Climate Water Loss Experiment - General Data Wrangling

Savannah Weaver

2021

Contents

Packages 1

Background and Goals 1

Load Data 2
- Variable Summary 2
- Lizard Data 2
- Join Osml & CEWL Data 4

Compute Scaled Mass Index 5
- Step 1: Simple Linear Regression 6
- Step 2: Identify Outliers 6
- Step 3: log-log Regression 10
- Step 4: Extract Values 11
- Step 5: Calculate Scaled Mass Index 11

Capture Data 11
- Split from Full 11
- Get & Join Weather 13
- Export 15

Full Data 15
- Format 15
- Export 16

Packages

```r
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse") # workflow and plots
if (!require("zoo")) install.packages("zoo")
library("zoo") # interpolation using na.approx (weather data)
if (!require("weathermetrics")) install.packages("weathermetrics")
library("weathermetrics") # F to C conversion (weather data)
```

Background and Goals

This data was collected June - August by Master's student Savannah Weaver, advisor Dr. Emily Taylor, and research assistants Tess McIntyre and Taylor Van Rossum. Adult male *Sceloporus occidentalis* were caught across the Cal Poly campus and in Poly Canyon. This R file compiles and formats the measurements taken. Please refer to the published scientific journal article for full details.

Load Data

Variable Summary

- measurement_date = date measurements were taken, including capture day
- collection/capture time for each lizard
- time_processed = when mass and blood draw were recorded
- time_c_temp = the time when cloacal temperature was recorded, immediately after CEWL measurements
- type = whether measurements were during experiment (exp) or after rehydration (post-rehab). For this R script/analysis, I'm only going to use capture day data, which is listed as "exp"
- day = whether measurements are from capture day or post-experiment, which was recorded in relation to CEWL & cloacal temp data. All observations used for this analysis will be from capture day
- individual ID for each lizard
- mass in grams
- hemolyzed = whether or not red blood cells burst and contaminated plasma
- hematocrit_percent = percent of blood that's red blood cells (measured in CRITOCAP microhematocrit capillary tubes)
- osmolality_mmol_kg_mean = the mean of 1-3 technical replicates of plasma osmolality measurements taken from plasma extracted from our blood samples and run on a VAPRO vapor pressure osmometer
- CEWL_g_m2h_mean = the mean of 3-5 technical replicates, after outliers were omitted, of CEWL measurements taken in the same area of the dorsum
- cloacal_temp_C = cloacal temperature recorded immediately after CEWL measurements
- capture_date = date of capture. For this dataset, it should be the same as measurement date
- day_n = numeric day of measurement. In this dataset, it should always be zero
- capture_date_time = combination of capture date and time
- SVL_mm = snout-to-vent length in mm

Lizard Data

```
lizard_dat <- read.csv("./data/mass_hct_notes.csv", # filename
                             na.strings=c("","NA") # fix empty cells
                             ) %>%
  # join with experiment treatment assignment data
  left_join(read.csv("./data/tmt_assignments.csv"),
            by = "individual_ID") %>%
          # fix date format
  mutate(measurement_date = as.Date(measurement_date,
                                        format = "%m/%d/%y"),
          # set categorical variables as factors
         individual_ID = as.factor(individual_ID),
         type = as.factor(type),
         blood_sample_eye = as.factor(blood_sample_eye),
         hemolyzed = as.factor(hemolyzed),
         trial_number = as.factor(trial_number),
         temp_tmt = as.factor(temp_tmt),
         humidity_tmt = as.factor(humidity_tmt),
         tmt = as.factor(paste(temp_tmt, humidity_tmt)),
         conclusion = as.factor(conclusion)
         ) %>%
```

```r
  group_by(individual_ID) %>%
          # for each individual, extract capture date
  mutate(capture_date = min(measurement_date),
         # create "day of experiment" variable, both numeric and factor
         day_n = as.numeric(measurement_date - capture_date),
         day_factor = as.factor(day_n))
summary(lizard_dat)
```

```
##  measurement_date     time_captured         type      individual_ID
##  Min.   :2021-06-16   Length:957         exp   :825   201    :  7
##  1st Qu.:2021-06-30   Class :character   rehab:132    202    :  7
##  Median :2021-07-25   Mode  :character                203    :  7
##  Mean   :2021-07-22                                   204    :  7
##  3rd Qu.:2021-08-14                                   205    :  7
##  Max.   :2021-09-01                                   206    :  7
##                                                       (Other):915
##  blood_sample_eye     mass_g      time_processed     hematocrit_percent
##  L   :  3         Min.   : 7.00   Length:957         Min.   :13.00
##  R   :543         1st Qu.: 9.50   Class :character   1st Qu.:26.00
##  NA's:411         Median :10.60   Mode  :character   Median :32.00
##                   Mean   :10.62                      Mean   :32.09
##                   3rd Qu.:11.60                      3rd Qu.:38.00
##                   Max.   :17.40                      Max.   :52.00
##                                                      NA's   :417
##  hemolyzed    notes.x          trial_number temp_tmt   humidity_tmt
##  N   :475   Length:957         1:179        cool:477   dry  :479
##  Y   : 70   Class :character   2:209        hot :480   humid:478
##  NA's:412   Mode  :character   3:236
##                                4:195
##                                5:138
##
##
##      SVL_mm         conclusion    notes.y              shed
##  Min.   :60.00   canceled: 21   Length:957         Length:957
##  1st Qu.:66.00   complete:936   Class :character   Class :character
##  Median :67.00                  Mode  :character   Mode  :character
##  Mean   :67.73
##  3rd Qu.:70.00
##  Max.   :77.00
##
##  tail_broken            died                    tmt       capture_date
##  Length:957         Length:957         cool dry  :238   Min.   :2021-06-16
##  Class :character   Class :character   cool humid:239   1st Qu.:2021-06-26
##  Mode  :character   Mode  :character   hot dry   :241   Median :2021-07-20
##                                        hot humid :239   Mean   :2021-07-17
##                                                         3rd Qu.:2021-08-08
##                                                         Max.   :2021-08-22
##
##      day_n        day_factor
##  Min.   : 0.000   0 :141
##  1st Qu.: 4.000   4 :138
##  Median : 6.000   5 :138
##  Mean   : 5.658   6 :138
##  3rd Qu.: 8.000   7 :135
```

```
##  Max.   :10.000   8 :135
##                   10:132
```

