Variola louti

Eva Schemmel

2023-12-07

Life history analysis for *Variola louti* from Guam

Code provided for the analyis of size/age at maturity, size/age at sex change, spawning season and lunar spawning frequency. Bayesian growth model and extracting SST from ERDDAP for comparision in edge analysis are in separate R files.

1. Load packages, and load and clean data.

```
library(dplyr)
library(reshape2)
library(lme4)
library(boot)
library(magrittr)
library(plotrix)
library(lubridate)
library(plyr)
library(car)
library(lunar)
library(mgcv)
library(itsadug)
library(ggplot2)
library(ggpubr)

#VALO Data
valo_ar<-read.csv("valo_data.csv")

#sub data sets for plotting use later
valo_ar$H_Sex<-as.factor(valo_ar$H_Sex)
valo_f<-valo_ar %>%
  subset(H_Sex=="F") #female
valo_m<-valo_ar %>%
  subset(H_Sex=="M" ) #male
valo_m_po<-valo_ar %>%
  subset(H_Sex=="M_PO" ) #male with primary oocytes
valo_t<-valo_ar %>%
  subset(H_Sex=="T" ) #transitional
```

##Spawning Season

First assess spawning season, then can assess L50. Need to constrain data for L50 to within the spawning season. Since some spawning occurs all year for this species (spawning capable individuals found all year), all the data is used for determining L50. Also look at relationships with gonadosomatic index (GSI) by fish length and reproductive phase.

```
gsi<- valo_ar%>% mutate(GSI=GonWeight/Weight.g.*100)
gsi$Month<-month(gsi$Date, label=TRUE) #get monthly labels for summarizing and plotting

#separate sexes for analysis
gsi_m<-subset(gsi, H_Sex=="M") #males
gsi<-subset(gsi, H_Sex=="F") #females
gsi<- subset(gsi, GSI>=0.00001) #clean and remove NA and zeros

###GSI means and sd by reproductive phase###
gsi<- gsi[complete.cases(gsi$Length.cm.),]  #subset data to get records that have Fork Lengths
gsi_phase<-gsi %>%
  dplyr::group_by(Phase) %>%
  dplyr::summarise(mean=mean(GSI), n=length(GSI), Sd=sd(GSI))
gsi_phase$SE <-gsi_phase$Sd/sqrt(gsi_phase$n) #adding standard error to the dataframe

gsi_phase_oocyte<-gsi %>%
  dplyr::group_by(Oocyte_Stage) %>%
  dplyr::summarise(mean=mean(GSI), n=length(GSI), Sd=sd(GSI))

#developing phase gsi and sd (VTI and VTII)
developing_gsi<-gsi %>% subset(Oocyte_Stage=="VTI" | Oocyte_Stage=="VTII")
developing_gsi<-developing_gsi %>%
  dplyr::summarise(mean=mean(GSI), n=length(GSI), Sd=sd(GSI))
developing_gsi$SE <-developing_gsi$Sd/sqrt(developing_gsi$n)

####GSI vs FL relationship#########
gsi<- gsi[complete.cases(gsi$GSI),]
gsi <- gsi[order(gsi$Length.cm.),]
gsi_sc<-gsi %>% subset(Phase=="SC" |Phase=="AS")
gsi_sc$GFBW=gsi_sc$Weight.g.-gsi_sc$GonWeight

######FL and GSI plots####
{par(mfrow=c(1,2),mar=c(4,4,3,3))
plot(gsi$GSI~gsi$Length.cm., bty="l", ylab="GSI", xlab="Fork Length (cm)", pch=19, las=1,xaxs="i",yaxs=
gsiU<-gsi %>% subset(Phase=="U")
gsiD<-gsi %>% subset(Phase=="D"|Phase=="RG"|Phase=="RE")
gsiSC<-gsi %>% subset(Phase=="SC"|Phase=="U")
points(gsiU$GSI~gsiU$Length.cm., pch=21,col="black",bg="white", add=TRUE)
points(gsiD$GSI~gsiD$Length.cm., pch=21,col="black",bg="grey", add=TRUE)
text(21,9.5, "a")
plot(log(gsi_sc$GonWeight)~gsi_sc$GFBW, bty="l", ylab="log GW", xlab="GFBW", pch=19,xaxs="i",yaxs="i",xl
axis(1, seq(200,1400,by=200), labels =seq(200,1400,by=200) )
expr<-lm(log(gsi_sc$GonWeight)~gsi_sc$GFBW)
leng<-seq(100,1200,by=10)
lines(gsi_sc$GFBW,(predict(expr, list(x=leng))), lwd=2)
text(260,4.8, "b")}
```

```
summary(expr)
```

```
##
```

