BNLL Plasma Osmolality Data Wrangling

Savannah Weaver

Contents

Packages

```
`%nin%` = Negate(`%in%`)
if (!require("tidyverse")) install.packages("tidyverse")
```

```
## Loading required package: tidyverse

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("tidyverse") # workflow and plots
```

Background and Goals

Blood was drawn from the postorbital sinus of Blunt-nosed Leopard Lizards (*Gambelia sila*) between April - July 2021. After centrifuging and separating, plasma was run on a VAPRO vapor pressure osmometer in 1-3 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 analyses R script file. Please refer to **doi:** for the published scientific paper and full details.

Load Data

```
osml_reps <- read.csv("./data/osmolality.csv",
                na.strings = c("","NA"),
                header = TRUE
                ) %>%
    dplyr::mutate(blood_draw_date = as.Date(blood_draw_date,
                                             format = "%m/%d/%y"),
                  individual_ID = as.factor(individual_ID),
                  replicate_no = as.factor(replicate_no),
                  osmolality_mmol_kg = as.numeric(osmolality_mmol_kg),
                  hemolyzed_Y_N = as.factor(hemolyzed_Y_N),
                  )
summary(osml_reps)
```

```
##  blood_draw_date      individual_ID replicate_no osmolality_mmol_kg
##  Min.   :2021-04-23   F-12   :  9   1:116        Min.   :306.0
##  1st Qu.:2021-04-24   M-19   :  9   2:107        1st Qu.:350.0
##  Median :2021-04-25   M-09   :  8   3: 84        Median :366.0
##  Mean   :2021-05-09   M-10   :  8                Mean   :368.5
##  3rd Qu.:2021-05-08   M-11   :  8                3rd Qu.:382.0
##  Max.   :2021-07-14   M-20   :  8                Max.   :452.0
##                       (Other):257
##  hemolyzed_Y_N
##  N   :222
##  Y   : 80
##  NA's:  5
##
##
##
##
```

```
unique(osml_reps$blood_draw_date)
```

```
## [1] "2021-04-23" "2021-04-24" "2021-04-25" "2021-05-07" "2021-05-08"
## [6] "2021-07-14"
```

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, blood_draw_date) %>%
  mutate(count = n()) %>%
  dplyr::filter(count > 2) %>%
  arrange(count)
enuf_reps
```

```
## # A tibble: 252 x 6
## # Groups:   individual_ID, blood_draw_date [84]
##    blood_draw_date individual_ID replicate_no osmolality_mmol_kg hemolyzed_Y_N
##    <date>          <fct>         <fct>                     <dbl> <fct>
##  1 2021-04-23      M-02          3                           382 N
##  2 2021-04-23      M-02          2                           339 N
##  3 2021-04-23      M-02          1                           349 N
##  4 2021-04-23      M-06          3                           346 N
##  5 2021-04-23      M-06          2                           340 N
##  6 2021-04-23      M-06          1                           351 N
##  7 2021-04-24      M-10          3                           391 Y
##  8 2021-04-24      M-10          2                           417 Y
##  9 2021-04-24      M-10          1                           424 Y
## 10 2021-04-23      F-01          3                           337 N
## # ... with 242 more rows, and 1 more variable: count <int>
```

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

```
## # A tibble: 55 x 6
## # Groups:   individual_ID, blood_draw_date [32]
##    blood_draw_date individual_ID replicate_no osmolality_mmol_kg hemolyzed_Y_N
##    <date>          <fct>         <fct>                     <dbl> <fct>
##  1 2021-04-23      M-04          1                           349 N
##  2 2021-04-23      M-05          1                           348 N
##  3 2021-04-23      M-08          1                           396 N
##  4 2021-04-23      W-002         1                           360 <NA>
##  5 2021-04-23      W-005         1                           361 <NA>
##  6 2021-04-24      W-006         1                           334 Y
##  7 2021-04-24      W-007         1                           409 Y
##  8 2021-05-08      W-001         1                           390 N
##  9 2021-05-08      W-017         1                           366 N
## 10 2021-04-23      M-03          1                           372 N
## # ... with 45 more rows, and 1 more variable: count <int>
```

```
# check total obs still add to original 307
nrow(enuf_reps) + nrow(not_reps)
```

```
## [1] 307
```

```
nrow(enuf_reps) + nrow(not_reps) == nrow(osml_reps)
```

```
## [1] TRUE
```

Most of the blood samples had enough plasma for 3 reps :)

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.

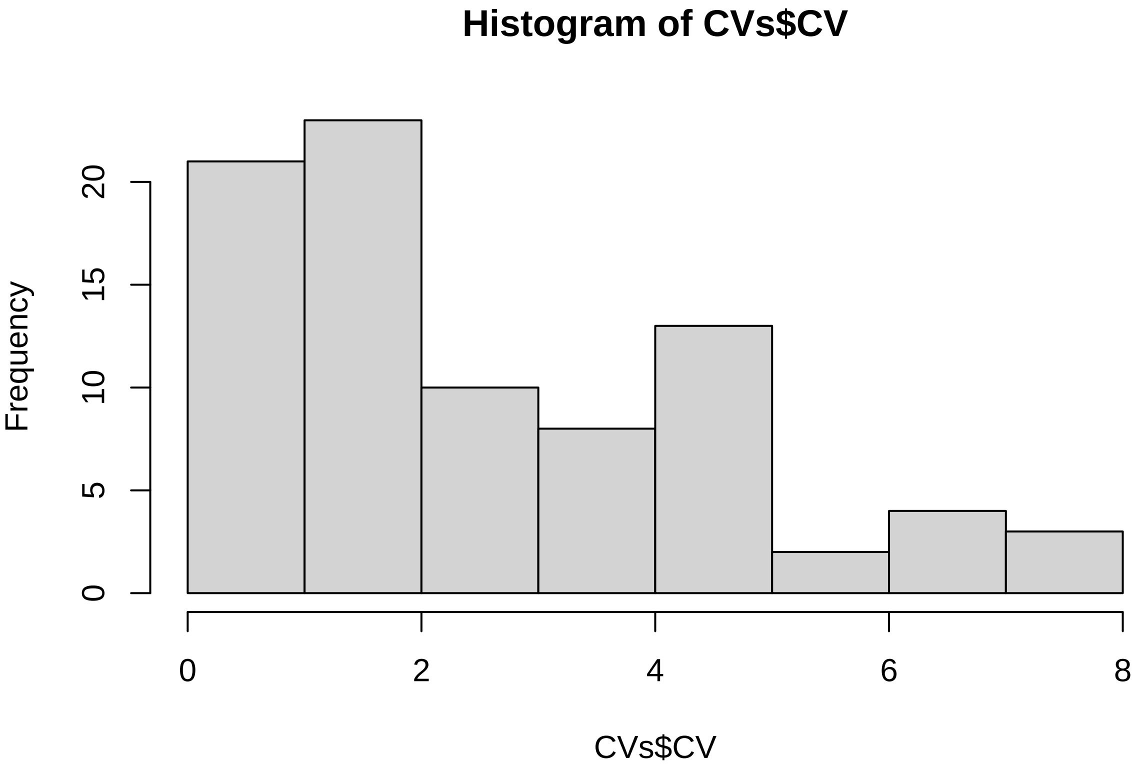
```
CVs <- enuf_reps %>%
  group_by(individual_ID, blood_draw_date) %>%
  summarise(mean = mean(osmolality_mmol_kg),
            SD = sd(osmolality_mmol_kg),
            CV = (SD/mean) *100,
            min = min(osmolality_mmol_kg),
            max = max(osmolality_mmol_kg),
            range = max - min
            )
```

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

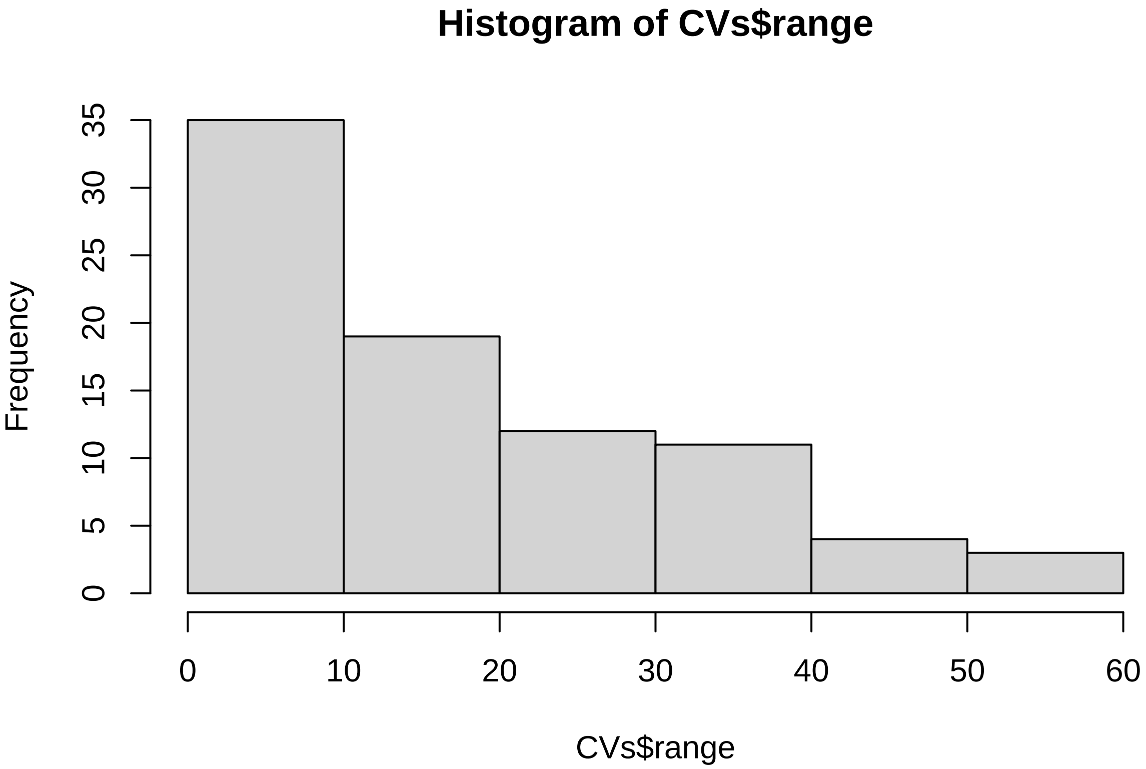
```
summary(CVs)
```

```
##  individual_ID blood_draw_date          mean             SD
##  F-12   : 3    Min.   :2021-04-23   Min.   :309.3   Min.   : 0.5774
##  M-19   : 3    1st Qu.:2021-04-24   1st Qu.:348.2   1st Qu.: 3.5119
##  F-01   : 2    Median :2021-04-25   Median :364.8   Median : 6.3705
##  F-03   : 2    Mean   :2021-05-09   Mean   :366.3   Mean   : 9.2897
##  F-06   : 2    3rd Qu.:2021-05-08   3rd Qu.:381.1   3rd Qu.:14.5373
##  F-10   : 2    Max.   :2021-07-14   Max.   :437.7   Max.   :29.8161
##  (Other):70
##        CV              min             max            range
##  Min.   :0.1701   Min.   :306.0   Min.   :313.0   Min.   : 1.00
##  1st Qu.:0.9814   1st Qu.:339.8   1st Qu.:358.0   1st Qu.: 7.00
##  Median :1.7355   Median :359.0   Median :372.5   Median :12.00
##  Mean   :2.5162   Mean   :358.1   Mean   :375.8   Mean   :17.69
##  3rd Qu.:4.1182   3rd Qu.:372.2   3rd Qu.:387.2   3rd Qu.:27.50
##  Max.   :7.6452   Max.   :434.0   Max.   :452.0   Max.   :58.00
##
```

```
hist(CVs$CV)
```



```
hist(CVs$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. CV values look really good, which is surprising because the range of values for some groups of replicates is 20-60 mmol/kg, which is very very high. So, I don't think we will find statistical outliers, but it's important to omit singular points increasing the range of their replicate set.

