ClearanceRate_F1Scallops

Samuel Gurr

10/12/2021

Merge to master file 'ClearRate_Master'

Date from: - 20210914 - 20210930

```
# 20210914 DATA ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
# formatting for merge
# call the 914 data and remove blanks
clear.rate_914_Scallops <- merge( # note - merging with th elengths removed the blanks from the clearan
  (clear.rate_914 %>% dplyr::filter(!Chamber_tank %in% 'Blank')),
  (length.resp.clear %>%
     dplyr::filter(Date %in% 20210914) %>%
     dplyr::mutate(Chamber_tank = sub("_", "", Chamber_tank))) ) %>%
  dplyr::select(!c('Food'))

# 20210930 DATA ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
# a bit more work with the 930 data to merge with the 914
clear.rate_930_Scallops <- merge( (data.frame(melt(clear.rate_930, id.vars = c('Date', 'Chamber_tank','
  dplyr::filter(!Fed_Unfed %in% 'blank') %>%
  dplyr::mutate(Time._min = (gsub(".*_","", variable))) %>%
  dplyr::rename(Count = value) %>%
  dplyr::arrange(Run,Plate,pH,Replicate,Number,Fed_Unfed,Time._min) ), #merge with...
    (length.resp.clear %>%
       dplyr::filter(Date %in% 20210930) %>%
       dplyr::mutate(Chamber_tank = sub("_", "", Chamber_tank)) %>%
       dplyr::mutate(Fed_Unfed = ifelse(Food == 1, 'fed', 'unfed')))
  ) %>%
  dplyr::select(!c('variable', 'Food', 'Sample.ID'))



ClearRate_Master          <- rbind(clear.rate_914_Scallops, clear.rate_930_Scallops)

# calculate the clearace rate normalized for shell length
ClearRate_Master$Cells_ml <- (ClearRate_Master$Count)*(1000/33)

print(head(ClearRate_Master))
```

```
##       Date Run  pH Chamber_tank Fed_Unfed Time._min Count Plate Replicate
## 1 20210914   1 7.5         7.5A      <NA>         0  1314    NA         A
## 2 20210914   1 7.5         7.5A      <NA>        10  1186    NA         A
## 3 20210914   1 7.5         7.5A      <NA>        27  1295    NA         A
```

```
## 4 20210914   1 7.5         7.5A      <NA>        47   984    NA         A
## 5 20210914   1 7.5         7.5A      <NA>        57   681    NA         A
## 6 20210914   1 7.5         7.5A      <NA>        70   849    NA         A
##   Number Length.um. Notes Cells_ml
## 1      1    1970.48       39818.18
## 2      1    1970.48       35939.39
## 3      1    1970.48       39242.42
## 4      1    1970.48       29818.18
## 5      1    1970.48       20636.36
## 6      1    1970.48       25727.27
```

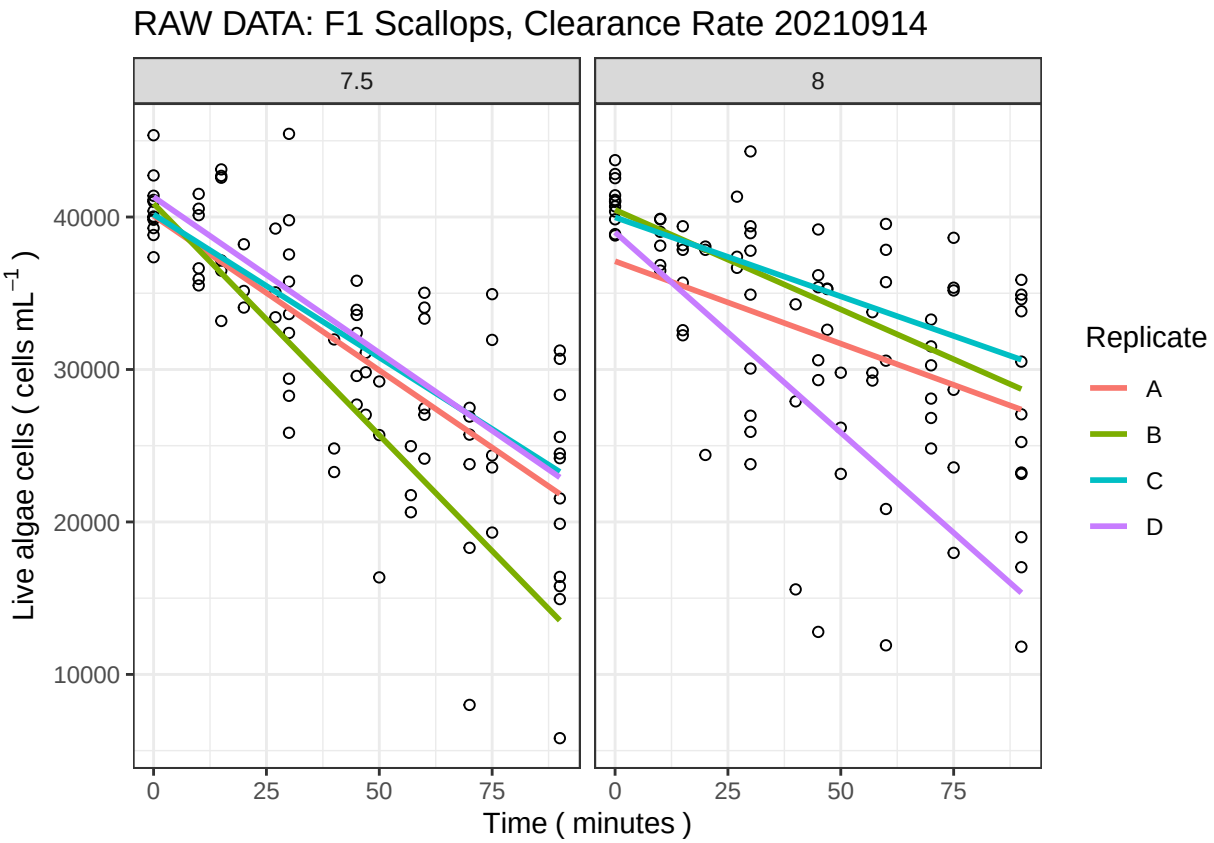
Visualize raw data & simple slopes/algae loss time-1

Plot the raw Flow cytometry output

- note: some data points were omitted in the worksheet, review the 'raw' vs 'worksheet_R' data files