Join Osml & CEWL Data

```
full_dat <- lizard_dat %>%
  # osmolality data
  left_join(read_rds("./data/osml_means_clean.RDS"),
            by = c("individual_ID", "measurement_date" = "date_blood_drawn")
            ) %>%
  # join CEWL data
  left_join(read_rds("./data/CEWL_dat_all_clean.RDS"),
            by = c("individual_ID", "measurement_date" = "date")
            ) %>%
  # remove some unnecessary variables
              # too few left eye or hemolyzed blood samples to matter statistically
  dplyr::select(-blood_sample_eye, -hemolyzed,
                # notes no longer necessary/useful
                -notes.x, -notes.y, -shed, -tail_broken, -died,
                -time_c_temp, -day
                ) %>%
          # compute vapor pressure deficit
  mutate(msmt_temp_K = msmt_temp_C + 273.15,
         # find saturation level first
         e_s_kPa_m = 0.611*exp((2500000/461.5)*
                                  ((1/273)-(1/msmt_temp_K))),
         # actual vapor pressure
         e_a_kPa_m = e_s_kPa_m * (msmt_RH_percent/100),
         # VPD
         msmt_VPD_kPa = e_s_kPa_m - e_a_kPa_m
         ) %>%
  # remove data for one lizard that was an accidental recapture
  dplyr::filter(individual_ID != 304)

summary(full_dat)
```

```
##  measurement_date     time_captured         type     individual_ID
##  Min.   :2021-06-16   Length:951         exp  :819   201    :  7
##  1st Qu.:2021-06-30   Class :character   rehab:132   202    :  7
##  Median :2021-07-25   Mode  :character               203    :  7
##  Mean   :2021-07-22                                  204    :  7
##  3rd Qu.:2021-08-14                                  205    :  7
##  Max.   :2021-09-01                                  206    :  7
##                                                      (Other):909
##      mass_g      time_processed     hematocrit_percent trial_number temp_tmt
##  Min.   : 7.00   Length:951         Min.   :13.00      1:179        cool:471
##  1st Qu.: 9.50   Class :character   1st Qu.:26.00      2:209        hot :480
##  Median :10.60   Mode  :character   Median :32.00      3:236
##  Mean   :10.62                      Mean   :32.09      4:189
##  3rd Qu.:11.60                      3rd Qu.:38.00      5:138
##  Max.   :17.40                      Max.   :52.00
##                                     NA's   :414
##  humidity_tmt     SVL_mm         conclusion          tmt
##  dry  :473    Min.   :60.00   canceled: 15    cool dry  :232
```

```
##  humid:478    1st Qu.:66.00   complete:936   cool humid:239
##               Median :67.00                  hot dry   :241
##               Mean   :67.73                  hot humid :239
##               3rd Qu.:70.00
##               Max.   :77.00
##
##   capture_date            day_n        day_factor osmolality_mmol_kg_mean
##  Min.   :2021-06-16   Min.   : 0.000   0 :140     Min.   :295.3
##  1st Qu.:2021-06-26   1st Qu.: 4.000   4 :137     1st Qu.:335.8
##  Median :2021-07-20   Median : 6.000   5 :137     Median :351.3
##  Mean   :2021-07-16   Mean   : 5.662   6 :137     Mean   :354.5
##  3rd Qu.:2021-08-08   3rd Qu.: 8.000   7 :134     3rd Qu.:370.0
##  Max.   :2021-08-22   Max.   :10.000   8 :134     Max.   :492.7
##                                        10:132     NA's   :421
##  CEWL_g_m2h_mean   msmt_temp_C    msmt_RH_percent cloacal_temp_C
##  Min.   : 7.152   Min.   :24.80   Min.   :25.52   Min.   :23.00
##  1st Qu.:19.659   1st Qu.:26.27   1st Qu.:46.07   1st Qu.:25.00
##  Median :24.091   Median :26.75   Median :47.80   Median :26.00
##  Mean   :24.709   Mean   :26.73   Mean   :46.67   Mean   :25.93
##  3rd Qu.:28.414   3rd Qu.:27.12   3rd Qu.:50.48   3rd Qu.:27.00
##  Max.   :56.066   Max.   :29.20   Max.   :56.16   Max.   :30.00
##  NA's   :679      NA's   :678     NA's   :678     NA's   :678
##    date_time                      msmt_temp_K      e_s_kPa_m       e_a_kPa_m
##  Min.   :2021-06-16 09:54:00   Min.   :297.9   Min.   :3.219   Min.   :0.9894
##  1st Qu.:2021-06-26 14:03:00   1st Qu.:299.4   1st Qu.:3.518   1st Qu.:1.6467
##  Median :2021-07-20 14:56:00   Median :299.9   Median :3.623   Median :1.7394
##  Mean   :2021-07-21 10:04:24   Mean   :299.9   Mean   :3.623   Mean   :1.6818
##  3rd Qu.:2021-08-08 15:22:00   3rd Qu.:300.3   3rd Qu.:3.704   3rd Qu.:1.7986
##  Max.   :2021-08-30 11:32:00   Max.   :302.4   Max.   :4.194   Max.   :1.9326
##  NA's   :678                   NA's   :678     NA's   :678     NA's   :678
##   msmt_VPD_kPa
##  Min.   :1.486
##  1st Qu.:1.784
##  Median :1.854
##  Mean   :1.941
##  3rd Qu.:2.017
##  Max.   :3.021
##  NA's   :678
```

```
# check
unique(full_dat$capture_date)
```

```
## [1] "2021-06-16" "2021-06-26" "2021-07-20" "2021-08-08" "2021-08-22"
```

