

F1 Bay Scallops: High vs. Low food availability under OA conditions

Samuel Gurr

10/12/2021

Last updates: october 6, 2021

Load libraries

Set Path

Time Series Analysis for Length Data

#Line plot for looking at meas shell length overtime (Growth) Color codes for ggPlot2: <http://sape.inf.usi.ch/quick-reference/ggplot2/colour>

```
#First make calculations for means and standard error
```

```
stL <- summarySE(df, measurevar="Length", groupvars=c("Day", "trt"))  
stL
```

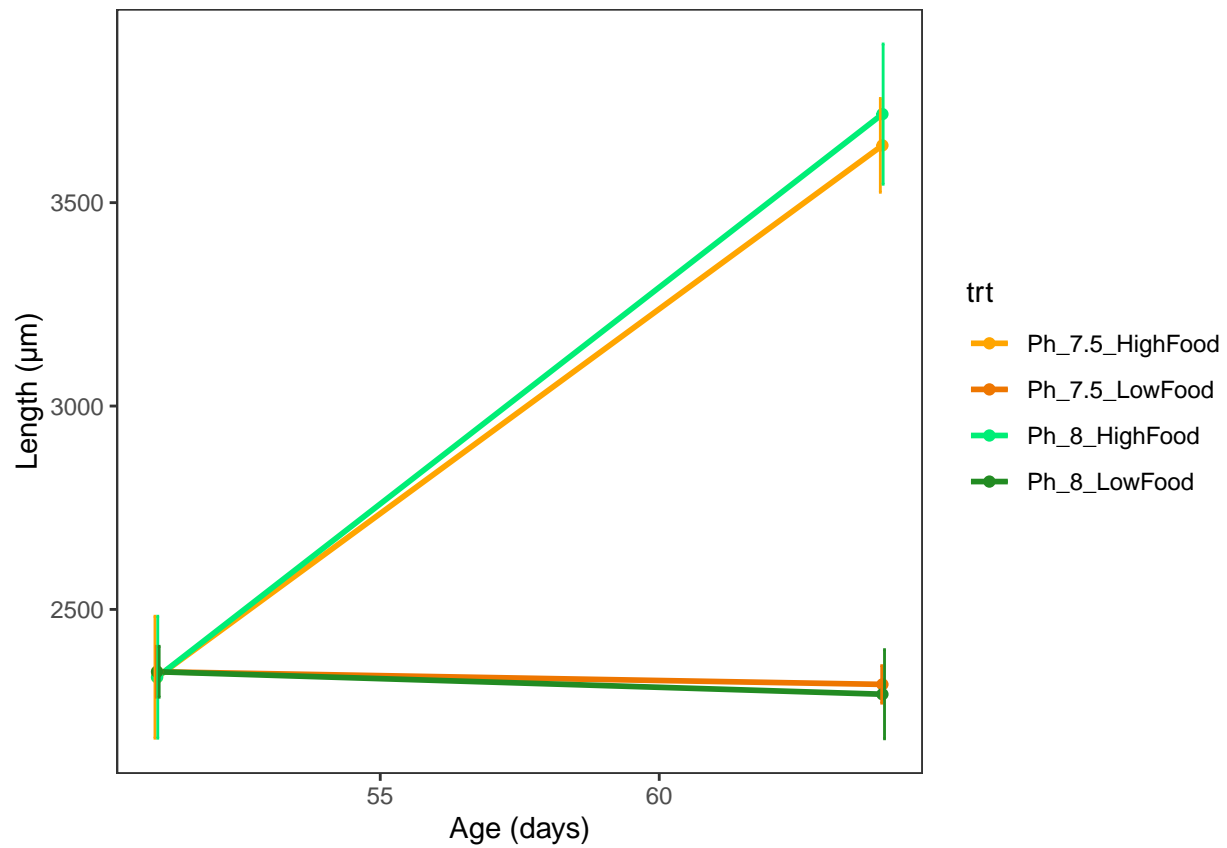
```
##   Day          trt N   Length      sd      se      ci  
## 1  51 Ph_7.5_HighFood 4 2333.317 299.90639 149.95320 477.2180  
## 2  51 Ph_7.5_LowFood 4 2346.736 125.10392  62.55196 199.0683  
## 3  51 Ph_8_HighFood 4 2333.317 299.90639 149.95320 477.2180  
## 4  51 Ph_8_LowFood 4 2346.736 125.10392  62.55196 199.0683  
## 5  64 Ph_7.5_HighFood 4 3640.847 230.56447 115.28223 366.8795  
## 6  64 Ph_7.5_LowFood 4 2315.662  92.12134  46.06067 146.5856  
## 7  64 Ph_8_HighFood 4 3717.715 344.79997 172.39998 548.6537  
## 8  64 Ph_8_LowFood 4 2291.467 219.08345 109.54172 348.6107
```

```
#stL$Treatment <- factor(stL$trt, # Relevel group factor  
# levels = c("Low OA", "Moderate OA", "High OA"))
```

```
## Use geom_line and geom_point to plot over time
```

