power significantly by dropping SMI, so we will consider the reduced model
(2) to be better than the full (1).

Check whether to continue dropping terms:
drop1(hydrat_mod2)

## Single term deletions
##
## Model:
## osmolality_mmol_kg ~ abs_humidity_g_m3_interpol * temp_C_interpol +
## (1 | date) + (1 | hemolyzed)
## npar AIC
## <none> 1070.7
## abs_humidity_g_m3_interpol:temp_C_interpol 1 1071.3

Dropping the current predictors will not improve AIC any more than its current value, so hydrat_mod2 is
the best model to predict osmolality.

Best Model

# save model 2 summary object
osml_best_mod <- summary(hydrat_mod2)
# extract stats table from summary object
osml_best_mod_vals <- data.frame(osml_best_mod$coefficients)

```

```
# export
write.csv(osml_best_mod_vals, "osml_best_mod_vals.csv")
```

Check LM Assumptions (Hydration Model)

First, get residuals:

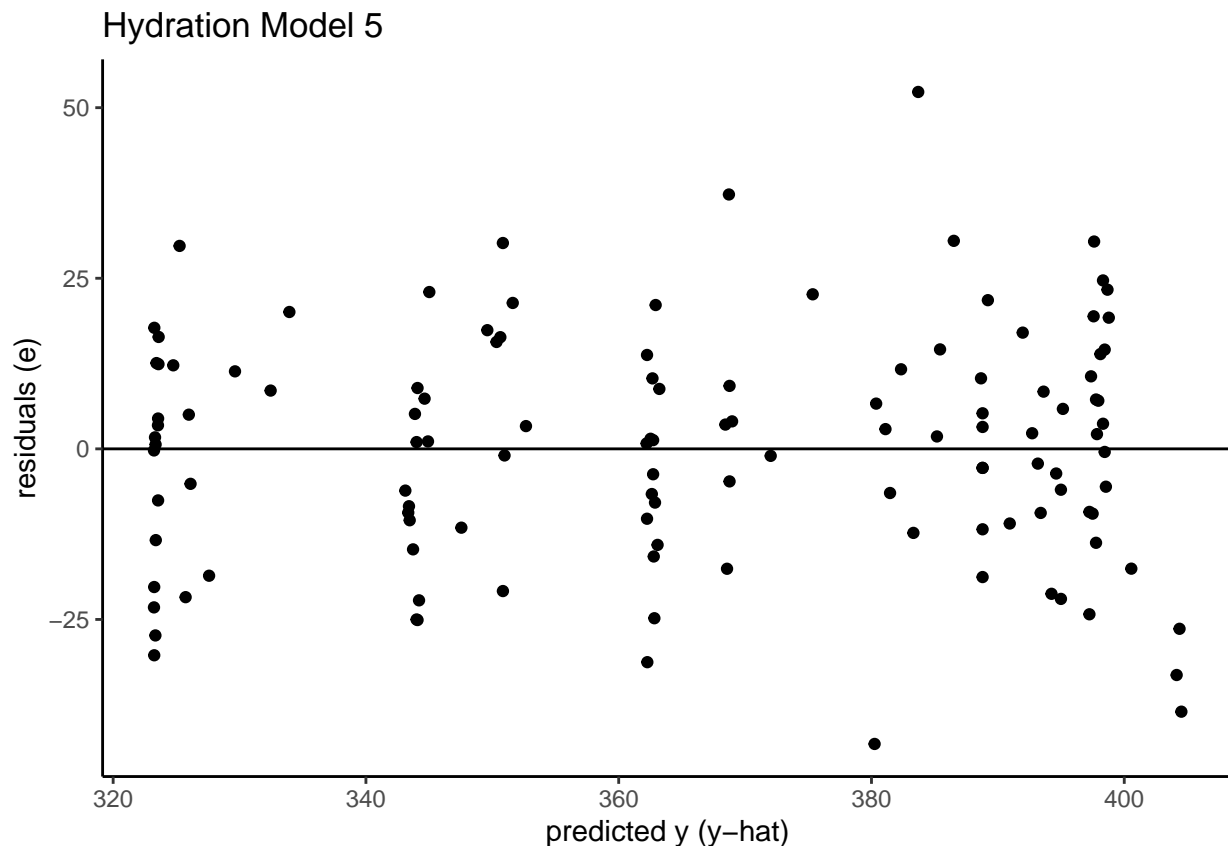
```
res_hydrat_mod <- hydrat_mod_dat %>%
  mutate(y_hat = predict(hydrat_mod2),
         e = residuals(hydrat_mod2))
```

Linearity and Equal Variance

Is the function **linear**? Is there **equal** variance of the residuals? The residuals should be homoskedastic relative to \hat{y} (or x). We don't care if there is a relationship between the residuals ~ dependent variable (actual y).

Plotting residuals shows us whether the data meets linearity and equal variance assumptions:

```
ggplot(data = res_hydrat_mod, aes(x = y_hat, y = e)) +
  geom_point() +
  theme_classic() +
  xlab("predicted y (y-hat)") +
  ylab("residuals (e)") +
  ggtitle("Hydration Model 5") +
  geom_hline(yintercept = 0)
```



It looks pretty evenly distributed. No clear patterning, so linearity and equal error variance should be satisfied.

Brown-Forsythe test to statistically check equal variance:

H0: normally distributed (non-sig test is GOOD) HA: NOT normally distributed (reject nul == assumption not satisfied)

```
# need to create the right data & format first
bf_data_temp <- res_hydrat_mod %>%
  dplyr::filter(complete.cases(temp_C_interpol)) %>%
  dplyr::mutate(middle = median(temp_C_interpol),
               side = temp_C_interpol > middle)
bf_data_temp$side <- as.factor(bf_data_temp$side)

# now run test
bf.test(formula = e ~ side, # y~x
        data = bf_data_temp, # dataframe
        alpha = 0.05, # default 0.05
        na.rm = TRUE, # remove missing data before running?
        verbose = TRUE # print output to console?
        )
```

```
##
##   Brown-Forsythe Test (alpha = 0.05)
## -----
##   data : e and side
##
##   statistic   : 0.004278733
##   num df      : 1
##   denom df    : 118.0462
##   p.value     : 0.9479566
##
##   Result      : Difference is not statistically significant.
## -----
```

```
# need to create the right data & format first
bf_data_abshum <- res_hydrat_mod %>%
  dplyr::filter(complete.cases(abs_humidity_g_m3_interpol)) %>%
  dplyr::mutate(middle = median(abs_humidity_g_m3_interpol),
               side = abs_humidity_g_m3_interpol > middle)
bf_data_abshum$side <- as.factor(bf_data_abshum$side)

# now run test
bf.test(formula = e ~ side, # y~x
        data = bf_data_abshum, # dataframe
        alpha = 0.05, # default 0.05
        na.rm = TRUE, # remove missing data before running?
        verbose = TRUE # print output to console?
        )
```

```
##
##   Brown-Forsythe Test (alpha = 0.05)
## -----
##   data : e and side
##
##   statistic   : 0.1445405
##   num df      : 1
##   denom df    : 118.9914
```

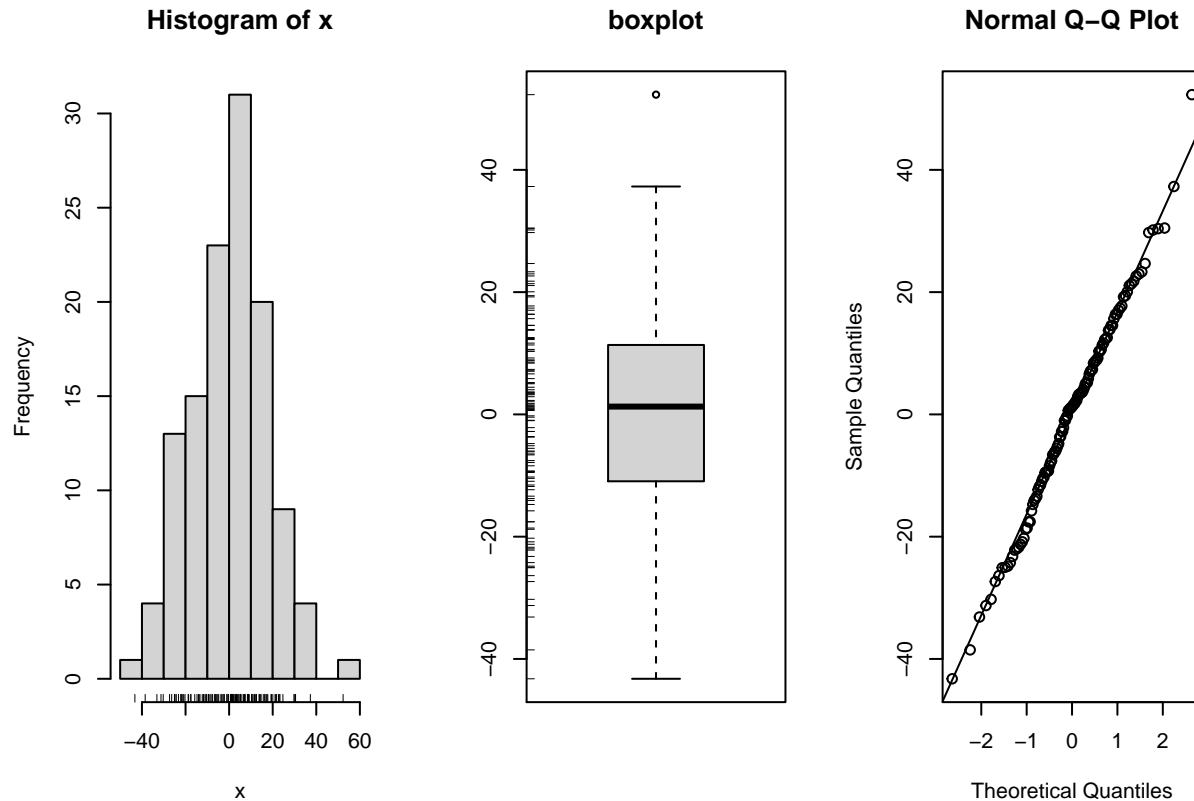
```
## p.value      : 0.704486
##
## Result       : Difference is not statistically significant.
## -----
```

Equal variance is satisfied.

Now check normality. Is the distribution of residuals **normal**?

use Shapiro-Wilk normality test: H0: data is NOT significantly different from normal distribution HA: data IS significantly different from normal distribution

