Choice Assay Exploratory Data Visualization

Table of Contents

# Intro

This document is where I will conduct preliminary data visualization and analysis for the choice assay experiment.

In the .html versions of this document, you can click on the boxes marked “code” to display the code that created each non-text section.

# Setup

This section contains code for the initial setup such as loading libraries and importing the data. Therefore, this section can be skipped in most cases.

All code is executed from the project root directory.

# # report numerical output with only 4 trailing digits  
# panderOptions('round', 4)  
# # don't remove the decimal if there is a whole number  
# panderOptions('keep.trailing.zeros', TRUE)

##-- function to provide linear equation from a model. It can also return the  
 ## predicted values when given x values  
lm\_eqtn <- function(model,x=NULL){  
 # extract model parameters  
 incpt <- coef(model)[1] # intercept  
 slope <- coef(model)[2] # linear slope  
 summ <- summary(model) # summary   
 r.sqr <- summ$r.squared # model r squared  
 adj.r.sqr <- summ$ajd.r.squared  
 # if x is not provided, print the equation as text  
 if(is.null(x)){  
 equation <- paste0("y=",round(slope,3),"\*x+(",round(incpt,3),"); r^2=",  
 round(r.sqr,4))  
 return(noquote(equation))  
 # if x is given, calculate the predicted value(s)  
 } else {  
 y <- as.vector((x\*slope)+incpt)  
 return(y)  
 }  
}  
##--

# load data  
getwd()

## [1] "/home/morrowcj/Documents/projects/holstrom-resistance-study"

## L4 gypsy moth wet-dry mass calibration data  
larv.calib.data <- read.csv("data/Choice\_Assay/larval-calibration-data\_L4.csv")  
## leaf wet-dry mass calibration data  
leaf.calib.data <- read.csv("data/Choice\_Assay/leaf-calibration-data.csv")  
## choice assay raw data ("long" format = 1 observation per row)  
choice.assay.data <- read.csv("data/Choice\_Assay/leaf-and-larvae-data\_long.csv")

##--- This code chunk is for modifying the data frames - adding columns,  
 ## updating variable classes, etc..  
  
# redifine class specific columns. This is mostly for converting variables of  
 ## class "character" or "numeric" into class "factor".   
## choice assay dataset:  
choice.assay.data <- within(choice.assay.data,{  
 ## turn the genet.lf (leaf genet) column into a factor  
 genet.lf <- as.factor(genet.lf)  
})  
## leaf calibration data:  
leaf.calib.data <- within(leaf.calib.data,{  
 ## turn the Genet column into a factor  
 Genet <- as.factor(Genet)  
 ## turn the grouping variable into a factor  
 group <- as.factor(group)  
})  
  
# calculate the number of leaves in each dish  
leaves <- choice.assay.data %>% group\_by(Dish.ID) %>%   
 summarise("leaves" = length(genet.lf))  
## add leaves/dish into the data frame  
choice.assay.data <- merge(choice.assay.data,leaves,by = "Dish.ID")  
  
# Time data:  
## convert date and time data to date-time format  
start.date.time <- dmy(choice.assay.data$start.date\_dmy) +  
 hms(choice.assay.data$start\_24.00.00) # trial start  
end.date.time <- dmy(choice.assay.data$end.date\_dmy) +  
 hms(choice.assay.data$end\_24.00.00) # trial end  
## calculate the elapsed time of the trial for each leaf  
choice.assay.data$elapsed.time\_hours <- as.numeric(end.date.time -   
 start.date.time)  
  
# calculate n (for testing)  
full.n <- nrow(choice.assay.data)

# Wet-dry mass calibrations

Because we are unable to directly measure the dry mass (biomass) of our larvae and leaves at the beginning of the trial, we have to estimate them according to a calibration. The calibration was done by taking weights of leaves/larvae, that were similar to those used in the trial, before and after dehydrating them (freeze drying).

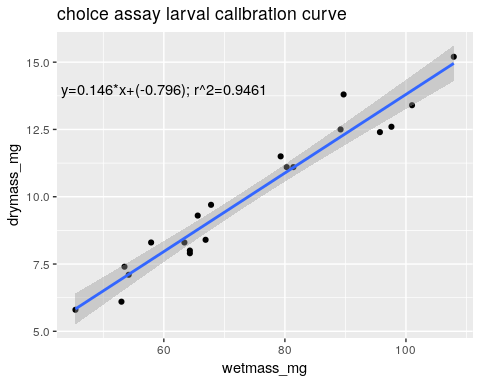
## Larval Calibration

To create a larval calibration curves, gypsy moth larvae of the same instar and approximate size were weighed before and after dehydration. Below is the linear model derived from these calibration data:

# declare linear model for larval calibration (wet-dry mass)  
larv.calib.model <- lm(larv.calib.data,formula = drymass\_mg~wetmass\_mg)  
  
# print the model call  
larv.calib.model$call

## lm(formula = drymass\_mg ~ wetmass\_mg, data = larv.calib.data)

# plot dry~wet mass  
ggplot(data = larv.calib.data, aes(y = drymass\_mg, x = wetmass\_mg))+  
 geom\_point() + #add points to the plot  
 stat\_smooth(method = "lm") + #add a trendline  
 labs(title = "choice assay larval calibration curve")+  
 annotate("text",label = lm\_eqtn(larv.calib.model),x = 60, y = 14)



This calibration should work perfectly. The line fits the data well. So, now we can plug our initial larval wet-mass values into the equation to get the predicted values which are our estimated dry mass values.

