Climate Water Loss Experiment - Treatment Hydration Analysis

Savannah Weaver

2021

Contents

Packages

```
`%nin%` = Negate(`%in%`)
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse") # workflow and plots
if (!require("lme4")) install.packages("lme4")
library("lme4") # for LMMs
if (!require("lmerTest")) install.packages("lmerTest")
library("lmerTest") # for p-values
if (!require("ggpubr")) install.packages("ggpubr")
library("ggpubr") # for multi-ggplot figs
if (!require("UsingR")) install.packages("UsingR")
library("UsingR") # simple.eda model assumption checker
if (!require("broom.mixed")) install.packages("broom.mixed")
library("broom.mixed") # lmer model export
if (!require("emmeans")) install.packages('emmeans')
library('emmeans')
if (!require("car")) install.packages("car")
library("car") # VIFs
if (!require("AICcmodavg")) install.packages("AICcmodavg")
library("AICcmodavg") # model selection
```

```r
if (!require("MuMIn")) install.packages("MuMIn")
library("MuMIn") # model selection
if (!require("RColorBrewer")) install.packages("RColorBrewer")
library("RColorBrewer") # color
```

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 then acclimated to 4 different climate treatments. **This R file analyzes the effect of experimental climate treatments on lizard body condition, osmotic balance, and osmoregulation.** Please refer to the published scientific journal article for full details.

Data

Load

Read-in data that was compiled, formatted, and checked for completeness in 'wrangling_general'. See that file for information related to the variables.

```r
dat <- read_rds("./data/analysis_data_experiment.RDS") %>%
  # add VPD values by tmt and trial group
  left_join(read_rds("./data/HOBO_tmt_trial_diffs.RDS"),
            by = c('tmt', 'trial_number'='trial')) %>%
  # add the per tmt group calculation of VPD from analysis_HOBO
  mutate(VPD_kPa = case_when(substr(tmt, 1, 6) == "Hot Dr" ~ 3.82,
                             substr(tmt, 1, 6) == "Hot Hu" ~ 1.07,
                             substr(tmt, 1, 6) == "Cool D" ~ 2.50,
                             substr(tmt, 1, 6) == "Cool H" ~ 0.64))
summary(dat)
```

```
##  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
##  Cool Humid (0.6 kPa):238   Min.   : 0.000   0 :134     Min.   :295.3
##  Hot Humid (1.1 kPa) :230   1st Qu.: 4.000   4 :134     1st Qu.:336.1
##  Cool Dry (2.5 kPa)  :231   Median : 6.000   5 :134     Median :351.3
##  Hot Dry (3.8 kPa)   :237   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         temp_mean_tmttrial temp_SD_tmttrial humidity_mean_tmttrial
##  Min.   : 6.747   Min.   :23.30      Min.   :0.5966   Min.   :13.75
##  1st Qu.: 9.714   1st Qu.:24.05      1st Qu.:0.7828   1st Qu.:29.21
##  Median :10.594   Median :24.88      Median :1.0461   Median :45.24
##  Mean   :10.599   Mean   :29.60      Mean   :1.1513   Mean   :52.94
##  3rd Qu.:11.390   3rd Qu.:35.05      3rd Qu.:1.5191   3rd Qu.:82.84
##  Max.   :15.063   Max.   :36.00      Max.   :1.8447   Max.   :93.15
##
##  humidity_SD_tmttrial    e_s_kPa      VPD_kPa_tmttrial    VPD_kPa
##  Min.   : 4.370       Min.   :2.859   Min.   :0.1958   Min.   :0.640
##  1st Qu.: 6.234       1st Qu.:2.992   1st Qu.:0.7925   1st Qu.:0.640
##  Median : 7.382       Median :3.142   Median :2.0310   Median :1.785
##  Mean   : 8.765       Mean   :4.330   Mean   :1.9985   Mean   :2.010
##  3rd Qu.:12.297       3rd Qu.:5.639   3rd Qu.:3.1520   3rd Qu.:3.820
##  Max.   :19.846       Max.   :5.944   Max.   :4.0640   Max.   :3.820
##
```

Split

Make sub-dataframes without recovery data / with only recovery-related data:

```
dat_no_rehab <- dat %>%
  dplyr::filter(day_n %in% c(seq(0,8)))

recovery_values <- dat %>%
  dplyr::filter(day_n == 10) %>%
  dplyr::select(individual_ID,
                end_hct = hematocrit_percent,
                end_osml = osmolality_mmol_kg_mean,
                end_SMI = SMI)

recovery_v_post_exp <- dat %>%
  dplyr::filter(day_n == 8) %>%
  left_join(recovery_values, by = 'individual_ID') %>%
  mutate(delta_osml_10_8 = end_osml - osmolality_mmol_kg_mean,
         delta_hct_10_8 = end_hct - hematocrit_percent,
```

```
       delta_SMI_10_8 = end_SMI - SMI)
summary(recovery_v_post_exp)
```

```
##  measurement_date        type     individual_ID     mass_g
##  Min.   :2021-06-24   exp  :134   201     :  1   Min.   : 7.00
##  1st Qu.:2021-07-04   rehab:  0   202     :  1   1st Qu.: 9.10
##  Median :2021-07-28               203     :  1   Median :10.20
##  Mean   :2021-07-25               204     :  1   Mean   :10.13
##  3rd Qu.:2021-08-16               205     :  1   3rd Qu.:11.10
##  Max.   :2021-08-30               206     :  1   Max.   :14.20
##                                   (Other):128
##  hematocrit_percent trial_number temp_tmt  humidity_tmt     SVL_mm
##  Min.   :13.00      1:25         Hot :67   Humid:67     Min.   :60.00
##  1st Qu.:25.00      2:29         Cool:67   Dry  :67     1st Qu.:66.00
##  Median :28.00      3:33                                Median :67.00
##  Mean   :28.53      4:27                                Mean   :67.73
##  3rd Qu.:32.50      5:20                                3rd Qu.:70.00
##  Max.   :48.00                                          Max.   :77.00
##  NA's   :3
##                       tmt         day_n   day_factor osmolality_mmol_kg_mean
##  Cool Humid (0.6 kPa):34   Min.   :8    0 :  0     Min.   :295.3
##  Hot Humid (1.1 kPa) :33   1st Qu.:8    4 :  0     1st Qu.:332.6
##  Cool Dry (2.5 kPa)  :33   Median :8    5 :  0     Median :345.0
##  Hot Dry (3.8 kPa)   :34   Mean   :8    6 :  0     Mean   :351.2
##                            3rd Qu.:8    7 :  0     3rd Qu.:362.5
##                            Max.   :8    8 :134     Max.   :445.5
##                                         10:  0     NA's   :10
##  CEWL_g_m2h_mean  msmt_temp_C    msmt_RH_percent cloacal_temp_C
##  Min.   :12.38   Min.   :24.80   Min.   :43.68   Min.   :23.00
##  1st Qu.:23.80   1st Qu.:25.83   1st Qu.:47.13   1st Qu.:25.00
##  Median :26.88   Median :26.29   Median :49.74   Median :26.00
##  Mean   :28.79   Mean   :26.25   Mean   :49.35   Mean   :25.39
##  3rd Qu.:33.22   3rd Qu.:26.77   3rd Qu.:51.77   3rd Qu.:26.00
##  Max.   :56.07   Max.   :27.23   Max.   :56.16   Max.   :28.00
##  NA's   :1
##   msmt_temp_K       e_s_kPa_m       e_a_kPa_m      msmt_VPD_kPa
##  Min.   :297.9   Min.   :3.219   Min.   :1.595   Min.   :1.486
##  1st Qu.:299.0   1st Qu.:3.427   1st Qu.:1.638   1st Qu.:1.661
##  Median :299.4   Median :3.523   Median :1.765   Median :1.802
##  Mean   :299.4   Mean   :3.517   Mean   :1.734   Mean   :1.784
##  3rd Qu.:299.9   3rd Qu.:3.627   3rd Qu.:1.809   3rd Qu.:1.855
##  Max.   :300.4   Max.   :3.728   Max.   :1.933   Max.   :2.098
##
##       SMI          temp_mean_tmttrial temp_SD_tmttrial humidity_mean_tmttrial
##  Min.   : 7.317   Min.   :23.30      Min.   :0.5966   Min.   :13.75
##  1st Qu.: 9.254   1st Qu.:24.11      1st Qu.:0.7828   1st Qu.:29.21
##  Median :10.141   Median :29.74      Median :1.0461   Median :45.24
##  Mean   :10.099   Mean   :29.61      Mean   :1.1502   Mean   :52.95
##  3rd Qu.:10.877   3rd Qu.:35.05      3rd Qu.:1.4894   3rd Qu.:82.84
##  Max.   :13.545   Max.   :36.00      Max.   :1.8447   Max.   :93.15
##
##  humidity_SD_tmttrial    e_s_kPa       VPD_kPa_tmttrial    VPD_kPa
##  Min.   : 4.370       Min.   :2.859   Min.   :0.1958   Min.   :0.6400
##  1st Qu.: 6.234       1st Qu.:3.001   1st Qu.:0.7925   1st Qu.:0.7475
```

```
##  Median : 7.382       Median :4.323   Median :2.0310   Median :1.7850
##  Mean   : 8.758       Mean   :4.333   Mean   :1.9993   Mean   :2.0108
##  3rd Qu.:11.490       3rd Qu.:5.639   3rd Qu.:3.0278   3rd Qu.:3.4900
##  Max.   :19.846       Max.   :5.944   Max.   :4.0640   Max.   :3.8200
##
##     end_hct         end_osml         end_SMI       delta_osml_10_8
##  Min.   :13.0   Min.   :308.0   Min.   : 6.747   Min.   :-78.33
##  1st Qu.:23.0   1st Qu.:341.2   1st Qu.: 9.603   1st Qu.:-12.25
##  Median :25.5   Median :358.3   Median :10.459   Median : 15.00
##  Mean   :26.6   Mean   :360.1   Mean   :10.549   Mean   : 10.05
##  3rd Qu.:30.0   3rd Qu.:374.8   3rd Qu.:11.532   3rd Qu.: 33.50
##  Max.   :47.0   Max.   :471.5   Max.   :15.063   Max.   : 86.67
##  NA's   :4      NA's   :3       NA's   :2        NA's   :12
##  delta_hct_10_8    delta_SMI_10_8
##  Min.   :-22.000   Min.   :-1.13086
##  1st Qu.: -5.000   1st Qu.:-0.09969
##  Median : -1.000   Median : 0.38598
##  Mean   : -2.063   Mean   : 0.45063
##  3rd Qu.:  1.000   3rd Qu.: 1.00670
##  Max.   : 11.000   Max.   : 2.46394
##  NA's   :7         NA's   :2
```

```
recovery_v_pre_exp <- dat %>%
  dplyr::filter(day_n == 0) %>%
  left_join(recovery_values, by = 'individual_ID') %>%
  mutate(delta_osml_10_0 = end_osml - osmolality_mmol_kg_mean,
         delta_hct_10_0 = end_hct - hematocrit_percent,
         delta_SMI_10_0 = end_SMI - SMI)
summary(recovery_v_pre_exp)
```

```
##  measurement_date        type     individual_ID     mass_g
##  Min.   :2021-06-16   exp  :134   201     :  1   Min.   : 8.80
##  1st Qu.:2021-06-26   rehab:  0   202     :  1   1st Qu.:10.60
##  Median :2021-07-20               203     :  1   Median :11.70
##  Mean   :2021-07-17               204     :  1   Mean   :11.74
##  3rd Qu.:2021-08-08               205     :  1   3rd Qu.:12.70
##  Max.   :2021-08-22               206     :  1   Max.   :17.40
##                                   (Other):128
##  hematocrit_percent trial_number temp_tmt  humidity_tmt     SVL_mm
##  Min.   :27.00      1:25         Hot :67   Humid:67     Min.   :60.00
##  1st Qu.:34.25      2:29         Cool:67   Dry  :67     1st Qu.:66.00
##  Median :39.00      3:33                                Median :67.00
##  Mean   :38.92      4:27                                Mean   :67.73
##  3rd Qu.:43.00      5:20                                3rd Qu.:70.00
##  Max.   :52.00                                          Max.   :77.00
##
##                   tmt         day_n   day_factor osmolality_mmol_kg_mean
##  Cool Humid (0.6 kPa):34   Min.   :0   0 :134     Min.   :305.0
##  Hot Humid (1.1 kPa) :33   1st Qu.:0   4 :  0     1st Qu.:334.1
##  Cool Dry (2.5 kPa)  :33   Median :0   5 :  0     Median :344.6
##  Hot Dry (3.8 kPa)   :34   Mean   :0   6 :  0     Mean   :348.2
##                            3rd Qu.:0   7 :  0     3rd Qu.:361.9
##                            Max.   :0   8 :  0     Max.   :395.0
##                                        10:  0
##  CEWL_g_m2h_mean   msmt_temp_C    msmt_RH_percent cloacal_temp_C
```

```
##  Min.   : 7.152   Min.   :25.90   Min.   :25.52   Min.   :25.00
##  1st Qu.:17.348   1st Qu.:26.72   1st Qu.:45.77   1st Qu.:26.00
##  Median :21.030   Median :26.96   Median :47.13   Median :26.00
##  Mean   :20.779   Mean   :27.19   Mean   :44.14   Mean   :26.46
##  3rd Qu.:24.416   3rd Qu.:27.50   3rd Qu.:48.51   3rd Qu.:27.00
##  Max.   :34.660   Max.   :29.20   Max.   :53.15   Max.   :30.00
##
##   msmt_temp_K      e_s_kPa_m       e_a_kPa_m       msmt_VPD_kPa
##  Min.   :299.1   Min.   :3.441   Min.   :0.9894   Min.   :1.612
##  1st Qu.:299.9   1st Qu.:3.616   1st Qu.:1.6916   1st Qu.:1.846
##  Median :300.1   Median :3.669   Median :1.7367   Median :1.937
##  Mean   :300.3   Mean   :3.724   Mean   :1.6331   Mean   :2.091
##  3rd Qu.:300.6   3rd Qu.:3.790   3rd Qu.:1.7880   3rd Qu.:2.053
##  Max.   :302.4   Max.   :4.194   Max.   :1.8502   Max.   :3.021
##
##       SMI         temp_mean_tmttrial temp_SD_tmttrial humidity_mean_tmttrial
##  Min.   : 9.122   Min.   :23.30      Min.   :0.5966   Min.   :13.75
##  1st Qu.:10.983   1st Qu.:24.11      1st Qu.:0.7828   1st Qu.:29.21
##  Median :11.687   Median :29.74      Median :1.0461   Median :45.24
##  Mean   :11.693   Mean   :29.61      Mean   :1.1502   Mean   :52.95
##  3rd Qu.:12.347   3rd Qu.:35.05      3rd Qu.:1.4894   3rd Qu.:82.84
##  Max.   :14.263   Max.   :36.00      Max.   :1.8447   Max.   :93.15
##
##  humidity_SD_tmttrial    e_s_kPa      VPD_kPa_tmttrial    VPD_kPa
##  Min.   : 4.370       Min.   :2.859   Min.   :0.1958   Min.   :0.6400
##  1st Qu.: 6.234       1st Qu.:3.001   1st Qu.:0.7925   1st Qu.:0.7475
##  Median : 7.382       Median :4.323   Median :2.0310   Median :1.7850
##  Mean   : 8.758       Mean   :4.333   Mean   :1.9993   Mean   :2.0108
##  3rd Qu.:11.490       3rd Qu.:5.639   3rd Qu.:3.0278   3rd Qu.:3.4900
##  Max.   :19.846       Max.   :5.944   Max.   :4.0640   Max.   :3.8200
##
##     end_hct        end_osml        end_SMI       delta_osml_10_0
##  Min.   :13.0   Min.   :308.0   Min.   : 6.747   Min.   :-67.33
##  1st Qu.:23.0   1st Qu.:341.2   1st Qu.: 9.603   1st Qu.:-12.08
##  Median :25.5   Median :358.3   Median :10.459   Median : 14.00
##  Mean   :26.6   Mean   :360.1   Mean   :10.549   Mean   : 12.19
##  3rd Qu.:30.0   3rd Qu.:374.8   3rd Qu.:11.532   3rd Qu.: 38.92
##  Max.   :47.0   Max.   :471.5   Max.   :15.063   Max.   :130.00
##  NA's   :4      NA's   :3       NA's   :2        NA's   :3
##  delta_hct_10_0  delta_SMI_10_0
##  Min.   :-32.0   Min.   :-4.1183
##  1st Qu.:-15.0   1st Qu.:-1.9496
##  Median :-12.0   Median :-1.1513
##  Mean   :-12.4   Mean   :-1.1462
##  3rd Qu.: -8.0   3rd Qu.:-0.2646
##  Max.   :  2.0   Max.   : 1.6871
##  NA's   :4       NA's   :2
```

Check

Dates:

```
unique(dat$measurement_date)
```

```
##  [1] "2021-06-16" "2021-06-20" "2021-06-21" "2021-06-22" "2021-06-23"
```

```
##  [6] "2021-06-24" "2021-06-26" "2021-06-30" "2021-07-01" "2021-07-02"
## [11] "2021-07-03" "2021-07-04" "2021-07-06" "2021-07-20" "2021-07-24"
## [16] "2021-07-25" "2021-07-26" "2021-07-27" "2021-07-28" "2021-07-30"
## [21] "2021-08-08" "2021-08-12" "2021-08-13" "2021-08-14" "2021-08-15"
## [26] "2021-08-16" "2021-08-18" "2021-08-22" "2021-08-26" "2021-08-27"
## [31] "2021-08-28" "2021-08-29" "2021-08-30" "2021-09-01"
```

Number of measurements for each lizard:

```
dat_no_rehab %>%
  group_by(individual_ID) %>%
  summarise(n = n()) %>%
  arrange(n)
```

```
## # A tibble: 134 x 2
##    individual_ID     n
##    <fct>         <int>
##  1 201               6
##  2 202               6
##  3 203               6
##  4 204               6
##  5 205               6
##  6 206               6
##  7 207               6
##  8 208               6
##  9 209               6
## 10 210               6
## # ... with 124 more rows
```

Every lizard has 6 experimental measurements: pre-tmt, mid-tmt, post-tmt, and mass checks on each of the 3 days between mid and post-tmt.

Did any of the treatment groups inherently start out with large differences in response variables?

```
dat %>%
  dplyr::filter(day_n == 0) %>%
  group_by(tmt) %>%
  summarise(mean(mass_g),
            sd(mass_g),
            mean(SMI),
            mean(hematocrit_percent),
            mean(osmolality_mmol_kg_mean),
            mean(CEWL_g_m2h_mean))
```

```
## # A tibble: 4 x 7
##   tmt                  `mean(mass_g)` sd(mass_~1 mean(~2 mean(~3 mean(~4 mean(~5
##   <fct>                         <dbl>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Cool Humid (0.6 kPa)           11.6       1.35    11.7    39.6    351.    20.9
## 2 Hot Humid (1.1 kPa)            11.6       1.75    11.5    37.9    347.    21.4
## 3 Cool Dry (2.5 kPa)             11.8       1.61    11.8    39.3    346.    20.0
## 4 Hot Dry (3.8 kPa)              12.0       1.68    11.8    38.9    347.    20.9
## # ... with abbreviated variable names 1: `sd(mass_g)`, 2: `mean(SMI)`,
## #   3: `mean(hematocrit_percent)`, 4: `mean(osmolality_mmol_kg_mean)`,
## #   5: `mean(CEWL_g_m2h_mean)`
```

There are slight differences, but overall the starting values across groups are more or less the same.

Temp & RH during (all, before and after exp) CEWL measurements:

```r
summary(dat_no_rehab)
```

```
##  measurement_date        type     individual_ID     mass_g     
##  Min.   :2021-06-16   exp  :804   201    :  6   Min.   : 7.00  
##  1st Qu.:2021-06-30   rehab:  0   202    :  6   1st Qu.: 9.50  
##  Median :2021-07-25               203    :  6   Median :10.60  
##  Mean   :2021-07-22               204    :  6   Mean   :10.65  
##  3rd Qu.:2021-08-13               205    :  6   3rd Qu.:11.60  
##  Max.   :2021-08-30               206    :  6   Max.   :17.40  
##                                   (Other):768                  
##  hematocrit_percent trial_number temp_tmt   humidity_tmt     SVL_mm     
##  Min.   :13.00      1:150        Hot :402   Humid:402    Min.   :60.00  
##  1st Qu.:28.25      2:174        Cool:402   Dry  :402    1st Qu.:66.00  
##  Median :33.00      3:198                                Median :67.00  
##  Mean   :33.75      4:162                                Mean   :67.73  
##  3rd Qu.:39.00      5:120                                3rd Qu.:70.00  
##  Max.   :52.00                                           Max.   :77.00  
##  NA's   :406                                                            
##                       tmt          day_n     day_factor osmolality_mmol_kg_mean
##  Cool Humid (0.6 kPa):204   Min.   :0.0   0 :134     Min.   :295.3          
##  Hot Humid (1.1 kPa) :198   1st Qu.:4.0   4 :134     1st Qu.:334.7          
##  Cool Dry (2.5 kPa)  :198   Median :5.5   5 :134     Median :348.3          
##  Hot Dry (3.8 kPa)   :204   Mean   :5.0   6 :134     Mean   :352.3          
##                             3rd Qu.:7.0   7 :134     3rd Qu.:367.4          
##                             Max.   :8.0   8 :134     Max.   :445.5          
##                                           10:  0     NA's   :413            
##  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   :537      NA's   :536     NA's   :536     NA's   :536    
##   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   :536     NA's   :536     NA's   :536      NA's   :536    
##       SMI         temp_mean_tmttrial temp_SD_tmttrial humidity_mean_tmttrial
##  Min.   : 7.317   Min.   :23.30      Min.   :0.5966   Min.   :13.75         
##  1st Qu.: 9.748   1st Qu.:24.05      1st Qu.:0.7828   1st Qu.:29.21         
##  Median :10.624   Median :29.74      Median :1.0461   Median :45.24         
##  Mean   :10.607   Mean   :29.61      Mean   :1.1502   Mean   :52.95         
##  3rd Qu.:11.348   3rd Qu.:35.05      3rd Qu.:1.5191   3rd Qu.:82.84         
##  Max.   :14.263   Max.   :36.00      Max.   :1.8447   Max.   :93.15         
##                                                                             
##  humidity_SD_tmttrial    e_s_kPa      VPD_kPa_tmttrial    VPD_kPa     
##  Min.   : 4.370       Min.   :2.859   Min.   :0.1958   Min.   :0.640  
##  1st Qu.: 6.234       1st Qu.:2.992   1st Qu.:0.7925   1st Qu.:0.640  
##  Median : 7.382       Median :4.323   Median :2.0310   Median :1.785  
```

```
##  Mean   : 8.758       Mean   :4.333   Mean   :1.9993   Mean   :2.011
##  3rd Qu.:12.297       3rd Qu.:5.639   3rd Qu.:3.1520   3rd Qu.:3.820
##  Max.   :19.846       Max.   :5.944   Max.   :4.0640   Max.   :3.820
##
```

```
dat_no_rehab %>%
  group_by(type) %>%
  summarise(mean(msmt_temp_C, na.rm = T),
            sd(msmt_temp_C, na.rm = T),
            mean(msmt_RH_percent, na.rm = T),
            sd(msmt_RH_percent, na.rm = T),
            mean(msmt_VPD_kPa, na.rm = T),
            mean(msmt_VPD_kPa, na.rm = T))
```

```
## # A tibble: 1 x 6
##   type  `mean(msmt_temp_C, na.rm = T)` sd(msmt_temp_C,~1 mean(~2 sd(ms~3 mean(~4
##   <fct>                          <dbl>             <dbl>   <dbl>   <dbl>   <dbl>
## 1 exp                             26.7             0.799    46.7    6.76    1.94
## # ... with abbreviated variable names 1: `sd(msmt_temp_C, na.rm = T)`,
## #   2: `mean(msmt_RH_percent, na.rm = T)`, 3: `sd(msmt_RH_percent, na.rm = T)`,
## #   4: `mean(msmt_VPD_kPa, na.rm = T)`
```

Means by Day

Calculate mean values per day per tmt group.

```
means <- dat %>% # use whole dat because want for both exp and rehyd
  group_by(day_n, tmt) %>%
  summarise(n_lizards = n(),
            mean_CEWL = mean(CEWL_g_m2h_mean, na.rm = T),
            sd_CEWL = sd(CEWL_g_m2h_mean, na.rm = T),
            mean_osml = mean(osmolality_mmol_kg_mean, na.rm = T),
            sd_osml = sd(osmolality_mmol_kg_mean, na.rm = T),
            mean_hct = mean(hematocrit_percent, na.rm = T),
            sd_hct = sd(hematocrit_percent, na.rm = T),
            mean_SMI = mean(SMI, na.rm = T),
            sd_SMI = sd(SMI, na.rm = T)) %>%
  mutate(se_CEWL = (sd_CEWL/sqrt(n_lizards)),
         se_osml = (sd_osml/sqrt(n_lizards)),
         se_hct = (sd_hct/sqrt(n_lizards)),
         se_SMI = (sd_SMI/sqrt(n_lizards)))
```

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

```
# get rid of non-defined points
means$mean_CEWL[is.nan(means$mean_CEWL)] <- NA
means$mean_osml[is.nan(means$mean_osml)] <- NA
means$mean_hct[is.nan(means$mean_hct)] <- NA
means$mean_SMI[is.nan(means$mean_SMI)] <- NA
means
```

```
## # A tibble: 28 x 15
## # Groups:   day_n [7]
##     day_n tmt      n_liz~1 mean_~2 sd_CEWL mean_~3 sd_osml mean_~4 sd_hct mean_~5
##     <dbl> <fct>      <int>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>
##  1     0 Cool Hu~      34    20.9    4.78    351.    20.3    39.6   5.30    11.7
```

```
## 2        0 Hot Hum~       33     21.4     4.85     347.     18.7     37.9     5.46     11.5
## 3        0 Cool Dr~       33     20.0     6.08     346.     20.6     39.3     5.96     11.8
## 4        0 Hot Dry~       34     20.9     5.93     347.     17.9     38.9     5.05     11.8
## 5        4 Cool Hu~       34     NA       NA       356.     24.5     34.5     5.34     11.2
## 6        4 Hot Hum~       33     NA       NA       359.     22.5     32.0     5.38     10.5
## 7        4 Cool Dr~       33     NA       NA       355.     27.0     35.0     7.02     11.1
## 8        4 Hot Dry~       34     NA       NA       361.     25.8     33.1     5.00     10.4
## 9        5 Cool Hu~       34     NA       NA        NA      NA       NA      NA        11.0
## 10       5 Hot Hum~       33     NA       NA        NA      NA       NA      NA        10.1
## # ... with 18 more rows, 5 more variables: sd_SMI <dbl>, se_CEWL <dbl>,
## #   se_osml <dbl>, se_hct <dbl>, se_SMI <dbl>, and abbreviated variable names
## #   1: n_lizards, 2: mean_CEWL, 3: mean_osml, 4: mean_hct, 5: mean_SMI
```

```
# get only means for the very end
end_means <- means %>%
  dplyr::filter(day_n == 8)
#write.csv(end_means, "./results_statistics/exp_end_means.csv")
```

End Values Only

Select for only day=8 values.

```
end_vals <- dat %>%
  dplyr::filter(day_n == 8)
```

delta CEWL

Get a df that only has complete observations that include CEWL values (only obs from before and after the experiment). Then, calculate the CHANGE (delta) in CEWL from before to after the experiment. Because we only measured CEWL at those two time points, it makes more sense to assess the **amount of change** in CEWL for each lizard, rather than measuring the change over time.

```
start_CEWL <- dat_no_rehab %>%
  dplyr::filter(day_n == 0) %>%
  dplyr::select(individual_ID, start_CEWL = CEWL_g_m2h_mean)
dat_no_rehab_deltaCEWL <- dat_no_rehab %>% # initiate new df
  dplyr::filter(complete.cases(CEWL_g_m2h_mean)) %>% # only use obs incl CEWL
  dplyr::filter(day_n == 8) %>% # get only obs for post-exp
  left_join(start_CEWL, by = 'individual_ID') %>% # add start CEWL to both obs for each lizard
  mutate(delta_CEWL = CEWL_g_m2h_mean - start_CEWL) # calculate deltaCEWL after-before experiment
```

Experiment Models

We predicted that there would be effects of day, humidity treatment, temperature treatment, and treatment VPD. However, we can't use the standard backwards model selection because the three treatment variables are collinear (VIF much higher than acceptable) and it leads to issues with changing the sign of the estimates when all three are included together. So, we will run singular models with each treatment variable alone:

response ~ day*humidity response ~ day*temperature response ~ day*VPD

Then, we will use AIC, RMSE, and R-sq to assess which treatment effect is most important to that response variable.

Body Condition

Building

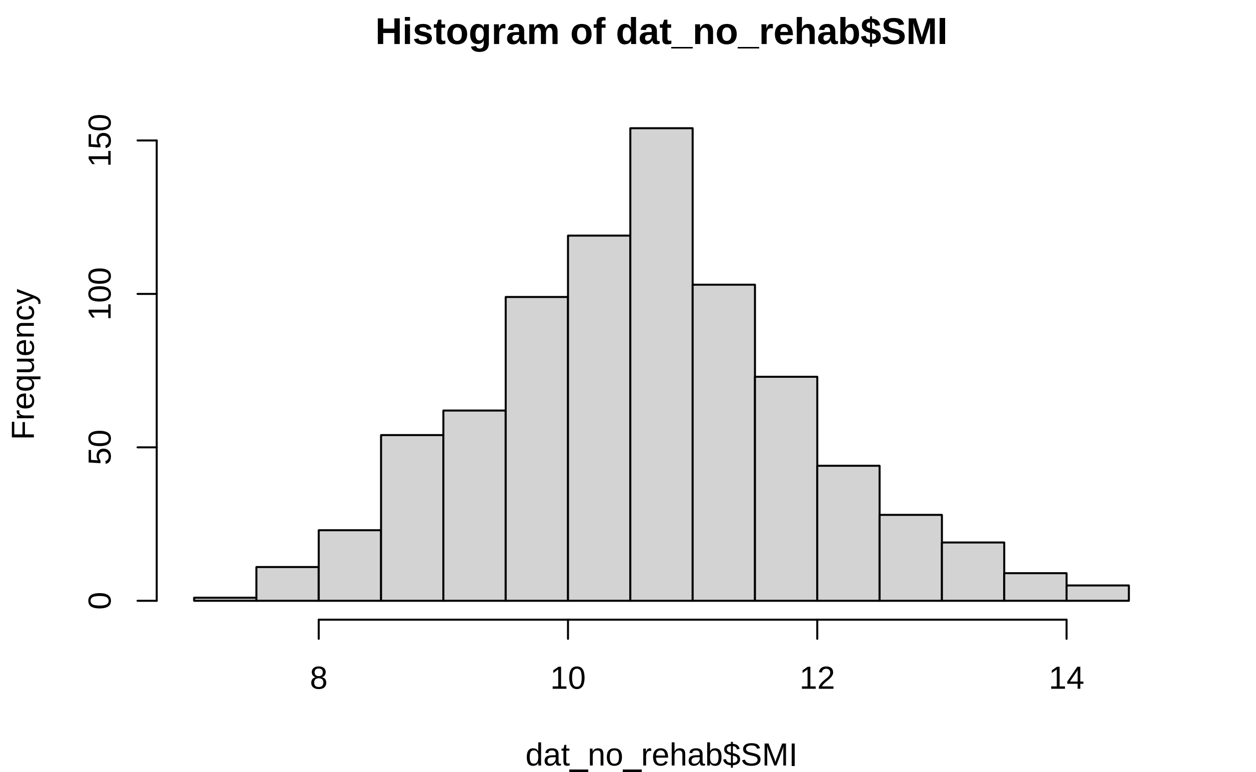
Build each treatment effect model.

```
SMI_mod_VPD <- lmerTest::lmer(data = dat_no_rehab,
                              SMI ~ day_n * VPD_kPa +
                              (1|trial_number/individual_ID))
SMI_mod_hum <- lmerTest::lmer(data = dat_no_rehab,
                              SMI ~ day_n * humidity_tmt +
                              (1|trial_number/individual_ID))
SMI_mod_temp <- lmerTest::lmer(data = dat_no_rehab,
                               SMI ~ day_n * temp_tmt +
                               (1|trial_number/individual_ID))
```

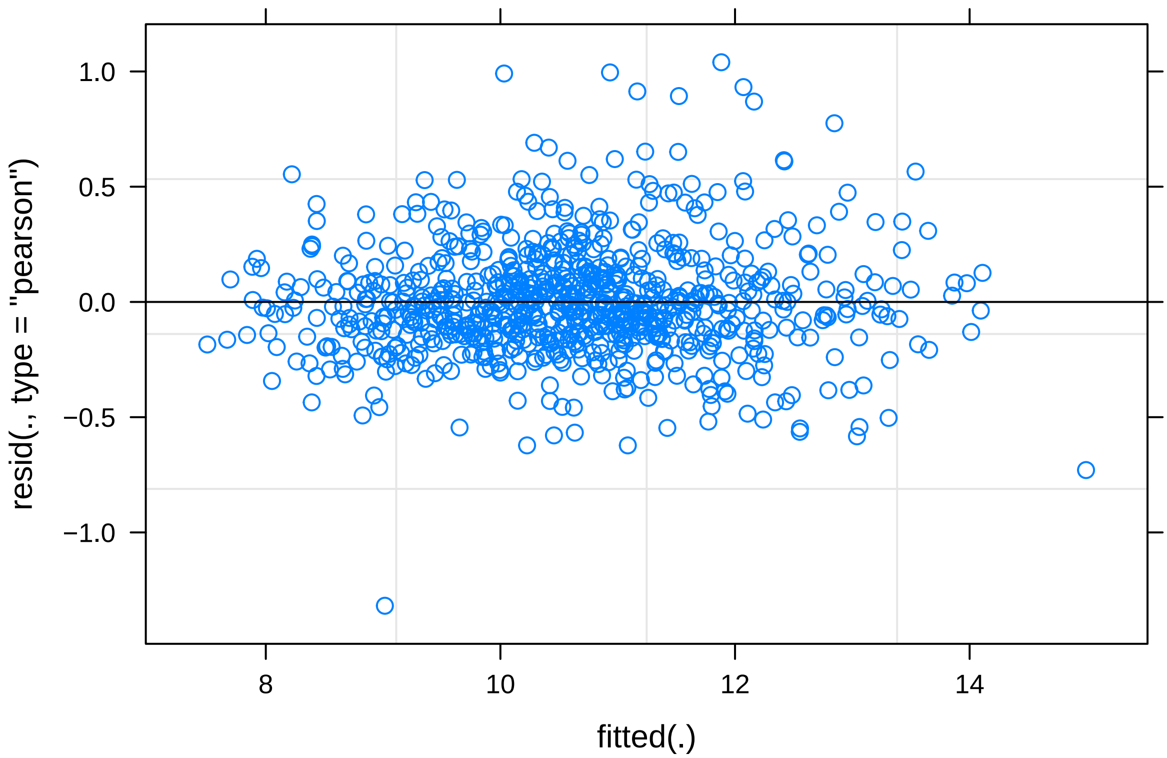
Assumptions

Check linear regression assumptions/conditions.