```
simple.eda(res_hydrat_mod$res)
```



```
shapiro.test(res_hydrat_mod$res)
```

```
##
## Shapiro-Wilk normality test
##
## data:  res_hydrat_mod$res
## W = 0.99535, p-value = 0.9629
```

Normality is satisfied.

Conclusion

hydrat_mod2 is the best model to predict osmolality, and the model satisfies all linear regression assumptions.

CEWL

Based on the simple linear models and figures above, CEWL should be predicted by: - body region - cloacal temperature at measurement - capture temperature, absolute humidity, wind speed, and solar radiation -

ambient temperature during measurement - SVL and mass - hold time (time between capture vs measurement)
 - individual ID (as a random effect)

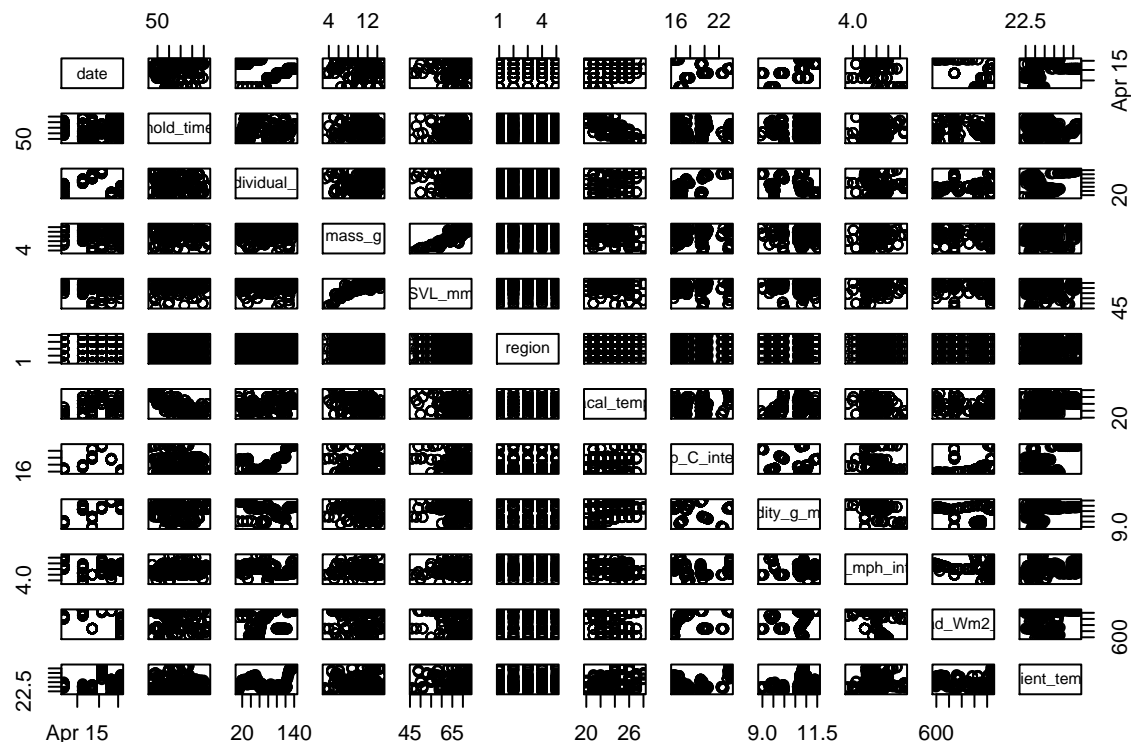
Prep dataframe for models:

```
CEWL_mod_dat <- CEWL_data_full %>%
  # select variables of interest
  dplyr::select(date,
                hold_time,
                individual_ID,
                mass_g,
                SVL_mm,
                TEWL_g_m2h,
                region,
                cloacal_temp_C,
                temp_C_interpol,
                abs_humidity_g_m3_interpol,
                Wind_mph_interpol,
                Solar_rad_Wm2_interpol,
                ambient_temp_C
                ) %>%
  dplyr::filter(complete.cases(.))
```

Multicollinearity

Check for multicollinearity among independent variables:

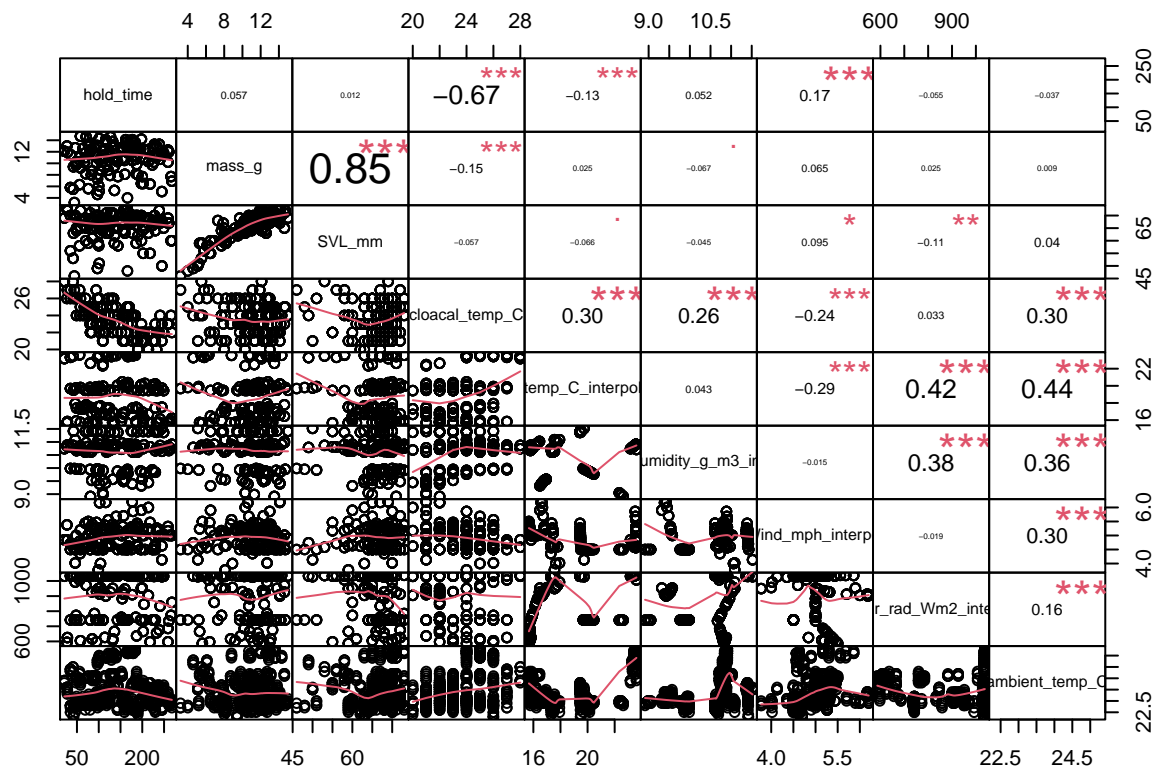
```
CEWL_mod_dat %>%
  # get rid of dependent variable
  dplyr::select(-TEWL_g_m2h, individual_ID) %>%
  # multicollinearity plot
  pairs(.)
```



```

# also make another plot with r-sq values
# non-numeric variables don't work for this
CEWL_mod_dat %>%
  # select variables of interest
  dplyr::select(-TEWL_g_m2h, -date, -region, -individual_ID) %>%
  # multicollinearity plot
  chart.Correlation(., histogram = F, pch = 19)

```



Mass and SVL are very collinear variables that should not be used in combination. Cloacal temp and hold time are pretty collinear, which makes sense; as we held them, they got cooler because they were inside and not basking. Individual ID and absolute humidity are also pretty collinear, but it's less intuitive. Temperature and solar radiation, as well as humidity and solar radiation, are each collinear pairs. We will use model selection to figure out which variable from each collinear pair is better to include in the model.

Models & Selection

This is the full model with every potential variable and interaction baed on figures and SLR.