Find Outliers

```
# write function to find outliers for each individual on each date
find_outliers <- function(df) {

  # initiate dataframe to compile info and list to compile plots
  outliers <- data.frame()
  #boxplots <- list()

  # initiate a for loop to go through every who in df
  for(indiv_ch in unique(df$individual_ID)) {

    # select data for only the individual of interest
    df_sub <- df %>%
      dplyr::filter(individual_ID == as.character(indiv_ch))

    # make a boxplot
    df_sub %>%
      ggplot(.) +
      geom_boxplot(aes(x = as.factor(blood_draw_date),
                       y = osmolality_mmol_kg,
                       fill = as.factor(blood_draw_date))) +
      ggtitle(paste("Individual", indiv_ch)) +
      theme_classic() -> plot

    # print/save
    print(plot)
    #boxplots[[indiv_ch]] <- plot

    # extract outliers
    outs <- df_sub %>%
      group_by(individual_ID, blood_draw_date) %>%
      summarise(outs = boxplot.stats(osmolality_mmol_kg)$out)

    # add to running dataframe of outliers
    outliers <- outliers %>%
      rbind(outs)
  }
  #return(boxplots)
  return(outliers)
}
```

Now apply the function to the data:

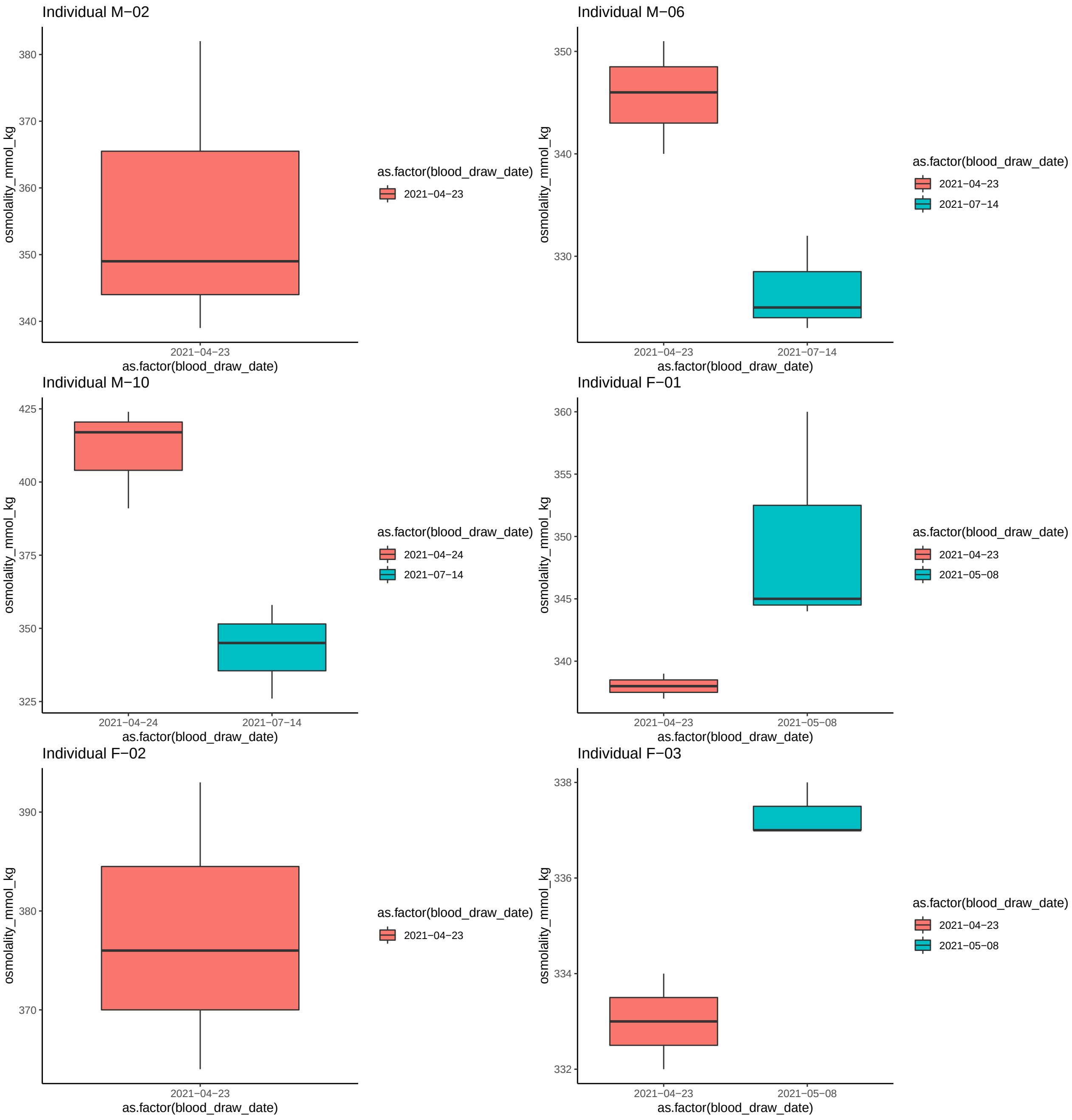
```
par(mfrow = c(71, 2))
outliers_found <- find_outliers(enuf_reps)
```

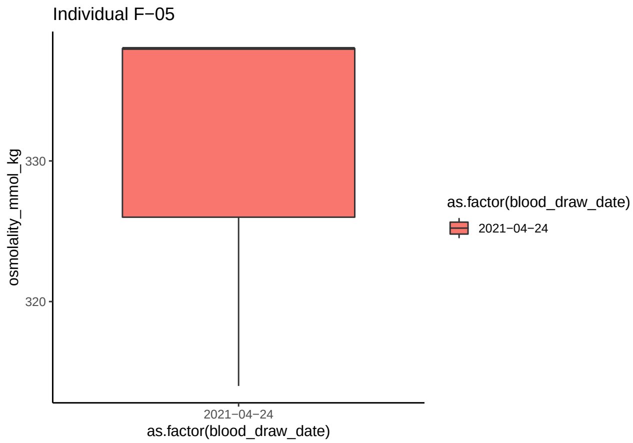
```
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
## `summarise()` regrouping output by 'individual_ID', 'blood_draw_date' (override with `.groups` argume
```

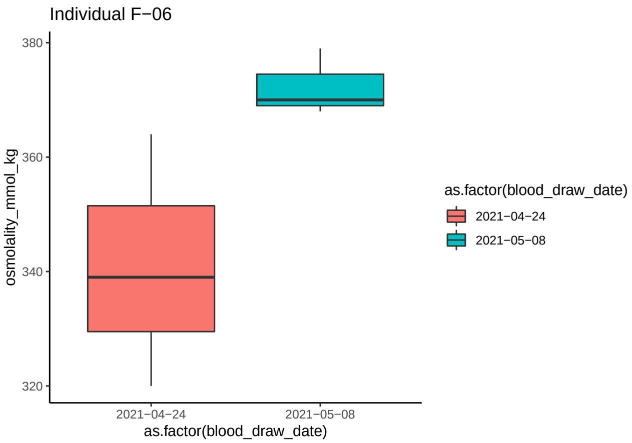
```
outliers_found
```

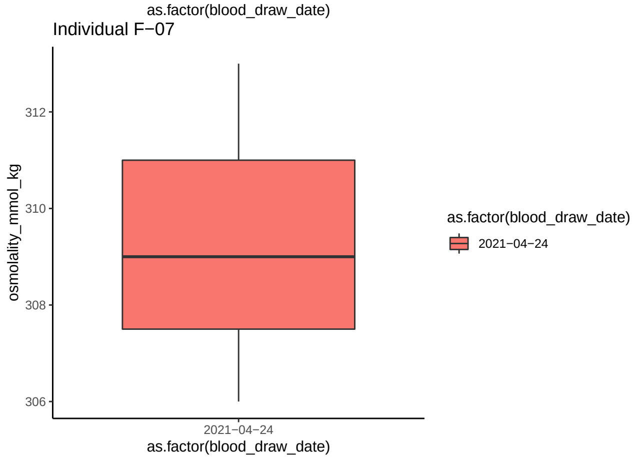
```
## # A tibble: 0 x 3
## # Groups:   individual_ID, blood_draw_date [0]
## # ... with 3 variables: individual_ID <fct>, blood_draw_date <date>, outs <dbl>
```

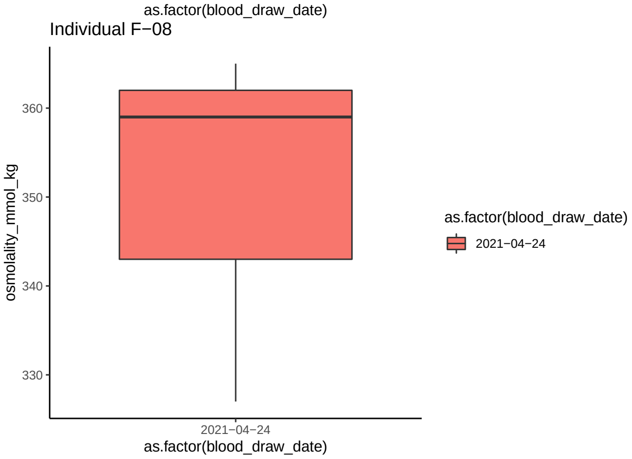
```
par(mfrow = c(1, 1))
```

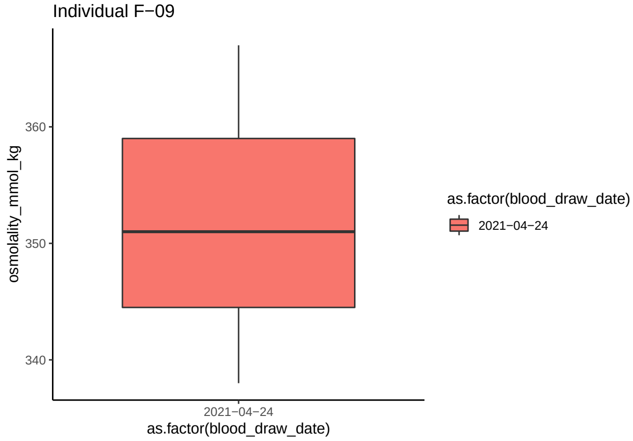

Individual M−02
osmolality_mmol_kg
380
370
360
350
340
2021−04−23
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23
Individual M−06
osmolality_mmol_kg
350
340
330
2021−04−23
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23
2021−07−14
Individual M−10
osmolality_mmol_kg
425
400
375
350
325
2021−04−24
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−24
2021−07−14
Individual F−01
osmolality_mmol_kg
360
355
350
345
340
2021−04−23
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23
2021−05−08
Individual F−02
osmolality_mmol_kg
390
380
370
2021−04−23
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23
Individual F−03
osmolality_mmol_kg
338
336
334
332
2021−04−23
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23
2021−05−08