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: Simple Linear Regression

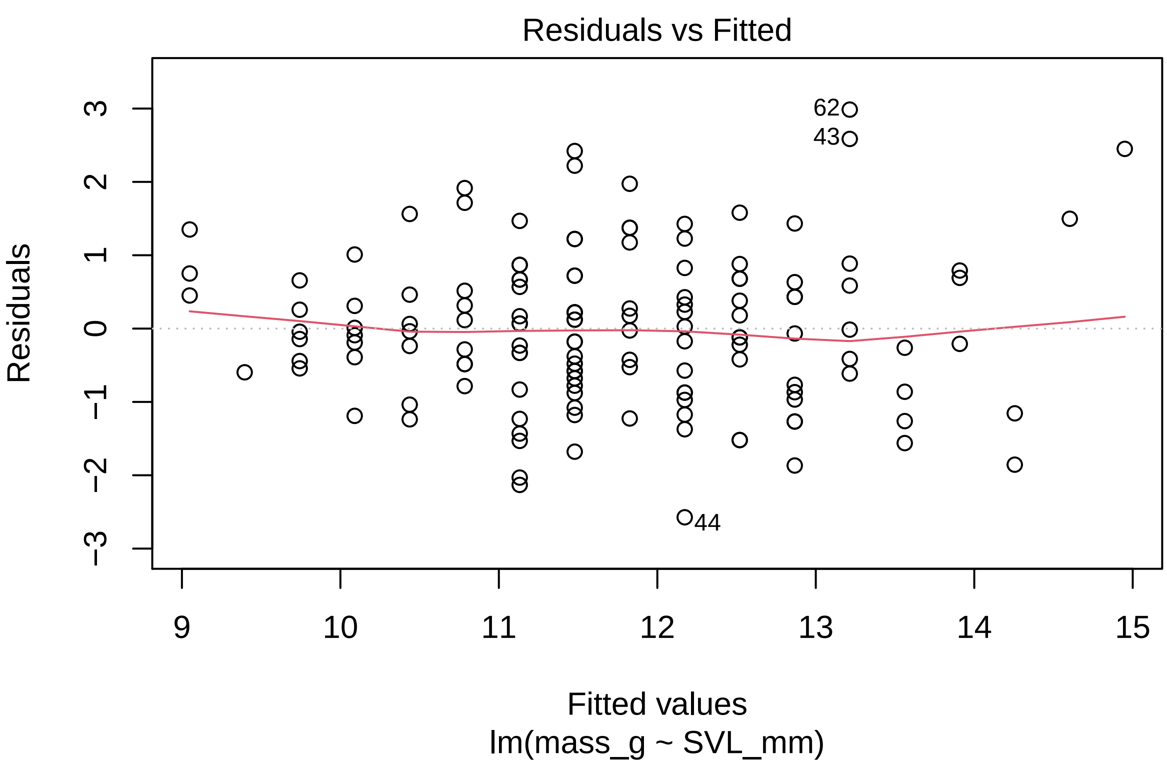
I only use capture mass measurements for these calculations because that's what's representative of body condition naturally.

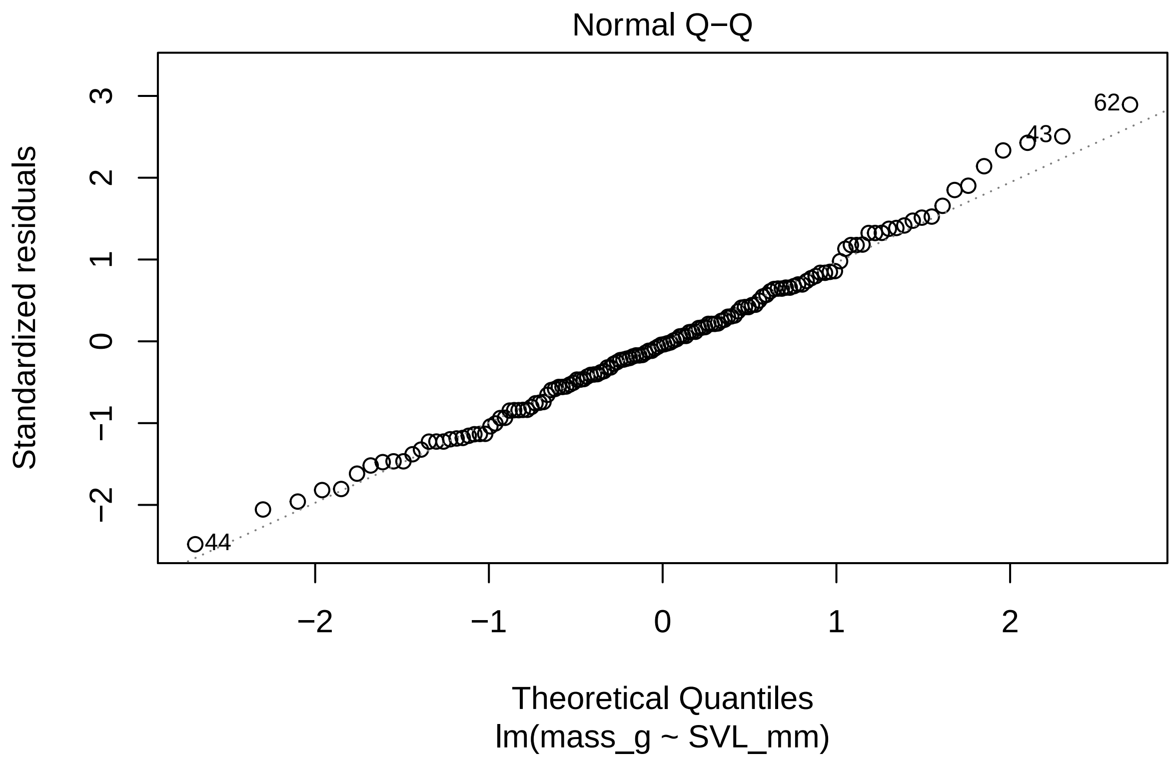
```
SLR_dat <- full_dat %>% dplyr::filter(day_n == 0)
mass_SVL_SLR <- lm(data = SLR_dat, mass_g ~ SVL_mm)
summary(mass_SVL_SLR)
```

```
##
## Call:
## lm(formula = mass_g ~ SVL_mm, data = SLR_dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.57294 -0.70082 -0.04022  0.66842  2.98570
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.7784     1.7481  -6.738 4.01e-10 ***
## SVL_mm        0.3471     0.0258  13.453  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.041 on 138 degrees of freedom
## Multiple R-squared:  0.5674, Adjusted R-squared:  0.5642
## F-statistic:   181 on 1 and 138 DF,  p-value: < 2.2e-16
```

