Climate Water Loss Experiment - Plasma Osmolality Data Wrangling

Savannah Weaver

2021

Contents

Packages

Background and Goals

Blood was drawn from the postorbital sinus of adult male *Sceloporus occidentalis* between June - August 2021. After centrifuging and separating, plasma was run on a VAPRO vapor pressure osmometer in 1-4 replicates, when plasma volume allowed. In this R script, I check the distribution of replicates, omit outliers, and average remaining replicates. The final values will be more precise and accurate estimates of the true plasma osmolality for each lizard, and those values will be used in the capture_analysis and experiment_analysis R script files. Please refer to **doi:** for the published scientific journal article and full details.

Load Data

```r
osml_reps <- read.csv("./data/osmolality.csv",
                na.strings = c("","NA"),
                header = TRUE
                ) %>%
    dplyr::mutate(date_blood_drawn = as.Date(date_blood_drawn,
```

```
                                            format = "%m/%d/%y"),
                  date_osmom_run = as.Date(date_osmom_run,
                                           format = "%m/%d/%y"),
                  time_osmom_run = as.POSIXct(time_osmom_run,
                                              format = "%H:%M"),
                  individual_ID = as.factor(individual_ID),
                  replicate_no = as.factor(replicate_no),
                  osmolality_mmol_kg = as.numeric(osmolality_mmol_kg)
                  )
summary(osml_reps)
```

```
##  date_blood_drawn     date_osmom_run       time_osmom_run
##  Min.   :2021-06-16   Min.   :2021-06-16   Min.   :2021-10-21 01:00:00
##  1st Qu.:2021-06-30   1st Qu.:2021-06-30   1st Qu.:2021-10-21 10:38:00
##  Median :2021-07-24   Median :2021-07-24   Median :2021-10-21 12:48:00
##  Mean   :2021-07-22   Mean   :2021-07-22   Mean   :2021-10-21 12:58:11
##  3rd Qu.:2021-08-16   3rd Qu.:2021-08-16   3rd Qu.:2021-10-21 15:14:00
##  Max.   :2021-09-01   Max.   :2021-09-02   Max.   :2021-10-21 21:22:00
##                                            NA's   :211
##  individual_ID  replicate_no osmolality_mmol_kg  notes
##  339    : 13    1:544        Min.   :247.0       Mode:logical
##  206    : 12    2:531        1st Qu.:336.0       NA's:1484
##  207    : 12    3:406        Median :351.0
##  208    : 12    4:  3        Mean   :357.3
##  211    : 12                 3rd Qu.:370.0
##  214    : 12                 Max.   :596.0
##  (Other):1411
```

Check Data

Dates

Blood was drawn on day 0, 4, 8, and 10 of the experiment. Create a list of the dates expected to have blood
draw data, then determine whether I have dates outside those.

Trail 1: June 16-24 Trail 2: June 26 - July 4 Trial 3: July 20-28 Trial 4: August 8-16 Trial 5: August 22-30

```
                                # trial 1
expected_dates <- as.Date(c("2021-06-16", "2021-06-20",
                            "2021-06-24", "2021-06-26",
                            # trial 2
                            "2021-06-26", "2021-06-30",
                            "2021-07-04", "2021-07-06",
                            # trial 3
                            "2021-07-20", "2021-07-24",
                            "2021-07-28", "2021-07-30",
                            # trial 4
                            "2021-08-08", "2021-08-12",
                            "2021-08-16", "2021-08-18",
                            # trial 5
                            "2021-08-22", "2021-08-26",
                            "2021-08-30", "2021-09-01"))

length(osml_reps$date_blood_drawn[osml_reps$date_blood_drawn %nin% expected_dates]
```

```
)
```

```
## [1] 0
```

There are zero blood draw dates that are not in our expected list.