```

# model 1
CEWL_mod1 <- lme4::lmer(data = CEWL_mod_dat,
  TEWL_g_m2h ~
    region * (mass_g + ambient_temp_C +
      temp_C_interpol +
      abs_humidity_g_m3_interpol) +
    hold_time + SVL_mm +
    cloacal_temp_C +
    temp_C_interpol:abs_humidity_g_m3_interpol +
    Wind_mph_interpol +
    Solar_rad_Wm2_interpol +
    (1|individual_ID))

```

```
summary(CEWL_mod1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: TEWL_g_m2h ~ region * (mass_g + ambient_temp_C + temp_C_interpol +
##      abs_humidity_g_m3_interpol) + hold_time + SVL_mm + cloacal_temp_C +
##      temp_C_interpol:abs_humidity_g_m3_interpol + Wind_mph_interpol +
##      Solar_rad_Wm2_interpol + (1 | individual_ID)
## Data: CEWL_mod_dat
##
## REML criterion at convergence: 4723.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.2331 -0.5426 -0.1011  0.4132  5.1591
##
## Random effects:
##   Groups             Name             Variance Std.Dev.
## individual_ID (Intercept) 30.02       5.479
## Residual                94.93       9.743
## Number of obs: 630, groups: individual_ID, 128
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)    -1.559e+02  1.576e+02  -0.990
## regiondors       8.701e+01  4.214e+01   2.065
## regionhead      9.480e+01  4.184e+01   2.266
## regionmite     1.483e+02  4.253e+01   3.487
## regionvent     1.091e+02  4.174e+01   2.615
## mass_g          1.132e+00  5.892e-01   1.921
## ambient_temp_C  -4.522e-01  1.893e+00  -0.239
## temp_C_interpol  3.792e+00  7.363e+00   0.515
## abs_humidity_g_m3_interpol 7.795e+00  1.297e+01   0.601
## hold_time       7.604e-03  1.481e-02   0.513
## SVL_mm          1.191e-01  2.461e-01   0.484
## cloacal_temp_C  2.455e+00  5.874e-01   4.179
## Wind_mph_interpol -3.363e-01  2.004e+00  -0.168
## Solar_rad_Wm2_interpol 1.266e-02  7.133e-03   1.775
## regiondors:mass_g -9.988e-01  4.884e-01  -2.045
## regionhead:mass_g -1.300e+00  4.807e-01  -2.705
## regionmite:mass_g  1.113e-01  4.970e-01   0.224
## regionvent:mass_g -1.059e-01  4.844e-01  -0.219
## regiondors:ambient_temp_C -2.018e+00  2.056e+00  -0.982
## regionhead:ambient_temp_C -2.673e+00  2.059e+00  -1.298
## regionmite:ambient_temp_C -2.372e+00  2.075e+00  -1.143
## regionvent:ambient_temp_C -3.947e+00  2.056e+00  -1.920
## regiondors:temp_C_interpol -1.797e+00  5.403e-01  -3.326
## regionhead:temp_C_interpol -1.214e+00  5.431e-01  -2.236
## regionmite:temp_C_interpol -4.837e-02  5.479e-01  -0.088
## regionvent:temp_C_interpol -4.230e-01  5.441e-01  -0.778
## regiondors:abs_humidity_g_m3_interpol 5.372e-01  2.067e+00   0.260
## regionhead:abs_humidity_g_m3_interpol 1.070e+00  2.074e+00   0.516
## regionmite:abs_humidity_g_m3_interpol -8.155e+00  2.083e+00  -3.915
## regionvent:abs_humidity_g_m3_interpol 3.362e-01  2.073e+00   0.162
## temp_C_interpol:abs_humidity_g_m3_interpol -2.726e-01  7.050e-01  -0.387
```

```
##
## Correlation matrix not shown by default, as p = 31 > 12.
## Use print(x, correlation=TRUE) or
##     vcov(x)         if you need it
```

Check which variables to drop:

```
drop1(CEWL_mod1)
```

```
## Single term deletions
##
## Model:
## TEWL_g_m2h ~ region * (mass_g + ambient_temp_C + temp_C_interpol +
##   abs_humidity_g_m3_interpol) + hold_time + SVL_mm + cloacal_temp_C +
##   temp_C_interpol:abs_humidity_g_m3_interpol + Wind_mph_interpol +
##   Solar_rad_Wm2_interpol + (1 | individual_ID)
##
##               npar      AIC
## <none>                4810.5
## hold_time              1 4808.8
## SVL_mm                 1 4808.7
## cloacal_temp_C         1 4826.3
## Wind_mph_interpol      1 4808.5
## Solar_rad_Wm2_interpol 1 4811.9
## region:mass_g           4 4817.0
## region:ambient_temp_C   4 4806.5
## region:temp_C_interpol  4 4819.7
## region:abs_humidity_g_m3_interpol 4 4830.6
## temp_C_interpol:abs_humidity_g_m3_interpol 1 4808.6
```

Based on AIC, dropping SVL, hold time, temp, humidity, wind speed, region:ambient_temp_C, and temp_C_interpol:abs_humidity_g_m3_interpol would result in a better model.

Start with region:ambient_temp_C and temp_C_interpol:abs_humidity_g_m3_interpol interactions:

```
# model 2
CEWL_mod2 <- lme4::lmer(data = CEWL_mod_dat,
  TEWL_g_m2h ~
    region * (mass_g +
              temp_C_interpol +
              abs_humidity_g_m3_interpol) +
  hold_time + SVL_mm +
  cloacal_temp_C +
  Wind_mph_interpol +
  Solar_rad_Wm2_interpol +
  (1|individual_ID))
summary(CEWL_mod2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
##   hold_time + SVL_mm + cloacal_temp_C + Wind_mph_interpol +
##   Solar_rad_Wm2_interpol + (1 | individual_ID)
## Data: CEWL_mod_dat
##
## REML criterion at convergence: 4747.5
##
## Scaled residuals:
```

```

##      Min      1Q  Median      3Q      Max
## -2.2159 -0.5667 -0.0867  0.4067  5.1995
##
## Random effects:
##   Groups      Name      Variance Std.Dev.
## individual_ID (Intercept) 30.90    5.559
## Residual                95.06    9.750
## Number of obs: 630, groups: individual_ID, 128
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)    -87.058039  24.523264  -3.550
## regiondors      51.945825  22.844519   2.274
## regionhead      49.495464  22.784245   2.172
## regionmite     107.535596  23.034315   4.668
## regionvent      42.587507  22.743292   1.873
## mass_g          1.200986   0.592606   2.027
## temp_C_interpol  0.710060   0.439275   1.616
## abs_humidity_g_m3_interpol 2.421998   1.732231   1.398
## hold_time       0.005608   0.014493   0.387
## SVL_mm          0.080709   0.247390   0.326
## cloacal_temp_C   2.244956   0.562955   3.988
## Wind_mph_interpol -2.965487   1.518371  -1.953
## Solar_rad_Wm2_interpol  0.013837   0.005330   2.596
## regiondors:mass_g -1.009311   0.488679  -2.065
## regionhead:mass_g -1.316901   0.480895  -2.738
## regionmite:mass_g  0.125620   0.497154   0.253
## regionvent:mass_g -0.154379   0.484223  -0.319
## regiondors:temp_C_interpol -2.033402   0.481402  -4.224
## regionhead:temp_C_interpol -1.538362   0.483775  -3.180
## regionmite:temp_C_interpol -0.355824   0.482327  -0.738
## regionvent:temp_C_interpol -0.904063   0.485072  -1.864
## regiondors:abs_humidity_g_m3_interpol -0.191633   1.919221  -0.100
## regionhead:abs_humidity_g_m3_interpol  0.018035   1.914084   0.009
## regionmite:abs_humidity_g_m3_interpol -9.039969   1.936276  -4.669
## regionvent:abs_humidity_g_m3_interpol -1.224058   1.913883  -0.640
##
## Correlation matrix not shown by default, as p = 25 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)      if you need it
# compare
anova(CEWL_mod2, CEWL_mod1)

## refitting model(s) with ML (instead of REML)

## Data: CEWL_mod_dat
## Models:
## CEWL_mod2: TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
## CEWL_mod2:      hold_time + SVL_mm + cloacal_temp_C + Wind_mph_interpol +
## CEWL_mod2:      Solar_rad_Wm2_interpol + (1 | individual_ID)
## CEWL_mod1: TEWL_g_m2h ~ region * (mass_g + ambient_temp_C + temp_C_interpol +
## CEWL_mod1:      abs_humidity_g_m3_interpol) + hold_time + SVL_mm + cloacal_temp_C +
## CEWL_mod1:      temp_C_interpol:abs_humidity_g_m3_interpol + Wind_mph_interpol +
## CEWL_mod1:      Solar_rad_Wm2_interpol + (1 | individual_ID)

```

```
##          npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## CEWL_mod2    27 4807.8 4927.8 -2376.9   4753.8
## CEWL_mod1    33 4810.5 4957.2 -2372.2   4744.5 9.3042  6    0.1572
```

AIC improved somewhat and we do not lose a significant amount of predictive power, so the reduced model is better. Continue dropping...

Check drop terms again:

```
drop1(CEWL_mod2)
```

```
## Single term deletions
##
## Model:
## TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
##   hold_time + SVL_mm + cloacal_temp_C + Wind_mph_interpol +
##   Solar_rad_Wm2_interpol + (1 | individual_ID)
##
##          npar    AIC
## <none>                4807.8
## hold_time              1 4805.9
## SVL_mm                 1 4805.9
## cloacal_temp_C         1 4821.9
## Wind_mph_interpol      1 4809.8
## Solar_rad_Wm2_interpol  1 4812.8
## region:mass_g           4 4814.4
## region:temp_C_interpol  4 4824.0
## region:abs_humidity_g_m3_interpol 4 4832.5
```

Hold time, SVL, and wind should still be deleted.

