

resp_calc_analysis_RMD

Samuel Gurr

9/22/2021

CALCULATE RESPIRATION RATES

(1) summary table of blanks to normalize respiration rates

- we will use 'Lpc' value (or L%) was chosen to generally choose the local regression with the majority of observations in the truly linear subset of data
- review Olito et al. 2017 Estimating monotonic rates from biological data using local linear regression

```
##           Date                Notes BLANK.mean.Lpc BLANK.mean.Leq
## 1 4/30/2021    20210319_new_sensor_7  0.0007473465  0.0006079785
## 2 4/30/2021    20210430_LOWtemp_HIGHsal  0.0189874195  0.0197653280
## 3 4/30/2021    20210430_LOWtemp_LOWsal  0.0013211745  0.0012705960
## 4 4/30/2021    20210430_raw            0.0005703090  0.0005747545
## 5 5/7/2021    20210507_HIGHtemp_HIGHsal  0.0011802100  0.0015554425
## 6 5/7/2021    20210507_HIGHtemp_LOWsal  0.0000902505  0.0001397385
## BLANK.mean.Lz
## 1 0.0006814560
## 2 0.0197631395
## 3 0.0012566205
## 4 0.0005747545
## 5 0.0013648970
## 6 0.0003289235
```

(2) Calculate respiration rates as ng O2 L⁻¹ indiv⁻¹ hr⁻¹

- normalize by mean value of blanks for each treatment, grouped by 'Date' and 'Notes' (as the run within day)
- omit rows respriting blank resp > sample resp - likely bad data, view to troubleshoot if needed

```
## [1] "number of blank resp > sample resp ="
## [2] "0"
```

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

Date	TempCarbSal	n
4/30/2021	HHH	3
4/30/2021	HHL	3

Date	TempCarbSal	n
4/30/2021	HLH	3
4/30/2021	HLL	3
4/30/2021	LHH	3
4/30/2021	LHL	3
4/30/2021	LLH	3
4/30/2021	LLL	3
5/7/2021	HHH	3
5/7/2021	HHL	1
5/7/2021	HLH	3
5/7/2021	LHH	6
5/7/2021	LHL	3
5/7/2021	LLH	4
5/7/2021	LLL	3

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

Date	TempCarbSal	n
4/30/2021	HHH	3
4/30/2021	HHL	3
4/30/2021	HLH	3
4/30/2021	HLL	3
4/30/2021	LHH	3
4/30/2021	LHL	3
4/30/2021	LLH	3
4/30/2021	LLL	3
5/7/2021	HHH	3
5/7/2021	HHL	1
5/7/2021	HLH	3
5/7/2021	LHH	3
5/7/2021	LHL	3
5/7/2021	LLH	3
5/7/2021	LLL	3

- calculate resp rates

```
##      Date Chamber_tank Channel resp_ng_L_indiv_hr
## 1 4/30/2021          1    CH1      0.010373786
## 2 4/30/2021         10    CH5      0.056411725
## 3 4/30/2021         11    CH6      0.085424982
## 4 4/30/2021         12    CH7      0.153077828
## 5 4/30/2021         13    CH1      0.045433306
## 6 4/30/2021         14    CH2      0.035187987
## 7 4/30/2021         15    CH3      0.042009624
## 8 4/30/2021         16    CH1      0.024826159
## 9 4/30/2021         17    CH2      0.030249248
## 10 4/30/2021        18    CH3      0.100367477
## 11 4/30/2021        19    CH5      0.009237778
## 12 4/30/2021          2    CH2      0.005418014
```

## 13	4/30/2021	20	CH6	0.006318191
## 14	4/30/2021	21	CH7	0.004381573
## 15	4/30/2021	22	CH5	0.016235173
## 16	4/30/2021	23	CH6	0.084238599
## 17	4/30/2021	24	CH7	0.228844946
## 18	4/30/2021	3	CH3	0.005221979
## 19	4/30/2021	4	CH1	0.022792656
## 20	4/30/2021	5	CH2	0.015750270
## 21	4/30/2021	6	CH3	0.017637781
## 22	4/30/2021	7	CH5	0.027893973
## 23	4/30/2021	8	CH6	0.017850225
## 24	4/30/2021	9	CH7	0.147881139
## 25	5/7/2021	1	CH1	0.132703528
## 26	5/7/2021	11	CH2	0.027388195
## 27	5/7/2021	11	CH2	0.048215423
## 28	5/7/2021	12	CH3	0.006369162
## 29	5/7/2021	12	CH3	0.044373735
## 30	5/7/2021	13	CH5	0.059501344
## 31	5/7/2021	13	CH5	0.046331672
## 32	5/7/2021	14	CH6	0.029162856
## 33	5/7/2021	14	CH6	0.003257876
## 34	5/7/2021	15	CH7	0.118937608
## 35	5/7/2021	15	CH7	0.076609967
## 36	5/7/2021	16	CH1	0.104695587
## 37	5/7/2021	17	CH2	0.491410327
## 38	5/7/2021	18	CH3	0.030979514
## 39	5/7/2021	2	CH2	0.041167203
## 40	5/7/2021	22	CH5	0.076267543
## 41	5/7/2021	23	CH6	0.038379673
## 42	5/7/2021	24	CH7	0.048322655
## 43	5/7/2021	3	CH3	0.227890553
## 44	5/7/2021	4	CH5	0.071063750
## 45	5/7/2021	5	CH6	0.074364906
## 46	5/7/2021	6	CH7	0.031637759
## 47	5/7/2021	8	CH2	0.140789280

