NMR analysis

Lauren Gill

10/01/2024

Load in libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(mdatools)
library(ggpubr)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:mdatools':
##
##     ellipse
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(emmeans)
library(cowplot)
```

```
##
```

```
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:ggpubr':
##
##     get_legend
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(multcompView)
library(agricolae)
library(openxlsx)
```

```
## Warning: package 'openxlsx' was built under R version 4.3.3
```

Load in data The data is the metabolites already normalized by wetweight

```
concentrations <- read_csv("../data/concentrations_names.csv")
```

```
## Rows: 36 Columns: 49
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): sample, group
## dbl (47): 1-Methylhistidine, 2-Hydroxybutyric acid, Acetic acid, Betaine, Ac...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
slug_weights <- read_csv("../data/slug_weights.csv")
```

```
## Rows: 74 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): slug_number
## dbl (2): dry_weight_g, wet_weight_g
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Transform data

First making the data into um intstead of uM The slugs were placed in 500uL (500/10^6 L) of acetonitrile so concentrations are relative to this

Concentrations in the original table are in uM uM = umol/L umol = uM * L

```
## renaming the sample column in slug_weights
slug_weights <- slug_weights %>%
  rename(sample = slug_number)
```

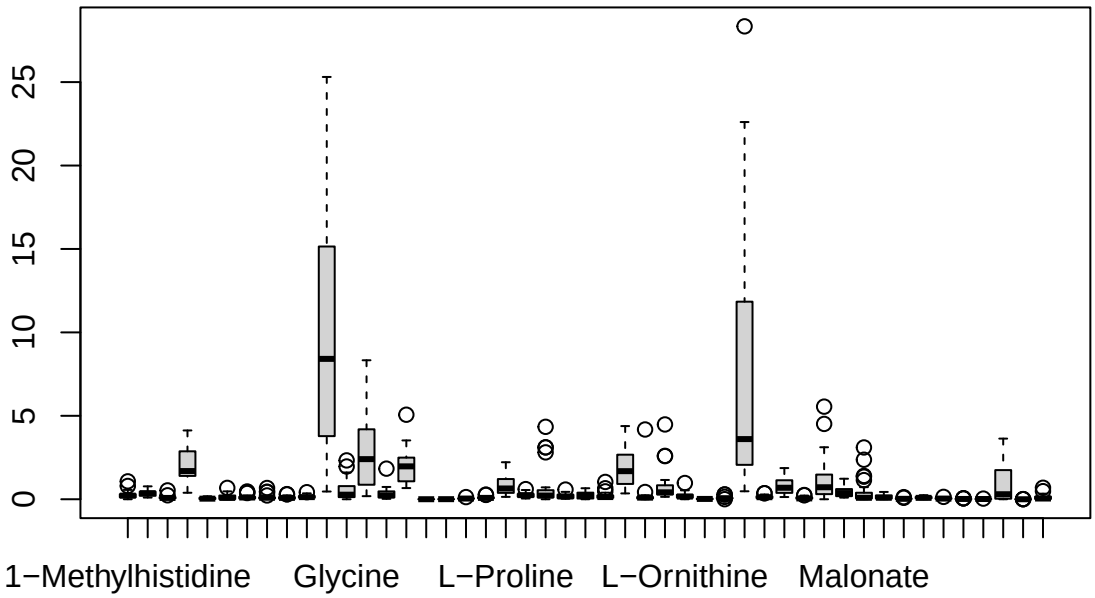
```
nmr <- right_join(slug_weights, concentrations, by = "sample")%>%
  #converting to umoles (see notes above)
  mutate(across(`1-Methylhistidine`:`Dimethyl sulfone`,
                ~ .*(500/(10^6)))) %>%
  #dividing each conc in each row by the dry weight (in g)
  mutate(across(`1-Methylhistidine`:`Dimethyl sulfone`,
                ~ ./wet_weight_g)) %>%
  #removing extrenuous colomns (for metaboanalyst)
  dplyr::select(-c(wet_weight_g, dry_weight_g))

#write_csv(nmr_wet, "new_wet_conc_moles.csv")
```