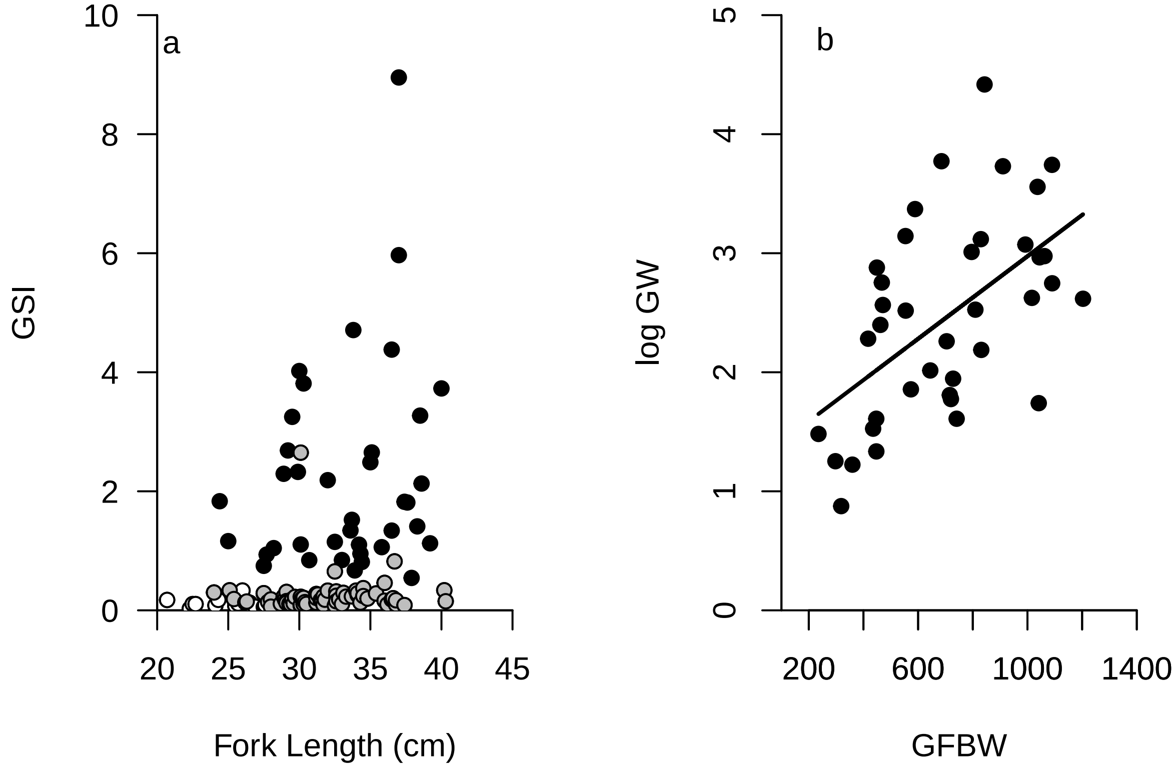


Figure 1: Female Guam Variola louti gonadosomatic index (GSI) versus fork length, and spawning capable and actively spawning female log gonad weight (GW) versus gonad free body weight (GFBW) (log GW = 1.3860 + GFBW * 0.0016, df = 40, R2=0.23) for fish sampled from Guam between 2010 and 2017. Spawning capable and actively spawning females are black, immature females are white, and all other female reproductive phases are grey.

```
## Call:
## lm(formula = log(gsi_sc$GonWeight) ~ gsi_sc$GFBW)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3054 -0.5015 -0.1432  0.4904  1.7153
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.2412614  0.3307442   3.753 0.000616 ***
## gsi_sc$GFBW 0.0017330  0.0004428   3.914 0.000387 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7089 on 36 degrees of freedom
## Multiple R-squared:  0.2985, Adjusted R-squared:  0.279
## F-statistic: 15.32 on 1 and 36 DF,  p-value: 0.0003873
```

```
#####box plot spawning season#####
#female and sample sizes per month
guam_total<-as.data.frame(gsi %>%
                            dplyr::group_by(Month)%>%
                            dplyr::summarise(total=length(Length.cm.))) #female
male_total<-as.data.frame(gsi_m %>%
                            dplyr::group_by(Month)%>%
                            dplyr::summarise(total=length(Length.cm.))) #male


{par(mfrow=c(2,1),mar=c(3.5,3.5,1.5
                       ,1),mgp=c(2,.7,0))
boxplot(gsi$GSI~gsi$Month,bty="l",ylab="GSI",xlab="", ylim=c(0.0,10),frame=T,cex.axis=1.0, cex.lab=1.0,
par(xpd=TRUE)
text(x =guam_total$Month, y=11.2, label = guam_total$total , offset=0, cex = 1.0)#too add sample size a
text(0.3,9, "a", cex=1.0)
par(xpd=TRUE)
boxplot(gsi_m$GSI~gsi_m$Month,bty="l",ylab="GSI",xlab="", ylim=c(0,.5), frame=T,cex.axis=1.0, cex.lab=1
text(x =male_total$Month, y=0.56, label = male_total$total , offset=0, cex = 1.0)#too add sample size a
text(0.3,.45, "b", cex=1.0)}
```

Proportion of Female Reproductive Phases per Month

```
###female histo reproductive phase per month frequency graph###
#number of females in each reproductive phase
temp<-gsi
temp <- temp[complete.cases(temp$Length.cm.),]
temp$PHASE<-factor(temp$Phase, levels=c( "AS","SC","D","RG","RE", "U"), ordered=TRUE)
tempAS<-subset(temp, PHASE=="AS")
tempSC<-subset(temp, PHASE=="SC")
tempD<-subset(temp, PHASE=="D")
tempRG<-subset(temp, PHASE=="RG")
tempRE<-subset(temp, PHASE=="RE")
```

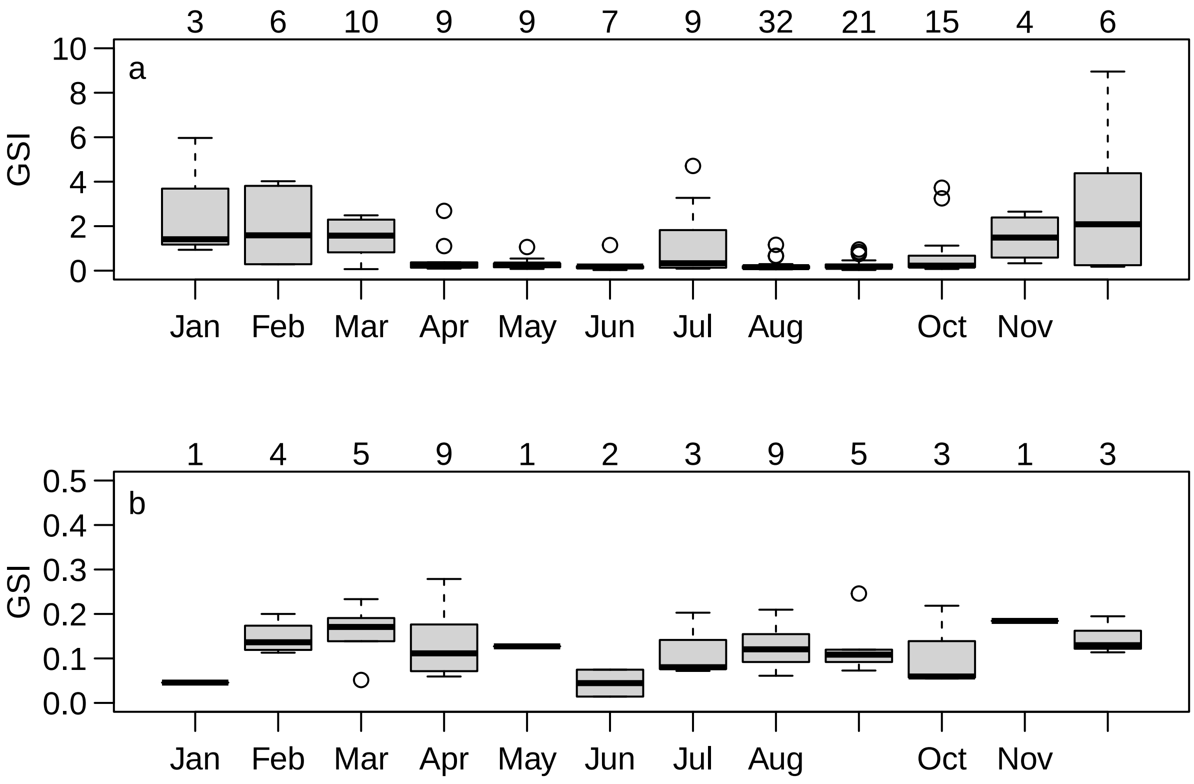


Figure 2: Median and interquartile range of gonadosomatic index (GSI) for mature female (a) and male (b) Variola louti by month

```
tempU<-subset(temp, PHASE=="U")

#to compare the lengths of immature and regenerating females - this can be useful to look for outliers
#hist(tempRE$Length.cm.)
#hist(tempU$Length.cm.)

##percent of females that are immature
fem_total<-length(temp$PHASE)
u=length(tempU$Phase)
immaturef<-u/fem_total

prop_month_cast <- dcast(temp, Month~PHASE, value.var="PHASE")

prop_month_cast$sum <- rowSums(prop_month_cast[2:7])# make sure it adds to 100%
prop_month_cast$AS<-(prop_month_cast$AS/prop_month_cast$sum)*100
prop_month_cast$D<-(prop_month_cast$D/prop_month_cast$sum)*100
prop_month_cast$SC<-(prop_month_cast$SC/prop_month_cast$sum)*100
prop_month_cast$RE<-(prop_month_cast$RE/prop_month_cast$sum)*100
prop_month_cast$U<-(prop_month_cast$U/prop_month_cast$sum)*100
prop_month_cast$RG<-(prop_month_cast$RG/prop_month_cast$sum)*100

# ###add in missing months of sampling if there are any, example below
# #install.packages("tibble")
# prop_month_cast<-add_row(prop_month_cast, Month=6, SC=0, D=0, RE=0,RG=0, U=0, sum=0)

prop_month_cast<-prop_month_cast %>% arrange(Month)
prop_month_cast <- prop_month_cast[complete.cases(prop_month_cast$Month),]

counts.L <- t(prop_month_cast[2:7])
count.labL <- c("Jan", "Feb", "Mar", "Apr", "May","Jun", "Jul","Aug","Sep","Oct","Nov","Dec")
sample_n=prop_month_cast$sum

{par(mfrow=c(1,1),mar=c(4,5,2,11))
sz<-barplot(counts.L,
            names.arg=count.labL, cex.lab=1.0, cex.axis=1.0,xaxs="i",yaxs="i",cex=1.0,ylab="Percent Fre
par(xpd=TRUE)
legend(15,80,bty="n",legend=c("Actively Spawning", "Spawning Capable","Developing","Regenerating","Regr
       ,col=c( "firebrick", "tan3","palegoldenrod", "mediumpurple2","lightcyan2", "steelblue"),pch=15, 
text(x = sz, y=104, label = sample_n , offset=10, cex = 1.0)}#too add sample size above plot
```

