Swimming analysis of zoea 1 larvae of *Taliepus dentatus* reared at different temperatures

Reproducible report

Lucas Bravo, Milena Cano, Mauricio F. Landaeta, Sergio Navarrete, Simone Baldanzi

This document reproduces and supports all data work and statistical analysis of the larval swimming in the paper: swimming performance and morphometrics of the kelp-crab *Taliepus dentatus* zoea 1 exposed to different rearing temperatures.

```
# Packages used for the analysis
library(tidyverse)
library(lme4)
library(plyr)
library(lmerTest)
library(lme4)
library(car)
library(sjPlot)
library(effects)
library(cowplot)
library(emmeans)
library(sjPlot)
```

```
#setting working directory and uploading data (you must set your own working directory)
#data are available in the online version of the article
setwd("~/Desktop/ECOLAB!/paper tesis/swimming R/NEW TRIAL/")
data<-read.table("copia de full_mean.csv", sep=",", header=TRUE)
```

```
#Tidying data
data1<-data%>%group_by(Temp,Pote,Larva) %>%
  dplyr::summarise(Ist_Vel = mean(IV),
                   Tot_Time= sum(Time), Max_Time=max(Time))%>%rowid_to_column(var='ID')
data1$Temp<-as.factor(data1$Temp)
```

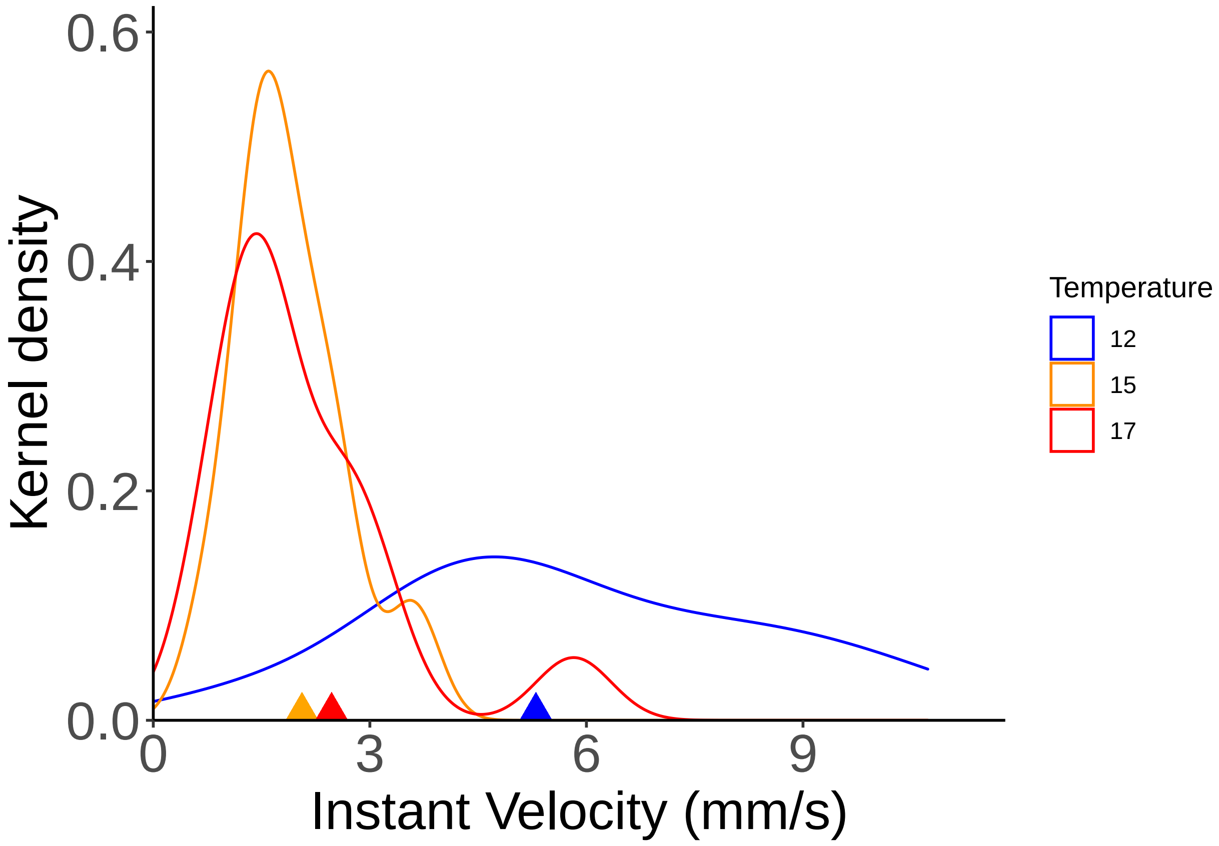
```
#Calculating median for each response variable
medIV <- ddply(data1, "Temp", summarise, grp.med=median(Ist_Vel))
head(medIV)
```

```
##   Temp  grp.med
## 1   12 5.138207
## 2   15 1.769275
## 3   17 1.568503
```

```
medTime <- ddply(data1, "Temp", summarise, grp.med=median(Tot_Time))
head(medTime)
```