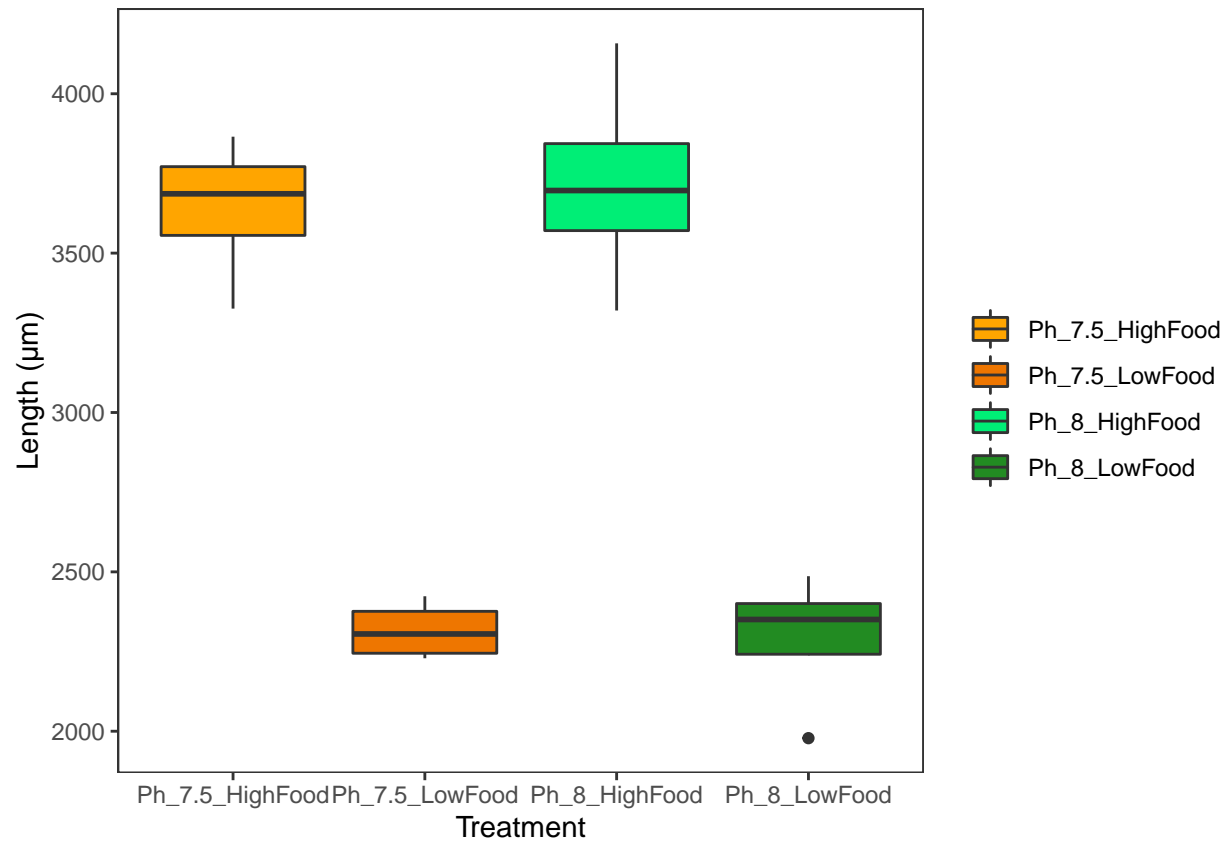
```
pPlot1<-ggplot(data=stL, aes(x=Day, y=Length, color=trt)) +  
  geom_line( stat = "identity", size=1.0)+  
  geom_point()+  
  scale_color_manual(values=c( "orange1","darkorange2","springgreen2","forestgreen"))+  
  geom_errorbar(aes(ymin=Length-se, ymax=Length+se), width=.2,  
                position=position_dodge(.1))+  
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+  
  scale_x_continuous(name ="Age (days)") +
```

```
scale_y_continuous(name = "Length (μm)")
pPlot1
```



```
data_64<- df%>%
  filter(Day=="64")

pBoxPlot<-ggplot(data=data_64, aes(x=trt, y=Length, fill=trt)) +
  geom_boxplot()+scale_fill_manual(values=c("orange1","darkorange2","springgreen2","forestgreen"))+
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), legend.title = "trt")
  #ggtitle("Age = 8 days")+
  scale_y_continuous(name = "Length (μm)") +
  scale_x_discrete(name = "Treatment")
pBoxPlot
```



```
library(lmerTest)
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'Matrix'
```

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

```
##
```

```
## expand, pack, unpack
```

```
##
```

```
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:nlme':
```

```
##
```

```
## lmList
```

```
##
```

```
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##      lmer

## The following object is masked from 'package:stats':
##
##      step
```

```
library(lme4)
library(pander)
library(performance)

data_64<- df%>%
  filter(Day=="64")

data_64$Fed_Unfed <- as.factor(gsub("^(:[^\_]+\_){2}([^\_]+).*", "\\1", data_64$trt))
levels(data_64$Fed_Unfed) <- c("High food", "Low food")

data_64$pCO2 <- as.factor(gsub("^(:[^\_]+\_){1}([^\_]+).*", "\\1", data_64$trt))
levels(data_64$pCO2) <- c("High", "Low")

SizeMod_D64 <- aov(lm(Length~pCO2*Fed_Unfed, data = data_64) )
pander(summary(SizeMod_D64), style='rmarkdown')
```