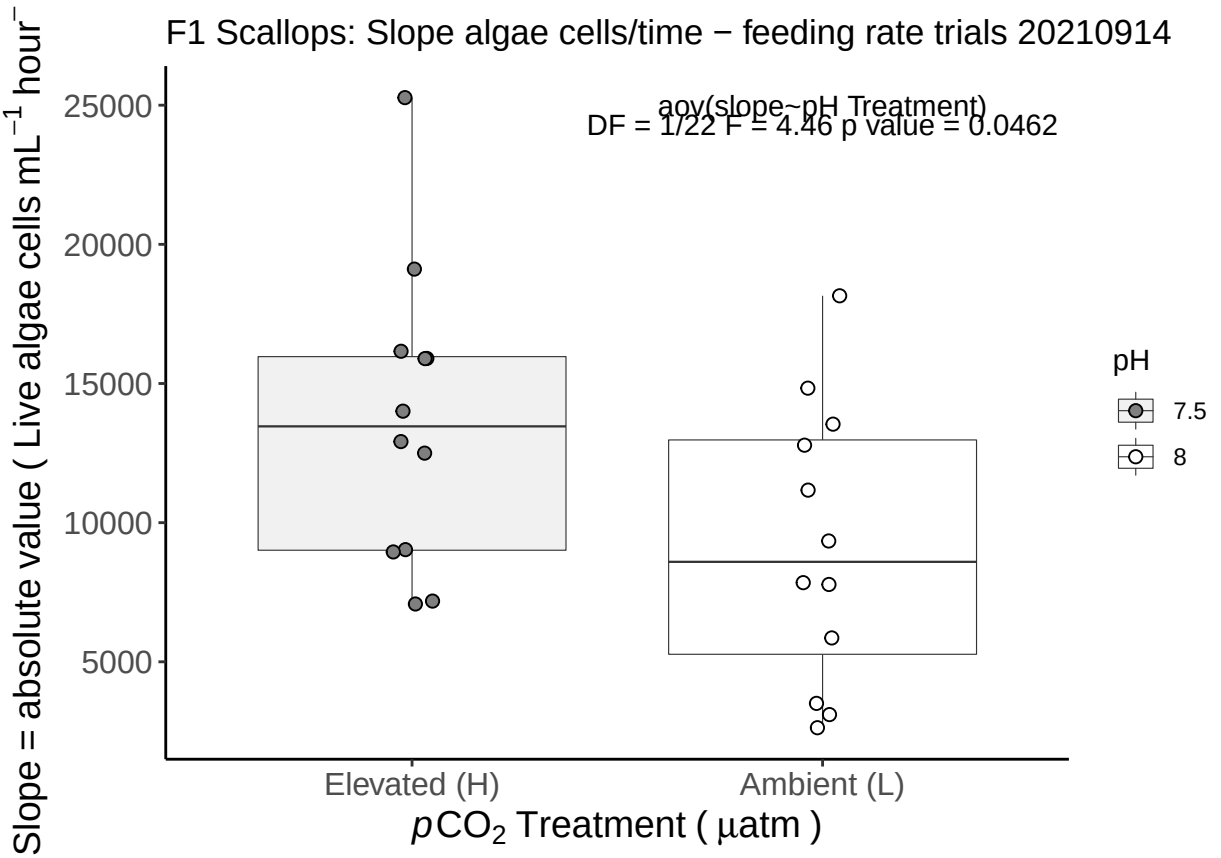
```
# 20210914 DATA ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
ClearRate_Master[!is.na(ClearRate_Master$Cells_ml),] %>%
  dplyr::filter(Date %in% 20210914) %>%
  dplyr::mutate(Time._min = as.numeric(as.character(Time._min))) %>%
  ggplot(aes(Time._min, Cells_ml, color=Replicate)) +
  geom_point(shape=1, color = "black")+
  ggtitle("RAW DATA: F1 Scallops, Clearance Rate 20210914")+
  labs(y = expression(Live~algae~cells~"("~cells~mL^{-1}~")"),
       x = expression(Time~"("~minutes~")")) +
  theme(plot.title= element_text(size =16, face ="bold",
                                 lineheight = 8, vjust=1), aspect.ratio=1)+
  stat_smooth(method="lm", se = F) +
  theme_bw() +
  scale_shape_identity() +
  facet_wrap( ~ pH, scales = "free_x" )
```

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



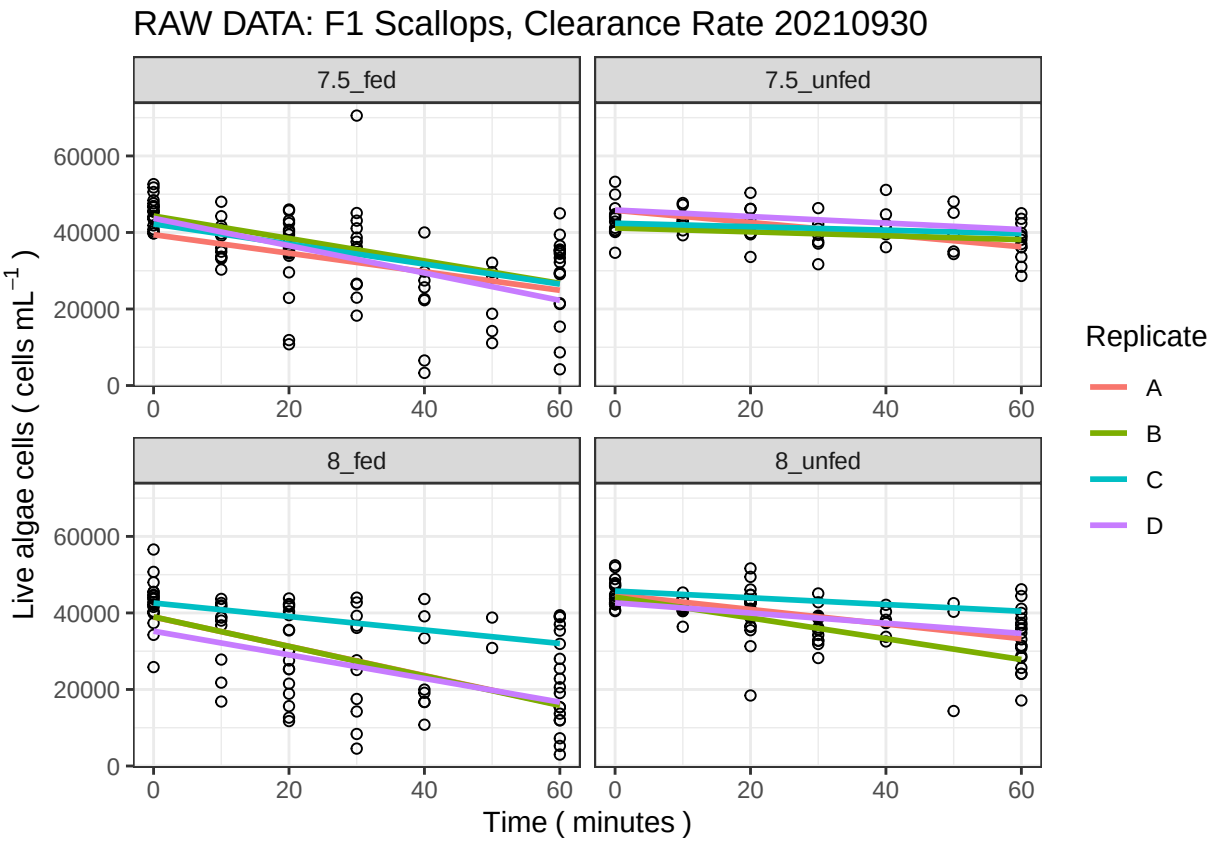

```
# Clearance rate anlaysis for 9/30 data
ClearRate_Master.914 <- ClearRate_Master %>%
  dplyr::filter(Date %in% 20210914) %>%
  dplyr::mutate(uniq_Identifier = paste(pH, "Run", Run, "Rep",Replicate, "Num", Number, sep='_'))

loop_914 <- as.data.frame(unique(ClearRate_Master.914$uniq_Identifier)) %>% dplyr::rename(ID = "unique(

SlopeTable_914 <- data.frame() # run this before the loop
for(i in 1:nrow(loop_914)){
  dat <- ClearRate_Master.914 %>%  filter(uniq_Identifier %in% loop_914[i,])
    slope<- summary(lm((dat$Cells_ml) ~ as.numeric(dat$Time._min)))$coef[2,"Estimate"]
    SLOPE  <- summary(lm((dat$Cells_ml) ~ as.numeric(dat$Time._min)))$r.squared
    pval <- summary(lm((dat$Cells_ml) ~ as.numeric(dat$Time._min)))$coef[2,"Pr(>|t|)"]
    mod  <- lm(as.numeric(dat$Time._min) ~ dat$Cells_ml)
    norm_assum <- shapiro.test(resid(mod))
    shapiro_pval <- norm_assum$p.value
    # assign the data table
    SLOPE.loop <- data.frame(matrix(nrow = 1, ncol = 5)) # create a new data table
    colnames(SLOPE.loop) <- c('pH', 'Replicate', 'slope', 'SLOPE', 'pval') # assign headers
    SLOPE.loop$pH        <- gsub("_.*", "\\1", loop_914[i,])
    SLOPE.loop$Replicate <- gsub("^(?:[^_]+_){4}([^_]+).*", "\\1", loop_914[i,])
    SLOPE.loop$slope     <- slope * 60  # cells per mL per hour
    SLOPE.loop$Number    <- gsub("^(?:[^_]+_){6}([^_]+).*", "\\1", loop_914[i,])
    SLOPE.loop$SLOPE       <- SLOPE
    SLOPE.loop$pval      <- pval
    SLOPE.loop$shapiro_pval <- shapiro_pval
```

```
    # loop additions
    df <- data.frame(SLOPE.loop) # name dataframe for this single row
    SlopeTable_914 <- rbind(SlopeTable_914,df) # bind to a cumulative list dataframe
   # print(SlopeTable_914) # show loop progress in the console
}# outside loo

SLOPE_mod_914 <- aov(lm(slope ~ pH, data= SlopeTable_914))
DF   <- paste( (summary(SLOPE_mod_914)[[1]][["Df"]])[1], (summary(SLOPE_mod_914)[[1]][["Df"]])[2], sep =
Fval <- (summary(SLOPE_mod_914)[[1]][["F value"]])[1]
pval <- (summary(SLOPE_mod_914)[[1]][["Pr(>F)"]])[1]

ggplot(SlopeTable_914, aes(pH , abs(slope) , fill = pH)) +
  theme(panel.grid=element_blank()) +
  geom_boxplot(size=0.2, alpha=0.1, aes(fill=pH)) +
  scale_fill_manual(values=c("grey50","white")) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.1)) +
  theme_classic() +
  scale_x_discrete(labels= c('Elevated (H)', 'Ambient (L)')) +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold")) +
  labs(title = "F1 Scallops: Slope algae cells/time - feeding rate trials 20210914",
       y = expression(Slope~"="~absolute~value~"("~Live~algae~cells~mL^{-1}~hour^{-1}~")"),
       x = expression(italic(p)*CO[2]~Treatment~"("~mu*atm~")")) +
  annotate("text", x=2, y= 25000, size = 4, label = "aov(slope~pH Treatment)") +
  annotate("text", x=2, y= 24300, size = 4, label= paste('DF =',DF,'F =', signif(Fval, digits=3), 'p va
```



```
# 20210930 DATA ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
ClearRate_Master[!is.na(ClearRate_Master$Cells_ml),] %>%
  dplyr::filter(Date %in% 20210930) %>%
  dplyr::mutate(pH_feed = paste(pH, Fed_Unfed, sep='_')) %>%
  dplyr::mutate(Time._min = as.numeric(as.character(Time._min))) %>%
  ggplot(aes(Time._min, Cells_ml, color=Replicate)) +
  geom_point(shape=1, fill = "white", color = "black")+
  ggtitle("RAW DATA: F1 Scallops, Clearance Rate 20210930")+
  labs(y = expression(Live~algae~cells~"("~cells~mL^{-1}~")"),
       x = expression(Time~"("~minutes~")")) +
  theme(plot.title= element_text(size =16, face ="bold",
                                 lineheight = 8, vjust=1), aspect.ratio=1)+
  stat_smooth(method="lm", se = F) +
  theme_bw() +
  scale_shape_identity() +
  facet_wrap( ~ pH_feed, scales = "free_x" )
```