Final units: umol metabolite/g ww

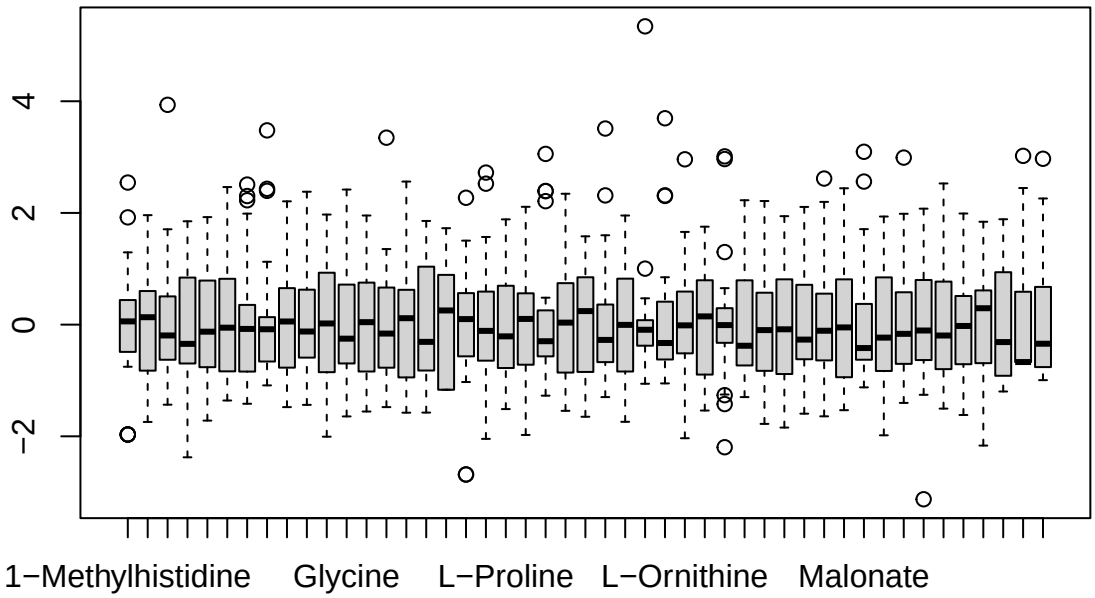
```
nmr_transformed <- nmr %>%
  #sqrt tranforming the data
  mutate(across(!c(sample, group), sqrt)) %>%
  #deleting the first two columns because "prep.autoscale" doesnt use them
  dplyr::select(!c(sample, group))

#Now autoscaling the data
nmr_scaled <- prep.autoscale(nmr_transformed, center = TRUE, scale = TRUE)

#Adding back in the first two columns
first_columns <- nmr %>%
  dplyr::select(c(sample, group))
nmr_normal <- cbind(first_columns, nmr_scaled)
```

Check out this transformed data

```
#Original NMR data (big variances)
nmr_original <- nmr %>%
  dplyr::select(!c(sample, group))
boxplot(nmr_original)
```




```
#the transformed data, looks a lot better!
boxplot(nmr_scaled)
```



ANOVA

Trying a two way anova (https://haowang47.github.io/files/2023-01-31-2wayANOVA.html)

Lets make a function Input = dataname, name of metabolite

```
metaboanova <- function(data, metabolite){
#first making a data table with just metabo concentrations

glut <- data %>%
  dplyr::select(sample, group, {{metabolite}}) %>%
  rename(conc = {{metabolite}}) %>%
  #seperating SD20 into SD and 20
  tidyr::separate(group,
                  into = c("day_length", "temp"),
                  sep = "(?<=[A-Za-z])(?=[0-9])")

glut_anova <- lm(conc ~ day_length*temp, data = glut,
                   contrasts = list(temp="contr.sum", day_length="contr.sum"))


aov(glut_anova)
}
```

```
metaboletters <- function(data, metabolite){
#first making a data table with just metabo concentrations

glut <- data %>%
  dplyr::select(sample, group, {{metabolite}}) %>%
  rename(conc = {{metabolite}}) %>%
  #seperating SD20 into SD and 20
  tidyr::separate(group,
           into = c("day_length", "temp"),
           sep = "(?<=[A-Za-z])(?=[0-9])")

glut_anova <- lm(conc ~ day_length*temp, data = glut,
                  contrasts = list(temp="contr.sum", day_length="contr.sum"))


multcompLetters4(glut_anova, TukeyHSD(aov(glut_anova)), reversed = TRUE)
}
```