# calculate the estimated drymasses using the lm\_eqtn() function created in  
 ## the setup section. Assign this value to a new column in the data.  
choice.assay.data$gm.init.drymass\_mg <-   
 lm\_eqtn(larv.calib.model, choice.assay.data$gm.init.wetmass\_mg) #predicted  
 ## values from the model  
  
## uncomment this to visualize the new variable. As expected, it is linear:  
# plot(choice.assay.data$gm.init.drymass\_mg ~   
# choice.assay.data$gm.init.wetmass\_mg)

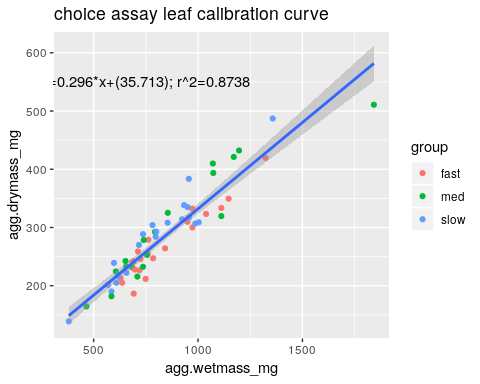
## Leaf Calibration

Here I will test different methods for calibrating the leaf wet-dry mass. This problem is not as straight forward as the larval calibration because of the genet.f and group

### Linear cal. method

First, we’ll try the linear calibration model for leaf wet-dry mass with no covariates:

# declare linear model for leaf calibration (wet-dry mass)  
leaf.calib.model <- lm(leaf.calib.data,formula =   
 agg.drymass\_mg~agg.wetmass\_mg)  
  
# plot dry mass as a function of wet mass  
ggplot(data = leaf.calib.data, # data set to look in  
 aes(y = agg.drymass\_mg, # variable to go on x axis  
 x = agg.wetmass\_mg)) + # ditto for y  
 labs(title = "choice assay leaf calibration curve") + # adds plot title  
 geom\_point(aes(col = group)) + # adds points to the plot  
 stat\_smooth(method = "lm") + # adds a trendline  
 annotate("text", label = lm\_eqtn(leaf.calib.model), # add model eqt. to plot  
 x = 750, y = 550) # coordinates of the annotation



# print the model's r.squared  
c("r.sqr" = summary(leaf.calib.model)$r.squared)

## r.sqr   
## 0.8738221

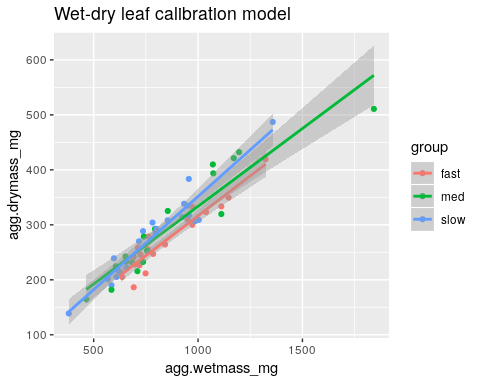
The fit is OK but, because there is substantial variation in dry-mass/wet-mass, we should look at the model grouped by growth category (group):

### Grouped cal. method

# update the cal. model to include group  
leaf.calib.model <- update(leaf.calib.model,. ~ . + group)  
  
# check if 'group' is a significant predictor of water loss:  
(leaf.cal.anova <- anova(leaf.calib.model))

## Analysis of Variance Table  
##   
## Response: agg.drymass\_mg  
## Df Sum Sq Mean Sq F value Pr(>F)   
## agg.wetmass\_mg 1 307529 307529 462.639 < 2.2e-16 \*\*\*  
## group 2 7182 3591 5.402 0.007158 \*\*   
## Residuals 56 37225 665   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# plot the data with lines for each group this time  
ggplot(data = leaf.calib.data,   
 aes(y = agg.drymass\_mg,   
 x = agg.wetmass\_mg,   
 group = group)) + # grouping factor  
 labs(title = "Wet-dry leaf calibration model",  
 xlab = "aggregate wet mass (mg)",   
 ylab = "aggregate dry mass (mg)") +   
 geom\_point(aes(col = group)) + # color points according to group  
 stat\_smooth(method = "lm", aes(col = group)) # ditto for trendline



There does seem to be an important effect of group on this relationship. Based on the above information, it should incorporated into the models. Here we will exctract the linear coefficients for each group:

## look at the model coefficients by group  
 ## library(data.table) allows the following  
dat.tab <- data.table(leaf.calib.data)  
coef.tab <- dat.tab[,list(intcpt = coef(lm(agg.drymass\_mg ~   
 agg.wetmass\_mg))[1],  
 slope = coef(lm(agg.drymass\_mg ~   
 agg.wetmass\_mg))[2],  
 r.sqr = summary(lm(agg.drymass\_mg ~  
 agg.wetmass\_mg))$r.squared),  
 by = group]  
  
within(coef.tab,{  
 r.sqr <- round(r.sqr,4)  
 slope <- round(slope,4)  
 intcpt <- round(intcpt,4)  
})

## group intcpt slope r.sqr  
## 1: slow 12.7560 0.3389 0.9151  
## 2: fast 25.9377 0.2904 0.9076  
## 3: med 50.8847 0.2828 0.8831

These models are improvements over the original non-grouped version.

#### Outlier testing

There may be outliers in the calibration data. For example, I suspect that the agg.wetmass\_mg that is > 1500 may be influencing the med line. I will officially test for outliers here:

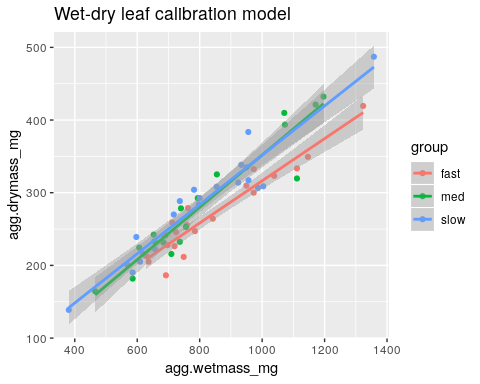
# bonferroni outlier test  
(out.test <- outlierTest(leaf.calib.model))

## rstudent unadjusted p-value Bonferonni p  
## 49 -4.15197 0.00011562 0.006937

# define test variables  
out.sig <- (out.test$bonf.p <= 0.5) # checks if significant  
wording <- ifelse(out.sig, "is", "is not") # changes conclusion based on this

The most likely outlier candidate is significant. Let’s see how it affects the model when we remove it

# check if the top outlier is significant:  
if(!out.sig) {  
 ## if it is not significant, print a message  
 print("no significant outlier")  
 ## otherwise, execute the rest:   
} else   
  
# create an outlier index from the test   
cal.outliers <- as.numeric(names(outlierTest(leaf.calib.model)[[1]]))  
  
# update the model to remove the outlier  
leaf.calib.model2 <- update(leaf.calib.model,   
 data = leaf.calib.data[-cal.outliers])  
  
