# Box 5.2 Estimating biomass using a length weight relationship

Commonly a subset of fish captured are measured for length and weight and the remaining fish are measured for length only. This is advantageous as the variance for total biomass can be reduced by accounting for variability in weight among fish by exploiting the relationship between length and weight. The length weight relationship predicts the expected weight of fish where only length was measured and the estimate of total biomass can be estimated by with a non-parametric boostrap that resamples the model residuals to make many new datasets of predicted values and then biomass estimated by summing the predicted weights. The 95% confidence intervals calculated by finding the 0.025 and the 0.975 quantiles. However the bias associated with predicting weight from length propagates when estimating biomass for a sample and results in an underestimate of biomass and therefore some caution is needed to account for the bias and some resampling to quantify the uncertainty in the biomass estimate. This box demonstrates: 1) how to fit length weight data and estimate weight, albeit biased, 2) apply a bias correction to estimated weights, 3) a non-parametric bootstrap approach that shuffles the residuals among observed values to estimate 95% confidence intervals for the estimated biomass.

# The data

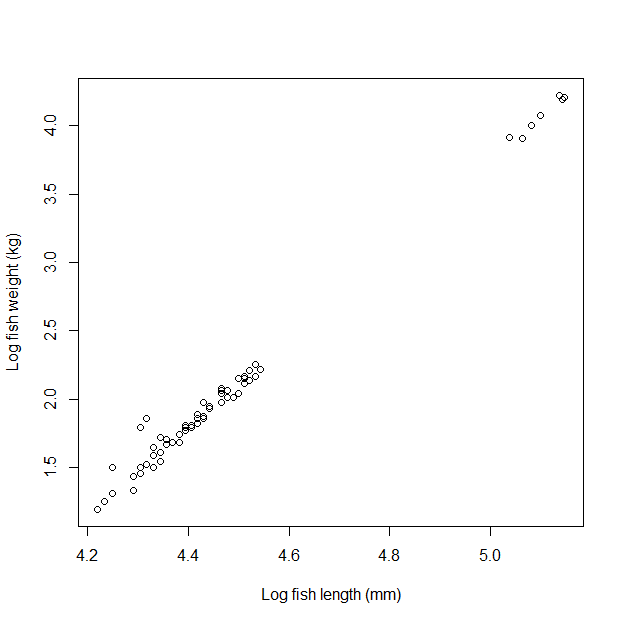
The data is 2 columns, length and weight for yellow bass *Morone mississippiensis* from Clear Lake, IA, in mm and g respectively. There were 104 yellow bass in th sample. Length was measured on all captured yellow bass and weight was measured for 63 fish. To estiamte biomass for the sample 41. The code chunk below reads in the data, looks at the first 6 rows, and plots the data with a rug plot of the underlying lengths. The rug plots illustrates where there may be fish that have lengths but not weight and is useful to ensure the data used to estimate the weight length relationship covers the lengths that need an estimated weight.

ylb\_lw<-read.csv("yellow-bass.csv")  
# look at the first few rows of the data  
head(ylb\_lw)  
# plot the data  
plot(weight\_g~length\_mm,data=ylb\_lw,  
 xlab="Length (mm)",  
 ylab="Weight (g)")  
# add a rug plot to see distribution of lengths  
# that do not have lengths to verify adequate coverage  
rug(ylb\_lw[which(!is.na(ylb\_lw$weight\_g)),]$length\_mm)

## Fitting the length weight relationship

The expected weight for a given length is estimated by log linear regression in the R code chunk below. First the length and weight data are log transformed with the log() function. In R the default for the log function is the natural log (i.e., ). Then a linear model is fit to the log transformed data. The linearized data is visualized using the plot() function.

ylb\_lw$ln\_length<-log(ylb\_lw$length)  
ylb\_lw$ln\_weight<-log(ylb\_lw$weight)  
plot(ln\_weight~ln\_length,data=ylb\_lw,  
 xlab="Log fish length (mm)",  
 ylab="Log fish weight (kg)")