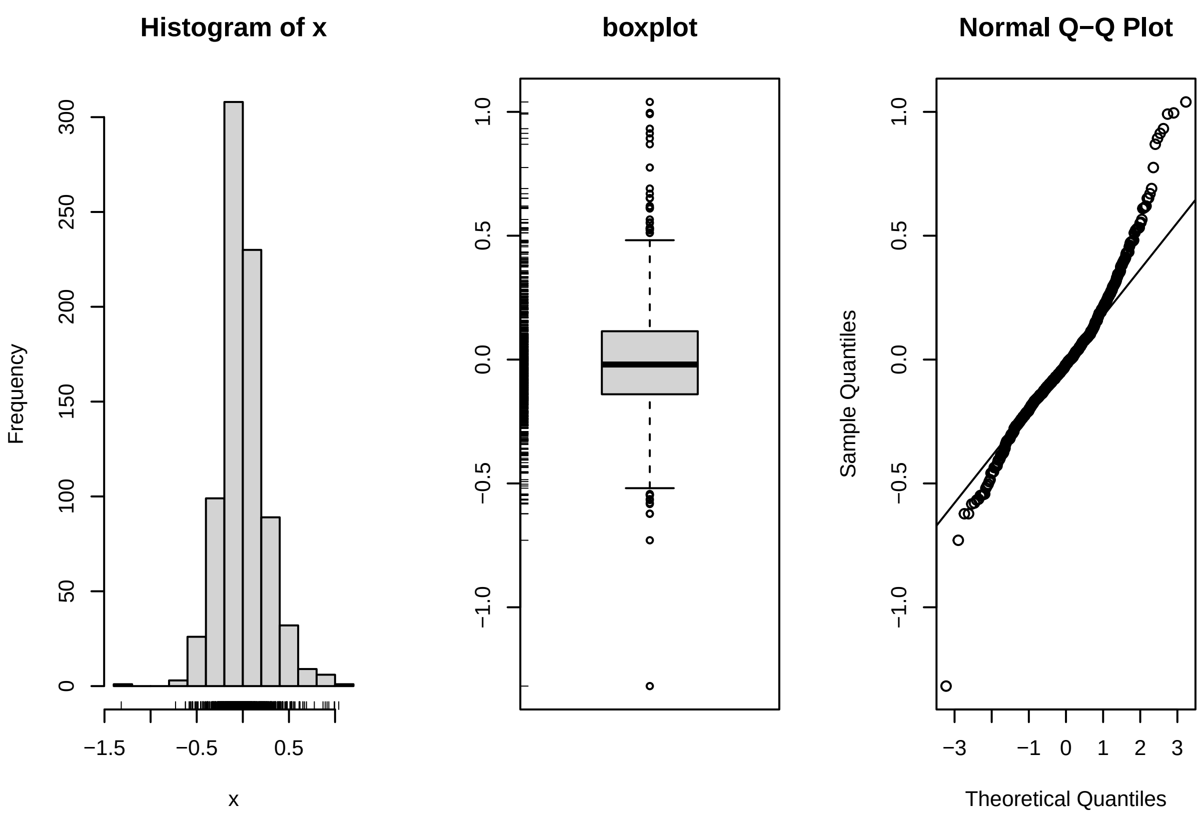
```
# distribution of SMI
hist(dat_no_rehab$SMI)
```




```
# VPD model
plot(SMI_mod_VPD)
```



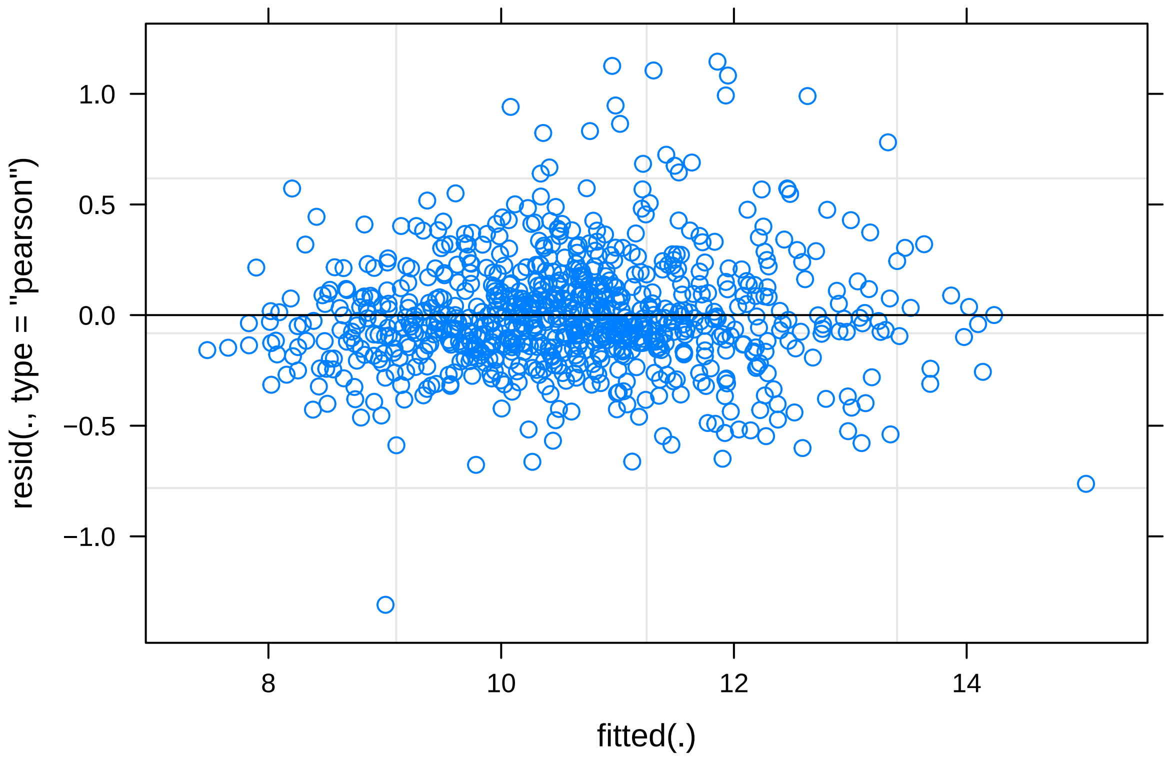

```
simple.eda(residuals(SMI_mod_VPD))
```



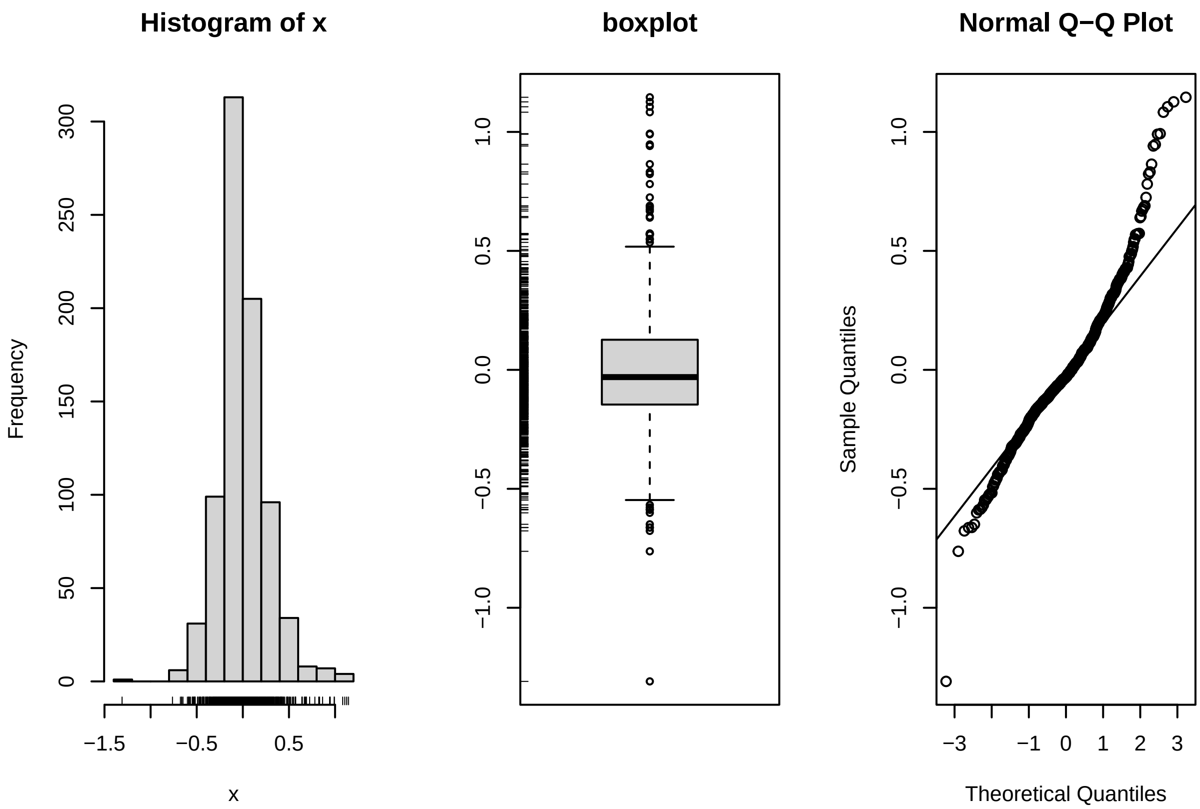
```r
shapiro.test(residuals(SMI_mod_VPD))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(SMI_mod_VPD)
## W = 0.96052, p-value = 6.691e-14
```

```r
# humidity model
plot(SMI_mod_hum)
```



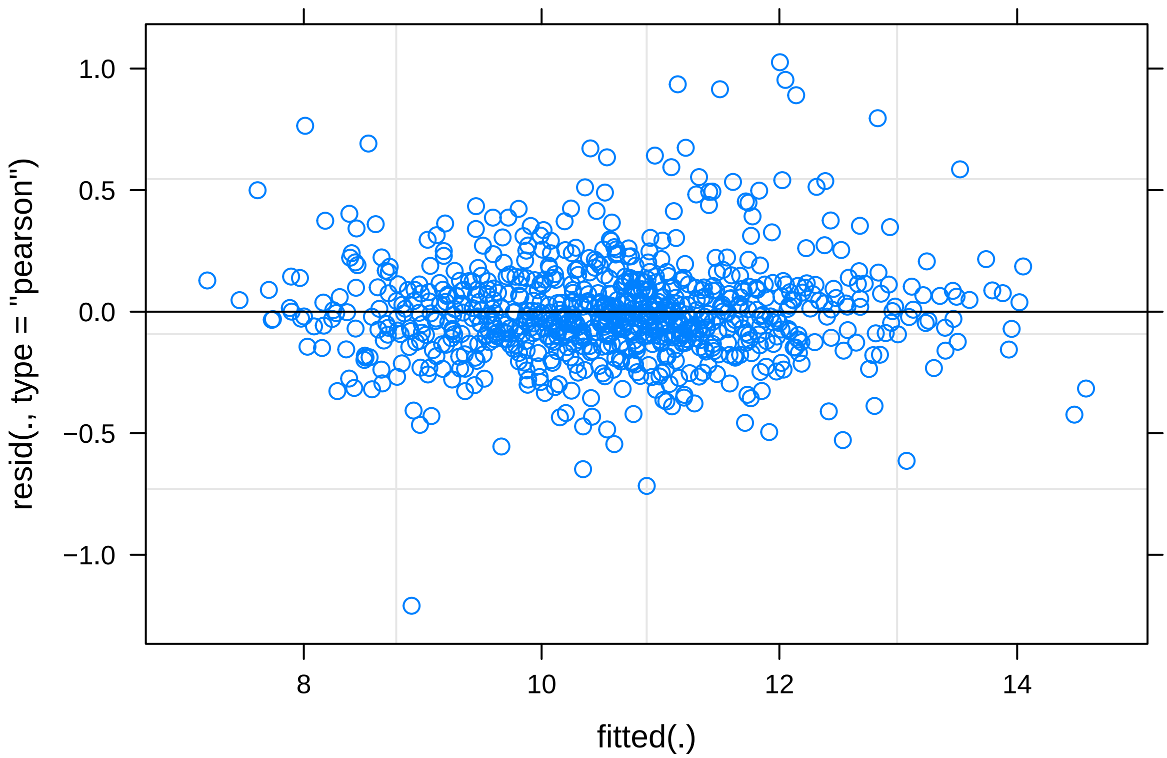

```r
simple.eda(residuals(SMI_mod_hum))
```



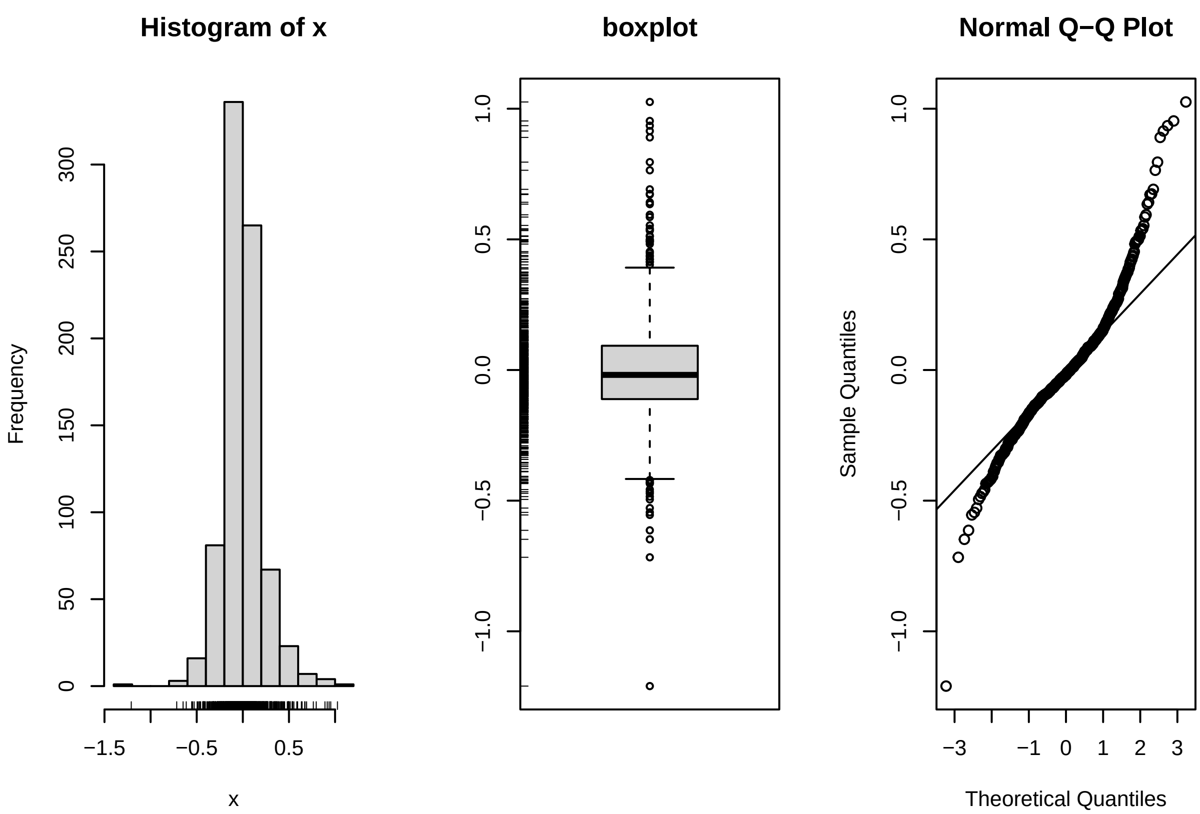

```
shapiro.test(residuals(SMI_mod_hum))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(SMI_mod_hum)
## W = 0.95569, p-value = 7.603e-15
```

```
# temperature model
plot(SMI_mod_temp)
```




```
simple.eda(residuals(SMI_mod_temp))
```



```
shapiro.test(residuals(SMI_mod_temp))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(SMI_mod_temp)
## W = 0.93877, p-value < 2.2e-16
```

Normality is violated, but linearity, equal error variance, and independence are all good.

Comparison

Now, compare the AIC, RMSE, and R^2 values across models, and the F and p values of the variables for each model.

We… calculate RMSE manually, use the r.squaredGLMM function in the MuMIn package to get the marginal R^2, which is how much of the total variance is explained by fixed effects, use the aictab function in the AICmodavg package to get AIC and deltaAIC values, and get the sum of squares, F, and p-values for each variable from the anova table for each model.

```
# calculate RMSE & R^2
SMI_RMSE_Rsq <- data.frame(model =
                              c('Day * VPD',
                                'Day * Humidity',
                                'Day * Temp'
                                ),
                          RMSE = c(sqrt(mean((residuals(SMI_mod_VPD))^2)),
                                   sqrt(mean((residuals(SMI_mod_hum))^2)),
                                   sqrt(mean((residuals(SMI_mod_temp))^2))),
                          # marginal Rsq for the amount of variance
                          # explained by fixed effects only
                          Rsq = c(MuMIn::r.squaredGLMM(SMI_mod_VPD)[,"R2m"],
                                  MuMIn::r.squaredGLMM(SMI_mod_hum)[,"R2m"],
                                  MuMIn::r.squaredGLMM(SMI_mod_temp)[,"R2m"]))
```

```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.
```

```
# calculate AIC
SMI_models <- list(SMI_mod_VPD, SMI_mod_hum, SMI_mod_temp)
EXP_mod_names <- data.frame(model =
                              c('Day * VPD',
                                'Day * Humidity',
                                'Day * Temp'
                                ))
SMI_AICc <- data.frame(aictab(cand.set = SMI_models,
                                modnames = EXP_mod_names))
```

```
## Warning in aictab.AIClmerModLmerTest(cand.set = SMI_models, modnames = EXP_mod_names):
## Model selection for fixed effects is only appropriate with ML estimation:
## REML (default) should only be used to select random effects for a constant set of fixed effects
```

```
# compare across models
SMI_across <- SMI_RMSE_Rsq %>%
  left_join(SMI_AICc, by = 'model') %>%
  mutate(response = "Body Condition (g')") %>%
  arrange(Delta_AICc)
```

```r
# calculate F & p-values
SMI_VPD_p <- data.frame(anova(SMI_mod_VPD,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * VPD',
         predictor = rownames(.))
SMI_hum_p <- data.frame(anova(SMI_mod_hum,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * Humidity',
         predictor = rownames(.))
SMI_temp_p <- data.frame(anova(SMI_mod_temp,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * Temp',
         predictor = rownames(.))

# save within model values
SMI_within <- SMI_VPD_p %>%
  rbind(SMI_hum_p) %>%
  rbind(SMI_temp_p) %>%
  mutate(df = paste((NumDF), round(DenDF, 0), sep = ", "),
         response = "Body Condition (g')")
```

Hematocrit

Building

Build each treatment effect model.

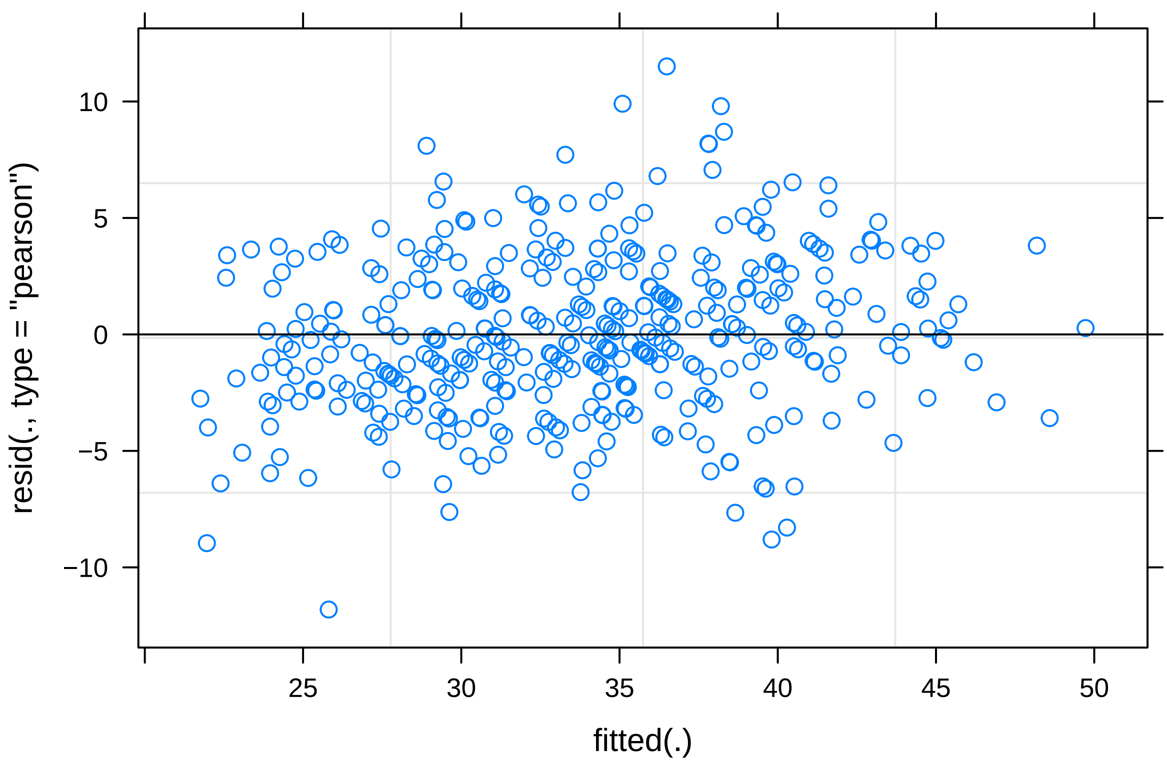
```r
hct_mod_VPD <- lmerTest::lmer(data = dat_no_rehab,
                               hematocrit_percent ~ day_n * VPD_kPa +
                              (1|trial_number/individual_ID))
hct_mod_hum <- lmerTest::lmer(data = dat_no_rehab,
                              hematocrit_percent ~ day_n * humidity_tmt +
                              (1|trial_number/individual_ID))
hct_mod_temp <- lmerTest::lmer(data = dat_no_rehab,
                              hematocrit_percent ~ day_n * temp_tmt +
                              (1|trial_number/individual_ID))
```

Assumptions

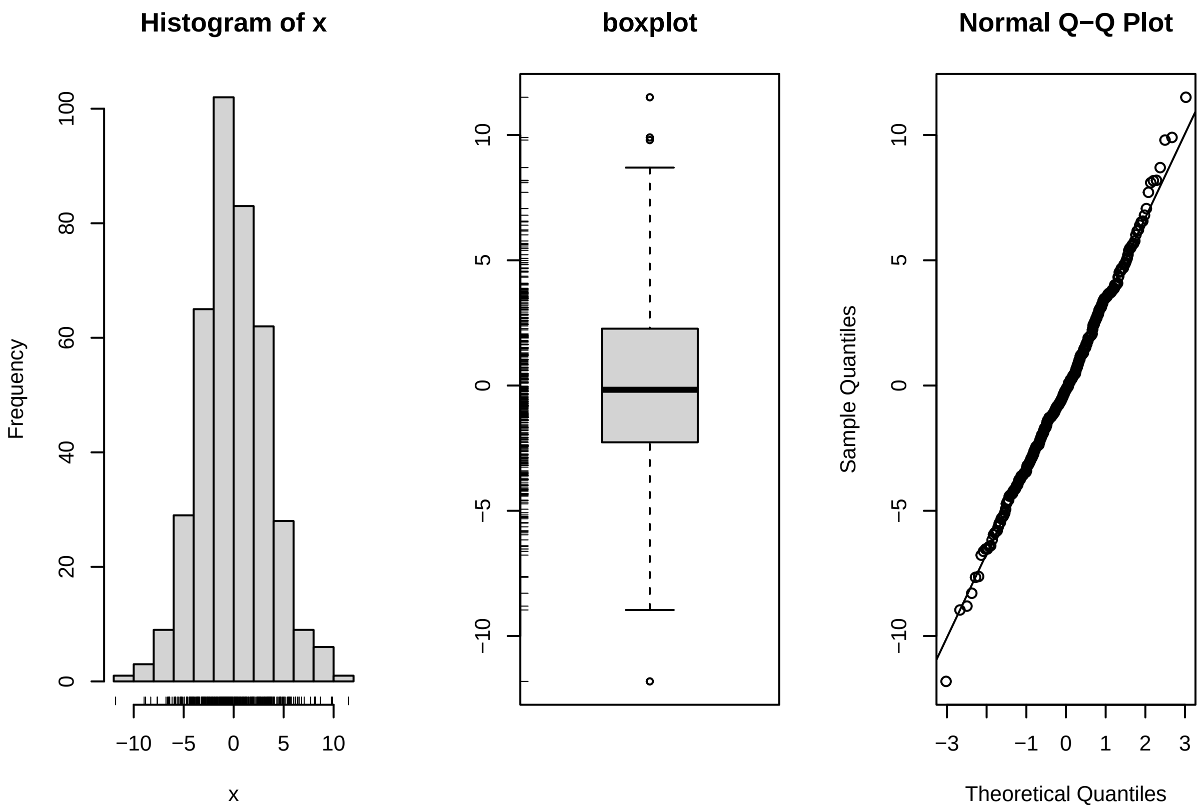
Check linear regression assumptions/conditions.

```r
# distribution of
hist(dat_no_rehab$hematocrit_percent)
```


```
# VPD model
plot(hct_mod_VPD)
```



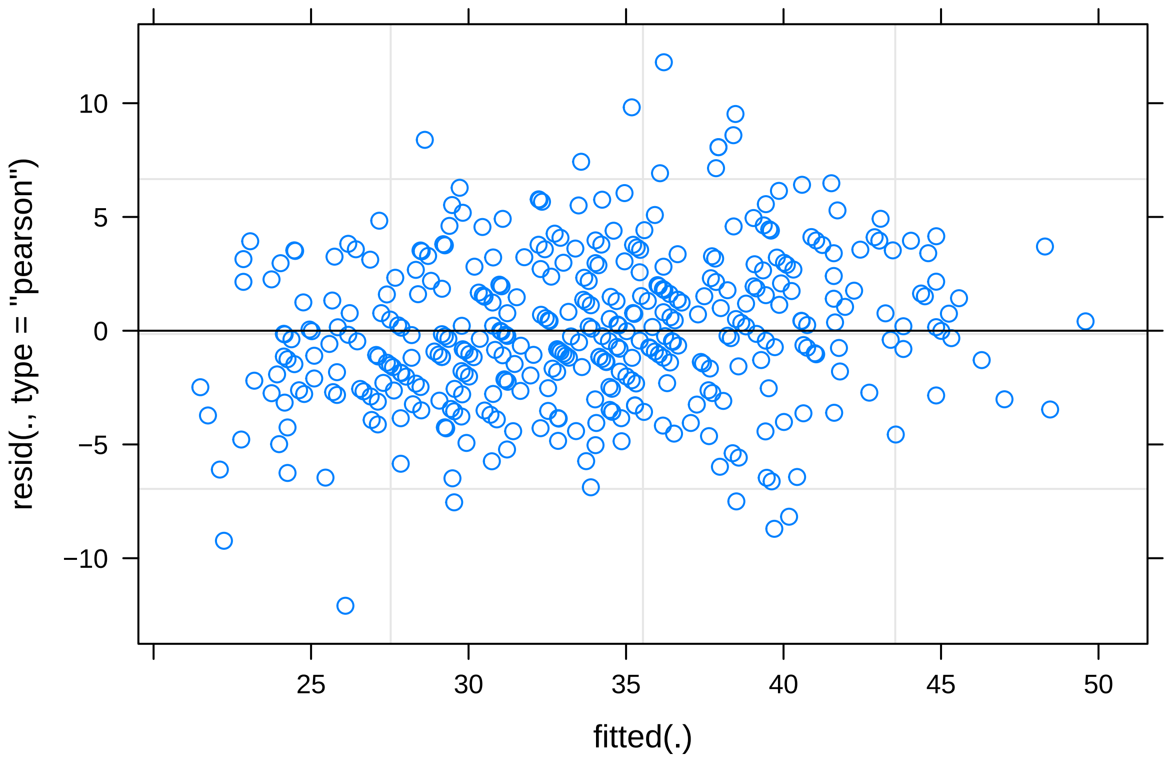

```
simple.eda(residuals(hct_mod_VPD))
```



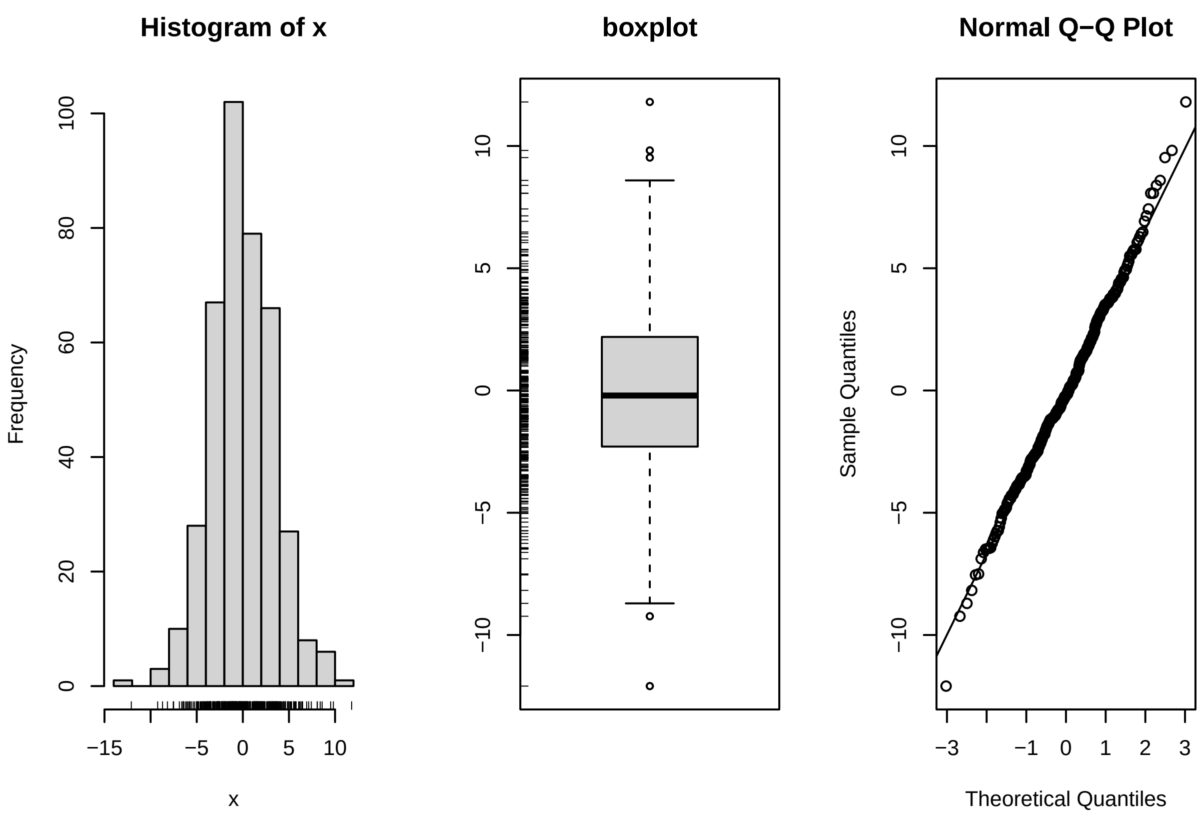

```
shapiro.test(residuals(hct_mod_VPD))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(hct_mod_VPD)
## W = 0.99651, p-value = 0.542
```

```
# humidity model
plot(hct_mod_hum)
```



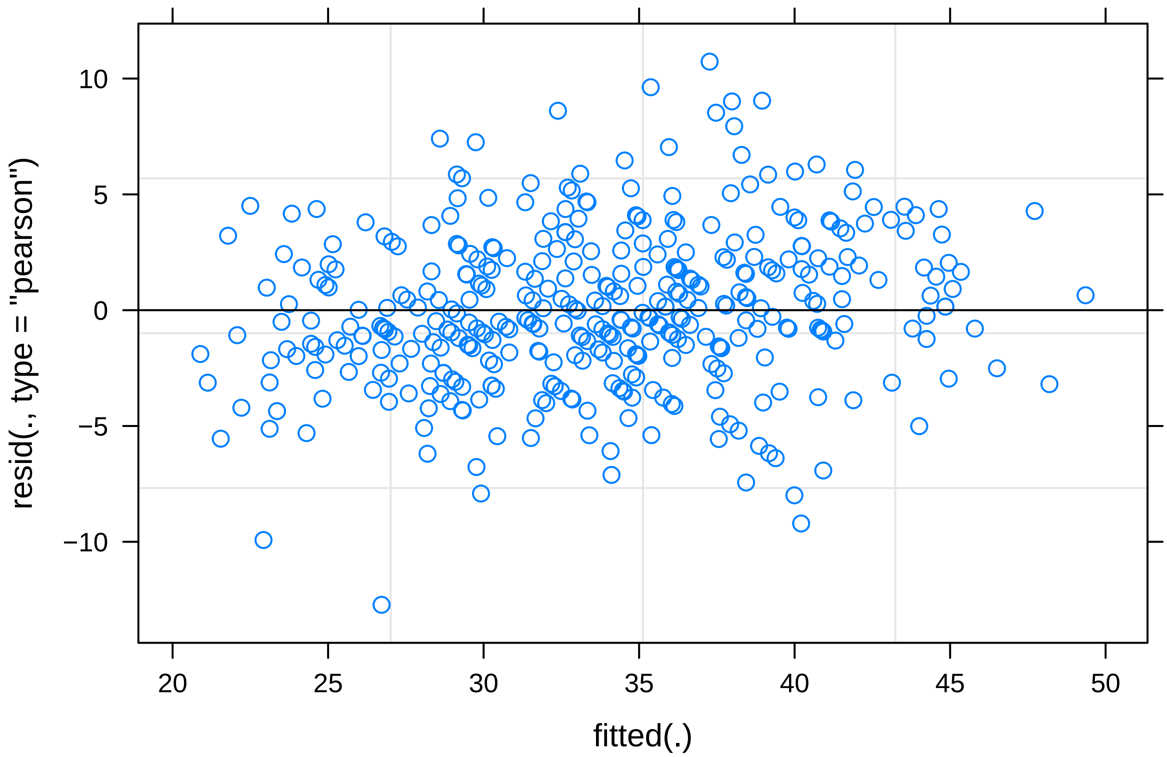

```
simple.eda(residuals(hct_mod_hum))
```



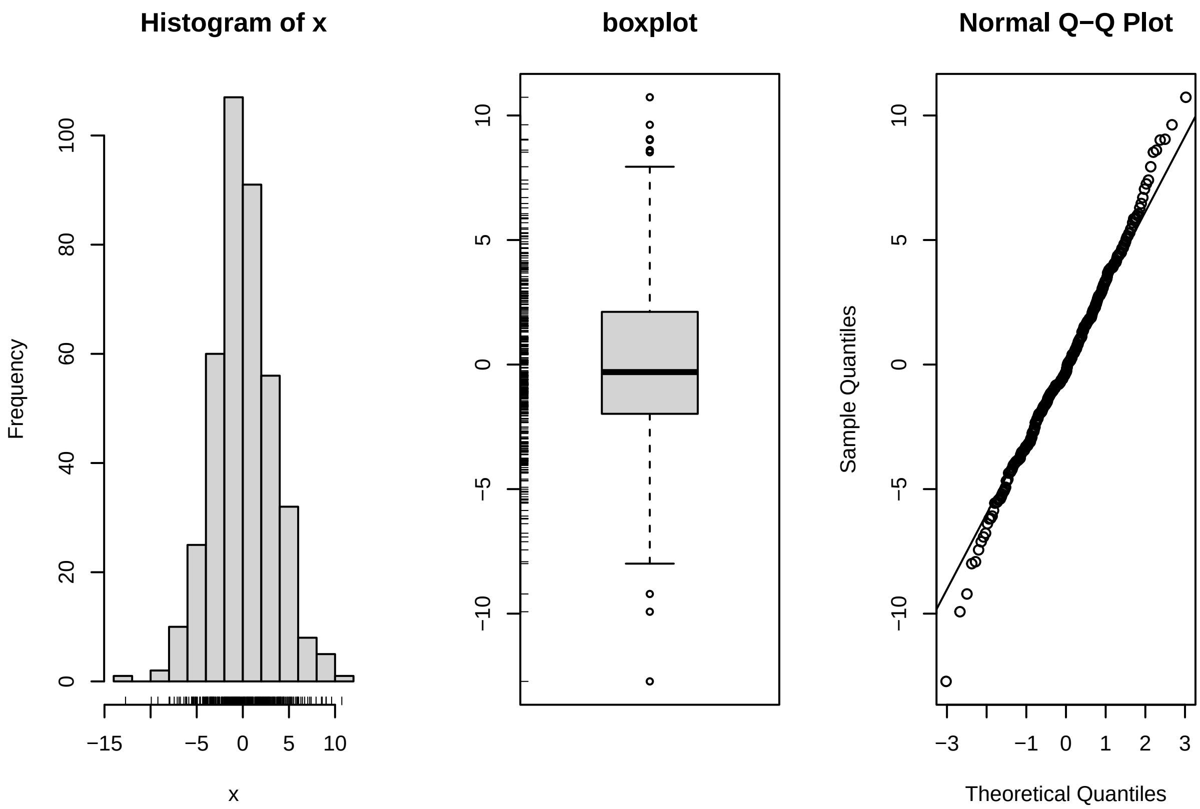
```r
shapiro.test(residuals(hct_mod_hum))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(hct_mod_hum)
## W = 0.99619, p-value = 0.4584
```

```r
# temperature model
plot(hct_mod_temp)
```




```r
simple.eda(residuals(hct_mod_temp))
```




```
shapiro.test(residuals(hct_mod_temp))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(hct_mod_temp)
## W = 0.99478, p-value = 0.1975
```

All assumptions, normality, linearity, equal error variance, and independence are all good.