Lunar Spawning Assessment

```
#####Lunar Spawning#####
#uses libraries lunar, mgcv, itsadug

#subsetting data for functionally mature females and males
gsi<-gsi %>% subset(M_VT==1) #functionally mature females
gsi_m<-gsi_m %>% subset(Mature==1) #mature males

gsi$Date<-as.Date(gsi$Date)
gsi$lunar.phase<-lunar::lunar.phase(gsi$Date,shift=-10, name=8)
```

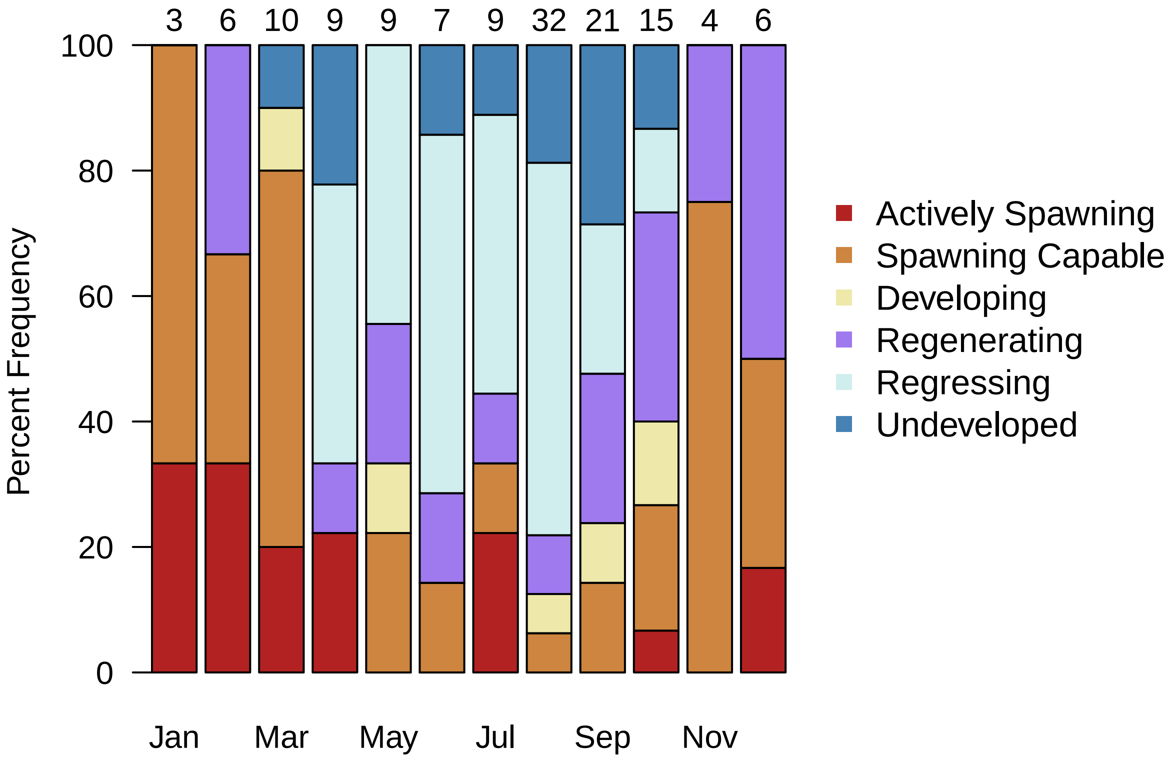


Figure 3: Trend in reproductive stages of female Variola louti by month for fish sampled from Guam between 2010 and 2017. Developing stages (Developing I & Developing II) were grouped together into "Developing." Numbers above each bar indicate aggregate monthly sample sizes.

```
lunar_total<-as.data.frame(gsi %>%
                            dplyr::group_by(lunar.phase)%>%
                            dplyr::summarise(total=length(Length.cm.)))

gsi <-
  gsi  %>%
  mutate(lunar_day = lunar.phase(as.Date(Date), shift=-10) / 0.212769)
gsi$lunar_day<-round(gsi$lunar_day,0)

# GAM uses library mgcv
#testing two models and which fits the data better
mod1<-gam(log(GSI)~Length.cm.+s(lunar_day), data=gsi)
summary(mod1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(GSI) ~ Length.cm. + s(lunar_day)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.39638    0.90437  -3.756 0.000285 ***
## Length.cm.   0.07758    0.02788   2.783 0.006391 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                edf Ref.df     F p-value
## s(lunar_day) 5.331  6.433 3.764 0.00152 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.216   Deviance explained = 26.1%
## GCV = 1.2339  Scale est. = 1.1531    n = 112
```

```
mod2<-gam(log(GSI)~s(lunar_day), data=gsi)
summary(mod2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(GSI) ~ s(lunar_day)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.8954     0.1048  -8.541 1.06e-13 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Approximate significance of smooth terms:
##                  edf Ref.df     F p-value
## s(lunar_day) 5.108  6.192 3.637 0.00236 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.163   Deviance explained = 20.2%
## GCV =  1.302  Scale est. = 1.231     n = 112
```

```
#To look at models graphically
#plot.gam(mod1, shade=TRUE)
#plot.gam(mod2, shade=TRUE)

#use AIC to select best model
AIC(mod1, mod2)
```

```
##            df      AIC
## mod1 8.330877 342.8787
## mod2 7.108468 349.0515
```

```
#to create a box plot of GSI by lunar day need to bin GIS by lunar day - used 3 day bins
gsitemp=gsi
brks <- seq(from=0,to=30,by=3)
gsitemp$binl <- cut(gsitemp[,"lunar_day"], breaks=brks, labels=brks[1:(length(brks)-1)],right=F)
gsitemp$binl<-as.factor((gsitemp$binl))
gsitemp<-gsitemp %>%
  dplyr::arrange(desc(lunar_day))%>%
  dplyr::group_by(binl)%>%
  mutate(Mean=mean(GSI, na.rm=T),lower=quantile(GSI,probs = 0.025, na.rm=TRUE), upper=quantile(GSI,probs
gsitemp$binl<-as.numeric(as.character((gsitemp$binl)))

gsilunarsum<-gsitemp %>%
  dplyr::group_by(binl) %>%
  dplyr::summarise(total=n())
gsilunarsum<-gsilunarsum[1:10,]
gsilunarsum$lunar_day<-gsilunarsum$binl

#make shift code to line up box plots on the x axis over the median lunar day for the bin
c(0,3,6,9,12,15,18,21,24,27)
```

```
##  [1]  0  3  6  9 12 15 18 21 24 27
```

```
gsitemp1<-gsitemp%>%
 ungroup()%>%
  select(GSI, binl)
l0<-gsitemp1%>%
  subset(binl==0)
l0<-l0 %>%
  select(GSI)
l3<-gsitemp1%>%
  subset(binl==3)
```

```
l6<-gsitemp1%>%
  subset(binl==6)
l9<-gsitemp1%>%
  subset(binl==9)
l12<-gsitemp1%>%
  subset(binl==12)
l15<-gsitemp1%>%
  subset(binl==15)
l18<-gsitemp1%>%
  subset(binl==18)
l21<-gsitemp1%>%
  subset(binl==21)
l24<-gsitemp1%>%
  subset(binl==24)
l27<-gsitemp1%>%
  subset(binl==27)
xplace=as.numeric(c(0,3,6,9,12,15,18,21,24,27))

#lunar spawning figure#
{plot(gsitemp$Mean~gsitemp$lunar_day, ylab="GSI", xlab="Lunar Day", bty="n",frame=T,cex.axis=1.0,ylim=c
#axis(1,at=c(seq(0,30,by=3)), cex.lab=1.0,cex.axis=1.0, las=1)
boxplot(l0$GSI, l3$GSI, l6$GSI, l9$GSI, l12$GSI,l15$GSI,l18$GSI,l21$GSI,l24$GSI,l27$GSI,
        at = c(1.5,4.5,7.5,10.5,13.5,16.5,19.5,22.5,25.5,28.5),
        names = c("0", "3", "6", "9","12","15","18","21","24","27"),
        las = 2,
        boxwex=3, add=TRUE, bty='n', xaxt="n")
plot_smooth(mod1, view="lunar_day", lwd=2,
            transform = exp, se=FALSE,rug=FALSE, plot_all=NULL,bty='n',add=TRUE) }#plot best fit gam mo
```

```
## Summary:
##  * Length.cm. : numeric predictor; set to the value(s): 31.8.
##  * lunar_day : numeric predictor; with 30 values ranging from 0.000000 to 30.000000.
##  * NOTE : No random effects in the model to cancel.
##
```