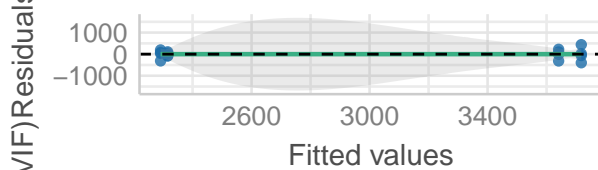
Table 1: Analysis of Variance Model

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
pCO2	1	2774	2774	0.04856	0.8293
Fed_Unfed	1	7570390	7570390	132.5	7.676e-08
pCO2:Fed_Unfed	1	10214	10214	0.1788	0.6799
Residuals	12	685593	57133	NA	NA

```
check_model(SizeMod_D64) # observe the diagnostics of the model
```

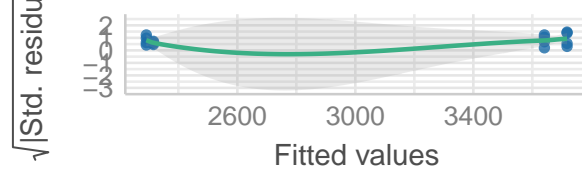
Linearity

Reference line should be flat and horizontal



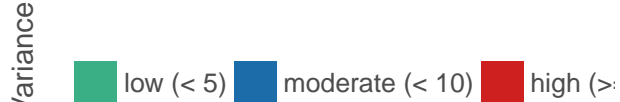
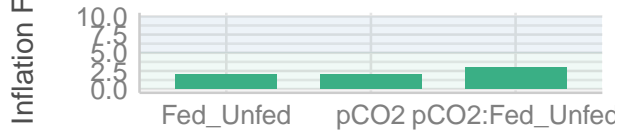
Homogeneity of Variance

Reference line should be flat and horizontal



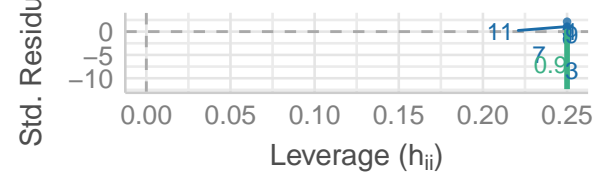
Collinearity

Higher bars (>5) indicate potential collinearity issue



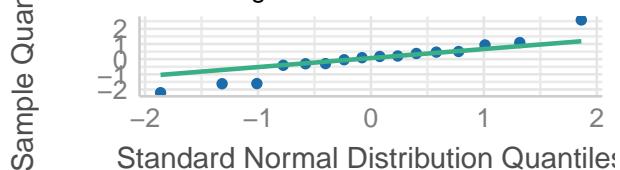
Influential Observations

Points should be inside the contour lines



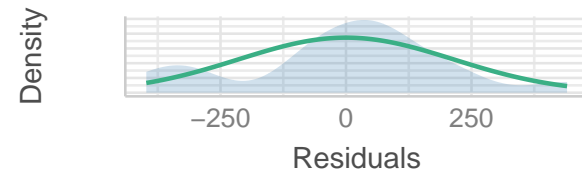
Normality of Residuals

Points should fall along the line



Normality of Residuals

Distribution should be close to the normal curve



```
shapiro.test(residuals(SizeMod_D64)) # non normal
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(SizeMod_D64)
## W = 0.95424, p-value = 0.5596
```

```
leveneTest(SizeMod_D64) # good
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 3  0.6666 0.5885
##      12
```

```
TukeyHSD(SizeMod_D64)
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lm.Length ~ pCO2 * Fed_Unfed, data = data_64))
##
## $pCO2
##      diff      lwr      upr      p adj
```

```
## Low-High 26.33676 -234.0581 286.7316 0.8292886
##
## $Fed_Unfed
##               diff          lwr          upr p adj
## Low food-High food -1375.717 -1636.112 -1115.322 1e-07
##
## $`pCO2:Fed_Unfed`
##               diff          lwr          upr    p adj
## Low:High food-High:High food   76.86825  -424.9231  578.6596 0.9674062
## High:Low food-High:High food -1325.18559 -1826.9770 -823.3942 0.0000239
## Low:Low food-High:High food  -1349.38032 -1851.1717 -847.5889 0.0000199
## High:Low food-Low:High food  -1402.05384 -1903.8452 -900.2625 0.0000134
## Low:Low food-Low:High food  -1426.24857 -1928.0399 -924.4572 0.0000113
## Low:Low food-High:Low food    -24.19473  -525.9861  477.5966 0.9988927
```

```
data_64 %>% dplyr::group_by(Fed_Unfed) %>% dplyr::summarise(mean = mean(Length))
```

```
## # A tibble: 2 x 2
##   Fed_Unfed mean
##   <fct>      <dbl>
## 1 High food 3679.
## 2 Low food  2304.
```

```
SurvivaleMod_D64 <- aov(lm(Survival~pCO2*Fed_Unfed, data = data_64) )
pander(summary(SurvivaleMod_D64), style='rmarkdown')
```