ANALYSIS

Day 1 Respirometry (20210430)

- a. filter the dataset for target data
- b. add column for mean aragonite sat. (note: this is from the entire experiment, not just day 1 chemistry!)
- c. run moels and diagnostics
- d. plot (reconfigure data as needed for plotting; i.e. melt/dcast, etc.)

```
# a.
Resp_age1 <- Resp.Master %>%
  dplyr::filter(Date %in% '4/30/2021') %>%
  dplyr::mutate(Treatment = as.character(paste('Temp.', Temp, '_OA.', pCO2, '_Sal.', Salinity, sep = ' ')))
  dplyr::mutate(pCO2_Salinity = as.character(substr(TempCarbSal, 2, 3))) %>%
```

```

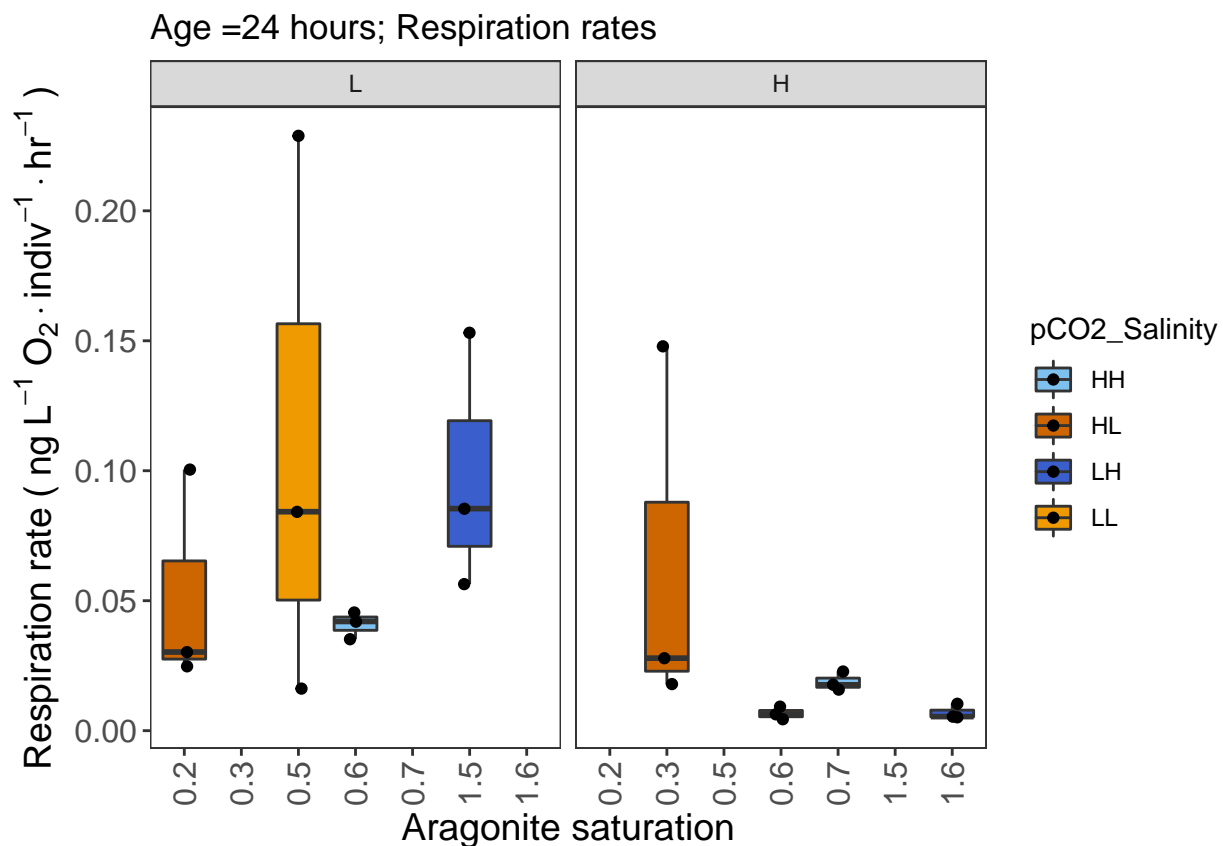
dplyr::mutate(Temp = (fct_relevel(Temp, c('L','H'))))

# b.
Resp_age1 <- merge(Resp_age1, aragonite_dat, by = 'TempCarbSal') # merge with the mean aragonite saturation

## Plot geeom boxlot with jitter
Resp_Age1_Boxplot <- ggplot(data=Resp_age1,
  aes(x=as.factor(Aragonite_saturation),
    #aes(x=meanAragonite,
      y=resp_ng_L_indiv_hr,
      fill=pCO2_Salinity)) +
  geom_boxplot()+
  geom_jitter(width = 0.1)+
  scale_fill_manual(values=c("skyblue2", "darkorange3","royalblue3", "orange2"))+
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  labs(title = "Age =24 hours; Respiration rates",
    y = expression(Respiration~rate~"("~ng~L^{-1}~O_2~indiv^{-1}~hr^{-1}~h),
    x = "Aragonite saturation") +
  theme(axis.title.y = element_text(size = rel(1.3), angle =90), axis.title.x = element_text(size=12)) +
  facet_wrap(~Temp )