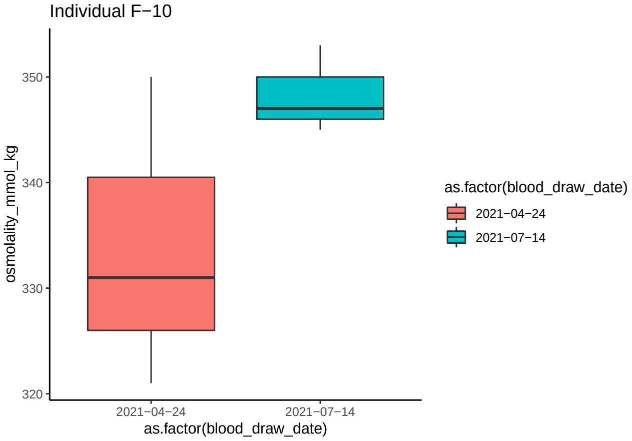

Individual F−05

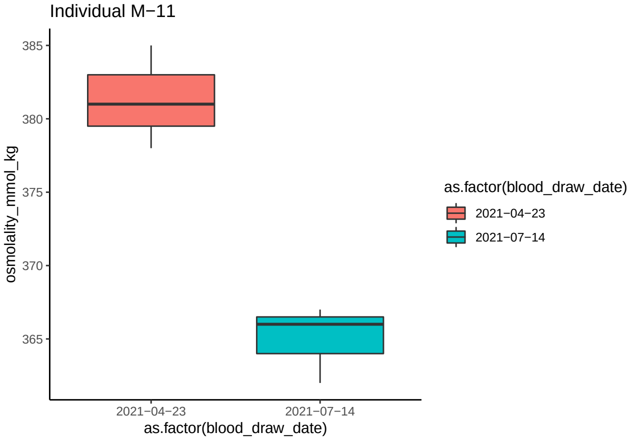

Individual F−06

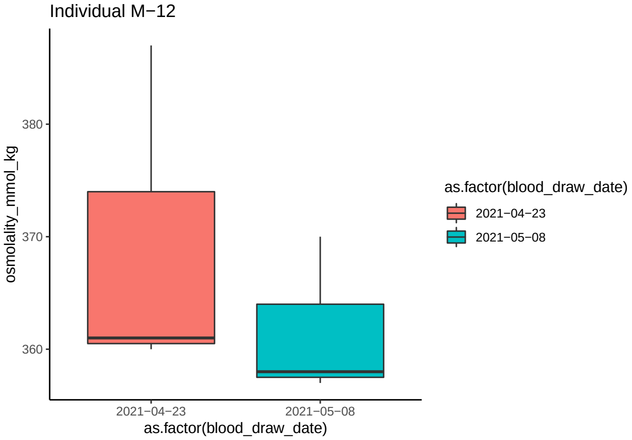

Individual F−07

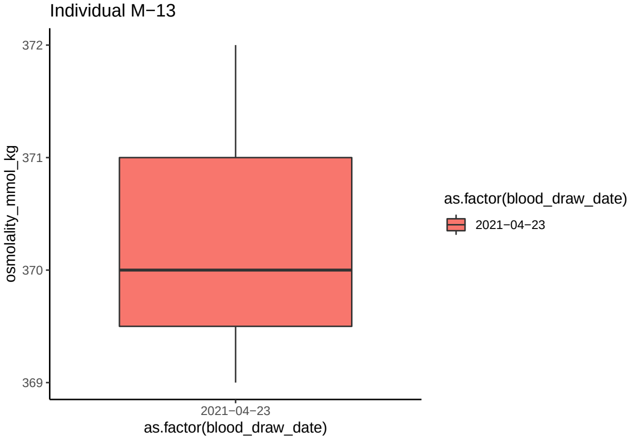

Individual F−08

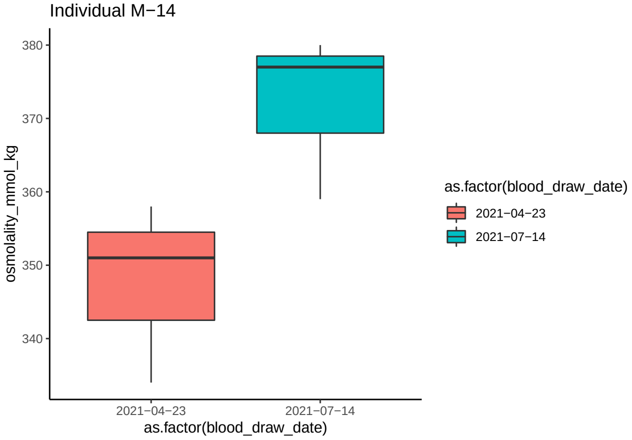

Individual F−09

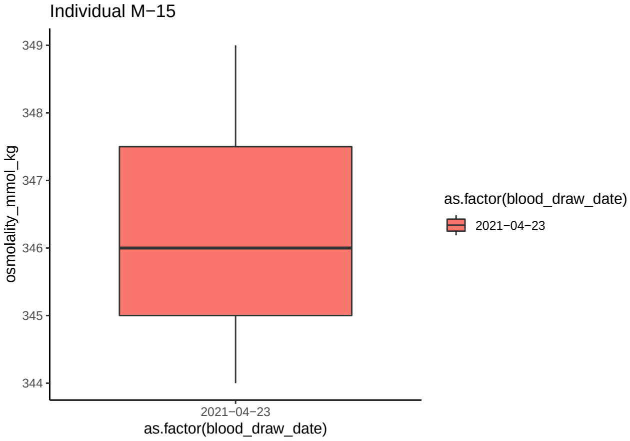

Individual F−10

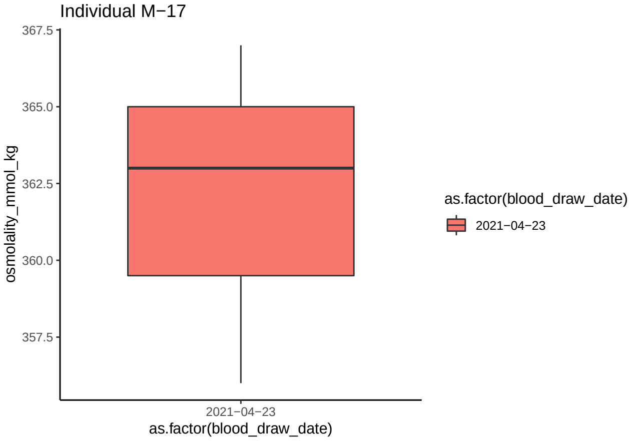

Individual M−11
osmolality_mmol_kg
385
380
375
370
365
2021−04−23
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23
2021−07−14

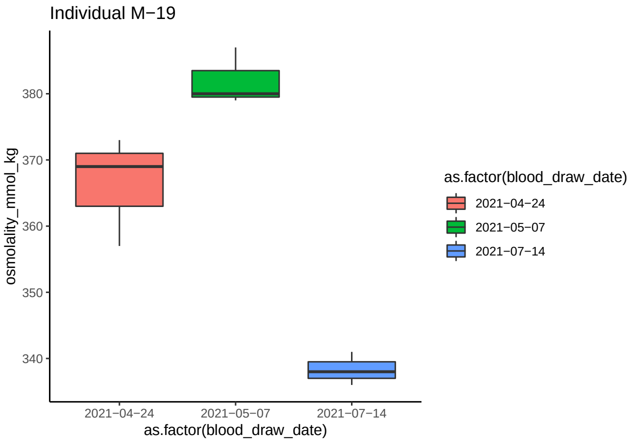

Individual M−12
osmolality_mmol_kg
380
370
360
2021−04−23
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23
2021−05−08

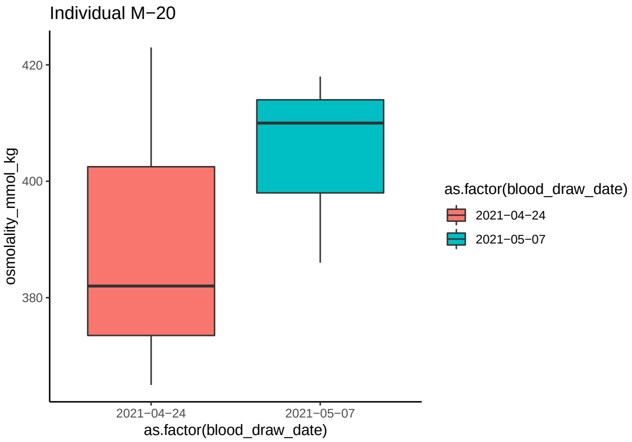

Individual M−13
osmolality_mmol_kg
372
371
370
369
2021−04−23
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23

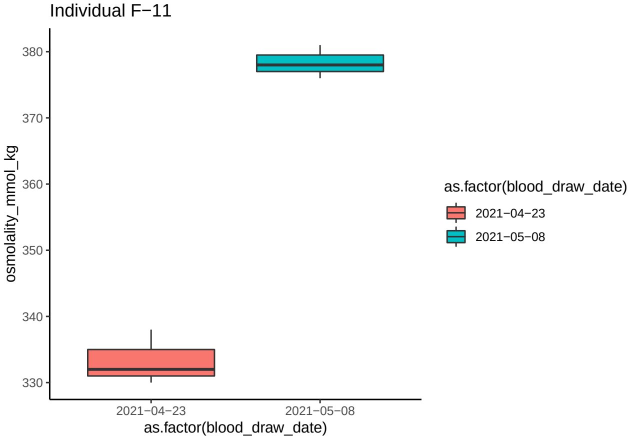

Individual M−14
osmolality_mmol_kg
380
370
360
350
340
2021−04−23
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23
2021−07−14