Comparison

Now, compare the AIC, RMSE, and R^2 values across models, and the F and p values of the variables for each model.

We… calculate RMSE manually, use the r.squaredGLMM function in the MuMIn package to get the marginal R^2, which is how much of the total variance is explained by fixed effects, use the aictab function in the AICmodavg package to get AIC and deltaAIC values, and get the sum of squares, F, and p-values for each variable from the anova table for each model.

```
# calculate RMSE & R^2
hct_RMSE_Rsq <- data.frame(model =
                              c('Day * VPD',
                                'Day * Humidity',
                                'Day * Temp'
                                ),
                           RMSE = c(sqrt(mean((residuals(hct_mod_VPD))^2)),
                                    sqrt(mean((residuals(hct_mod_hum))^2)),
                                    sqrt(mean((residuals(hct_mod_temp))^2))),
                           # marginal Rsq for the amount of variance
```

```r
                        # explained by fixed effects only
                        Rsq = c(MuMIn::r.squaredGLMM(hct_mod_VPD)[,"R2m"],
                                MuMIn::r.squaredGLMM(hct_mod_hum)[,"R2m"],
                                MuMIn::r.squaredGLMM(hct_mod_temp)[,"R2m"]))

# calculate AIC
hct_models <- list(hct_mod_VPD, hct_mod_hum, hct_mod_temp)
hct_AICc <- data.frame(aictab(cand.set = hct_models,
                              modnames = EXP_mod_names))
```

```
## Warning in aictab.AIClmerModLmerTest(cand.set = hct_models, modnames = EXP_mod_names):
## Model selection for fixed effects is only appropriate with ML estimation:
## REML (default) should only be used to select random effects for a constant set of fixed effects
```

```r
# compare across models
hct_across <- hct_RMSE_Rsq %>%
  left_join(hct_AICc, by = 'model') %>%
  mutate(response = "Hematocrit (%)") %>%
  arrange(Delta_AICc)

# calculate F & p-values
hct_VPD_p <- data.frame(anova(hct_mod_VPD,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * VPD',
         predictor = rownames(.))
hct_hum_p <- data.frame(anova(hct_mod_hum,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * Humidity',
         predictor = rownames(.))
hct_temp_p <- data.frame(anova(hct_mod_temp,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * Temp',
         predictor = rownames(.))

# save within model values
hct_within <- hct_VPD_p %>%
  rbind(hct_hum_p) %>%
  rbind(hct_temp_p) %>%
  mutate(df = paste((NumDF), round(DenDF, 0), sep = ", "),
         response = "Hematocrit (%)")
```

Osmolality

Building

Build each treatment effect model.

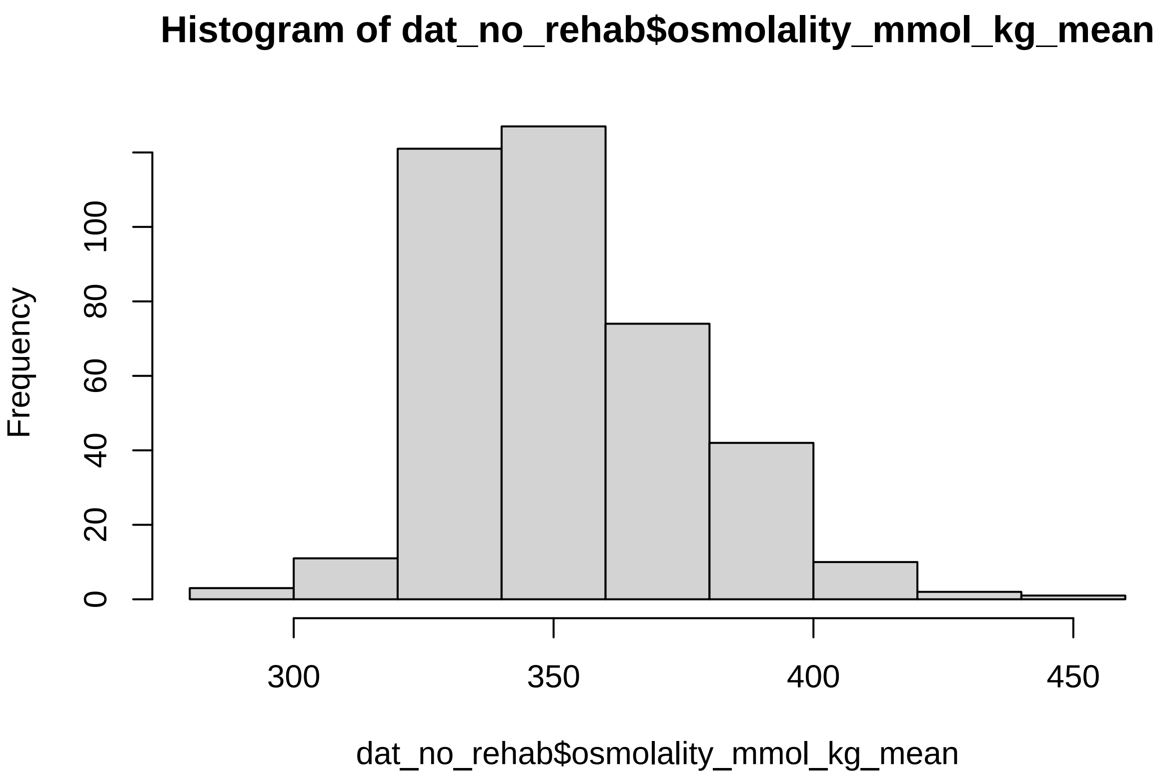
```r
osml_mod_VPD <- lmerTest::lmer(data = dat_no_rehab,
                               osmolality_mmol_kg_mean ~ day_n * VPD_kPa +
                              (1|trial_number/individual_ID))
osml_mod_hum <- lmerTest::lmer(data = dat_no_rehab,
                               osmolality_mmol_kg_mean ~ day_n * humidity_tmt +
```

```r
                               (1|trial_number/individual_ID))
osml_mod_temp <- lmerTest::lmer(data = dat_no_rehab,
                                osmolality_mmol_kg_mean ~ day_n * temp_tmt +
                               (1|trial_number/individual_ID))
```

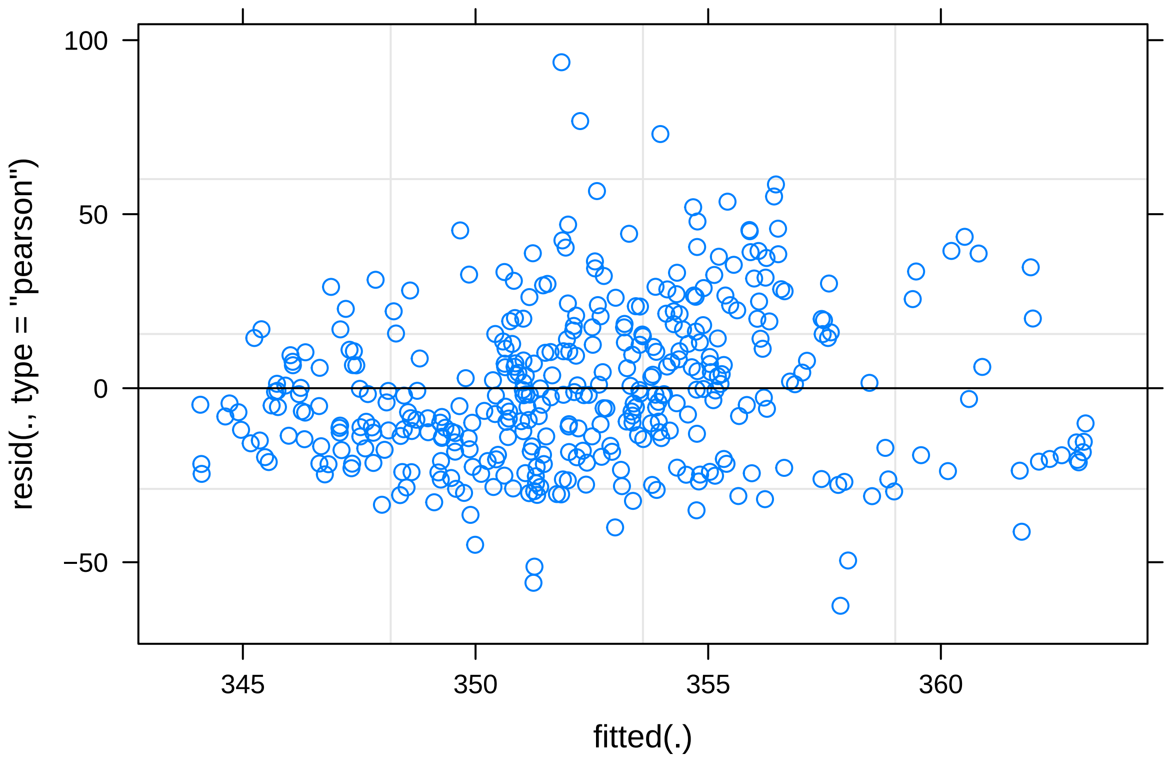
Assumptions

Check linear regression assumptions/conditions.

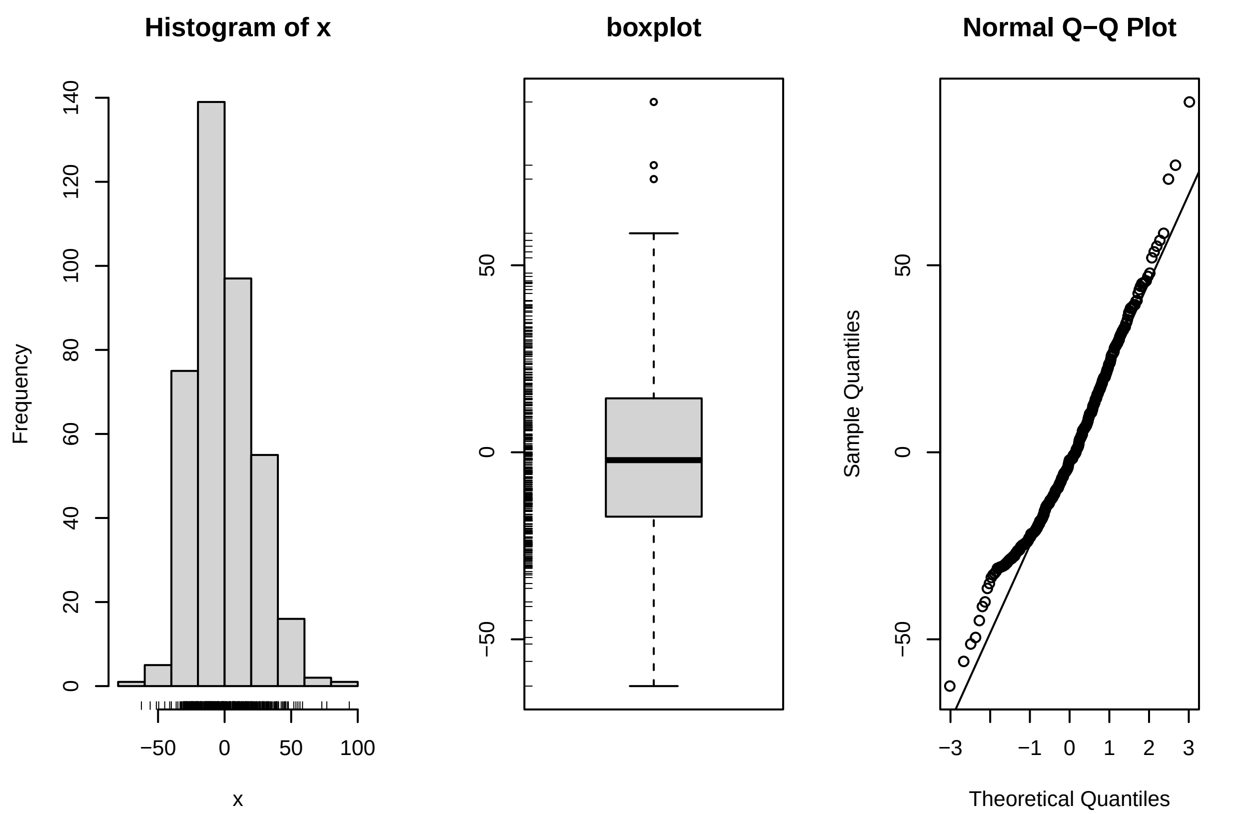
```r
# distribution of
hist(dat_no_rehab$osmolality_mmol_kg_mean)
```




```r
# VPD model
plot(osml_mod_VPD)
```



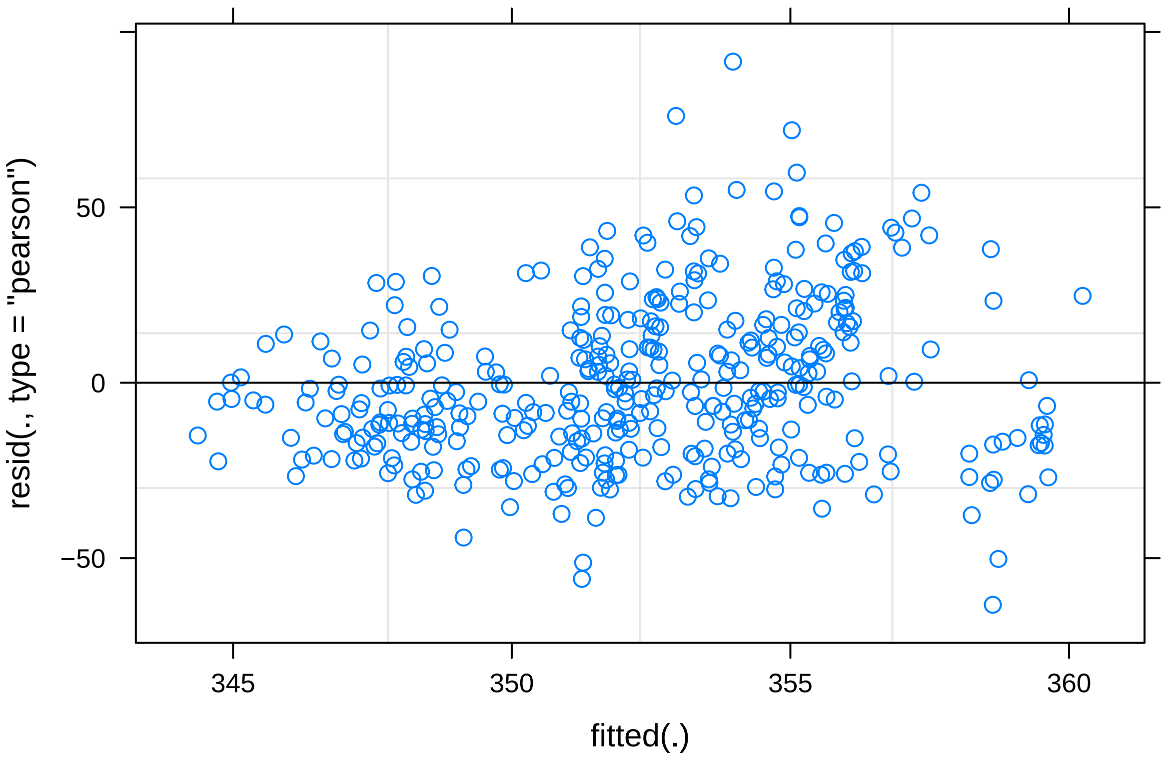

```
simple.eda(residuals(osml_mod_VPD))
```



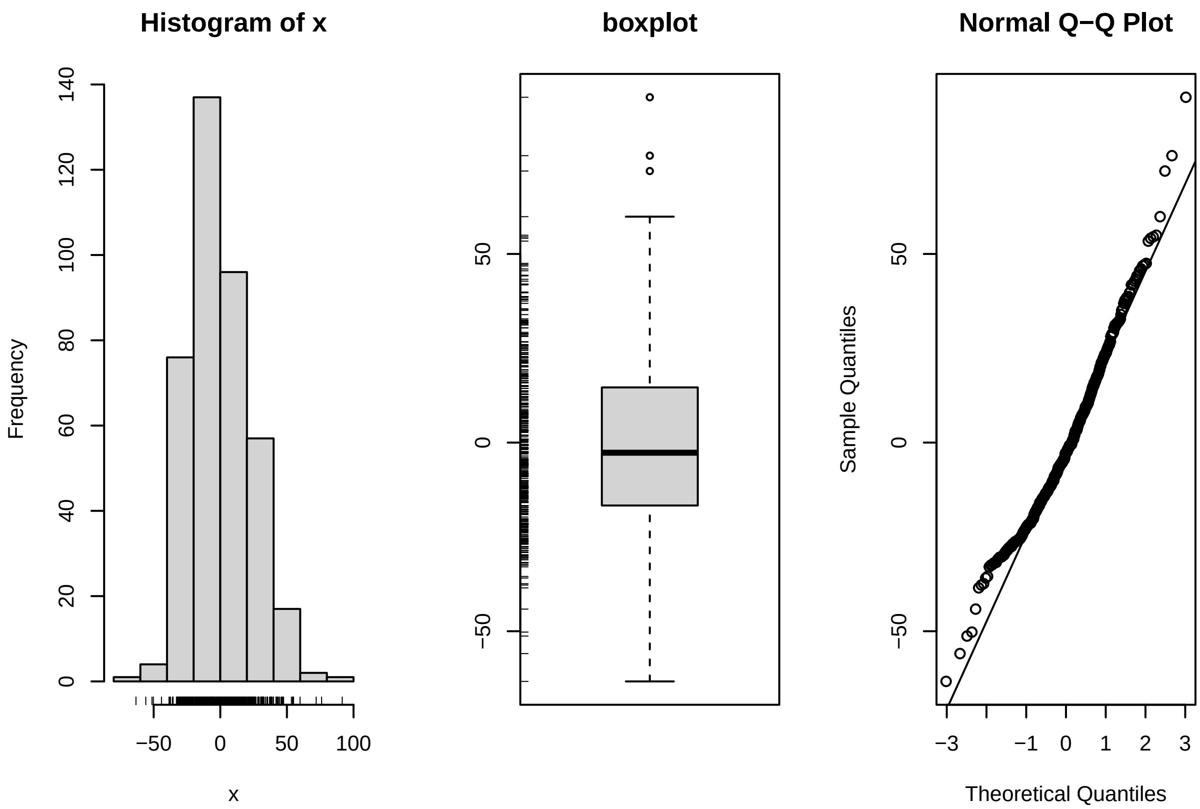
```
shapiro.test(residuals(osml_mod_VPD))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(osml_mod_VPD)
## W = 0.9769, p-value = 6.836e-06
```

```
# humidity model
plot(osml_mod_hum)
```



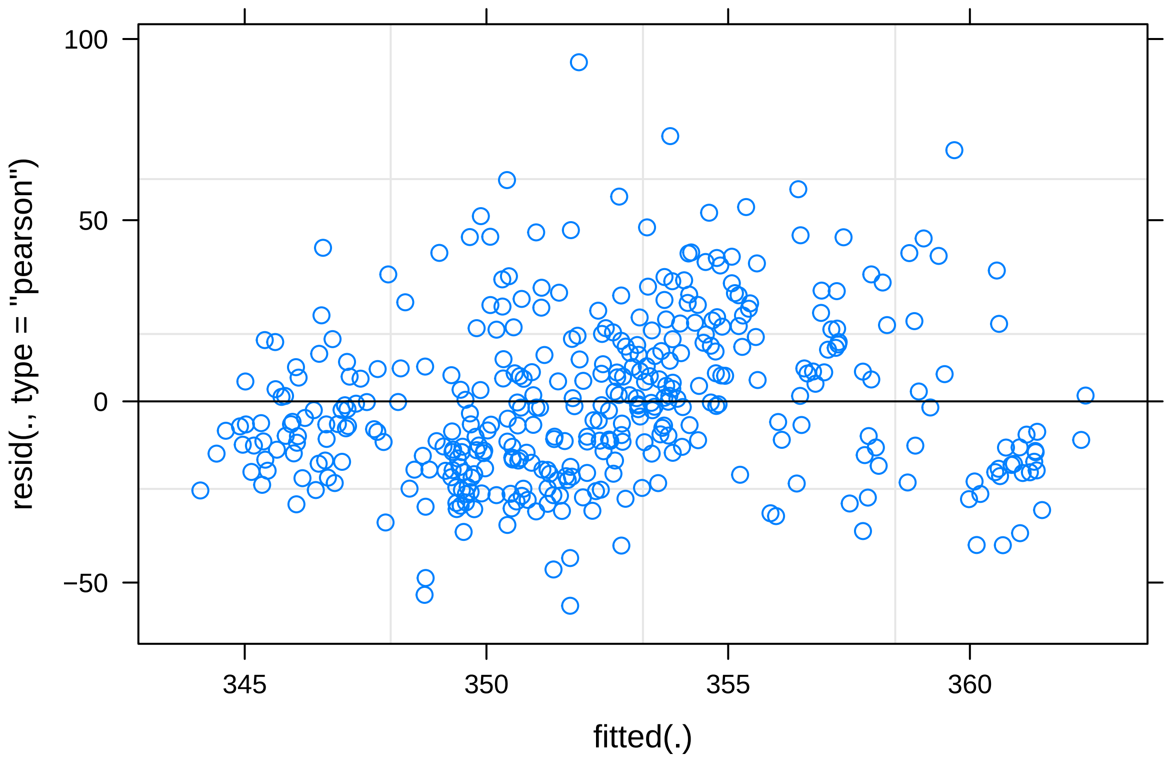

```
simple.eda(residuals(osml_mod_hum))
```



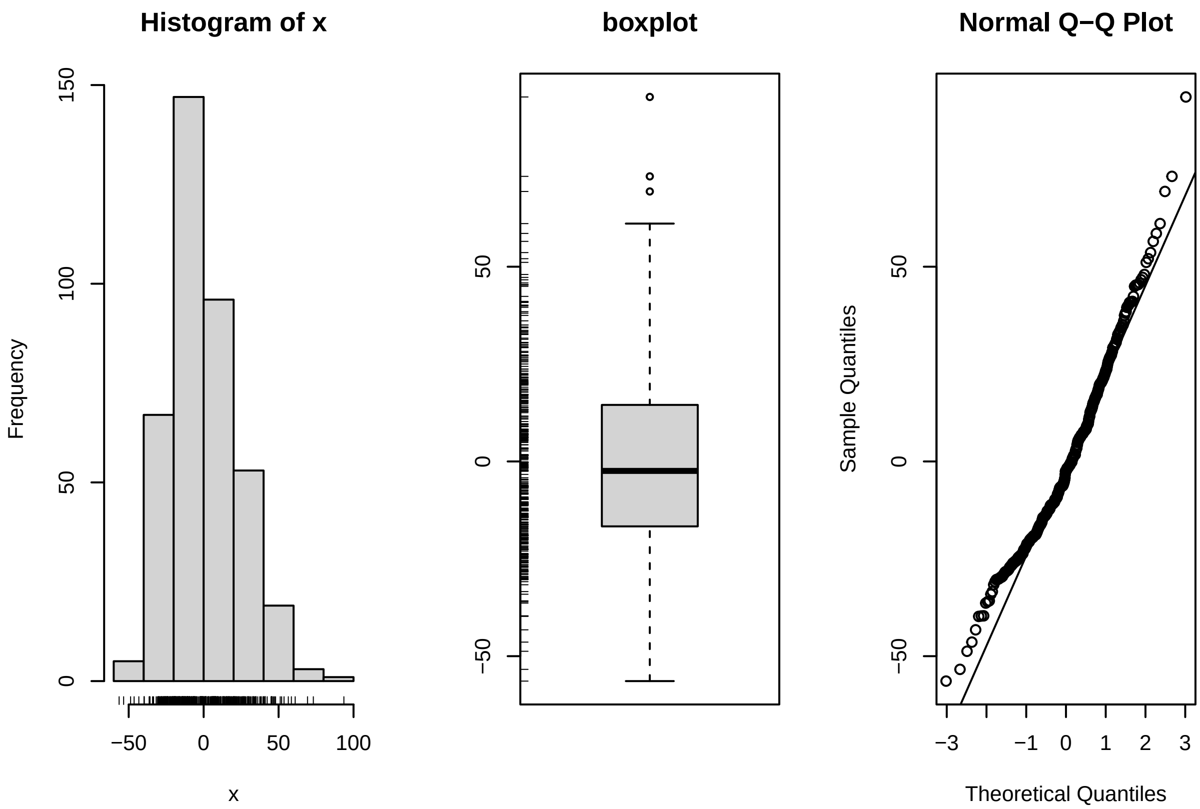

```
shapiro.test(residuals(osml_mod_hum))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(osml_mod_hum)
## W = 0.97746, p-value = 8.914e-06
```

```
# temperature model
plot(osml_mod_temp)
```




```
simple.eda(residuals(osml_mod_temp))
```



```
shapiro.test(residuals(osml_mod_temp))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(osml_mod_temp)
## W = 0.97346, p-value = 1.44e-06
```

Normality is violated, but linearity, equal error variance, and independence are all okay.

Comparison

Now, compare the AIC, RMSE, and R^2 values across models, and the F and p values of the variables for each model.

We… calculate RMSE manually, use the r.squaredGLMM function in the MuMIn package to get the marginal R^2, which is how much of the total variance is explained by fixed effects, use the aictab function in the AICmodavg package to get AIC and deltaAIC values, and get the sum of squares, F, and p-values for each variable from the anova table for each model.

```
# calculate RMSE & R^2
osml_RMSE_Rsq <- data.frame(model =
                              c('Day * VPD',
                                'Day * Humidity',
                                'Day * Temp'
                                ),
                          RMSE = c(sqrt(mean((residuals(osml_mod_VPD))^2)),
                                   sqrt(mean((residuals(osml_mod_hum))^2)),
                                   sqrt(mean((residuals(osml_mod_temp))^2))),
                          # marginal Rsq for the amount of variance
                          # explained by fixed effects only
                          Rsq = c(MuMIn::r.squaredGLMM(osml_mod_VPD)[,"R2m"],
                                  MuMIn::r.squaredGLMM(osml_mod_hum)[,"R2m"],
                                  MuMIn::r.squaredGLMM(osml_mod_temp)[,"R2m"]))

# calculate AIC
osml_models <- list(osml_mod_VPD, osml_mod_hum, osml_mod_temp)
osml_AICc <- data.frame(aictab(cand.set = osml_models,
                               modnames = EXP_mod_names))
```

```
## Warning in aictab.AIClmerModLmerTest(cand.set = osml_models, modnames = EXP_mod_names):
## Model selection for fixed effects is only appropriate with ML estimation:
## REML (default) should only be used to select random effects for a constant set of fixed effects
```

```
# compare across models
osml_across <- osml_RMSE_Rsq %>%
  left_join(osml_AICc, by = 'model') %>%
  mutate(response = "Plasma Osmolality (mmol/kg)") %>%
  arrange(Delta_AICc)

# calculate F & p-values
osml_VPD_p <- data.frame(anova(osml_mod_VPD,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * VPD',
         predictor = rownames(.))
```

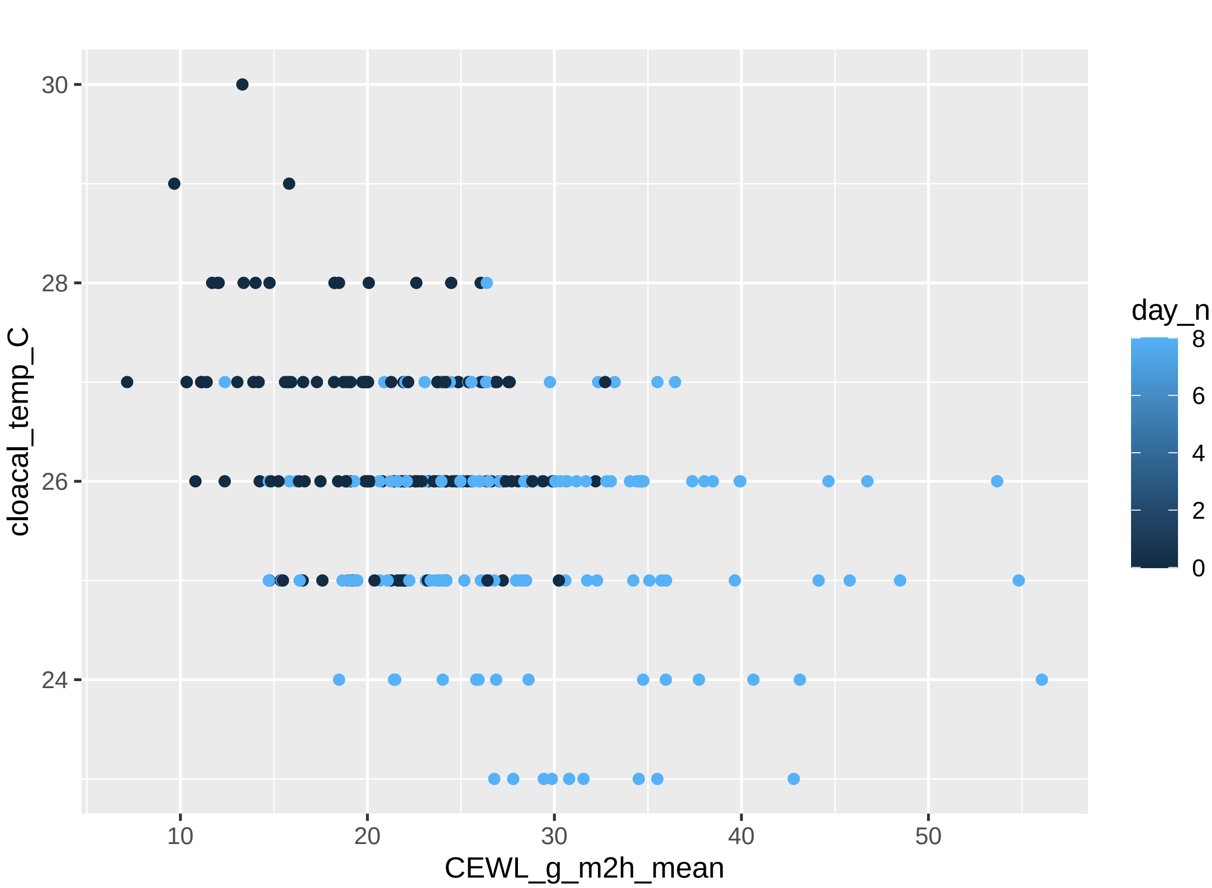
```r
osml_hum_p <- data.frame(anova(osml_mod_hum,
                               type = "1",
                               ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * Humidity',
         predictor = rownames(.))
osml_temp_p <- data.frame(anova(osml_mod_temp,
                               type = "1",
                               ddf = "Kenward-Roger")) %>%
  mutate(model = 'Day * Temp',
         predictor = rownames(.))

# save within model values
osml_within <- osml_VPD_p %>%
  rbind(osml_hum_p) %>%
  rbind(osml_temp_p) %>%
  mutate(df = paste((NumDF), round(DenDF, 0), sep = ", "),
         response = "Plasma Osmolality (mmol/kg)")
```

Body Temp

I need to double check whether CEWL has a relationship with body temperature at the point of measurement.

```r
ggplot(dat_no_rehab) +
  geom_point(aes(x = CEWL_g_m2h_mean,
                 y = cloacal_temp_C,
                 color = day_n))
```

```
## Warning: Removed 537 rows containing missing values (`geom_point()`).
```



Test an lm of raw CEWL ~ body temp for day 8 measurements only.

```
body_temp_test_dat <- dat_no_rehab_deltaCEWL %>%
  dplyr::filter(complete.cases(CEWL_g_m2h_mean)) %>%
  dplyr::filter(day_n == 8)

CEWL_body_temp_mod <- lmerTest::lmer(data = body_temp_test_dat,
                         CEWL_g_m2h_mean ~ cloacal_temp_C * tmt +
                           (1|trial_number))
summary(CEWL_body_temp_mod)
```

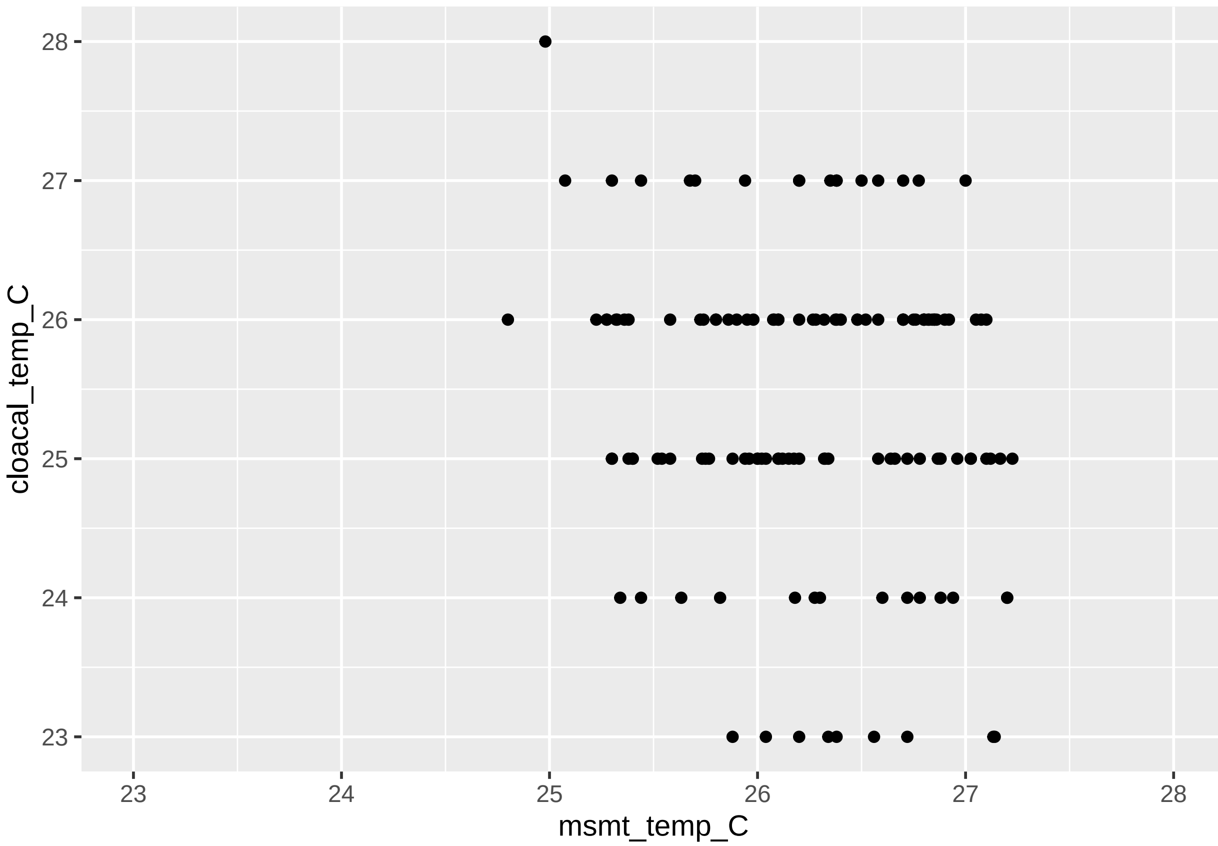
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: CEWL_g_m2h_mean ~ cloacal_temp_C * tmt + (1 | trial_number)
##    Data: body_temp_test_dat
##
## REML criterion at convergence: 839.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.02426 -0.58867 -0.09161  0.46172  3.09926
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  trial_number (Intercept)  4.977   2.231
##  Residual                 36.926   6.077
## Number of obs: 133, groups:  trial_number, 5
##
```