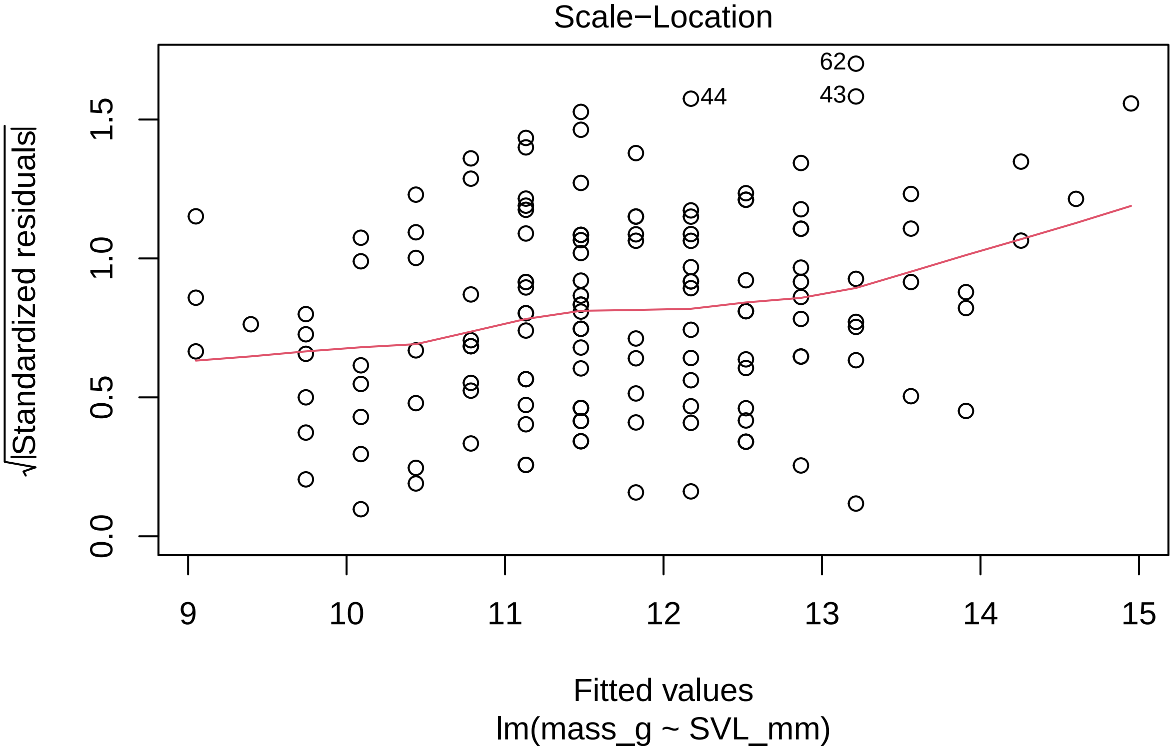
Step 2: Identify Outliers

```
plot(mass_SVL_SLR)
```