Number of Blood Draws

Each lizard should have had their blood drawn on 4 different dates, unless they were taken out of the experiment early.

```r
# get ID's of the individuals that completed treatment
individuals <- read.csv("./data/tmt_assignments.csv") %>%
  dplyr::select(conclusion, individual_ID) %>%
  dplyr::filter(conclusion == "complete") %>%
  mutate(individual_ID = as.factor(individual_ID),
         conclusion = as.factor(conclusion))
summary(individuals)
```

```
##      conclusion  individual_ID
##  complete:134   201    :  1
##                 202    :  1
##                 203    :  1
##                 204    :  1
##                 205    :  1
##                 206    :  1
##                 (Other):128
```

```r
# calculate the number of dates for each individual
osml_reps %>%
  dplyr::filter(individual_ID %in% individuals$individual_ID) %>%
  group_by(individual_ID, date_blood_drawn) %>%
  summarise(n()) %>%
  group_by(individual_ID) %>%
  summarise(n()) %>%
  arrange(n())
```

```
## `summarise()` regrouping output by 'individual_ID' (override with `.groups` argument)
```

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

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

Wahoo, every lizard has blood draws from 4 different dates.

Replicates

Now, I will try to identify outliers within the replicates for a given individual on a given date. There must be at least 3 replicates to do this, so the first thing I need to do is figure out which individuals/dates have enough replicates, then subset my data to be only those individuals.

Individuals w 3+ Replicates

```
# identify individuals with 3-4 reps
enuf_reps <- osml_reps %>%
  group_by(individual_ID, date_blood_drawn) %>%
  mutate(count = n()) %>%
  dplyr::filter(count > 2) %>%
  arrange(count)
enuf_reps
```

```
## # A tibble: 1,224 x 8
## # Groups:   individual_ID, date_blood_drawn [407]
##    date_blood_drawn date_osmom_run time_osmom_run      individual_ID
##    <date>           <date>         <dttm>              <fct>
##  1 2021-06-16       2021-06-16     2021-10-21 11:00:00 201
##  2 2021-06-16       2021-06-16     2021-10-21 11:00:00 201
##  3 2021-06-16       2021-06-16     2021-10-21 11:00:00 201
##  4 2021-06-16       2021-06-16     2021-10-21 11:00:00 202
##  5 2021-06-16       2021-06-16     2021-10-21 11:00:00 202
##  6 2021-06-16       2021-06-16     2021-10-21 11:00:00 202
##  7 2021-06-16       2021-06-16     2021-10-21 11:10:00 203
##  8 2021-06-16       2021-06-16     2021-10-21 11:10:00 203
##  9 2021-06-16       2021-06-16     2021-10-21 11:10:00 203
## 10 2021-06-16       2021-06-16     2021-10-21 11:25:00 205
## # ... with 1,214 more rows, and 4 more variables: replicate_no <fct>,
## #   osmolality_mmol_kg <dbl>, notes <lgl>, count <int>
```

```
# identify individuals with 1-2 reps
not_reps <- osml_reps %>%
  group_by(individual_ID, date_blood_drawn) %>%
  mutate(count = n()) %>%
  dplyr::filter(count < 3) %>%
  arrange(count)
not_reps
```

```
## # A tibble: 260 x 8
## # Groups:   individual_ID, date_blood_drawn [136]
##    date_blood_drawn date_osmom_run time_osmom_run      individual_ID
##    <date>           <date>         <dttm>              <fct>
##  1 2021-06-24       2021-06-24     2021-10-21 09:26:00 203
##  2 2021-06-24       2021-06-26     NA                  204
##  3 2021-06-24       2021-06-26     NA                  223
##  4 2021-06-26       2021-06-26     2021-10-21 09:47:00 201
##  5 2021-06-26       2021-06-26     2021-10-21 11:40:00 230
##  6 2021-07-04       2021-07-05     2021-10-21 10:03:00 227
##  7 2021-07-06       2021-07-06     2021-10-21 09:59:00 239
##  8 2021-07-28       2021-07-28     2021-10-21 10:32:00 259
##  9 2021-07-28       2021-07-28     2021-10-21 12:07:00 274
## 10 2021-07-30       2021-07-31     2021-10-21 13:55:00 291
```

```
## # ... with 250 more rows, and 4 more variables: replicate_no <fct>,
## #   osmolality_mmol_kg <dbl>, notes <lgl>, count <int>
# check total obs still add to original 1484
nrow(enuf_reps) + nrow(not_reps)
```

```
## [1] 1484
```