```
## Fixed effects:
##                                    Estimate Std. Error        df t value
## (Intercept)                         42.83759   26.65061 123.21107   1.607
## cloacal_temp_C                      -0.49420    1.05450 122.93003  -0.469
## tmtHot Humid (1.1 kPa)             -16.46275   34.17703 122.09739  -0.482
## tmtCool Dry (2.5 kPa)              -23.23095   47.80497 123.10602  -0.486
## tmtHot Dry (3.8 kPa)                -3.95989   36.73317 123.70420  -0.108
## cloacal_temp_C:tmtHot Humid (1.1 kPa)   0.90966    1.35821 122.13178   0.670
## cloacal_temp_C:tmtCool Dry (2.5 kPa)    0.65396    1.86629 123.06213   0.350
## cloacal_temp_C:tmtHot Dry (3.8 kPa)    -0.06271    1.44351 123.65911  -0.043
##                                    Pr(>|t|)
## (Intercept)                           0.111
## cloacal_temp_C                        0.640
## tmtHot Humid (1.1 kPa)                0.631
## tmtCool Dry (2.5 kPa)                 0.628
## tmtHot Dry (3.8 kPa)                  0.914
## cloacal_temp_C:tmtHot Humid (1.1 kPa)    0.504
## cloacal_temp_C:tmtCool Dry (2.5 kPa)     0.727
## cloacal_temp_C:tmtHot Dry (3.8 kPa)      0.965
##
## Correlation of Fixed Effects:
##             (Intr) clc__C tHH(1k tCD(2k tHD(3k c__H(k c__C:CD(k
## clocl_tmp_C -0.999
## tmHH(1.1kP) -0.745  0.744
## tmCD(2.5kP) -0.528  0.528  0.434
## tmHD(3.8kP) -0.694  0.694  0.564  0.422
## c__C:HH(1.k  0.741 -0.741 -0.999 -0.432 -0.562
## c__C:CD(2.k  0.536 -0.536 -0.439 -0.999 -0.427  0.438
## c__C:HD(3.k  0.699 -0.700 -0.567 -0.425 -0.999  0.566  0.430
```

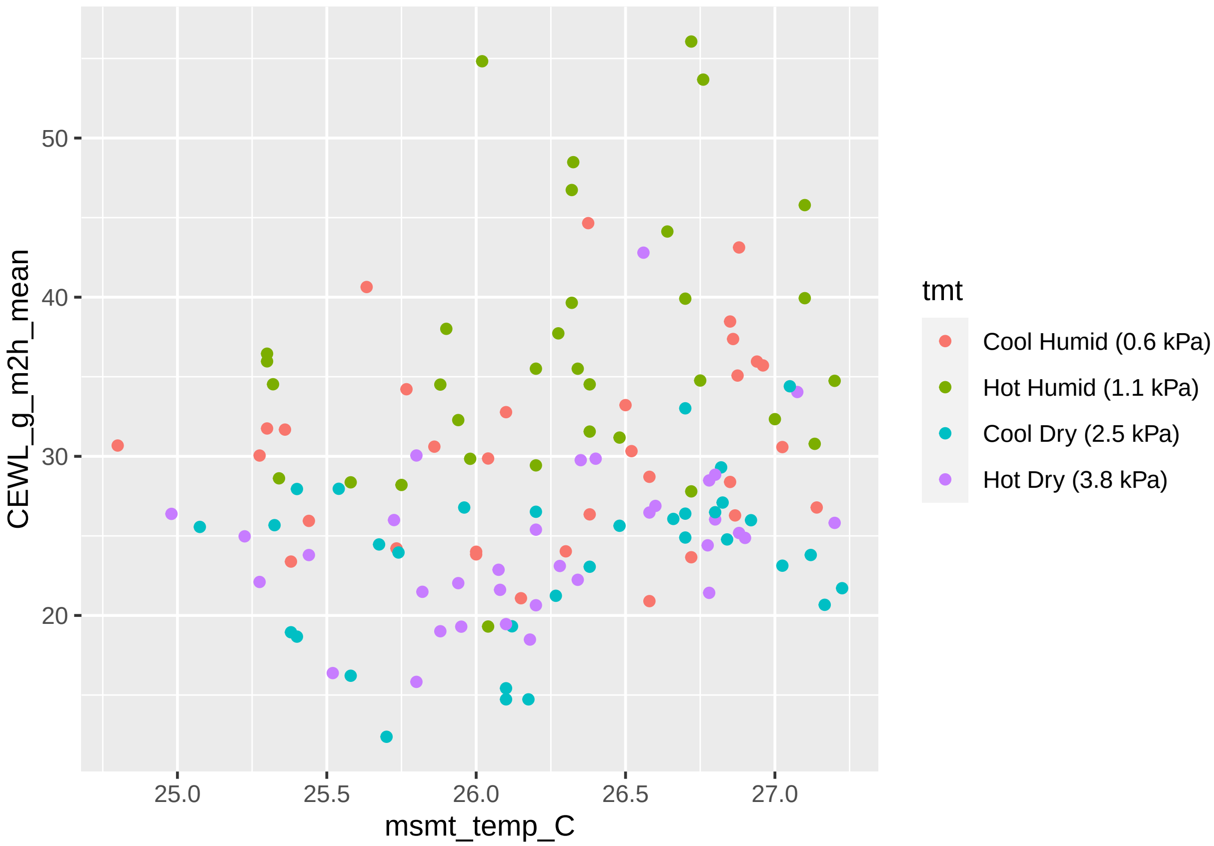
```
anova(CEWL_body_temp_mod, type = "1", ddf = "Kenward-Roger")
```

```
## Type I Analysis of Variance Table with Kenward-Roger's method
##                    Sum Sq Mean Sq NumDF  DenDF F value   Pr(>F)
## cloacal_temp_C     230.75  230.75     1 115.89  6.2491  0.01383 *
## tmt               3115.05 1038.35     3 122.21 28.1193 6.75e-14 ***
## cloacal_temp_C:tmt   27.57    9.19     3 122.56  0.2489  0.86199
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggplot(body_temp_test_dat) +
  geom_point(aes(x = msmt_temp_C,
                 y = cloacal_temp_C)) +
  xlim(23, 28) + ylim(23,28)
```




```
ggplot(body_temp_test_dat) +
  geom_point(aes(x = msmt_temp_C,
                 y = CEWL_g_m2h_mean,
                 color = tmt))
```



CEWL

First, get a separate df for each tmt group:

```
HH8 <- body_temp_test_dat %>%
  dplyr::filter(substr(tmt, 1, 6) == "Hot Hu")
HD8 <- body_temp_test_dat %>%
  dplyr::filter(substr(tmt, 1, 6) == "Hot Dr")
CH8 <- body_temp_test_dat %>%
  dplyr::filter(substr(tmt, 1, 6) == "Cool H")
CD8 <- body_temp_test_dat %>%
  dplyr::filter(substr(tmt, 1, 6) == "Cool D")
```

t-tests

```
CEWL_ttest_HH <- t.test(HH8$delta_CEWL,
                        mu = 0, alternative = "two.sided")
CEWL_ttest_HH
```

```
##
##  One Sample t-test
##
## data:  HH8$delta_CEWL
## t = 8.7341, df = 32, p-value = 5.573e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  11.76821 18.92672
```

```
## sample estimates:
## mean of x
##  15.34746
```

```r
CEWL_ttest_HD <- t.test(HD8$delta_CEWL,
                        mu = 0, alternative = "two.sided")
CEWL_ttest_HD
```

```
##
##  One Sample t-test
##
## data:  HD8$delta_CEWL
## t = 2.513, df = 33, p-value = 0.01703
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.7042497 6.6929758
## sample estimates:
## mean of x
##  3.698613
```

```r
CEWL_ttest_CH <- t.test(CH8$delta_CEWL,
                        mu = 0, alternative = "two.sided")
CEWL_ttest_CH
```

```
##
##  One Sample t-test
##
## data:  CH8$delta_CEWL
## t = 8.7488, df = 32, p-value = 5.363e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   7.296566 11.725292
## sample estimates:
## mean of x
##  9.510929
```

```r
CEWL_ttest_CD <- t.test(CD8$delta_CEWL,
                        mu = 0, alternative = "two.sided")
CEWL_ttest_CD
```

```
##
##  One Sample t-test
##
## data:  CD8$delta_CEWL
## t = 2.75, df = 32, p-value = 0.009721
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.9302207 6.2446177
## sample estimates:
## mean of x
##  3.587419
```

Building

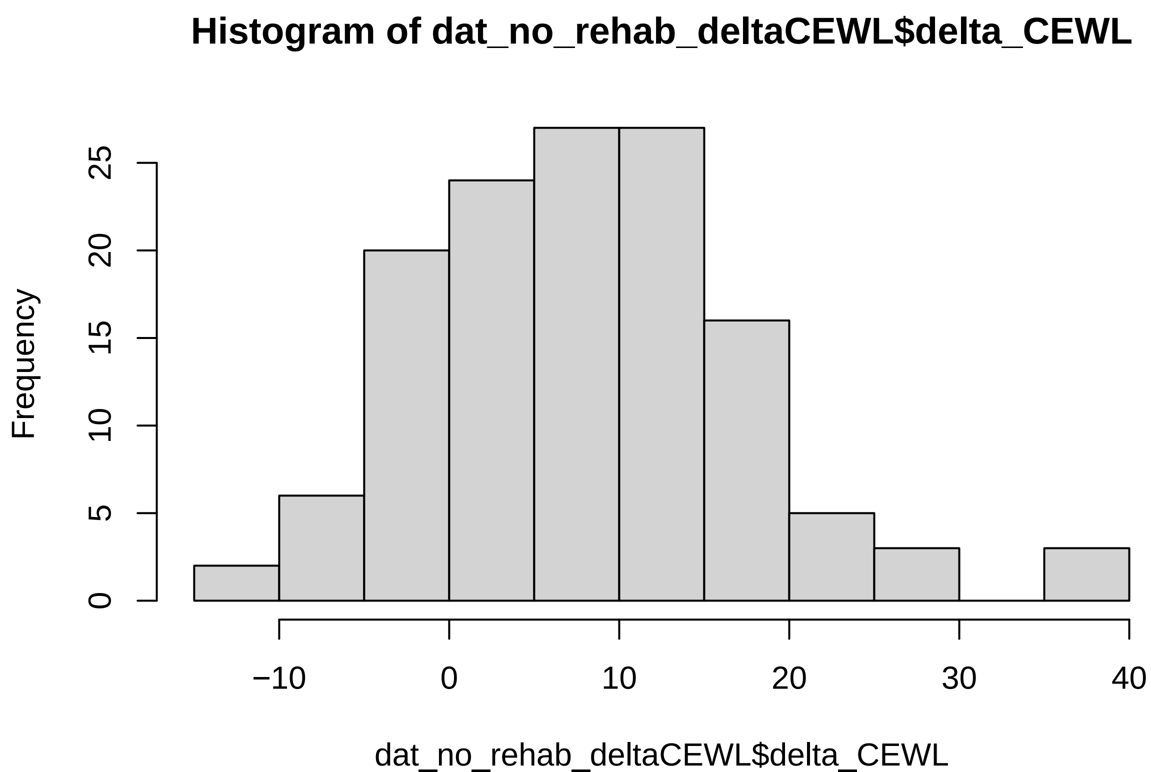
Build each treatment effect model.

```r
CEWL_mod_VPD <- lmerTest::lmer(data = dat_no_rehab_deltaCEWL,
                               delta_CEWL ~ VPD_kPa +
                              (1|trial_number))
CEWL_mod_hum <- lmerTest::lmer(data = dat_no_rehab_deltaCEWL,
                               delta_CEWL ~ humidity_tmt +
                              (1|trial_number))
CEWL_mod_temp <- lmerTest::lmer(data = dat_no_rehab_deltaCEWL,
                               delta_CEWL ~ temp_tmt +
                              (1|trial_number))
```

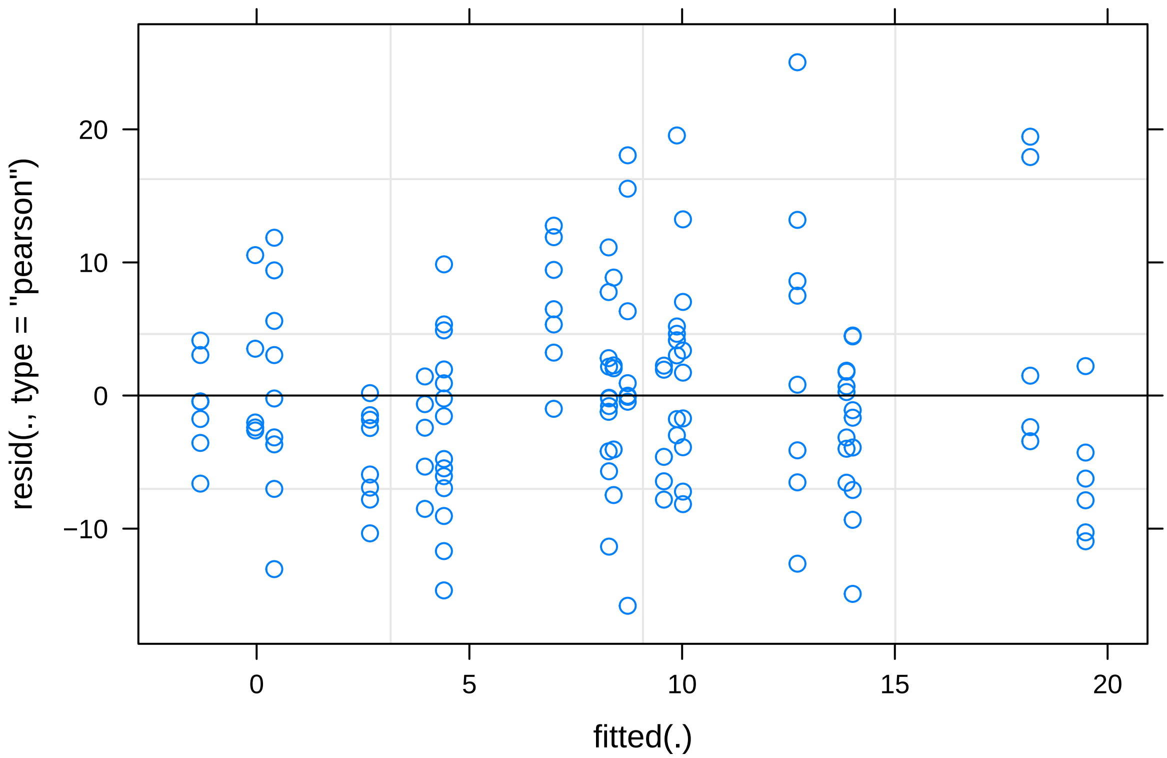
Assumptions

Check linear regression assumptions/conditions.

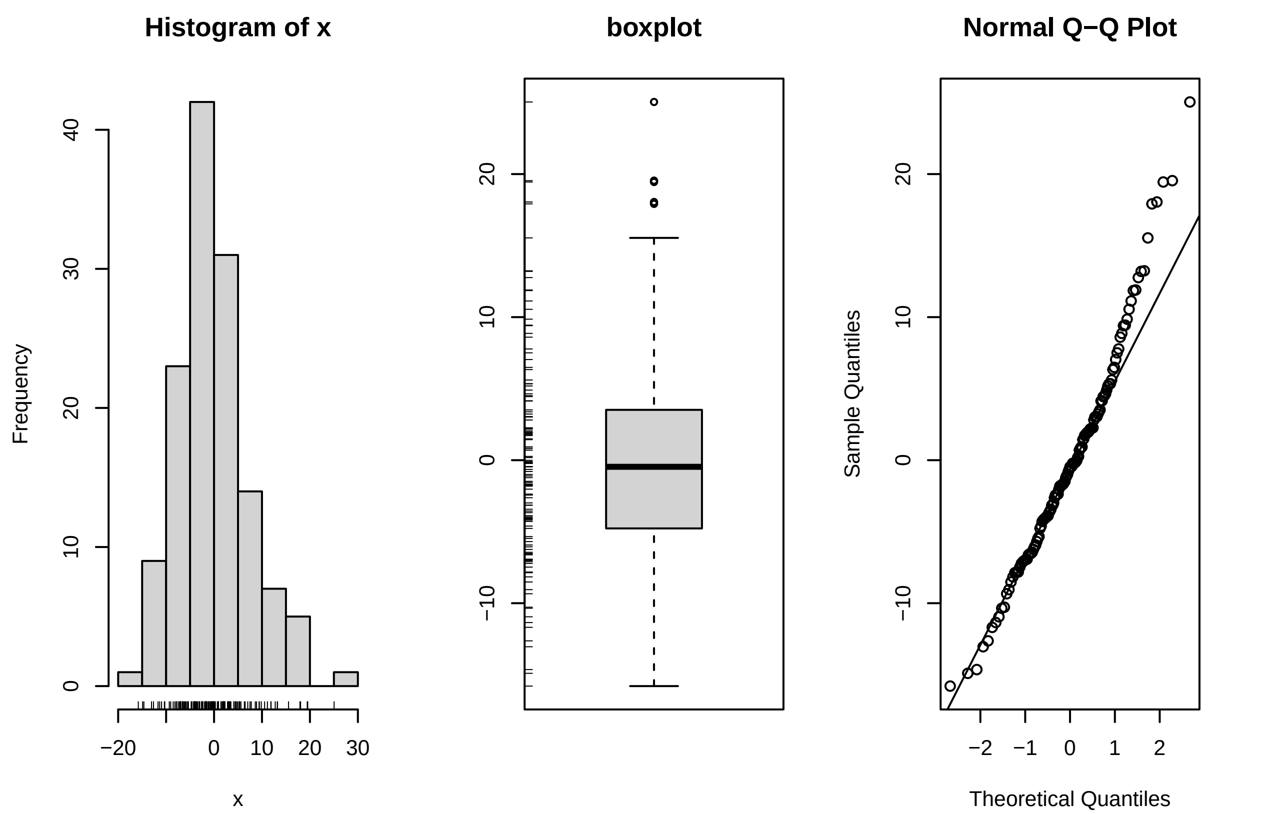
```r
# distribution of
hist(dat_no_rehab_deltaCEWL$delta_CEWL)
```




```r
# VPD model
plot(CEWL_mod_VPD)
```



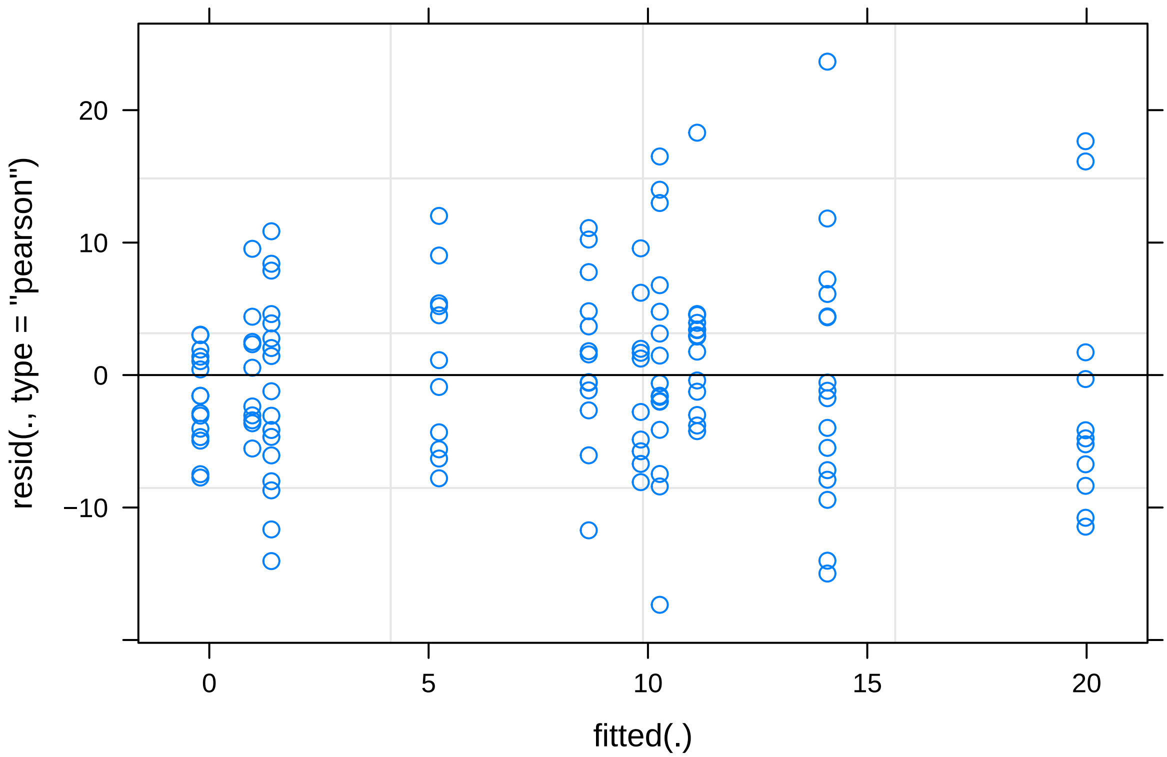

```
simple.eda(residuals(CEWL_mod_VPD))
```



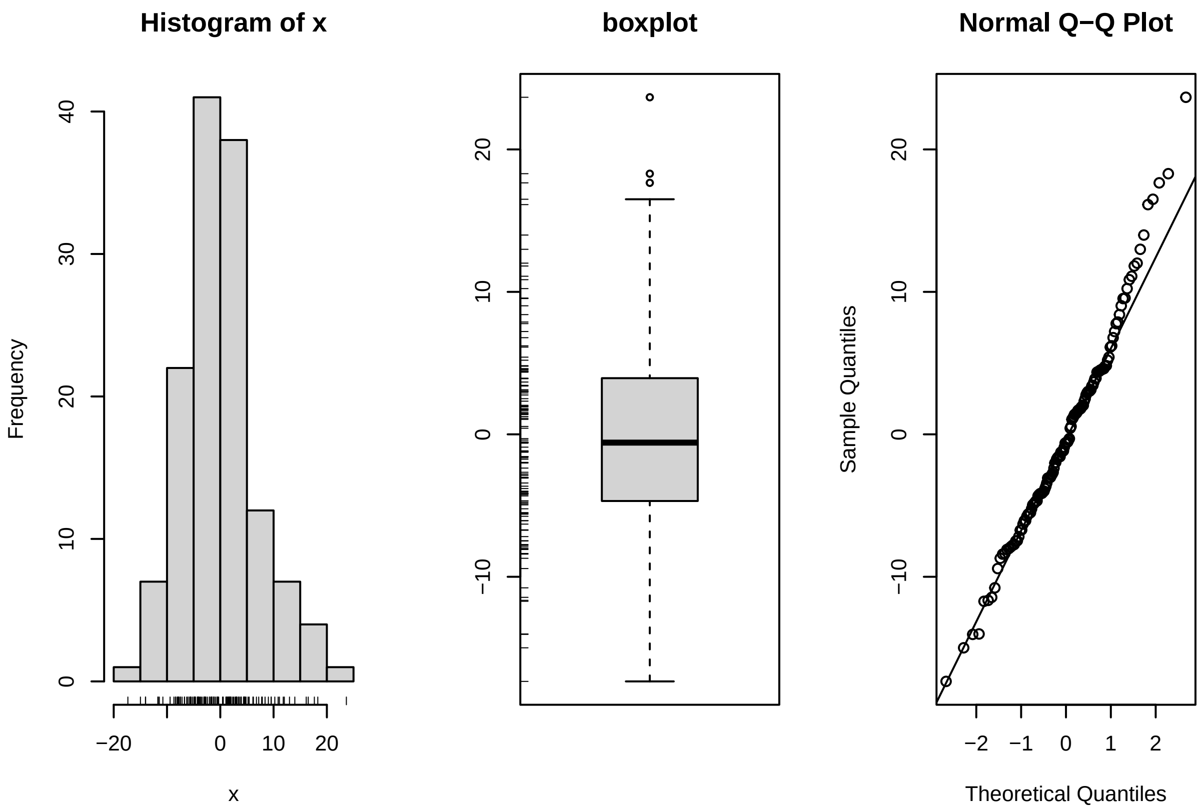
```r
shapiro.test(residuals(CEWL_mod_VPD))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(CEWL_mod_VPD)
## W = 0.97332, p-value = 0.01016
```

```r
# humidity model
plot(CEWL_mod_hum)
```



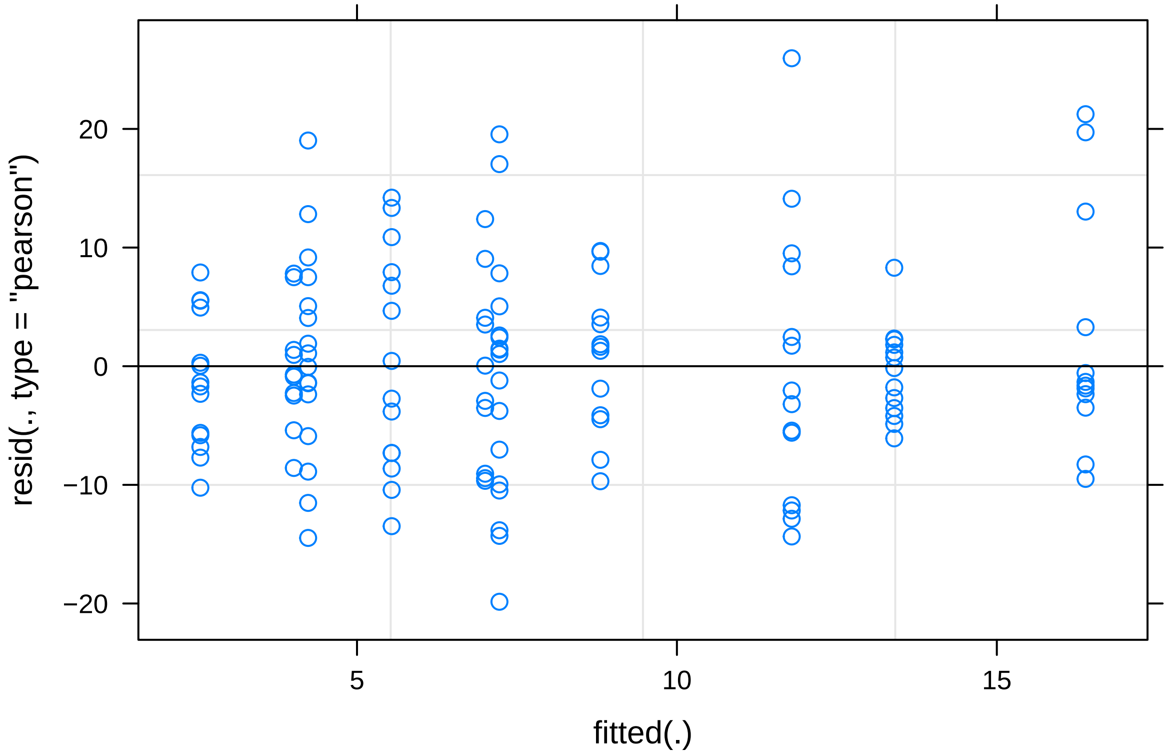

```r
simple.eda(residuals(CEWL_mod_hum))
```



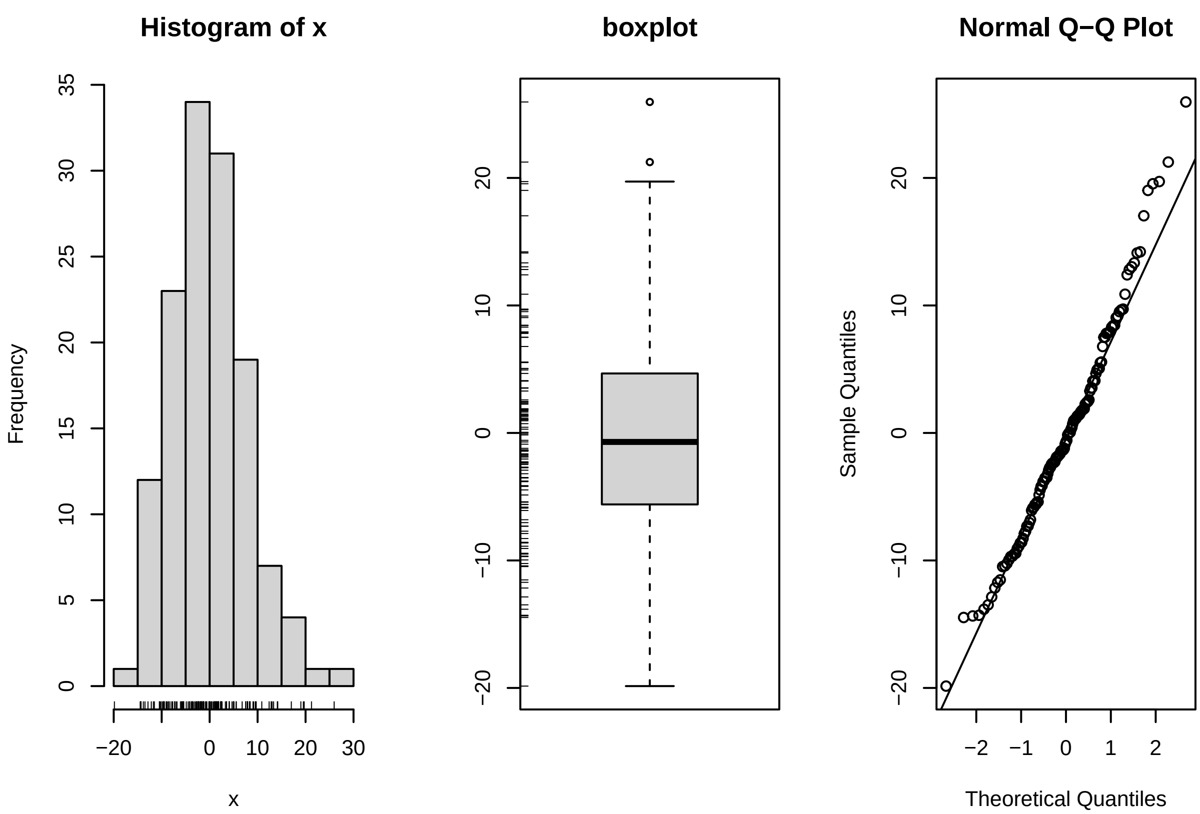

```r
shapiro.test(residuals(CEWL_mod_hum))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(CEWL_mod_hum)
## W = 0.98319, p-value = 0.1002
```

```r
# temperature model
plot(CEWL_mod_temp)
```




```
simple.eda(residuals(CEWL_mod_temp))
```



```r
shapiro.test(residuals(CEWL_mod_temp))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(CEWL_mod_temp)
## W = 0.98262, p-value = 0.08761
```

All assumptions are fine.

Comparison

Now, compare the AIC, RMSE, and R^2 values across models, and the F and p values of the variables for each model.

We… calculate RMSE manually, use the r.squaredGLMM function in the MuMIn package to get the marginal R^2, which is how much of the total variance is explained by fixed effects, use the aictab function in the AICmodavg package to get AIC and deltaAIC values, and get the sum of squares, F, and p-values for each variable from the anova table for each model.

```r
# calculate RMSE & R^2
CEWL_RMSE_Rsq <- data.frame(model =
                              c('VPD',
                                'Humidity',
                                'Temp'
                                ),
                          RMSE = c(sqrt(mean((residuals(CEWL_mod_VPD))^2)),
                                   sqrt(mean((residuals(CEWL_mod_hum))^2)),
                                   sqrt(mean((residuals(CEWL_mod_temp))^2))),
                          # marginal Rsq for the amount of variance
                          # explained by fixed effects only
                          Rsq = c(MuMIn::r.squaredGLMM(CEWL_mod_VPD)[,"R2m"],
                                  MuMIn::r.squaredGLMM(CEWL_mod_hum)[,"R2m"],
                                  MuMIn::r.squaredGLMM(CEWL_mod_temp)[,"R2m"]))

# calculate AIC
CEWL_models <- list(CEWL_mod_VPD, CEWL_mod_hum, CEWL_mod_temp)
CEWL_mod_names <- data.frame(model =
                              c('VPD',
                                'Humidity',
                                'Temp'
                                ))
CEWL_AICc <- data.frame(aictab(cand.set = CEWL_models,
                                modnames = CEWL_mod_names))
```

```
## Warning in aictab.AIClmerModLmerTest(cand.set = CEWL_models, modnames = CEWL_mod_names):
## Model selection for fixed effects is only appropriate with ML estimation:
## REML (default) should only be used to select random effects for a constant set of fixed effects
```

```r
# compare across models
CEWL_across <- CEWL_RMSE_Rsq %>%
  left_join(CEWL_AICc, by = 'model') %>%
  mutate(response = "deltaCEWL") %>%
  arrange(Delta_AICc)

# calculate F & p-values
```

```
CEWL_VPD_p <- data.frame(anova(CEWL_mod_VPD,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'VPD',
         predictor = rownames(.))
CEWL_hum_p <- data.frame(anova(CEWL_mod_hum,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Humidity',
         predictor = rownames(.))
CEWL_temp_p <- data.frame(anova(CEWL_mod_temp,
                              type = "1",
                              ddf = "Kenward-Roger")) %>%
  mutate(model = 'Temp',
         predictor = rownames(.))

# save within model values
CEWL_within <- CEWL_VPD_p %>%
  rbind(CEWL_hum_p) %>%
  rbind(CEWL_temp_p) %>%
  mutate(df = paste((NumDF), round(DenDF, 0), sep = ", "),
         response = "deltaCEWL")
```

Group Export

Put all the model statistics into one df/csv - one for among-model comparisons, and one for within-model stats.

```
experiment_model_compare <- CEWL_across %>%
  rbind(osml_across) %>%
  rbind(hct_across) %>%
  rbind(SMI_across) %>%
  dplyr::select(response, model,
                RMSE, Rsq, AICc, Delta_AICc) %>%
  mutate(RMSE = round(RMSE, 2),
         Rsq = round(Rsq, 2),
         AICc = round(AICc, 2),
         Delta_AICc = round(Delta_AICc, 2))
write.csv(experiment_model_compare,
          "./results_statistics/exp_model_comparisons.csv")

experiment_model_values <- CEWL_within %>%
  rbind(osml_within) %>%
  rbind(hct_within) %>%
  rbind(SMI_within) %>%
  dplyr::select(response, model, predictor,
                seq_sum_of_squares = Sum.Sq,
                df,
                F_statistic = F.value,
                p_value = Pr..F.) %>%
  mutate(seq_sum_of_squares = round(seq_sum_of_squares, 0),
         F_statistic = round(F_statistic, 2))
write.csv(experiment_model_values, "./results_statistics/exp_model_values.csv")
```

Effect Estimates

End Value CIs

Now, we can use the emmeans function from the emmeans package to estimate marginal means and confidence intervals for the values among treatment groups at the end of the experiment. But, to get for each treatment group, we need to run a new model with day 8 data only and tmt as a single, 4-category variable.