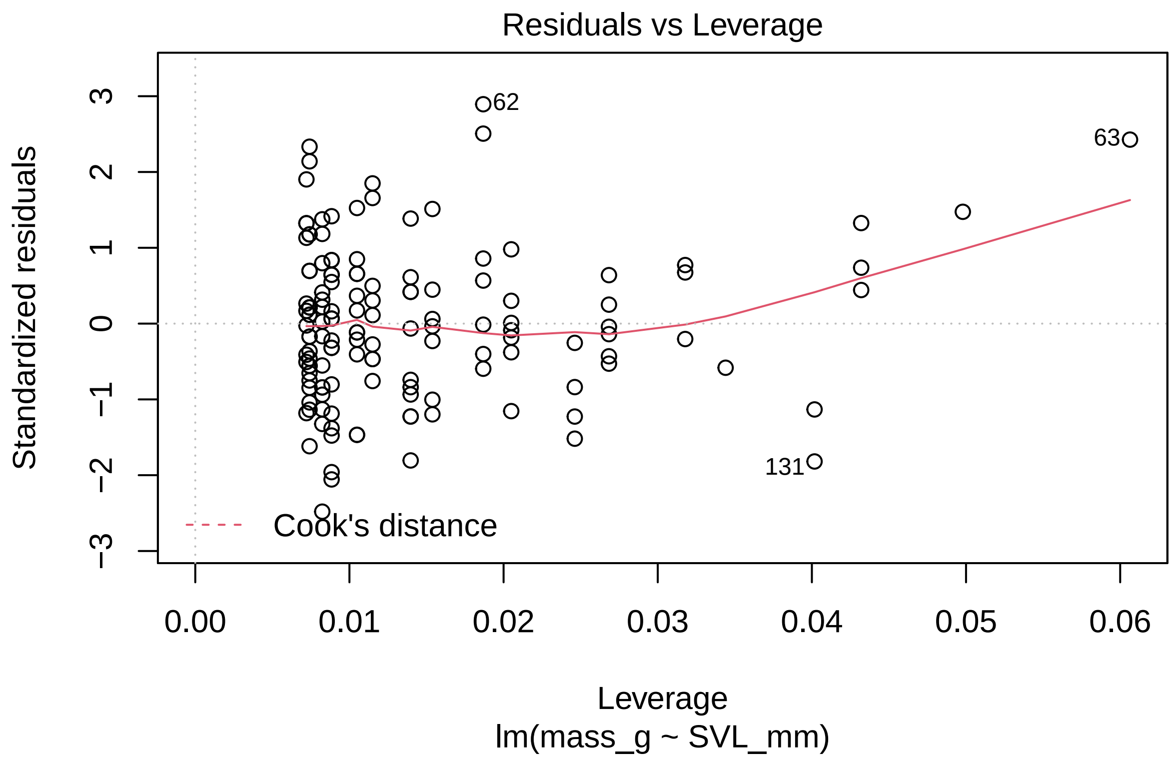




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(mass_g ~ SVL_mm)
44
43
62

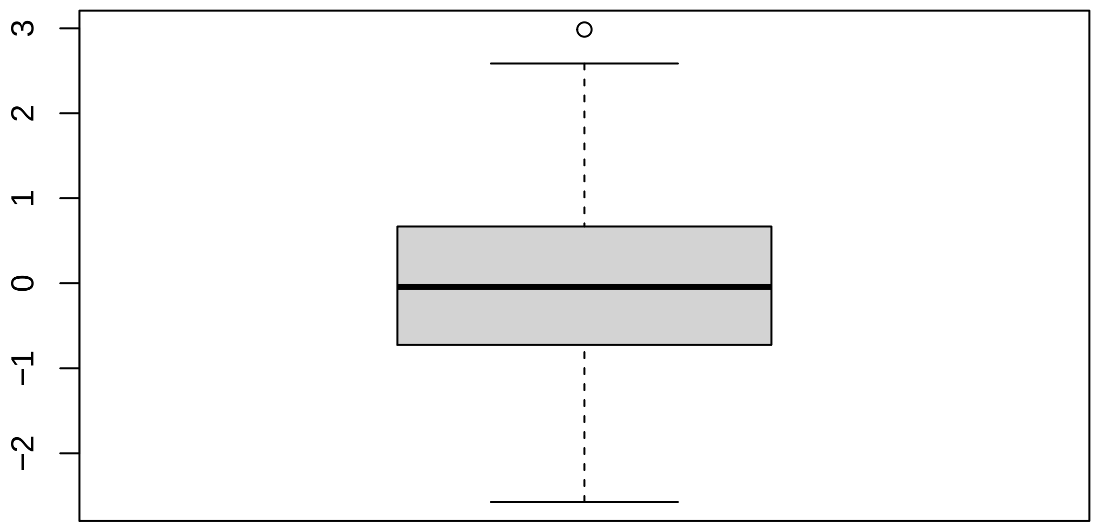


Scale−Location
√|Standardized residuals|
Fitted values
lm(mass_g ~ SVL_mm)
44
43
62

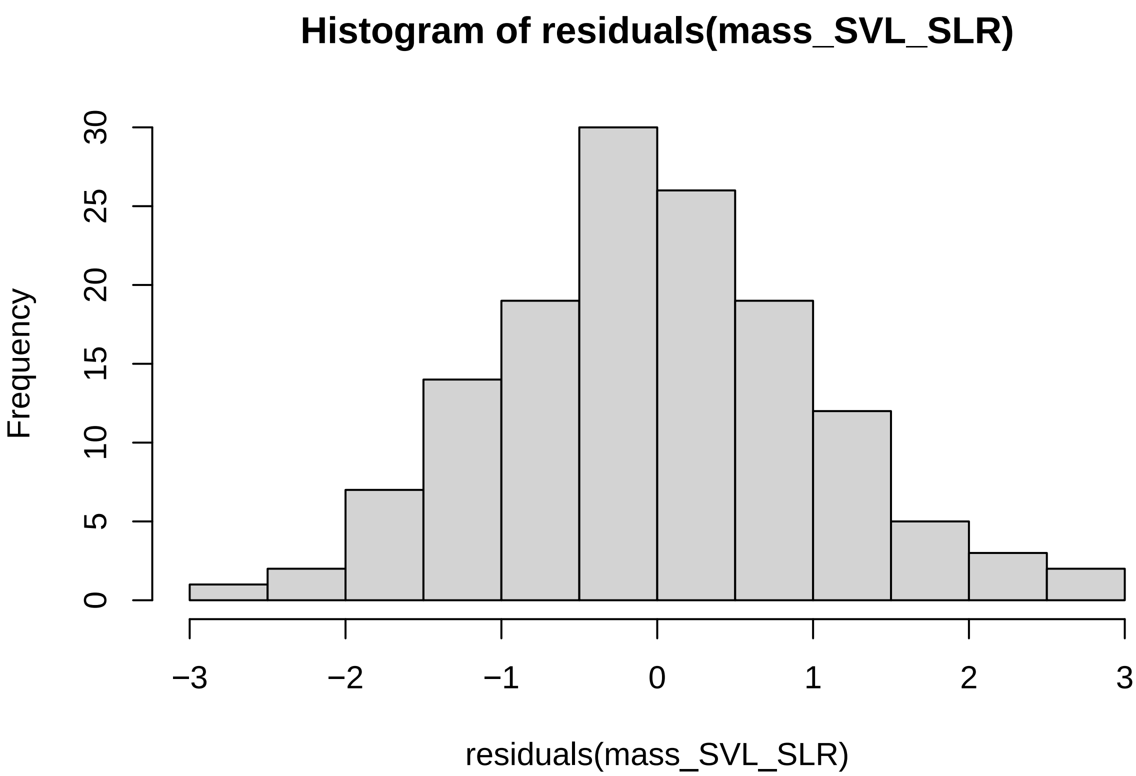


The conditions of linearity, equal error variance, and normality are all satisfied. It doesn't look like any residuals are >3 or <-3.

```
boxplot(residuals(mass_SVL_SLR))
```




```
hist(residuals(mass_SVL_SLR))
```



From the boxplot, there is one individual with a much higher residual than the rest of the distribution. The histogram looks beautiful and incredibly normally distributed.

Check average residual value:

```
mean(residuals(mass_SVL_SLR))
```

```
## [1] 6.468505e-17
```

```
median(residuals(mass_SVL_SLR))
```

```
## [1] -0.04021575
```

The mean is basically zero and the median is pretty close to zero, which is very good.