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



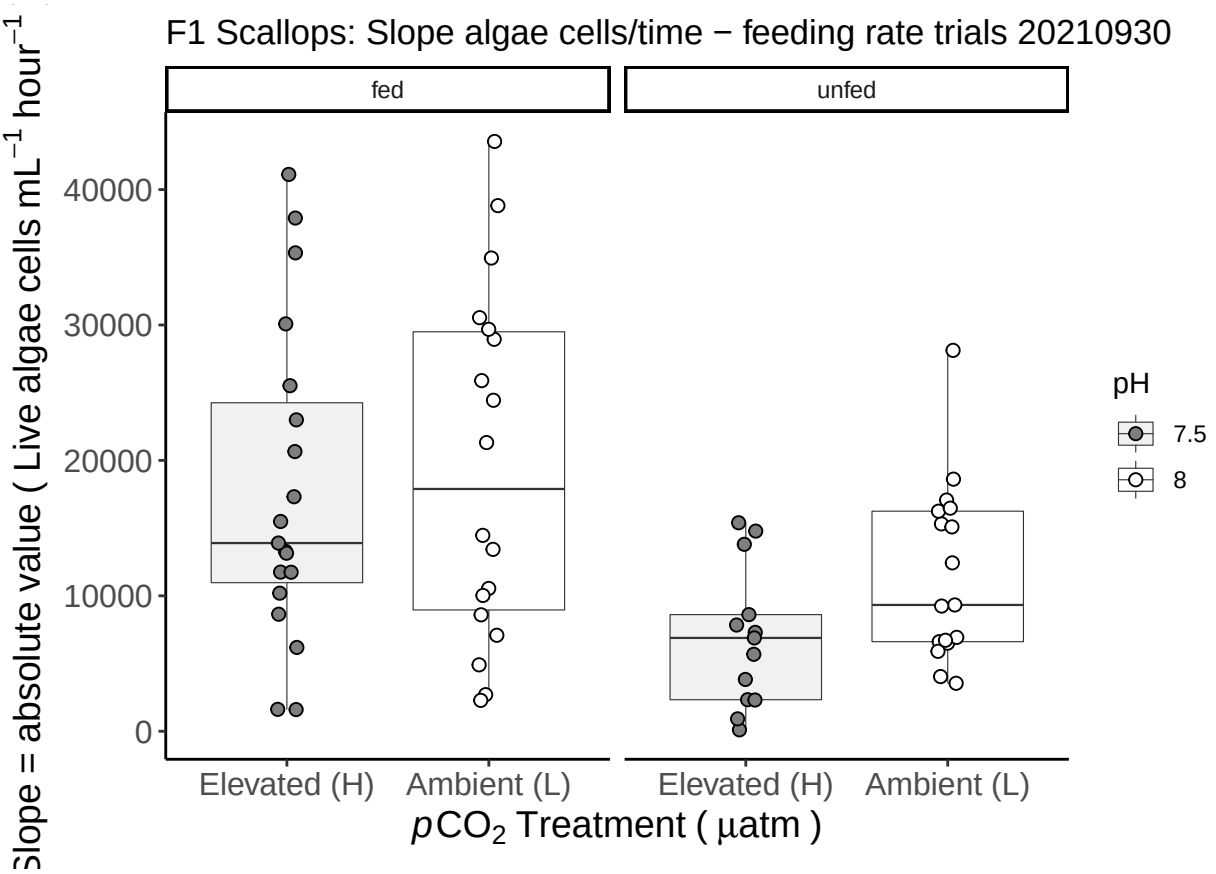

```
# Clearance rate anlaysis for 9/30 data
ClearRate_Master.930       <- ClearRate_Master %>%
  dplyr::filter(Date %in% 20210930) %>%
  dplyr::mutate(uniq_Identifier = paste(pH, Fed_Unfed, "Run", Run, "Rep",Replicate, "Num", Number, sep=
```

```r
loop_930 <- as.data.frame(unique(ClearRate_Master.930$uniq_Identifier)) %>% dplyr::rename(ID = "unique(

SlopeTable_930 <- data.frame() # run this before the loop
for(i in 1:nrow(loop_930)){
  dat <- ClearRate_Master.930 %>%  filter(uniq_Identifier %in% loop_930[i,])
  slope<- summary(lm((dat$Cells_ml) ~ as.numeric(dat$Time._min)))$coef[2,"Estimate"]
  SLOPE  <- summary(lm((dat$Cells_ml) ~ as.numeric(dat$Time._min)))$r.squared
  pval <- summary(lm((dat$Cells_ml) ~ as.numeric(dat$Time._min)))$coef[2,"Pr(>|t|)"]
  mod  <- lm(as.numeric(dat$Time._min) ~ dat$Cells_ml)
  norm_assum <- shapiro.test(resid(mod))
  shapiro_pval <- norm_assum$p.value
  # assign the data table
  SLOPE.loop <- data.frame(matrix(nrow = 1, ncol = 6)) # create a new data table
  colnames(SLOPE.loop) <- c('pH', 'Replicate', 'Fed_Unfed','slope', 'SLOPE', 'pval') # assign headers
  SLOPE.loop$pH        <- gsub("_.*", "\\1", loop_930[i,])
  SLOPE.loop$Replicate <- gsub("^(?:[^_]+_){5}([^_]+).*", "\\1", loop_930[i,])
  SLOPE.loop$Fed_Unfed <- gsub("^(?:[^_]+_){1}([^_]+).*", "\\1", loop_930[i,])
  SLOPE.loop$slope     <- slope * 60  # cells per mL per hour
  SLOPE.loop$Number    <- gsub("^(?:[^_]+_){7}([^_]+).*", "\\1", loop_930[i,])
  SLOPE.loop$SLOPE       <- SLOPE
  SLOPE.loop$pval      <- pval
  SLOPE.loop$shapiro_pval <- shapiro_pval
  # loop additions
  df <- data.frame(SLOPE.loop) # name dataframe for this single row
  SlopeTable_930 <- rbind(SlopeTable_930,df) # bind to a cumulative list dataframe
  #print(SlopeTable_930) # show loop progress in the console
}# outside loo

SLOPE_mod_930 <- aov(lm(slope ~ pH*Fed_Unfed, data= SlopeTable_930))
summary(SLOPE_mod_930)
```

```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## pH            1 1.176e+08 1.176e+08   1.136 0.290525
## Fed_Unfed     1 1.478e+09 1.478e+09  14.284 0.000351 ***
## pH:Fed_Unfed  1 4.590e+07 4.590e+07   0.444 0.507794
## Residuals    63 6.518e+09 1.035e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
DF.930   <- paste( (summary(SLOPE_mod_930)[[1]][["Df"]])[1], (summary(SLOPE_mod_930)[[1]][["Df"]])[2], s
Fval.930 <- (summary(SLOPE_mod_930)[[1]][["F value"]])[1]
pval.930 <- (summary(SLOPE_mod_930)[[1]][["Pr(>F)"]])[1]

SlopeTable_930 %>%  mutate(pH_feed = paste(pH, Fed_Unfed, sep = '_')) %>%
ggplot(aes(pH , abs(slope) , fill = pH)) +
  theme(panel.grid=element_blank()) +
  geom_boxplot(size=0.2, alpha=0.1, aes(fill=pH)) +
  scale_fill_manual(values=c("grey50","white")) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.1)) +
  theme_classic() +
  scale_x_discrete(labels= c('Elevated (H)', 'Ambient (L)')) +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold")) +
```

```
labs(title = "F1 Scallops: Slope algae cells/time - feeding rate trials 20210930",
     y = expression(Slope~"="~absolute~value~"("~Live~algae~cells~mL^{-1}~hour^{-1}~")"),
     x = expression(italic(p)*CO[2]~Treatment~"("~mu*atm~")")) +
facet_wrap(~ Fed_Unfed)
```