# # uncomment print the updated anova  
# anova(leaf.calib.model2)  
  
# look at the model coefficients by group again (see above for code notes)  
dat.tab2 <- data.table(leaf.calib.data[-cal.outliers,])  
coef.tab2 <- dat.tab2[,list(intcpt = coef(lm(agg.drymass\_mg ~   
 agg.wetmass\_mg))[1],  
 slope = coef(lm(agg.drymass\_mg ~   
 agg.wetmass\_mg))[2],  
 r.sqr = summary(lm(agg.drymass\_mg ~   
 agg.wetmass\_mg))$r.squared),  
 by = group]  
  
ggplot(data = leaf.calib.data[-cal.outliers,],   
 aes(y = agg.drymass\_mg,   
 x = agg.wetmass\_mg,   
 group = group)) +  
 labs(title = "Wet-dry leaf calibration model",  
 xlab = "aggregate wet mass (mg)",   
 ylab = "aggregate dry mass (mg)") +   
 geom\_point(aes(col = group)) + #add points to the plot  
 stat\_smooth(method = "lm", aes(col = group)) #add a trendline



within(coef.tab2,{  
 r.sqr <- round(r.sqr,4)  
 slope <- round(slope,4)  
 intcpt <- round(intcpt,4)  
})

## group intcpt slope r.sqr  
## 1: slow 12.7560 0.3389 0.9151  
## 2: fast 25.9377 0.2904 0.9076  
## 3: med -7.7621 0.3596 0.9090

Removing the potential outlier does not affect the curve alot but, removing it places the range of the calibration, closer to the range of the larvae in the experiment. For this reason, I will remove the outlier from the calibration model.

# update the calibration model (remove outlier)  
leaf.calib.model <- leaf.calib.model2

### Genet calibration:

The r-squared values are a bit better when grouped linear models are used over the traditional linear model, but it still may be best to use 1 correction factor per genotype (which is the average dry-mass/wet-mass for 4-5 leaves per genet.)

We can create a “key” for with the calibration factor for each genet. Below is the first few rows of that key:

# create the cal. key  
## start with the full data  
leaf.cal\_lookup.tab <- leaf.calib.data %>%   
 ## select only the relavent columns  
 select(Genet,group,biomass.proportion\_dry.div.wet) %>%   
 ## group the data by grouping factors  
 group\_by(Genet,group) %>%  
 ## calculate the means for each group  
 summarise(mean.biomass.prop = mean(biomass.proportion\_dry.div.wet,  
 na.rm = TRUE))  
  
# print top 5 rows of the key  
head(leaf.cal\_lookup.tab, n = 5)

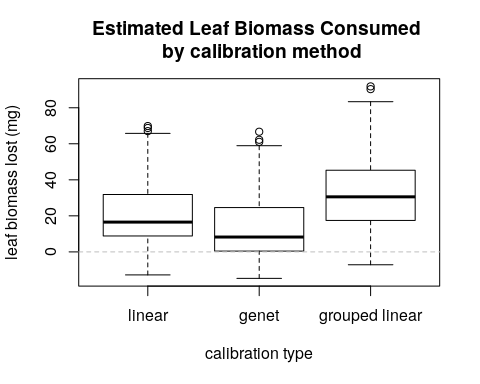
## # A tibble: 5 x 3  
## # Groups: Genet [5]  
## Genet group mean.biomass.prop  
## <fct> <fct> <dbl>  
## 1 5 med 0.335  
## 2 6 slow 0.350  
## 3 16 fast 0.338  
## 4 24 slow 0.344  
## 5 58 med 0.355

# merge the lookup table into the choice assay data  
choice.assay.data <- merge(choice.assay.data,leaf.cal\_lookup.tab,  
 by.x = "genet.lf",by.y = "Genet",all.x = TRUE)  
  
# add in coeficients from grouped model  
choice.assay.data <- merge(choice.assay.data,coef.tab,by = "group",all.x = TRUE)

### Calculate leaf drymass estimates

Next we can estimate the initial dry-mass with each of the calibration methods and then compare them.

# calculate initial drymass (predicted values)  
## full linear model  
choice.assay.data$lf.init.drymass\_no.calib\_mg <-  
 lm\_eqtn(model = leaf.calib.model, choice.assay.data$lf.init.wetmass\_mg)  
## genet calibration method  
choice.assay.data$lf.init.drymass\_genet.calib\_mg <-  
 choice.assay.data$mean.biomass.prop \* choice.assay.data$lf.init.wetmass\_mg  
## grouped linear model  
choice.assay.data$lf.init.drymass\_group.calib\_mg <-  
 (choice.assay.data$lf.init.wetmass\_mg \* choice.assay.data$slope) +   
 choice.assay.data$intcpt  
  
# calculate leaf mass lost/consumed  
## the same value is being stored as an outside variable (dif<x>) and as a col.  
choice.assay.data$bmass.consumed\_no.cal <- # full  
 dif0 <- choice.assay.data$lf.init.drymass\_no.calib\_mg -   
 choice.assay.data$lf.end.drymass\_mg  
choice.assay.data$bmass.consumed\_genet.cal <- # genet  
 dif1 <- choice.assay.data$lf.init.drymass\_genet.calib\_mg -  
 choice.assay.data$lf.end.drymass\_mg  
choice.assay.data$bmass.consumed\_group.cal <- # grouped  
 dif2 <- choice.assay.data$lf.init.drymass\_group.calib\_mg -  
 choice.assay.data$lf.end.drymass\_mg  
  
  
# caculate proportion of leaves where the leaf consumed is estimated to be  
 ## negative  
prop0 <- length(which(dif0 < 0)) / length(dif0) #full  
prop1 <- length(which(dif1 < 0)) / length(dif1) #genet  
prop2 <- length(which(dif2 < 0)) / length(dif2) #grouped  
  
# side-by-side boxplots of the mass consumed by method  
boxplot(dif0,dif1,dif2,  
 names = c("linear","genet","grouped linear"),  
 main = "Estimated Leaf Biomass Consumed \n by calibration method",  
 ylab = "leaf biomass lost (mg)",  
 xlab = "calibration type")  
## add a horixzontol line at y = 0  
abline(h = 0, col = "grey", lty = 2)



# Print the boxplot stats:  
rbind("full" = summary(dif0),  
 "genet" = summary(dif1),  
 "grouped" =summary(dif2))

## Min. 1st Qu. Median Mean 3rd Qu. Max.  
## full -12.79052 8.8700974 16.530717 21.19119 31.81893 69.83824  
## genet -14.72744 0.4606336 8.221228 13.38288 24.29538 66.69243  
## grouped -7.14974 17.5071182 30.563253 33.52664 45.18473 91.85552

# print the proportion of observations where leaves are estimated to gain mass  
 ## for each treatmentt  
round(data.frame("linear" = prop0,   
 "genet" = prop1,   
 "grouped linear" = prop2,  
 row.names = "proportion of X<0:"),  
 digits = 4)