Check for high leverage points:

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

# 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 <- SLR_dat %>%
  mutate(row = row_number()) %>%
  left_join(., high_leverage, by = "row") %>%
  dplyr::filter(H > h_bar)
high_leverage_dat
```

```
## # A tibble: 0 x 28
```

```
## # Groups:   individual_ID [0]
## # ... with 28 variables: measurement_date <date>, time_captured <chr>,
## #   type <fct>, individual_ID <fct>, mass_g <dbl>, time_processed <chr>,
## #   hematocrit_percent <int>, trial_number <fct>, temp_tmt <fct>,
## #   humidity_tmt <fct>, SVL_mm <int>, conclusion <fct>, tmt <fct>,
## #   capture_date <date>, day_n <dbl>, day_factor <fct>,
## #   osmolality_mmol_kg_mean <dbl>, CEWL_g_m2h_mean <dbl>, msmt_temp_C <dbl>,
## #   msmt_RH_percent <dbl>, cloacal_temp_C <dbl>, date_time <dttm>,
## #   msmt_temp_K <dbl>, e_s_kPa_m <dbl>, e_a_kPa_m <dbl>, msmt_VPD_kPa <dbl>,
## #   row <int>, H <dbl>
```

No points are considered high leverage, which is fantastic.

Check for influential points based on Cook's distance:

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

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

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

```
## # A tibble: 0 x 28
## # Groups:   individual_ID [0]
## # ... with 28 variables: measurement_date <date>, time_captured <chr>,
## #   type <fct>, individual_ID <fct>, mass_g <dbl>, time_processed <chr>,
## #   hematocrit_percent <int>, trial_number <fct>, temp_tmt <fct>,
## #   humidity_tmt <fct>, SVL_mm <int>, conclusion <fct>, tmt <fct>,
## #   capture_date <date>, day_n <dbl>, day_factor <fct>,
## #   osmolality_mmol_kg_mean <dbl>, CEWL_g_m2h_mean <dbl>, msmt_temp_C <dbl>,
## #   msmt_RH_percent <dbl>, cloacal_temp_C <dbl>, date_time <dttm>,
## #   msmt_temp_K <dbl>, e_s_kPa_m <dbl>, e_a_kPa_m <dbl>, msmt_VPD_kPa <dbl>,
## #   row <int>, c <dbl>
```

There are no infuential points based on Cook's distance, so there's nothing to potentially remove.

We could remove the one outlier found using the boxplot, but it's the only one, so we will leave it in the dataset. No points were indicated to be outliers based on residuals or a histogram, and there were no high leverage or influential points. Thus I can create a log-log model using the data as-is. Observation omissions are unlikely to increase generalizability.

Step 3: log-log Regression

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

```
##
## Call:
## lm(formula = log(mass_g) ~ log(SVL_mm), data = SLR_dat)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.230964 -0.062344 -0.001499  0.057523  0.208260
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.8665     0.6133  -9.565   <2e-16 ***
## log(SVL_mm)   1.9743     0.1456  13.563   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08683 on 138 degrees of freedom
## Multiple R-squared:  0.5714, Adjusted R-squared:  0.5683
## F-statistic:   184 on 1 and 138 DF,  p-value: < 2.2e-16
```

Step 4: Extract Values

compute standardized major axis using the log-log regression equation:

```
r <- sqrt(0.5714) # Pearson's correlection coefficient (sqrt of R-squared)
b_OLS <- 1.9743 # regression slope
b_SMA <- b_OLS/r
```

mean length in capture data:

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

Step 5: Calculate Scaled Mass Index

Add SMI to an updated full_dat df - full_dat2

```
full_dat2 <- full_dat %>%
  # compute SMI
  mutate(SMI = mass_g * ((L0/SVL_mm) ^ b_SMA))
```

Capture Data

Split from Full

Extract only the data from capture day (1 row of observations for each individual) for some initial/baseline physiology analyses. The experiment analyses will use the full dataframe.

```
capture_dat <- full_dat2 %>%
          # select only data from capture days
  dplyr::filter(day_n == 0) %>%
  # make sure only complete data included
  dplyr::filter(complete.cases(osmolality_mmol_kg_mean,
                               CEWL_g_m2h_mean, cloacal_temp_C)) %>%
  # make capture_date_time variable
  mutate(capture_date_time = as.POSIXct(paste(capture_date, time_captured,
                                               sep = " "),
                              format = "%Y-%m-%d %H:%M"),
         hold_time_sec = (date_time - capture_date_time),
         hold_time_min = (hold_time_sec)/60,
         hold_time_hr = (hold_time_min)/60
         ) %>%
```

```r
  # remove experiment variables not relevant to capture analysis
  dplyr::select(-type, -trial_number, -temp_tmt, -humidity_tmt, -tmt,
                -conclusion, -day_n, -day_factor,
                # redundant variables
                -time_captured, -time_processed,
                -measurement_date)
summary(capture_dat)
```