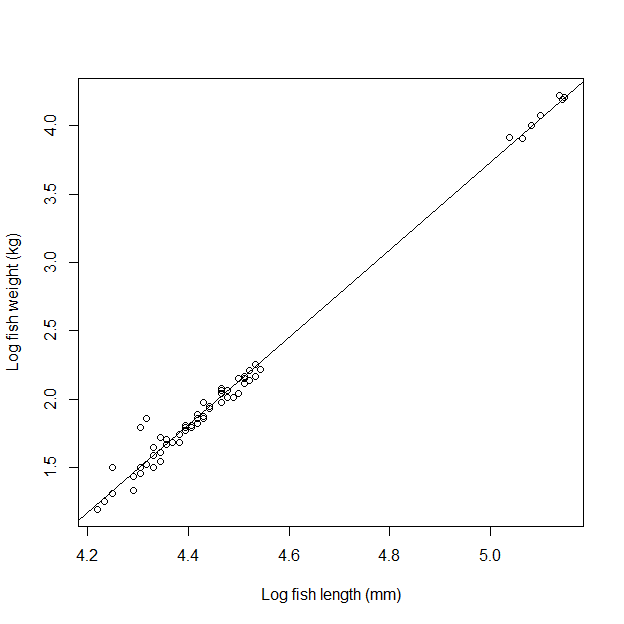


A linear regression is fit to the log transformed length and weight data using the lm() function. And the model output summarized and model fit added to the plot of raw data using the abline() function to visualize model fit.

# fit the length weight model for log transformed length  
# and weight  
fit<- lm(ln\_weight~ln\_length,data=ylb\_lw)  
# linear model summary information  
summary(fit)

##   
## Call:  
## lm(formula = ln\_weight ~ ln\_length, data = ylb\_lw)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.12195 -0.04186 -0.01244 0.01789 0.31261   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -12.31097 0.17976 -68.48 <2e-16 \*\*\*  
## ln\_length 3.20896 0.04006 80.10 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07431 on 61 degrees of freedom  
## (41 observations deleted due to missingness)  
## Multiple R-squared: 0.9906, Adjusted R-squared: 0.9904   
## F-statistic: 6416 on 1 and 61 DF, p-value: < 2.2e-16

# plot the raw log transformed length and weight data  
plot(ln\_weight~ln\_length,data=ylb\_lw,  
 xlab="Log fish length (mm)",  
 ylab="Log fish weight (kg)")  
# add a line of predicted weights on log scale  
abline(coef(fit)[1],coef(fit)[2])



## Bias corrected predicted weight

The variance of the predicted weight is used to correct the bias in the predicted weight. To calculate the bias corrected weights for a given length the predict() function is used to get the uncorrected predicted weight.

# calculate the estimated mean weight for a fish of  
# a given length  
ylb\_lw$predicted\_ln\_weight<-predict(fit,  
 newdata=data.frame(ln\_length=ylb\_lw$ln\_length))  
# take a look at the predicted mean weights and standard errors  
head(ylb\_lw)

## length\_mm weight\_g ln\_length ln\_weight predicted\_ln\_weight  
## 1 80 NA 4.382027 NA 1.750786  
## 2 81 NA 4.394449 NA 1.790649  
## 3 82 NA 4.406719 NA 1.830023  
## 4 90 NA 4.499810 NA 2.128747  
## 5 75 NA 4.317488 NA 1.543684  
## 6 90 NA 4.499810 NA 2.128747

The overall model mean squared error is used to peform the bias correction which is extracted from the fitted lm() object and squared to get the means squared error.

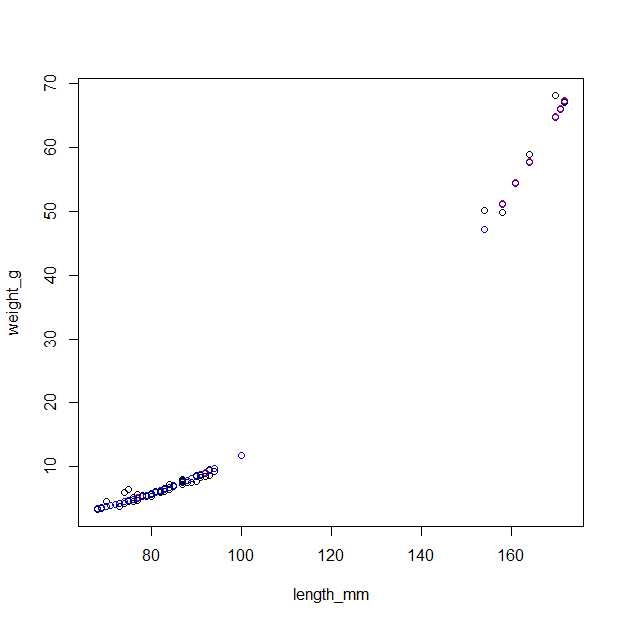
# get mean squared error from model fit  
mse<-summary(fit)$sigma^2