```
# model 3
CEWL_mod3 <- lme4::lmer(data = CEWL_mod_dat,
  TEWL_g_m2h ~
    region * (mass_g +
      temp_C_interpol +
      abs_humidity_g_m3_interpol) +
    cloacal_temp_C +
    Solar_rad_Wm2_interpol +
    (1|individual_ID))
summary(CEWL_mod3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
##   cloacal_temp_C + Solar_rad_Wm2_interpol + (1 | individual_ID)
## Data: CEWL_mod_dat
##
## REML criterion at convergence: 4746.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.2165 -0.5681 -0.0845  0.3897  5.2348
##
## Random effects:
##   Groups             Name             Variance Std.Dev.
## individual_ID (Intercept) 31.28         5.593
## Residual                 95.06         9.750
```

```

## Number of obs: 630, groups:  individual_ID, 128
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)    -99.785318   19.355118  -5.156
## regiondors       51.879222   22.844484   2.271
## regionhead       49.554223   22.783334   2.175
## regionmite      107.365993   23.033169   4.661
## regionvent       42.581631   22.742456   1.872
## mass_g           1.312125    0.398001   3.297
## temp_C_interpol   0.875234    0.425830   2.055
## abs_humidity_g_m3_interpol  2.635599    1.663082   1.585
## cloacal_temp_C    2.210436    0.383892   5.758
## Solar_rad_Wm2_interpol  0.012051    0.005007   2.407
## regiondors:mass_g  -1.007466    0.488673  -2.062
## regionhead:mass_g  -1.317803    0.480831  -2.741
## regionmite:mass_g   0.126792    0.497084   0.255
## regionvent:mass_g  -0.153665    0.484142  -0.317
## regiondors:temp_C_interpol -2.030832    0.481401  -4.219
## regionhead:temp_C_interpol -1.537802    0.483770  -3.179
## regionmite:temp_C_interpol -0.354780    0.482324  -0.736
## regionvent:temp_C_interpol -0.901930    0.485071  -1.859
## regiondors:abs_humidity_g_m3_interpol -0.191317    1.919226  -0.100
## regionhead:abs_humidity_g_m3_interpol  0.012735    1.914041   0.007
## regionmite:abs_humidity_g_m3_interpol -9.027196    1.936220  -4.662
## regionvent:abs_humidity_g_m3_interpol -1.227483    1.913844  -0.641
##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it
# compare
anova(CEWL_mod3, CEWL_mod1)

## refitting model(s) with ML (instead of REML)
## Data: CEWL_mod_dat
## Models:
## CEWL_mod3: TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
## CEWL_mod3:      cloacal_temp_C + Solar_rad_Wm2_interpol + (1 | individual_ID)
## CEWL_mod1: TEWL_g_m2h ~ region * (mass_g + ambient_temp_C + temp_C_interpol +
## CEWL_mod1:      abs_humidity_g_m3_interpol) + hold_time + SVL_mm + cloacal_temp_C +
## CEWL_mod1:      temp_C_interpol:abs_humidity_g_m3_interpol + Wind_mph_interpol +
## CEWL_mod1:      Solar_rad_Wm2_interpol + (1 | individual_ID)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## CEWL_mod3    24 4805.9 4912.6 -2379.0   4757.9
## CEWL_mod1    33 4810.5 4957.2 -2372.2   4744.5 13.45  9    0.1433
anova(CEWL_mod3, CEWL_mod2)

## refitting model(s) with ML (instead of REML)
## Data: CEWL_mod_dat
## Models:
## CEWL_mod3: TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
## CEWL_mod3:      cloacal_temp_C + Solar_rad_Wm2_interpol + (1 | individual_ID)

```

```
## CEWL_mod2: TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
## CEWL_mod2:      hold_time + SVL_mm + cloacal_temp_C + Wind_mph_interpol +
## CEWL_mod2:      Solar_rad_Wm2_interpol + (1 | individual_ID)
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## CEWL_mod3    24 4805.9 4912.6 -2379.0   4757.9
## CEWL_mod2    27 4807.8 4927.8 -2376.9   4753.8 4.1461  3      0.2461
```

Once again, the AIC is slightly lower, and our model fit is not significantly worse by going to model 3, so the reduced model is better.

Check drop terms again to make sure nothing else could potentially be removed:

```
drop1(CEWL_mod3)
```

```
## Single term deletions
##
## Model:
## TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
##      cloacal_temp_C + Solar_rad_Wm2_interpol + (1 | individual_ID)
##           npar      AIC
## <none>                        4805.9
## cloacal_temp_C                  1 4834.7
## Solar_rad_Wm2_interpol           1 4809.9
## region:mass_g                    4 4812.6
## region:temp_C_interpol           4 4822.1
## region:abs_humidity_g_m3_interpol 4 4830.5
```

Check whether removing Solar_rad_Wm2_interpol is okay:

```
# model 4
CEWL_mod4 <- lme4::lmer(data = CEWL_mod_dat,
                        TEWL_g_m2h ~
                          region * (mass_g +
                                      temp_C_interpol +
                                      abs_humidity_g_m3_interpol) +
                          cloacal_temp_C +
                          (1|individual_ID))
summary(CEWL_mod4)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
##      cloacal_temp_C + (1 | individual_ID)
## Data: CEWL_mod_dat
##
## REML criterion at convergence: 4743.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.2684 -0.5593 -0.0981  0.3787  5.1696
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## individual_ID (Intercept) 33.36   5.776
## Residual                95.01   9.747
## Number of obs: 630, groups: individual_ID, 128
##
```



```

## Fixed effects:
##
##               Estimate Std. Error t value
## (Intercept)    -103.22837   19.46944  -5.302
## regiondors       51.84649   22.84083   2.270
## regionhead       49.38419   22.77852   2.168
## regionmite      107.53197   23.02916   4.669
## regionvent       42.41691   22.73810   1.865
## mass_g           1.32056    0.40108   3.292
## temp_C_interpol   1.21248    0.40537   2.991
## abs_humidity_g_m3_interpol  3.84425    1.59777   2.406
## cloacal_temp_C    1.99607    0.38092   5.240
## regiondors:mass_g  -1.00927    0.48865  -2.065
## regionhead:mass_g  -1.31378    0.48076  -2.733
## regionmite:mass_g   0.12224    0.49701   0.246
## regionvent:mass_g  -0.14769    0.48408  -0.305
## regiondors:temp_C_interpol -2.03061    0.48130  -4.219
## regionhead:temp_C_interpol -1.53934    0.48365  -3.183
## regionmite:temp_C_interpol -0.35235    0.48223  -0.731
## regionvent:temp_C_interpol -0.90458    0.48498  -1.865
## regiondors:abs_humidity_g_m3_interpol -0.18693    1.91886  -0.097
## regionhead:abs_humidity_g_m3_interpol  0.02664    1.91363   0.014
## regionmite:abs_humidity_g_m3_interpol -9.04168    1.93588  -4.671
## regionvent:abs_humidity_g_m3_interpol -1.21423    1.91344  -0.635

##
## Correlation matrix not shown by default, as p = 21 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it

# compare
anova(CEWL_mod4, CEWL_mod1)

## refitting model(s) with ML (instead of REML)

## Data: CEWL_mod_dat
## Models:
## CEWL_mod4: TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
## CEWL_mod4:      cloacal_temp_C + (1 | individual_ID)
## CEWL_mod1: TEWL_g_m2h ~ region * (mass_g + ambient_temp_C + temp_C_interpol +
## CEWL_mod1:      abs_humidity_g_m3_interpol) + hold_time + SVL_mm + cloacal_temp_C +
## CEWL_mod1:      temp_C_interpol:abs_humidity_g_m3_interpol + Wind_mph_interpol +
## CEWL_mod1:      Solar_rad_Wm2_interpol + (1 | individual_ID)
##      npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## CEWL_mod4    23 4809.9 4912.1 -2381.9   4763.9
## CEWL_mod1    33 4810.5 4957.2 -2372.2   4744.5 19.382 10    0.03567 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(CEWL_mod4, CEWL_mod3)

## refitting model(s) with ML (instead of REML)

## Data: CEWL_mod_dat
## Models:
## CEWL_mod4: TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
## CEWL_mod4:      cloacal_temp_C + (1 | individual_ID)
## CEWL_mod3: TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +

```

```
## CEWL_mod3:      cloacal_temp_C + Solar_rad_Wm2_interpol + (1 | individual_ID)
##              npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## CEWL_mod4    23 4809.9 4912.1 -2381.9  4763.9
## CEWL_mod3    24 4805.9 4912.6 -2379.0  4757.9 5.9316  1    0.01487 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We do lose significant predictive power when we go from model 3 to 4... So model 3 is the best model to predict CEWL.