Calculate Clearance Rate

Note: the following clusters calculate CR for the start/end Algae counts (i.e. 20210914 time 0 and time 90 minutes; 20210930 time 0 and time 60 minutes)

CLEARANCE RATE EQUATION

FR =(V/t * (ln(C_0/C_t) - A))/ L

- **V** == the volume of the vessel
- **t** == time of the trial interval (i.e. 60 minutes or 1 hour)
- **ln(C_0/C_t)** == ratio of the live algae concentration (cells ml-1) at time 0 (C0) and at the elapsed time interval(s) (Ct) - take the natural log of this number
- **A** == ln(C_0/C_t) for the 'blank' values in each treatment, accounts for the sink or stuck algae cells
- **L** == normalization factor between individuals - here we will use the shell length in mm

'A' Calculate the blank values for each trial

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

```
##     pH Time._min meanBlank_ln_AlgaeLoss sdBlank_ln_AlgaeLoss
## 1  7.5        10            0.005925943                   NA
## 2  7.5        15            0.008379498          0.001565165
## 3  7.5        20            0.083124754                   NA
## 4  7.5        30            0.066622217          0.034126783
## 5  7.5        40            0.065622318                   NA
## 6  7.5        50            0.094424309                   NA
## 7  7.5        57            0.059777204                   NA
## 8  7.5        60            0.098939948                   NA
## 9  7.5        70            0.101756318                   NA
## 10 7.5        75            0.075223421                   NA
## 11 7.5        90            0.089593318          0.053841127
## 12 8.0        10            0.015870864          0.001415206
## 13 8.0        15            0.081890972                   NA
## 14 8.0        30            0.112614144                   NA
## 15 8.0        40            0.067635835                   NA
## 16 8.0        45            0.076359279          0.067200682
## 17 8.0        47            0.046063258                   NA
## 18 8.0        50            0.040346822                   NA
## 19 8.0        57            0.044518856                   NA
## 20 8.0        60            0.158305235                   NA
## 21 8.0        70            0.024924890          0.015757351
## 22 8.0        75            0.112731849                   NA
## 23 8.0        90            0.068248862          0.056330183
##    seBlank_ln_AlgaeLoss n
## 1                    NA 1
## 2           0.001106739 2
## 3                    NA 1
## 4           0.019703108 3
## 5                    NA 1
## 6                    NA 1
## 7                    NA 1
## 8                    NA 1
## 9                    NA 1
## 10                   NA 1
## 11          0.031085189 3
## 12          0.001000702 2
## 13                   NA 1
## 14                   NA 1
## 15                   NA 1
## 16          0.047518058 2
## 17                   NA 1
## 18                   NA 1
## 19                   NA 1
## 20                   NA 1
## 21          0.011142130 2
## 22                   NA 1
## 23          0.028165092 4
```

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

```
##    pH Time._min meanBlank_ln_AlgaeLoss sdBlank_ln_AlgaeLoss
## 1 7.5        20             0.07899175           0.09373166
## 2 7.5        30             0.01338708                   NA
## 3 7.5        40             0.14527004                   NA
## 4 7.5        60             0.06080047           0.07900592
## 5 8.0        20             0.06528042           0.03669390
## 6 8.0        30             0.04766821           0.00226676
## 7 8.0        40             0.12783337                   NA
## 8 8.0        60             0.06835669           0.03782168
##   seBlank_ln_AlgaeLoss n
## 1          0.066278294 2
## 2                   NA 1
## 3                   NA 1
## 4          0.045614090 3
## 5          0.021185232 3
## 6          0.001602841 2
## 7                   NA 1
## 8          0.021836358 3
```

'A' Call the blank values for each trial - continued....

- Call the blank value **specifically for the final timepoint** of the clearance rate trial(s)!
- this value will be called in the for loop when calculating CR in following cluster(s)

```
# 20210914 DATA ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Blank_914_pH7.5 <- (A_914_BlankMeans %>% dplyr::filter(pH == 7.5) %>% dplyr::arrange(desc(Time._min)))$m
Blank_914_pH8.0 <- (A_914_BlankMeans %>% dplyr::filter(pH == 8.0) %>% dplyr::arrange(desc(Time._min)))$m

# 20210930 DATA ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Blank_930_pH7.5 <- (A_930_BlankMeans %>% dplyr::filter(pH == 7.5) %>% dplyr::arrange(desc(Time._min)))$m
Blank_930_pH8.0 <- (A_930_BlankMeans %>% dplyr::filter(pH == 8.0) %>% dplyr::arrange(desc(Time._min)))$m
```