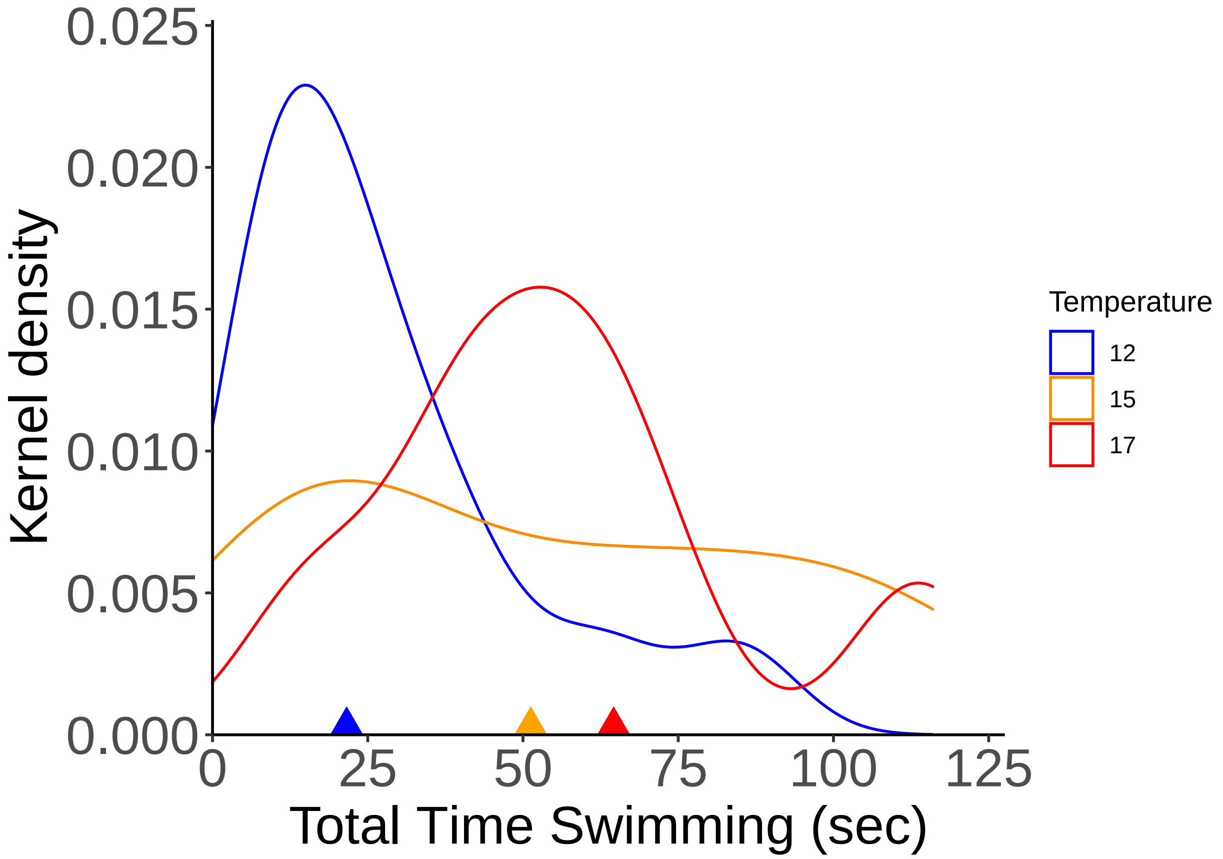
```
##   Temp grp.med
## 1   12   20.30
## 2   15   51.20
## 3   17   54.35
```

```
#plotting as Kernel Densities
dplot_IV<-ggplot(data1, aes(Ist_Vel, color=Temp))+
  geom_density(kernel = "gaussian")+
  geom_point(aes(x=5.30, y=0),colour="blue",shape=17, size=6)+
  geom_point(aes(x=2.06, y=0),colour="orange",shape=17, size=6)+
  geom_point(aes(x=2.47, y=0),colour="red",shape=17, size=6)+
  theme_classic()+
  labs(colour="Temperature", x="Instant Velocity (mm/s)", y="Kernel density")+
  scale_x_continuous(limits = c(0, NA), expand = expansion(mult = c(0, 0.1))) +
  scale_y_continuous(limits = c(0, NA),expand = expansion(mult = c(0, 0.1))) +
  theme(axis.title = element_text(size = 20),axis.text = element_text(size = 20)) +
  scale_colour_manual(values =c("blue","darkorange", "red"))
dplot_IV
```



```
dplot_Time<-ggplot(data1, aes(Tot_Time, color=Temp))+
  geom_density(kernel="gaussian")+
  geom_point(aes(x=21.60, y=0),color="blue",shape=17, size=6)+
  geom_point(aes(x=51.25, y=0),color="orange",shape=17, size=6)+
  geom_point(aes(x=64.6, y=0),color="red",shape=17, size=6)+
  theme_classic()+
  labs(colour="Temperature", x="Total Time Swimming (sec)", y="Kernel density")+
  scale_x_continuous(limits = c(0, NA), expand = expansion(mult = c(0, 0.1))) +
  scale_y_continuous(limits = c(0, NA),expand = expansion(mult = c(0, 0.1))) +
  theme(axis.title = element_text(size = 20),axis.text = element_text(size = 20)) +
  scale_color_manual(values =c("blue","darkorange", "red"))
```

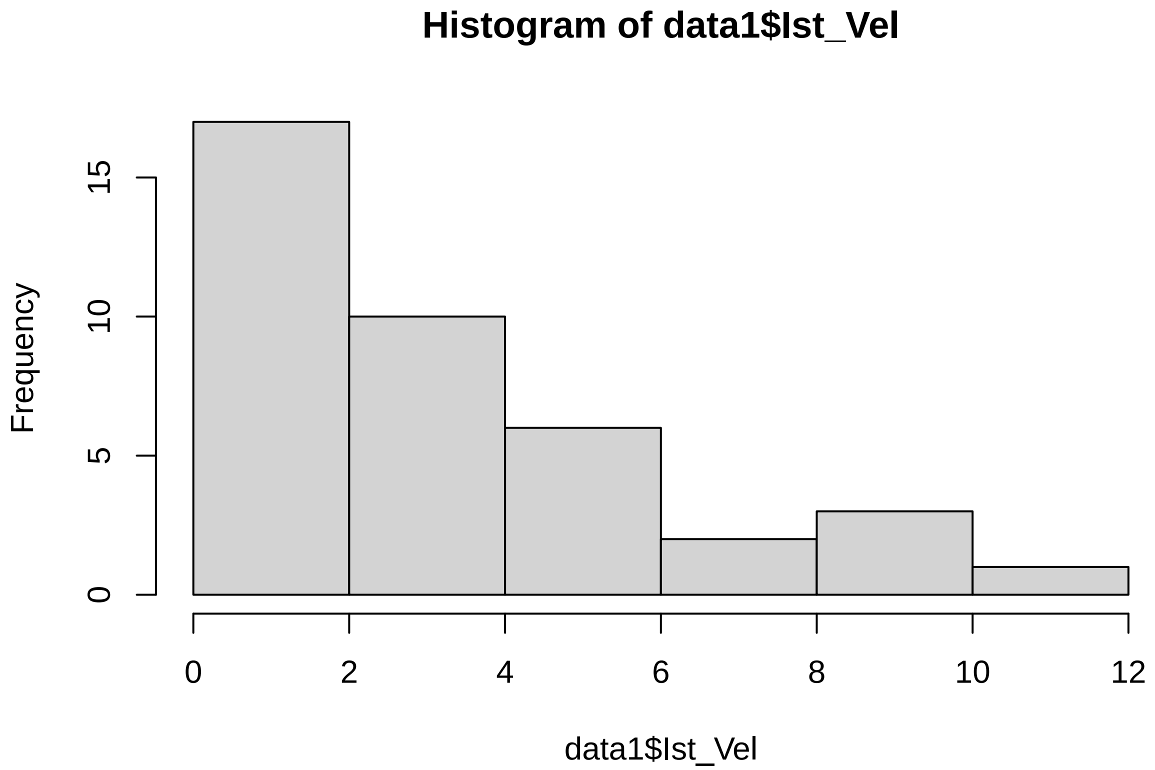
```
dplot_Time
```



```
#Checking for normality
shapiro.test(data1$Ist_Vel)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data1$Ist_Vel
## W = 0.84219, p-value = 7.07e-05
```