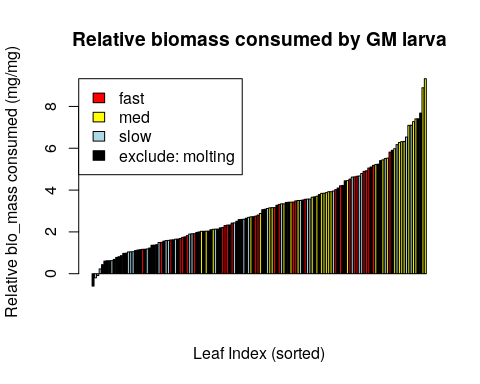
## linear genet grouped.linear  
## proportion of X<0: 0.0845 0.2465 0.0211

The comparison shows that both linear methods estimate fewer observations with “leaf biomass lost” < 0. Additionally, the grouped linear method estimates even fewer negative values than the original linear model:

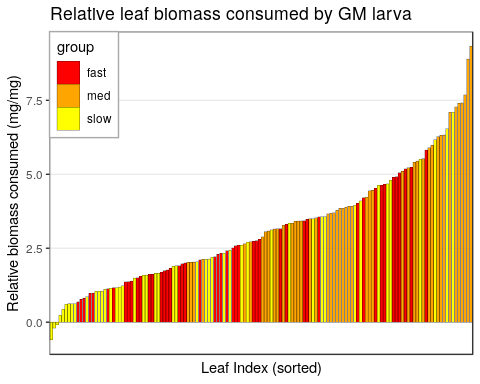
#### Relative Biomass consumed

“Biomass consumed” isn’t really the metric that we are interested in. Instead, we really want “relative biomass consumed” or in other words: “mg of leaf biomass consumed per mg initial larval biomass”. We’ll calculate it here before we filter the data (i.e. omitting molting larvae):

# calculate the relative biomass consumed and create new variables   
choice.assay.data <- within(choice.assay.data,{  
 rel.bmass.cons\_no.cal <- bmass.consumed\_no.cal/ #linear  
 gm.init.drymass\_mg  
 rel.bmass.cons\_genet.cal <- bmass.consumed\_genet.cal/ #genet  
 gm.init.drymass\_mg  
 rel.bmass.cons\_group.cal <- bmass.consumed\_group.cal/ #grouped  
 gm.init.drymass\_mg  
})  
  
# create a plot showing the relative consumption data by group  
## make an index to sort the data for plotting  
sort.index <- choice.assay.data$rel.bmass.cons\_group.cal %>% order()  
## make a vector of colors according to the group  
colors <- ifelse(test = choice.assay.data$molting\_bin == 1, yes = "black",  
 no = ifelse(choice.assay.data$group == "fast",yes = "red",  
 no = ifelse(choice.assay.data$group == "med", yes = "yellow",  
 no = "lightblue")))  
## initiate the plot - orded by the index  
barplot(choice.assay.data$rel.bmass.cons\_group.cal[sort.index],   
 ylab = "Relative bio\_mass consumed (mg/mg)",  
 xlab = "Leaf Index (sorted)",  
 main = "Relative biomass consumed by GM larva",  
 col = colors[sort.index]) # use our vector of colors  
## add a legend to the plot  
legend("topleft",legend = c("fast","med","slow","exclude: molting"),  
 fill = c("red","yellow","lightblue", "black"))



# Using ggplot:  
## first, re-order the levels of the leaf vial IDs by relative bmass consumed  
choice.assay.data$ordered.levels <-   
 factor(choice.assay.data$vial.num,   
 levels = choice.assay.data$vial.num[  
 order(choice.assay.data$rel.bmass.cons\_group.cal)  
 ]  
 )  
  
# create a plot object with the appropiare values:  
rel.bmass.plot <- ggplot(data = choice.assay.data,   
 aes(y = rel.bmass.cons\_group.cal, x = ordered.levels, fill = group)) +  
 geom\_bar(stat = "identity",width = .9, col = "black", lwd = .1) +  
 labs(y = "Relative biomass consumed (mg/mg)",  
 x = "Leaf Index (sorted)",   
 title = "Relative leaf biomass consumed by GM larva") +  
 scale\_fill\_manual(values = c("red", "orange", "yellow","white","black")) +  
 theme\_bw() +   
 theme(axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.grid.major.x = element\_blank(),  
 legend.justification = c(0,1),  
 legend.position = c(0,1),  
 legend.background = element\_rect(fill = "white",  
 linetype = "solid",  
 color = "darkgrey"))  
  
## print the plot  
rel.bmass.plot



There are a few realtive consumption values that are negative, but they are within the error of estimation by predicted value. **Note** in my most recent meeting with Rick (08-Jan-2019), we discussed what to do with these negative values - leave them as is or express them then as 0s. The latter option is more biologically relavent, as the insects did not consume anything. However, this has the potential of introducing a new bias. If we only correct some values, but not others, realizing that the predicted values will over-estimate as often as they underestimate the true value. For this reason, I have decided to leave the values as is for now. The negative values are easier to justify than altering some of the data.

#### Calculate biomass consumed by larva

Currently, the data are on a per-leaf basis but we would like to add up the biomass consumed in each larva (sum of 2 or 3 leaves in a dish). Below is a subset of those dish-wide consumption calculations:

# sum up the biomass consumed for each dish (group cal)  
cons.key <- choice.assay.data %>% group\_by(Dish.ID) %>%   
 summarise("dish.bmass.consumed" = sum(bmass.consumed\_group.cal))  
# subset the key  
cons.key[sample(1:nrow(cons.key),size = 5),]

## # A tibble: 5 x 2  
## Dish.ID dish.bmass.consumed  
## <fct> <dbl>  
## 1 24 73.6  
## 2 6 55.3  
## 3 24B 82.7  
## 4 9B 55.0  
## 5 34 79.5

# update the data with this new variable  
choice.assay.data <- merge(choice.assay.data,cons.key,by = "Dish.ID")  
# test that data merged properly  
stopifnot(nrow(choice.assay.data) == full.n)

We will then calculate the ‘preference’ to a leaf by dividing the leaf biomass consumed by the total biomass consumed by the larva:

# calculate this preference value  
choice.assay.data$leaf.preference <- with(choice.assay.data,  
 ## divide leaf biomass consumed   
 bmass.consumed\_group.cal/  
 ## by dish biomass consumed  
 dish.bmass.consumed  
 )  
  