Assess Variation

We want the Coefficient of Variation (CV) among our technical replicates to be small. We need to calculate it to identify whether there may be outliers.

```
CVs <- enuf_reps %>%
  group_by(individual_ID, date_blood_drawn) %>%
  summarise(osml_mean = mean(osmolality_mmol_kg),
            osml_SD = sd(osmolality_mmol_kg),
            CV = (osml_SD/osml_mean) *100,
            min_osml = min(osmolality_mmol_kg),
            max_osml = max(osmolality_mmol_kg),
            osml_range = max_osml - min_osml
            )
```

```
## `summarise()` regrouping output by 'individual_ID' (override with `.groups` argument)
```

```
summary(CVs)
```

```
##  individual_ID date_blood_drawn      osml_mean        osml_SD
##  206    :  4   Min.   :2021-06-16   Min.   :295.3   Min.   : 0.000
##  207    :  4   1st Qu.:2021-06-30   1st Qu.:335.0   1st Qu.: 2.581
##  208    :  4   Median :2021-07-24   Median :350.0   Median : 4.359
##  211    :  4   Mean   :2021-07-23   Mean   :356.2   Mean   : 5.596
##  214    :  4   3rd Qu.:2021-08-16   3rd Qu.:368.0   3rd Qu.: 6.658
##  220    :  4   Max.   :2021-09-01   Max.   :576.0   Max.   :69.816
##  (Other):383
##        CV             min_osml        max_osml        osml_range
##  Min.   : 0.0000   Min.   :247.0   Min.   :300.0   Min.   :  0.00
##  1st Qu.: 0.7278   1st Qu.:330.5   1st Qu.:339.0   1st Qu.:  5.00
##  Median : 1.2021   Median :345.0   Median :356.0   Median :  8.00
##  Mean   : 1.5627   Mean   :351.1   Mean   :361.7   Mean   : 10.66
##  3rd Qu.: 1.8746   3rd Qu.:364.0   3rd Qu.:374.0   3rd Qu.: 12.50
##  Max.   :21.4600   Max.   :566.0   Max.   :596.0   Max.   :134.00
##
```
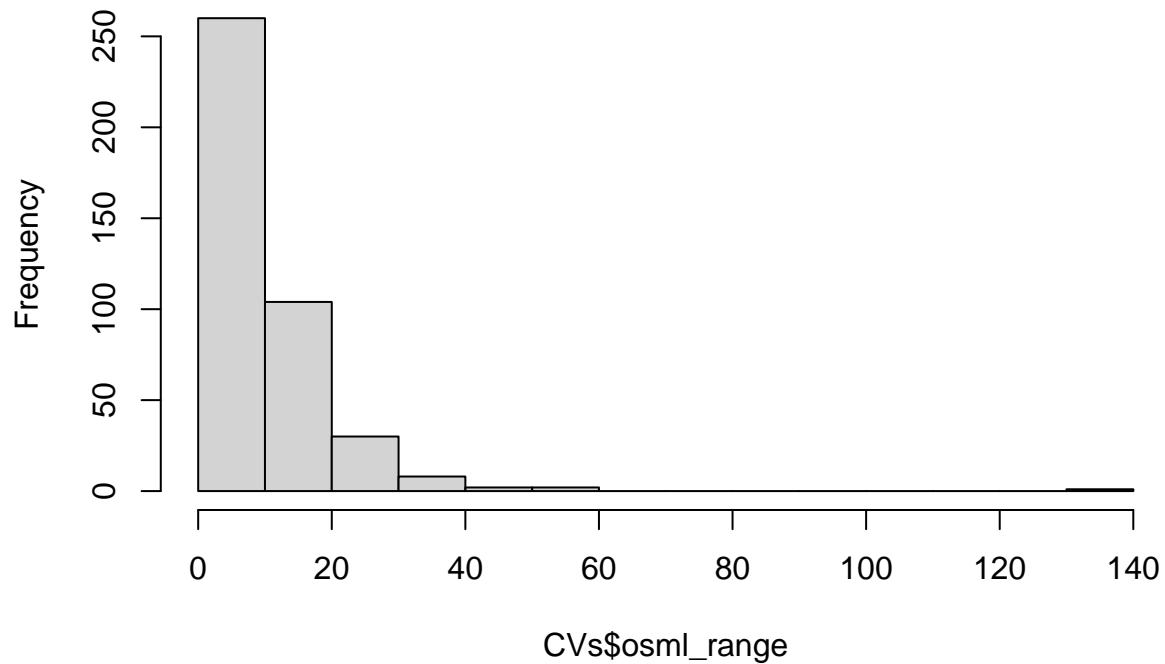
```
hist(CVs$CV)
```