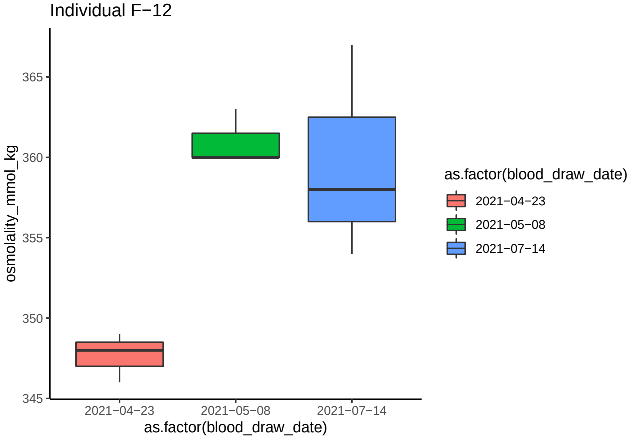

Individual M−15
osmolality_mmol_kg
349
348
347
346
345
344
2021−04−23
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23

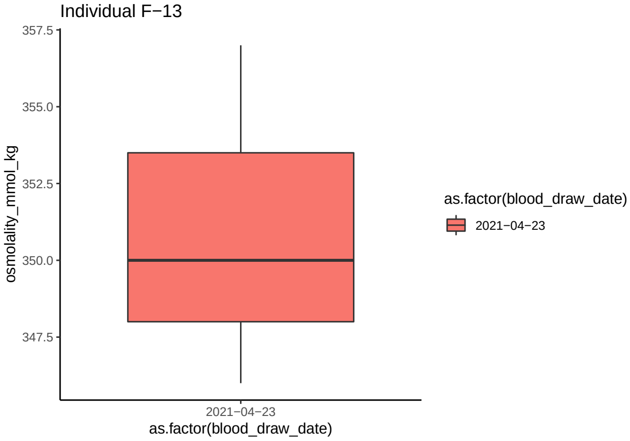

Individual M−17
osmolality_mmol_kg
367.5
365.0
362.5
360.0
357.5
2021−04−23
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23

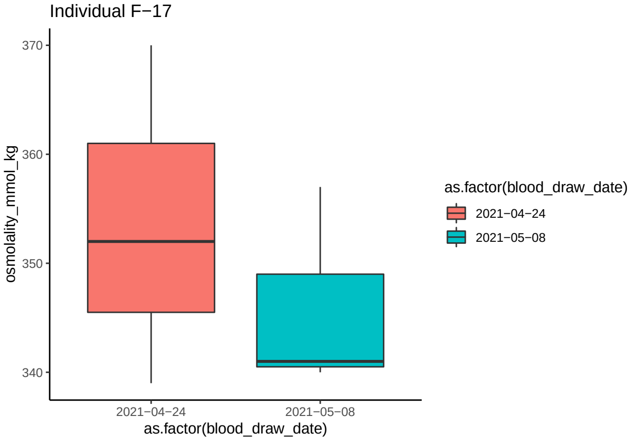

Individual M−19
osmolality_mmol_kg
380
370
360
350
340
2021−04−24
2021−05−07
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
🟥 2021−04−24
🟩 2021−05−07
🟦 2021−07−14

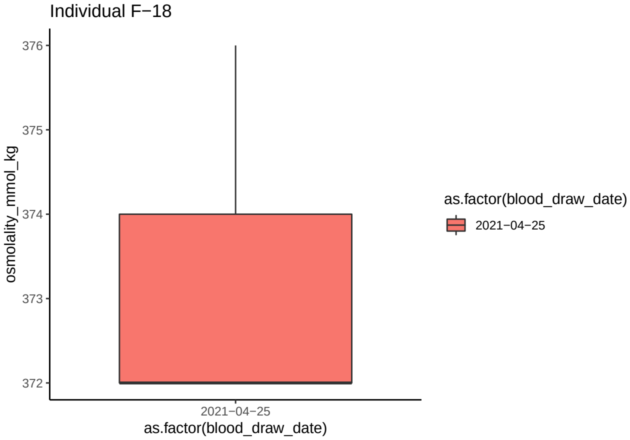

Individual M−20
osmolality_mmol_kg
420
400
380
2021−04−24
2021−05−07
as.factor(blood_draw_date)
as.factor(blood_draw_date)
🟥 2021−04−24
🟦 2021−05−07

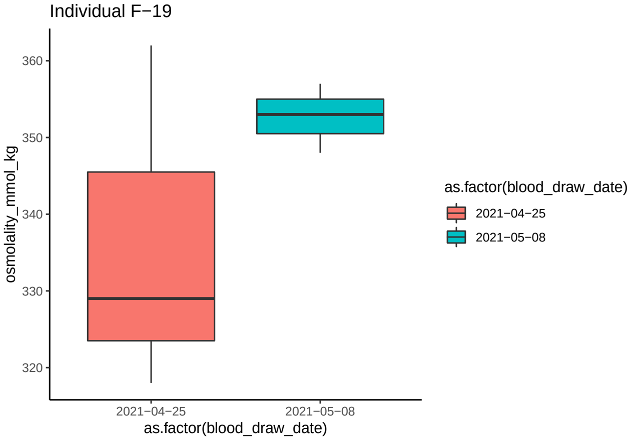

Individual F−11
osmolality_mmol_kg
380
370
360
350
340
330
2021−04−23
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
🟥 2021−04−23
🟦 2021−05−08

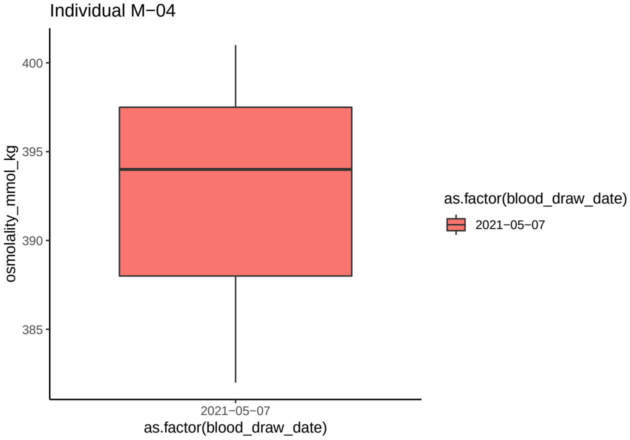

Individual F−12
osmolality_mmol_kg
365
360
355
350
345
2021−04−23
2021−05−08
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
🟥 2021−04−23
🟩 2021−05−08
🟦 2021−07−14

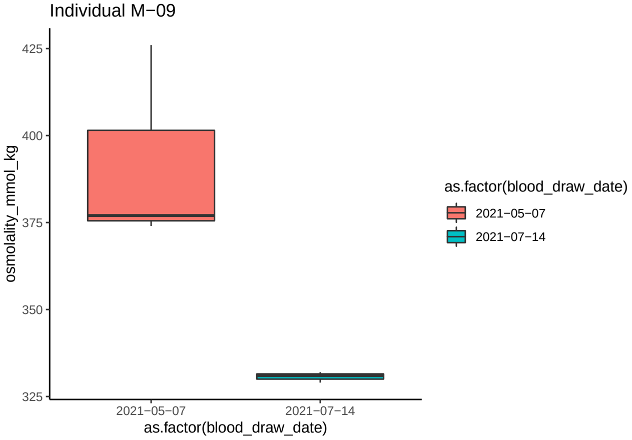

Individual F−13
osmolality_mmol_kg
357.5
355.0
352.5
350.0
347.5
2021−04−23
as.factor(blood_draw_date)
as.factor(blood_draw_date)
🟥 2021−04−23

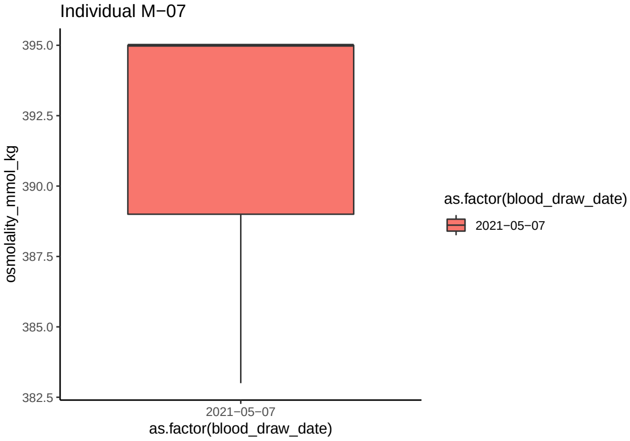

Individual F−17
osmolality_mmol_kg
370
360
350
340
2021−04−24
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
🟥 2021−04−24
🟦 2021−05−08