```

Resp_Age1_Boxplot



```
# c.
LMmod.age1 <- aov(lm( resp_ng_L_indiv_hr~Temp*pCO2*Salinity,data=Resp_age1))
print('LM model == aov( lm ( resp_ng_L_indiv_hr ~ Temp * pCO2 * Salinity, data = Resp_APRIL ) )')

## [1] "LM model == aov( lm ( resp_ng_L_indiv_hr ~ Temp * pCO2 * Salinity, data = Resp_APRIL ) )"

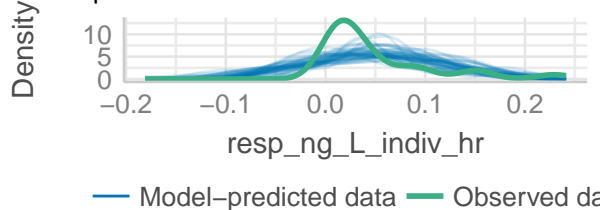
summary(LMmod.age1)

##              Df Sum Sq Mean Sq F value Pr(>F)
## Temp          1  0.01558  0.015583   5.849  0.0279 *
## pCO2           1  0.00079  0.000786   0.295  0.5946
## Salinity       1  0.00173  0.001727   0.648  0.4326
## Temp:pCO2      1  0.01284  0.012835   4.817  0.0433 *
## Temp:Salinity   1  0.00020  0.000199   0.075  0.7880
## pCO2:Salinity   1  0.00078  0.000781   0.293  0.5956
## Temp:pCO2:Salinity 1  0.00082  0.000818   0.307  0.5872
## Residuals     16  0.04263  0.002664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

check_model(LMmod.age1) # observe the diagnostics of the model
```