Getting letters to put on the plot

```
th_letter <- metaboletters(nmr_normal, `L-Threonine`)
th_letter_1 <- th_letter$`day_length:temp`
th_letter_1
```

```
## LD:15 LD:20 SD:20 SD:15
##   "b"   "a"   "a"   "a"
```

```
metaboletters(nmr_normal, `L-Glutamine`)
```

```
## $day_length
##  LD  SD
## "b" "a"
##
## $temp
##  20  15
## "b" "a"
##
## $`day_length:temp`
## LD:20 SD:20 LD:15 SD:15
##   "b"   "b"  "ab"   "a"
```

```
metaboletters(nmr_normal, `Formate`)
```

```
## $day_length
## $day_length$Letters
##  LD  SD
## "a" "a"
##
## $day_length$LetterMatrix
##       a
## LD TRUE
```

```
## SD TRUE
##
##
## $temp
##  20  15
## "b" "a"
##
## $`day_length:temp`
## SD:20 LD:20 LD:15 SD:15
##   "b"  "ab"  "ab"   "a"
```

```
th <- metaboanova(nmr_normal, `L-Threonine`)
summary(th)
```

```
##                 Df Sum Sq Mean Sq F value  Pr(>F)
## day_length       1  7.769   7.769  12.140 0.00145 **
## temp             1  0.469   0.469   0.733 0.39817
## day_length:temp  1  6.285   6.285   9.822 0.00368 **
## Residuals       32 20.477   0.640
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
th_tukey <- TukeyHSD(th)
letters <- multcompLetters4(th, TukeyHSD(aov(th)), reversed = TRUE)
th_tukey
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = glut_anova)
##
## $day_length
##             diff       lwr        upr     p adj
## SD-LD -0.9290788 -1.472222 -0.3859355 0.0014531
##
## $temp
##             diff        lwr      upr     p adj
## 20-15 -0.2226362 -0.7657794 0.320507 0.4099388
##
## $`day_length:temp`
##                    diff        lwr         upr     p adj
## SD:15-LD:15 -1.8382162 -2.8861086 -0.79032386 0.0002285
## LD:20-LD:15 -1.0912944 -2.1391867 -0.04340199 0.0387015
## SD:20-LD:15 -1.2153253 -2.3738139 -0.05683671 0.0368870
## LD:20-SD:15  0.7469219 -0.1772323  1.67107608 0.1478120
## SD:20-SD:15  0.6228909 -0.4250015  1.67078329 0.3872466
## SD:20-LD:20 -0.1240310 -1.1719233  0.92386142 0.9883773
```

```
glut <- metaboanova(nmr_normal, `L-Glutamine`)
summary(glut)
```

```
##                 Df Sum Sq Mean Sq F value   Pr(>F)
```

```
## day_length       1  4.083   4.083   6.304 0.017297 *
## temp             1  9.502   9.502  14.670 0.000563 ***
## day_length:temp  1  0.690   0.690   1.065 0.309896
## Residuals       32 20.726   0.648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
form <- metaboanova(nmr_normal, `Formate`)
summary(form)
```

```
##                 Df Sum Sq Mean Sq F value  Pr(>F)
## day_length       1  0.176   0.176   0.212 0.64794
## temp             1  7.738   7.738   9.317 0.00454 **
## day_length:temp  1  0.507   0.507   0.611 0.44027
## Residuals       32 26.578   0.831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Trying to add p values

```
nmr_summarized <-  nmr %>%
  dplyr::select(sample, group, Formate, `L-Threonine`, `L-Glutamine`)%>%
  pivot_longer(!c(sample, group), names_to = "metabolite", values_to = "conc") %>%
  group_by(group) %>%
  summarize(max_conc=max(conc))

hsd <- HSD.test(th, "day_length", group=T)
```

Mean Comparisons

Checking simple effects #### Glutamine