20210914 Clearance Rate data

```
df_total.914              <- data.frame() # start dataframe
for (i in 1:nrow(loop_914)) {
  dat <- ClearRate_Master.914 %>%
    dplyr::filter(uniq_Identifier == loop_914[i,]) %>%
    dplyr::arrange(Time._min)
  C0 <- (dat %>% dplyr::filter(Time._min == 0))$Cells_ml[1]
  dat2 <- dat %>%
    #dplyr::mutate(diff = as.numeric(Time._min) - lag(as.numeric(Time._min), default = first(as.numeric
    dplyr::filter(Time._min %in% 90) %>%
    dplyr::mutate(Blank = if(pH == 7.5) Blank_914_pH7.5 else Blank_914_pH8.0 ) %>%
    dplyr::mutate(AlgaeLossRatio = C0 / as.numeric(Cells_ml) ) %>%
    dplyr::filter(!AlgaeLossRatio < 1) %>%
    dplyr::mutate(ln_AlgaeLossRatio = ln(AlgaeLossRatio)) %>%
   # dplyr::mutate(ClearanceRate =  ( (25/1000) / (diff/60)  * # V / t == Volume of the vessel (in Lite
   #                                  ( ln_AlgaeLossRatio ) / Length.um. ) ) %>%
    dplyr::mutate(ClearanceRate_L_hour_mm = ( (25/1000) * # V / t == Volume of the vessel (in Liters as
                                     (( ln_AlgaeLossRatio /  (as.numeric(Time._min)/60) - Blank))) /
```

```
    dplyr::mutate(Time_period = paste((as.numeric(substr(Time._min,1,1)) -1), "0-", Time._min, sep ='')
    dat2OM <- dat2 %>% dplyr::filter(!Time._min == 0)

  if (nrow(dat2OM) > 0) {
    ClearRate.table           <- data.frame(matrix(nrow = nrow(dat2OM), ncol = 9)) # create dataframe t
    colnames(ClearRate.table) <- c('Date', 'ID', 'pH', 'Replicate', 'Num','Run','Time_period', 'AlgaeLos

    ClearRate.table$Date                       <- dat2OM$Date
    ClearRate.table$ID                         <- loop_914[i,]
    ClearRate.table$pH                         <- gsub("_.*", "\\1", ClearRate.table$ID)
    ClearRate.table$Replicate                  <- gsub("^(?:[^_]+_){4}([^_]+).*", "\\1", ClearRate.table
    ClearRate.table$Num                        <- gsub("^(?:[^_]+_){6}([^_]+).*", "\\1", ClearRate.table
    ClearRate.table$Run                        <- gsub("^(?:[^_]+_){2}([^_]+).*", "\\1", ClearRate.table
    ClearRate.table$Time_period                <- paste((as.numeric(substr(dat2OM$Time._min,1,1)) -1),
    ClearRate.table$AlgaeLossRatio             <- dat2OM$AlgaeLossRatio
    ClearRate.table$ClearanceRate_L_hour_mm <- dat2OM$ClearanceRate_L_hour_mm

    df       <- data.frame(ClearRate.table) # name dataframe for this single row
    df_total.914 <- rbind(df_total.914,df) #bind to a cumulative list dataframe
    #print(df_total.914) # print to monitor progress
  }
  else {}
}

ClearRates_914_Means <- df_total.914 %>%
  #dplyr::filter(!ClearanceRate_L_hour_meter %in% '-Inf') %>%
  dplyr::group_by(pH) %>%
  dplyr::summarise(
    meanCR = mean(ClearanceRate_L_hour_mm),
    sdCR   = sd(ClearanceRate_L_hour_mm),
    seCR   = sd(ClearanceRate_L_hour_mm) / sqrt(length(ClearanceRate_L_hour_mm)),
    n = n()) %>%
  na.omit()

print(ClearRates_914_Means)
```

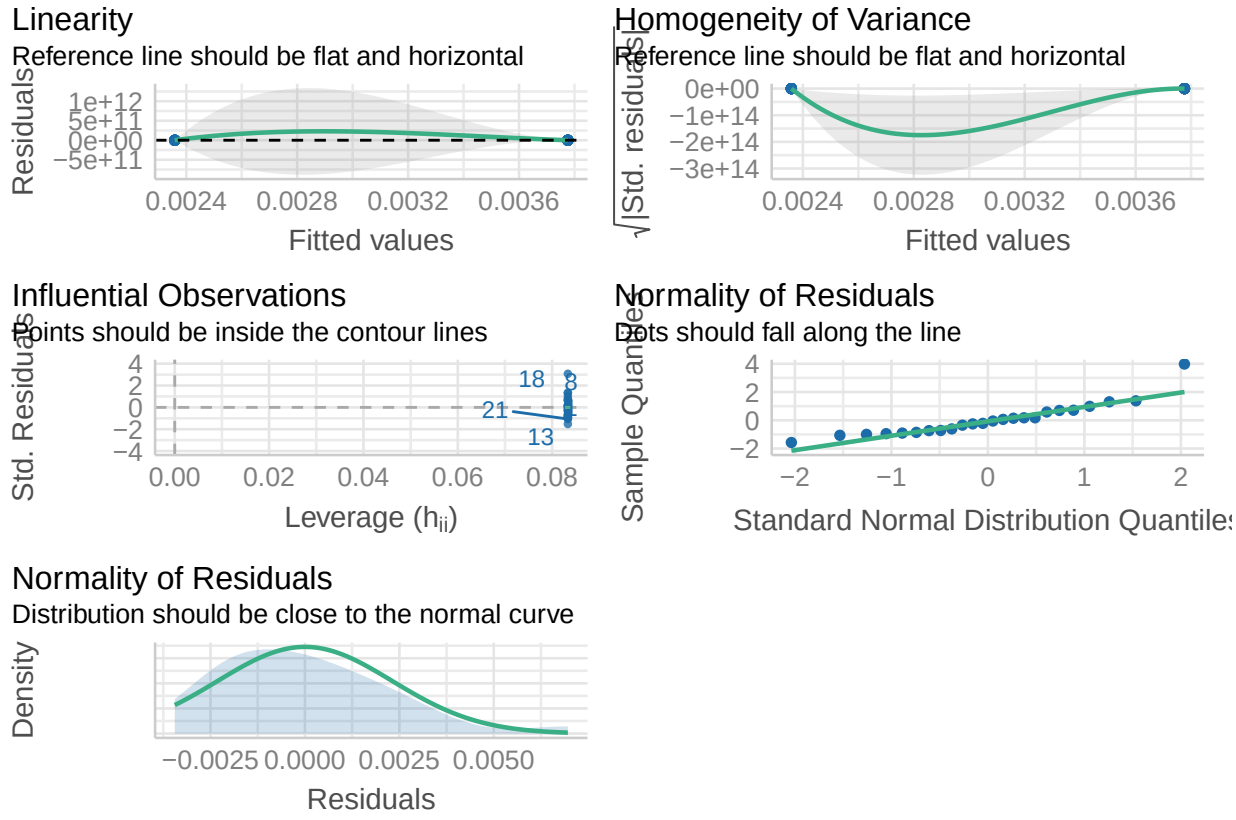
```
## # A tibble: 2 x 5
##   pH     meanCR    sdCR     seCR     n
##   <chr>   <dbl>   <dbl>    <dbl> <int>
## 1 7.5   0.00378 0.00288 0.000831    12
## 2 8     0.00236 0.00170 0.000490    12
```

```
# summary(lmer(meanCR~pH+ (1|Time_period), data=ClearRates_914_Means))
# summary(aov(lm(meanCR~pH*Time_period, data=ClearRates_914_Means)))


mod914CR <- aov(lm(ClearanceRate_L_hour_mm~ pH , data = (df_total.914 %>%  dplyr::filter(!ClearanceRate
pander(summary(mod914CR), style='rmarkdown')
```

Table 1: Analysis of Variance Model

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
pH	1	1.206e-05	1.206e-05	2.158	0.156
Residuals	22	0.0001229	5.588e-06	NA	NA

```
check_model(mod914CR) # observe the diagnostics of the model
```




```
shapiro.test(residuals(mod914CR)) # non normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(mod914CR)
## W = 0.92006, p-value = 0.05859
```