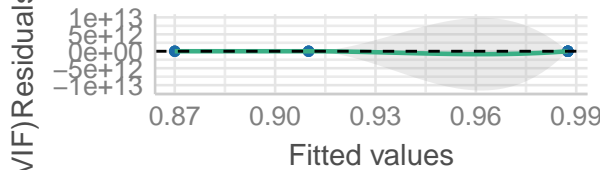
Table 2: Analysis of Variance Model

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
pCO2	1	0.0016	0.0016	0.1138	0.7417
Fed_Unfed	1	0.03802	0.03802	2.704	0.126
pCO2:Fed_Unfed	1	0.0016	0.0016	0.1138	0.7417
Residuals	12	0.1687	0.01406	NA	NA

```
check_model(SurvivaleMod_D64) # observe the diagnostics of the model
```

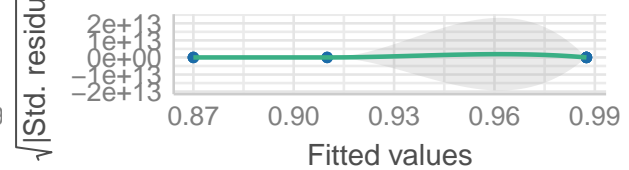
Linearity

Reference line should be flat and horizontal



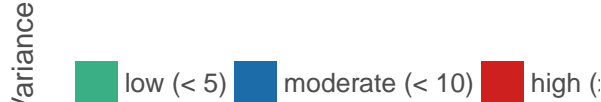
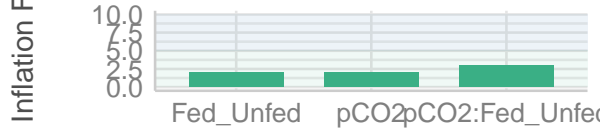
Homogeneity of Variance

Reference line should be flat and horizontal



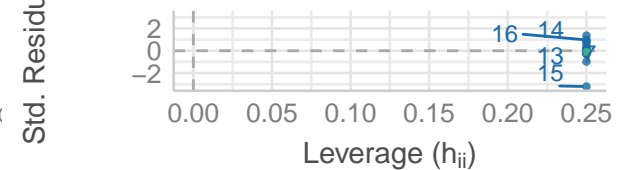
Collinearity

Higher bars (>5) indicate potential collinearity issue



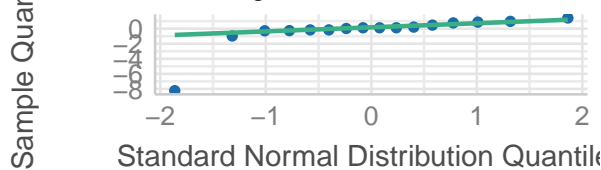
Influential Observations

Points should be inside the contour lines



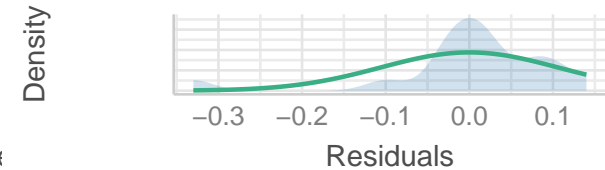
Normality of Residuals

Points should fall along the line



Normality of Residuals

Distribution should be close to the normal curve



```
shapiro.test(residuals(SurvivaleMod_D64)) # non normal
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(SurvivaleMod_D64)
## W = 0.80657, p-value = 0.003334
```

```
leveneTest(SurvivaleMod_D64) # good
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 3  0.933 0.4548
##      12
```

```
data_64 %>% dplyr::group_by(pCO2) %>% dplyr::summarise(mean = mean(Survival))
```

```
## # A tibble: 2 x 2
##   pCO2 mean
##   <fct> <dbl>
## 1 High  0.929
## 2 Low   0.949
```

```
data_64 %>% dplyr::group_by(Fed_Unfed) %>% dplyr::summarise(mean = mean(Survival), sd = sd(Survival))
```

```
## # A tibble: 2 x 3
##   Fed_Unfed mean    sd
##   <fct>      <dbl> <dbl>
## 1 High food 0.988 0.0183
## 2 Low food  0.89  0.156
```

```
data_64 %>% dplyr::group_by(trt) %>% dplyr::summarise(mean = mean(Survival))
```

```
## # A tibble: 4 x 2
##   trt          mean
##   <chr>         <dbl>
## 1 Ph_7.5_HighFood 0.988
## 2 Ph_7.5_LowFood  0.87
## 3 Ph_8_HighFood   0.988
## 4 Ph_8_LowFood    0.91
```

```
Length_64dpf <- ggplot(data_64, aes(x = factor(pCO2, level = c('Low', 'High')), Length, fill = pCO2)) +
  theme(panel.grid=element_blank()) +
  geom_boxplot(size=0.2, alpha=0.1, aes(fill=pCO2)) +
  scale_fill_manual(values=c("grey50", "white")) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.1)) +
  theme_classic() +
  ylim(2000, 4000) +
  scale_x_discrete(labels=c('~500 atm', '~800 atm')) +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14, face="bold")) +
  stat_summary(fun.y=mean, geom="point", shape=18, size=4, color="black", fill="white") +
  labs(title = "Shell Length; F1 Scallops (juveniles) 66 days post-fertilization",
       y = expression(Shell~length~"(" ~ m ~ ")"),
       x = expression(italic(p)*CO[2]~Treatment~"(" ~ *atm ~ ")")) +
  facet_wrap(~factor(Fed_Unfed, level = c('Low food', 'High food')))
```