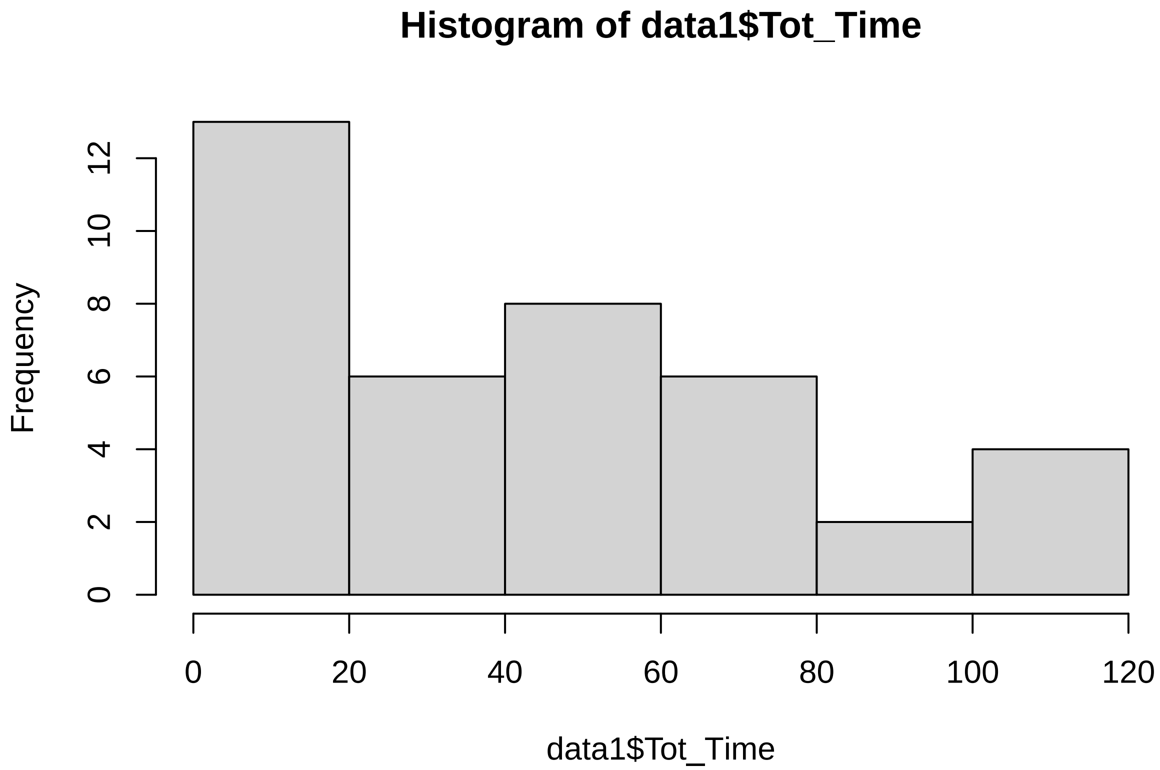
```
hist(data1$Ist_Vel)
```



```
shapiro.test(data1$Tot_Time)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data1$Tot_Time
## W = 0.9082, p-value = 0.003824
```

```
hist(data1$Tot_Time)
```



```
#Checking for heteroscedasticity
leveneTest(data1$Ist_Vel, data1$Temp)
```