# print a subset of the preference values  
choice.assay.data %>% select(Dish.ID, genet.lf, leaf.preference,   
 bmass.consumed\_group.cal,   
 dish.bmass.consumed) %>%   
 filter(Dish.ID %in% c("1","10","20"))

## Dish.ID genet.lf leaf.preference bmass.consumed\_group.cal  
## 1 1 135 0.2301887 12.50290  
## 2 1 203 0.7698113 41.81297  
## 3 10 118 0.3437712 36.84260  
## 4 10 310 0.6562288 70.32925  
## 5 20 237 0.3535038 20.52014  
## 6 20 256 0.6464962 37.52772  
## dish.bmass.consumed  
## 1 54.31587  
## 2 54.31587  
## 3 107.17185  
## 4 107.17185  
## 5 58.04786  
## 6 58.04786

# check that the proportions sum to 1  
x <- choice.assay.data %>%   
 group\_by(Dish.ID) %>%   
 summarise("total" = sum(leaf.preference))  
# stop if they don't all sum to 1  
stopifnot(all.equal(rep(1,nrow(x)), x$total))

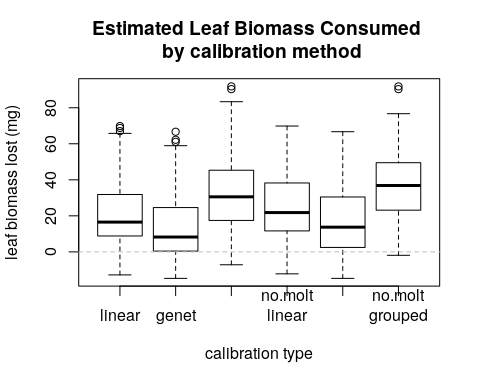
For example, in dish 1, 77% of the bmass consumed was from genet 203 and the other 23% from genet 135.

These preference values will be used later.

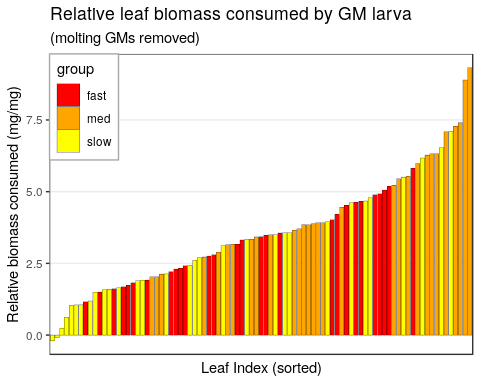
### Filter data by molting

Until now, the choice.assay.data object has contained all data. However, some of the larvae molted during the feeding trial. To account for this, most of the dishes where re-run with new larvae. Here I will remove those dishes where molting occured:

# filter the data by the molting column: 1=molted, 0=not  
data.reduced <- choice.assay.data %>% filter(molting\_bin < 1)  
  
# n of reduced data set (for testing)  
reduced.n <- nrow(data.reduced)  
  
# calculate the proportion of negative values  
prop.red <- length(which(data.reduced$bmass.consumed\_group.cal < 0))/  
 nrow(data.reduced) # grouped  
prop.genet.red <- length(which(data.reduced$bmass.consumed\_genet.cal < 0))/  
 nrow(data.reduced) # genet  
  
# create a side-by-side boxplot of the leaf mass consumed estimates (see above)  
boxplot(dif0,dif1,dif2,  
 data.reduced$bmass.consumed\_no.cal,  
 data.reduced$bmass.consumed\_genet.cal,  
 data.reduced$bmass.consumed\_group.cal,  
 names = c("linear","genet","grouped\n linear",   
 "no.molt\nlinear","no.molt\n genet","no.molt\ngrouped"),  
 main = "Estimated Leaf Biomass Consumed \n by calibration method",  
 ylab = "leaf biomass lost (mg)",xlab = "calibration type")  
abline(h = 0, col = "grey", lty = 2)



# abline(v = 3.5)  
  
# Re-print the above plot with the filtered data  
rel.bmass.plot %+% data.reduced + labs(subtitle = "(molting GMs removed)")



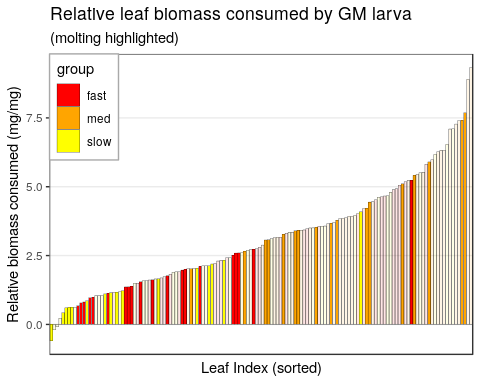
From the first plot, it appears that our estimates have not changed much - the grouped linear estimator is still the best. Most of the removed data came from those leaves that lost very little biomass (molting larvae don’t feed).

# proportion of dishes with molting larvae  
molt.prop <- (1 - (nrow(data.reduced)/nrow(choice.assay.data))) #0.3732

The proportion of larvae that underwent a molt during the drial is 0.3732.

The next plot highlights those larvae that molted:

rel.bmass.plot + (aes(alpha = factor(molting\_bin+1))) +   
 scale\_alpha\_discrete(guide = FALSE) +   
 labs(subtitle = "(molting highlighted)")



This shows that many of the molting larvae were from the lower end of the distribution, but they seemed to have similar preferences to their non-molting counterparts.

### Check Data

Here we will look at the data to see if any trends our outliers emerge.

#### Biomass consumed

First, let’s look at final leaf biomass as a function of initial leaf biomass:

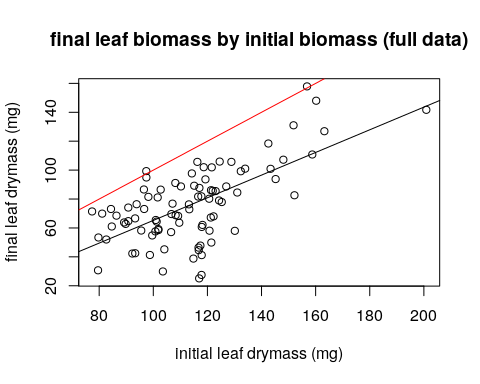
# create a linear model to draw a line  
model <- lm(data = data.reduced, lf.end.drymass\_mg ~   
 lf.init.drymass\_group.calib\_mg)  
  
cat("original r-squared (with group covariate):\n")

## original r-squared (with group covariate):

summary(model)$r.squared

## [1] 0.420395

plot(data.reduced$lf.end.drymass\_mg ~   
 data.reduced$lf.init.drymass\_group.calib\_mg,  
 xlab = "initial leaf drymass (mg)",  
 ylab = "final leaf drymass (mg)",  
 main = "final leaf biomass by initial biomass (full data)"  
 )  
suppressWarnings(abline(model))  
abline(c(0,1),col = "red")



The red line is the 1:1 line. Leaves on this line lost no biomass.

caption <- "The red line is the 1:1 line. Leaves on this line lost no biomass."