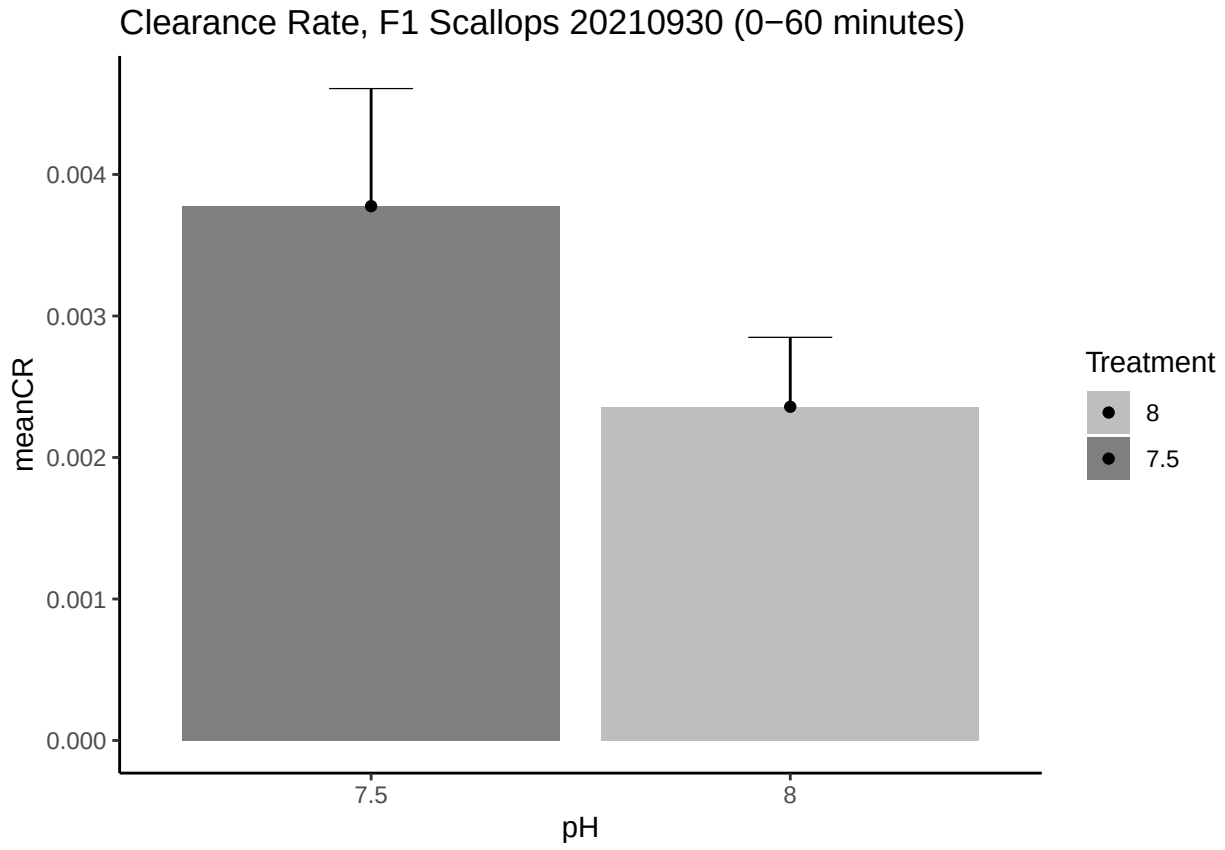
```
leveneTest(mod914CR) # good
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = median)
```

```
##       Df F value Pr(>F)
## group  1  1.2472 0.2761
##       22
```

```
#summary(lmer(ClearanceRate_L_hour_mm~ pH + (1|Replicate), data = (df_total.914 %>%  dplyr::filter(!Cle
```

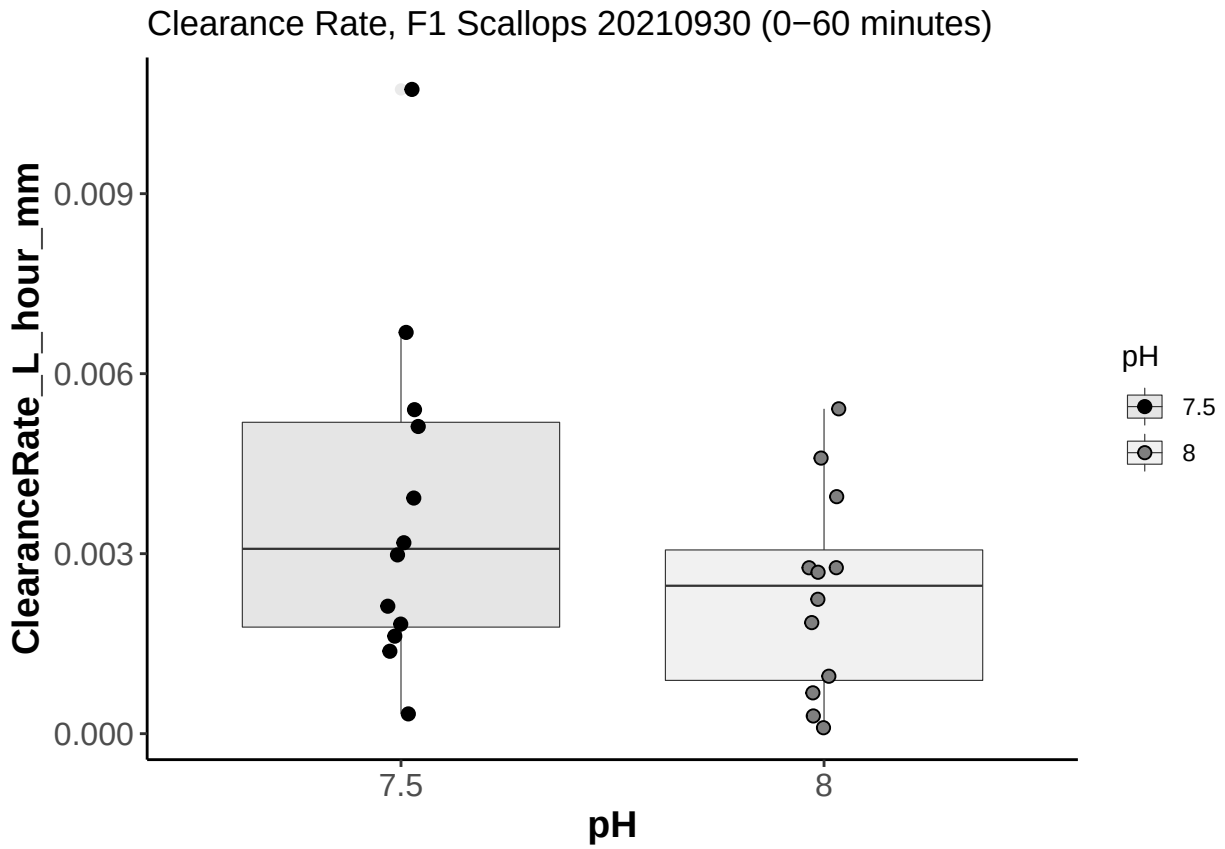
```
ClearRates_914_Means %>%
  #dplyr::filter(!Time_period %in% c('40-50', '50-60')) %>%
  ggplot(aes(x=pH , y=meanCR, fill = pH)) +
  geom_bar(position=position_dodge(), aes(y=meanCR), stat="identity", alpha=0.5) +
  scale_fill_manual("Treatment", values = c("8" = "grey50", "7.5" = "black")) +
  geom_errorbar(position=position_dodge(width=0.9), aes(ymin=meanCR+seCR, ymax=meanCR+seCR), width=0.2,
  geom_linerange(aes(ymin = meanCR, ymax = meanCR+seCR)) +
  geom_point(position=position_dodge(width=0.9), aes(y=meanCR)) +
  theme_classic() +
  ggtitle("Clearance Rate, F1 Scallops 20210930 (0-60 minutes)") # +
```




```
  # facet_wrap(~Time_period)
```

```
df_total.914 %>%
  #dplyr::filter(!ClearanceRate_L_hour_mm %in% '-Inf') %>%
  #dplyr::filter(!Time_period %in% c('40-50', '50-60')) %>%
```

```
  ggplot(aes(pH , ClearanceRate_L_hour_mm , fill = pH)) +
  theme(panel.grid=element_blank()) +
  geom_boxplot(size=0.2, alpha=0.1, aes(fill=pH)) +
  scale_fill_manual(values=c("black", "grey50")) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.1)) +
  theme_classic() +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold")) +
  # labs(title = "F1 Scallops: Slope algae cells/time - feeding rate trials 20210930",
  #      y = expression(Slope~"="~absolute~value~"("~Live~algae~cells~mL^{-1}~hour^{-1}~")"),
  #      x = expression(italic(p)*CO[2]~Treatment~"("~mu*atm~")")) +
  ggtitle("Clearance Rate, F1 Scallops 20210930 (0-60 minutes)")
```