```
#first making a data table with just metabo concentrations
glut <- nmr_normal %>%
  dplyr::select(sample, group, `L-Glutamine`) %>%
  rename(conc = `L-Glutamine`) %>%
  #seperating SD20 into SD and 20
  tidyr::separate(group,
           into = c("day_length", "temp"),
           sep = "(?<=[A-Za-z])(?=[0-9])")

glut_anova <- lm(conc ~ day_length*temp, data = glut,
                 contrasts = list(temp="contr.sum",
                                  day_length="contr.sum"))

#by day length
tempbydaylength <- emmeans(glut_anova, ~temp|day_length)
tempbydaylength
```

```
## day_length = LD:
##  temp emmean    SE df lower.CL upper.CL
```

```
##  15   -0.134 0.304 32   -0.753    0.486
##  20    0.636 0.243 32    0.142    1.130
##
## day_length = SD:
##  temp emmean    SE df lower.CL upper.CL
##  15   -0.857 0.243 32   -1.351   -0.363
##  20    0.481 0.304 32   -0.139    1.100
##
## Confidence level used: 0.95
```

```
pairs(tempbydaylength) #signficant!
```

```
## day_length = LD:
##  contrast        estimate    SE df t.ratio p.value
##  temp15 - temp20    -0.77 0.389 32  -1.979  0.0565
##
## day_length = SD:
##  contrast        estimate    SE df t.ratio p.value
##  temp15 - temp20    -1.34 0.389 32  -3.438  0.0016
```

```
#by temp
daylengthbytemp <- emmeans(glut_anova, ~day_length|temp)
daylengthbytemp
```

```
## temp = 15:
##  day_length emmean    SE df lower.CL upper.CL
##  LD         -0.134 0.304 32   -0.753    0.486
##  SD         -0.857 0.243 32   -1.351   -0.363
##
## temp = 20:
##  day_length emmean    SE df lower.CL upper.CL
##  LD          0.636 0.243 32    0.142    1.130
##  SD          0.481 0.304 32   -0.139    1.100
##
## Confidence level used: 0.95
```

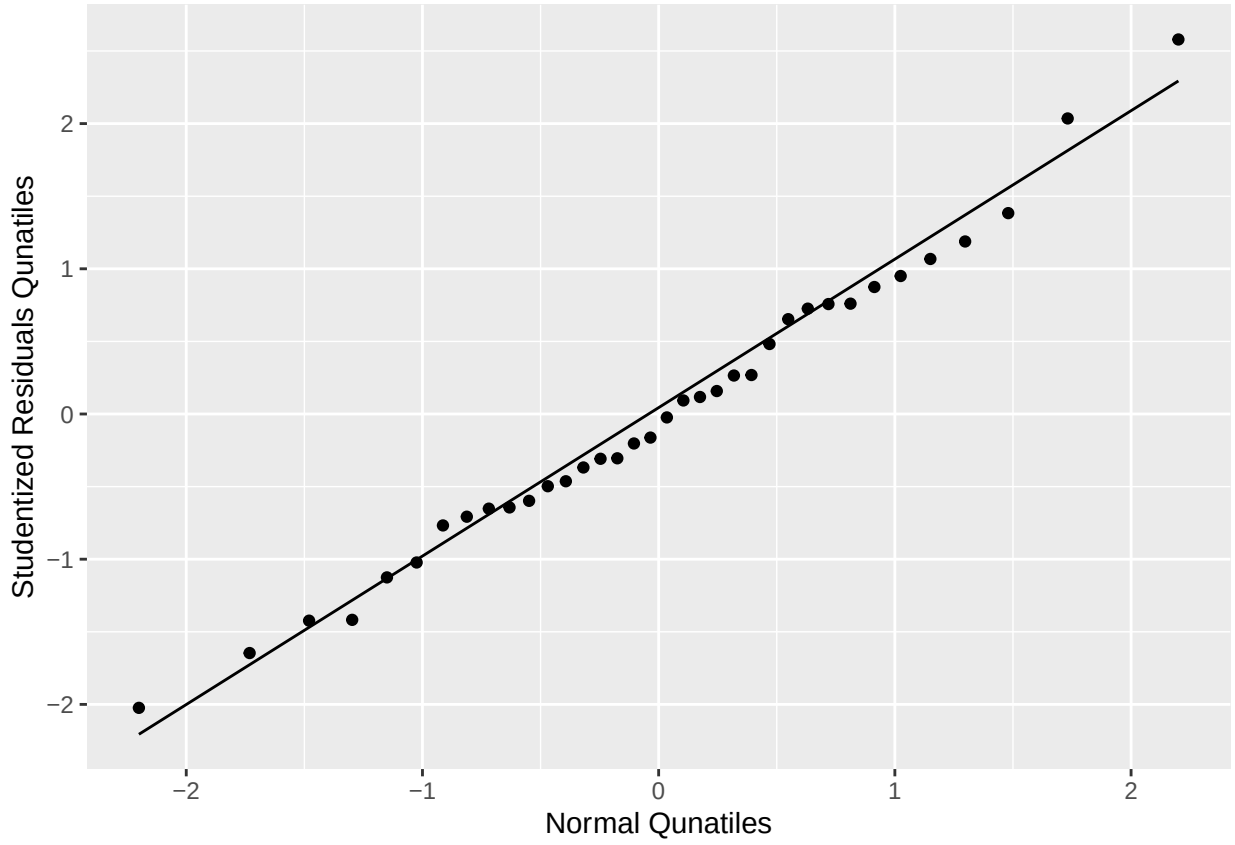
```
pairs(daylengthbytemp) #nothing significant!
```