This shows that most leaves had some biomass consumed and larger leaves lost a smaller proportion of their biomass than smaller leaves. this makes sense because a larva can only consume so much biomass relative to it’s size and the time.

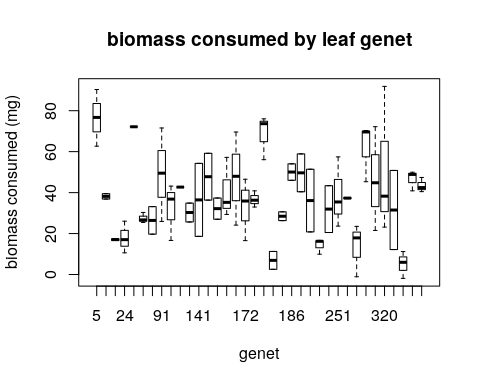
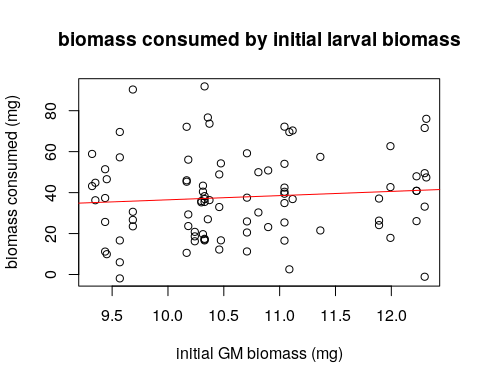
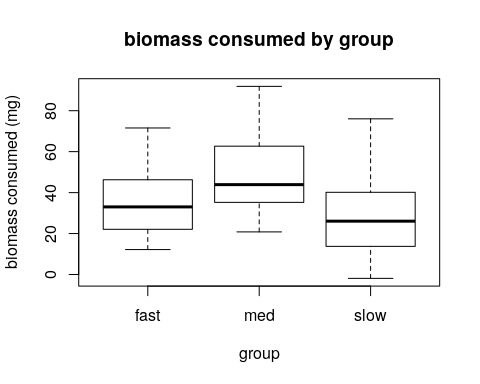
#### Leaf Consumption:

Next, we will look at factors that might affect consumption rates:

# create our consumption model  
cons.model <- lm(data = data.reduced, bmass.consumed\_group.cal ~ group +   
 genet.lf + gm.init.drymass\_mg)  
# print the anova table  
anova(cons.model)

## Analysis of Variance Table  
##   
## Response: bmass.consumed\_group.cal  
## Df Sum Sq Mean Sq F value Pr(>F)   
## group 2 6128.4 3064.18 12.9094 2.805e-05 \*\*\*  
## genet.lf 33 18518.6 561.17 2.3642 0.002654 \*\*   
## gm.init.drymass\_mg 1 16.8 16.78 0.0707 0.791378   
## Residuals 52 12342.8 237.36   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# print plots  
with(data.reduced,{  
 ## boxplots of consumption by group (all treatments pooled)  
 boxplot(bmass.consumed\_group.cal ~ group,  
 xlab = "group",  
 ylab = "biomass consumed (mg)",  
 main = "biomass consumed by group")  
 ## plot against initial larval biomass  
 plot(bmass.consumed\_group.cal ~ gm.init.drymass\_mg,  
 ylab = "biomass consumed (mg)",  
 xlab = "initial GM biomass (mg)",  
 main = "biomass consumed by initial larval biomass")  
 abline(lm(bmass.consumed\_group.cal ~ gm.init.drymass\_mg),  
 col = "red")  
 ## plot against genet  
 plot(bmass.consumed\_group.cal ~ genet.lf,  
 ylab = "biomass consumed (mg)",  
 xlab = "genet",  
 main = "biomass consumed by leaf genet")  
})



Overall, the moderate genotypes were the most consumed leaves. Genet is clearly an important covariate in this relationship but, surprisingly, larval size is not. We will still leave larval size in the model to allign with the literature and common practice. It is known that consumption does increase with larval size.

#### Outlier test:

We’ll do an official outlier test here:

# print and assign the outlier test to a variable  
(outlr.test <- outlierTest(cons.model,n.max = Inf))

## No Studentized residuals with Bonferonni p < 0.05  
## Largest |rstudent|:  
## rstudent unadjusted p-value Bonferonni p  
## 38 3.581615 0.00076137 0.066239

# test significance   
if (outlr.test$bonf.p <= outlr.test$cutoff){ #if the test is significant:  
 ## assign wording to the object (for printing in text body)  
 outlr.concl <- "is" #significance  
 # list of outliers (as rows of original dataset) (assign and print)  
 (outliers <- as.numeric(names(outlr.test[[1]])))  
 ## only the outlier row of data   
 out.obs <- data.reduced[outliers,]  
 ## select only relevant columns  
 out.obs %>% select("genet.lf","group","lf.init.drymass\_group.calib\_mg",  
 "lf.end.drymass\_mg","bmass.consumed\_group.cal",   
 "rel.bmass.cons\_group.cal")  
   
 # Show the biomass difference  
 (outlier.val <- round(data.reduced[outliers, "lf.end.drymass\_mg"],digits = 2))  
 # assign the rounded outlier variable for body printing   
 rel.cons.outlier <- round(data.reduced$rel.bmass.cons\_group.cal[outliers],3)  
# if the outlier test is not significant, just assign the conclusion value:  
} else {outlr.concl <- "is not"} #significance

There is not a statistical outlier.

## Visualization

Here I will further explore the data through visualization. I will also conduct some tests to check for variable significance.

Here I’ll add in the treatment grouping factor which is the combination of all groups present in one dish. Below is a subset of the data with the new treatment variable and the number of samples (dishes) in each treatment group.