```
##  individual_ID     mass_g       hematocrit_percent     SVL_mm
##  201    :  1   Min.   : 8.80   Min.   :27.00      Min.   :60.00
##  202    :  1   1st Qu.:10.60   1st Qu.:34.25      1st Qu.:66.00
##  203    :  1   Median :11.65   Median :39.00      Median :67.00
##  204    :  1   Mean   :11.73   Mean   :38.93      Mean   :67.71
##  205    :  1   3rd Qu.:12.70   3rd Qu.:43.00      3rd Qu.:70.00
##  206    :  1   Max.   :17.40   Max.   :52.00      Max.   :77.00
##  (Other):132
##   capture_date        osmolality_mmol_kg_mean CEWL_g_m2h_mean   msmt_temp_C
##  Min.   :2021-06-16   Min.   :305.0           Min.   : 7.152   Min.   :25.90
##  1st Qu.:2021-06-26   1st Qu.:334.3           1st Qu.:17.255   1st Qu.:26.72
##  Median :2021-07-20   Median :344.6           Median :21.030   Median :26.96
##  Mean   :2021-07-16   Mean   :348.3           Mean   :20.760   Mean   :27.20
##  3rd Qu.:2021-08-08   3rd Qu.:361.9           3rd Qu.:24.416   3rd Qu.:27.50
##  Max.   :2021-08-22   Max.   :395.0           Max.   :34.660   Max.   :29.20
##
##  msmt_RH_percent cloacal_temp_C    date_time                     msmt_temp_K
##  Min.   :25.52   Min.   :25.00   Min.   :2021-06-16 09:54:00   Min.   :299.1
##  1st Qu.:45.77   1st Qu.:26.00   1st Qu.:2021-06-26 12:59:30   1st Qu.:299.9
##  Median :47.09   Median :26.00   Median :2021-07-20 13:17:00   Median :300.1
##  Mean   :44.08   Mean   :26.45   Mean   :2021-07-17 06:56:12   Mean   :300.3
##  3rd Qu.:48.44   3rd Qu.:27.00   3rd Qu.:2021-08-08 13:39:00   3rd Qu.:300.6
##  Max.   :53.15   Max.   :30.00   Max.   :2021-08-22 15:19:00   Max.   :302.4
##
##    e_s_kPa_m       e_a_kPa_m       msmt_VPD_kPa        SMI
##  Min.   :3.441   Min.   :0.9894   Min.   :1.612   Min.   : 9.122
##  1st Qu.:3.616   1st Qu.:1.6913   1st Qu.:1.846   1st Qu.:10.926
##  Median :3.669   Median :1.7342   Median :1.942   Median :11.687
##  Mean   :3.724   Mean   :1.6312   Mean   :2.093   Mean   :11.690
##  3rd Qu.:3.790   3rd Qu.:1.7865   3rd Qu.:2.053   3rd Qu.:12.347
##  Max.   :4.194   Max.   :1.8502   Max.   :3.021   Max.   :14.263
##
##  capture_date_time             hold_time_sec     hold_time_min
##  Min.   :2021-06-16 08:28:00   Length:138        Length:138
##  1st Qu.:2021-06-26 09:44:45   Class :difftime   Class :difftime
##  Median :2021-07-20 09:52:00   Mode  :numeric    Mode  :numeric
##  Mean   :2021-07-14 14:50:11
##  3rd Qu.:2021-08-08 09:56:45
##  Max.   :2021-08-22 13:25:00
##  NA's   :14
##  hold_time_hr
##  Length:138
##  Class :difftime
##  Mode  :numeric
##
##
```

```
##
##
```

Also do some capture-based summary stats for permit reporting:

```
permit_stats <- capture_dat %>%
  group_by(capture_date) %>%
  summarise(n = n()) %>%
  mutate(sex = "M")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
permit_stats
```

```
## # A tibble: 5 x 3
##   capture_date     n sex
##   <date>       <int> <chr>
## 1 2021-06-16      26 M
## 2 2021-06-26      31 M
## 3 2021-07-20      34 M
## 4 2021-08-08      27 M
## 5 2021-08-22      20 M
```

```
sum(permit_stats$n)
```

```
## [1] 138
```

```
write.csv(permit_stats, "./data/collection_summary.csv")
```

Get & Join Weather

This data was obtained from http://www.itrc.org/databases/precip/ (Adcon Server Data) to test the effect of ambient conditions on CEWL.

Load and format:

```
weather <- read.csv("./data/weather.csv", sep = ';') %>%
         # add a variable for combined date-time
  mutate(capture_date_time = as.POSIXct(paste(date, time),
                                      format = "%m/%d/%y %I:%M %p"))
```

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 on each capture day.

```
all_times <- data.frame(capture_date_time = c(
                           # June 16
                           seq(from = as.POSIXct("2021-06-16 07:00"),
                               to = as.POSIXct("2021-06-16 19:00"),
                               by="min"),
                           # June 26
                           seq(from = as.POSIXct("2021-06-26 07:00"),
                               to = as.POSIXct("2021-06-26 19:00"),
                               by="min"),
                           # July 20
                           seq(from = as.POSIXct("2021-07-20 07:00"),
                               to = as.POSIXct("2021-07-20 19:00"),
                               by="min"),
```

```
                          # August 8
                          seq(from = as.POSIXct("2021-08-08 07:00"),
                              to = as.POSIXct("2021-08-08 19:00"),
                              by="min"),
                          # August 22
                          seq(from = as.POSIXct("2021-08-22 07:00"),
                              to = as.POSIXct("2021-08-22 19:00"),
                              by="min")
                          ))
```

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

```
weather_every_minute <- all_times %>% # time only dataframe
  # add weather measurements based on matching date-time
  left_join(weather, by = 'capture_date_time') %>%
         # convert temperature units F->C
  mutate(temp_C = fahrenheit.to.celsius(temperature_F, round = 2),
         # interpolate temperatures
         temp_C_interpol = na.approx(temp_C),
         # also get temperature C-> K
         temp_K_interpol = temp_C_interpol + 273.15,
         # interpolate humidities
         RH_percent_interpol = na.approx(relative_humidity_percent),
         # interpolate Wind Speeds
         wind_mph_interpol = na.approx(wind_speed_mph),
         # interpolate solar radiation
         solar_rad_W_sqm_interpol = na.approx(solar_radiation_W_sqm),
         # compute vapor pressure deficit
         # find saturation level first
         e_s_kPa_int = 0.611*exp((2500000/461.5)*
                            ((1/273)-(1/temp_K_interpol))),
         # actual vapor pressure
         e_a_kPa_int = e_s_kPa_int * (RH_percent_interpol/100),
         # VPD
         VPD_kPa_int = e_s_kPa_int - e_a_kPa_int
         ) %>%
  # keep only the relevant variables
  dplyr::select(capture_date_time,
                temp_C_interpol,
                RH_percent_interpol,
                VPD_kPa_int,
                wind_mph_interpol,
                solar_rad_W_sqm_interpol)
summary(weather_every_minute)
```