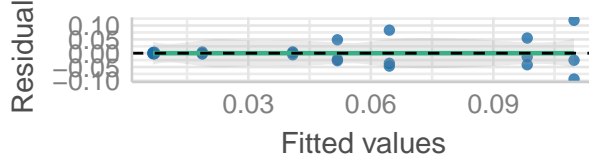
Posterior Predictive Check

Model-predicted lines should resemble observed c



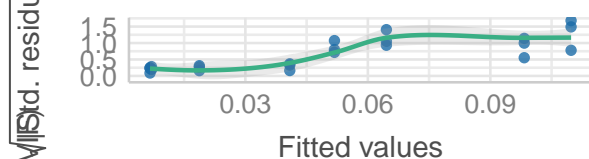
Linearity

Reference line should be flat and horizontal



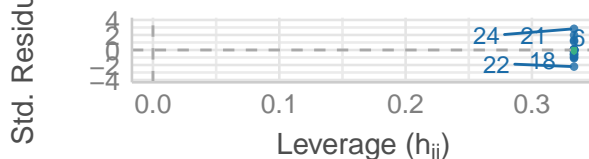
Homogeneity of Variance

Reference line should be flat and horizontal



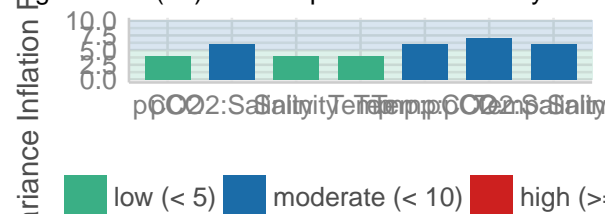
Influential Observations

Points should be inside the contour lines



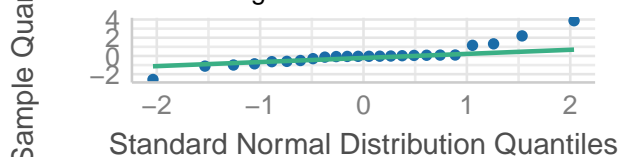
Collinearity

Higher bars (>5) indicate potential collinearity issues



Normality of Residuals

Points should fall along the line



```
shapiro.test(residuals(LMmod.age1)) # non normal
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: residuals(LMmod.age1)  
## W = 0.88542, p-value = 0.0107
```

```
leveneTest(LMmod.age1) # good
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group 7  1.2514 0.3331  
##      16
```

```
# LME model here however we do only have one rep per chamber_tank in some cases, thus the lm (above is  
# MEmod.APRIL <- lmer(resp_ng_L_indiv_hr~Temp*pCO2*Salinity + (1/Chamber_tank),REML=F, data=Resp_APRIL)  
# summary(MEmod.APRIL)  
# check_model(MEmod.APRIL)  
# shapiro.test(residuals(MEmod.APRIL)) # non normal 0.0107  
# leveneTest(MEmod.APRIL) # good
```

```
# Post-hoc tests and exploration of sig effects  
TukeyHSD(LMmod.age1, 'Temp:pCO2')
```

```
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##  
## Fit: aov(formula = lm(resp_ng_L_indiv_hr ~ Temp * pCO2 * Salinity, data = Resp_age1))  
##  
## $'Temp:pCO2'  
##      diff      lwr      upr      p adj  
## H:H-L:L  -0.004711293 -0.08997364  0.08055106 0.9985320  
## L:L-L:L  0.057693242 -0.02756911  0.14295559 0.2526911  
## H:L-L:L  -0.039520413 -0.12478276  0.04574194 0.5605082  
## L:L-H:H  0.062404535 -0.02285782  0.14766689 0.1970100  
## H:L-H:H  -0.034809121 -0.12007147  0.05045323 0.6545779  
## H:L-L:L  -0.097213656 -0.18247601 -0.01195130 0.0227961
```