```
# Body Condition
SMI_mod_end <- lmerTest::lmer(data = end_vals,
                              SMI ~ tmt +
                              (1|trial_number))
SMI_emmeans <- data.frame(emmeans(SMI_mod_end, "tmt")) %>%
  mutate(response = "Body Condition (g')")
SMI_pairwise <- data.frame(pairs(emmeans(SMI_mod_end, "tmt"))) %>%
  mutate(response = "Body Condition (g')")

# Hematocrit
hct_mod_end <- lmerTest::lmer(data = end_vals,
                              hematocrit_percent ~ tmt +
                              (1|trial_number))
hct_emmeans <- data.frame(emmeans(hct_mod_end, "tmt")) %>%
  mutate(response = "Hematocrit (%)")
hct_pairwise <- data.frame(pairs(emmeans(hct_mod_end, "tmt"))) %>%
  mutate(response = "Hematocrit (%)")

# Plasma Osmolality
osml_mod_end <- lmerTest::lmer(data = end_vals,
                              osmolality_mmol_kg_mean ~ tmt +
                              (1|trial_number))
osml_emmeans <- data.frame(emmeans(osml_mod_end, "tmt")) %>%
  mutate(response = "Plasma Osmolality (mmol/kg)")
osml_pairwise <- data.frame(pairs(emmeans(osml_mod_end, "tmt"))) %>%
  mutate(response = "Plasma Osmolality (mmol/kg)")

# CEWL
CEWL_mod_end <- lmerTest::lmer(data = end_vals,
                              CEWL_g_m2h_mean ~ tmt +
                              (1|trial_number))
CEWL_emmeans <- data.frame(emmeans(CEWL_mod_end, "tmt")) %>%
  mutate(response = "CEWL (g/m2h)")
CEWL_pairwise <- data.frame(pairs(emmeans(CEWL_mod_end, "tmt"))) %>%
  mutate(response = "CEWL (g/m2h)")

# put together?
# use indiv for boxplots
```

Rate of Change

```
# Body Condition
SMI_mod_day <- lmerTest::lmer(data = dat_no_rehab,
                              SMI ~ day_n * tmt +
                              (1|trial_number/individual_ID))
SMI_emtrends <- data.frame(emtrends(SMI_mod_day, "tmt", var = "day_n")) %>%
```

```r
  mutate(response = "Body Condition (g')")
SMI_pairtrend <- data.frame(pairs(emtrends(SMI_mod_day, "tmt", var = "day_n"))) %>%
  mutate(response = "Body Condition (g')")

# Hematocrit
hct_mod_day <- lmerTest::lmer(data = dat_no_rehab,
                              hematocrit_percent ~ day_n * tmt +
                              (1|trial_number/individual_ID))
hct_emtrends <- data.frame(emtrends(hct_mod_day, "tmt", var = "day_n")) %>%
  mutate(response = "Hematocrit (%)")
hct_pairtrend <- data.frame(pairs(emtrends(hct_mod_day, "tmt", var = "day_n"))) %>%
  mutate(response = "Hematocrit (%)")

# Plasma Osmolality
osml_mod_day <- lmerTest::lmer(data = dat_no_rehab,
                              osmolality_mmol_kg_mean ~ day_n * tmt +
                              (1|trial_number/individual_ID))
osml_emtrends <- data.frame(emtrends(osml_mod_day, "tmt", var = "day_n")) %>%
  mutate(response = "Plasma Osmolality (mmol/kg)")
osml_pairtrend <- data.frame(pairs(emtrends(osml_mod_day, "tmt", var = "day_n"))) %>%
  mutate(response = "Plasma Osmolality (mmol/kg)")

# CEWL
CEWL_mod_day <- lmerTest::lmer(data = dat_no_rehab,
                              CEWL_g_m2h_mean ~ day_n * tmt +
                              (1|trial_number))
CEWL_emtrends <- data.frame(emtrends(CEWL_mod_day, "tmt", var = "day_n")) %>%
  mutate(response = "CEWL (g/m2h)")
CEWL_pairtrend <- data.frame(pairs(emtrends(CEWL_mod_day, "tmt", var = "day_n"))) %>%
  mutate(response = "CEWL (g/m2h)")

# put together
all_emtrends <- CEWL_emtrends %>%
  rbind(osml_emtrends) %>%
  rbind(hct_emtrends) %>%
  rbind(SMI_emtrends) %>%
  mutate(confint95 = paste(round(lower.CL, 2), round(upper.CL, 2), sep = ", ")) %>%
  dplyr::select(response, tmt,
                per_day_change = day_n.trend,
                confint95,
                SE, df)
#write.csv(all_emtrends, "./results_statistics/exp_emtrends_per_day_change.csv")
```

Given the daily trends for plasma osmolality, in total for acclimation change in osml had 1.04 (0.13x8) to 13 (1.587x8) mmol/kg of change.

CEWL ~ osmolality

Use separate df for each tmt group created for body temp relationship tests.

Then, run models for CEWL ~ osmolality:

```r
HH_CEWL_osml <- lmerTest::lmer(data = HH8,
                               CEWL_g_m2h_mean ~ osmolality_mmol_kg_mean +
```

```r
                                (1|trial_number))
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
summary(HH_CEWL_osml)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: CEWL_g_m2h_mean ~ osmolality_mmol_kg_mean + (1 | trial_number)
##    Data: HH8
##
## REML criterion at convergence: 222.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1357 -0.6398 -0.1682  0.4261  2.4440
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  trial_number (Intercept)  0.00    0.00
##  Residual                 62.57    7.91
## Number of obs: 32, groups:  trial_number, 5
##
## Fixed effects:
##                         Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)             19.77938   23.68747 30.00000   0.835    0.410
## osmolality_mmol_kg_mean  0.04658    0.06734 30.00000   0.692    0.494
##
## Correlation of Fixed Effects:
##             (Intr)
## osmllty_m__ -0.998
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
anova(HH_CEWL_osml, type = "1", ddf = "Kenward-Roger")
```

```
## Type I Analysis of Variance Table with Kenward-Roger's method
##                         Sum Sq Mean Sq NumDF  DenDF F value Pr(>F)
## osmolality_mmol_kg_mean 24.554  24.554     1 18.124  0.3924 0.5388
```

```r
HD_CEWL_osml <- lmerTest::lmer(data = HD8,
                                CEWL_g_m2h_mean ~ osmolality_mmol_kg_mean +
                                (1|trial_number))
summary(HD_CEWL_osml)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: CEWL_g_m2h_mean ~ osmolality_mmol_kg_mean + (1 | trial_number)
##    Data: HD8
##
## REML criterion at convergence: 171.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.55684 -0.59339 -0.07624  0.40035  2.96626
##
```

```
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  trial_number (Intercept)  6.99    2.644
##  Residual                 11.28    3.358
## Number of obs: 31, groups:  trial_number, 5
##
## Fixed effects:
##                           Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)                2.11489   11.95116 25.69290   0.177    0.861
## osmolality_mmol_kg_mean  0.06101    0.03291 26.24328   1.854    0.075 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## osmllty_m__ -0.994
```

```r
anova(HD_CEWL_osml, type = "1", ddf = "Kenward-Roger")
```

```
## Type I Analysis of Variance Table with Kenward-Roger's method
##                          Sum Sq Mean Sq NumDF  DenDF F value Pr(>F)
## osmolality_mmol_kg_mean 31.917  31.917     1 26.042  2.8298 0.1045
```

```r
CH_CEWL_osml <- lmerTest::lmer(data = CH8,
                               CEWL_g_m2h_mean ~ osmolality_mmol_kg_mean +
                               (1|trial_number))
summary(CH_CEWL_osml)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: CEWL_g_m2h_mean ~ osmolality_mmol_kg_mean + (1 | trial_number)
##    Data: CH8
##
## REML criterion at convergence: 192.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.2869 -0.7878 -0.1247  0.6553  2.0089
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  trial_number (Intercept)  6.866   2.620
##  Residual                 30.995   5.567
## Number of obs: 30, groups:  trial_number, 5
##
## Fixed effects:
##                           Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)               22.02423   13.70141 27.96594   1.607    0.119
## osmolality_mmol_kg_mean  0.02489    0.03901 27.97274   0.638    0.529
##
## Correlation of Fixed Effects:
##             (Intr)
## osmllty_m__ -0.993
```

```r
anova(CH_CEWL_osml, type = "1", ddf = "Kenward-Roger")
```

```
## Type I Analysis of Variance Table with Kenward-Roger's method
##                          Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## osmolality_mmol_kg_mean 11.376  11.376     1 27.97   0.367 0.5495
```

```
CD_CEWL_osml <- lmerTest::lmer(data = CD8,
                               CEWL_g_m2h_mean ~ osmolality_mmol_kg_mean +
                               (1|trial_number))
summary(CD_CEWL_osml)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: CEWL_g_m2h_mean ~ osmolality_mmol_kg_mean + (1 | trial_number)
##    Data: CD8
##
## REML criterion at convergence: 170
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8562 -0.5019  0.1381  0.4084  1.8415
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  trial_number (Intercept) 21.14    4.598
##  Residual                 11.34    3.368
## Number of obs: 30, groups:  trial_number, 5
##
## Fixed effects:
##                          Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)             -1.45589   11.16955 27.92145  -0.130   0.8972
## osmolality_mmol_kg_mean  0.07357    0.03133 27.93722   2.348   0.0262 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## osmllty_m__ -0.981
```

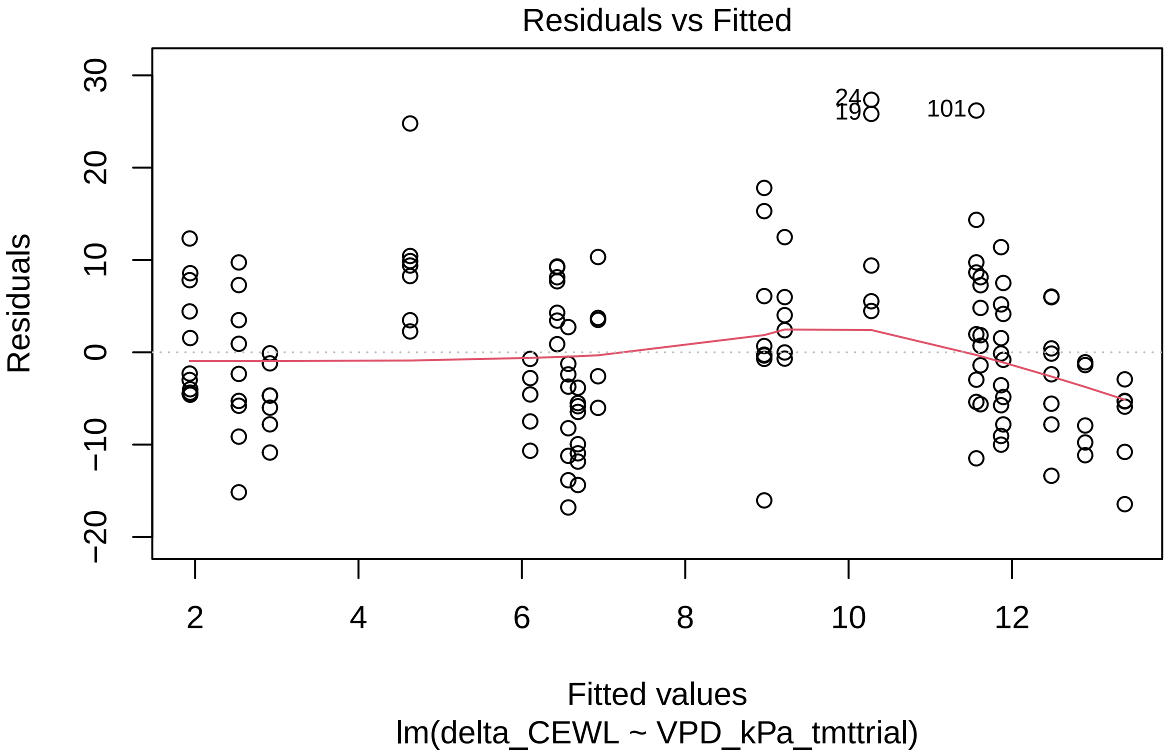
```
anova(CD_CEWL_osml, type = "1", ddf = "Kenward-Roger")
```

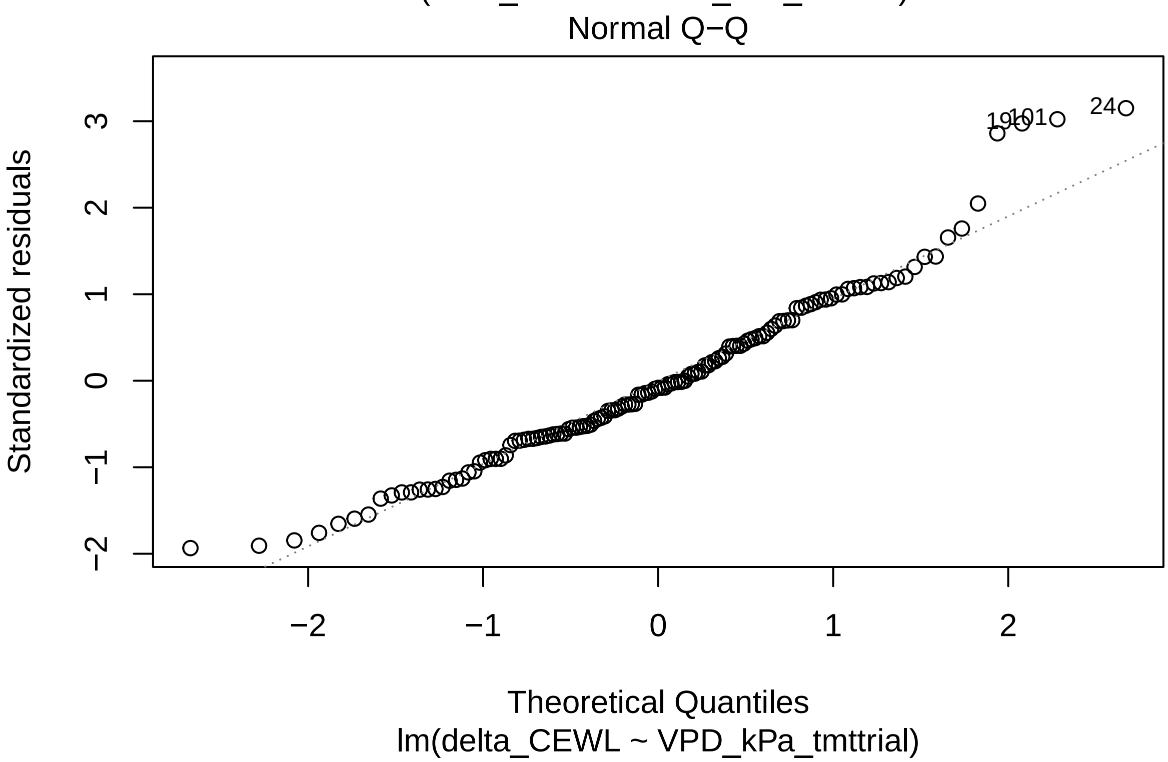
```
## Type I Analysis of Variance Table with Kenward-Roger's method
##                          Sum Sq Mean Sq NumDF  DenDF F value  Pr(>F)
## osmolality_mmol_kg_mean 55.393  55.393     1 27.937  4.8837 0.03548 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

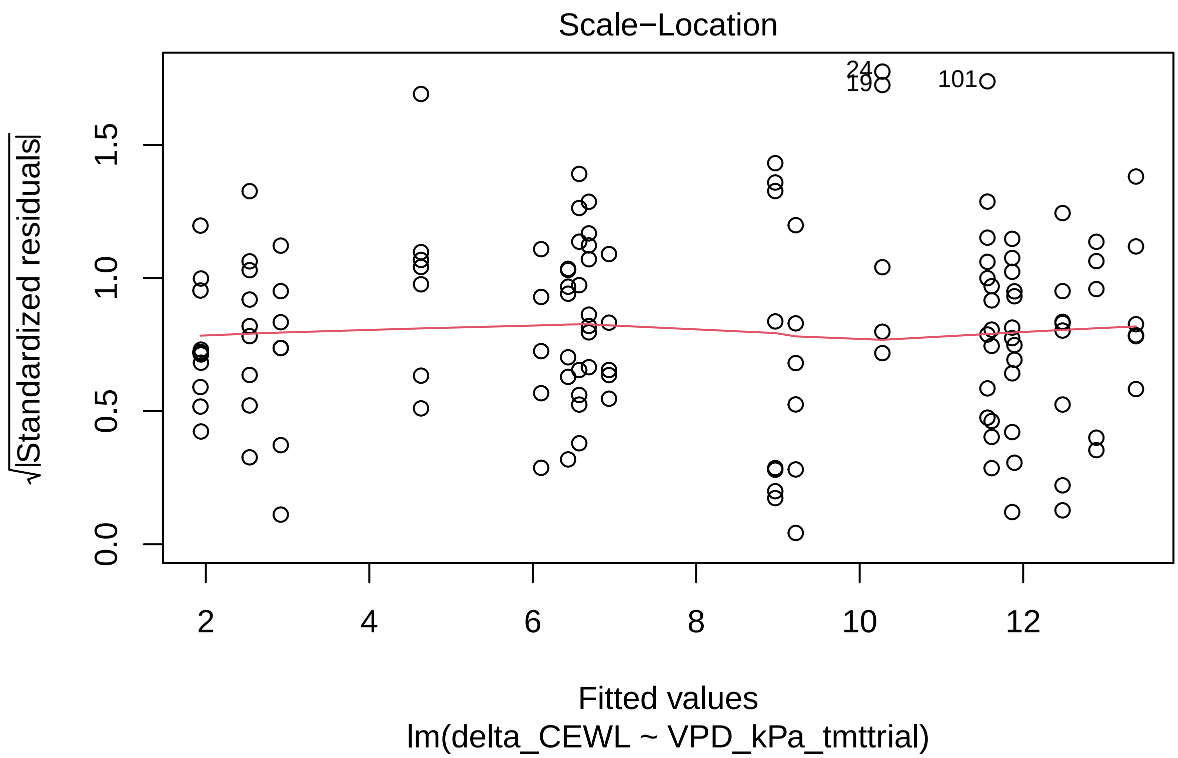
Even though only data for one treatment group was significant, all have a positive relationship

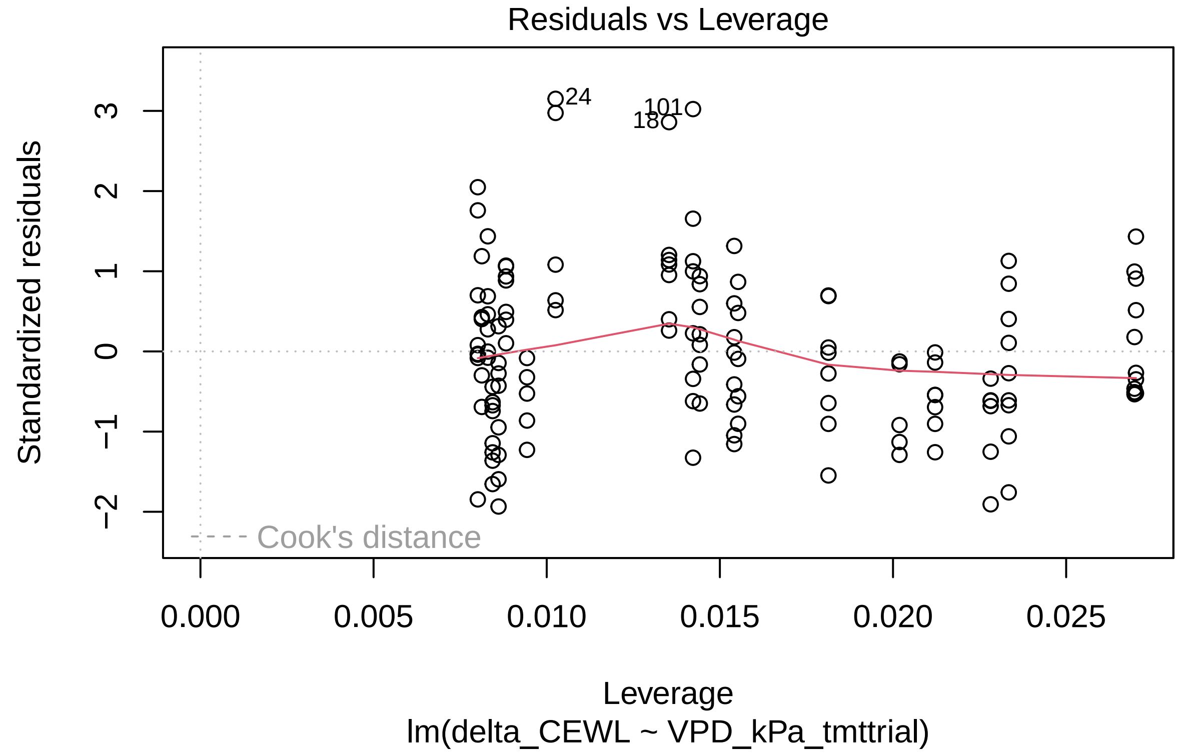
CEWL ~ VPD

```
CEWL_VPD_lm <- lm(data = dat_no_rehab_deltaCEWL,
                               delta_CEWL ~ VPD_kPa_tmttrial)
plot(CEWL_VPD_lm)
```

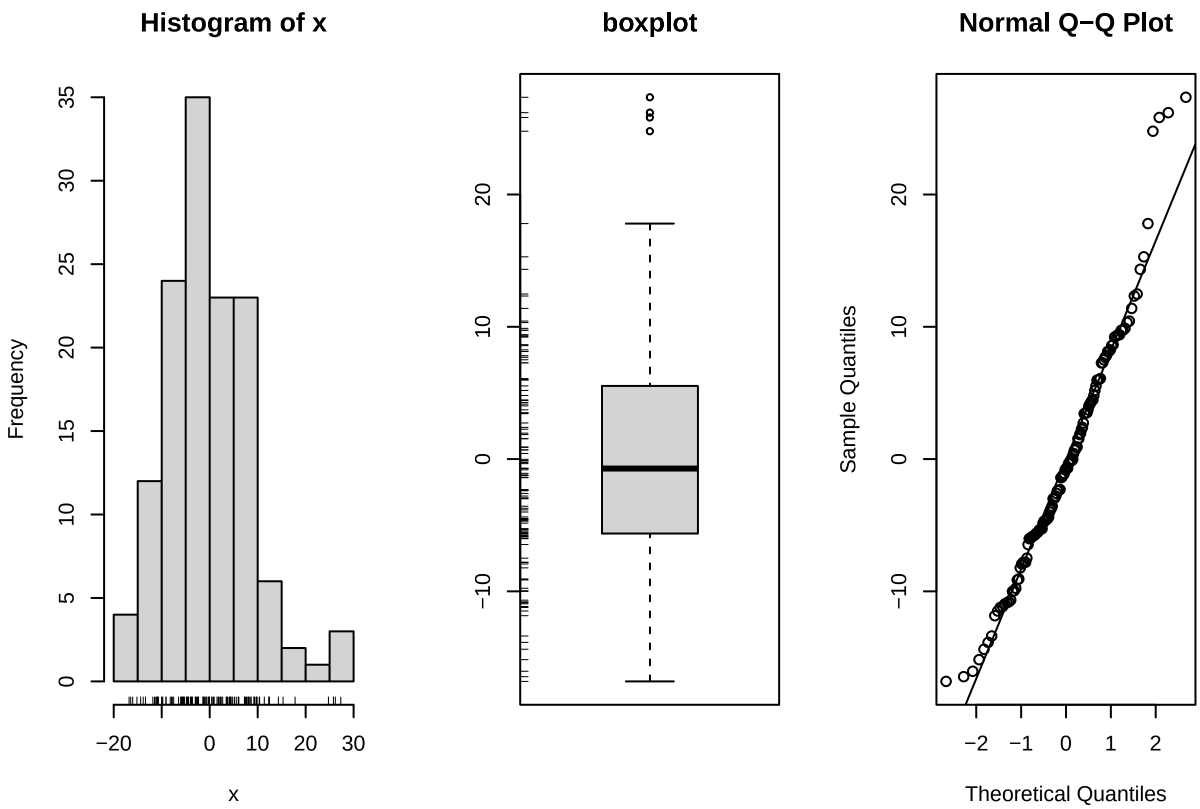

Residuals vs Fitted
Residuals
30
20
10
0
−10
−20
2
4
6
8
10
12
24
19
101
Fitted values
lm(delta_CEWL ~ VPD_kPa_tmttrial)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
−2
−1
0
1
2
19
101
24
Theoretical Quantiles
lm(delta_CEWL ~ VPD_kPa_tmttrial)






```
simple.eda(residuals(CEWL_VPD_lm))
```




```
shapiro.test(residuals(CEWL_VPD_lm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(CEWL_VPD_lm)
## W = 0.96694, p-value = 0.002527
```

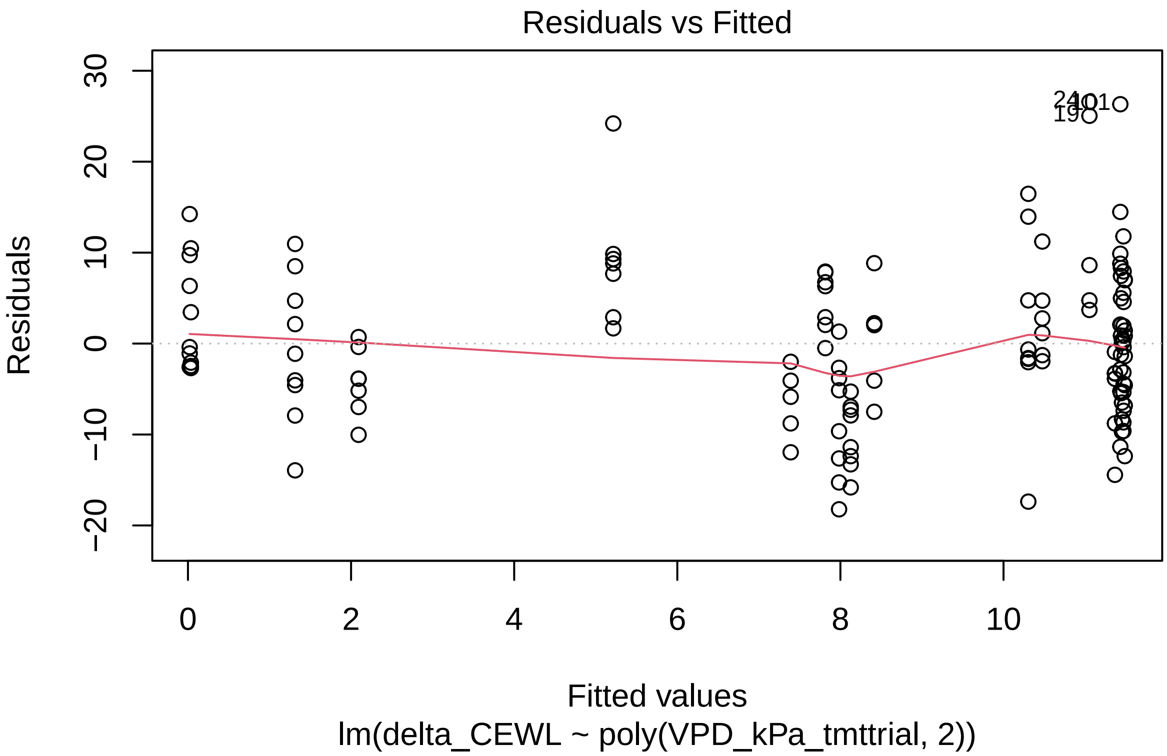
```
summary(CEWL_VPD_lm)
```

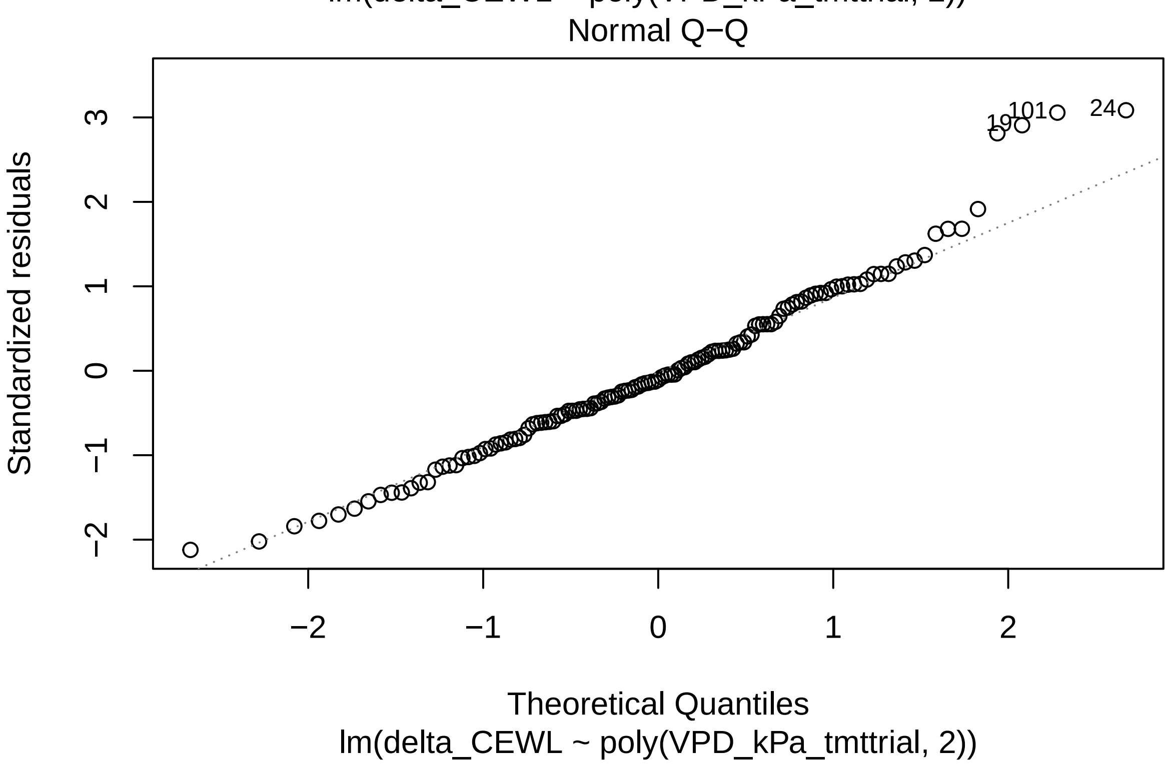
```
##
## Call:
## lm(formula = delta_CEWL ~ VPD_kPa_tmttrial, data = dat_no_rehab_deltaCEWL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.8069  -5.6289  -0.7159   5.5303  27.3517
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        13.960      1.415   9.866  < 2e-16 ***
## VPD_kPa_tmttrial   -2.959      0.594  -4.982 1.95e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.726 on 131 degrees of freedom
## Multiple R-squared:  0.1593, Adjusted R-squared:  0.1529
## F-statistic: 24.82 on 1 and 131 DF,  p-value: 1.952e-06
```

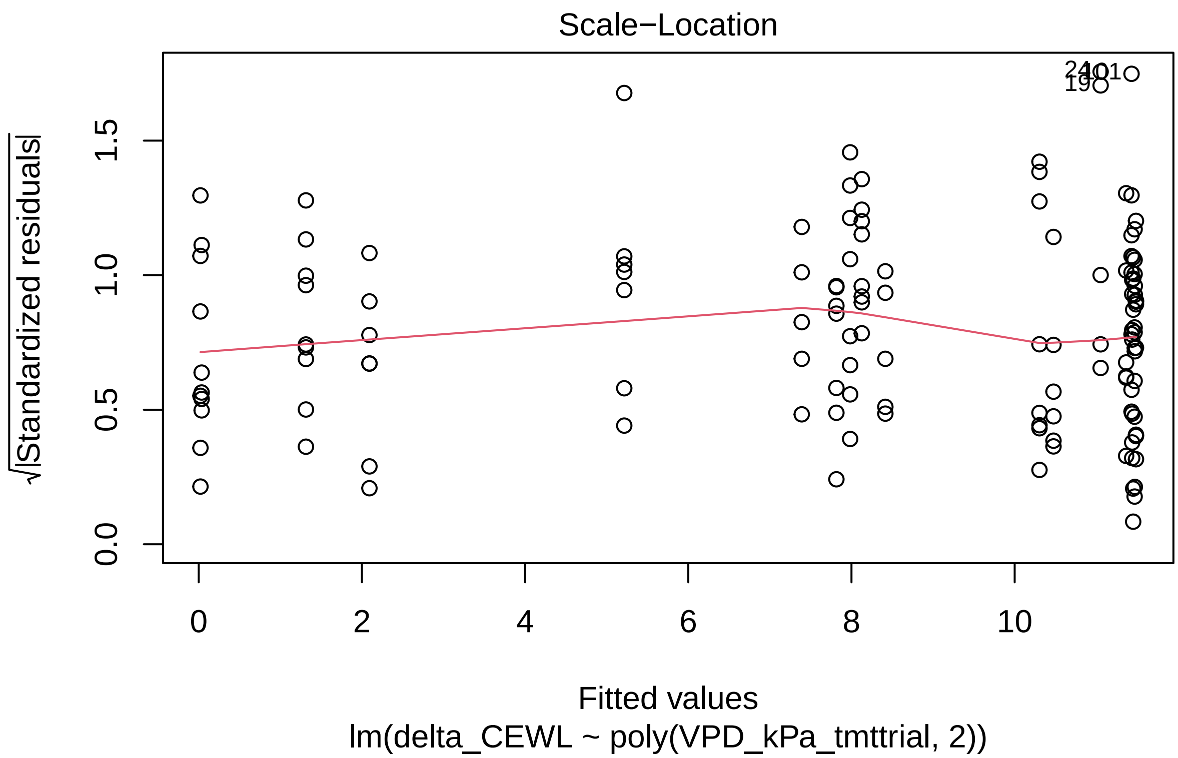
Even though the data is slightly nonlinear, a linear model does a fine job explaining the data.

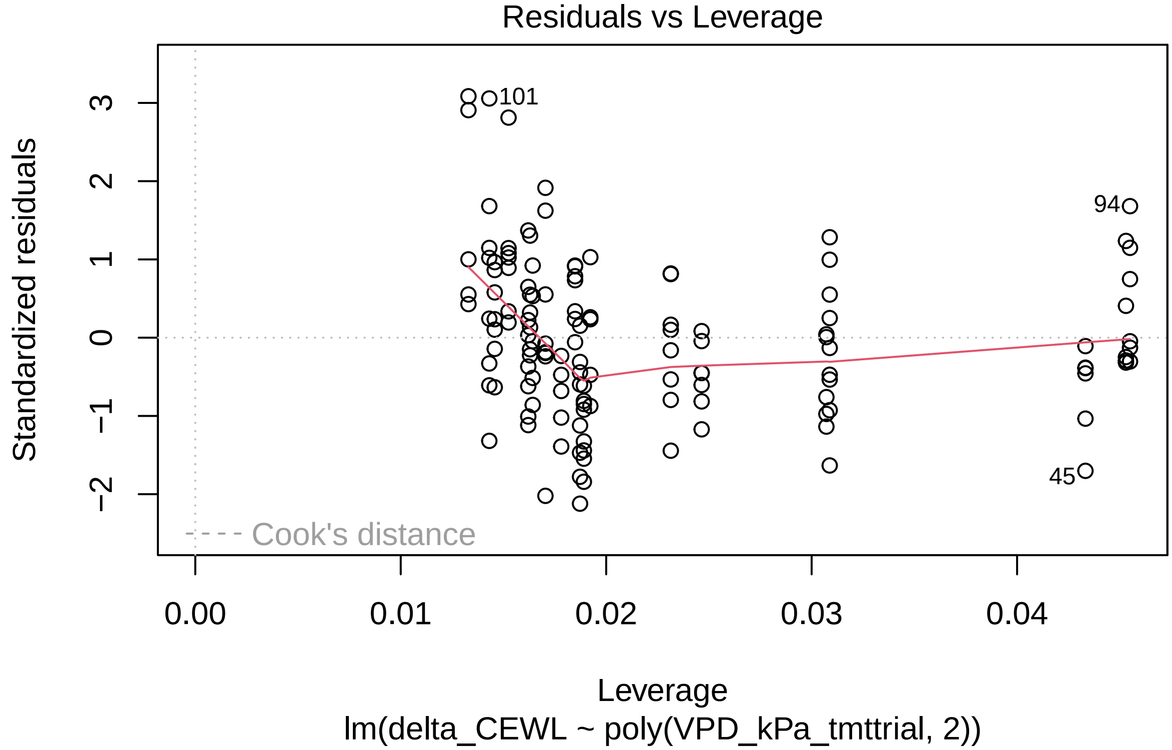
We will double check a comparison of a polynomial model, just to be sure:

```
CEWL_VPD_poly <- lm(data = dat_no_rehab_deltaCEWL,
                            delta_CEWL ~ poly(VPD_kPa_tmttrial, 2))
plot(CEWL_VPD_poly)
```

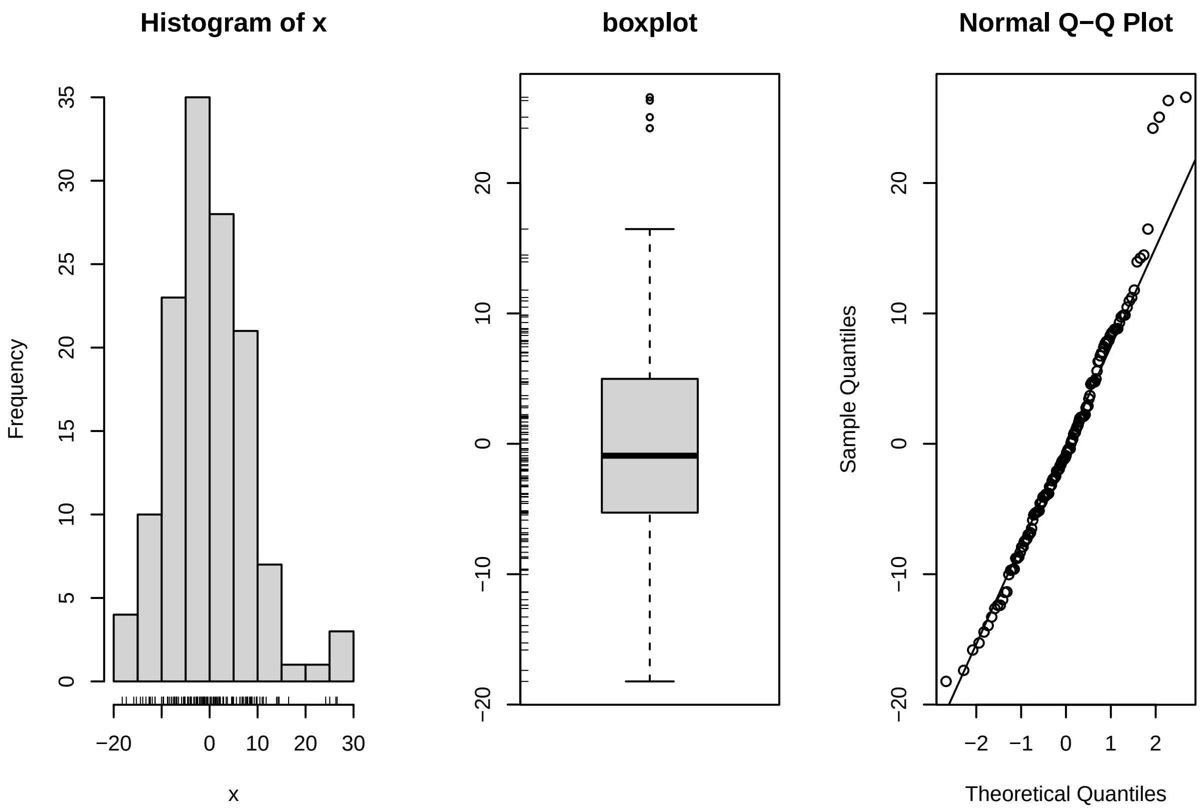









```
simple.eda(residuals(CEWL_VPD_poly))
```




```
shapiro.test(residuals(CEWL_VPD_poly))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(CEWL_VPD_poly)
## W = 0.97276, p-value = 0.00896
```

```
summary(CEWL_VPD_poly)
```

```
##
## Call:
## lm(formula = delta_CEWL ~ poly(VPD_kPa_tmttrial, 2), data = dat_no_rehab_deltaCEWL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.222  -5.281  -0.914   4.981  26.576
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   8.004      0.752  10.643  < 2e-16 ***
## poly(VPD_kPa_tmttrial, 2)1  -43.470      8.672  -5.013 1.72e-06 ***
## poly(VPD_kPa_tmttrial, 2)2  -14.069      8.672  -1.622    0.107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.672 on 130 degrees of freedom
## Multiple R-squared:  0.176,  Adjusted R-squared:  0.1633
## F-statistic: 13.88 on 2 and 130 DF,  p-value: 3.442e-06
```

LINE assumptions are equally-well-met.