Best Model

The best model is CEWL predicted by: body region and its interaction with mass and capture temp and humidity, cloacal temperature, solar radiation at capture time, and individual ID as a random effect.

```
# save best model summary object
CEWL_best_mod <- summary(CEWL_mod3)
# extract stats table from summary object
CEWL_best_mod_vals <- data.frame(CEWL_best_mod$coefficients)
# export
write.csv(CEWL_best_mod_vals, "CEWL_best_mod_vals.csv")
```

Check LM Assumptions

First, get residuals:

```
CEWL_mod_res <- CEWL_mod_dat %>%
  mutate(y_hat = predict(CEWL_mod3),
         e = residuals(CEWL_mod3))
```

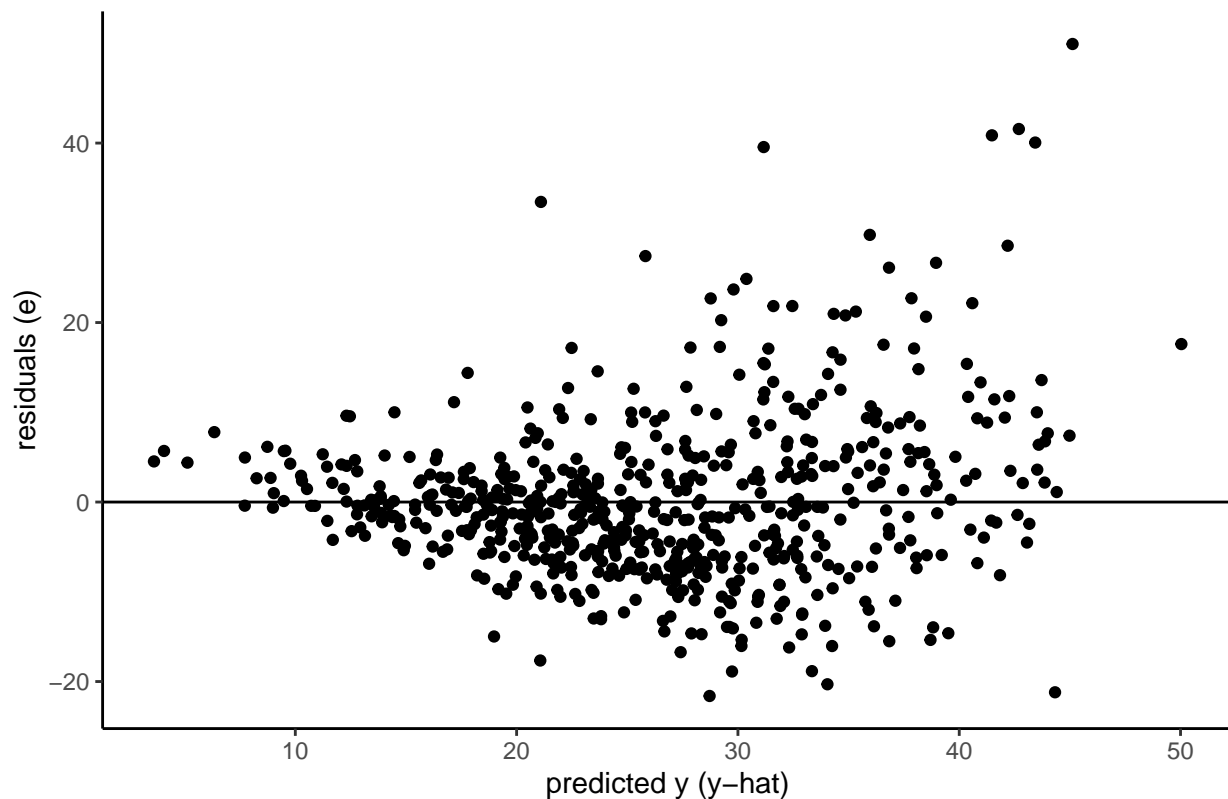
Linearity and Equal Variance

Is the function **linear**? Is there **equal** variance of the residuals? The residuals should be homoskedastic relative to \hat{y} (or x). We don't care if there is a relationship between the residuals ~ dependent variable (actual y).

Plotting residuals shows us whether the data meets linearity and equal variance assumptions:

```
ggplot(data = CEWL_mod_res, aes(x = y_hat, y = e)) +
  geom_point() +
  theme_classic() +
  xlab("predicted y (y-hat)") +
  ylab("residuals (e)") +
  ggtitle("CEWL Model Residuals") +
  geom_hline(yintercept = 0)
```

CEWL Model Residuals



It's definitely making a fan shape. :(

Brown-Forsythe test to statistically check equal variance, for each continuous predictor variable:

H0: normally distributed (non-sig test is GOOD) HA: NOT normally distributed (reject nul == assumption not satisfied)

```
# need to create the right data & format first
bf_data_CEWL <- CEWL_mod_res %>%
  dplyr::mutate(middle_mass = median(mass_g), # mass
               side_mass = as.factor(mass_g > middle_mass),
               # solar radiation
               middle_sorad = median(Solar_rad_Wm2_interpol),
               side_sorad = as.factor(Solar_rad_Wm2_interpol > middle_sorad),
               # temperature
               middle_temp = median(temp_C_interpol),
               side_temp = as.factor(temp_C_interpol > middle_temp),
               # absolute humidity
               middle_absh = median(abs_humidity_g_m3_interpol),
               side_absh = as.factor(abs_humidity_g_m3_interpol > middle_absh),
               middle_ct = median(cloacal_temp_C),
               side_ct = as.factor(cloacal_temp_C > middle_ct)
  )

# now run test
bf.test(formula = e ~ side_absh, # y~x
        data = bf_data_CEWL, # dataframe)
```

```

    alpha = 0.05, # default 0.05
    na.rm = TRUE, # remove missing data before running?
    verbose = TRUE # print output to console?
  )

##
## Brown-Forsythe Test (alpha = 0.05)
## -----
## data : e and side_absh
##
## statistic : 0.08139769
## num df : 1
## denom df : 615.6298
## p.value : 0.7755081
##
## Result : Difference is not statistically significant.
## -----

bf.test(formula = e ~ side_temp, # y~x
  data = bf_data_CEWL, # dataframe
  alpha = 0.05, # default 0.05
  na.rm = TRUE, # remove missing data before running?
  verbose = TRUE # print output to console?
)

##
## Brown-Forsythe Test (alpha = 0.05)
## -----
## data : e and side_temp
##
## statistic : 0.03237498
## num df : 1
## denom df : 585.0611
## p.value : 0.8572694
##
## Result : Difference is not statistically significant.
## -----

bf.test(formula = e ~ side_sorad, # y~x
  data = bf_data_CEWL, # dataframe
  alpha = 0.05, # default 0.05
  na.rm = TRUE, # remove missing data before running?
  verbose = TRUE # print output to console?
)

##
## Brown-Forsythe Test (alpha = 0.05)
## -----
## data : e and side_sorad
##
## statistic : 0.0762158
## num df : 1
## denom df : 626.0987
## p.value : 0.7825838
##
## Result : Difference is not statistically significant.