```
# d.
```

```
Resp_age1_select <- Resp_age1 %>% dplyr::select(c('resp_ng_L_indiv_hr', 'Temp', 'pCO2', 'Salinity', 'Age1'))  
Resp_age1L_melt <- tidyr::gather(Resp_age1_select, variable, value, -resp_ng_L_indiv_hr)
```

```
## Warning: attributes are not identical across measure variables;  
## they will be dropped
```

```
# MAIN EFFECT PLOT  
# ggplot(Resp_APRIL_melt, aes(value , resp_ng_L_indiv_hr , fill = factor(value ))) +  
#   theme(panel.grid=element_blank()) +  
#   scale_color_manual(values=c("#56B4E9", "#D55E00")) +
```

```

#   geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.5))+
#   geom_boxplot(size=0.2, alpha=0.1) +
#   theme_bw() +
#   facet_wrap(~variable, scales = "free_y")

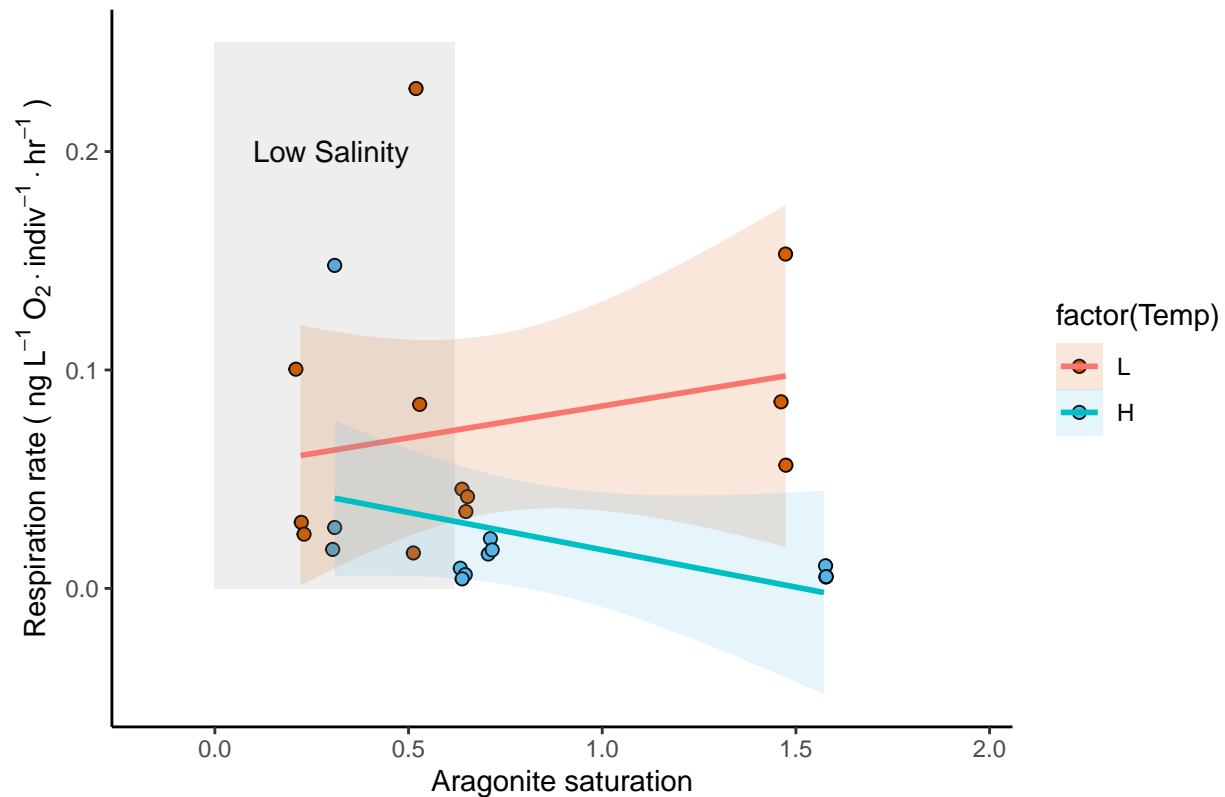
# INTERACTION PLOT
Resp_age1 %>%
  dplyr::mutate(full.treatment = (paste(Temp, pCO2, Salinity, sep=''))) %>%
  dplyr::mutate(full.treatment = fct_relevel(full.treatment,
      "HHH", "HLH", "LHH", 'LLH',
      "HHL", "HLL", "LHL", 'LLL')) %>%
  ggplot(aes(meanAragonite, resp_ng_L_indiv_hr , group =full.treatment, fill = factor(Temp))) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.05))+
  #geom_boxplot(size=.2, alpha=0.1, width =0.2) +
  scale_fill_manual(values=c("#D55E00", "#56B4E9", "#D55E00", "#56B4E9",
      "#D55E00", "#56B4E9", "#D55E00", "#56B4E9")) +
  labs(title = "Respiration rates; C virginica day 1",
      y = expression(Respiration~rate~"(" ~ng~L^{-1}~0[2]%.%indiv^{-1}%.% hr^{-1}~")"),
      x = "Aragonite saturation") +
  annotate("text", x=0.3, y=0.2, label = "Low Salinity") +
  annotate("rect", xmin = 0, xmax = 0.62, ymin = 0, ymax = 0.25, alpha = .1) +
  geom_smooth(method = "lm", level = 0.95, alpha = .15, aes(group=Temp, colour = factor(Temp))) +
  theme_classic()