The polynomial factor is not significant, but the R-sq is slightly higher for the poly model.

Compare RMSE and AIC:

```
sqrt(mean((residuals(CEWL_VPD_lm))^2))
```

```
## [1] 8.660173
```

```
sqrt(mean((residuals(CEWL_VPD_poly))^2))
```

```
## [1] 8.573822
```

```
CEWL_VPD_models <- list(CEWL_VPD_lm, CEWL_VPD_poly)
CEWL_VPD_mod_names <- data.frame(model =
                               c('linear',
                                 'polynomial'
                                 ))
CEWL_VPD_AICc <- data.frame(aictab(cand.set = CEWL_VPD_models,
                                 modnames = CEWL_VPD_mod_names))
CEWL_VPD_AICc
```

```
##        model K     AICc Delta_AICc  ModelLik    AICcWt        LL    Cum.Wt
## 2 polynomial 4 957.3080  0.0000000 1.0000000 0.5669881 -474.4977 0.5669881
## 1     linear 3 957.8471  0.5391459 0.7637056 0.4330119 -475.8305 1.0000000
```

RMSE is slightly lower for the polynomial model. But, AIC is not meaningfully different between the two versions.

I'll use the lm as the best model/

```
summary(CEWL_VPD_lm)
```

```
##
## Call:
## lm(formula = delta_CEWL ~ VPD_kPa_tmttrial, data = dat_no_rehab_deltaCEWL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.8069  -5.6289  -0.7159   5.5303  27.3517
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        13.960      1.415   9.866  < 2e-16 ***
## VPD_kPa_tmttrial   -2.959      0.594  -4.982 1.95e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.726 on 131 degrees of freedom
## Multiple R-squared:  0.1593, Adjusted R-squared:  0.1529
## F-statistic: 24.82 on 1 and 131 DF,  p-value: 1.952e-06
```

```
anova(CEWL_VPD_lm)
```

```
## Analysis of Variance Table
##
## Response: delta_CEWL
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## VPD_kPa_tmttrial   1 1889.7 1889.67  24.817 1.952e-06 ***
```

```
## Residuals        131 9974.8   76.14
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(estimate = -2.96, SE = 0.6, df = (1, 131), F = 24.82, $p < 0.0001$, Rsq = 0.16)

Recovery Models

I want to know how the 2-day recovery period affects physiology relative to post- and pre- experiment. To do this, I'll use a two-sided t-test comparing delta to the hypothesis of mu=0.

First, calculate the mean delta and SEM?

```
recovery_v_post_exp_summary <- recovery_v_post_exp %>%
  group_by(type) %>%
  summarise(mean_delta_osml = mean(delta_osml_10_8, na.rm = T),
            mean_delta_hct = mean(delta_hct_10_8, na.rm = T),
            mean_delta_SMI = mean(delta_SMI_10_8, na.rm = T))
```

SMI

```
SMI_rmod_post_exp <- t.test(recovery_v_post_exp$delta_SMI_10_8,
                            mu = 0, alternative = "two.sided")
SMI_rmod_post_exp
```

```
##
##  One Sample t-test
##
## data:  recovery_v_post_exp$delta_SMI_10_8
## t = 6.677, df = 131, p-value = 6.292e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.3171205 0.5841444
## sample estimates:
## mean of x
## 0.4506324
```

```
SMI_rmod_pre_exp <- t.test(recovery_v_pre_exp$delta_SMI_10_0,
                           mu = 0, alternative = "two.sided")
SMI_rmod_pre_exp
```

```
##
##  One Sample t-test
##
## data:  recovery_v_pre_exp$delta_SMI_10_0
## t = -11.542, df = 131, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.3426322 -0.9497445
## sample estimates:
## mean of x
## -1.146188
```

Hematocrit

```r
hct_rmod_post_exp <- t.test(recovery_v_post_exp$delta_hct_10_8,
                            mu = 0, alternative = "two.sided")
hct_rmod_post_exp
```

```
##
##  One Sample t-test
##
## data:  recovery_v_post_exp$delta_hct_10_8
## t = -4.2083, df = 126, p-value = 4.85e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -3.033119 -1.092866
## sample estimates:
## mean of x
## -2.062992
```

```r
hct_rmod_pre_exp <- t.test(recovery_v_pre_exp$delta_hct_10_0,
                            mu = 0, alternative = "two.sided")
hct_rmod_pre_exp
```

```
##
##  One Sample t-test
##
## data:  recovery_v_pre_exp$delta_hct_10_0
## t = -22.249, df = 129, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -13.50271 -11.29729
## sample estimates:
## mean of x
##     -12.4
```

Osmolality

```r
osml_rmod_post_exp <- t.test(recovery_v_post_exp$delta_osml_10_8,
                            mu = 0, alternative = "two.sided")
osml_rmod_post_exp
```

```
##
##  One Sample t-test
##
## data:  recovery_v_post_exp$delta_osml_10_8
## t = 3.4782, df = 121, p-value = 0.0007021
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   4.330203 15.772256
## sample estimates:
## mean of x
##  10.05123
```

```r
osml_rmod_pre_exp <- t.test(recovery_v_pre_exp$delta_osml_10_0,
                            mu = 0, alternative = "two.sided")
osml_rmod_pre_exp
```

```
##
##  One Sample t-test
##
## data:  recovery_v_pre_exp$delta_osml_10_0
## t = 3.75, df = 130, p-value = 0.0002651
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   5.758213 18.618378
## sample estimates:
## mean of x
##   12.1883
```

Group Export

```
recovery_stats <- broom.mixed::tidy(osml_rmod_pre_exp) %>%
  rbind(broom.mixed::tidy(osml_rmod_post_exp)) %>%
  rbind(broom.mixed::tidy(hct_rmod_pre_exp)) %>%
  rbind(broom.mixed::tidy(hct_rmod_post_exp)) %>%
  rbind(broom.mixed::tidy(SMI_rmod_pre_exp)) %>%
  rbind(broom.mixed::tidy(SMI_rmod_post_exp)) %>%
  mutate(response = c(rep("Plasma Osmolality (mmol/kg)", 2),
                      rep("Hematocrit (%)", 2),
                      rep("Body Condition (g')", 2)),
         pre_post_exp = c(rep(c("pre", "post"), 3)))
write.csv(recovery_stats,
          "./results_statistics/recovery_stats.csv")
```

Figures

Colors & Shapes

```
CH_color <- brewer.pal(4, "Spectral")[c(4)]
HH_color <- brewer.pal(4, "Spectral")[c(2)]
CD_color <- brewer.pal(4, "Spectral")[c(3)]
HD_color <- brewer.pal(4, "Spectral")[c(1)]
my_colors <- c(CH_color, HH_color, CD_color, HD_color)

CH_shp <- 15
HH_shp <- 19
CD_shp <- 22
HD_shp <- 21
CH_shp_box <- 22
HH_shp_box <- 21
my_shapes <- c(CH_shp, HH_shp, CD_shp, HD_shp)
my_shapes_box <- c(CH_shp_box, HH_shp_box, CD_shp, HD_shp)

my_labels <- c("Cool Humid\n0.6 kPa",
                               "Hot Humid\n1.1 kPa",
                               "Cool Dry\n2.5 kPa",
                               "Hot Dry\n3.8 kPa")
```

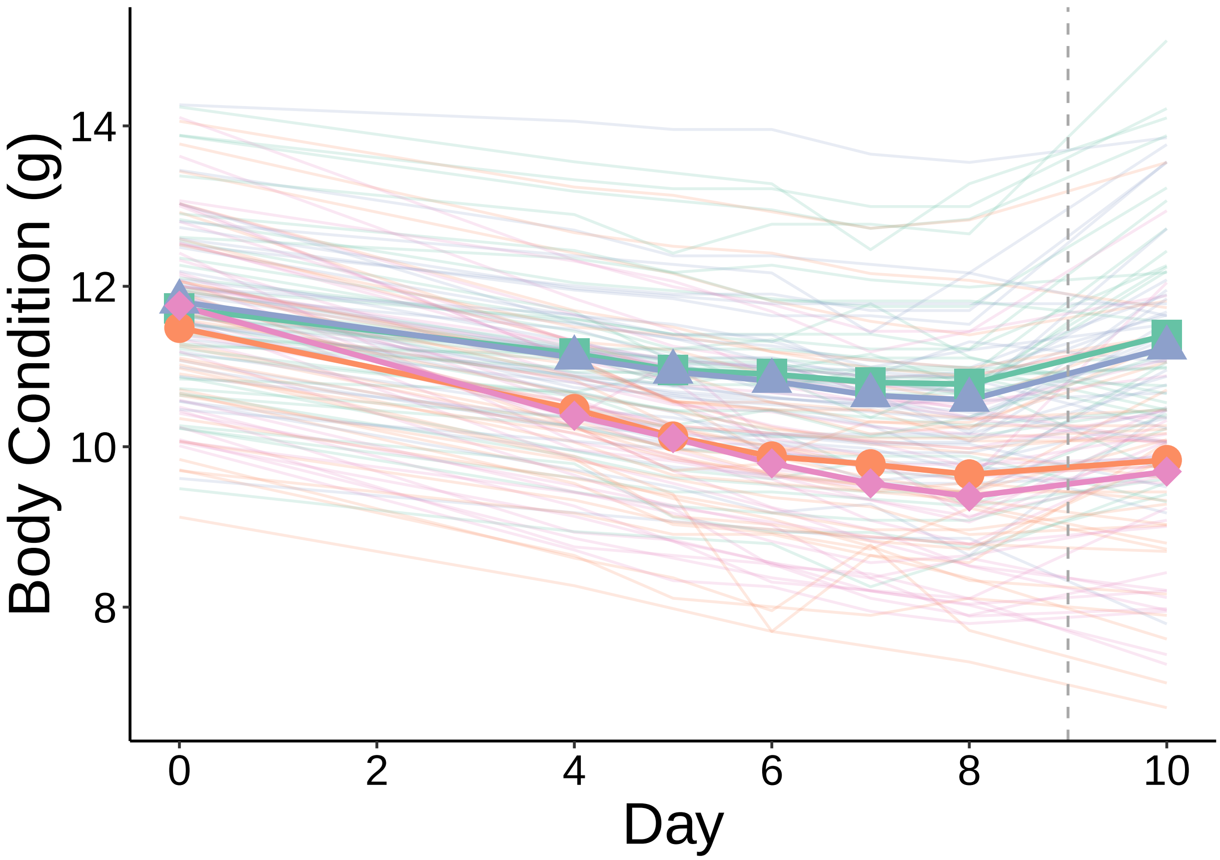
SMI

Ind + Means

```
ggplot() +
  geom_line(data = dat,
            aes(x = day_n,
                y = SMI,
                color = tmt,
                group = individual_ID),
            alpha = 0.2) +
  geom_line(data = means,
            aes(x = day_n,
                y = mean_SMI,
                color = tmt,
                group = tmt),
            alpha = 1,
            size = 1) +
  geom_point(data = means,
            aes(x = day_n,
                y = mean_SMI,
                color = tmt,
                shape = tmt),
            alpha = 1,
            size = 5) +
  geom_vline(xintercept = 9,
             linetype = "dashed",
             color = "darkgrey") +
  theme_classic() +
  scale_shape_manual(values = c(15:18), name = "") +
  scale_x_continuous(breaks = c(0, 2, 4, 6, 8, 10)) +
  scale_color_brewer(palette = "Set2", name = "") +
  xlab("Day") +
  ylab("Body Condition (g)") +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 16),
        legend.text = element_text(color = "black",
                                   family = "sans",
                                   size = 22),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = unit(c(0.1, #top
                             0.1, #right
                             0.1, #bottom
                             0.41 #left
                             ), "cm")) -> SMI_fig
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```

```
SMI_fig
```



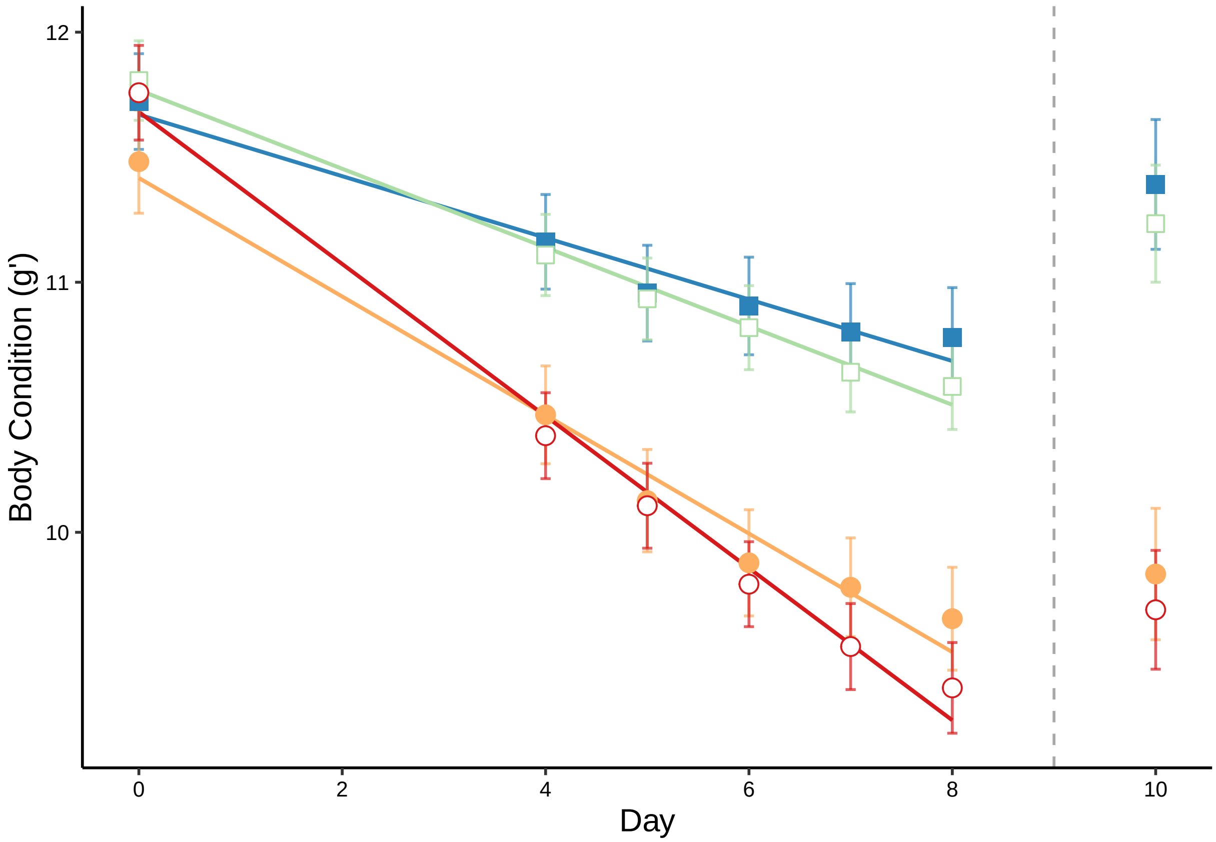

```
#ggsave(filename = "SMI_fig1.jpeg",
 #      plot = SMI_fig,
  #     path = "./results_figures",
   #    device = "jpeg",
    #   dpi = 1200,
     #  width = 6, height = 6)
```

Means Only *MS*

```
ggplot() +
  #plot these first so they end up on the "bottom"
  geom_smooth(data = dat_no_rehab,
                aes(x = day_n,
              y = SMI,
              color = tmt,
              group = tmt),
            method = "lm",
            se = F,
            size = 0.7) +
  geom_errorbar(data = means,
                aes(x = day_n,
              y = mean_SMI,
              color = tmt,
              group = tmt,
```

```r
                ymin = mean_SMI-se_SMI,
                ymax = mean_SMI+se_SMI),
                width = .1,
                #position=position_dodge(.01),
                alpha = 0.7) +
  #geom_line(data = means,
   #         aes(x = day_n,
    #            y = mean_SMI,
     #           color = tmt,
      #          #linetype = tmt,
       #         group = tmt),
        #    alpha = 1,
         #   size = 0.5) +
  geom_point(data = means,
            aes(x = day_n,
                y = mean_SMI,
                color = tmt,
                #fill = tmt,
                shape = tmt),
            fill = "white",
            alpha = 1,
            size = 3) +

  theme_classic() +
  scale_shape_manual(values = my_shapes, name = "",
                     labels = my_labels) +
  scale_fill_manual(values = my_colors, name = "",
                     labels = my_labels) +
  scale_color_manual(values = my_colors, name = "",
                     labels = my_labels) +
  scale_x_continuous(breaks = c(0, 2, 4, 6, 8, 10)) +
  scale_y_continuous(breaks = c(seq(8,12)),
                     labels = c(seq(8,12))) +
  xlab("Day") +
  ylab("Body Condition (g')") +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  geom_vline(xintercept = 9,
             linetype = "dashed",
             color = "darkgrey") +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 12),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = margin(t = 6, r = 6, b = 6, l = 10.8, unit = "pt")
) -> SMI_fig_min
SMI_fig_min
```

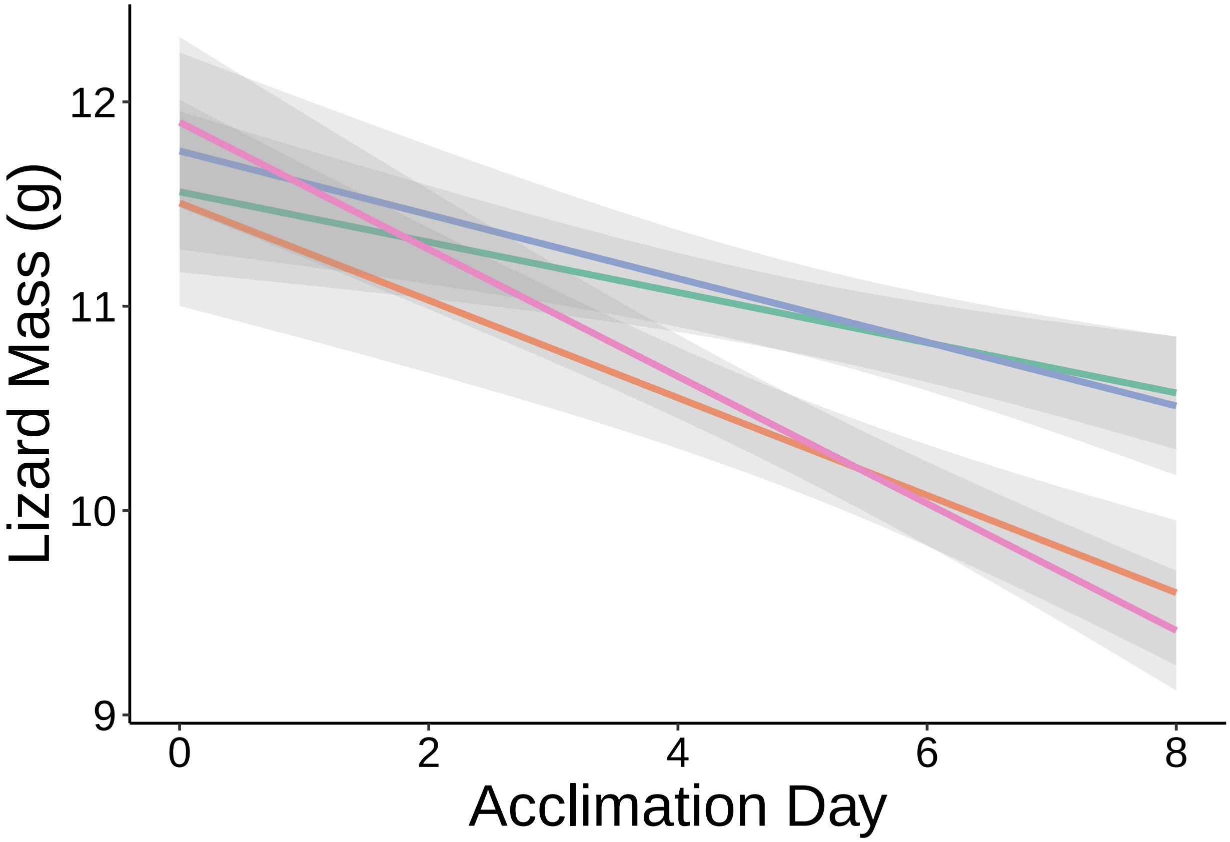
```
## `geom_smooth()` using formula = 'y ~ x'
```



LM + SE

```
ggplot() +
  geom_smooth(data = dat_no_rehab,
              aes(x = day_n,
                  y = mass_g,
                  color = tmt,
                  group = tmt),
              formula = y ~ x,
              method = "lm",
              se = T,
              size = 1.2,
              alpha = 0.2) +
  theme_classic() +
  #scale_shape_manual(values = c(15:18), name = "") +
  scale_x_continuous(limits = c(0,8),
                     breaks = c(0, 2, 4, 6, 8)) +
  scale_color_brewer(palette = "Set2", name = "") +
  xlab("Acclimation Day") +
  ylab("Lizard Mass (g)") +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
      axis.text = element_text(color = "black",
```

```
                                   family = "sans",
                                   size = 16),
        legend.text = element_text(color = "black",
                                   family = "sans",
                                   size = 22),
        legend.text.align = 0,
        legend.position = "none"
) -> SMI_lm_fig
SMI_lm_fig
```

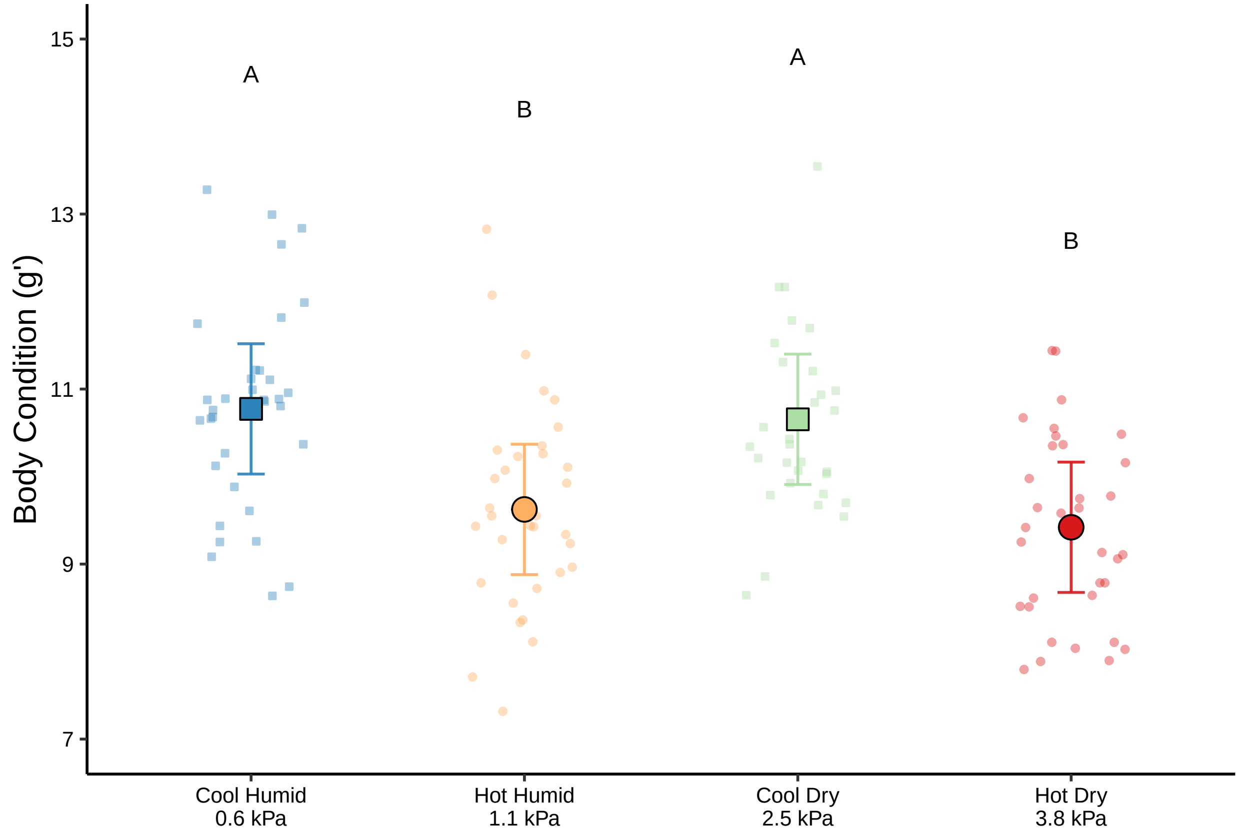


Ending Values *MS*

```
ggplot() +
  geom_jitter(data = end_vals,
              aes(x = tmt,
                   y = SMI,
                   color = tmt,
                 fill = tmt,
                 shape = tmt),
               size = 1,
              alpha = 0.4,
              position = position_jitter(height = 0, width = 0.2)) +
  geom_errorbar(data = SMI_emmeans,
                aes(x = tmt,
                    y = emmean,
                    color = tmt,
```

```
                    group = tmt,
                    ymin = lower.CL,
                    ymax = upper.CL),
                width = .1,
                alpha = 0.9) +
geom_point(data = SMI_emmeans,
           aes(x = tmt,
                 y = emmean,
                 #color = tmt,
               shape = tmt,
               fill = tmt),
           color = "black",
             size = 4) +
theme_classic() +
scale_shape_manual(values = my_shapes_box, name = "") +
scale_fill_manual(values = my_colors, name = "") +
scale_color_manual(values = my_colors, name = "") +
scale_y_continuous(limits = c(7,15),
                   breaks = c(seq(7,15, by = 2)),
                   labels = c(seq(7,15, by = 2))) +
scale_x_discrete(labels = c("Cool Humid\n0.6 kPa",
                             "Hot Humid\n1.1 kPa",
                             "Cool Dry\n2.5 kPa",
                             "Hot Dry\n3.8 kPa")) +
xlab("") +

annotate(geom = "text", x = 4, y = 12.7, label = "B",
         size = 3) +
#annotate(geom = "text", x = 4, y = 12.2, label = "9.4", #HD
 #        size = 3) +
annotate(geom = "text", x = 2, y = 14.2, label = "B",
         size = 3) +
#annotate(geom = "text", x = 2, y = 13.7, label = "9.6", #HH
 #        size = 3) +
annotate(geom = "text", x = 3, y = 14.8, label = "A",
         size = 3) +
#annotate(geom = "text", x = 3, y = 14.3, label = "10.7", #CD
 #        size = 3) +
annotate(geom = "text", x = 1, y = 14.6, label = "A",
         size = 3) +
#annotate(geom = "text", x = 1, y = 14.1, label = "10.8", #CH
 #        size = 3) +

ylab("Body Condition (g')") +
theme(text = element_text(color = "black",
                          family = "sans",
                          size = 12),
      axis.text = element_text(color = "black",
                               family = "sans",
                               size = 8),
      #axis.text.x = element_blank(),
      legend.text = element_text(color = "black",
                               family = "sans",
```

```
                                    size = 8),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = unit(c(0, #top
                             0, #right
                             0, #bottom
                             3.4), "mm")
        ) -> SMI_end_boxplot
SMI_end_boxplot
```




```
SMI_emmeans
```

```
##                      tmt    emmean        SE       df  lower.CL upper.CL
## 1 Cool Humid (0.6 kPa) 10.773306 0.3101046 6.511477 10.028724 11.51789
## 2  Hot Humid (1.1 kPa)  9.624568 0.3116093 6.634876  8.879428 10.36971
## 3   Cool Dry (2.5 kPa) 10.654259 0.3115933 6.634043  9.909138 11.39938
## 4    Hot Dry (3.8 kPa)  9.420686 0.3101705 6.516199  8.676064 10.16531
##                response
## 1 Body Condition (g')
## 2 Body Condition (g')
## 3 Body Condition (g')
## 4 Body Condition (g')
```

```
SMI_pairwise
```

```
##                                        contrast  estimate        SE       df
## 1 Cool Humid (0.6 kPa) - Hot Humid (1.1 kPa)  1.1487389 0.2384728 126.0137
## 2  Cool Humid (0.6 kPa) - Cool Dry (2.5 kPa)  0.1190474 0.2391374 126.1449
```

```
## 3   Cool Humid (0.6 kPa) - Hot Dry (3.8 kPa)  1.3526208 0.2370101 126.0772
## 4   Hot Humid (1.1 kPa) - Cool Dry (2.5 kPa) -1.0296915 0.2411556 126.1941
## 5    Hot Humid (1.1 kPa) - Hot Dry (3.8 kPa)  0.2038818 0.2390104 126.1219
## 6     Cool Dry (2.5 kPa) - Hot Dry (3.8 kPa)  1.2335733 0.2385858 126.0337
##      t.ratio      p.value            response
## 1  4.8170640 2.420102e-05 Body Condition (g')
## 2  0.4978201 9.594278e-01 Body Condition (g')
## 3  5.7070166 4.643724e-07 Body Condition (g')
## 4 -4.2698224 2.211875e-04 Body Condition (g')
## 5  0.8530250 8.288557e-01 Body Condition (g')
## 6  5.1703556 5.287879e-06 Body Condition (g')
```

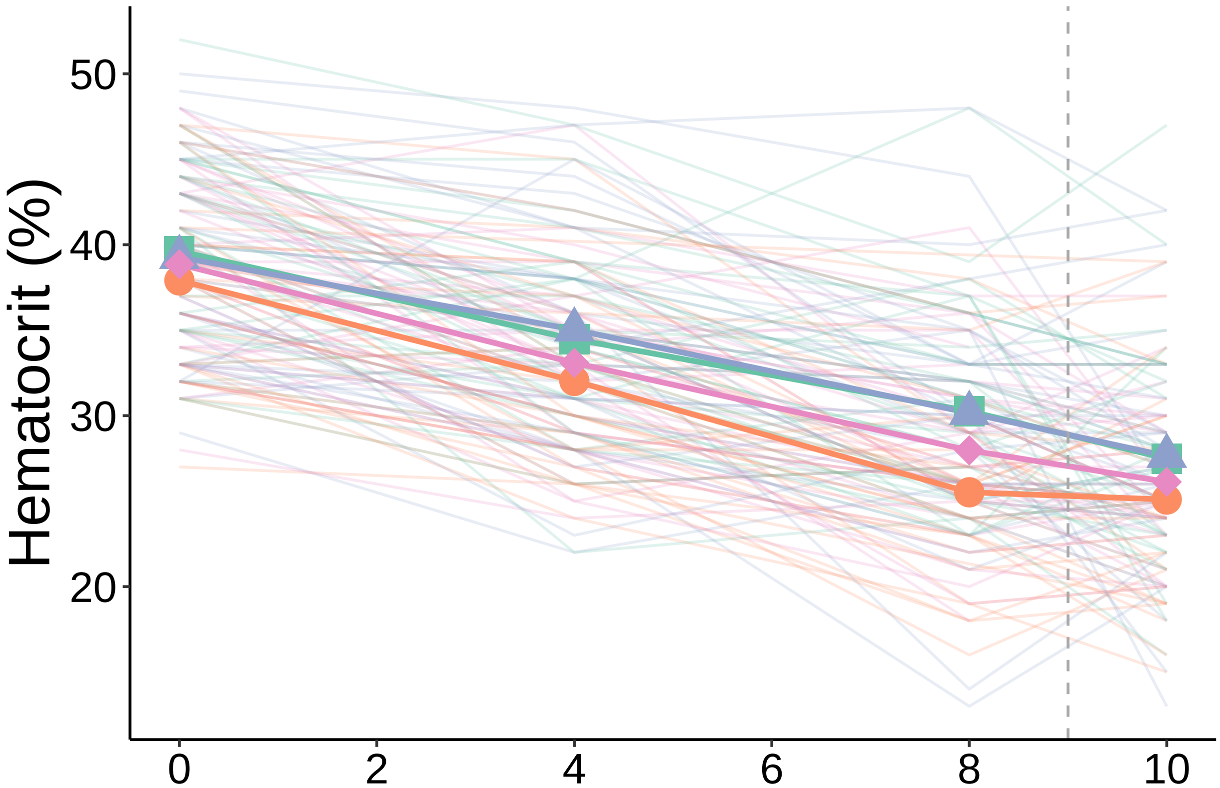
Hct

Ind + Means

```
ggplot() +
  geom_line(data = dat[complete.cases(dat$hematocrit_percent),],
            aes(x = day_n,
                y = hematocrit_percent,
                color = tmt,
                group = individual_ID),
            alpha = 0.2) +
  geom_line(data = means[complete.cases(means$mean_hct),],
            aes(x = day_n,
                y = mean_hct,
                color = tmt,
                group = tmt),
            alpha = 1,
            size = 1) +
  geom_point(data = means,
            aes(x = day_n,
                y = mean_hct,
                color = tmt,
                shape = tmt),
            alpha = 1,
            size = 5) +
  geom_vline(xintercept = 9,
             linetype = "dashed",
             color = "darkgrey") +
  theme_classic() +
  scale_shape_manual(values = c(15:18), name = "") +
  scale_x_continuous(breaks = c(0, 2, 4, 6, 8, 10)) +
  scale_color_brewer(palette = "Set2", name = "") +
  xlab("") +
  ylab("Hematocrit (%)") +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 16),
        legend.text = element_text(color = "black",
```

```
                                 family = "sans",
                                 size = 22),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = unit(c(0.1, #top
                             0.1, #right
                             0.1, #bottom
                             0.41 #left
                             ), "cm")
        ) -> hct_fig
hct_fig
```

```
## Warning: Removed 12 rows containing missing values (`geom_point()`).
```