```
##  capture_date_time              temp_C_interpol RH_percent_interpol
##  Min.   :2021-06-16 07:00:00   Min.   :12.50   Min.   : 16.50
##  1st Qu.:2021-06-26 10:00:00   1st Qu.:20.04   1st Qu.: 56.83
##  Median :2021-07-20 13:00:00   Median :22.35   Median : 67.10
##  Mean   :2021-07-19 08:12:00   Mean   :23.22   Mean   : 63.15
##  3rd Qu.:2021-08-08 16:00:00   3rd Qu.:25.17   3rd Qu.: 76.13
##  Max.   :2021-08-22 19:00:00   Max.   :38.33   Max.   :100.00
##   VPD_kPa_int     wind_mph_interpol solar_rad_W_sqm_interpol
##  Min.   :0.0000   Min.   : 0.100    Min.   :  13.6
```

```
##  1st Qu.:0.5724   1st Qu.: 2.800    1st Qu.: 370.0
##  Median :0.9074   Median : 4.700    Median : 699.6
##  Mean   :1.4591   Mean   : 4.820    Mean   : 624.2
##  3rd Qu.:1.4235   3rd Qu.: 5.833    3rd Qu.: 902.6
##  Max.   :5.8841   Max.   :13.600    Max.   :1011.7
```

add the weather data in:

```
capture_dat2 <- capture_dat %>%
  left_join(weather_every_minute, by = 'capture_date_time')
```

Export

```
write_rds(capture_dat2, "./data/analysis_data_capture.RDS")
```

Full Data

Format

Remove data for individuals with canceled experimental treatments:

```
full_dat3 <- full_dat2 %>%
  dplyr::filter(conclusion == "complete") %>%
  dplyr::select(-time_captured, -time_processed,
                -capture_date, -date_time,
                -conclusion)
```

Rename some factors:

```
full_dat3$humidity_tmt <- factor(full_dat3$humidity_tmt,
                         levels = c("humid", "dry"),
                         labels = c("Humid", "Dry"))
full_dat3$temp_tmt <- factor(full_dat3$temp_tmt,
                         levels = c("hot", "cool"),
                         labels = c("Hot", "Cool"))
full_dat3$tmt <- factor(full_dat3$tmt,
                         levels = c("hot humid", "hot dry",
                                    "cool humid", "cool dry"),
                         labels = c("Hot Humid", "Hot Dry",
                                    "Cool Humid", "Cool Dry"))
summary(full_dat3)
```

```
##  measurement_date        type     individual_ID     mass_g
##  Min.   :2021-06-16   exp  :804   201    :  7   Min.   : 7.00
##  1st Qu.:2021-07-01   rehab:132   202    :  7   1st Qu.: 9.50
##  Median :2021-07-25               203    :  7   Median :10.60
##  Mean   :2021-07-22               204    :  7   Mean   :10.64
##  3rd Qu.:2021-08-14               205    :  7   3rd Qu.:11.70
##  Max.   :2021-09-01               206    :  7   Max.   :17.40
##                                   (Other):894
##  hematocrit_percent trial_number temp_tmt   humidity_tmt     SVL_mm
##  Min.   :13.00      1:175        Hot :467   Humid:468    Min.   :60.00
##  1st Qu.:26.00      2:203        Cool:469   Dry  :468    1st Qu.:66.00
##  Median :32.00      3:231                                Median :67.00
##  Mean   :31.99      4:189                                Mean   :67.74
```

```
##  3rd Qu.:37.00      5:138                               3rd Qu.:70.00
##  Max.   :52.00                                          Max.   :77.00
##  NA's   :408
##          tmt          day_n        day_factor osmolality_mmol_kg_mean
##  Hot Humid :230   Min.   : 0.000   0 :134     Min.   :295.3
##  Hot Dry   :237   1st Qu.: 4.000   4 :134     1st Qu.:336.1
##  Cool Humid:238   Median : 6.000   5 :134     Median :351.3
##  Cool Dry  :231   Mean   : 5.705   6 :134     Mean   :354.3
##                   3rd Qu.: 8.000   7 :134     3rd Qu.:370.0
##                   Max.   :10.000   8 :134     Max.   :471.5
##                                    10:132     NA's   :414
##  CEWL_g_m2h_mean   msmt_temp_C    msmt_RH_percent cloacal_temp_C
##  Min.   : 7.152   Min.   :24.80   Min.   :25.52   Min.   :23.00
##  1st Qu.:19.755   1st Qu.:26.20   1st Qu.:46.11   1st Qu.:25.00
##  Median :24.152   Median :26.74   Median :47.88   Median :26.00
##  Mean   :24.767   Mean   :26.72   Mean   :46.74   Mean   :25.92
##  3rd Qu.:28.505   3rd Qu.:27.11   3rd Qu.:50.50   3rd Qu.:27.00
##  Max.   :56.066   Max.   :29.20   Max.   :56.16   Max.   :30.00
##  NA's   :669      NA's   :668     NA's   :668     NA's   :668
##   msmt_temp_K      e_s_kPa_m       e_a_kPa_m       msmt_VPD_kPa
##  Min.   :297.9   Min.   :3.219   Min.   :0.9894   Min.   :1.486
##  1st Qu.:299.4   1st Qu.:3.504   1st Qu.:1.6464   1st Qu.:1.767
##  Median :299.9   Median :3.620   Median :1.7411   Median :1.853
##  Mean   :299.9   Mean   :3.620   Mean   :1.6833   Mean   :1.937
##  3rd Qu.:300.3   3rd Qu.:3.701   3rd Qu.:1.7992   3rd Qu.:2.012
##  Max.   :302.4   Max.   :4.194   Max.   :1.9326   Max.   :3.021
##  NA's   :668     NA's   :668     NA's   :668      NA's   :668
##       SMI
##  Min.   : 6.747
##  1st Qu.: 9.714
##  Median :10.594
##  Mean   :10.599
##  3rd Qu.:11.390
##  Max.   :15.063
##
```

Export

```
write_rds(full_dat3, "./data/analysis_data_experiment.RDS")
```