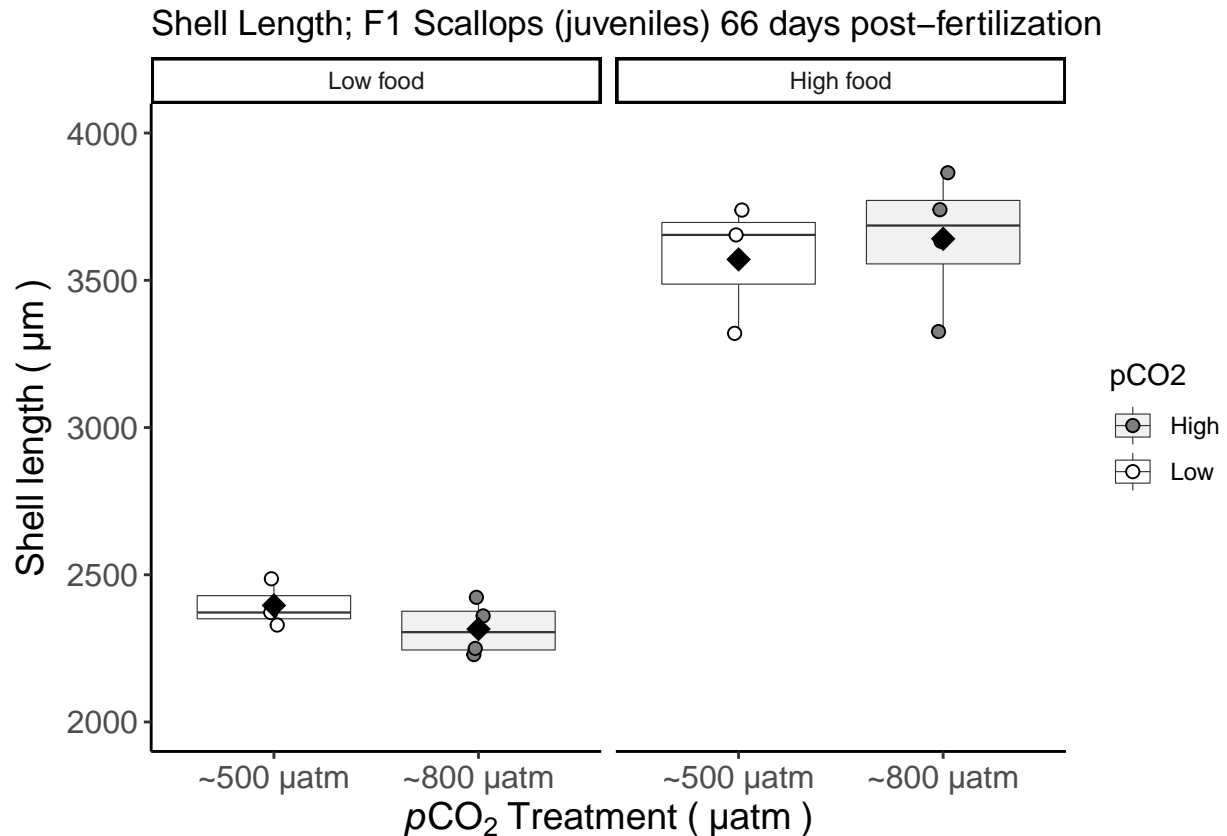
```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
Length_64dpf
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 2 rows containing non-finite values (stat_summary).
```