Size/age at Maturity

```
#L50 functions
# setup functions ---------------------------------------------------------
data.setup <- function(dataframe,xname,yname){
  templ <- dataframe
  brks <- seq(from=min(templ[,xname]),to=max(templ[,xname]),by=2)
  templ$binl <- cut(templ[,xname], breaks=brks, labels=brks[1:(length(brks)-1)],right=F)
  templ <- dcast(templ, binl~templ[,yname])
  colnames(templ) <- c("binl","a","b")
  templ$binl <- as.character(templ$binl)
  templ$binl <- as.numeric(templ$binl)
  templ$sum <- rowSums(templ[2:3])
  templ$a.p <- templ$a/templ$sum
  templ$b.p <- templ$b/templ$sum
  return(templ)
```

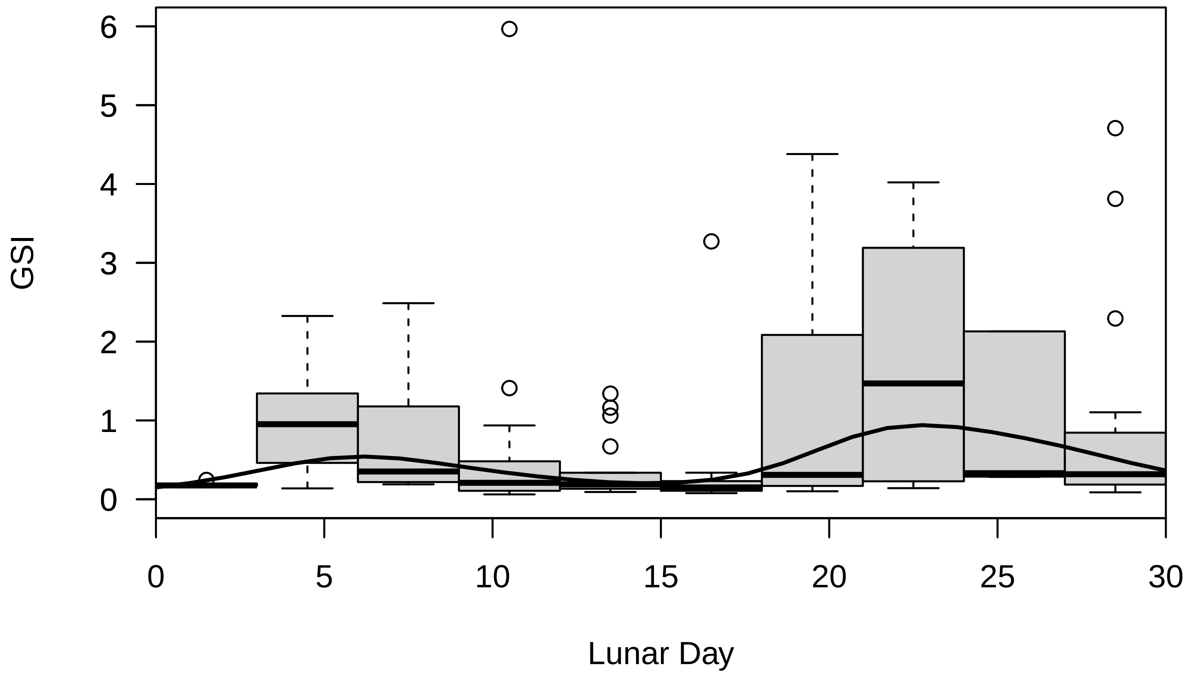


Figure 4: Generalized additive model GSI ~ Fork Length+s(Lunar Day) fit to raw data points (black line) by lunar day (3 day mean).