Histogram of CVs$CV
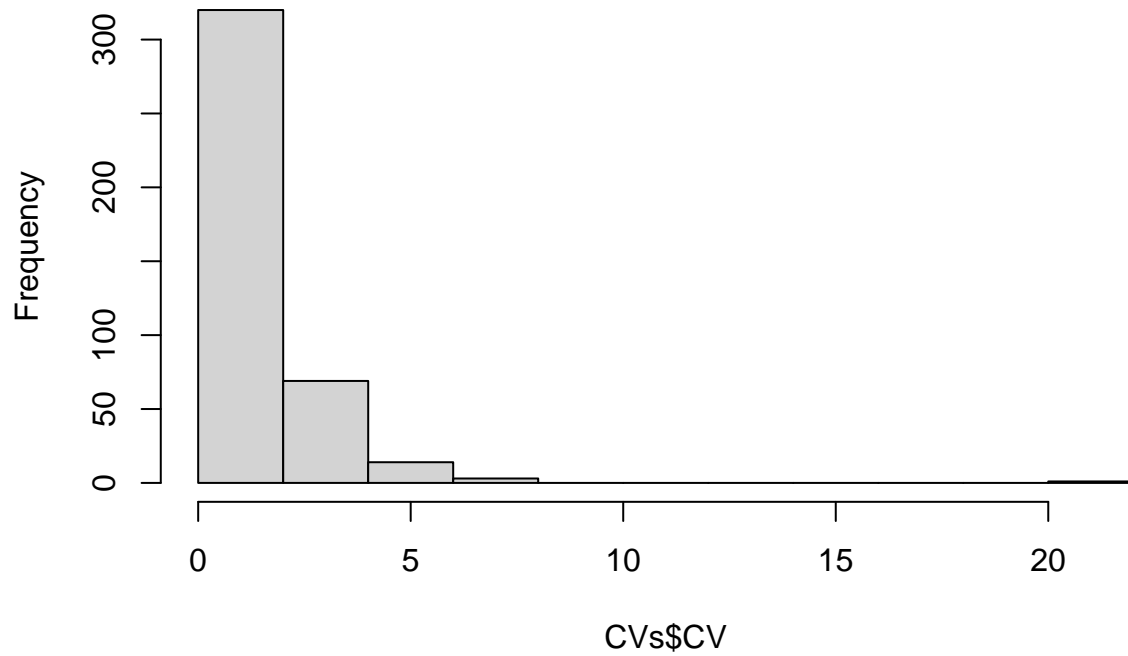

```r
hist(CVs$osml_range)
```

Histogram of CVs$osml_range

Ideally, CV would be 10-15%. If it's larger, and one of the replicates is very different than the others, we can assume that the replicates that are closer together are more reliable.

The CV is >10 for only one lizard on one date, so our replicates are likely to accurately represent the true

value. We don't want to lose accuracy by searching for precision, so we will only remove the one point driving the enormous CV value.