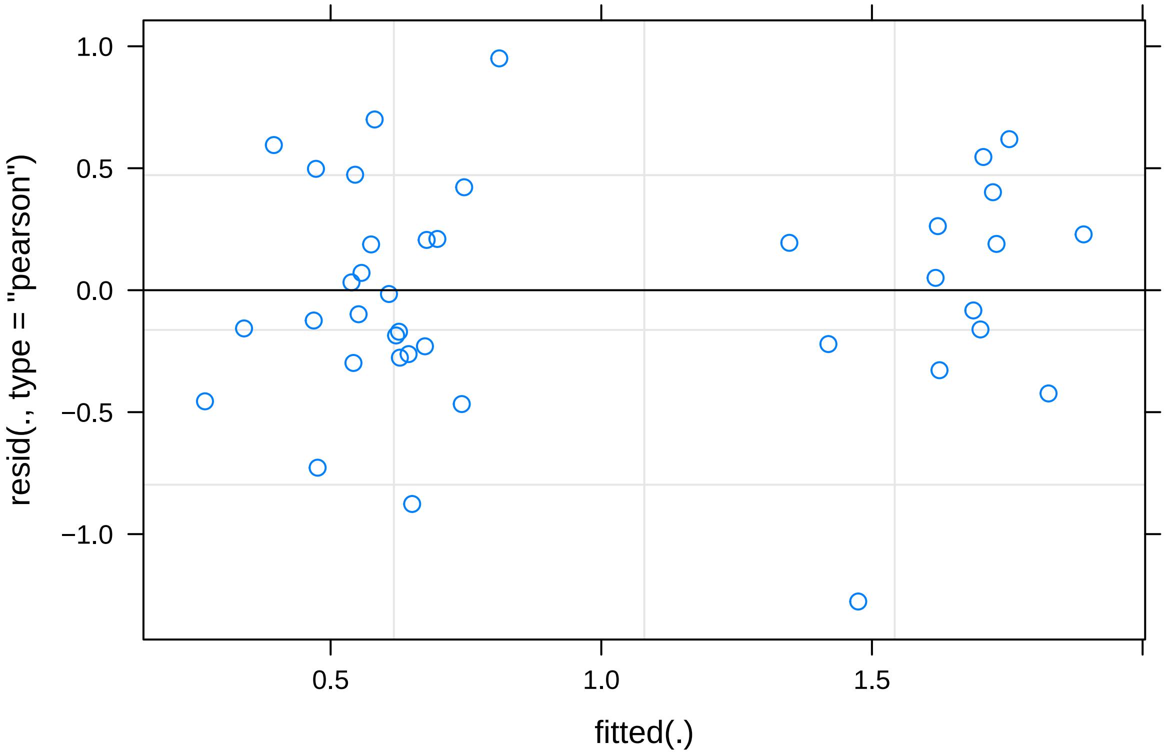
```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value   Pr(>F)
## group  2  5.4257 0.008718 **
##       36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(data1$Tot_Time,data1$Temp)
```

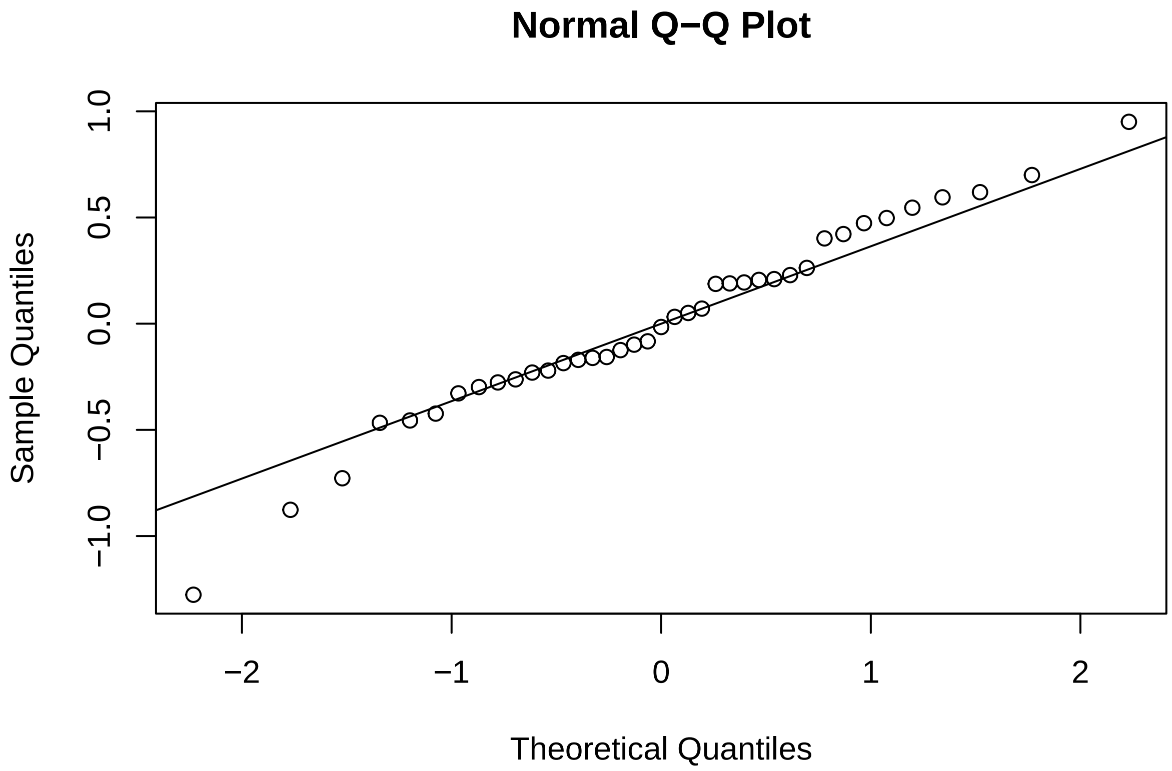
```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value  Pr(>F)
## group  2  2.6217 0.08651 .
##       36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Linear mixed effects model with Log transformed data.
#Pote is a factor with random effects and larva is nested in Pote
modIV<-lmer(log(Ist_Vel)~Temp + (Pote|Larva), data=data1)

#model diagnostic
plot(modIV,which = 1)
```



```
qqnorm(resid(modIV))
qqline(resid(modIV))
```