20210930 Clearance Rate data

```
df_total.930              <- data.frame() # start dataframe
for (i in 1:nrow(loop_930)) {
  dat <- ClearRate_Master.930 %>%
    dplyr::filter(uniq_Identifier == loop_930[i,]) %>%
    dplyr::arrange(Time._min)
  C0 <- (dat %>% dplyr::filter(Time._min == 0))$Cells_ml[1]
  dat2 <- dat %>%
    #dplyr::mutate(diff = as.numeric(Time._min) - lag(as.numeric(Time._min), default = first(as.numeric
```

```
    dplyr::filter(Time._min %in% 60) %>%
    dplyr::mutate(Blank = if(pH == 7.5) Blank_930_pH7.5 else Blank_930_pH8.0 ) %>%
    dplyr::mutate(AlgaeLossRatio = C0 / as.numeric(Cells_ml) ) %>%
    dplyr::filter(!AlgaeLossRatio < 1) %>%
    dplyr::mutate(ln_AlgaeLossRatio = ln(AlgaeLossRatio)) %>%
    # dplyr::mutate(ClearanceRate =  ( (25/1000) / (diff/60)  * # V / t == Volume of the vessel (in Lit
    #                                  ( ln_AlgaeLossRatio ) / Length.um. ) ) %>%
    dplyr::mutate(ClearanceRate_L_hour_mm = ( (25/1000) * # V / t == Volume of the vessel (in Liters as
                                            (( ln_AlgaeLossRatio /  (as.numeric(Time._min)/60) - Bla
    dplyr::mutate(Time_period = paste((as.numeric(substr(Time._min,1,1)) -1), "0-", Time._min, sep ='')
  # dat2OM <- dat2 %>% dplyr::filter(!Time._min == 0)

    if (nrow(dat2) > 0) {
    ClearRate.table           <- data.frame(matrix(nrow = nrow(dat2), ncol = 10)) # create dataframe to
    colnames(ClearRate.table) <- c('Date', 'ID', 'pH', 'Fed_Unfed', 'Replicate', 'Num','Run','Time_perio

    ClearRate.table$Date                        <- dat2$Date
    ClearRate.table$ID                          <- loop_930[i,]
    ClearRate.table$pH                          <- gsub("_.*", "\\1", ClearRate.table$ID)
    ClearRate.table$Fed_Unfed                   <- gsub("^(?:[^_]+_){1}([^_]+).*", "\\1", ClearRate.table
    ClearRate.table$Replicate                   <- gsub("^(?:[^_]+_){5}([^_]+).*", "\\1", ClearRate.table
    ClearRate.table$Num                         <- gsub("^(?:[^_]+_){7}([^_]+).*", "\\1", ClearRate.table
    ClearRate.table$Run                         <- gsub("^(?:[^_]+_){3}([^_]+).*", "\\1", ClearRate.table
    ClearRate.table$Time_period                 <- paste((as.numeric(substr(dat2$Time._min,1,1)) -1), "0
    ClearRate.table$AlgaeLossRatio              <- dat2$AlgaeLossRatio
    ClearRate.table$ClearanceRate_L_hour_mm     <- dat2$ClearanceRate_L_hour_mm

    df       <- data.frame(ClearRate.table) # name dataframe for this single row
    df_total.930 <- rbind(df_total.930,df) #bind to a cumulative list dataframe
    #print(df_total.930) # print to monitor progress
    }
    else {}
}
```

```
## Warning in if (pH == 7.5) Blank_930_pH7.5 else Blank_930_pH8.0: the condition
## has length > 1 and only the first element will be used
```

```
## Warning in if (pH == 7.5) Blank_930_pH7.5 else Blank_930_pH8.0: the condition
## has length > 1 and only the first element will be used
```

```
ClearRates_930_Means <- df_total.930 %>%
  dplyr::filter(!ClearanceRate_L_hour_mm  < 0) %>%
  dplyr::group_by(pH, Fed_Unfed, Time_period) %>%
  dplyr::summarise(
    meanCR = mean(ClearanceRate_L_hour_mm),
    sdCR   = sd(ClearanceRate_L_hour_mm),
    seCR   = sd(ClearanceRate_L_hour_mm) / sqrt(length(ClearanceRate_L_hour_mm)),
    n = n()) %>%
  na.omit()
```

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

```
ClearRates_930_Means
```

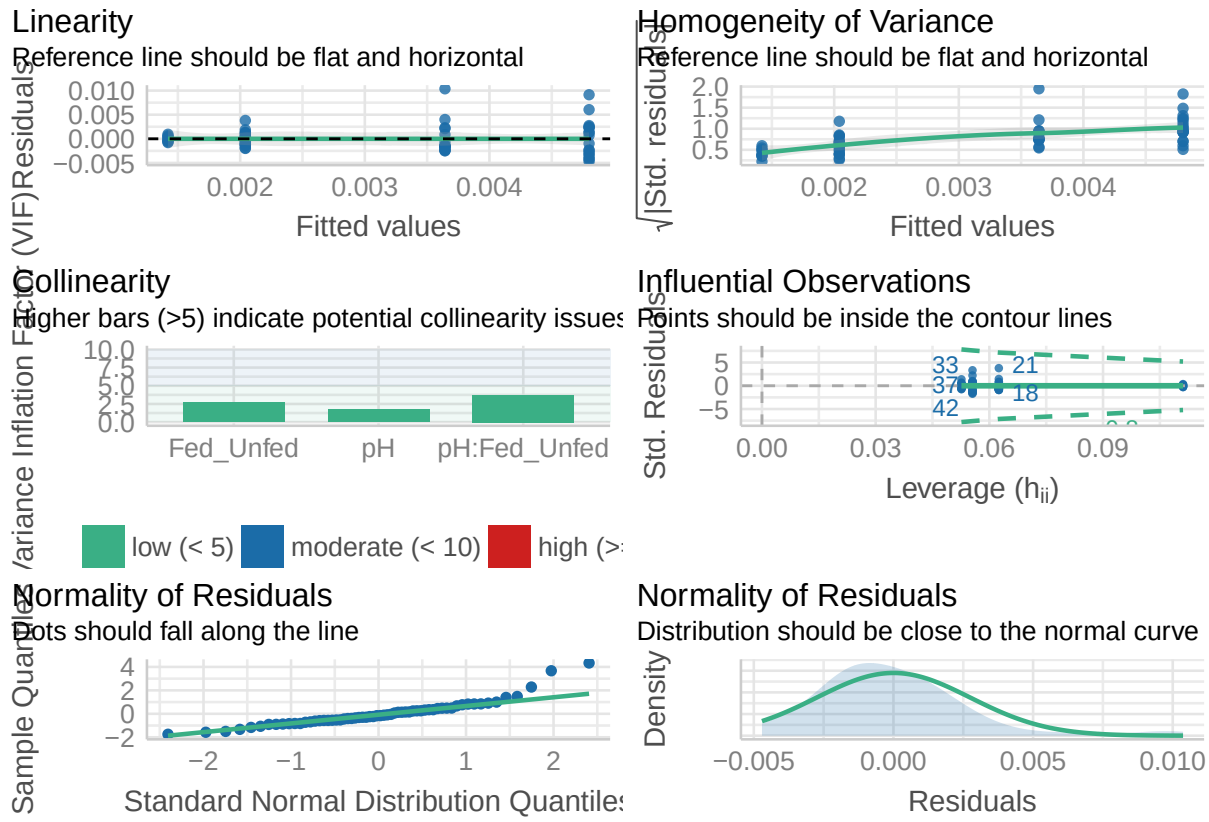
```
## # A tibble: 4 x 7
## # Groups:   pH, Fed_Unfed [4]
##   pH    Fed_Unfed Time_period  meanCR     sdCR     seCR     n
##   <chr> <chr>     <chr>         <dbl>    <dbl>    <dbl> <int>
## 1 7.5   fed       50-60       0.00364 0.00339  0.000846    16
## 2 7.5   unfed     50-60       0.00142 0.000604 0.000201     9
## 3 8     fed       50-60       0.00480 0.00383  0.000903    18
## 4 8     unfed     50-60       0.00204 0.00146  0.000334    19
```

```
# summary(lmer(meanCR~pH*Fed_Unfed + (1|Time_period), data=ClearRates_930_Means))
# summary(aov(lm(meanCR~pH*Fed_Unfed, data=ClearRates_930_Means)))

mod930CR <- aov(lm(ClearanceRate_L_hour_mm~ pH * Fed_Unfed, data = (df_total.930 %>%  dplyr::filter(!Cl
pander(summary(mod930CR), style='rmarkdown')
```