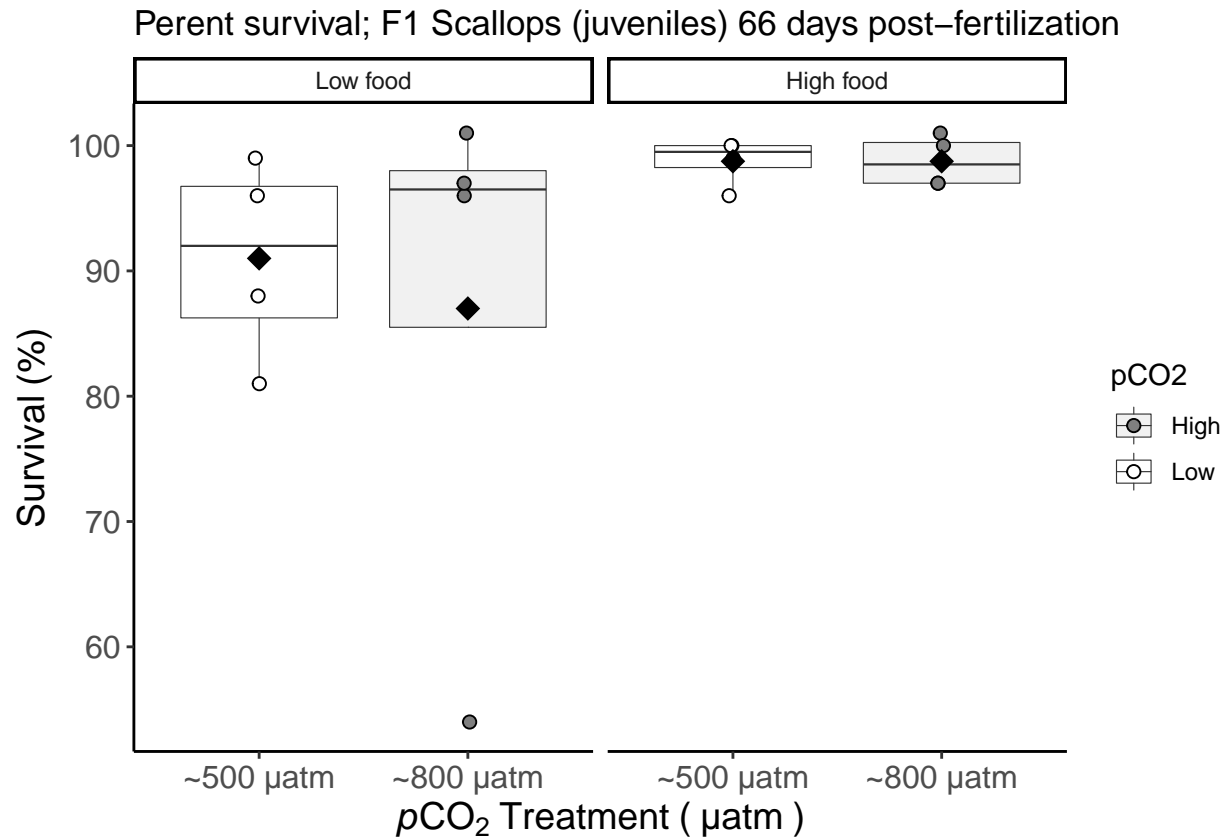
```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
Survival_64DPF <- ggplot(data_64, aes(x = factor(pCO2, level = c('Low', 'High')), (Survival*100) , fill =
  theme(panel.grid=element_blank()) +
  geom_boxplot(size=0.2, alpha=0.1, aes(fill=pCO2)) +
  scale_fill_manual(values=c("grey50","white")) +
  geom_point(shape = 21, size = 2, position = position_jitterdodge(jitter.width = 0.1)) +
  theme_classic() +
  #ylim(2, 4.5) +
  scale_x_discrete(labels=c('~500 atm', '~800 atm')) +
  theme(axis.text=element_text(size=12),
    axis.title=element_text(size=14,face="bold")) +
  stat_summary(fun.y=mean, geom="point", shape=18, size=4, color="black", fill="white") +
  labs(title = "Perent survival; F1 Scallops (juveniles) 66 days post-fertilization",
    y = expression(Survival~"(%)" ),
    x = expression(italic(p)*CO[2]~Treatment~"(~ *atm~)") ) +
  facet_wrap(~factor(Fed_Unfed, level = c('Low food', 'High food')))
```

Warning: `fun.y` is deprecated. Use `fun` instead.

Survival_64DPF



```
# pdf(paste0("C:/Users/samuel.gurr/Documents/Github_repositories/Airradians_OA/RAnalysis/Output/Respira
# pdf(paste0("C:/Users/samuel.gurr/Documents/Github_repositories/Airradians_OA/RAnalysis/Output/Surviva
pdf(paste0("C:/Users/samjg/Documents/Github_repositories/Airradians_OA/RAnalysis/Output/Survival_size/2
Length_64dpf
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 2 rows containing non-finite values (stat_summary).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
## 2
```

```
# pdf(paste0("C:/Users/samuel.gurr/Documents/Github_repositories/Airradians_OA/RAnalysis/Output/Surviva
pdf(paste0("C:/Users/samjg/Documents/Github_repositories/Airradians_OA/RAnalysis/Output/Survival_size/2
Survival_64DPF
dev.off()
```

```
## pdf
## 2
```