```
## temp = 15:
##  contrast estimate    SE df t.ratio p.value
##  LD - SD     0.723 0.389 32   1.859  0.0723
##
## temp = 20:
##  contrast estimate    SE df t.ratio p.value
##  LD - SD     0.155 0.389 32   0.399  0.6922
```

Model fit

Does the Model fit the data?? QQplot - looks good

```
ggplot(glut_anova, aes(sample = rstandard(glut_anova))) + geom_qq() + stat_qq_line()+ylab("Studentized R
```



Plotting

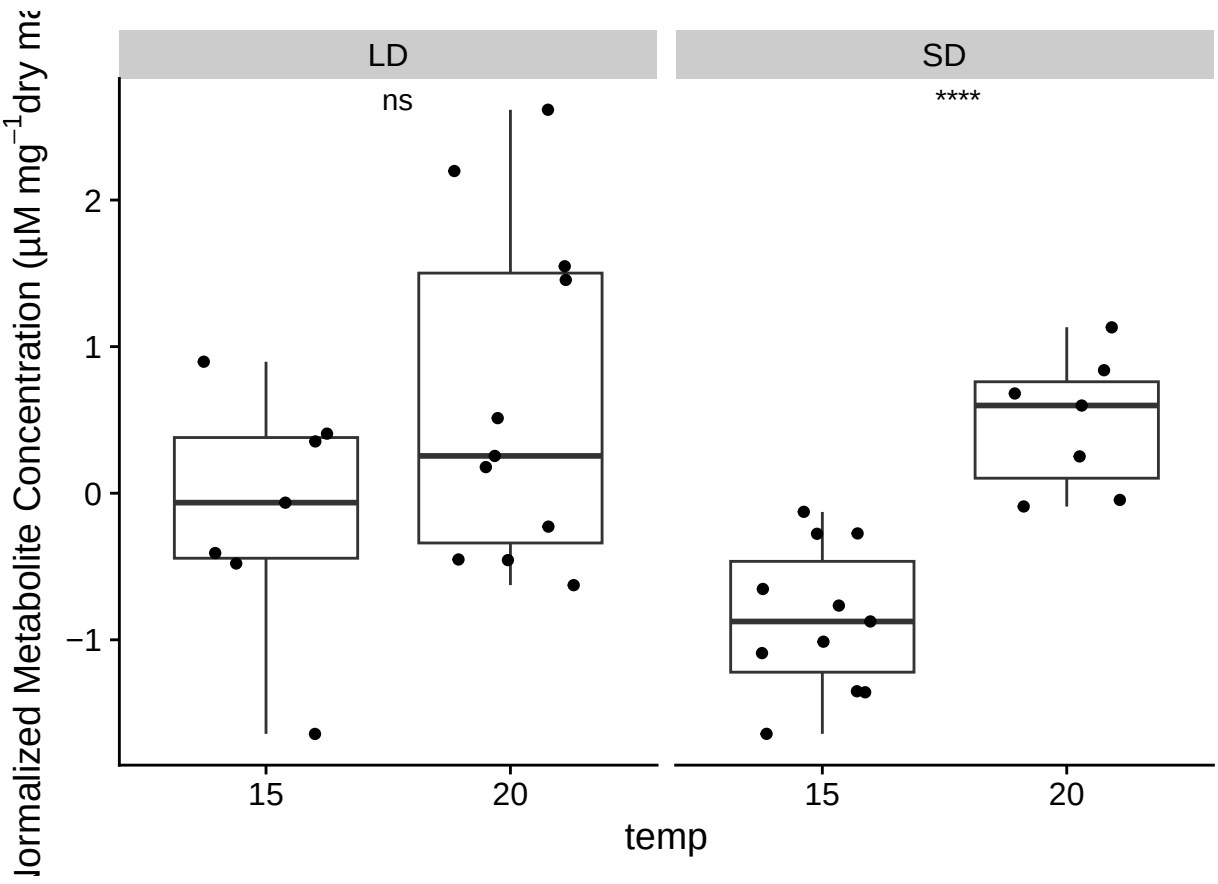
```
glut_graph <- ggplot(glut, aes(x = temp, y = conc)) +
  geom_boxplot() +
  geom_jitter() +
  facet_wrap(~day_length) +
  theme_cowplot() +
  stat_compare_means(label =  "p.signif", label.x = 1.5)
```

Making a Metabo Graphing Function

now make a function for graphing by metabolite