```

```
## -----
bf.test(formula = e ~ side_ct, # y~x
        data = bf_data_CEWL, # dataframe
        alpha = 0.05, # default 0.05
        na.rm = TRUE, # remove missing data before running?
        verbose = TRUE # print output to console?
        )
```

```
##
## Brown-Forsythe Test (alpha = 0.05)
## -----
## data : e and side_ct
##
## statistic : 0.951633
## num df : 1
## denom df : 575.2982
## p.value : 0.329714
##
## Result : Difference is not statistically significant.
## -----
```

```
bf.test(formula = e ~ side_mass, # y~x
        data = bf_data_CEWL, # dataframe
        alpha = 0.05, # default 0.05
        na.rm = TRUE, # remove missing data before running?
        verbose = TRUE # print output to console?
        )
```

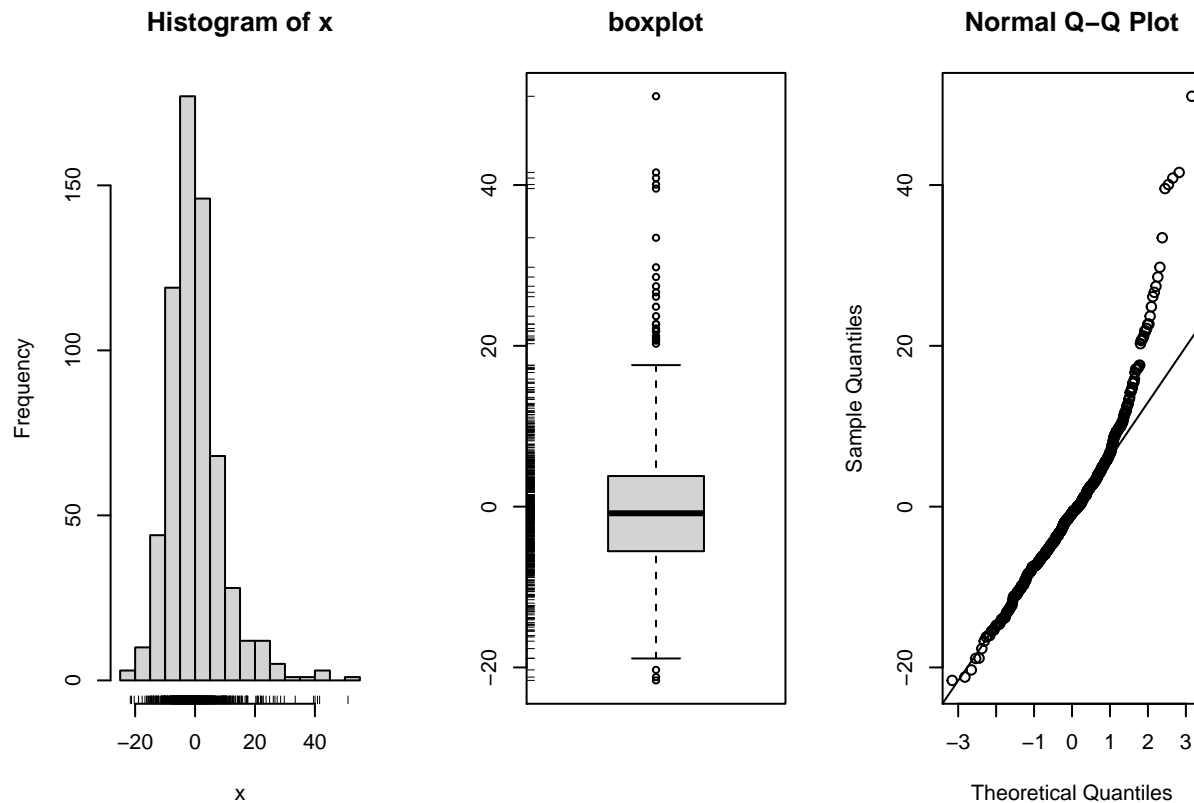
```
##
## Brown-Forsythe Test (alpha = 0.05)
## -----
## data : e and side_mass
##
## statistic : 0.1110193
## num df : 1
## denom df : 627.9847
## p.value : 0.7390978
##
## Result : Difference is not statistically significant.
## -----
```

Equal variance is satisfied for all 5 continuous predictor variables.

Now check normality. Is the distribution of residuals **normal**?

use Shapiro-Wilk normality test: H0: data is NOT significantly different from normal distribution HA: data IS significantly different from normal distribution

```
simple.eda(CEWL_mod_res$e)
```



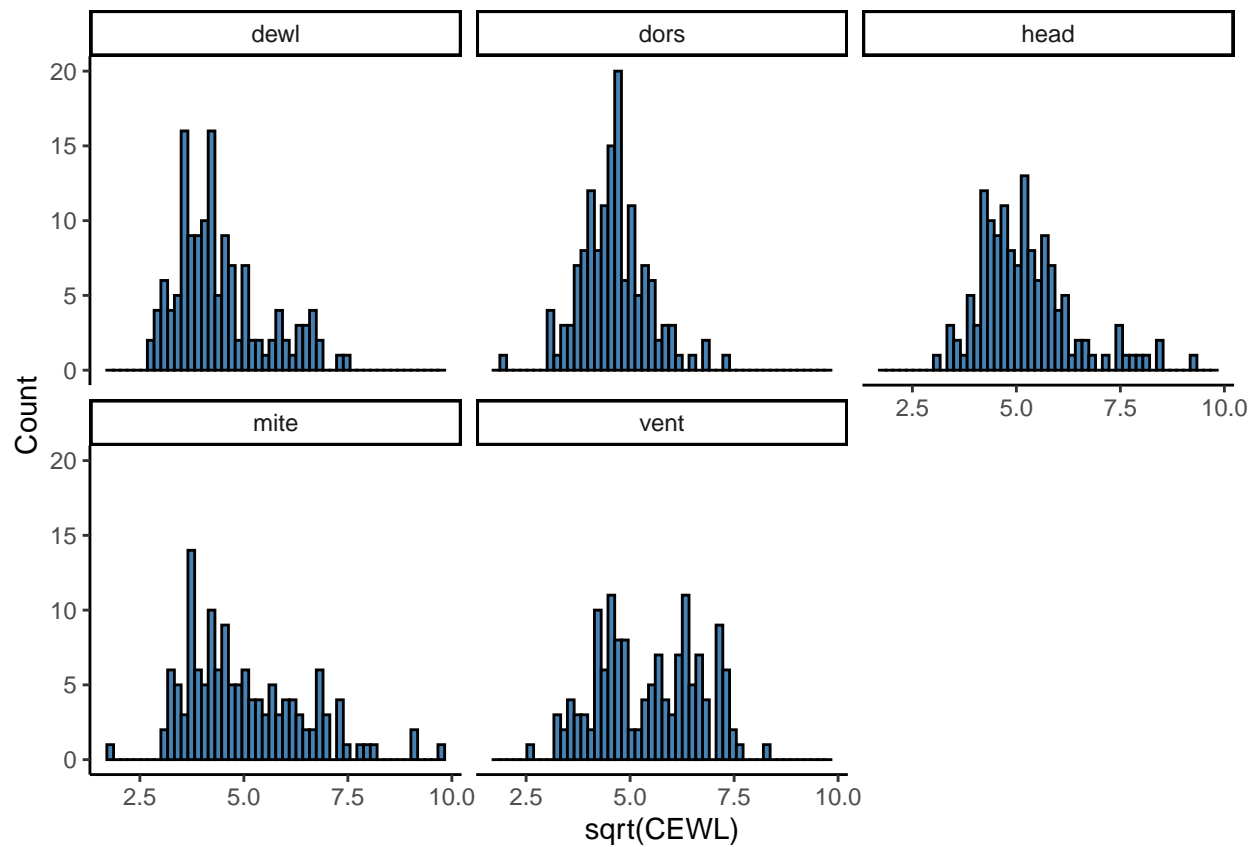
```
shapiro.test(CEWL_mod_res$res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  CEWL_mod_res$res
## W = 0.92375, p-value < 2.2e-16
not normal!
```