```

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

```
## Warning: position_jitterdodge requires non-overlapping x intervals
```

Respiration rates; C virginica day 1



```
pd <- position_dodge(0.1) # adjust the jitter for the different treatments
library(Rmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: plyr
```

```
## Warning: package 'plyr' was built under R version 4.1.3
```

```
## -----
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
```

```
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
```

```
## library(plyr); library(dplyr)
```

```
## -----
```

```
##
```

```
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
```

```
##
```

```
## arrange, count, desc, failwith, id, mutate, rename, summarise,
```

```
## summarize
```



```

Resp_age1 %>%
  dplyr::mutate(full.treatment = (paste(Temp, pCO2, Salinity, sep=''))) %>%
  dplyr::mutate(full.treatment = fct_relevel(full.treatment,
      "HHH", "HLH", "LHH", "LLH",
      "HHL", "HLL", "LHL", "LLL")) %>%
  summarySE(measurevar="resp_ng_L_indiv_hr", groupvars=c("full.treatment", "meanAragonite", "Temp")) %>%
  ggplot(aes(meanAragonite, resp_ng_L_indiv_hr, group=full.treatment, fill = factor(Temp))) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.05)) +
  geom_errorbar(aes(ymin=resp_ng_L_indiv_hr-se, ymax=resp_ng_L_indiv_hr+se), colour="black", width=.1) +
  geom_line(position=pd) +
  scale_fill_manual(values=c("#D55E00", "#56B4E9", "#D55E00", "#56B4E9",
      "#D55E00", "#56B4E9", "#D55E00", "#56B4E9")) +
  labs(title = "Respiration rates; C virginica day 1",
      y = expression(Respiration~rate~("("~ng~L^{-1}~O[2]~.%indiv^{-1}~.% hr^{-1}~)"),
      x = "Aragonite saturation") +
  annotate("text", x=0.3, y=0.24, label = "Low Salinity") +
  annotate("text", x=1.25, y=0.24, label = "High Salinity") +
  annotate("text", x=1.58, y=0.14, label = "a") +
  annotate("text", x=1.65, y=0.018, label = "b") +
  annotate("rect", xmin = 0, xmax = 0.62, ymin = 0, ymax = 0.25, alpha = .05) +
  # geom_smooth(method = "lm", level = 0.95, alpha = .15, aes(group=Temp, colour = factor(Temp))) +
  theme_classic()

```

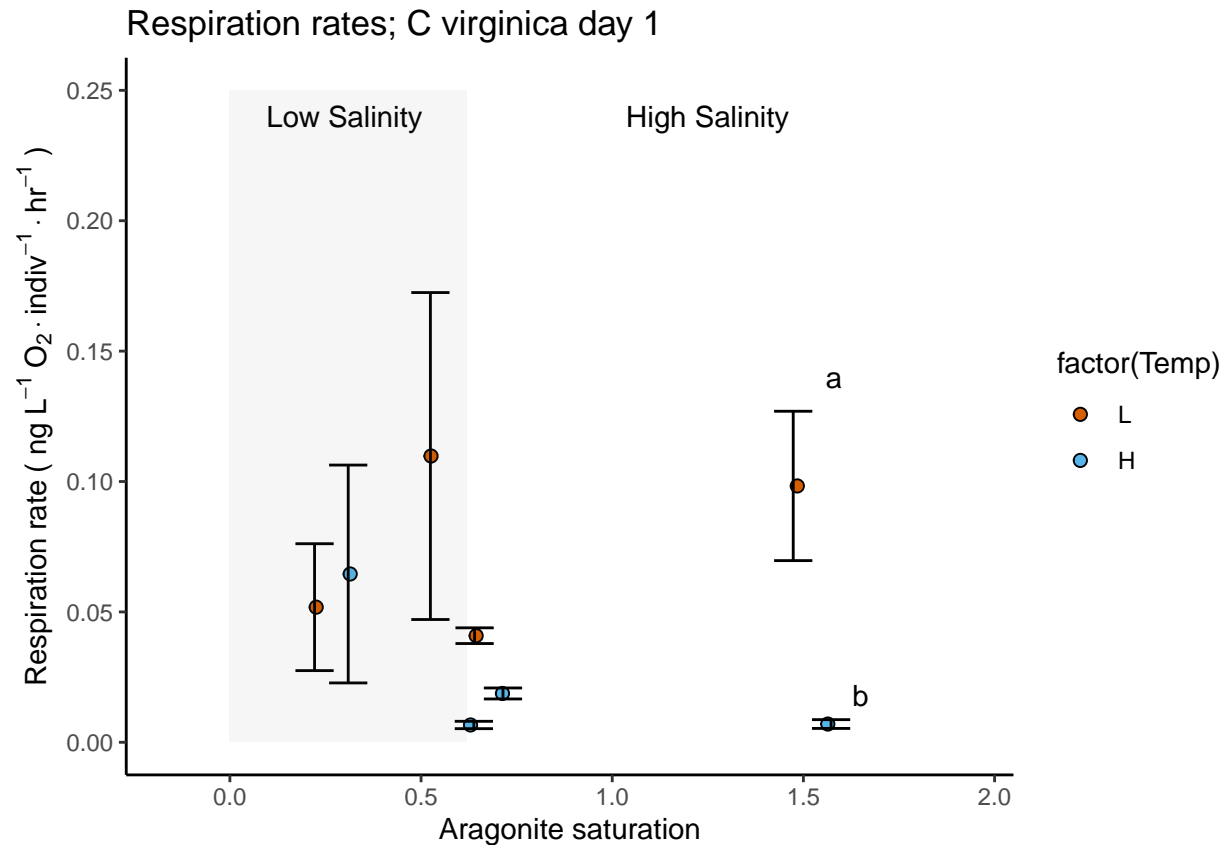
```
## Warning: position_jitterdodge requires non-overlapping x intervals
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