```
metabograph <- function(data, metabolite, x, wrap){

  data %>%
  dplyr::select(sample, group, {{metabolite}}) %>%
  rename(conc = {{metabolite}}) %>%
  #seperating SD20 into SD and 20
  separate(group,
```

```
            into = c("day_length", "temp"),
            sep = "(?<=[A-Za-z])(?=[0-9])") %>%
    ggplot(aes(x = {{x}}, y =conc)) +
  geom_boxplot() +
  geom_jitter(width = 0.28) +
  facet_wrap(enquo(wrap)) +
    theme_cowplot() +
    labs(y = expression(paste("Normalized Metabolite Concentration (µM m",g^-1,"dry mass)")))+
    stat_compare_means(label =  "p.signif", label.x = 1.5)

}

metabograph(nmr_normal, `L-Glutamine`, temp, day_length)
```



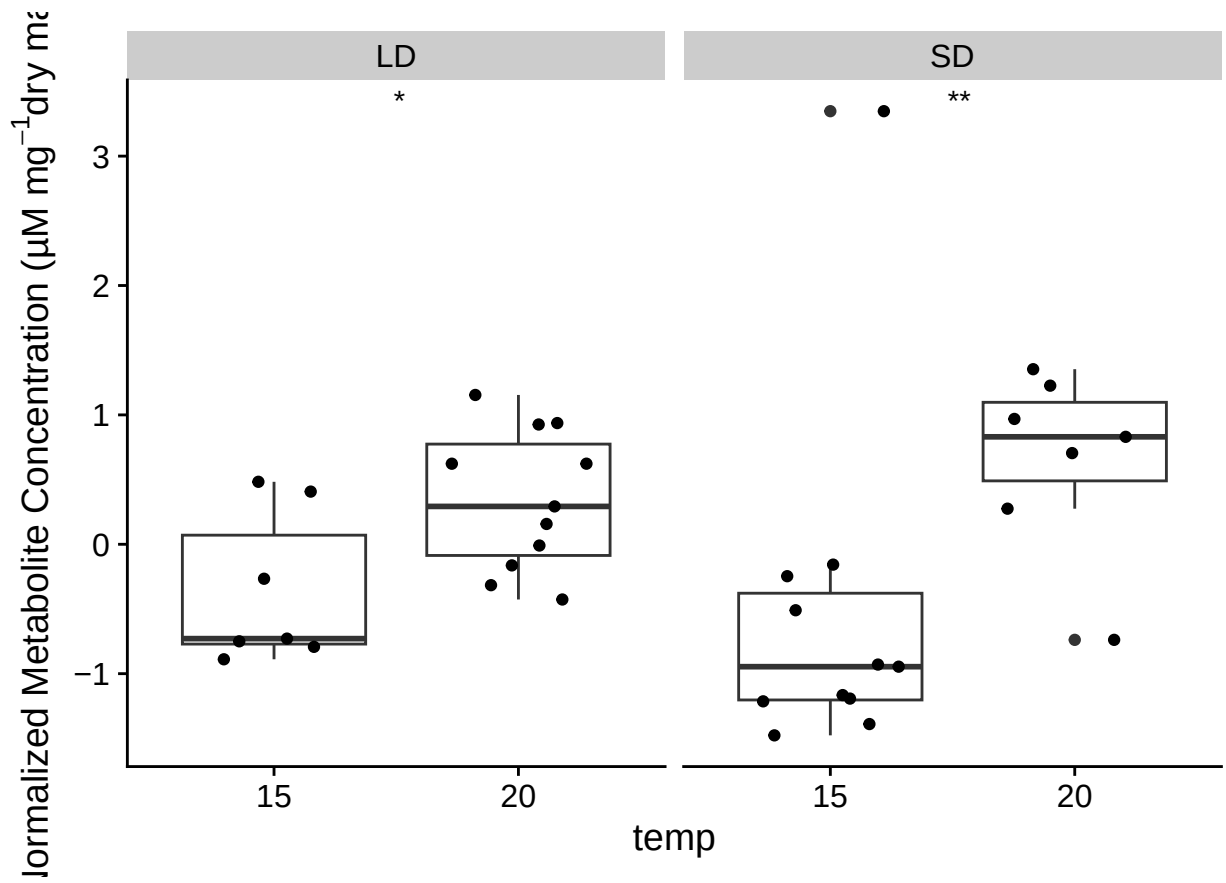

```
#ggsave("glutamine.pdf")
```

Lets try this function on threonine

```
p1 <- metabograph(nmr_normal, `L-Threonine`, day_length, temp)
#ggsave("threonine.pdf")
```

Lets try this function on formate

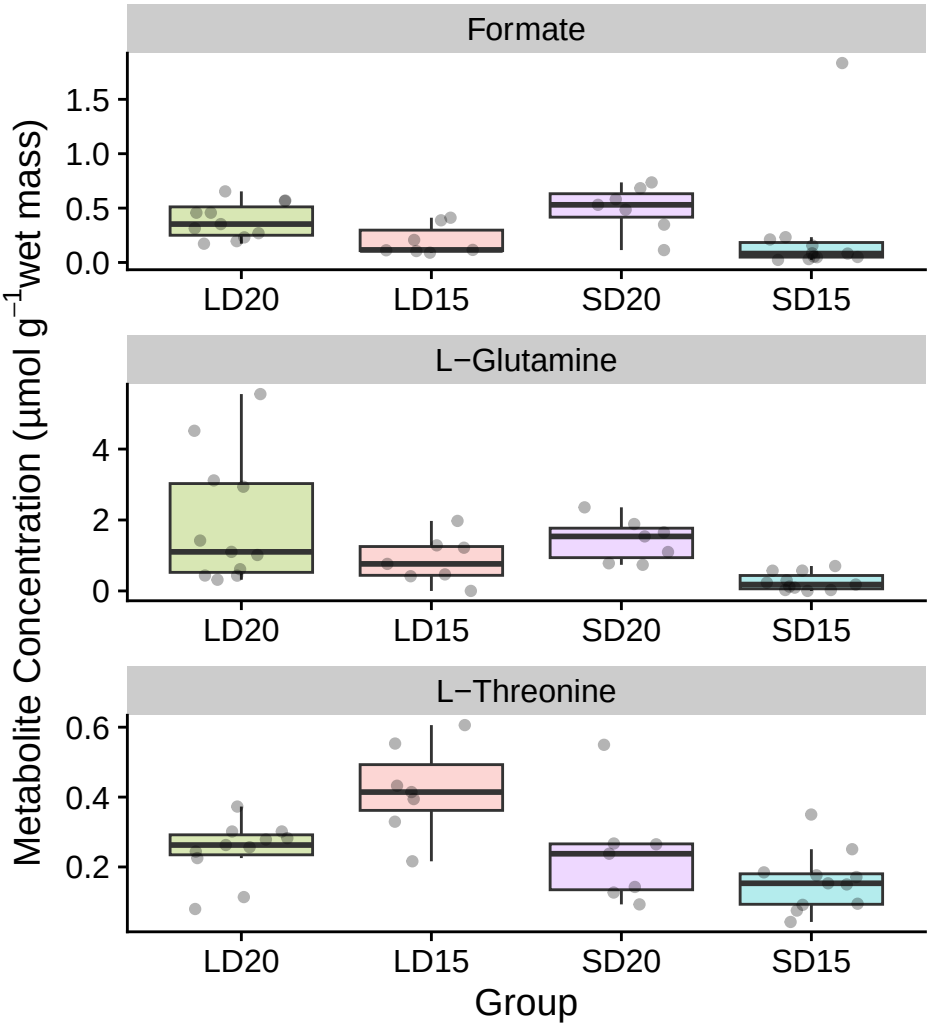
```
metabograph(nmr_normal, Formate, temp, day_length)
```