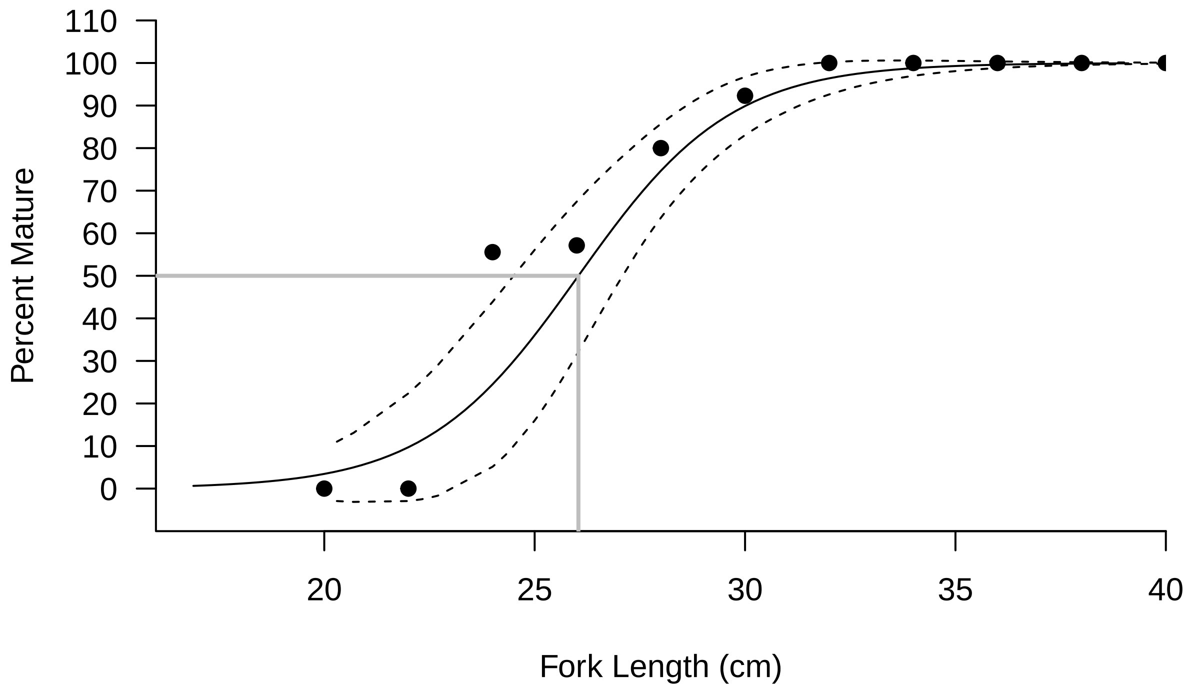
```
}

data.setup <- function(dataframe,xname,yname){
  templ <- dataframe
  brks <- seq(from=20,to=50,by=2)
  templ$binl <- cut(templ[,xname], breaks=brks, labels=brks[1:(length(brks)-1)],right=F)
  templ <- dcast(templ, binl~templ[,yname])
  colnames(templ) <- c("binl","a","b")
  templ$binl <- as.character(templ$binl)
  templ$binl <- as.numeric(templ$binl)
  templ$sum <- rowSums(templ[2:3])
  templ$a.p <- templ$a/templ$sum
  templ$b.p <- templ$b/templ$sum
  return(templ)
}

data.setup_age <- function(dataframe,xname,yname){
  templ <- dataframe
  brks <- seq(from=min(templ[,xname]),to=max(templ[,xname]),by=1)
  templ$binl <- cut(templ[,xname], breaks=brks, labels=brks[1:(length(brks)-1)],right=F)
  templ <- dcast(templ, binl~templ[,yname])
  colnames(templ) <- c("binl","a","b")
  templ$binl <- as.character(templ$binl)
  templ$binl <- as.numeric(templ$binl)
  templ$sum <- rowSums(templ[2:3])
  templ$a.p <- templ$a/templ$sum
  templ$b.p <- templ$b/templ$sum
  return(templ)
}


model.ci <- function(dataframe,xname,yname){
  templ <- dataframe
  glm.out <- glm(templ[,yname] ~ templ[,xname], family=binomial(link=logit), data=templ)
  newdata.l=data.frame(templ[,xname])
  colnames(newdata.l) <- xname
  templ <- predict(glm.out,newdata=newdata.l, type="response", se.fit=T)
  ci <- matrix(c(templ$fit+1.96*templ$se.fit,templ$fit-1.96*templ$se.fit),ncol=2)
  ci <- data.frame(ci)
  ci <- cbind(ci,newdata.l)
  ci <- ci[with(ci,order(ci[,3])),]
  return(ci)
}

est <- function(coef,p) (log(p/(1-p))-coef[1])/coef[2] #p is the proportion

boot.est <- function(dataframe,xname,yname){
  myfun <- function(dataframe,xname,yname){
    templ <- dataframe
    srows <- sample(1:nrow(templ),nrow(templ),TRUE)
    glm.out <- glm(templ[,yname] ~ templ[,xname], family=binomial(link=logit), data=templ[srows,])
    return(est(coef(glm.out),0.5))
  }
  bootdist <- replicate(1000, myfun(dataframe,xname,yname))
  boot.ci <- quantile(unlist(bootdist),c(.025,.975)) # 95% CI
  return(boot.ci)
}


repro_f<-subset(valo_ar, H_Sex=="F")
#repro_f<-subset(repro_f, Phase!="RE")
repro_f <- repro_f[complete.cases(repro_f$Length.cm.),]
#repro_f<-repro_f %>% subset(Month=="Apr"|Month=="May"| Month=="Jun" |Month=="Jul" | Month=="Aug"|Month
lbins.l50vt <- data.setup(repro_f,"Length.cm.","M_VT")
glm.l50vt <- glm(M_VT ~ Length.cm., family=binomial(link=logit), data=repro_f)
l50.civt <- model.ci(repro_f,"Length.cm.","M_VT")

length(repro_f$Length.cm.)
```

```
## [1] 136
```

```
bcL_tlvt<-bootCase(glm.l50vt, R=1000)
lrPerc <- function(cf,p) (log(p/(1-p))-cf[1])/cf[2]#p is the proportion
l50vt<-lrPerc(coef(glm.l50vt), 0.5)
l50vt<-apply(bcL_tlvt, 1, lrPerc, p=0.5)
l50civt<-quantile(l50vt, c(0.25,0.975))
l50vt <- est(coef(glm.l50vt),.5)

#physiological maturity
lbins.l50ca <- data.setup(repro_f,"Length.cm.","M_CA")
glm.l50ca <- glm(M_CA ~ Length.cm., family=binomial(link=logit), data=repro_f)
l50.cica <- model.ci(repro_f,"Length.cm.","M_CA")

length(repro_f$Length.cm.)
```

```
## [1] 136
```

```
bcL_tlca<-bootCase(glm.l50ca, R=1000)
lrPerc <- function(cf,p) (log(p/(1-p))-cf[1])/cf[2]#p is the proportion
l50ca<-lrPerc(coef(glm.l50ca), 0.5)
l50ca<-apply(bcL_tlca, 1, lrPerc, p=0.5)
l50cica<-quantile(l50ca, c(0.25,0.975))
l50ca <- est(coef(glm.l50ca),.5)

######Size at functional maturity####
plot(lbins.l50vt$bin,lbins.l50vt$b.p,pch=19,cex.axis=1.0, cex=1.0,cex.lab=1.0, bty="l",xlab="Fork Lengt
     ,ylim=c(-0.1,1.1),yaxt="n",yaxs="i",xaxs="i", xlim=c(16,40))
axis(2,at=c(seq(0,1.1,by=.1)),labels=c(seq(0,110,by=10)), cex.lab=1.0,cex.axis=1.0, las=1)
curve((exp(coef(glm.l50vt)[1]+coef(glm.l50vt)[2]*x))/(1+exp(coef(glm.l50vt)[1]+coef(glm.l50vt)[2]*x)),a
lines(l50.civt[,3],l50.civt[,1],lty=2)
lines(l50.civt[,3],l50.civt[,2],lty=2)
lines(c(l50vt,l50vt),c(-0.1,.5), lwd=2,col="grey",lty=1, font=2)
lines(c(16,l50vt),c(.5,.5), lwd=2,col="grey",lty=1,font=2)
```



```
#age at maturity
valo_ar_f<-valo_ar %>%
  subset(H_Sex=="F")

valo_ar_f<-valo_ar_f[complete.cases(valo_ar_f$final_age),] #make sure there are no blanks in ages
abins.a50vt <- data.setup_age(valo_ar_f,"final_age","M_VT")
glm.a50vt <- glm(M_VT ~ final_age, family=binomial(link=logit), data=valo_ar_f)
a50.civt <- model.ci(valo_ar_f,"final_age","M_VT")
coef(glm.a50vt)
```

```
## (Intercept)   final_age
##   -2.893769    1.103650
```

```
length(valo_ar_f$final_age)
```