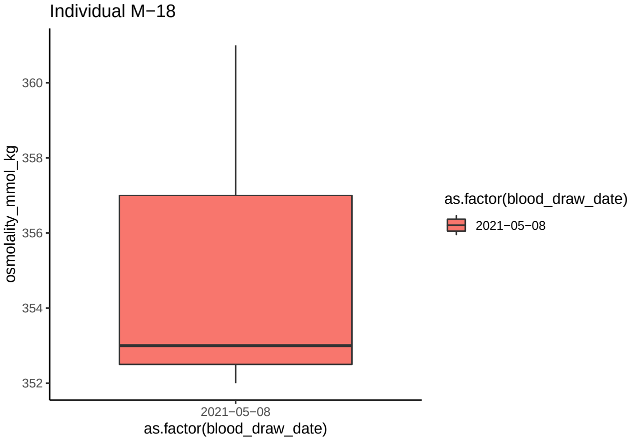

Individual F−18

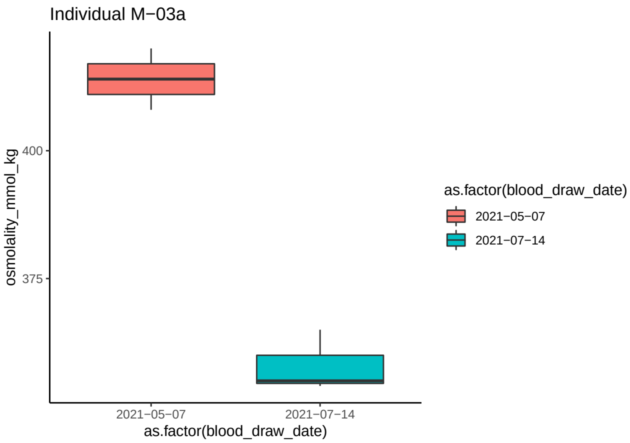

Individual F−19


Individual M−04

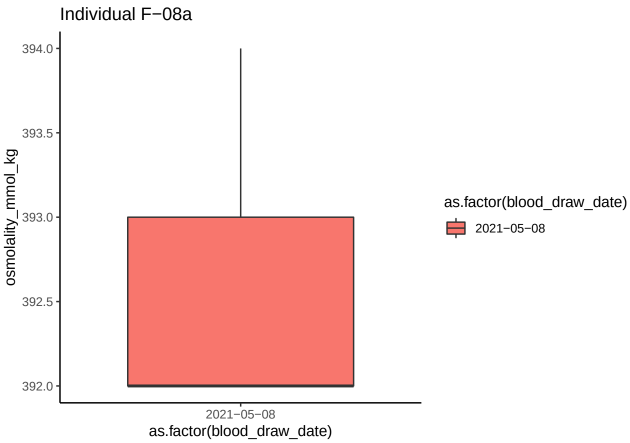

Individual M−09


Individual M−07

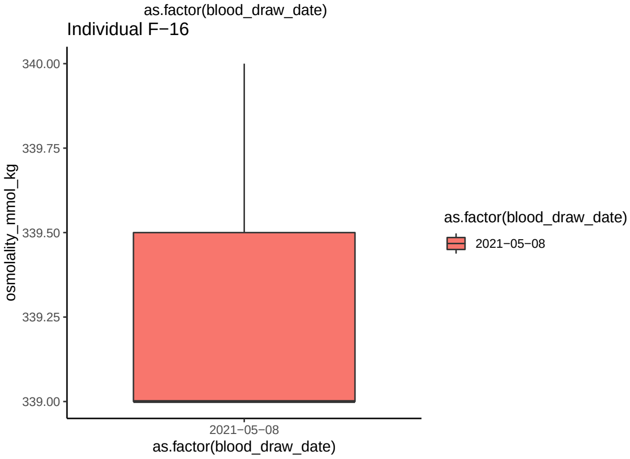

Individual M−18



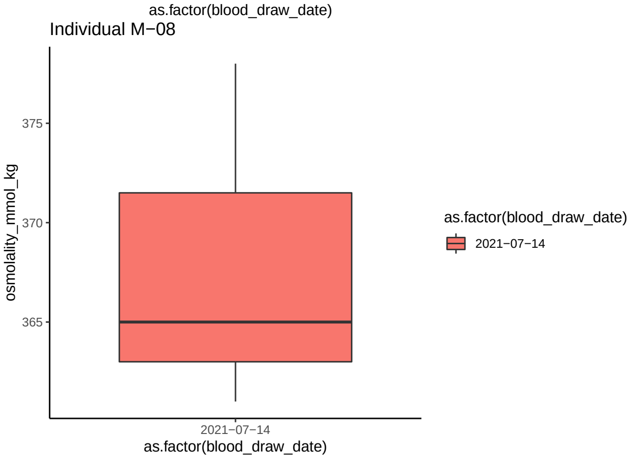
Individual M−03a
osmolality_mmol_kg
400
375
2021−05−07
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−05−07
2021−07−14



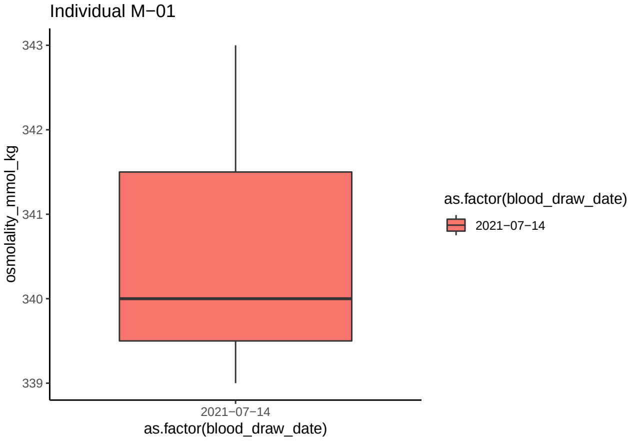
Individual F−08a
osmolality_mmol_kg
394.0
393.5
393.0
392.5
392.0
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−05−08



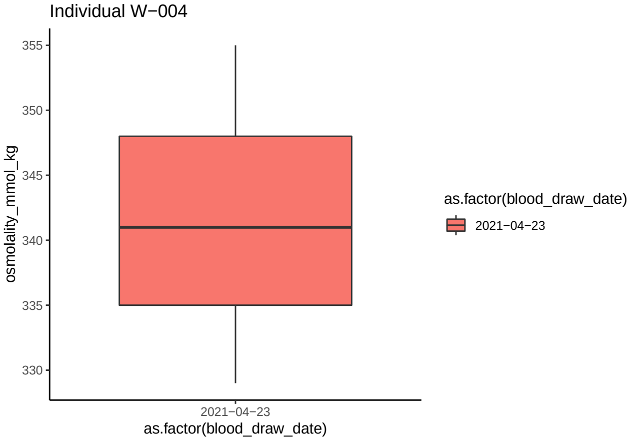
Individual F−16
osmolality_mmol_kg
340.00
339.75
339.50
339.25
339.00
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−05−08



Individual M−08
osmolality_mmol_kg
375
370
365
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−07−14

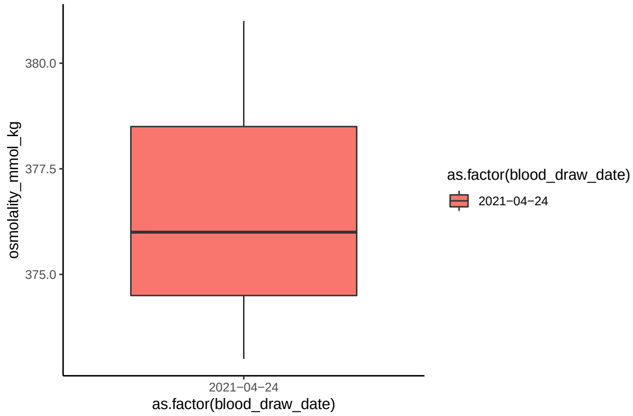


Individual M−01
osmolality_mmol_kg
343
342
341
340
339
2021−07−14
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−07−14

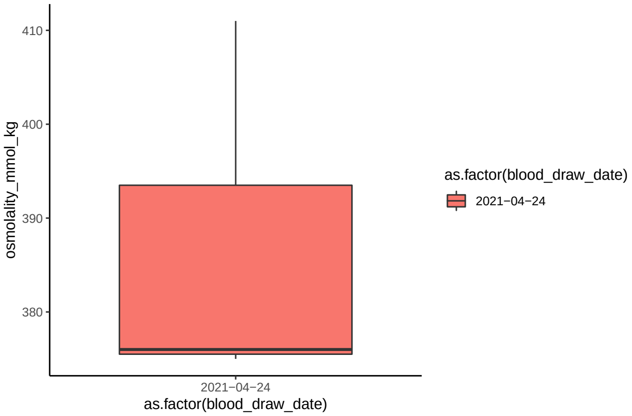


Individual W−004
osmolality_mmol_kg
355
350
345
340
335
330
2021−04−23
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−23