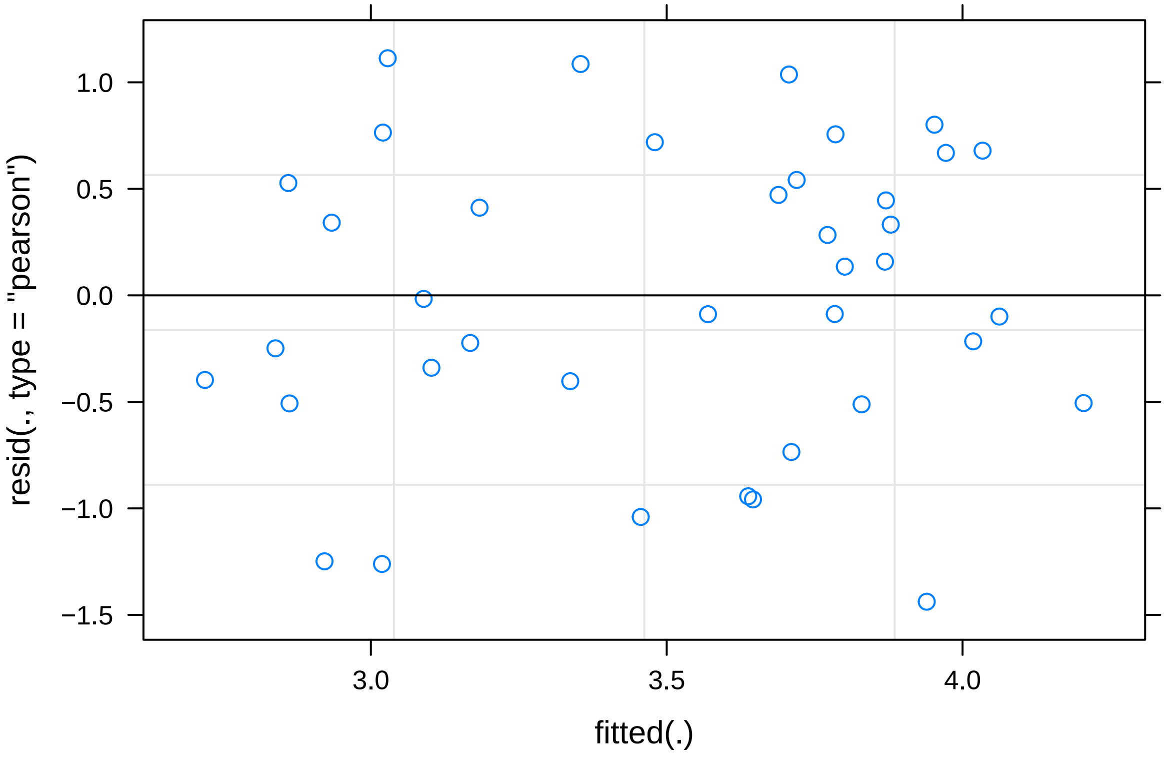
```
#model output
print(summary(modIV))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: log(Ist_Vel) ~ Temp + (Pote | Larva)
##    Data: data1
##
## REML criterion at convergence: 60.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6676 -0.5143 -0.0327  0.5135  1.9865
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Larva    (Intercept) 0.039770 0.19942
##           PoteB       0.004512 0.06717   1.00
##           PoteC       0.002630 0.05128  -1.00 -1.00
##  Residual             0.228902 0.47844
## Number of obs: 39, groups:  Larva, 5
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   1.6338     0.1566 10.4830  10.435 7.26e-07 ***
## Temp15       -1.1488     0.1961 32.8937  -5.857 1.49e-06 ***
## Temp17       -1.0795     0.1814 31.4611  -5.952 1.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr) Temp15
## Temp15 -0.550
## Temp17 -0.590  0.459
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

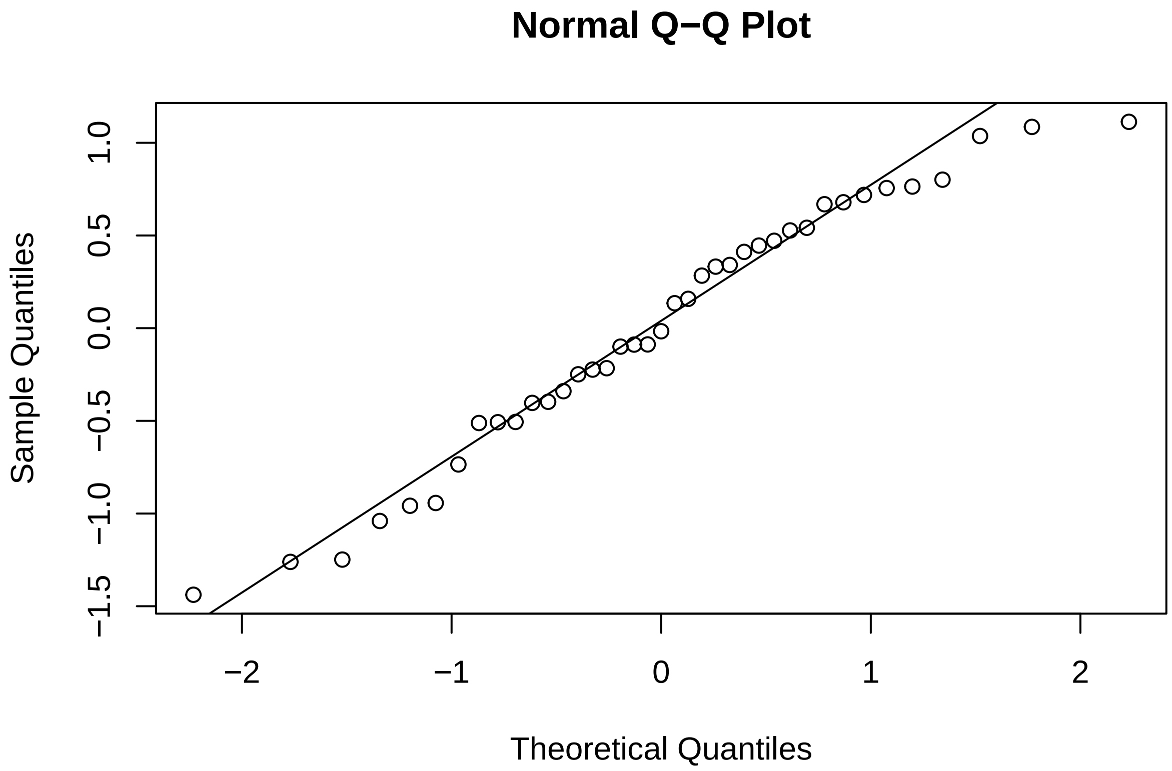
```
anova(modIV)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##      Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## Temp 10.937  5.4685     2 32.411   23.89 4.209e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#model
modTime<-lmer(log(Tot_Time)~Temp + (Pote|Larva), data=data1)
#model diagnostic
plot(modTime,which = 1)
```