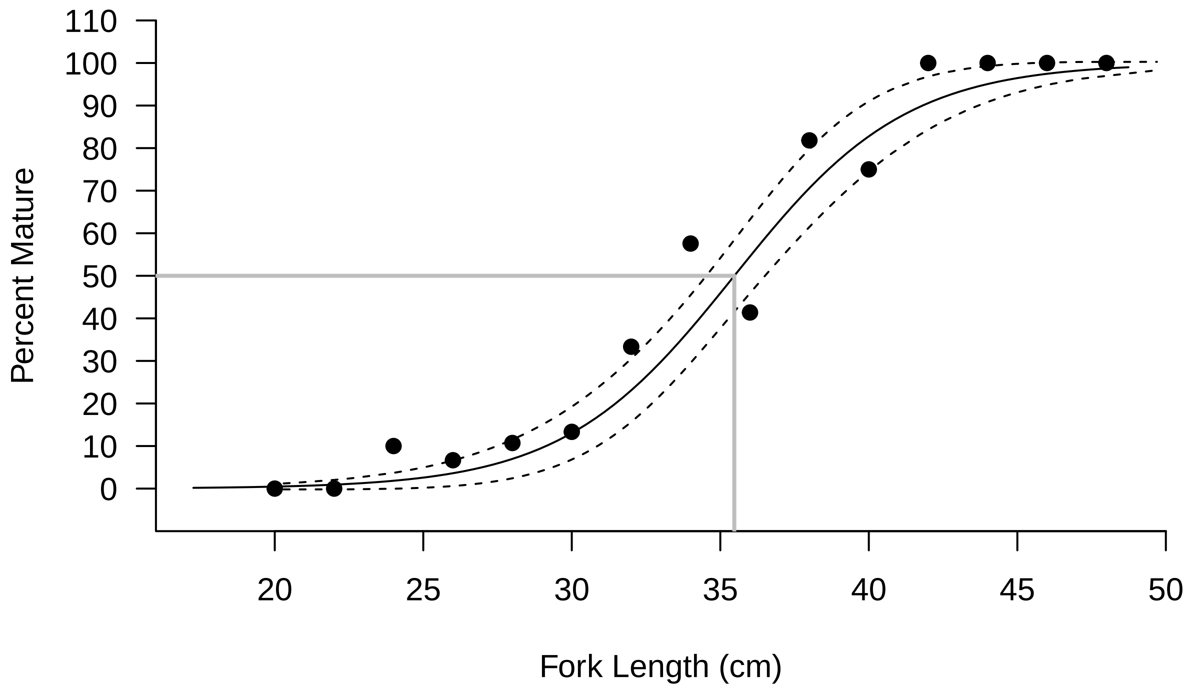
```
## [1] 134
```

```
bcL_tlvt<-bootCase(glm.a50vt, R=1000)
lrPerc <- function(cf,p) (log(p/(1-p))-cf[1])/cf[2]#p is the proportion
a50vt<-lrPerc(coef(glm.a50vt), 0.5)
a50vt<-apply(bcL_tlvt, 1, lrPerc, p=0.5)
a50civt<-quantile(a50vt, c(0.25,0.975))
a50vt <- est(coef(glm.a50vt),.5)
```

```
#physiological
abins.a50ca <- data.setup_age(valo_ar_f,"final_age","M_CA")
glm.a50ca <- glm(M_CA ~ final_age, family=binomial(link=logit), data=valo_ar_f)
a50.cica <- model.ci(valo_ar_f,"final_age","M_CA")

length(valo_ar_f$final_age)
```

```
## [1] 134
```

```
bcL_tlca<-bootCase(glm.a50ca, R=1000)
lrPerc <- function(cf,p) (log(p/(1-p))-cf[1])/cf[2]#p is the proportion
a50ca<-lrPerc(coef(glm.a50ca), 0.5)
a50ca<-apply(bcL_tlca, 1, lrPerc, p=0.5)
a50cica<-quantile(a50ca, c(0.25,0.975))
a50ca <- est(coef(glm.a50ca),.5)



#####Size at sex change######
valo_ar$SexID<-valo_ar$H_Sex
valo_ar$SexID<-revalue(valo_ar$SexID, c("F"=0, "T"=1, "M"=1,"M_PO"=1))
valo_ar$SexID<-as.factor(valo_ar$SexID)
valo_ar<-valo_ar[complete.cases(valo_ar$Length.cm.),]

#remove rows with blanks
valo_ar<-valo_ar[!(is.na(valo_ar$SexID) | valo_ar$SexID==""), ] #removed two rows with blank H_Sex
lbins.l50sex <- data.setup(valo_ar,"Length.cm.","SexID")
glm.l50sex <- glm(SexID ~ Length.cm., family=binomial(link=logit), data=valo_ar)
l50.cisex <- model.ci(valo_ar,"Length.cm.","SexID")

bcL_tlsex<-bootCase(glm.l50sex, R=1000)
lrPerc <- function(cf,p) (log(p/(1-p))-cf[1])/cf[2]#p is the proportion
l50sex<-lrPerc(coef(glm.l50sex), 0.5)
l50sex<-apply(bcL_tlsex, 1, lrPerc, p=0.5)
l50cisex<-quantile(l50sex, c(0.25,0.975))
l50sex <- est(coef(glm.l50sex),.5)

{plot(lbins.l50sex$bin,lbins.l50sex$b.p,pch=19,cex.axis=1.0, cex=1.0,cex.lab=1.0, bty="l",xlab="Fork Le
     ,ylim=c(-0.1,1.1),yaxt="n",yaxs="i",xaxs="i", xlim=c(16,50))
axis(2,at=c(seq(0,1.1,by=.1)),labels=c(seq(0,110,by=10)), cex.lab=1.0,cex.axis=1.0, las=1)
curve((exp(coef(glm.l50sex)[1]+coef(glm.l50sex)[2]*x))/(1+exp(coef(glm.l50sex)[1]+coef(glm.l50sex)[2]*x
lines(l50.cisex[,3],l50.cisex[,1],lty=2)
lines(l50.cisex[,3],l50.cisex[,2],lty=2)
lines(c(l50sex,l50sex),c(-0.1,.5), lwd=2,col="grey",lty=1, font=2)
lines(c(16,l50sex),c(.5,.5), lwd=2,col="grey",lty=1,font=2)}
```