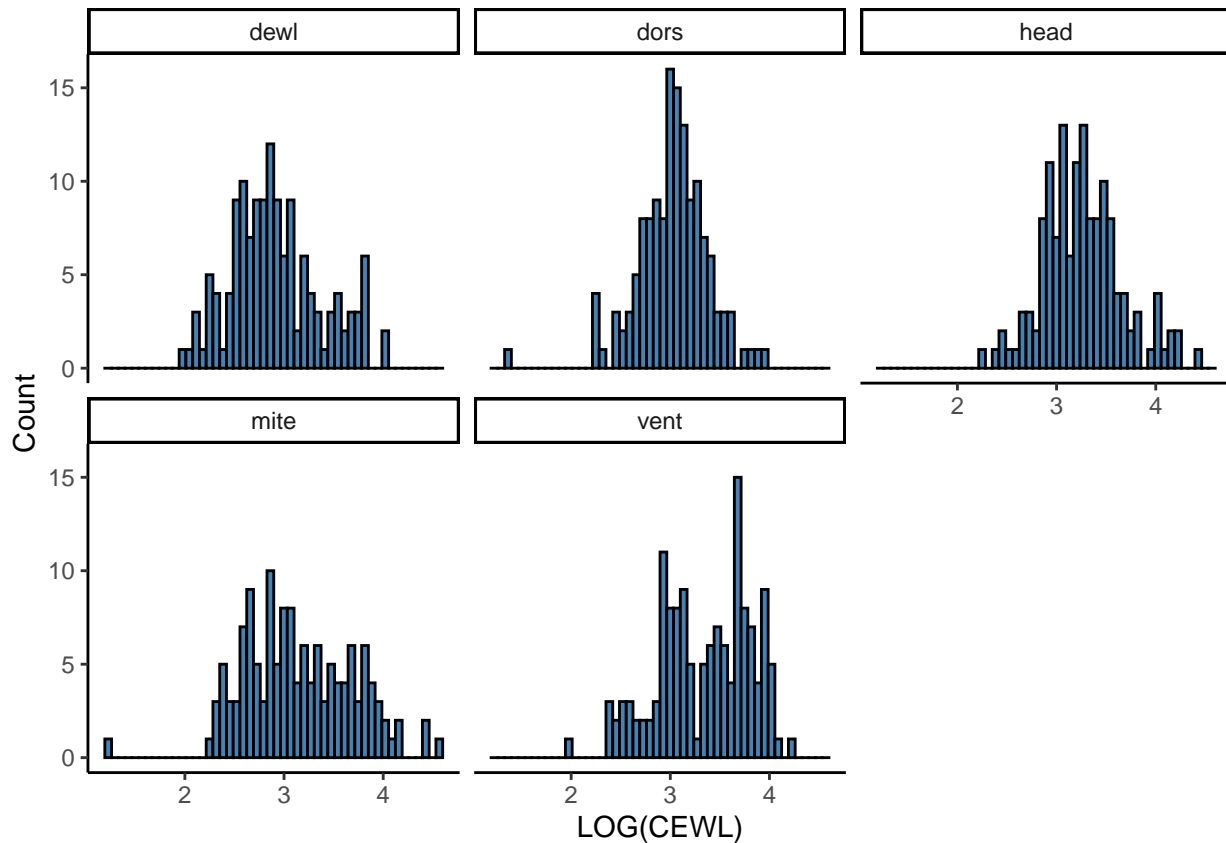
Test Transformations

Can I improve satisfaction of LM assumptions by transforming the dependent variable?

```
# sqrt(TEWL)
CEWL_data_full %>%
  ggplot(., aes(x = sqrt(TEWL_g_m2h))) +
  geom_histogram(color = "black", fill="steelblue", bins=50) +
  theme_classic() +
  xlab("sqrt(CEWL)") +
  ylab("Count") +
  facet_wrap(~region)
```



```
# log(temperature)
CEWL_data_full %>%
  ggplot(., aes(x = log(TEWL_g_m2h))) +
  geom_histogram(color = "black", fill="steelblue", bins=50) +
  theme_classic() +
  xlab("LOG(CEWL)") +
  ylab("Count") +
  facet_wrap(~region)
```



Log transforming seems to be pretty effective across body regions.

Transform & Re-Model

I will log-transform CEWL and see whether it makes the models satisfy LMM assumptions better.

Run CEWL model 4 with log-transformed CEWL:

```
# log-transformed model 3
CEWL_mod3_t <- lme4::lmer(data = CEWL_mod_dat,
                          log(TEWL_g_m2h) ~
                            region * (mass_g +
                                      temp_C_interpol +
                                      abs_humidity_g_m3_interpol) +
                            cloacal_temp_C + Solar_rad_Wm2_interpol +
                            (1|individual_ID))
summary(CEWL_mod3_t)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## log(TEWL_g_m2h) ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
##   cloacal_temp_C + Solar_rad_Wm2_interpol + (1 | individual_ID)
##   Data: CEWL_mod_dat
##
## REML criterion at convergence: 638.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.3124 -0.5294 -0.0337  0.4981  3.8402
```



```
##
## Random effects:
##   Groups      Name      Variance Std.Dev.
## individual_ID (Intercept) 0.0434  0.2083
## Residual          0.1078  0.3283
## Number of obs: 630, groups: individual_ID, 128
##
## Fixed effects:
##                                     Estimate Std. Error t value
## (Intercept)                      -2.4491457  0.6712670  -3.649
## regiondors                        2.1145282  0.7695182   2.748
## regionhead                        2.2521822  0.7673448   2.935
## regionmite                        4.1254137  0.7758522   5.317
## regionvent                        2.6094520  0.7660117   3.407
## mass_g                           0.0635451  0.0137578   4.619
## temp_C_interpol                   0.0380952  0.0148088   2.572
## abs_humidity_g_m3_interpol        0.1505318  0.0577753   2.605
## cloacal_temp_C                    0.0865716  0.0137910   6.277
## Solar_rad_Wm2_interpol             0.0004175  0.0001799   2.320
## regiondors:mass_g                 -0.0422830  0.0164664  -2.568
## regionhead:mass_g                 -0.0579755  0.0161973  -3.579
## regionmite:mass_g                  0.0011536  0.0167446   0.069
## regionvent:mass_g                 -0.0172561  0.0163107  -1.058
## regiondors:temp_C_interpol        -0.0905381  0.0162139  -5.584
## regionhead:temp_C_interpol        -0.0640707  0.0162917  -3.933
## regionmite:temp_C_interpol        -0.0274852  0.0162453  -1.692
## regionvent:temp_C_interpol        -0.0455998  0.0163389  -2.791
## regiondors:abs_humidity_g_m3_interpol  0.0125548  0.0646434   0.194
## regionhead:abs_humidity_g_m3_interpol -0.0119491  0.0644646  -0.185
## regionmite:abs_humidity_g_m3_interpol -0.3212611  0.0652199  -4.926
## regionvent:abs_humidity_g_m3_interpol -0.1073542  0.0644585  -1.665
##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it
# compare
anova(CEWL_mod3_t, CEWL_mod3)

## refitting model(s) with ML (instead of REML)
## Data: CEWL_mod_dat
## Models:
## CEWL_mod3_t: log(TEWL_g_m2h) ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
## CEWL_mod3_t:      cloacal_temp_C + Solar_rad_Wm2_interpol + (1 | individual_ID)
## CEWL_mod3: TEWL_g_m2h ~ region * (mass_g + temp_C_interpol + abs_humidity_g_m3_interpol) +
## CEWL_mod3:      cloacal_temp_C + Solar_rad_Wm2_interpol + (1 | individual_ID)
##      npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## CEWL_mod3_t   24  549.8  656.5  -250.92    501.8
## CEWL_mod3     24 4805.9 4912.6 -2378.96   4757.9     0  0
```

Re-Check Assumptions (transformed model 3)

First, get residuals:

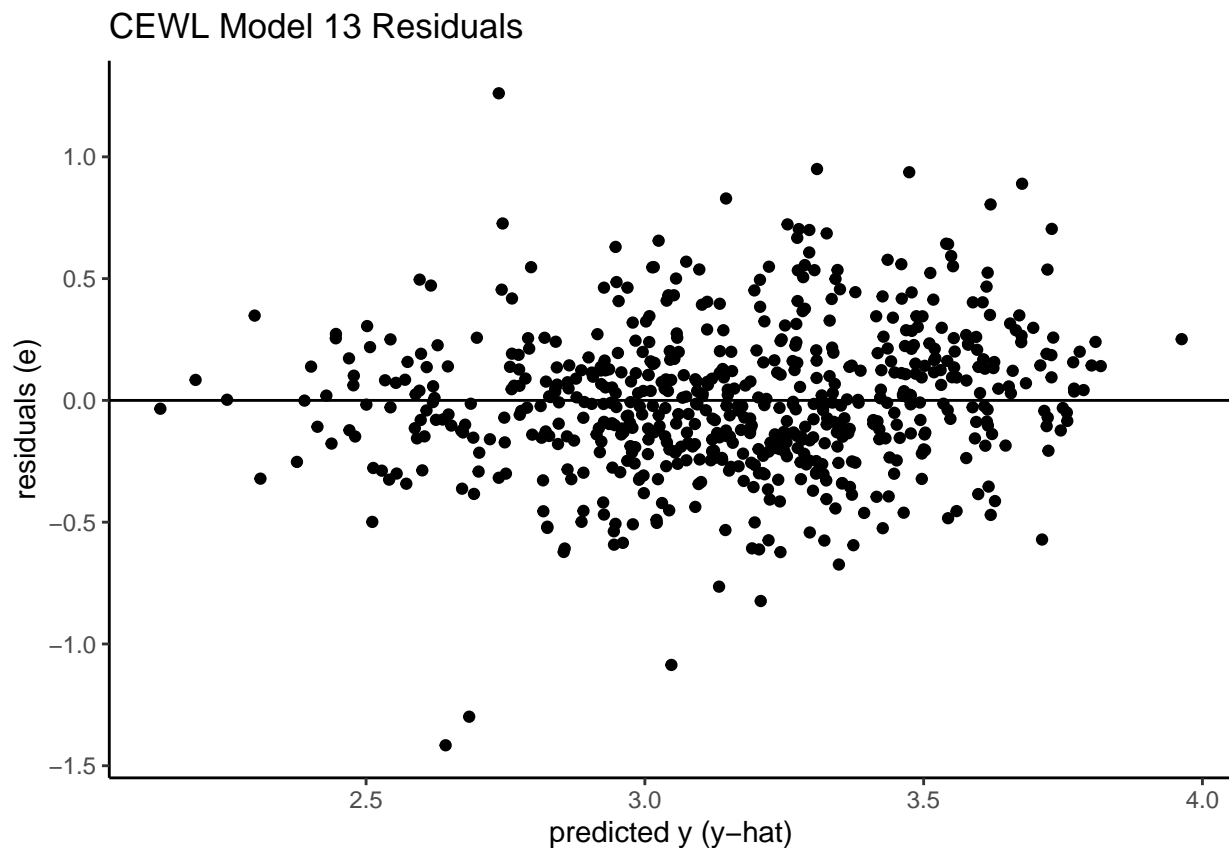
```
CEWL_t_mod_res <- CEWL_mod_dat %>%
  mutate(y_hat = predict(CEWL_mod3_t),
         e = residuals(CEWL_mod3_t))
```

Linearity and Equal Variance

Is the function **linear**? Is there **equal** variance of the residuals? The residuals should be homoskedastic relative to y_{hat} (or x). We don't care if there is a relationship between the residuals ~ dependent variable (actual y).