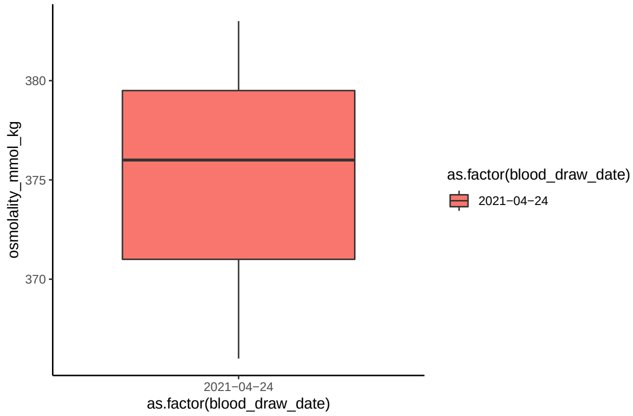
Individual W−009



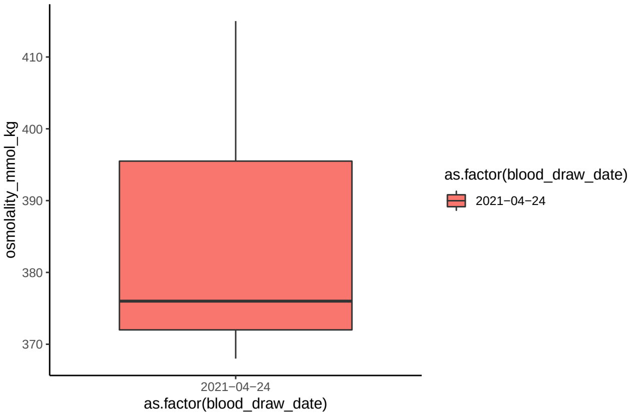
Individual W−010



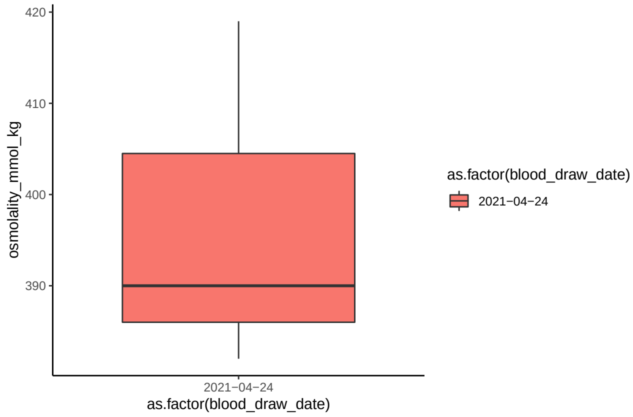
Individual W−011



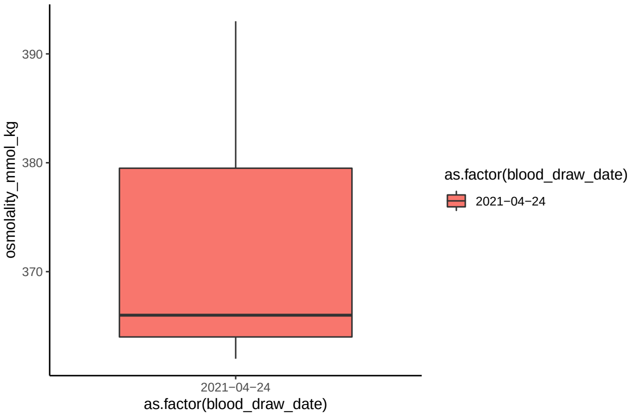
Individual W−012

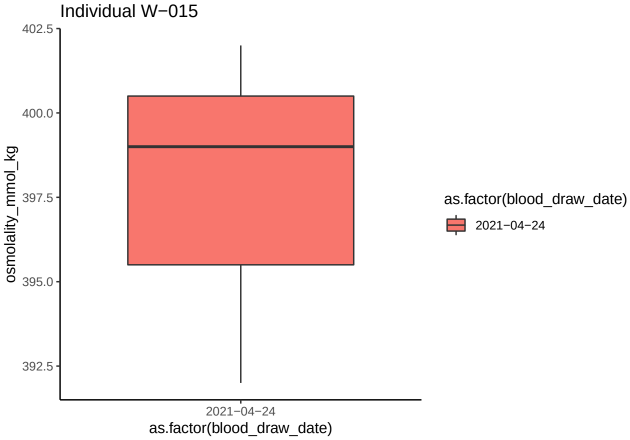


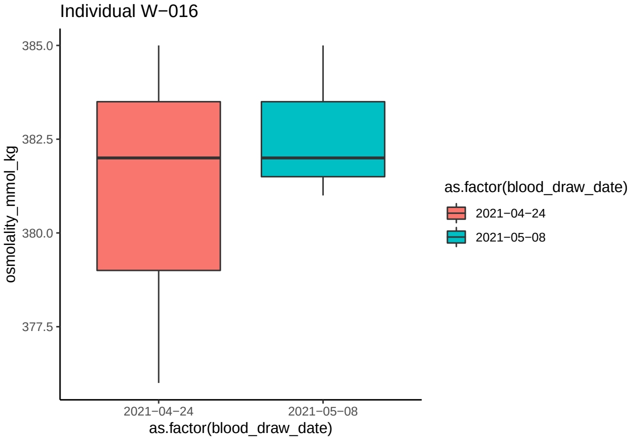
Individual W−013

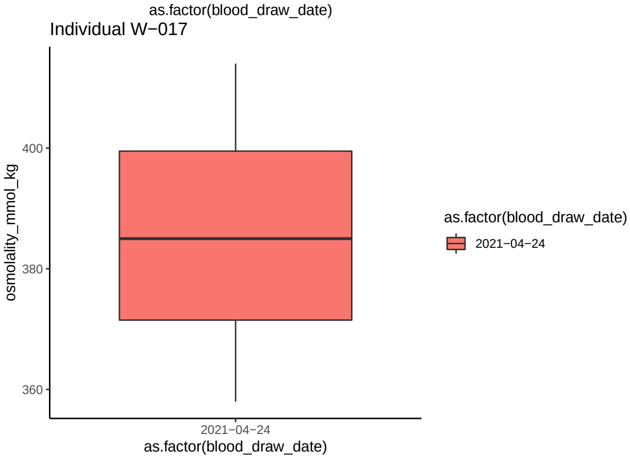


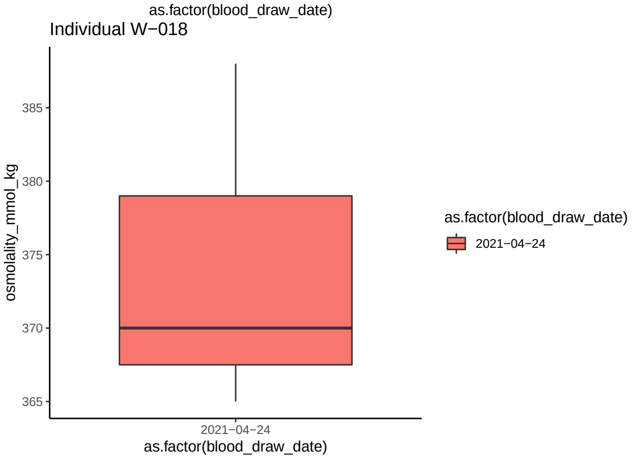
Individual W−014

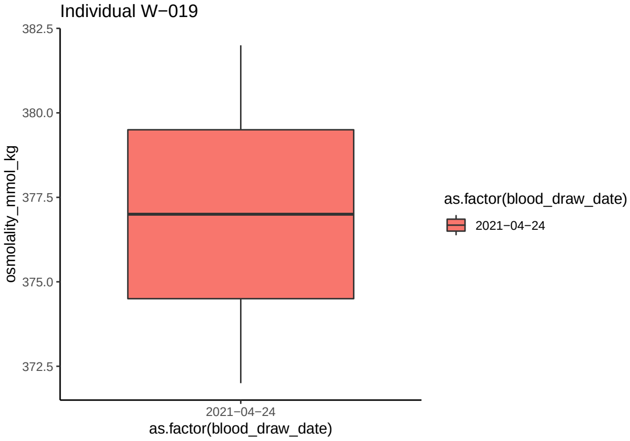


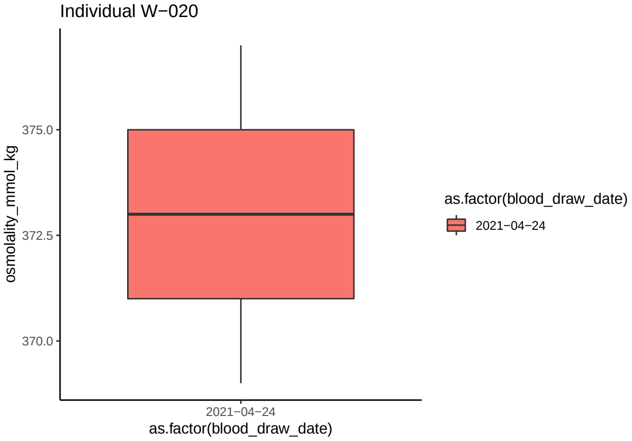

Individual W−015
402.5
400.0
397.5
395.0
392.5
osmolality_mmol_kg
2021−04−24
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−24


Individual W−016
385.0
382.5
380.0
377.5
osmolality_mmol_kg
2021−04−24
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−24
2021−05−08

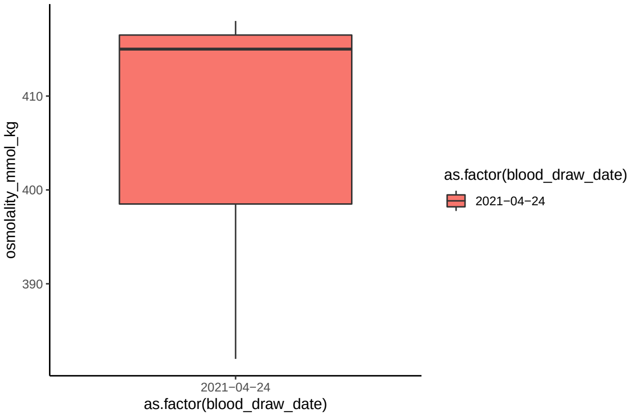

Individual W−017
400
380
360
osmolality_mmol_kg
2021−04−24
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−24


Individual W−018
385
380
375
370
365
osmolality_mmol_kg
2021−04−24
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−24

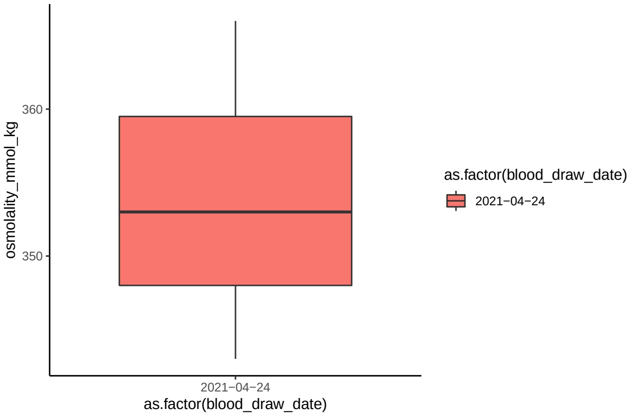

Individual W−019
382.5
380.0
377.5
375.0
372.5
osmolality_mmol_kg
2021−04−24
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−24


Individual W−020
375.0
372.5
370.0
osmolality_mmol_kg
2021−04−24
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−24