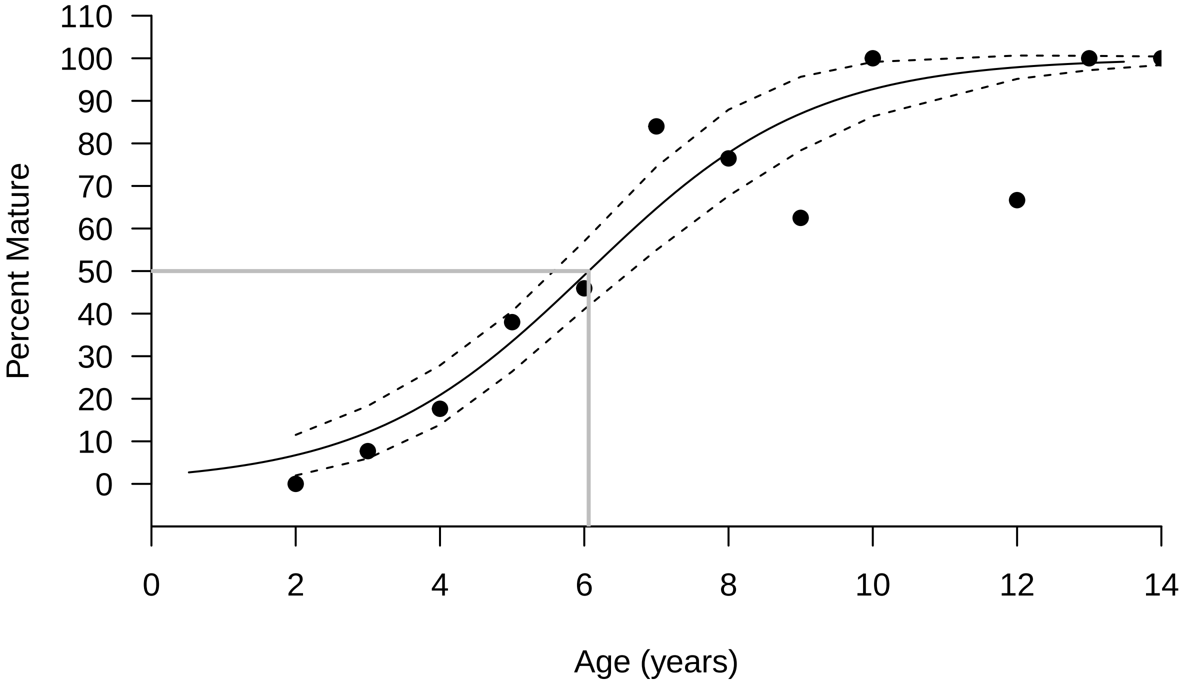


```
#Age at sex change
valo_ar$SexID<-valo_ar$H_Sex
valo_ar$SexID<-revalue(valo_ar$SexID, c("F"=0, "T"=1, "M"=1,"M_PO"=1))
valo_ar$SexID<-as.factor(valo_ar$SexID)
valo_ar<-valo_ar[complete.cases(valo_ar$SexID),]
valo_ar<-valo_ar[complete.cases(valo_ar$final_age),]
valo_ar<-valo_ar[complete.cases(valo_ar$Length.cm.),] ###went from 273 to 257
valo_ar<-valo_ar[!(is.na(valo_ar$SexID) | valo_ar$SexID==""), ] #removed two rows with blank H_Sex

abins.a50sex <- data.setup_age(valo_ar,"final_age","SexID")
abins.a50sex[12,1]<-14 #not sure why it dropped the last age
glm.a50sex <- glm(SexID ~ final_age, family=binomial(link=logit), data=valo_ar)
a50.cisex <- model.ci(valo_ar,"final_age","SexID")

length(valo_ar$Length.cm.)
```

```
## [1] 229
```

```
bcL_tlsex<-bootCase(glm.a50sex, R=1000)
lrPerc <- function(cf,p) (log(p/(1-p))-cf[1])/cf[2]#p is the proportion
a50sex<-lrPerc(coef(glm.a50sex), 0.5)
a50sex<-apply(bcL_tlsex, 1, lrPerc, p=0.5)
a50cisex<-quantile(a50sex, c(0.25,0.975))
a50sex <- est(coef(glm.a50sex),.5)
```

```
#Age sex change
plot(abins.a50sex$bin,abins.a50sex$b.p,pch=19,cex.axis=1.0, cex=1.0,cex.lab=1.0, bty="l",xlab="Age (yea
     ,ylim=c(-0.1,1.1),yaxt="n",yaxs="i",xaxs="i", xlim=c(0,14))
axis(2,at=c(seq(0,1.1,by=.1)),labels=c(seq(0,110,by=10)), cex.lab=1.0,cex.axis=1.0, las=1)
curve((exp(coef(glm.a50sex)[1]+coef(glm.a50sex)[2]*x))/(1+exp(coef(glm.a50sex)[1]+coef(glm.a50sex)[2]*x
lines(a50.cisex[,3],a50.cisex[,1],lty=2)
lines(a50.cisex[,3],a50.cisex[,2],lty=2)
lines(c(a50sex,a50sex),c(-0.1,.5), lwd=2,col="grey",lty=1, font=2)
lines(c(0,a50sex),c(.5,.5), lwd=2,col="grey",lty=1,font=2)
```