# create the empty key  
treatment.key <- data.frame("Dish.ID" = data.reduced$Dish.ID, "treatment" = NA)  
  
# loop over all dishes and extract the groups present  
for (ID in data.reduced$Dish.ID){  
 trts <- unique(data.reduced$group[which(data.reduced$Dish.ID == ID)])  
 trts <- sort(trts)  
 ## paste the dishes groups together into one string  
 treatment <- paste(trts, collapse = ".")  
 ## at the treatment string to the key  
 treatment.key$treatment[which(treatment.key$Dish.ID == ID)] <- treatment  
}  
  
## check that there are only 4 treatments  
stopifnot(length(unique(treatment.key$treatment))==4)  
## if the IDs aren't equal, stop.  
stopifnot(all.equal(treatment.key$Dish.ID,data.reduced$Dish.ID))  
## same for number of observations  
stopifnot(nrow(treatment.key) == nrow(data.reduced))  
  
match.index <- match(x = treatment.key$Dish.ID, table = data.reduced$Dish.ID)  
  
data.reduced$treatment <- factor(treatment.key$treatment,levels = unique(treatment.key$treatment))  
  
## check that the number of rows hasn't changed:  
stopifnot(nrow(data.reduced) == reduced.n)  
  
# print:  
## show that it worked:  
data.reduced[data.reduced$Dish.ID %in% data.reduced$Dish.ID[c(1,5,31,80)],] %>% select(Dish.ID, group, treatment)

## Dish.ID group treatment  
## 1 10 fast fast.med  
## 2 10 med fast.med  
## 5 12B slow med.slow  
## 6 12B med med.slow  
## 31 2B fast fast.slow  
## 32 2B slow fast.slow  
## 79 48 slow fast.med.slow  
## 80 48 med fast.med.slow  
## 81 48 fast fast.med.slow

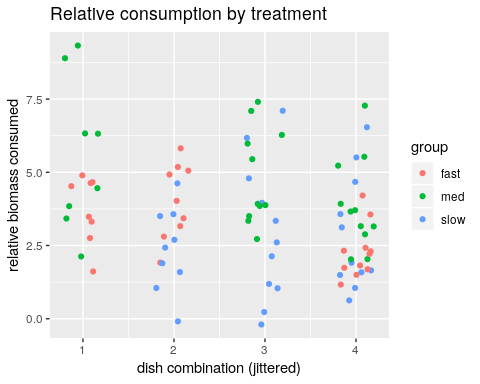
## print the treatment level n:  
(treatment.n <- data.reduced %>% group\_by(treatment) %>% summarise("dish n" = length(unique(Dish.ID))))

## # A tibble: 4 x 2  
## treatment `dish n`  
## <fct> <int>  
## 1 fast.med 8  
## 2 fast.slow 9  
## 3 med.slow 11  
## 4 fast.med.slow 11

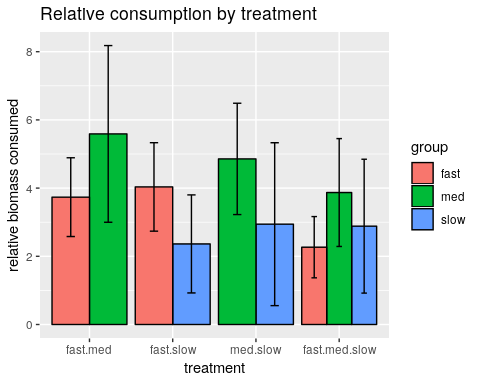
# Statistical Analyses

## Influence on rel. cons.

# plot the data points for each treatment:  
ggplot(data = data.reduced, aes(x = jitter(as.numeric(treatment)),  
 y = rel.bmass.cons\_group.cal,   
 fill = group)) +  
 geom\_point(aes(col = group)) +  
 labs(title = "Relative consumption by treatment",   
 y = "relative biomass consumed",   
 x = "dish combination (jittered)")



## summarize the grouped data by mean   
plot.table <- data.reduced %>%   
 group\_by(treatment, group) %>%   
 summarize(mean.consumed = mean(rel.bmass.cons\_group.cal),   
 sd.consumed = sd(rel.bmass.cons\_group.cal))  
  
  
ggplot(data = plot.table, aes(x = treatment,   
 y = mean.consumed,   
 fill = group)) +  
 geom\_bar(stat = "identity",   
 position = "dodge",   
 col = "black")+  
 labs(title = "Relative consumption by treatment",  
 y = "relative biomass consumed") +  
 geom\_errorbar(aes(ymin = mean.consumed - sd.consumed,  
 ymax = mean.consumed + sd.consumed),  
 width = .2,  
 position = position\_dodge(.9))



Let’s test the significance of these relationships:

# Full model  
full.model <- lm(data = data.reduced, rel.bmass.cons\_group.cal ~   
 group:treatment + # group within treatment  
 genet.lf + elapsed.time\_hours + dish.bmass.consumed  
 )  
# full model call  
formula(full.model)

## rel.bmass.cons\_group.cal ~ group:treatment + genet.lf + elapsed.time\_hours +   
## dish.bmass.consumed

## print type 2 anova  
(full.anova <- car::Anova(full.model,type = 2))

## Anova Table (Type II tests)  
##   
## Response: rel.bmass.cons\_group.cal  
## Sum Sq Df F value Pr(>F)   
## genet.lf 112.646 33 1.6814 0.05241 .  
## elapsed.time\_hours 0.531 1 0.2617 0.61144   
## dish.bmass.consumed 1.920 1 0.9457 0.33601   
## group:treatment 26.235 6 2.1537 0.06547 .  
## Residuals 91.360 45   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The above type II anova shows that group within treatment is moderately significant (). This suggests that there is a weak to moderate effect of our ‘vigor’ classification on larval leaf preference.

We also can see that the duration of time that a larva was allowed to feed in the dish did not have a significant effect on relative consumption but the genet of the leaves did.