Plotting residuals shows us whether the data meets linearity and equal variance assumptions:

```
ggplot(data = CEWL_t_mod_res, aes(x = y_hat, y = e)) +
  geom_point() +
  theme_classic() +
  xlab("predicted y (y-hat)") +
  ylab("residuals (e)") +
  ggtitle("CEWL Model 13 Residuals") +
  geom_hline(yintercept = 0)
```



It looks much much better. :) Equal error variance doesn't look perfect, but there's no more fan shape.

Brown-Forsythe test to statistically check equal variance:

H0: normally distributed (non-sig test is GOOD) HA: NOT normally distributed (reject nul == assumption not satisfied)

```
# need to create the right data & format first
bf_data_CEWL_t <- CEWL_t_mod_res %>%
  dplyr::mutate(middle_mass = median(mass_g), # mass
```

```

side_mass = as.factor(mass_g > middle_mass),
# solar radiation
middle_sorad = median(Solar_rad_Wm2_interpol),
side_sorad = as.factor(Solar_rad_Wm2_interpol > middle_sorad),
# temperature
middle_temp = median(temp_C_interpol),
side_temp = as.factor(temp_C_interpol > middle_temp),
# absolute humidity
middle_absh = median(abs_humidity_g_m3_interpol),
side_absh = as.factor(abs_humidity_g_m3_interpol > middle_absh),
middle_ct = median(cloacal_temp_C),
side_ct = as.factor(cloacal_temp_C > middle_ct)
)

# now run test
bf.test(formula = e ~ side_absh, # y~x
data = bf_data_CEWL_t, # dataframe
alpha = 0.05, # default 0.05
na.rm = TRUE, # remove missing data before running?
verbose = TRUE # print output to console?
)

```

```

##
## Brown-Forsythe Test (alpha = 0.05)
## -----
## data : e and side_absh
##
## statistic : 0.1208918
## num df : 1
## denom df : 626.2323
## p.value : 0.7281862
##
## Result : Difference is not statistically significant.
## -----

```

```

bf.test(formula = e ~ side_temp, # y~x
data = bf_data_CEWL_t, # dataframe
alpha = 0.05, # default 0.05
na.rm = TRUE, # remove missing data before running?
verbose = TRUE # print output to console?
)

```

```

##
## Brown-Forsythe Test (alpha = 0.05)
## -----
## data : e and side_temp
##
## statistic : 0.04963674
## num df : 1
## denom df : 579.6971
## p.value : 0.8237751
##
## Result : Difference is not statistically significant.
## -----

```

```
bf.test(formula = e ~ side_sorad, # y~x
        data = bf_data_CEWL_t, # dataframe
        alpha = 0.05, # default 0.05
        na.rm = TRUE, # remove missing data before running?
        verbose = TRUE # print output to console?
        )
```

```
##
##   Brown-Forsythe Test (alpha = 0.05)
## -----
##   data : e and side_sorad
##
##   statistic   : 0.04627073
##   num df      : 1
##   denom df    : 627.9542
##   p.value     : 0.8297543
##
##   Result      : Difference is not statistically significant.
## -----
```

```
bf.test(formula = e ~ side_ct, # y~x
        data = bf_data_CEWL_t, # dataframe
        alpha = 0.05, # default 0.05
        na.rm = TRUE, # remove missing data before running?
        verbose = TRUE # print output to console?
        )
```

```
##
##   Brown-Forsythe Test (alpha = 0.05)
## -----
##   data : e and side_ct
##
##   statistic   : 0.7705684
##   num df      : 1
##   denom df    : 614.2173
##   p.value     : 0.3803845
##
##   Result      : Difference is not statistically significant.
## -----
```

```
bf.test(formula = e ~ side_mass, # y~x
        data = bf_data_CEWL_t, # dataframe
        alpha = 0.05, # default 0.05
        na.rm = TRUE, # remove missing data before running?
        verbose = TRUE # print output to console?
        )
```

```
##
##   Brown-Forsythe Test (alpha = 0.05)
## -----
##   data : e and side_mass
##
##   statistic   : 0.1630792
##   num df      : 1
##   denom df    : 623.8621
```

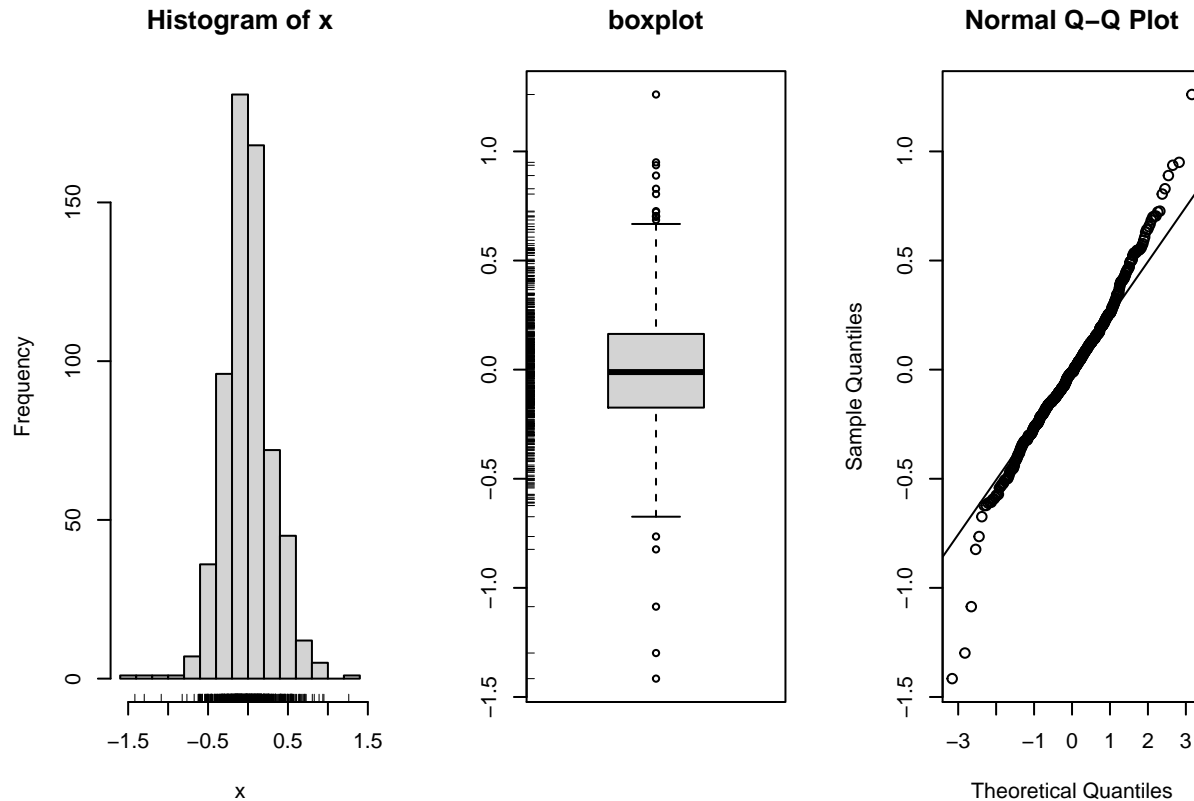
```
## p.value      : 0.6864756
##
## Result       : Difference is not statistically significant.
## -----
```

Equal variance is still satisfied for all continuous predictor variables.

Now check normality. Is the distribution of residuals **normal**?

use Shapiro-Wilk normality test: H0: data is NOT significantly different from normal distribution HA: data IS significantly different from normal distribution

```
simple.eda(CEWL_t_mod_res$e)
```



```
shapiro.test(CEWL_t_mod_res$e)
```

```
##
## Shapiro-Wilk normality test
##
## data:  CEWL_t_mod_res$e
## W = 0.98182, p-value = 4.674e-07
```

Still not statistically normal... but the distribution looks a lot better.

Conclusion

The best CEWL model should use log-transformed CEWL because this greatly improves the model based on AIC and it allows the model to satisfy the linearity assumption of LMM.

```
# save log model 3 summary object
CEWL_best_t_mod <- summary(CEWL_mod3_t)
# extract stats table from summary object
```

```
CEWL_best_t_mod_vals <- data.frame(CEWL_best_t_mod$coefficients)
# export
write.csv(CEWL_best_t_mod_vals, "CEWL_best_mod_vals.csv")
```

What to Present in the Paper

- figures (exported)
- hct SLR
- best osml mod (transformed?)
- best CEWL mod (CEWL transformed)