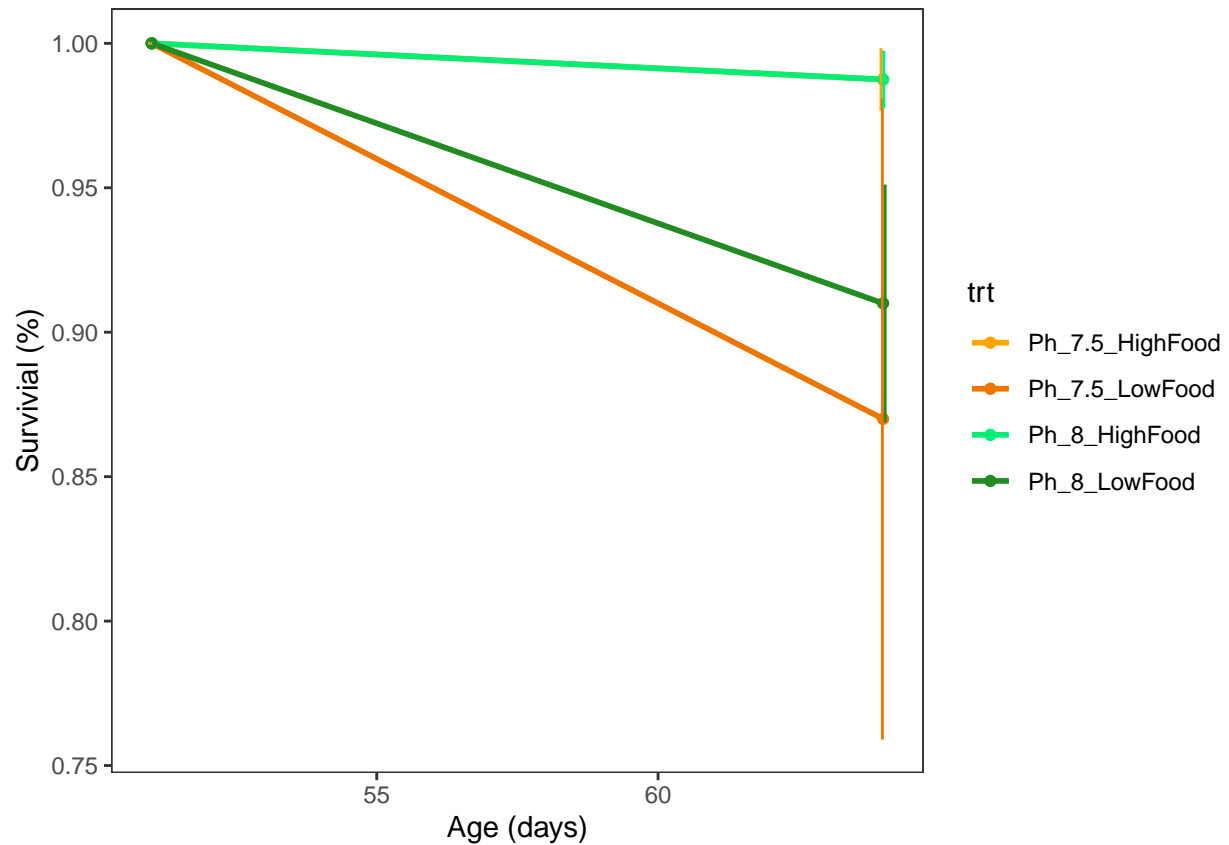
```
#First make calculations for means and standard error
stL <- summarySE(df, measurevar="Survival", groupvars=c("Day", "trt"))
stL
```

```
##   Day          trt N Survival      sd      se      ci
## 1  51 Ph_7.5_HighFood 4   1.0000 0.00000000 0.00000000 0.00000000
## 2  51 Ph_7.5_LowFood 4   1.0000 0.00000000 0.00000000 0.00000000
## 3  51 Ph_8_HighFood 4   1.0000 0.00000000 0.00000000 0.00000000
## 4  51 Ph_8_LowFood 4   1.0000 0.00000000 0.00000000 0.00000000
## 5  64 Ph_7.5_HighFood 4   0.9875 0.02061553 0.010307764 0.03280391
## 6  64 Ph_7.5_LowFood 4   0.8700 0.22105806 0.110529031 0.35175271
## 7  64 Ph_8_HighFood 4   0.9875 0.01892969 0.009464847 0.03012137
## 8  64 Ph_8_LowFood 4   0.9100 0.08124038 0.040620192 0.12927158
```

```
#na.omit(stL)

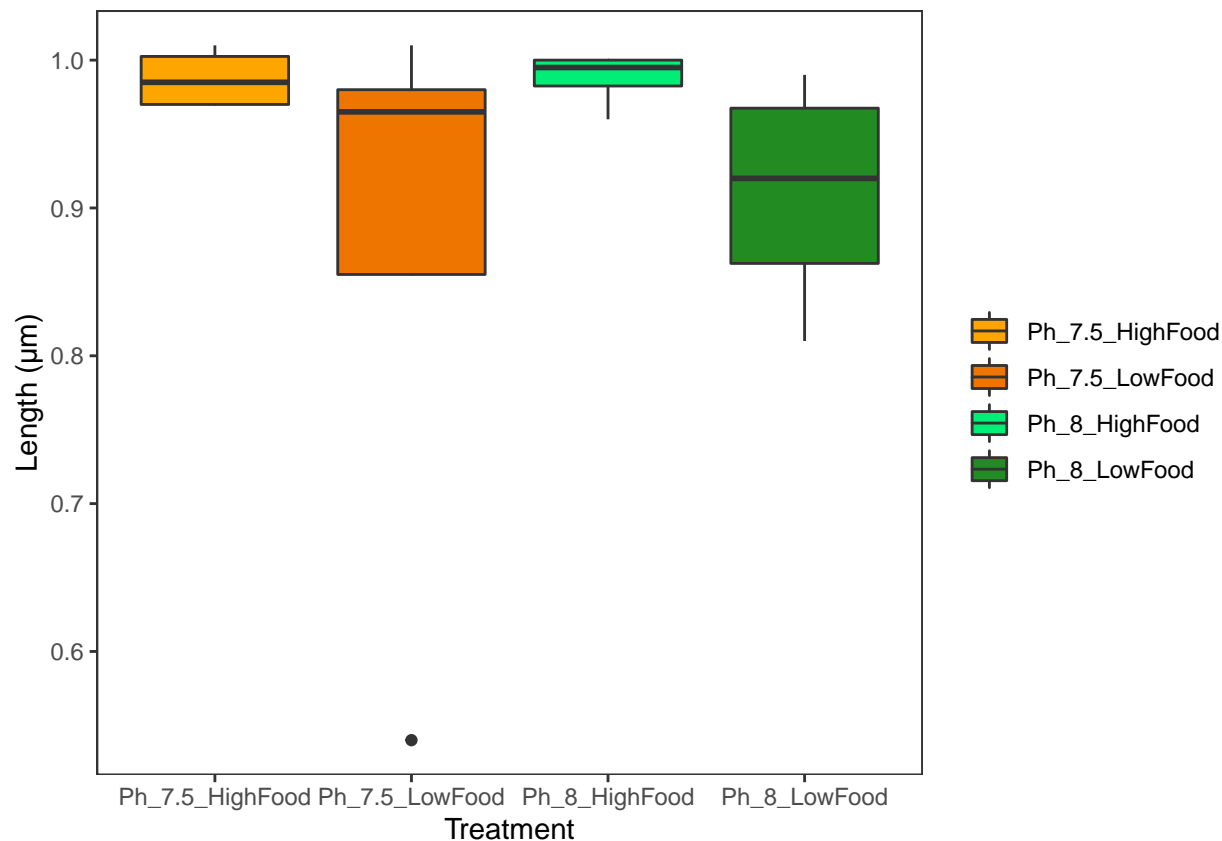
#stL$Treatment <- factor(stL$trt, # Relevel group factor
#                               levels = c("Low OA", "Moderate OA", "High OA"))

## Use geom_line and geom_point to plot over time
pPlot1<-ggplot(data=stL, aes(x=Day, y=Survival, color=trt)) +
  geom_line( stat = "identity", size=1.0)+
  geom_point()+
  scale_color_manual(values=c( "orange1","darkorange2","springgreen2","forestgreen"))+
  geom_errorbar(aes(ymin=Survival-se, ymax=Survival+se), width=.2,
                position=position_dodge(.1))+
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
  scale_x_continuous(name ="Age (days)" ) +
  scale_y_continuous(name ="Survivial (%)")
pPlot1
```



```
data_64<- df%>%
  filter(Day=="64")

pBoxPlot<-ggplot(data=data_64, aes(x=trt, y=Survival, fill=trt)) +
  geom_boxplot()+scale_fill_manual(values=c("orange1","darkorange2","springgreen2","forestgreen"))+
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), legend.title = "trt")
#ggtitle("Age = 8 days")+
  scale_y_continuous(name = "Length (µm)")+
  scale_x_discrete(name = "Treatment")
pBoxPlot
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] performance_0.7.3  pander_0.6.4      lmerTest_3.1-3     lme4_1.1-27.1