The next code chunk is a bit complicated but in order to correct the predictions the first step is to create the model design matrix and solve for the inverse. This process needs to be applied to each predicted. The bias correction is then the predicted weight on log scale plus the variance divided by 2.

X<-model.matrix(~ln\_length,ylb\_lw)  
X\_prime<-t(X)# transpose the design matrix  
ylb\_lw$var\_i<-sapply(1:nrow(ylb\_lw),function(i)  
 {  
 mse\*(1+t(as.matrix(X[i,1:2]))%\*%(X\_prime%\*%X)^-1%\*%as.matrix(X[i,1:2]))  
 })  
ylb\_lw$W\_hat\_corrected<- exp(ylb\_lw$predicted\_ln\_weight+ylb\_lw$var\_i/2)

Plotting the corrected and the uncorrected predicted weights shows that the corrected weight is slighly higher, reflecting the bias correction.

plot(weight\_g~length\_mm,ylb\_lw)  
points(W\_hat\_corrected~length\_mm,ylb\_lw,col="red")  
# uncorrected predicted weight  
ylb\_lw$W\_hat\_uncorrected<-exp(ylb\_lw$predicted\_ln\_weight)  
points(W\_hat\_uncorrected~length\_mm,ylb\_lw,col="blue")



The corrected weight can now be used to estimate biomass by summing up the values. Comparing the corrected and the uncorrected values, you notice the uncorrected biomass is less than the bias corrected biomass.

biomass<-sum(ylb\_lw$W\_hat\_corrected)  
biomass\_uncorrected<-sum(ylb\_lw$W\_hat\_uncorrected)

The log transformation used to linearize the length weight relationship makes calculating a confidence interval for the biomass estimate difficult and in some cases unreasonable because of the long tails the occur with a log normal distribution. Bootstrapping provides a practical way to correct the bias associated with the log transformation and quantify the uncertainty around the estimated biomass. The code chunk below uses the simpleboot package to resample the residuals from the fitted model and the model estimated mean squared error to estimate unbiased weight and then sum up the predictions for 1000 replicates. The 0.025 and the 0.975 quantiles are used for the 95% confidence intervals.

library(simpleboot)

## Simple Bootstrap Routines (1.1-7)

# fit the model, as done above  
fit<-lm(ln\_weight~ln\_length,ylb\_lw)  
# mean squared error from the fitted model  
mse<-summary(fit)$sigma^2   
# the design matrix  
X<-model.matrix(~ln\_length,ylb\_lw)  
# transpose of the design matrix  
X\_prime<-t(X)  
# resample the residuals from the fitted model  
lboot <- lm.boot(fit, R = 1000,rows=FALSE)

The result of the bootstrap is contained in a list the first element of the list is returned below.

lboot$boot.list[[1]]