```
qqnorm(resid(modTime))
qqline(resid(modTime))
```



```
#model output
print(summary(modTime))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite’s method [
## lmerModLmerTest]
## Formula: log(Tot_Time) ~ Temp + (Pote | Larva)
##    Data: data1
##
## REML criterion at convergence: 90.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.96091 -0.61973 -0.02265  0.72864  1.51772
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Larva    (Intercept) 0.12318  0.3510
##           PoteB       0.03008  0.1734   -1.00
##           PoteC       0.02091  0.1446   -1.00  1.00
##  Residual             0.53782  0.7334
## Number of obs: 39, groups:  Larva, 5
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   3.0088     0.2232 15.2943  13.481 6.76e-10 ***
## Temp15        0.6174     0.2993 33.5858   2.062   0.0470 *
## Temp17        0.8502     0.2775 32.1512   3.063   0.0044 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##        (Intr) Temp15
## Temp15 -0.577
## Temp17 -0.620  0.464
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help(’isSingular’)
```

```
anova(modTime)
```

```
## Type III Analysis of Variance Table with Satterthwaite’s method
##      Sum Sq Mean Sq NumDF  DenDF F value  Pr(>F)
## Temp 5.3292  2.6646     2 32.989  4.9544 0.01314 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

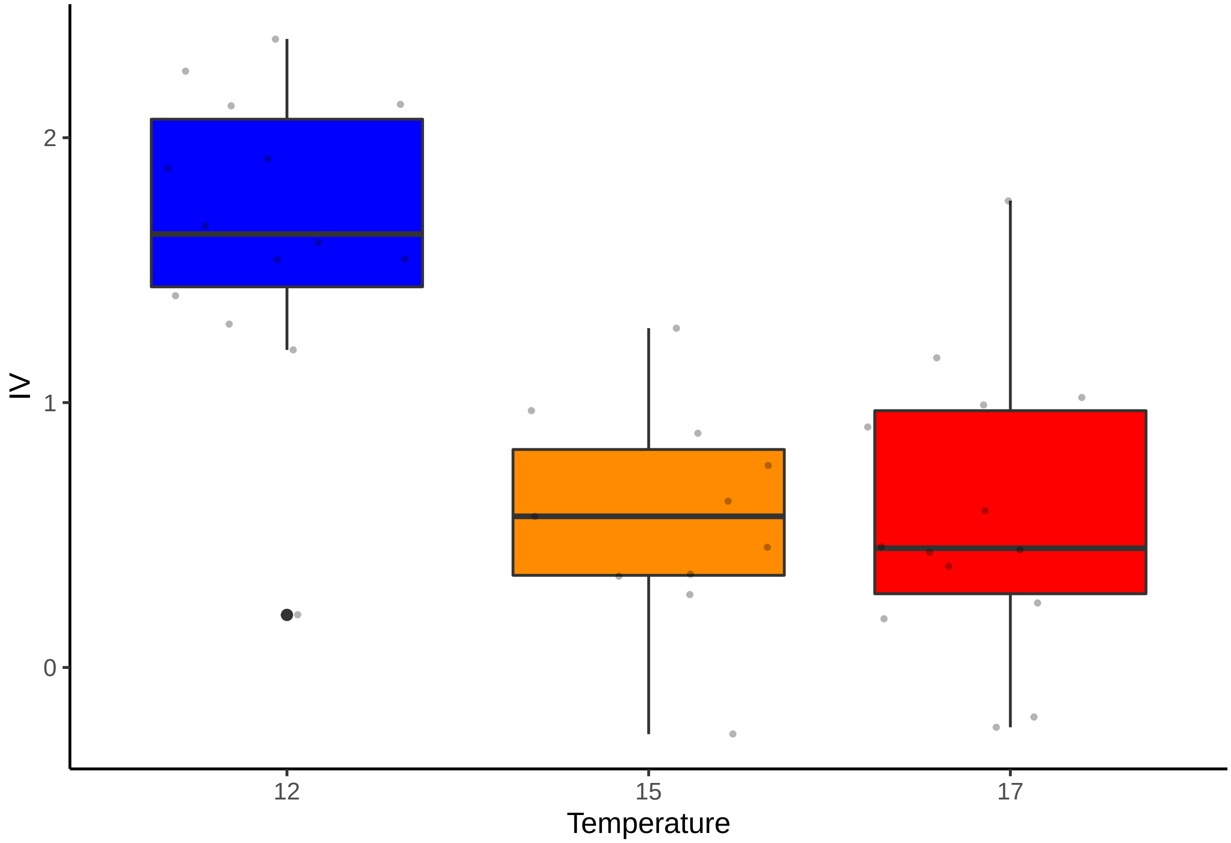
```
#pairwise comparison using estimated means
emmeans(modIV, pairwise ~ Temp, adjust="bonferroni")
```

```
## $emmeans
##  Temp emmean    SE   df lower.CL upper.CL
##  12    1.634 0.200 7.66   1.1691    2.099
##  15    0.485 0.221 9.01  -0.0138    0.984
##  17    0.554 0.200 6.95   0.0797    1.029
##
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $contrasts
##  contrast        estimate    SE   df t.ratio p.value
##  Temp12 - Temp15   1.1488 0.207 24.8   5.554  <.0001
##  Temp12 - Temp17   1.0795 0.186 23.4   5.799  <.0001
##  Temp15 - Temp17  -0.0693 0.207 24.7  -0.335  1.0000
##
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: bonferroni method for 3 tests
```