Find Outlier

Determine which replicate is an outlier and pulls the CV above our acceptable range (10-15%).

```r
CV_all <- enuf_reps %>%
    group_by(individual_ID, date_blood_drawn) %>%
    summarise(mean = mean(osmolality_mmol_kg),
              SD = sd(osmolality_mmol_kg),
              CV_all = (SD/mean) *100) %>%
    dplyr::select(-mean, -SD)
```

```
## `summarise()` regrouping output by 'individual_ID' (override with `.groups` argument)
```

```r
CV_12 <- enuf_reps %>%
    dplyr::filter(replicate_no != 3) %>%
    group_by(individual_ID, date_blood_drawn) %>%
    summarise(mean = mean(osmolality_mmol_kg),
              SD = sd(osmolality_mmol_kg),
              CV_12 = (SD/mean) *100) %>%
    dplyr::select(-mean, -SD)
```

```
## `summarise()` regrouping output by 'individual_ID' (override with `.groups` argument)
```

```r
CV_23 <- enuf_reps %>%
    dplyr::filter(replicate_no != 1) %>%
    group_by(individual_ID, date_blood_drawn) %>%
    summarise(mean = mean(osmolality_mmol_kg),
              SD = sd(osmolality_mmol_kg),
              CV_23 = (SD/mean) *100) %>%
    dplyr::select(-mean, -SD)
```

```
## `summarise()` regrouping output by 'individual_ID' (override with `.groups` argument)
```

```r
CV_31 <- enuf_reps %>%
    dplyr::filter(replicate_no != 2) %>%
    group_by(individual_ID, date_blood_drawn) %>%
    summarise(mean = mean(osmolality_mmol_kg),
              SD = sd(osmolality_mmol_kg),
              CV_31 = (SD/mean) *100) %>%
    dplyr::select(-mean, -SD)
```

```
## `summarise()` regrouping output by 'individual_ID' (override with `.groups` argument)
```

```r
compare <- CV_all %>%
    left_join(CV_12, by = c('individual_ID', 'date_blood_drawn')) %>%
    left_join(CV_23, by = c('individual_ID', 'date_blood_drawn')) %>%
    left_join(CV_31, by = c('individual_ID', 'date_blood_drawn')) %>%
  arrange(desc(CV_all)) %>%
  dplyr::filter(CV_all>10)
compare
```

```
## # A tibble: 1 x 6
## # Groups:   individual_ID [1]
##    individual_ID date_blood_drawn CV_all CV_12 CV_23 CV_31
##    <fct>         <date>            <dbl> <dbl> <dbl> <dbl>
```

```
## 1 225              2021-06-26           21.5   30.2   24.0   6.40
```

We should remove replicate 2 for individual 225 on June 26.

Remove Outlier

```r
enuf_reps[200, "osmolality_mmol_kg"] = NA

# check
enuf_reps %>%
  dplyr::filter(individual_ID == 225,
                date_blood_drawn == as.Date("2021-06-26",
                                            format= "%Y-%m-%d")
                #replicate_no == 2
                )
```

```
## # A tibble: 3 x 8
## # Groups:   individual_ID, date_blood_drawn [1]
##   date_blood_drawn date_osmom_run time_osmom_run      individual_ID replicate_no
##   <date>           <date>         <dttm>              <fct>         <fct>
## 1 2021-06-26       2021-06-26     2021-10-21 09:42:00 225           1
## 2 2021-06-26       2021-06-26     2021-10-21 09:42:00 225           2
## 3 2021-06-26       2021-06-26     2021-10-21 09:42:00 225           3
## # ... with 3 more variables: osmolality_mmol_kg <dbl>, notes <lgl>, count <int>
```

```r
# filter out the one NA
enuf_reps_trimmed <- enuf_reps %>%
  dplyr::filter(complete.cases(osmolality_mmol_kg))
```

Average Remaining Replicates

Now that the outliers are removed from the technical replicates when there were enough replicates to identify them, I will average the remaining replicates of the rejoined data for lizards with 1-2 and 3-4 replicates.

```r
osml_means <- not_reps %>%
  rbind(enuf_reps_trimmed) %>%
  group_by(date_blood_drawn, individual_ID) %>%
  summarise(osmolality_mmol_kg_mean = mean(osmolality_mmol_kg))
```

```
## `summarise()` regrouping output by 'date_blood_drawn' (override with `.groups` argument)
```

Export

```r
write.csv(osml_means, "./data/osml_means_clean.csv")
```