Table 2: Analysis of Variance Model

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
pH	1	4.353e-06	4.353e-06	0.5458	0.463
Fed_Unfed	1	9.774e-05	9.774e-05	12.26	0.0008984
pH:Fed_Unfed	1	1.018e-06	1.018e-06	0.1277	0.7222
Residuals	58	0.0004625	7.974e-06	NA	NA

```
check_model(mod930CR) # observe the diagnostics of the model
```




```
shapiro.test(residuals(mod930CR)) # non normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(mod930CR)
## W = 0.89661, p-value = 7.692e-05
```

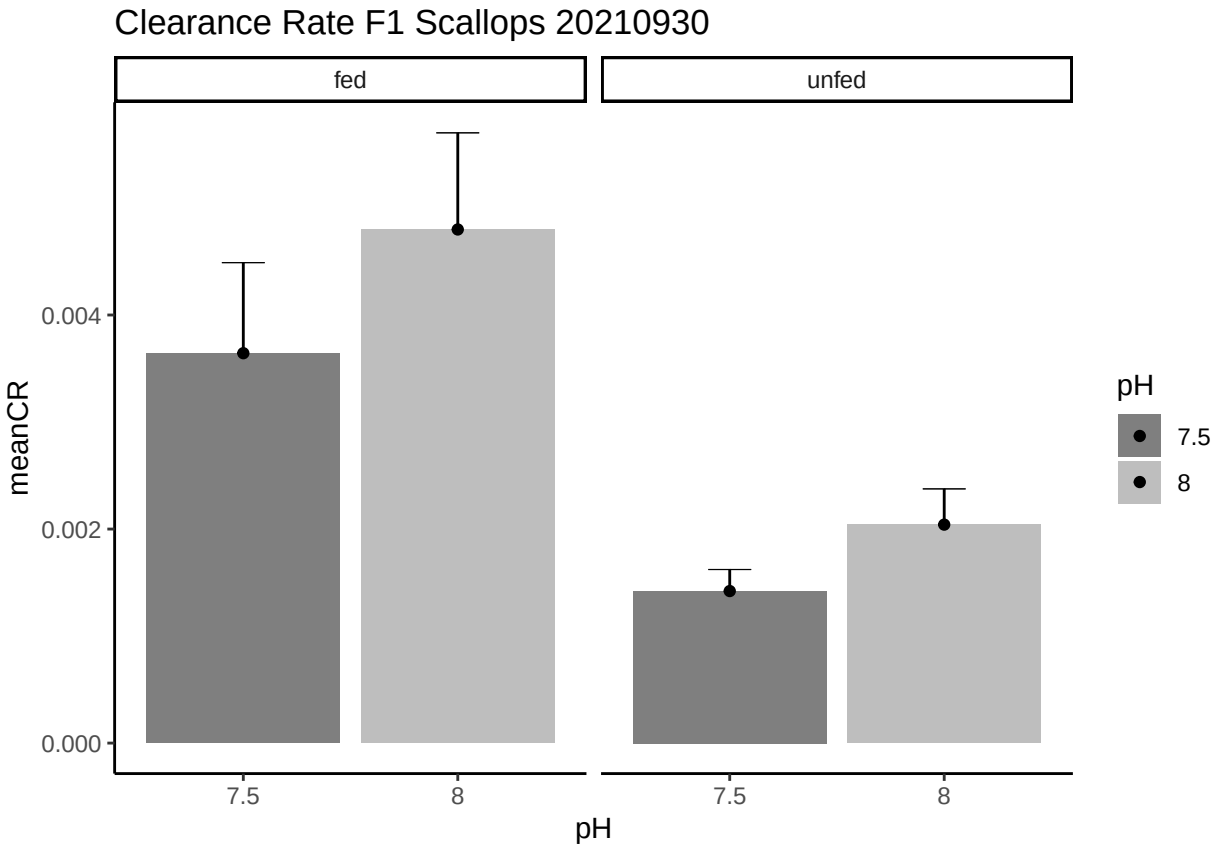
```
leveneTest(mod930CR) # good
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value  Pr(>F)
## group  3  4.7854 0.00478 **
##       58
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# summary(lmer(ClearanceRate_L_hour_mm~ pH * Fed_Unfed + (1|Replicate), data = (df_total.930 %>%  dplyr
```

```
ClearRates_930_Means$pH_feed <- paste(ClearRates_930_Means$pH, ClearRates_930_Means$Fed_Unfed, sep='_')
ClearRates_930_Means %>%
#dplyr::filter(!Time_period %in% c('40-50', '50-60')) %>%
```

```
ggplot(aes(x=pH , y=meanCR, fill = pH)) +
  geom_bar(position=position_dodge(), aes(y=meanCR), stat="identity", alpha=0.5) +
  scale_fill_manual(values=c("black", "grey50")) +
  geom_errorbar(position=position_dodge(width=0.9), aes(ymin=meanCR+seCR, ymax=meanCR+seCR), width=0.2,
  geom_linerange(aes(ymin = meanCR, ymax = meanCR+seCR)) +
  geom_point(position=position_dodge(width=0.9), aes(y=meanCR)) +
  theme_classic() +
  ggtitle("Clearance Rate F1 Scallops 20210930") +
  facet_wrap(~ Fed_Unfed)
```




```
df_total.930 %>%
  dplyr::filter(!ClearanceRate_L_hour_mm  < 0) %>%
  #dplyr::filter(!ClearanceRate_L_hour_mm %in% '-Inf') %>%
  #dplyr::filter(!Time_period %in% c('40-50', '50-60')) %>%
  ggplot(aes(pH , ClearanceRate_L_hour_mm , fill = pH)) +
  theme(panel.grid=element_blank()) +
  geom_boxplot(size=0.2, alpha=0.1, aes(fill=pH)) +
  scale_fill_manual(values=c("black", "grey50")) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.1)) +
  theme_classic() +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14,face="bold")) +
  # labs(title = "F1 Scallops: Slope algae cells/time - feeding rate trials 20210930",
  #      y = expression(Slope~"="~absolute~value~"("~Live~algae~cells~mL^{-1}~hour^{-1}~")"),
  #      x = expression(italic(p)*CO[2]~Treatment~"("~mu*atm~")")) +
```

```
ggtitle("Clearance Rate F1 Scallops 20210930") +
facet_wrap(~ Fed_Unfed)
```