# treatment-wise models  
## Loop over the treatment levels to create models and p values  
for(trt in levels(data.reduced$treatment)){  
 ## filter the data according to treatment  
 assign(paste0(trt,".data"),   
 as.data.frame(data.reduced[which(data.reduced$treatment == trt),]))  
 ## build model by treatment (slope and intercept)  
 assign(paste0(trt,".model"),   
 lm(data = get(paste0(trt,".data")), rel.bmass.cons\_group.cal ~   
 group +   
 elapsed.time\_hours +  
 dish.bmass.consumed  
 )  
 )  
 ## extract the p.values   
 assign(paste0(trt,".pval"), anova(get(paste0(trt,".model")))$`Pr(>F)`[1])  
}  
  
## create a table of the p values  
p.vals <- c("fast:med" = fast.med.pval,   
 "fast:slow" = fast.slow.pval,   
 "med:slow" = med.slow.pval,  
 "fast:med:slow" = fast.med.slow.pval)  
### make sig stars p < 0.1 = "." and p < 0.05 = "\*"  
sig.stars <- ifelse(p.vals <= .1,   
 yes = ifelse(p.vals <= .05, yes = "\*", no = "."),  
 no = "")  
## print the p values and sidnificance stars  
as.data.frame(rbind(round(p.vals,4),sig.stars),  
 row.names = c("p.vals", "significance"))

## fast:med fast:slow med:slow fast:med:slow  
## p.vals 0.0806 0.0159 0.036 0.0554  
## significance . \* \* .

Furthermore, when we look at the treatments individually we see that there was a significant () preference for fast over slow leaves, and med over slow. There was also a moderate () preference for med over slow leaves.

We also observed a moderately significant difference among our groups in the fast:med:slow treatment but to see the specifics, we need to test that treatment alone.

# Further testing for the triplets - we can see that there is a moderately  
 ## significant difference among groups within this treatment, but we  
 ## don't yet know how the groups differ from each other.  
  
## test for difference of variance (pooled or not)  
with(fast.med.slow.data,  
 ## relative biomass consumed - accounts for total biomass consumed and  
 ## trial time  
leveneTest(y = rel.bmass.cons\_group.cal /  
 (dish.bmass.consumed \* elapsed.time\_hours),   
 group = group)  
)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 2 2.1366 0.1357  
## 30

## with a p > 0.1, I will choose to pool the SD

The Levene’s Test p value of 0.18 means that the variances for each group are not significantly different from each other. Therefore, when testing for difference in mean, a pooled standard deviation should be used. Below is that pair-wise t-test:

## test for difference of mean  
trip.test <- with(fast.med.slow.data,{  
 pairwise.t.test(x = rel.bmass.cons\_group.cal /  
 (elapsed.time\_hours \* dish.bmass.consumed),   
 g = group, pool.sd = TRUE)  
})   
  
round(trip.test$p.value,4)

## fast med  
## med 0.0302 NA  
## slow 0.3782 0.1477

# noquote(ifelse(trip.test$p.value <= .05, yes = "\*", no = ""))

med and fast leaves are the only groups for which differences in means are significant. The other differences are not significant.

### Conclusions

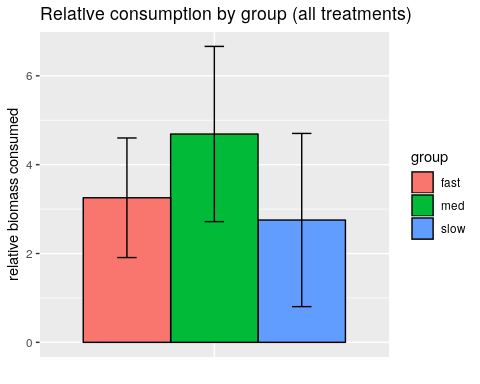
In conclusion (based on only significant results):

1. In a choice between fast and slow-growing genets, larvae preferred fast growers
2. in a choice between moderate- and slow-growing genets, larvae preferred moderate growers
3. in a choice between all 3 groups, moderate-growing genets were preferred over fast growers. **Note:** This is supported by the trend (non-significant) of larvae preferring moderate over fast genets in the first paired treatment.

## Full pairwase t-test

We’ll do another pairwise t-test of the entire data set (ignoring treatment) to test for overall differences in the trial.

## summarize the grouped data by mean   
plot.table <- data.reduced %>%   
 group\_by(group) %>%   
 summarize(mean.consumed = mean(rel.bmass.cons\_group.cal),   
 sd.consumed = sd(rel.bmass.cons\_group.cal))  
  
  
ggplot(data = plot.table, aes(x = "", y = mean.consumed, fill = group)) +  
 geom\_bar(stat = "identity",   
 position = "dodge",   
 col = "black")+  
 labs(title = "Relative consumption by group (all treatments)",  
 y = "relative biomass consumed") +  
 theme(axis.title.x = element\_blank(), axis.ticks.x = element\_blank())+  
 geom\_errorbar(aes(ymin = mean.consumed - sd.consumed,  
 ymax = mean.consumed + sd.consumed),  
 width = .2,  
 position = position\_dodge(.9))



## test for difference of variance (pooled or not)  
l.test <- with(data.reduced,  
 ## relative biomass consumed - accounts for total biomass consumed and  
 ## trial time  
leveneTest(y = rel.bmass.cons\_group.cal /  
 (dish.bmass.consumed \* elapsed.time\_hours),   
 group = group)  
)  
# if the p value is not significant, use pooled sd  
pool <- (l.test$`Pr(>F)`[1] > .1)  
## test for difference of mean  
trip.test <- with(data.reduced,{  
 pairwise.t.test(x = rel.bmass.cons\_group.cal /  
 (elapsed.time\_hours \* dish.bmass.consumed),   
 g = group, pool.sd = pool)  
})   
  
round(trip.test$p.value,4)

## fast med  
## med 0.0521 NA  
## slow 0.1342 7e-04

Overall, there is a strong preference for med over slow leaves and a moderate preference for fast over med when accounting for elapsed time, total biomass consumed by the larva, and vigor group.

### Conclusions

# Needed

# Write updated data

# remove uneccessary columns  
data.reduced <- data.reduced %>% select(-c(slope,r.sqr,intcpt,ordered.levels,  
 bmass.consumed\_no.cal,bmass.consumed\_genet.cal,  
 lf.init.drymass\_no.calib\_mg,lf.init.drymass\_genet.calib\_mg,  
 rel.bmass.cons\_genet.cal,rel.bmass.cons\_genet.cal,  
 start.time\_2400,end.time\_2400)) %>%  
 ## copy the drymass estimate into the original column  
 mutate(lf.init.drymass\_mg = lf.init.drymass\_group.calib\_mg)  
  
write.csv(data.reduced,file = "data/Choice\_Assay/leaf-and-larvae-data\_long\_calibrated-&-reduced.csv")

write.csv(choice.assay.data,file = "data/Choice\_Assay/leaf-and-larvae-data\_long\_calibrated.csv")