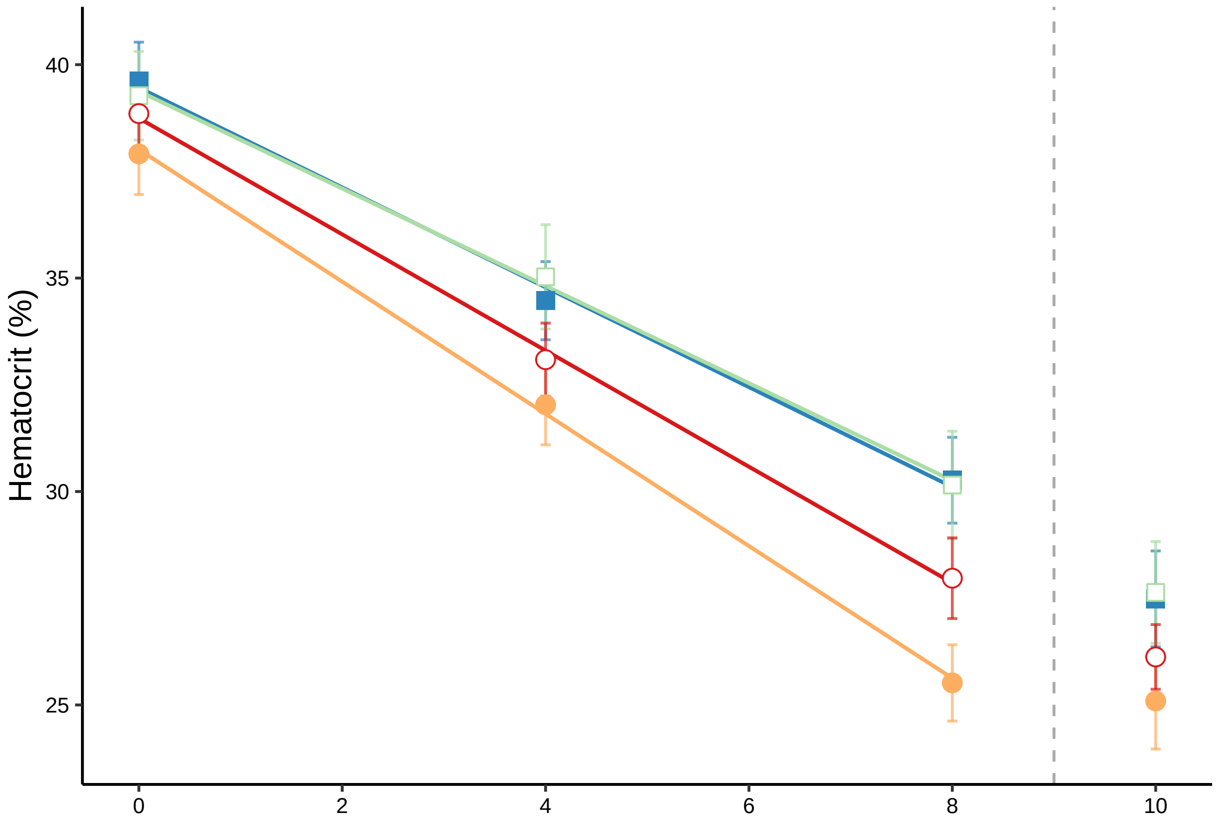
Means Only ***MS***

```
ggplot() +
  geom_smooth(data = dat_no_rehab[complete.cases(dat_no_rehab$hematocrit_percent),],
                aes(x = day_n,
              y = hematocrit_percent,
              color = tmt,
              group = tmt),
            method = "lm",
            se = F,
            size = 0.7) +
  geom_errorbar(data = means[complete.cases(means$mean_hct),],
                aes(x = day_n,
              y = mean_hct,
```

```
                color = tmt,
                group = tmt,
                ymin = mean_hct-se_hct,
                ymax = mean_hct+se_hct),
                width = .1,
                #position=position_dodge(.1),
                alpha = 0.7) +
#geom_line(data = means[complete.cases(means$mean_hct),],
 #         aes(x = day_n,
  #            y = mean_hct,
   #           color = tmt,
    #          #linetype = tmt,
     #         group = tmt),
      #    alpha = 1,
       #   size = 0.5) +
geom_point(data = means[complete.cases(means$mean_hct),],
          aes(x = day_n,
              y = mean_hct,
              color = tmt,
              #fill = tmt,
              shape = tmt),
          alpha = 1,
          fill = "white",
          size = 3) +
theme_classic() +
scale_shape_manual(values = my_shapes, name = "",
                   labels = my_labels) +
scale_fill_manual(values = my_colors, name = "",
                   labels = my_labels) +
scale_color_manual(values = my_colors, name = "",
                   labels = my_labels) +
scale_x_continuous(breaks = c(0, 2, 4, 6, 8, 10)) +
scale_y_continuous(breaks = c(25, 30, 35, 40),
                   labels = c(25, 30, 35, 40),) +
xlab("") +
ylab("Hematocrit (%)") +
guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
geom_vline(xintercept = 9,
           linetype = "dashed",
           color = "darkgrey") +
theme(text = element_text(color = "black",
                          family = "sans",
                          size = 12),
      axis.text = element_text(color = "black",
                               family = "sans",
                               size = 8),
      legend.text = element_text(color = "black",
                               family = "sans",
                               size = 8),
      legend.text.align = 0,
      legend.position = "none",
      plot.margin = margin(t = 6, r = 6, b = 0, l = 10.8, unit = "pt")
      ) -> hct_fig_min
```

```
hct_fig_min
```

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

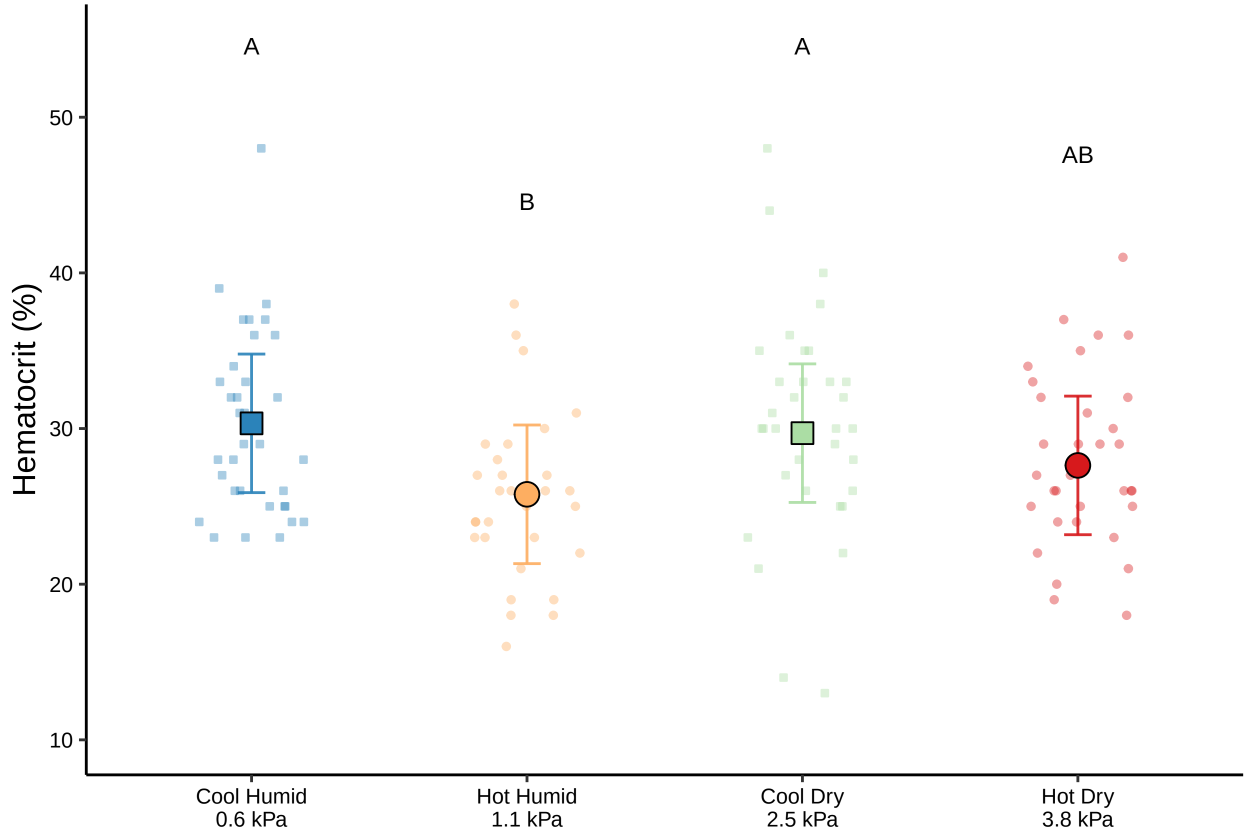


Ending Values *MS*

```
ggplot() +
  geom_jitter(data = end_vals,
              aes(x = tmt,
                   y = hematocrit_percent,
                   color = tmt,
                 fill = tmt,
                 shape = tmt),
               size = 1,
              alpha = 0.4,
              position = position_jitter(height = 0, width = 0.2)) +
  geom_errorbar(data = hct_emmeans,
                aes(x = tmt,
                    y = emmean,
                    color = tmt,
                    group = tmt,
                    ymin = lower.CL,
                    ymax = upper.CL),
                width = .1,
                alpha = 0.9) +
  geom_point(data = hct_emmeans,
             aes(x = tmt,
```

```r
                    y = emmean,
                    #color = tmt,
                  shape = tmt,
                  fill = tmt),
              color = "black",
                size = 4) +
  theme_classic() +
  scale_shape_manual(values = my_shapes_box, name = "") +
  scale_fill_manual(values = my_colors, name = "") +
  scale_color_manual(values = my_colors, name = "") +
  xlab("") +
  scale_y_continuous(limits = c(10,55),
                     breaks = c(seq(10,50, by = 10)),
                     labels = c(seq(10,50, by = 10))) +
  scale_x_discrete(labels = c("Cool Humid\n0.6 kPa",
                              "Hot Humid\n1.1 kPa",
                              "Cool Dry\n2.5 kPa",
                              "Hot Dry\n3.8 kPa")) +

  annotate(geom = "text", x = 4, y = 47.6, label = "AB",
           size = 3) +
  #annotate(geom = "text", x = 4, y = 45, label = "28", #HD
   #        size = 3) +
  annotate(geom = "text", x = 2, y = 44.6, label = "B",
           size = 3) +
  #annotate(geom = "text", x = 2, y = 42, label = "26", #HH
   #        size = 3) +
  annotate(geom = "text", x = 3, y = 54.6, label = "A",
           size = 3) +
  #annotate(geom = "text", x = 3, y = 52, label = "30", #CD
   #        size = 3) +
  annotate(geom = "text", x = 1, y = 54.6, label = "A",
           size = 3) +
  #annotate(geom = "text", x = 1, y = 52, label = "30", #CH
   #        size = 3) +

  ylab("Hematocrit (%)") +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 12),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.text = element_text(color = "black",
                                   family = "sans",
                                   size = 8),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = unit(c(0, #top
                             0, #right
                             0, #bottom
                             2.24), "mm")
        ) -> hct_end_boxplot
hct_end_boxplot
```

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




```
hct_emmeans
```

```
##                     tmt   emmean       SE       df lower.CL upper.CL
## 1 Cool Humid (0.6 kPa) 30.33352 1.808392 5.860156 25.88283 34.78422
## 2  Hot Humid (1.1 kPa) 25.77326 1.830146 6.142261 21.32008 30.22645
## 3   Cool Dry (2.5 kPa) 29.70335 1.815476 5.950456 25.25206 34.15464
## 4    Hot Dry (3.8 kPa) 27.63092 1.815279 5.948653 23.17978 32.08205
##         response
## 1 Hematocrit (%)
## 2 Hematocrit (%)
## 3 Hematocrit (%)
## 4 Hematocrit (%)
```

```
hct_pairwise
```

```
##                                      contrast   estimate       SE       df
## 1 Cool Humid (0.6 kPa) - Hot Humid (1.1 kPa)  4.5602630 1.272328 123.0398
## 2  Cool Humid (0.6 kPa) - Cool Dry (2.5 kPa)  0.6301769 1.254531 123.1123
## 3   Cool Humid (0.6 kPa) - Hot Dry (3.8 kPa)  2.7026074 1.253006 123.0665
## 4   Hot Humid (1.1 kPa) - Cool Dry (2.5 kPa) -3.9300861 1.288296 123.2298
## 5    Hot Humid (1.1 kPa) - Hot Dry (3.8 kPa) -1.8576556 1.283706 123.1026
## 6     Cool Dry (2.5 kPa) - Hot Dry (3.8 kPa)  2.0724305 1.262089 123.0607
##     t.ratio     p.value       response
## 1  3.5841894 0.002703439 Hematocrit (%)
## 2  0.5023206 0.958383983 Hematocrit (%)
```

```
## 3  2.1568990 0.141276348 Hematocrit (%)
## 4 -3.0506087 0.014665591 Hematocrit (%)
## 5 -1.4471033 0.472669199 Hematocrit (%)
## 6  1.6420637 0.359065321 Hematocrit (%)
```

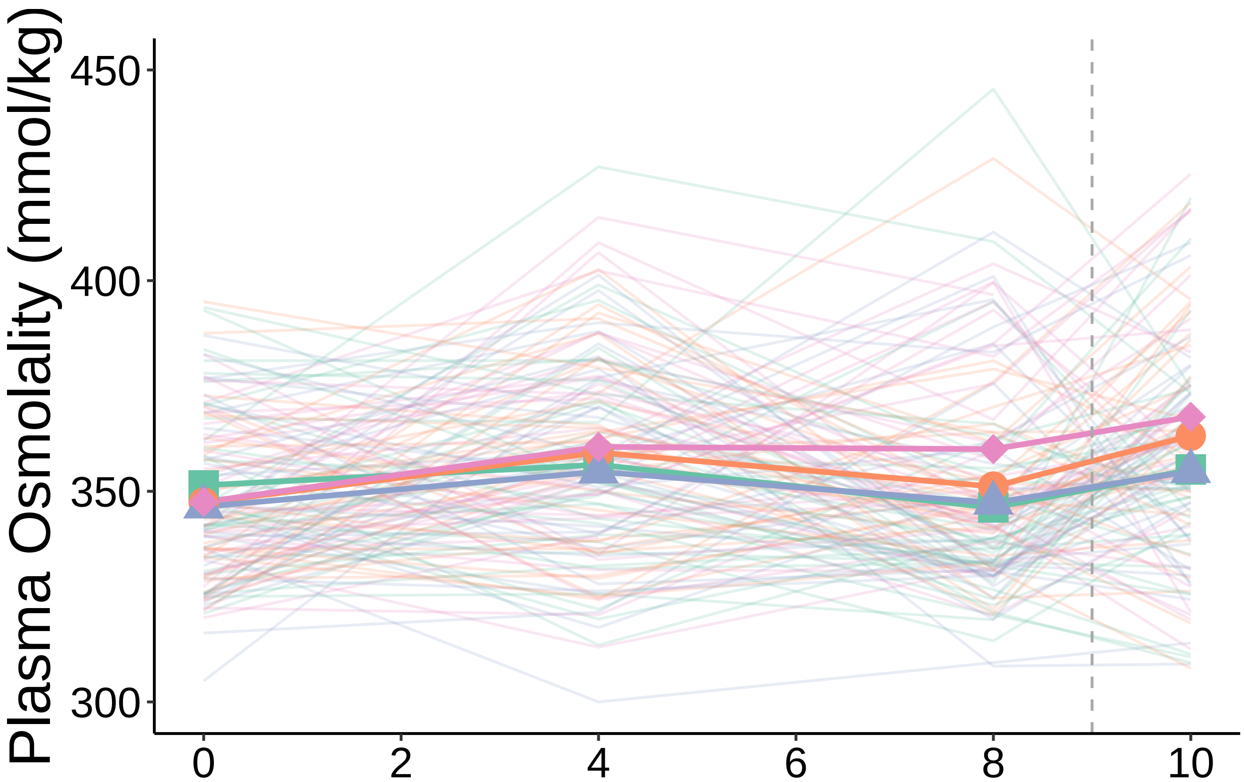
Osml

Ind + Means

```
ggplot() +
  geom_line(data = dat[complete.cases(dat$osmolality_mmol_kg_mean),],
            aes(x = day_n,
                y = osmolality_mmol_kg_mean,
                color = tmt,
                group = individual_ID),
            alpha = 0.2) +
  geom_line(data = means[complete.cases(means$mean_osml),],
            aes(x = day_n,
                y = mean_osml,
                color = tmt,
                group = tmt),
            alpha = 1,
            size = 1) +
  geom_point(data = means,
            aes(x = day_n,
                y = mean_osml,
                color = tmt,
                shape = tmt),
            alpha = 1,
            size = 5) +
  geom_vline(xintercept = 9,
             linetype = "dashed",
             color = "darkgrey") +
  scale_shape_manual(values = c(15:18), name = "") +
  scale_x_continuous(breaks = c(0, 2, 4, 6, 8, 10)) +
  ylim(300,450) +
  scale_color_brewer(palette = "Set2", name = "") +
  xlab("") +
  ylab("Plasma Osmolality (mmol/kg)") +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme_classic() +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 16),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 22),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = unit(c(0.6, #top
                            0.1, #right
```

```
                              0.1, #bottom
                              0.1 #left
                              ), "cm")
        ) -> osml_fig
osml_fig
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 12 rows containing missing values (`geom_point()`).
```



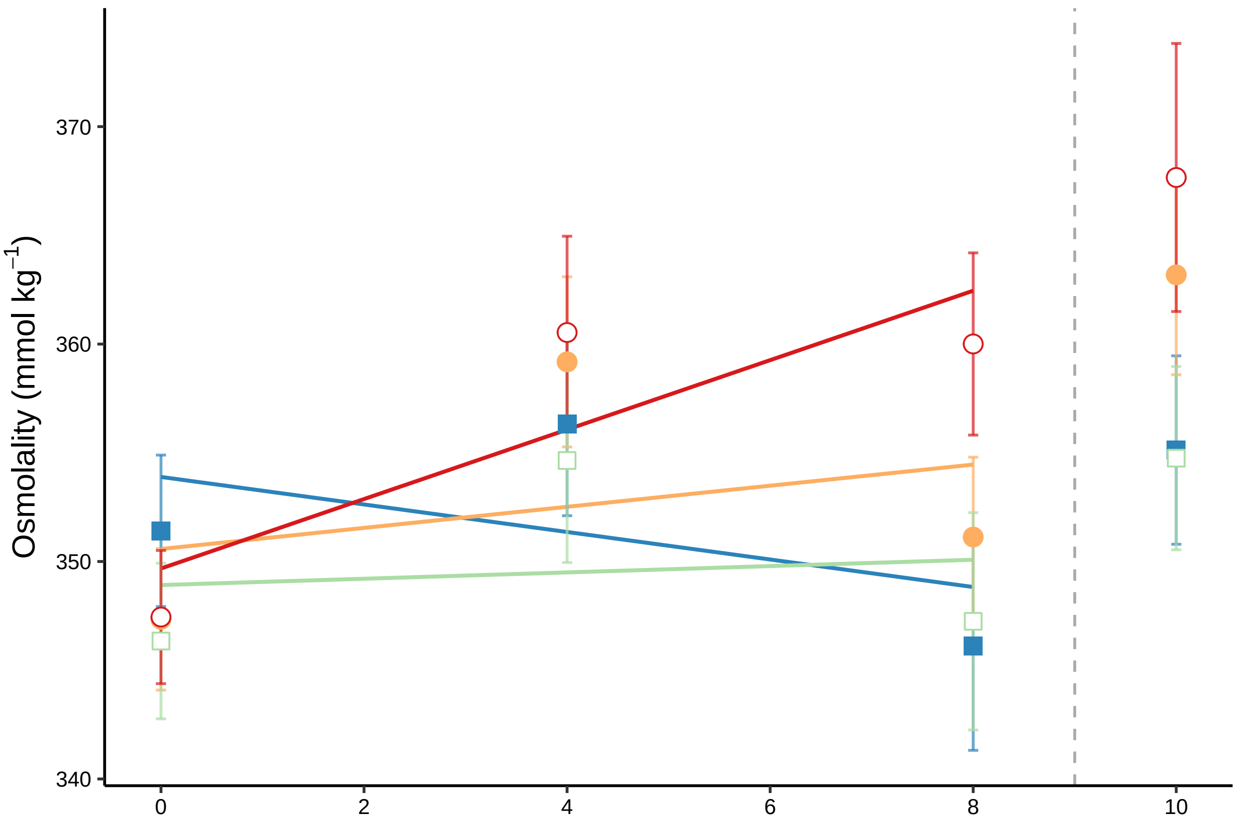
Means Only *MS*

```
ggplot() +
  geom_smooth(data = dat_no_rehab[complete.cases(dat_no_rehab$osmolality_mmol_kg_mean),],
                aes(x = day_n,
              y = osmolality_mmol_kg_mean,
              color = tmt,
              group = tmt),
            method = "lm",
            se = F,
            size = 0.7) +
  geom_errorbar(data = means,
                aes(x = day_n,
              y = mean_osml,
              color = tmt,
              group = tmt,
              ymin = mean_osml-se_osml,
              ymax = mean_osml+se_osml),
              width = .1,
```

```
               #position=position_dodge(.1),
               alpha = 0.7) +
  #geom_line(data = means[complete.cases(means$mean_osml),],
   #         aes(x = day_n,
    #            y = mean_osml,
     #           color = tmt,
      #          #linetype = tmt,
       #         group = tmt),
        #    alpha = 1,
         #   size = 0.5) +
  geom_point(data = means,
            aes(x = day_n,
                y = mean_osml,
                color = tmt,
                #fill = tmt,
                shape = tmt),
            fill = "white",
            alpha = 1,
            size = 3) +
  theme_classic() +
  scale_shape_manual(values = my_shapes, name = "",
                     labels = my_labels) +
  scale_fill_manual(values = my_colors, name = "",
                     labels = my_labels) +
  scale_color_manual(values = my_colors, name = "",
                     labels = my_labels) +
  scale_x_continuous(breaks = c(0, 2, 4, 6, 8, 10)) +
  scale_y_continuous(breaks = c(seq(320,400, by = 10)),
                     labels = c(seq(320,400, by = 10))) +
  xlab("") +
  ylab(bquote('Osmolality (mmol '*kg^-1*')')) +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  geom_vline(xintercept = 9,
             linetype = "dashed",
             color = "darkgrey") +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 12),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = margin(t = 6, r = 6, b = 0, l = 1, unit = "pt")
        ) -> osml_fig_min
osml_fig_min
```

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

```
## Warning: Removed 12 rows containing missing values (`geom_point()`).
```



Stats! Check Pairwise Diffs ~ Time

Since Plasma osmolality has such a nonlinear trend, we need to test whether the elevated values in the middle of the experiment are significantly different than the values taken before and/or after.

```
# first make sub-dfs for each tmt group
HH <- dat_no_rehab %>%
  dplyr::filter(substr(tmt, 1, 6) == "Hot Hu")
HD <- dat_no_rehab %>%
  dplyr::filter(substr(tmt, 1, 6) == "Hot Dr")
CH <- dat_no_rehab %>%
  dplyr::filter(substr(tmt, 1, 6) == "Cool H")
CD <- dat_no_rehab %>%
  dplyr::filter(substr(tmt, 1, 6) == "Cool D")

# next do pairwise tests for osml on the diff exp days, for each tmt group
pair_HH <- TukeyHSD(aov(data = HH, osmolality_mmol_kg_mean ~ day_factor)) #nonsig
pair_HD <- TukeyHSD(aov(data = HD, osmolality_mmol_kg_mean ~ day_factor)) #nonsig
pair_CH <- TukeyHSD(aov(data = CH, osmolality_mmol_kg_mean ~ day_factor)) #nonsig
pair_CD <- TukeyHSD(aov(data = CD, osmolality_mmol_kg_mean ~ day_factor)) #nonsig

# put into a df and export
osml_pairwise_df <- as.data.frame(pair_HD[[1]]) %>%
  rbind(as.data.frame(pair_HH[[1]])) %>%
  rbind(as.data.frame(pair_CD[[1]])) %>%
  rbind(as.data.frame(pair_CH[[1]])) %>%
  mutate(day_diff = paste("day", substr(rownames(.), 1, 3)),
         tmt = c(rep("Hot Dry",3),
```

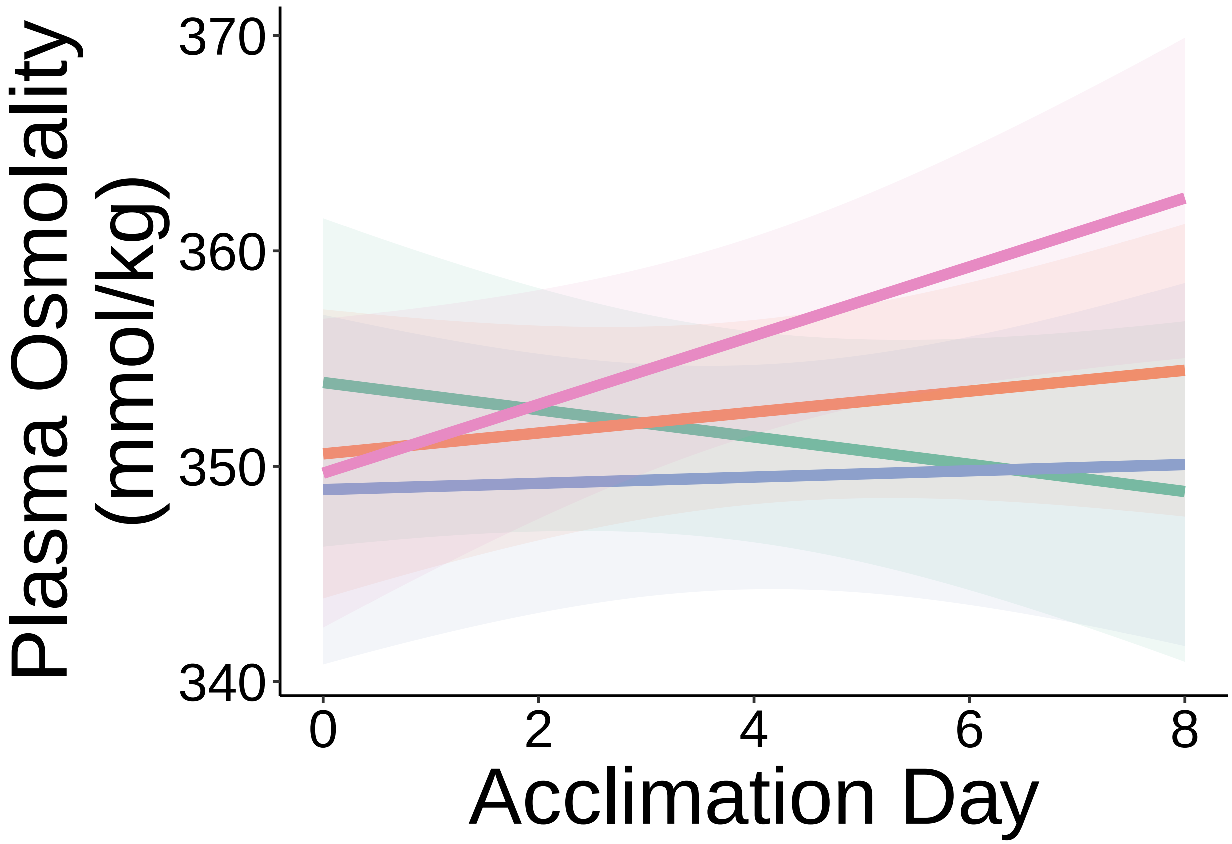
```r
                 rep("Hot Humid",3),
                 rep("Cool Dry",3),
                 rep("Cool Humid",3)),
         CI_95 = paste(round(lwr, digits = 2), round(upr, digits = 2), sep = ", "),
         diff = round(diff, digits = 2)) %>%
  dplyr::select(tmt, day_diff, diff, CI_95, p_adj = "p adj")
write.csv(osml_pairwise_df, "./results_statistics/osmolality_pairwise_diffs.csv")
```

nope, none of the differences between days within tmt groups are significantly different

LM + SE

```r
ggplot() +
  stat_smooth(data = dat_no_rehab,
              aes(x = day_n,
                  y = osmolality_mmol_kg_mean,
                  color = tmt,
                  fill = tmt,
                  group = tmt),
              formula = y ~ x,
              method = "lm",
              se = T,
              size = 2,
              alpha = 0.1) +
  theme_classic() +
  #scale_shape_manual(values = c(15:18), name = "") +
  scale_x_continuous(limits = c(0,8),
                     breaks = c(0, 2, 4, 6, 8)) +
  scale_color_brewer(palette = "Set2", name = "") +
  scale_fill_brewer(palette = "Set2", name = "") +
  xlab("Acclimation Day") +
  #ylim(340,370) + #CANNOT put ylimits in bc gets rid of pts that are used to make lines
  ylab("Plasma Osmolality\n(mmol/kg)") +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 30),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 20),
        legend.text = element_text(color = "black",
                                   family = "sans",
                                   size = 22),
        legend.text.align = 0,
        legend.position = "none"
        ) -> osml_lm_fig
osml_lm_fig
```

```
## Warning: Removed 413 rows containing non-finite values (`stat_smooth()`).
```

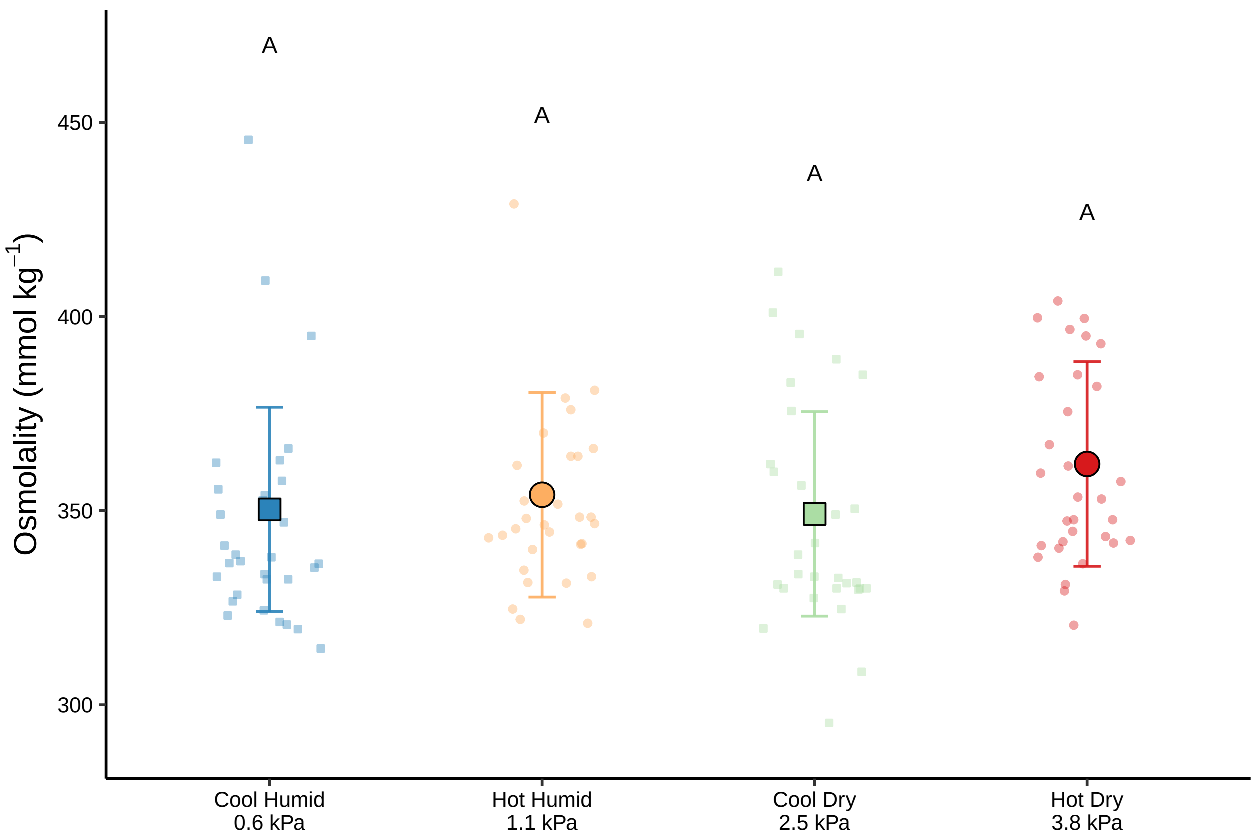


Ending Values ***MS***

```
ggplot() +
  geom_jitter(data = end_vals,
              aes(x = tmt,
                   y = osmolality_mmol_kg_mean,
                   color = tmt,
                 fill = tmt,
                 shape = tmt),
               size = 1,
              alpha = 0.4,
              position = position_jitter(height = 0, width = 0.2)) +
  geom_errorbar(data = osml_emmeans,
                aes(x = tmt,
                    y = emmean,
                    color = tmt,
                    group = tmt,
                    ymin = lower.CL,
                    ymax = upper.CL),
                width = .1,
                alpha = 0.9) +
  geom_point(data = osml_emmeans,
             aes(x = tmt,
                  y = emmean,
                  #color = tmt,
                shape = tmt,
```

```r
                 fill = tmt),
             color = "black",
               size = 4) +
  theme_classic() +
  scale_shape_manual(values = my_shapes_box, name = "") +
  scale_fill_manual(values = my_colors, name = "") +
  scale_color_manual(values = my_colors, name = "") +
  scale_y_continuous(limits = c(290,470),
                     breaks = c(seq(300,450, by = 50)),
                     labels = c(seq(300,450, by = 50))) +
  scale_x_discrete(labels = c("Cool Humid\n0.6 kPa",
                              "Hot Humid\n1.1 kPa",
                              "Cool Dry\n2.5 kPa",
                              "Hot Dry\n3.8 kPa")) +
  xlab("") +

  annotate(geom = "text", x = 4, y = 427, label = "A",
           size = 3) +
  #annotate(geom = "text", x = 4, y = 417, label = "362", #HD
   #        size = 3) +
  annotate(geom = "text", x = 2, y = 452, label = "A",
           size = 3) +
  #annotate(geom = "text", x = 2, y = 442, label = "354", #HH
   #        size = 3) +
  annotate(geom = "text", x = 3, y = 437, label = "A",
           size = 3) +
  #annotate(geom = "text", x = 3, y = 427, label = "349", #CD
   #        size = 3) +
  annotate(geom = "text", x = 1, y = 470, label = "A",
           size = 3) +
  #annotate(geom = "text", x = 1, y = 460, label = "350", #CH
   #        size = 3) +

  ylab(bquote('Osmolality (mmol '*kg^-1*')')) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 12),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        #axis.text.x = element_blank(),
        legend.text = element_text(color = "black",
                                   family = "sans",
                                   size = 8),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = unit(c(0, #top
                             0, #right
                             0, #bottom
                             0), "mm")
        ) -> osml_end_boxplot
osml_end_boxplot
```

```
## Warning: Removed 10 rows containing missing values (`geom_point()`).
```




```
osml_emmeans
```

```
##                      tmt   emmean       SE       df lower.CL upper.CL
## 1 Cool Humid (0.6 kPa) 350.3202 10.16423 4.874923 323.9893 376.6511
## 2  Hot Humid (1.1 kPa) 354.0963 10.13690 4.824373 327.7506 380.4420
## 3   Cool Dry (2.5 kPa) 349.1611 10.17830 4.903510 322.8413 375.4808
## 4    Hot Dry (3.8 kPa) 362.0244 10.15727 4.863220 335.6918 388.3569
##                     response
## 1 Plasma Osmolality (mmol/kg)
## 2 Plasma Osmolality (mmol/kg)
## 3 Plasma Osmolality (mmol/kg)
## 4 Plasma Osmolality (mmol/kg)
```

```
osml_pairwise
```