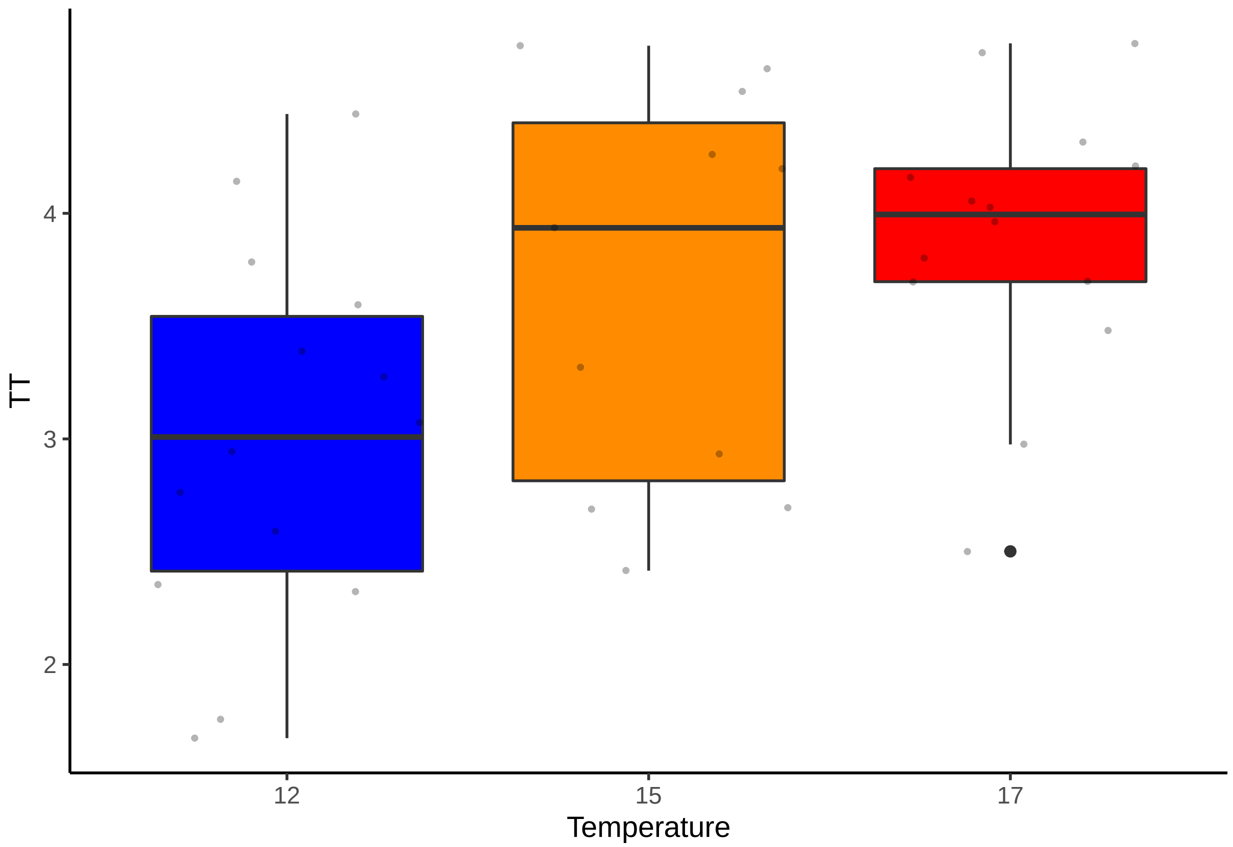
```
emmeans(modTime, pairwise ~ Temp, adjust="bonferroni")
```

```
## $emmeans
##  Temp emmean    SE   df lower.CL upper.CL
##  12     3.01 0.282 7.35     2.35     3.67
##  15     3.63 0.315 9.87     2.92     4.33
##  17     3.86 0.281 7.46     3.20     4.52
##
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $contrasts
##  contrast        estimate    SE   df t.ratio p.value
##  Temp12 - Temp15   -0.617 0.318 25.1  -1.944  0.1897
##  Temp12 - Temp17   -0.850 0.286 23.6  -2.976  0.0199
##  Temp15 - Temp17   -0.233 0.317 25.1  -0.735  1.0000
##
## Degrees-of-freedom method: kenward-roger
## Results are given on the log (not the response) scale.
## P value adjustment: bonferroni method for 3 tests
```

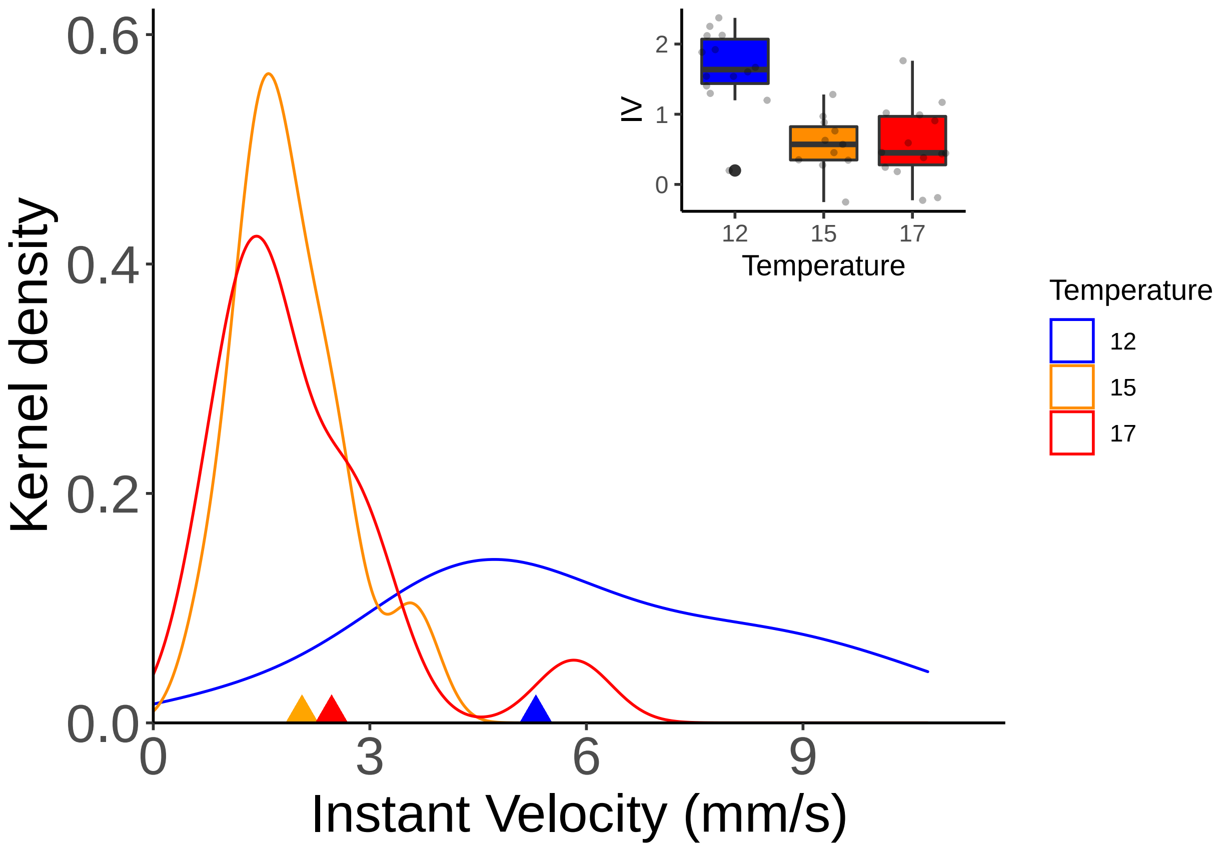
```
#Plotting as boxplots with log transformed data
p1<-ggplot(data1, aes(x=Temp, y=log(Ist_Vel), fill=Temp))+geom_boxplot()+
  geom_jitter(color="black", size=0.6, alpha=0.3)+
  theme_classic()+labs(x="Temperature", y="IV")+theme(legend.position="none")+
   scale_fill_manual(values =c("blue","darkorange", "red"))+ scale_y_continuous(breaks = c(1,2,0))
p1
```