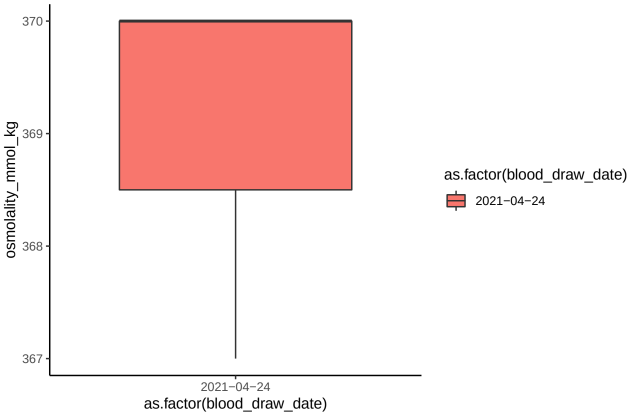
Individual W−021



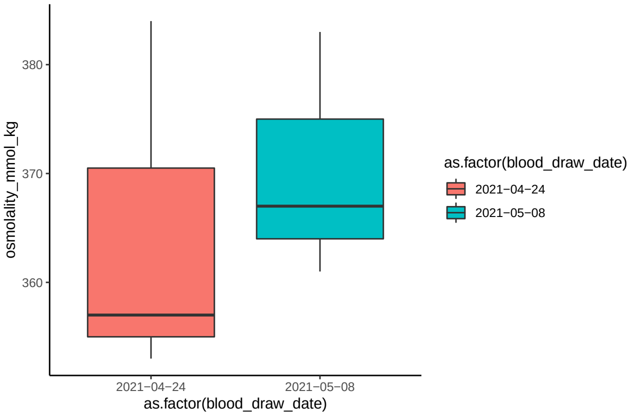
Individual W−022



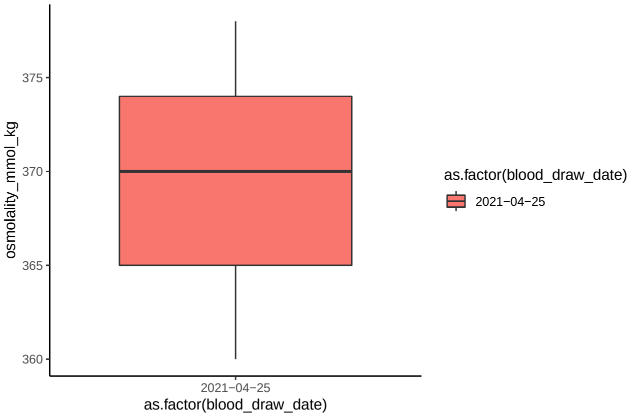
Individual W−023



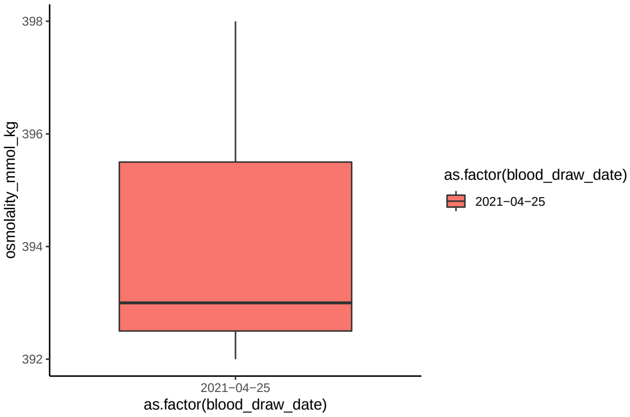
Individual W−025

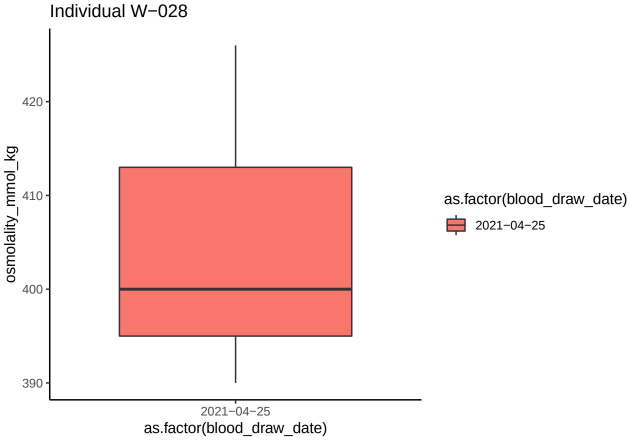


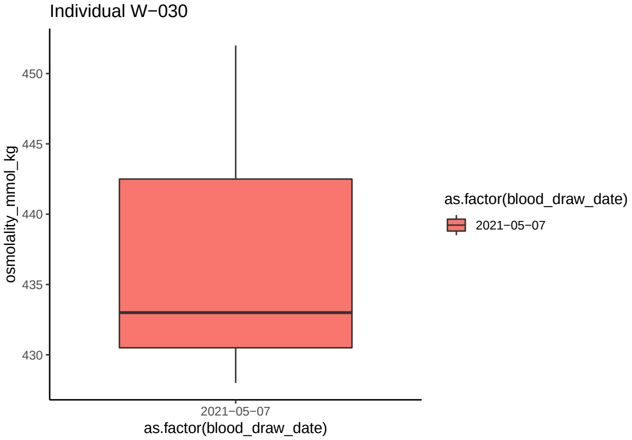
Individual W−026

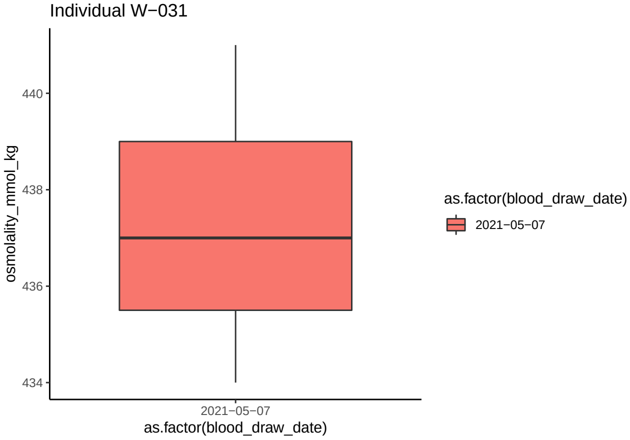


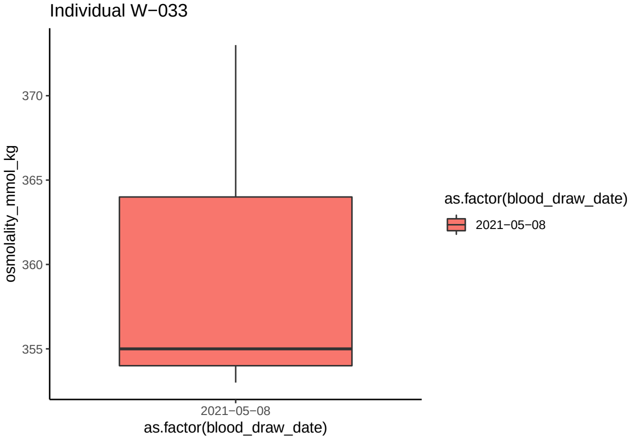
Individual W−027

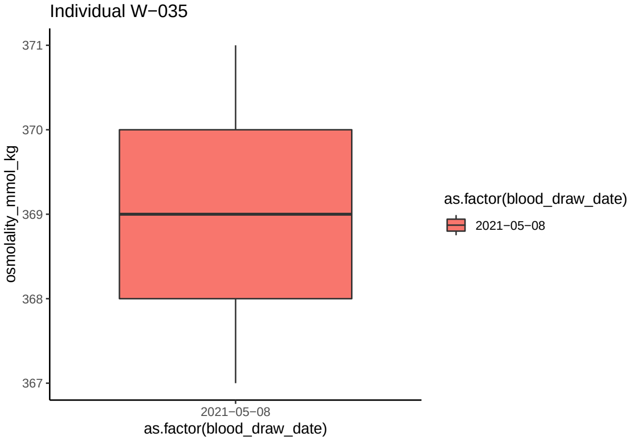


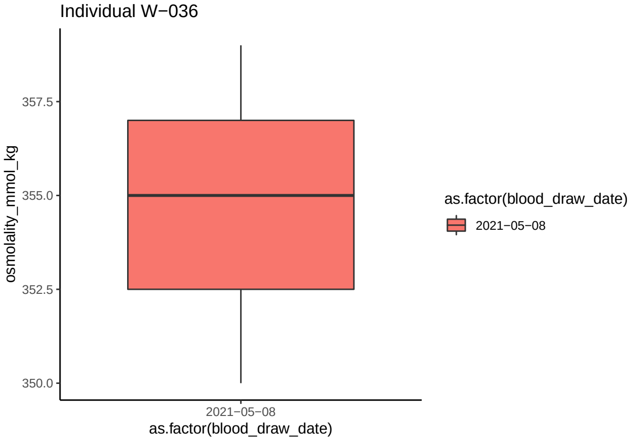

Individual W−028
osmolality_mmol_kg
420
410
400
390
2021−04−25
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−04−25

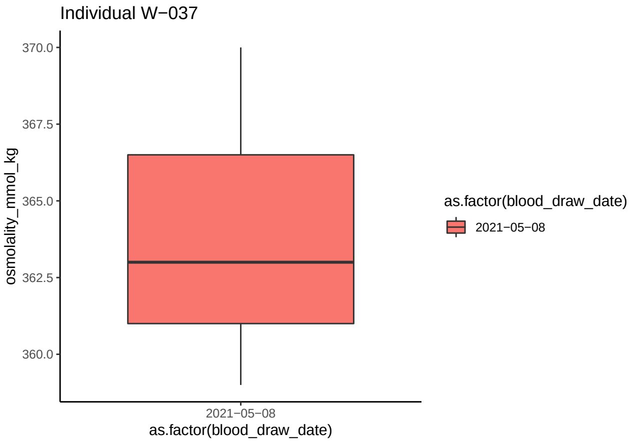

Individual W−030
osmolality_mmol_kg
450
445
440
435
430
2021−05−07
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−05−07

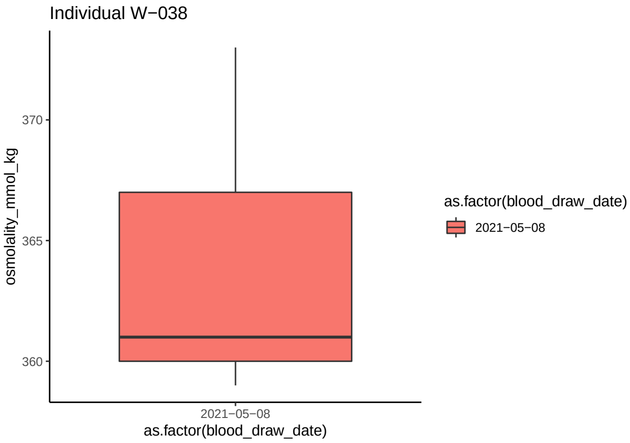

Individual W−031
osmolality_mmol_kg
440
438
436
434
2021−05−07
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−05−07

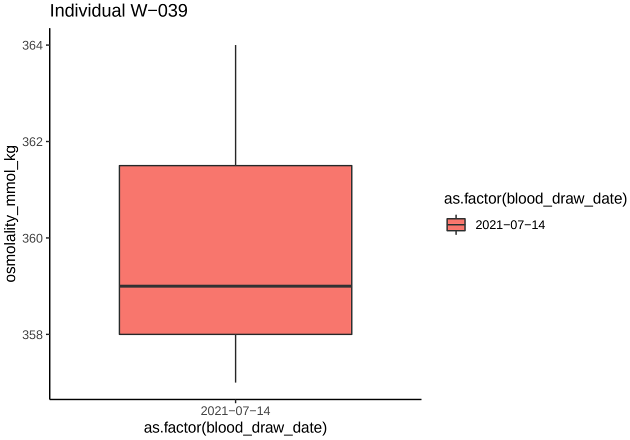

Individual W−033
osmolality_mmol_kg
370
365
360
355
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−05−08


Individual W−035
osmolality_mmol_kg
371
370
369
368
367
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−05−08


Individual W−036
osmolality_mmol_kg
357.5
355.0
352.5
350.0
2021−05−08
as.factor(blood_draw_date)
as.factor(blood_draw_date)
2021−05−08







As expected, based on boxplots, there are no outliers, but we still want to find and omit the points severely increasing the replicate ranges.

Determine which replicates lead to an increased CV… The osmometer is supposed to have measurement precision within a range of ±3 mmol/kg, so generously, we will investigate variability in replicate sets with a range of >10.

```
# first, look at just the replicate sets with very (badly) high value ranges
high_ranges <- enuf_reps %>%
  group_by(individual_ID, blood_draw_date) %>%
  dplyr::mutate(mean = mean(osmolality_mmol_kg),
                SD = sd(osmolality_mmol_kg),
                CV = (SD/mean) *100,
                min = min(osmolality_mmol_kg),
                max = max(osmolality_mmol_kg),
                range = max - min
            ) %>%
  dplyr::filter(range > 10) %>%
  dplyr::select(individual_ID, blood_draw_date, CV, osmolality_mmol_kg, replicate_no)

CV_12 <- high_ranges %>% # get CV for only reps 1&2
    dplyr::filter(replicate_no != 3) %>%
    group_by(individual_ID, blood_draw_date) %>%
    summarise(mean = mean(osmolality_mmol_kg),
              SD = sd(osmolality_mmol_kg),
              CV = (SD/mean) *100) %>%
    mutate(rep_excluded = "3") %>%
    dplyr::select(-mean, -SD)
```

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

```
CV_23 <- high_ranges %>% # get CV for only reps 2&3
    dplyr::filter(replicate_no != 1) %>%
    group_by(individual_ID, blood_draw_date) %>%
    summarise(mean = mean(osmolality_mmol_kg),
              SD = sd(osmolality_mmol_kg),
              CV = (SD/mean) *100) %>%
    mutate(rep_excluded = "1") %>%
    dplyr::select(-mean, -SD)
```