```
## position_dodge requires non-overlapping x intervals
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
```

```
## the group aesthetic?
```



Day 8 Respirometry (20210507)

```
# a.
Resp_age8 <- Resp.Master %>%
  dplyr::filter(Date %in% '5/7/2021') %>%
  dplyr::mutate(Treatment = as.character(paste('Temp.', Temp, '_OA.', pCO2, '_Sal.', Salinity, sep = ')))
  dplyr::mutate(pCO2_Salinity = as.character(substr(TempCarbSal, 2, 3))) %>%
  dplyr::mutate(Temp = (fct_relevel(Temp, c('L', 'H'))))

# b.
Resp_age8 <- merge(Resp_age8, aragonite_dat, by = 'TempCarbSal') # merge with the mean aragonite saturation

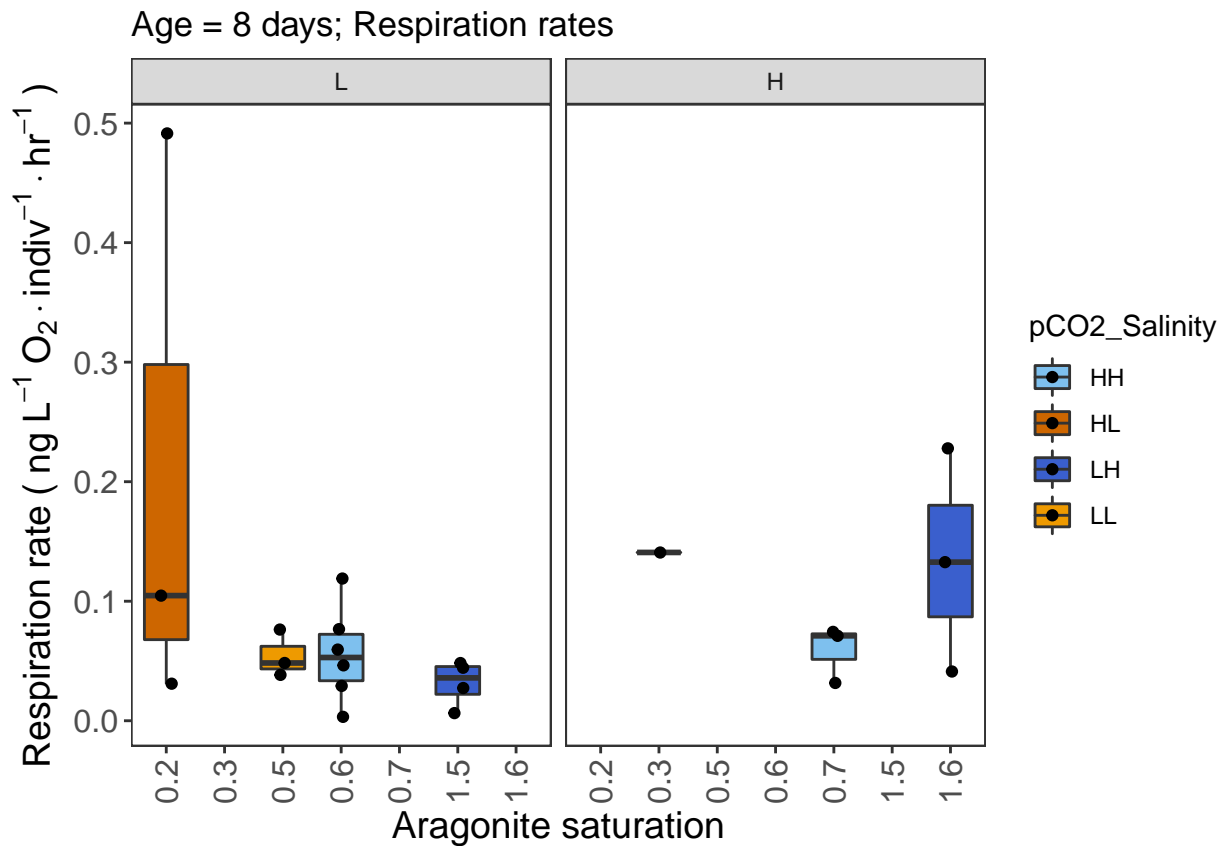
## Plot geom boxplot with jitter
Resp_Age8_Boxplot <- ggplot(data=Resp_age8,
  aes(x=as.factor(Aragonite_saturation),
    #aes(x=meanAragonite,
      y=resp_ng_L_indiv_hr,
      fill=pCO2_Salinity)) +
  geom_boxplot()+
  geom_jitter(width = 0.1)+
  scale_fill_manual(values=c("skyblue2", "darkorange3", "royalblue3", "orange2")) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

```

theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
labs(title = "Age = 8 days; Respiration rates",
      y = expression(Respiration~rate~"("~ng~L^{-1}~0[2]~.%~indiv^{-1}~.%~h),
      x = "Aragonite saturation") +
theme(axis.title.y = element_text(size = rel(1.3), angle =90), axis.title.x =
theme(axis.text=element_text(size=12)) +
facet_wrap(~Temp )