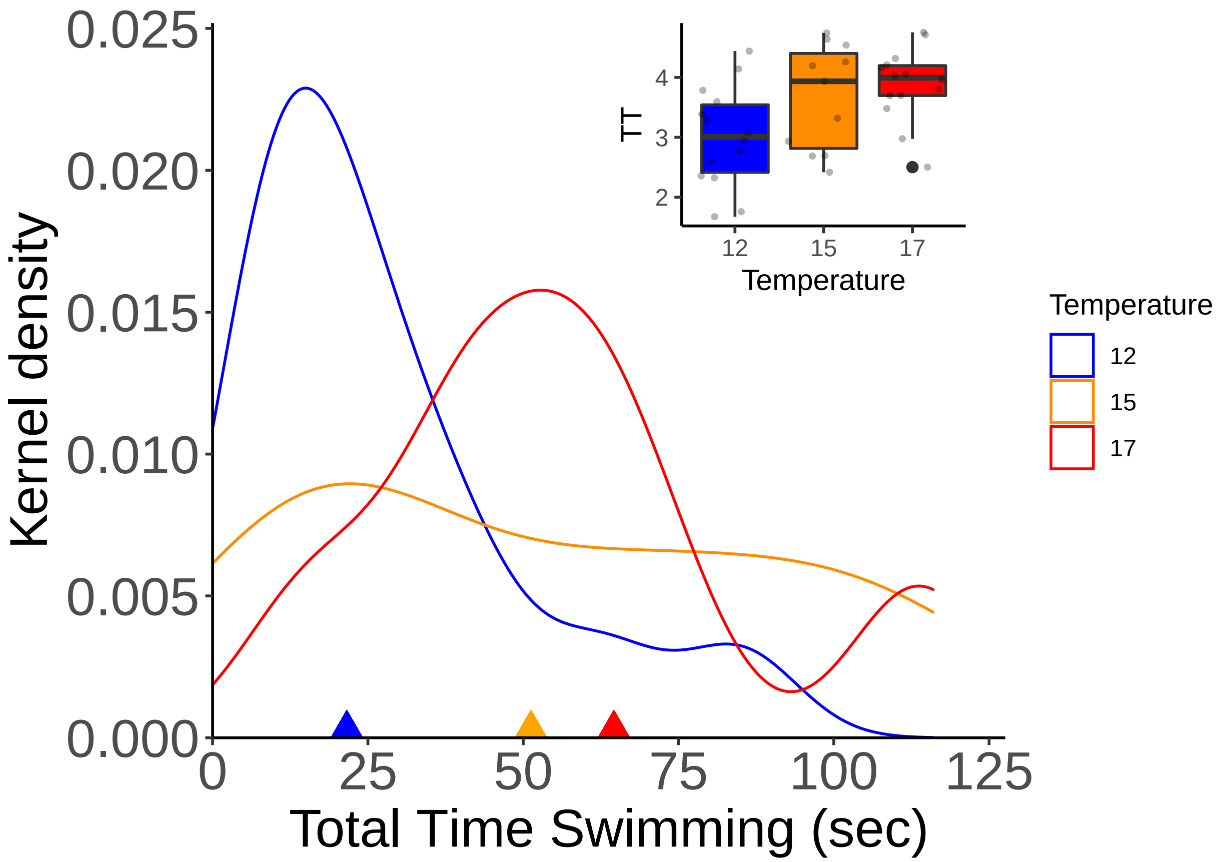
```
p2<-ggplot(data1, aes(x=Temp, y=log(Tot_Time), fill=Temp))+geom_boxplot()+
  geom_jitter(color="black", size=0.6, alpha=0.3)+
  theme_classic() + theme(legend.position="none") +
  labs(x="Temperature", y="TT")+
   scale_fill_manual(values =c("blue","darkorange", "red"))
p2
```



```
#plotting final kernel graphs with insets
plot.with.inset.IV <-
  ggdraw() +
  draw_plot(dplot_IV) +
  draw_plot(p1, x = .49, y = 0.65, width = .3, height = .35)
plot.with.inset.IV
```



```
plot.with.inset.Tot_Time <-
  ggdraw() +
  draw_plot(dplot_Time) +
  draw_plot(p2, x = .49, y = 0.65, width = .3, height = .35)
plot.with.inset.Tot_Time
```


Kernel density
0.025
0.020
0.015
0.010
0.005
0.000
0
25
50
75
100
125
Total Time Swimming (sec)
TT
4
3
2
12
15
17
Temperature
Temperature
12
15
17