```
#ggsave("fomate.pdf")
```

Now making a new graph that has a multiplot set up This time I am using the un-transformed data

```
nmr %>%
  dplyr::select(sample, group, Formate, `L-Threonine`, `L-Glutamine`)%>%
  pivot_longer(!c(sample, group), names_to = "metabolite", values_to = "conc") %>%
    ggplot(aes(x = group, y =conc, fill = group)) +
  geom_boxplot(outlier.shape = NA, alpha = 0.3) +
  geom_jitter(width = 0.25, alpha = 0.3) +
  scale_x_discrete(limits = c("LD20", "LD15", "SD20", "SD15"))+

  facet_wrap(vars(metabolite), nrow = 3, scales = "free", strip.position = "top") +
    theme_cowplot() +
  #get rid of legend
theme(legend.position="none")+
    labs(y = expression(paste("Metabolite Concentration (µmol ",g^-1,"wet mass)")),
         x = "Group")
```




```
#getting n values
nmr %>%
  group_by(group) %>%
  summarize(count = n())
```

```
## # A tibble: 4 x 2
##   group count
##   <chr> <int>
## 1 LD15      7
## 2 LD20     11
## 3 SD15     11
## 4 SD20      7
```

```
#ggsave("all_metabolites_plot_updatedoct2025.pdf")
```

Table for results

```
#read in file with metabolite groups
mgroup <- read_csv("../data/metabolite_groups.csv")
```

```
## Rows: 21 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Organic Acids, Peptides and Analogues, Amino acids, Sugars, alcohol...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mgroup_long <- mgroup %>%
  pivot_longer(cols = everything(),
               names_to = "category",
               values_to = "metabolite",
               values_drop_na = TRUE)

table_for_results <- nmr %>%
  dplyr::select(!c(group)) %>%
  pivot_longer(!sample, names_to = "metabolite", values_to = "concentrations") %>%
  group_by(metabolite) %>%
  summarise(N    = length(concentrations),
                    mean = mean(concentrations, na.rm = TRUE),
                    sd   = sd(concentrations, na.rm = TRUE),
                    std   = sd / sqrt(N)) %>%
  #rounding
  mutate_at(vars(-metabolite), funs(round(., 2))) %>%
  inner_join(mgroup_long,
             by = "metabolite") %>%
#paste together mean and std and metabolite
  mutate(all = paste("[",mean,"±",std, "]"),
         #get rid of white spaces
         all = gsub(" ", "", all),
         #paste the metabolite name again
         all2 = paste(metabolite, all))%>%
   #get rid of N and sd
  dplyr::select(all2, category) %>%
  ungroup() %>%
mutate(row_id = row_number()) %>%
  # Pivot wider while using 'row_id' to avoid aggregation issues
  pivot_wider(
    names_from = category,
    values_from = all2,
    id_cols = row_id) %>%
  dplyr::select(-c(row_id))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
```

```
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
#write.xlsx(table_for_results, 'table_for_results_updated.xlsx')
```