```

Resp_Age8_Boxplot



```

# c.
LMmod.age8 <- aov(lm(resp_ng_L_indiv_hr~Temp*pCO2*Salinity,data=Resp_age8))
print('LM model == aov( lm ( resp_ng_L_indiv_hr ~ Temp * pCO2 * Salinity, data = Resp_APRIL ) )')

```

```
## [1] "LM model == aov( lm ( resp_ng_L_indiv_hr ~ Temp * pCO2 * Salinity, data = Resp_APRIL ) )"
```

```
summary(LMmod.age8)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## Temp       1  0.00296  0.002962   0.314  0.583
## pCO2       1  0.00478  0.004776   0.507  0.487
## Salinity   1  0.02777  0.027769   2.947  0.105
## Temp:pCO2  1  0.02652  0.026520   2.815  0.113
## Temp:Salinity 1  0.00008  0.000080   0.008  0.928
## pCO2:Salinity 1  0.01576  0.015758   1.673  0.214
## Residuals 16  0.15075  0.009422
```

```
# check_model(LMmod.age8) # observe the diagnostics of the model
shapiro.test(residuals(LMmod.age8)) # non normal
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(LMmod.age8)
## W = 0.82009, p-value = 0.0008179
```

```
leveneTest(LMmod.age8) # good
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  6  1.3835 0.2802
##      16
```

```
# LME model here however we do only have one rep per chamber_tank in some cases, thus the lm (above is
# MEmod.APRIL <- lmer(resp_ng_L_indiv_hr~Temp*pCO2*Salinity + (1/Chamber_tank),REML=F, data=Resp_APR
# summary(MEmod.APRIL)
# check_model(MEmod.APRIL)
# shapiro.test(residuals(MEmod.APRIL)) # non normal
# leveneTest(MEmod.APRIL) # good
```

```
# d.
Resp_age8_select <- Resp_age8 %>% dplyr::select(c('resp_ng_L_indiv_hr', 'Temp', 'pCO2', 'Salinity'))
Resp_age8_melt <- tidyr::gather(Resp_age8_select, variable, value, -resp_ng_L_indiv_hr)
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
# MAIN EFFECT PLOT
# ggplot(Resp_APRIL_melt, aes(value , resp_ng_L_indiv_hr , fill = factor(value ))) +
#   theme(panel.grid=element_blank()) +
#   scale_color_manual(values=c("#56B4E9", "#D55E00")) +
#   geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.5))+
#   geom_boxplot(size=0.2, alpha=0.1) +
#   theme_bw() +
#   facet_wrap(~variable, scales = "free_y")
```

```
# INTERACTION PLOT
Resp_age8 %>%
  dplyr::mutate(full.treatment = (paste(Temp, pCO2, Salinity, sep=''))) %>%
  dplyr::mutate(full.treatment = fct_relevel(full.treatment,
      "HHH", "HLH", "LHH", 'LLH',
      "HHL", "LHL", 'LLL')) %>%
  ggplot(aes(meanAragonite, resp_ng_L_indiv_hr , group =full.treatment, fill = factor(Temp))) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.05))+
  #geom_boxplot(size=.2, alpha=0.1, width =0.2) +
  scale_fill_manual(values=c("#D55E00", "#56B4E9", "#D55E00", "#56B4E9",
      "#D55E00", "#56B4E9", "#D55E00", "#56B4E9")) +
```

```
labs(title = "Respiration rates; C virginica day 8",
     y = expression(Respiration~rate~("ng~L^{-1}~O_2~indiv^{-1}~hr^{-1}")),
     x = "Aragonite saturation") +
annotate("text", x=0.3, y=0.55, label = "Low Salinity") +
annotate("rect", xmin = 0, xmax = 0.62, ymin = 0, ymax = 0.6, alpha = .1) +
geom_smooth(method = "lm", level = 0.95, alpha = .15, aes(group=Temp, colour = factor(Temp))) +
theme_classic()
```

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

```
## Warning: position_jitterdodge requires non-overlapping x intervals
```