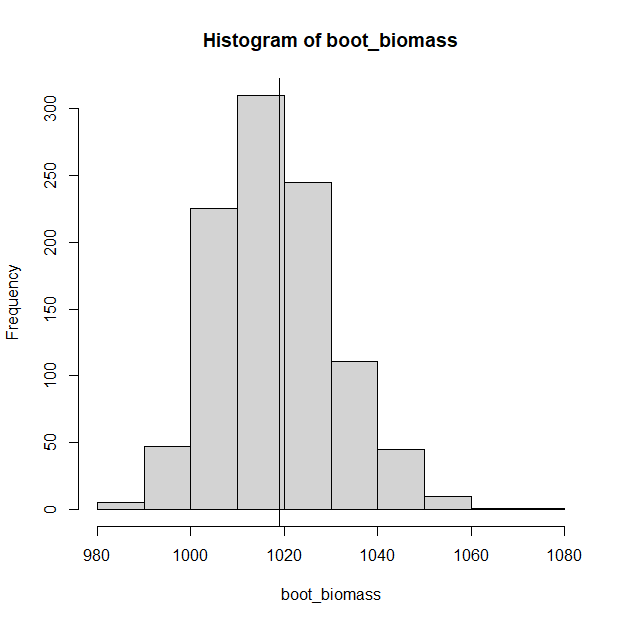
## $coef  
## (Intercept) ln\_length   
## -12.352048 3.216623   
##   
## $rss  
## [1] 0.1935874  
##   
## $rsquare  
## [1] 0.999362  
##   
## $rstderr  
## [1] 0.05633439  
##   
## $fitted  
## 1 2 3 4 5 6 7 8   
## 1.220516 1.250667 1.280819 1.310970 1.341122 1.371273 1.401424 1.431576   
## 9 10 11 12 13 14 15 16   
## 1.461727 1.491878 1.522030 1.552181 1.582332 1.612484 1.642635 1.672786   
## 17 18 19 20 21 22 23 24   
## 1.702938 1.733089 1.763240 1.793392 1.823543 1.853694 1.883846 1.913997   
## 25 26 27 28 29 30 31 32   
## 1.944148 1.974300 2.004451 2.034602 2.064754 2.094905 2.125057 2.155208   
## 33 34 35 36 37 38 39 40   
## 2.185359 2.215511 2.245662 2.275813 2.305965 2.336116 2.366267 2.396419   
## 41 42 43 44 45 46 47 48   
## 2.426570 2.456721 2.486873 2.517024 2.547175 2.577327 2.607478 2.637629   
## 49 50 51 52 53 54 55 56   
## 2.667781 2.697932 2.728083 2.758235 2.788386 2.818537 2.848689 2.878840   
## 57 58 59 60 61 62 63 64   
## 2.908992 2.939143 2.969294 2.999446 3.029597 3.059748 3.089900 3.120051   
## 65 66 67 68 69 70 71 72   
## 3.150202 3.180354 3.210505 3.240656 3.270808 3.300959 3.331110 3.361262   
## 73 74 75 76 77 78 79 80   
## 3.391413 3.421564 3.451716 3.481867 3.512018 3.542170 3.572321 3.602472   
## 81 82 83 84 85 86 87 88   
## 3.632624 3.662775 3.692927 3.723078 3.753229 3.783381 3.813532 3.843683   
## 89 90 91 92 93 94 95 96   
## 3.873835 3.903986 3.934137 3.964289 3.994440 4.024591 4.054743 4.084894   
## 97 98 99 100   
## 4.115045 4.145197 4.175348 4.205499

The sapply() function takes each list element and estimates the predicted weight for each weight, corrects the bias, and then returns the estimated biomass for each bootstrap replicate.

boot\_biomass<-sapply(lboot$boot.list,function(xx)  
 {  
 mse<-xx$rstderr^2  
 var\_i<-sapply(1:nrow(ylb\_lw),function(i)  
 {  
 mse\*(1+t(as.matrix(X[i,1:2]))%\*%(X\_prime%\*%X)^-1%\*%as.matrix(X[i,1:2]))  
 })  
   
 return(sum(exp(xx$coef[1]+xx$coef[2]\*ylb\_lw$ln\_length+var\_i/2)))  
 })

The vector of biomass values are plotted below and the 95% confidence intervals determined using thequantile() function. The confidence interval is narrow as we would expect given the relationship of length and weight.

hist(boot\_biomass)  
abline(v=biomass)



ci<-quantile(boot\_biomass,prob=c(0.025,0.975))  
ci

## 2.5% 97.5%   
## 997.2802 1045.0269

# References

Colvin, M. E., C. L. Pierce, T. W. Stewart, and S. Grummer. 2012. Strategies to control a common carp (*Cyprinus carpio*) population by pulsed commercial harvest. North American Journal of Fisheries Management 32:1251-1264.