## [5] Matrix_1.3-4       Rmisc_1.5          plyr_1.8.6         lattice_0.20-44
## [9] agricolae_1.3-5    multcompView_0.1-8 nlme_3.1-152       car_3.0-11
## [13] carData_3.0-4      forcats_0.5.1     stringr_1.4.0      dplyr_1.0.7
## [17] purrr_0.3.4        readr_2.0.1       tidyr_1.1.3        tibble_3.1.4
## [21] tidyverse_1.3.1    ggplot2_3.3.5
```

```
##
## loaded via a namespace (and not attached):
## [1] minqa_1.2.4           colorspace_2.0-2      ellipsis_0.3.2
## [4] rio_0.5.27            fs_1.5.0              rstudioapi_0.13
## [7] farver_2.1.0          ggrepel_0.9.1         fansi_0.5.0
## [10] lubridate_1.7.10      xml2_1.3.2            splines_4.1.1
## [13] knitr_1.34            jsonlite_1.7.2        nloptr_1.2.2.2
## [16] broom_0.7.9           cluster_2.1.2         dbplyr_2.1.1
## [19] shiny_1.7.1           compiler_4.1.1        httr_1.4.2
## [22] backports_1.2.1       assertthat_0.2.1      fastmap_1.1.0
## [25] cli_3.0.1             later_1.3.0           htmltools_0.5.2
## [28] tools_4.1.1           gtable_0.3.0          glue_1.4.2
## [31] Rcpp_1.0.7            cellranger_1.1.0      vctrs_0.3.8
## [34] insight_0.14.4        xfun_0.25             openxlsx_4.2.4
## [37] rvest_1.0.1           mime_0.11             miniUI_0.1.1.1
## [40] lifecycle_1.0.0       MASS_7.3-54           scales_1.1.1
## [43] hms_1.1.0             promises_1.2.0.1      yaml_2.2.1
## [46] curl_4.3.2            see_0.6.7             labelled_2.8.0
## [49] stringi_1.7.4         highr_0.9             bayestestR_0.11.0
## [52] klaR_0.6-15           AlgDesign_1.2.0       boot_1.3-28
## [55] zip_2.2.0             rlang_0.4.11          pkgconfig_2.0.3
## [58] evaluate_0.14         patchwork_1.1.1       labeling_0.4.2
## [61] tidyselect_1.1.1      magrittr_2.0.1        R6_2.5.1
## [64] generics_0.1.0        combinat_0.0-8        DBI_1.1.1
## [67] mgcv_1.8-36           pillar_1.6.2          haven_2.4.3
## [70] foreign_0.8-81        withr_2.4.2           datawizard_0.2.0.1
## [73] abind_1.4-5           modelr_0.1.8          crayon_1.4.1
## [76] questionr_0.7.5       utf8_1.2.2            tzdb_0.1.2
## [79] rmarkdown_2.11        grid_4.1.1           readxl_1.3.1
## [82] data.table_1.14.0     reprex_2.0.1          digest_0.6.27
## [85] xtable_1.8-4          httpuv_1.6.3          numDeriv_2016.8-1.1
## [88] munsell_0.5.0
```

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.