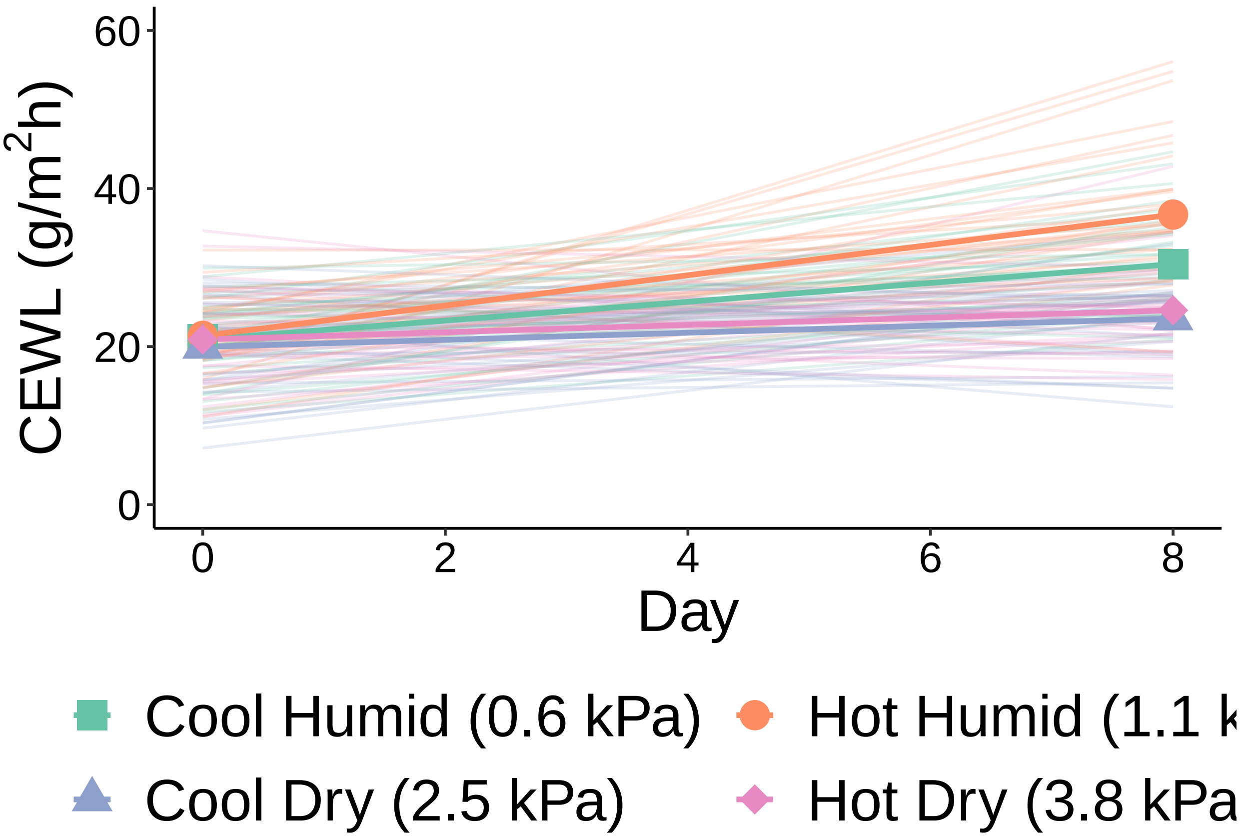
```
##                                      contrast   estimate       SE       df
## 1 Cool Humid (0.6 kPa) - Hot Humid (1.1 kPa)  -3.776107 5.031118 116.0115
## 2  Cool Humid (0.6 kPa) - Cool Dry (2.5 kPa)   1.159115 5.128594 116.0589
## 3   Cool Humid (0.6 kPa) - Hot Dry (3.8 kPa) -11.704175 5.078569 116.0347
## 4   Hot Humid (1.1 kPa) - Cool Dry (2.5 kPa)   4.935222 5.085161 116.0412
## 5    Hot Humid (1.1 kPa) - Hot Dry (3.8 kPa)  -7.928068 5.035558 116.0197
## 6     Cool Dry (2.5 kPa) - Hot Dry (3.8 kPa) -12.863291 5.114607 116.0115
##       t.ratio    p.value                    response
## 1 -0.7505502 0.87626200 Plasma Osmolality (mmol/kg)
## 2  0.2260104 0.99590771 Plasma Osmolality (mmol/kg)
## 3 -2.3046207 0.10288296 Plasma Osmolality (mmol/kg)
## 4  0.9705144 0.76645671 Plasma Osmolality (mmol/kg)
## 5 -1.5744172 0.39720576 Plasma Osmolality (mmol/kg)
```

```
## 6 -2.5150105 0.06282759 Plasma Osmolality (mmol/kg)
```

CEWL

Ind + Means

```
ggplot() +
  geom_line(data = dat[complete.cases(dat$CEWL_g_m2h_mean),],
            aes(x = day_n,
                y = CEWL_g_m2h_mean,
                color = tmt,
                group = individual_ID),
            alpha = 0.2) +
  geom_line(data = means[complete.cases(means$mean_CEWL),],
            aes(x = day_n,
                y = mean_CEWL,
                color = tmt,
                group = tmt),
            alpha = 1,
            size = 1) +
  geom_point(data = means[complete.cases(means$mean_CEWL),],
            aes(x = day_n,
                y = mean_CEWL,
                color = tmt,
                shape = tmt),
            alpha = 1,
            size = 5) +
  theme_classic() +
  scale_shape_manual(values = c(15:18), name = "") +
  scale_x_continuous(breaks = c(0, 2, 4, 6, 8)) +
  scale_color_brewer(palette = "Set2", name = "") +
  xlab("Day") +
  ylim(0,60) +
  ylab(bquote('CEWL (g/'*m^2*'h)')) +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 22),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 16),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 22),
        legend.text.align = 0,
        legend.position = "bottom"
        #legend.position = c(0.25,0.85)
        ) -> CEWL_fig
CEWL_fig
```

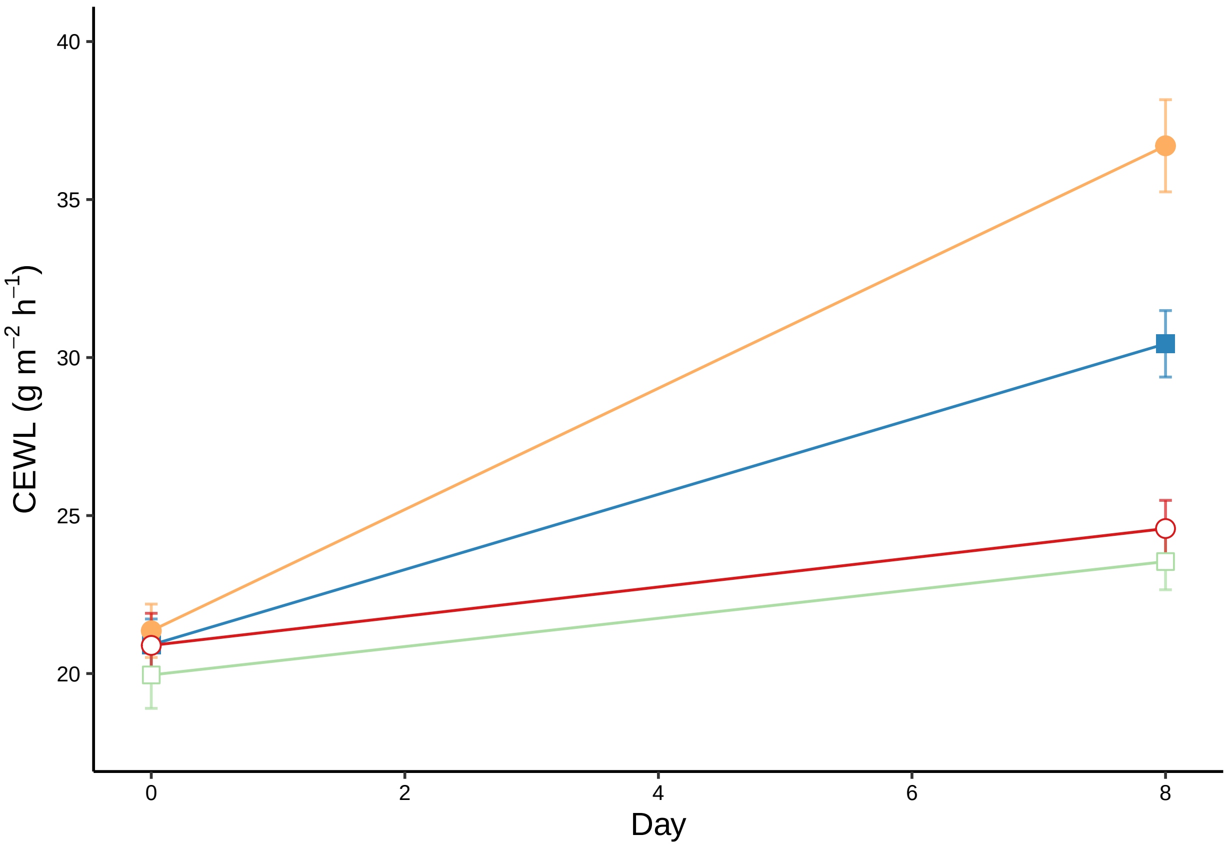



```
#ggsave(filename = "CEWL_fig1.jpeg",
 #      plot = CEWL_fig,
  #     path = "./results_figures",
   #    device = "jpeg",
    #   dpi = 1200,
     #  width = 6, height = 6)
```

Means Only *MS*

```
ggplot() +
  geom_errorbar(data = means[complete.cases(means$mean_CEWL),],
                aes(x = day_n,
               y = mean_CEWL,
               color = tmt,
               group = tmt,
               ymin = mean_CEWL-se_CEWL,
               ymax = mean_CEWL+se_CEWL),
               width = .1,
               #position=position_dodge(.1),
               alpha = 0.7) +
  geom_line(data = means[complete.cases(means$mean_CEWL),],
            aes(x = day_n,
               y = mean_CEWL,
               color = tmt,
               #linetype = tmt,
               group = tmt),
            alpha = 1,
```

```
            size = .5) +
  geom_point(data = means[complete.cases(means$mean_CEWL),],
            aes(x = day_n,
                y = mean_CEWL,
                color = tmt,
                shape = tmt),
            fill = "white",
            alpha = 1,
            size = 3) +
  theme_classic() +
  scale_shape_manual(values = my_shapes, name = "",
                     labels = my_labels) +
  scale_fill_manual(values = my_colors, name = "",
                     labels = my_labels) +
  scale_color_manual(values = my_colors, name = "",
                     labels = my_labels) +
  scale_x_continuous(breaks = c(0, 2, 4, 6, 8)) +
  scale_y_continuous(breaks = c(20, 25, 30, 35, 40),
                     limits = c(18,40)) +
  xlab("Day") +
  ylab(bquote('CEWL (g '*m^-2*' '*h^-1*')')) +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 12),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.text.align = 0,
        legend.position = "none"
        #legend.position = c(0.25,0.85)
        ) -> CEWL_fig_min
CEWL_fig_min
```



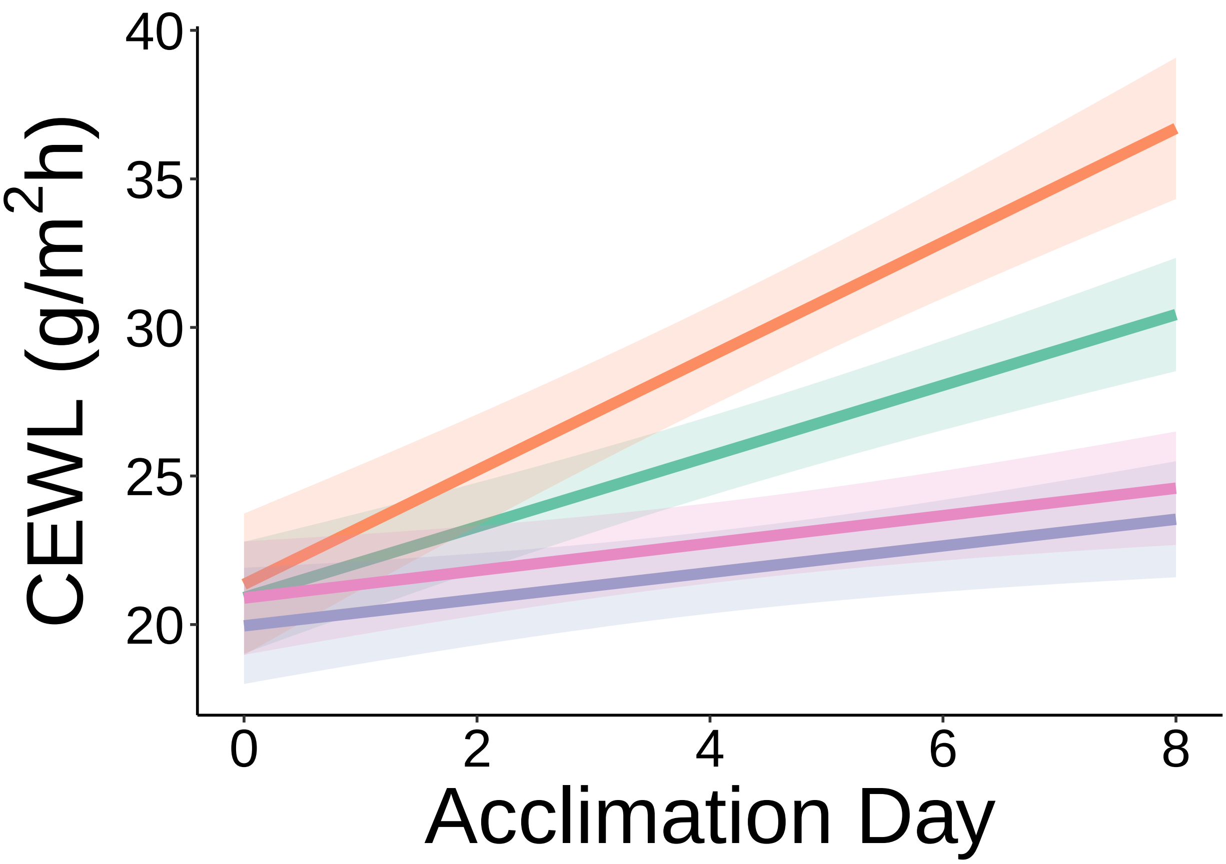

```
# use ggarrange so legend is centered
CEWL_fig_formatted <- ggarrange(CEWL_fig_min,
                                ncol = 1, nrow = 1,
                                common.legend = TRUE,
                                legend = "bottom")
# save
ggsave(filename = "experiment_CEWL_fig.pdf",
       plot = CEWL_fig_formatted,
       path = "./results_figures",
       device = "pdf",
       dpi = 600,
       units = "mm",
       width = 80, height = 90)
```

LM + SE

```
ggplot() +
  stat_smooth(data = dat,
              aes(x = day_n,
                  y = CEWL_g_m2h_mean,
                  color = tmt,
                  fill = tmt,
                  group = tmt),
              formula = y ~ x,
              method = "lm",
              se = T,
              size = 2,
```

```
              alpha = 0.2) +
  theme_classic() +
  scale_x_continuous(breaks = c(0, 2, 4, 6, 8)) +
  scale_color_brewer(palette = "Set2", name = "") +
  scale_fill_brewer(palette = "Set2", name = "") +
  xlab("Acclimation Day") +
  ylab(bquote('CEWL (g/'*m^2*'h)')) +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 30),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 20),
        legend.text = element_text(color = "black",
                                   family = "sans",
                                   size = 22),
        legend.text.align = 0,
        legend.position = "none"
        #legend.position = c(0.25,0.85)
        ) -> CEWL_lm_fig
CEWL_lm_fig
```

```
## Warning: Removed 669 rows containing non-finite values (`stat_smooth()`).
```

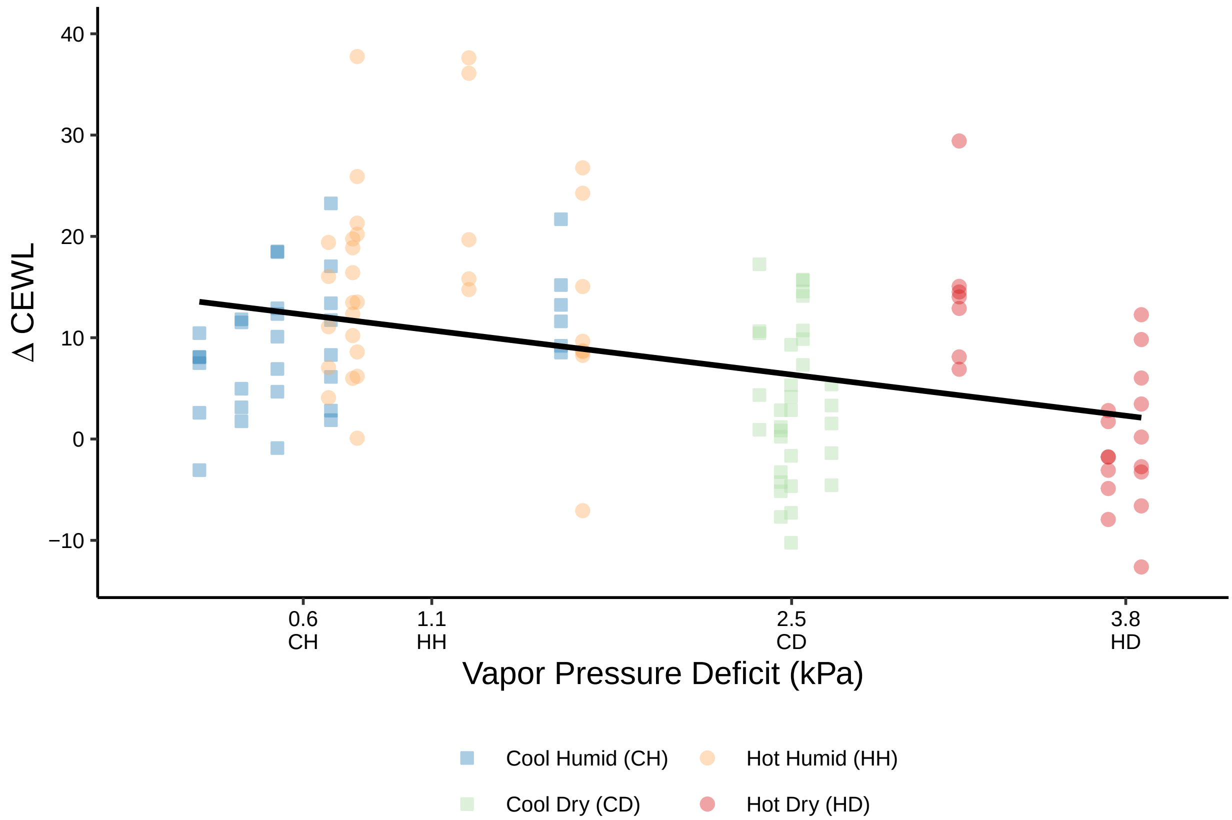


delta CEWL ~ VPD

```
ggplot(data = dat_no_rehab_deltaCEWL) +
  geom_point(aes(x = VPD_kPa_tmttrial,
                 y = delta_CEWL,
                 color = tmt,
               fill = tmt,
               shape = tmt),
             size = 2,
            alpha = 0.4
           ) +
  geom_smooth(aes(x = VPD_kPa_tmttrial,
                  y = delta_CEWL),
              se = F,
              formula = y ~ x,
              method = "lm",
              color = "black") +
  #geom_smooth(aes(x = VPD_kPa_tmttrial,
   #                y = delta_CEWL),
    #          se = F,
     #         formula = y ~ poly(x, 2),
      #        method = "lm",
       #       color = "red") +
  theme_classic() +
  scale_shape_manual(values = my_shapes_box, name = "",
                     labels = c("Cool Humid (CH)",
                                "Hot Humid (HH)",
                                "Cool Dry (CD)",
                                "Hot Dry (HD)")) +
  scale_fill_manual(values = my_colors, name = "",
                     labels = c("Cool Humid (CH)",
                                "Hot Humid (HH)",
                                "Cool Dry (CD)",
                                "Hot Dry (HD)")) +
  scale_color_manual(values = my_colors, name = "",
                     labels = c("Cool Humid (CH)",
                                "Hot Humid (HH)",
                                "Cool Dry (CD)",
                                "Hot Dry (HD)")) +
  scale_y_continuous(limits = c(-13,40),
                     breaks = c(seq(-10,40, by = 10)),
                     labels = c(seq(-10,40, by = 10))
                     ) +
  scale_x_continuous(limits = c(0,4),
                     breaks = c(0.6, 1.1, 2.5, 3.8),
                     labels = c("0.6\nCH",
                                 "1.1\nHH",
                                 "2.5\nCD",
                                 "3.8\nHD")) +
  xlab("Vapor Pressure Deficit (kPa)") +
  ylab(expression(Delta ~ 'CEWL')) +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
```

```
                                    size = 12),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        #axis.text.x = element_blank(),
        legend.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.text.align = 0,
        legend.position = "bottom",
        legend.spacing.y = unit(0, "mm")
        #plot.margin = unit(c(0, #top
         #                    0, #right
          #                   0, #bottom
           #                  0), "mm")
        ) -> CEWL_VPD_fig
CEWL_VPD_fig
```

```
## Warning: Removed 11 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 11 rows containing missing values (`geom_point()`).
```




```
# use ggarrange so legend is centered
CEWL_VPD_fig_formatted <- ggarrange(CEWL_VPD_fig,
                                    ncol = 1, nrow = 1,
                                    common.legend = TRUE,
                                    legend = "bottom")
```

```
## Warning: Removed 11 rows containing non-finite values (`stat_smooth()`).
```

```
## Removed 11 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 11 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```

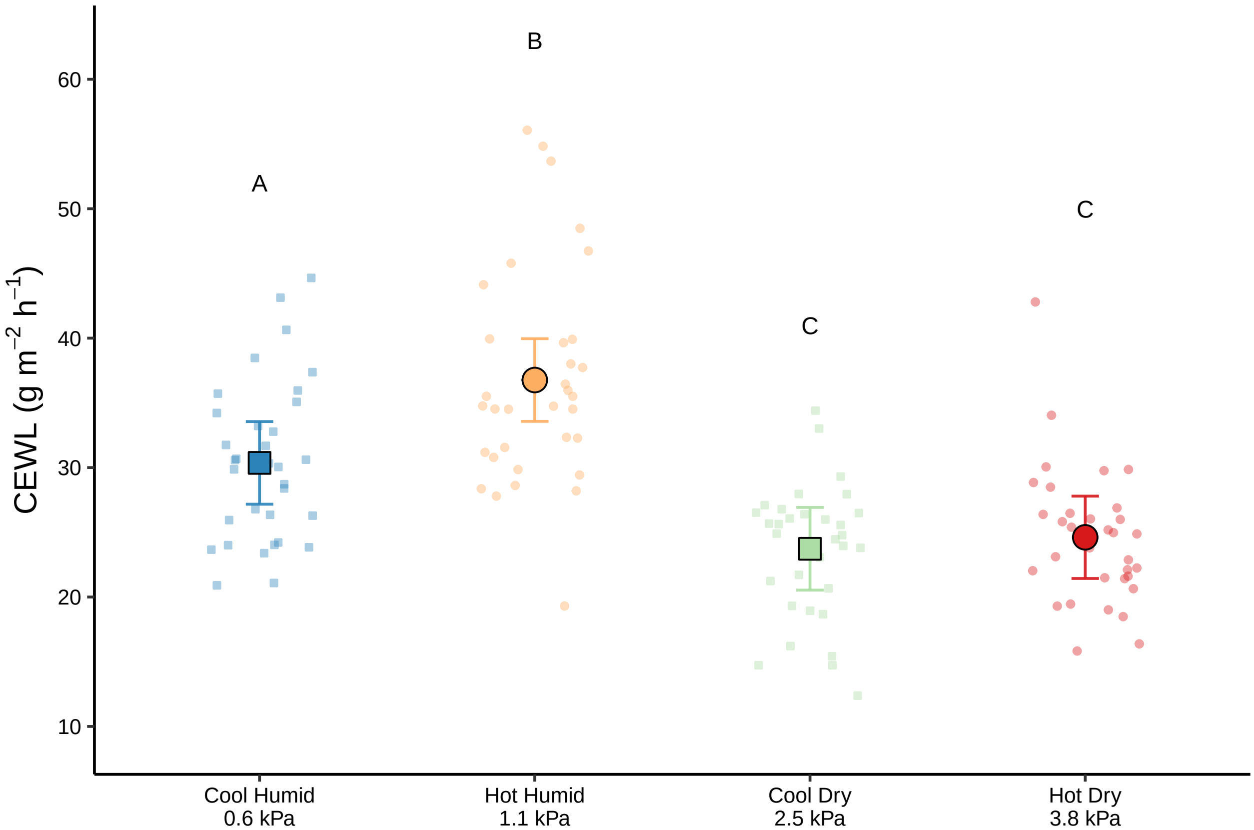
```r
# save
ggsave(filename = "exp_CEWL_delta_VPD.pdf",
       plot = CEWL_VPD_fig_formatted,
       path = "./results_figures",
       device = "pdf",
       dpi = 600,
       units = "mm",
       width = 80, height = 90)
```

Ending Values *MS*

```r
ggplot() +
  geom_jitter(data = end_vals,
              aes(x = tmt,
                   y = CEWL_g_m2h_mean,
                   color = tmt,
                 fill = tmt,
                 shape = tmt),
               size = 1,
              alpha = 0.4,
              position = position_jitter(height = 0, width = 0.2)) +
  geom_errorbar(data = CEWL_emmeans,
                aes(x = tmt,
                    y = emmean,
                    color = tmt,
                    group = tmt,
                    ymin = lower.CL,
                    ymax = upper.CL),
                width = .1,
                alpha = 0.9) +
  geom_point(data = CEWL_emmeans,
             aes(x = tmt,
                   y = emmean,
                   #color = tmt,
                 shape = tmt,
                 fill = tmt),
             color = "black",
               size = 4) +
  theme_classic() +
  scale_shape_manual(values = my_shapes_box, name = "") +
  scale_fill_manual(values = my_colors, name = "") +
  scale_color_manual(values = my_colors, name = "") +
  scale_x_discrete(labels = c("Cool Humid\n0.6 kPa",
                              "Hot Humid\n1.1 kPa",
                              "Cool Dry\n2.5 kPa",
                              "Hot Dry\n3.8 kPa")) +
  scale_y_continuous(limits = c(9,63),
                     breaks = c(seq(10,60, by = 10)),
                     labels = c(seq(10,60, by = 10))) +
```

```
  xlab("") +
  ylab(bquote('CEWL (g '*m^-2*' '*h^-1*')')) +

  annotate(geom = "text", x = 4, y = 50, label = "C",
           size = 3) +
  #annotate(geom = "text", x = 4, y = 47, label = "25", #HD
   #        size = 3) +
  annotate(geom = "text", x = 2, y = 63, label = "B",
           size = 3) +
  #annotate(geom = "text", x = 2, y = 60, label = "37", #HH
  #         size = 3) +
  annotate(geom = "text", x = 3, y = 41, label = "C",
           size = 3) +
  #annotate(geom = "text", x = 3, y = 38, label = "24", #CD
   #        size = 3) +
  annotate(geom = "text", x = 1, y = 52, label = "A",
           size = 3) +
  #annotate(geom = "text", x = 1, y = 49, label = "30", #CH
   #        size = 3) +

  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 12),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        #axis.text.x = element_blank(),
        legend.text = element_text(color = "black",
                                   family = "sans",
                                   size = 8),
        legend.text.align = 0,
        legend.position = "none",
        plot.margin = unit(c(0, #top
                             0, #right
                             0, #bottom
                             0), "mm")
        ) -> CEWL_end_boxplot
CEWL_end_boxplot
```

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




```
CEWL_emmeans
```

```
##                      tmt   emmean       SE       df lower.CL upper.CL
## 1 Cool Humid (0.6 kPa) 30.36157 1.445712 10.68500 27.16810 33.55505
## 2  Hot Humid (1.1 kPa) 36.76428 1.446643 10.69577 33.56916 39.95941
## 3   Cool Dry (2.5 kPa) 23.72439 1.446548 10.69551 20.52946 26.91931
## 4    Hot Dry (3.8 kPa) 24.61233 1.434910 10.37491 21.43074 27.79392
##       response
## 1 CEWL (g/m2h)
## 2 CEWL (g/m2h)
## 3 CEWL (g/m2h)
## 4 CEWL (g/m2h)
```

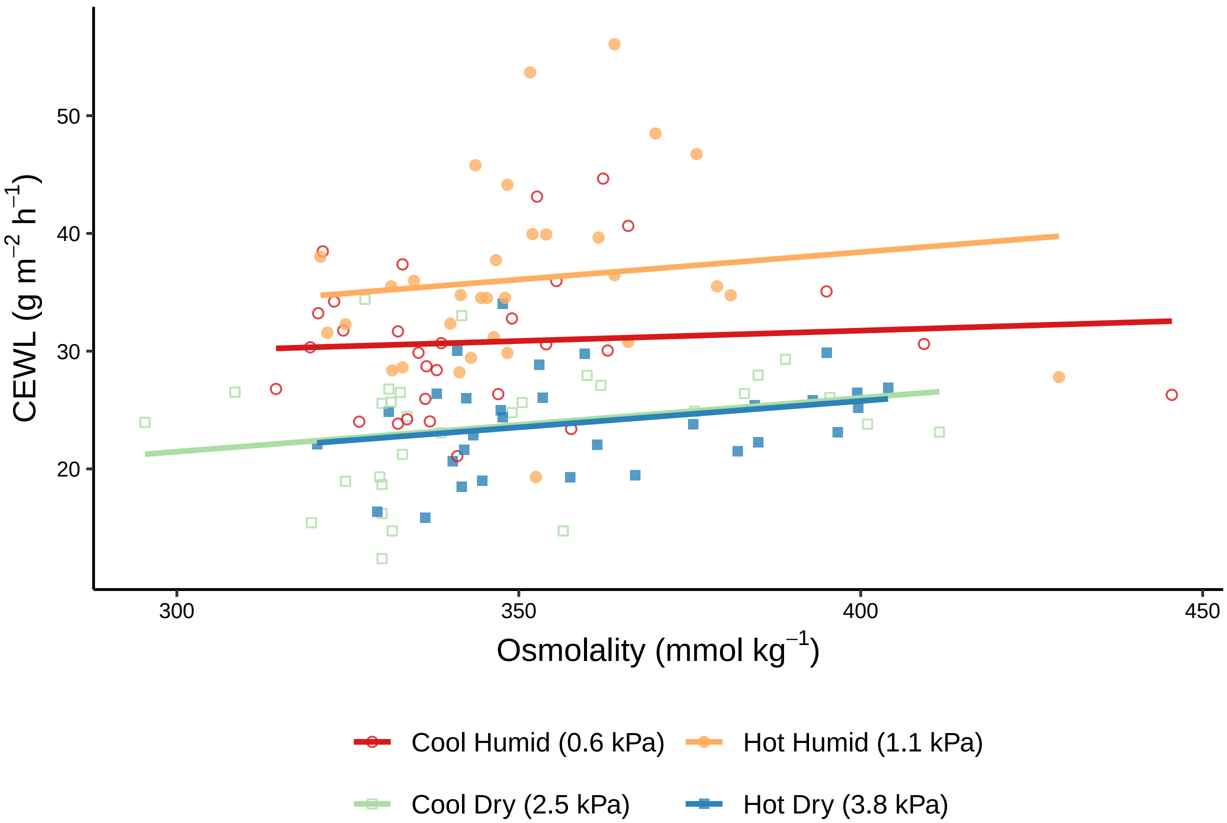
```
CEWL_pairwise
```

```
##                                       contrast   estimate       SE       df
## 1 Cool Humid (0.6 kPa) - Hot Humid (1.1 kPa) -6.4027092 1.477747 125.0660
## 2  Cool Humid (0.6 kPa) - Cool Dry (2.5 kPa)  6.6371867 1.482776 125.4155
## 3   Cool Humid (0.6 kPa) - Hot Dry (3.8 kPa)  5.7492451 1.468760 125.2015
## 4   Hot Humid (1.1 kPa) - Cool Dry (2.5 kPa) 13.0398959 1.482912 125.4316
## 5    Hot Humid (1.1 kPa) - Hot Dry (3.8 kPa) 12.1519543 1.469693 125.2717
## 6     Cool Dry (2.5 kPa) - Hot Dry (3.8 kPa) -0.8879416 1.467098 125.0811
##      t.ratio      p.value     response
## 1 -4.3327496 1.740823e-04 CEWL (g/m2h)
## 2  4.4761900 9.849699e-05 CEWL (g/m2h)
## 3  3.9143543 8.436015e-04 CEWL (g/m2h)
## 4  8.7934385 9.026113e-14 CEWL (g/m2h)
## 5  8.2683648 1.057265e-12 CEWL (g/m2h)
```

```
## 6 -0.6052366 9.302438e-01 CEWL (g/m2h)
```

Exp CEWL ~ Osml

```
end_vals_CEWL_osml <- dat %>%
  dplyr::filter(day_n == 8) %>%
  dplyr::filter(complete.cases(CEWL_g_m2h_mean, osmolality_mmol_kg_mean))

ggplot(end_vals_CEWL_osml) +
  aes(x = osmolality_mmol_kg_mean,
      y = CEWL_g_m2h_mean,
      color = tmt,
      shape = tmt) +
  geom_point(size = 1.5,
             alpha = 0.8) +
  stat_smooth(formula = y ~ x,
              method = "lm",
              se = F,
              size = 1,
              alpha = 0.9) +
  theme_classic() +
  xlab(bquote('Osmolality (mmol '*kg^-1*')')) +
  ylab(bquote('CEWL (g '*m^-2*' '*h^-1*')')) +
  #xlim(300, 400) +
  #ylim(0, 40) +
  scale_shape_manual(values = c(21,19, 22,15), name = "") +
  scale_fill_brewer(palette = "Spectral", name = "") +
  scale_color_brewer(palette = "Spectral", name = "") +
  scale_x_continuous(breaks = c(300, 350, 400, 450)) +
  scale_y_continuous(breaks = c(20, 30, 40, 50),
                     limits = c(12,57)) +
  guides(shape = guide_legend(nrow = 2, byrow = TRUE)) +
  theme(text = element_text(color = "black",
                            family = "sans",
                            size = 12),
        axis.text = element_text(color = "black",
                                 family = "sans",
                                 size = 8),
        legend.position = "bottom"
        #axis.text.y = element_blank(),
        #plot.margin = unit(c(0.1,0,0.1,0.45), "cm")
        ) -> exp_end_CEWL_osml_fig
exp_end_CEWL_osml_fig
```




```
# will need to save using figure arrange to make legend centered if I use that fig
ggsave(filename = "exp_CEWL_osml_fig.pdf",
       plot = exp_end_CEWL_osml_fig,
       path = "./results_figures",
       device = "pdf",
       dpi = 600,
       units = "mm",
       width = 80, height = 90)
```

Multi-Figures

over time:

```
ggarrange(osml_fig_min,
          hct_fig_min,
          SMI_fig_min,
          ncol = 1, nrow = 3,
          labels = c("A", "B", "C"),
          common.legend = TRUE,
          legend = "bottom"
          ) -> experiment_multi_fig
```

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

## Warning: Removed 12 rows containing missing values (`geom_point()`).

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

## Warning: Removed 12 rows containing missing values (`geom_point()`).
```

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

```
#experiment_multi_fig
# export figure
ggsave(filename = "experiment_multi_fig.pdf",
       plot = experiment_multi_fig,
       path = "./results_figures",
       device = "pdf",
       dpi = 600,
       units = "mm",
       width = 80, height = 210)
```

end values:

```
ggarrange(CEWL_end_boxplot,
          osml_end_boxplot,
          hct_end_boxplot,
          SMI_end_boxplot,
          ncol = 2, nrow = 2,
          labels = c("A", "B", "C", "D"),
          widths = c(2, 2.045), heights = c(2, 2),
          common.legend = FALSE
          ) -> ending_values_multi_fig
```

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

## Warning: Removed 10 rows containing missing values (`geom_point()`).

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

```
#ending_values_multi_fig

ggsave(filename = "exp_end_val_multi_fig.pdf",
       plot = ending_values_multi_fig,
       path = "./results_figures",
       device = "pdf",
       dpi = 600,
       units = "mm",
       width = 180, height = 150)
```