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

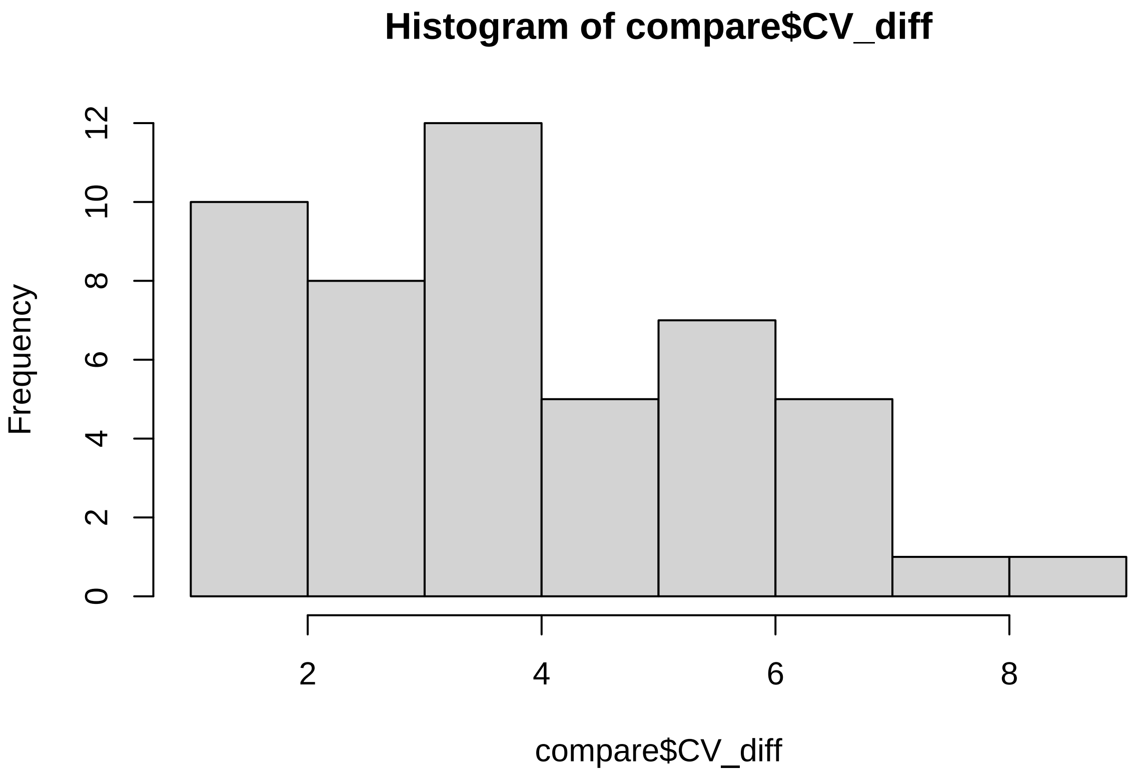
```
CV_31 <- high_ranges %>% # get CV for only reps 3&1
    dplyr::filter(replicate_no != 2) %>%
    group_by(individual_ID, blood_draw_date) %>%
    summarise(mean = mean(osmolality_mmol_kg),
              SD = sd(osmolality_mmol_kg),
              CV = (SD/mean) *100) %>%
    mutate(rep_excluded = "2") %>%
    dplyr::select(-mean, -SD)
```

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

```
# figure out what replicate inflates CV (and range)
compare <- high_ranges %>%
    dplyr::select(individual_ID, blood_draw_date, CV) %>%
    mutate(rep_excluded = "none") %>%
    rbind(CV_12) %>%
    rbind(CV_23) %>%
    rbind(CV_31) %>%
  group_by(individual_ID, blood_draw_date) %>%
  dplyr::mutate(min_CV = min(CV),
                max_CV = max(CV),
                CV_diff = max_CV - min_CV) %>%
  dplyr::filter(min_CV == CV)
compare
```

```
## # A tibble: 49 x 7
## # Groups:   individual_ID, blood_draw_date [49]
##    individual_ID blood_draw_date    CV rep_excluded min_CV max_CV CV_diff
##    <fct>         <date>          <dbl> <chr>         <dbl>  <dbl>   <dbl>
##  1 F-08          2021-04-24      1.17  3             1.17    7.77    6.59
##  2 F-09          2021-04-24      2.67  3             2.67    5.82    3.15
##  3 F-17          2021-04-24      2.66  3             2.66    6.18    3.52
##  4 M-02          2021-04-23      2.06  3             2.06    8.43    6.38
##  5 M-03a         2021-07-14      0.199 3             0.199   2.16    1.96
##  6 M-04          2021-05-07      1.25  3             1.25    3.43    2.19
##  7 M-09          2021-05-07      0.565 3             0.565   9.19    8.63
##  8 M-10          2021-04-24      1.18  3             1.18    5.73    4.55
##  9 M-14          2021-04-23      1.40  3             1.40    4.90    3.51
## 10 M-20          2021-05-07      1.37  3             1.37    5.63    4.26
## # ... with 39 more rows
```

```
hist(compare$CV_diff)
```



Remove Outliers

```
# need to save these or they get filtered out by default with the NAs
save <- enuf_reps %>%
  left_join(compare, by = c("individual_ID", "blood_draw_date")) %>%
  dplyr::filter(is.na(rep_excluded) == TRUE)

# remove "outlying" replicates
cleaned <- enuf_reps %>%
  left_join(compare, by = c("individual_ID", "blood_draw_date")) %>%
  dplyr::filter(replicate_no != rep_excluded)

# check number of data obs
nrow(enuf_reps) == (nrow(compare) + nrow(save) + nrow(cleaned))
```

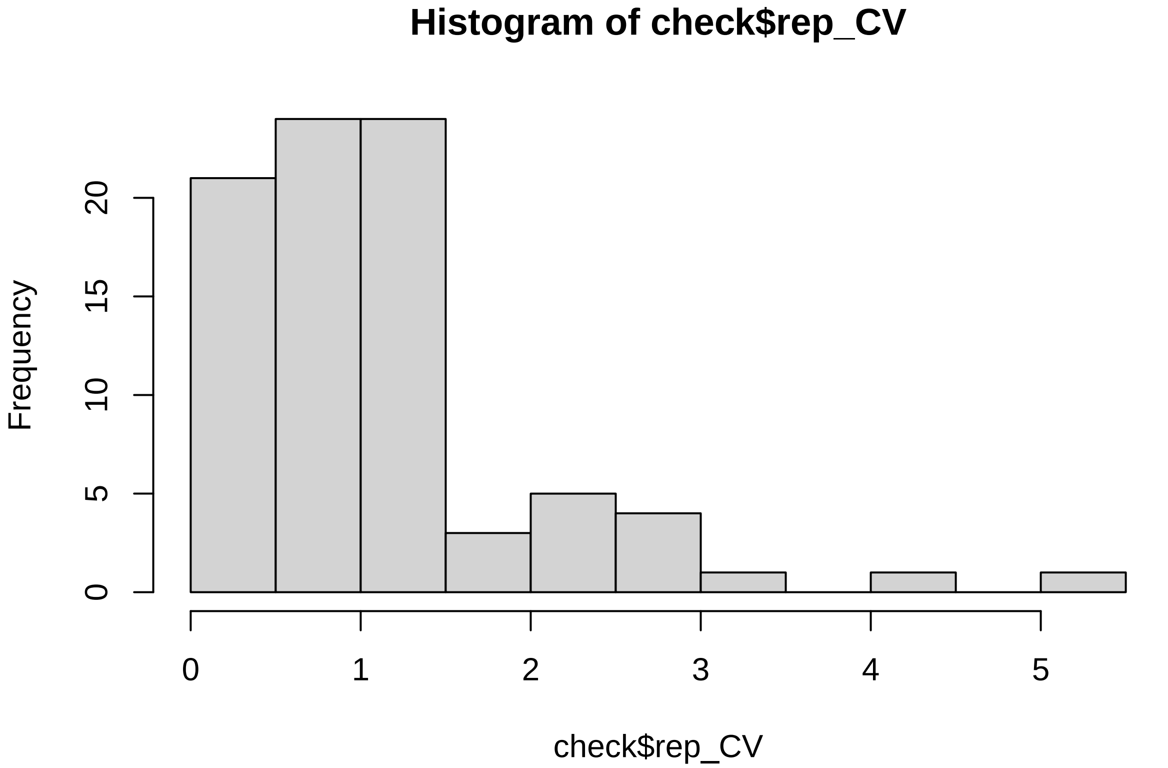
```
## [1] TRUE
```

```
# check that we improved things
check <- save %>%
  rbind(cleaned) %>%
  group_by(individual_ID, blood_draw_date) %>%
  summarise(osmolality_mmol_kg_mean = mean(osmolality_mmol_kg),
            rep_SD = sd(osmolality_mmol_kg),
            rep_CV = (rep_SD/osmolality_mmol_kg_mean) *100,
            rep_min = min(osmolality_mmol_kg),
            rep_max = max(osmolality_mmol_kg),
            rep_range = rep_max - rep_min
```

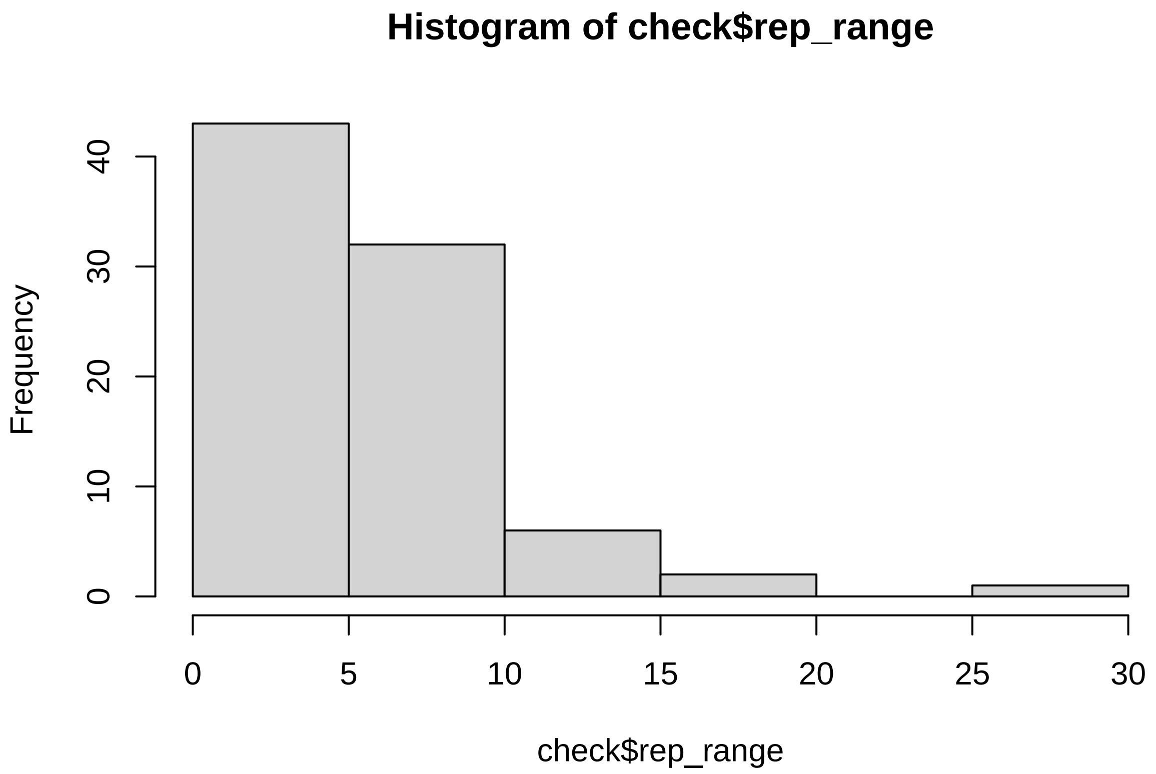
```
             )
```

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

```
hist(check$rep_CV)
```



```
hist(check$rep_range)
```



Much, MUCH better! :D

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 and check that it meaningfully improved things.

```
osml_means <- not_reps %>%
  rbind(save) %>%
  rbind(cleaned) %>%
  group_by(individual_ID, blood_draw_date) %>%
  summarise(osmolality_mmol_kg_mean = mean(osmolality_mmol_kg))
```

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

Export

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