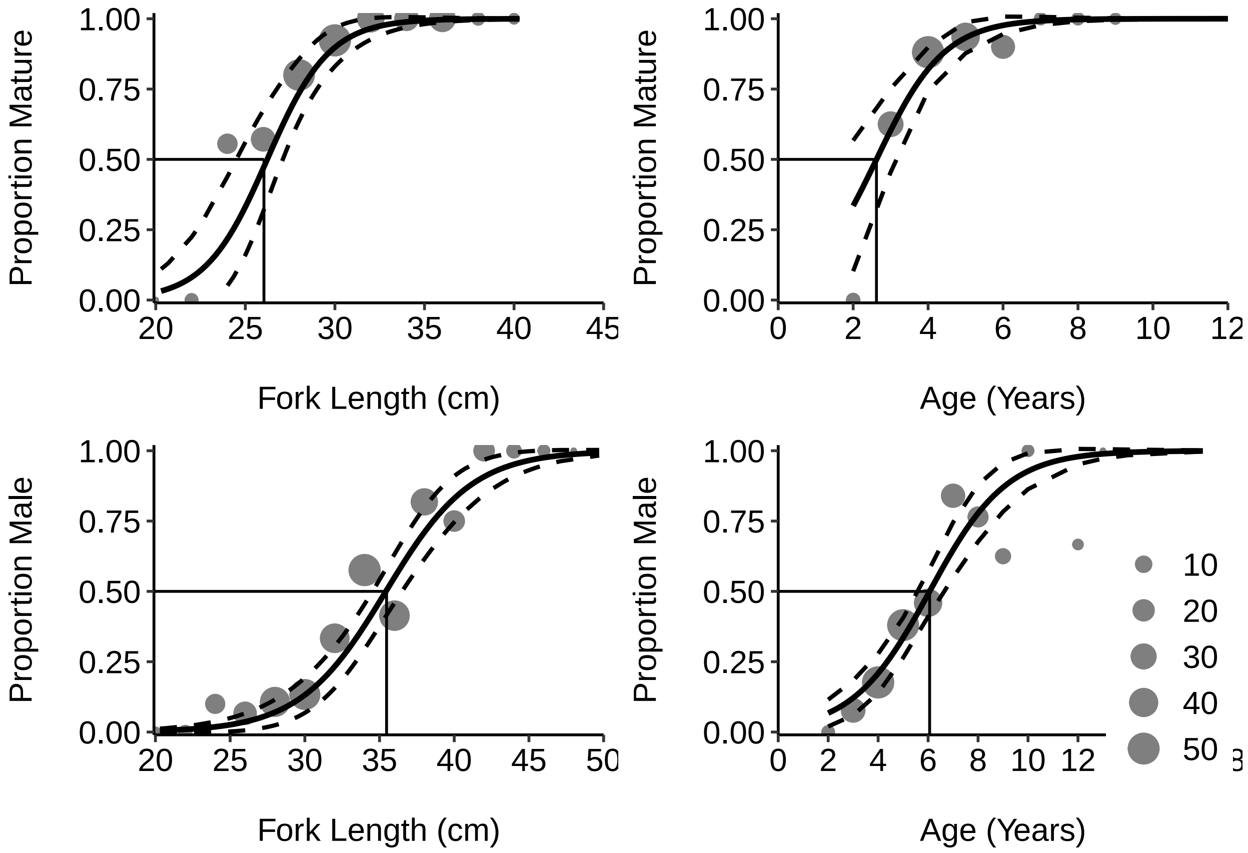


Bubble Plot for Size/Age at Maturity and Size/Age at Sex Chnage

```
#length at maturity
valo_f<-valo_ar %>%
  subset(H_Sex=="F")
l50bubble<- ggplot()+
  geom_smooth(data = valo_f, aes(x = Length.cm., y = M_VT),
              method = "glm", method.args = list(family = "binomial"),
              se = FALSE, color="black") +
  geom_point(data=lbins.l50vt, aes(x = binl, y =b.p, size=sum), alpha=0.5) +
  scale_size(range = c(.5, 5), name="Sample Size")+
  geom_line(aes(x=l50.civt[,3],y=l50.civt[,1], group=1), linetype="dashed", size= 0.75)+
  geom_line(aes(x=l50.civt[,3],y=l50.civt[,2], group=1), linetype="dashed", size= 0.75)+
  geom_segment(aes(x = 19.9, y = 0.5, xend =l50vt , yend = 0.5))+ #to add L50 lines if wanted
  geom_segment(aes(x = l50vt, y = -0.01, xend =l50vt , yend = 0.5))+ #to add L50 lines if wanted
  theme_bw() +
  theme(
    panel.border = element_blank(),
    panel.grid = element_blank(),
    axis.title = element_text(size = 12),
    axis.title.y = element_text(margin= margin(r= 15)), # move title away from axis,
    axis.title.x = element_text(margin= margin(t= 15)),
    legend.position = "none",
   # legend.text = element_text(colour="black", size=10, face="bold"),
    #legend.title = element_text(colour="black", size=10, face="bold"),
    axis.text = element_text(colour = "black", size = 12),
    axis.line = element_line(colour = "black")) +
  xlab('Fork Length (cm)') +
  ylab("Proportion Mature")+
  scale_x_continuous(name = "Fork Length (cm)",breaks=seq(20,45,5))

#l50bubble<-l50bubble+theme(legend.position = c(.87,.2),
#legend.text = element_text(colour="black", size=12),
#legend.title = element_text(colour="black", size=12))

l50bubble<- l50bubble+ scale_y_continuous(expand = c(0, 0), limits = c(-0.01, 1.02))+
  scale_x_continuous(expand = c(0, 0), limits = c(19.9,45), breaks=seq(20,45,5))


#Age at maturity
 a50bubble<- ggplot()+
  geom_smooth(data = valo_f, aes(x = final_age, y = M_VT),
              method = "glm", method.args = list(family = "binomial"),
              se = FALSE, color="black") +
  geom_point(data=abins.a50vt, aes(x = binl, y =b.p, size=sum), alpha=0.5) +
  scale_size(range = c(.5, 5), name="Sample Size")+
  geom_line(aes(x=a50.civt[,3],y=a50.civt[,1], group=1), linetype="dashed", size= 0.75)+
  geom_line(aes(x=a50.civt[,3],y=a50.civt[,2], group=1), linetype="dashed", size= 0.75)+
  geom_segment(aes(x = 0, y = 0.5, xend =a50vt , yend = 0.5))+ #to add L50 lines if wanted
  geom_segment(aes(x = a50vt, y = -0.01, xend =a50vt , yend = 0.5))+ #to add L50 lines if wanted
  theme_bw() +
  theme(
    panel.border = element_blank(),
    panel.grid = element_blank(),
    axis.title = element_text(size = 12),
    axis.title.y = element_text(margin= margin(r= 15)), # move title away from axis,
    axis.title.x = element_text(margin= margin(t= 15)),
    legend.position = "none",
    legend.text = element_blank(),
    legend.title = element_blank(),
    axis.text = element_text(colour = "black", size = 12),
    axis.line = element_line(colour = "black")) +
  xlab('Age (Years)') +
  ylab("Proportion Mature")+
  scale_x_continuous(name = "Age (years)",breaks=seq(0,14,2))
  a50bubble<- a50bubble+ scale_y_continuous(expand = c(0, 0), limits = c(-0.01, 1.02))+
    scale_x_continuous(expand = c(0, 0), limits = c(0, 12), breaks=seq(0,12,2))



##length at sex change
  l50sexbubble<- ggplot()+
    geom_smooth(data = valo_ar, aes(x = Length.cm., y = as.numeric(as.character(SexID))),
                method = "glm", method.args = list(family = "binomial"),
                se = FALSE, color="black") +
    geom_point(data=lbins.l50sex, aes(x = binl, y =b.p, size=sum), alpha=0.5) +
    scale_size(range = c(.5, 5), name="Sample Size")+
    geom_line(aes(x=l50.cisex[,3],y=l50.cisex[,1], group=1), linetype="dashed", size= 0.75)+
    geom_line(aes(x=l50.cisex[,3],y=l50.cisex[,2], group=1), linetype="dashed", size= 0.75)+
    geom_segment(aes(x = 19.9, y = 0.5, xend =l50sex , yend = 0.5))+ #to add L50 lines if wanted
    geom_segment(aes(x = l50sex, y = -0.01, xend =l50sex , yend = 0.5))+ #to add L50 lines if wanted
    theme_bw() +
    theme(
      panel.border = element_blank(),
      panel.grid = element_blank(),
      axis.title = element_text(size = 12),
      axis.title.y = element_text(margin= margin(r= 15)), # move title away from axis,
      axis.title.x = element_text(margin= margin(t= 15)),
      legend.position = "none",
      legend.text = element_blank(),
      legend.title = element_blank(),
      axis.text = element_text(colour = "black", size = 12),
      axis.line = element_line(colour = "black")) +
    xlab('Fork Length (cm)') +
    ylab("Proportion Male")+
    scale_x_continuous(name = "Fork Length (cm)",breaks=seq(20,50,5))

  l50sexbubble<- l50sexbubble+ scale_y_continuous(expand = c(0, 0), limits = c(-0.01, 1.02), breaks=seq
    scale_x_continuous(expand = c(0, 0), limits = c(19.9,50), breaks=seq(20,50,5))


#age at sex chage
  valo_ar <- valo_ar[!(valo_ar$SexID == ""), ]
  valo_ar$SexID[valo_ar$SexID==' ']=NA
  valo_ar$SexID= droplevels(valo_ar$SexID)
  a50sexbubble<- ggplot()+
    geom_smooth(data = valo_ar, aes(x = final_age, y = as.numeric(as.character(SexID))),
                method = "glm", method.args = list(family = "binomial"),
                se = FALSE, color="black") +
       geom_point(data=abins.a50sex, aes(x = binl, y =b.p, size=sum), alpha=0.5) +
    scale_size(range = c(.5, 5), name=" ")+
    geom_line(aes(x=a50.cisex[,3],y=a50.cisex[,1], group=1), linetype="dashed", size= 0.75)+
    geom_line(aes(x=a50.cisex[,3],y=a50.cisex[,2], group=1), linetype="dashed", size= 0.75)+
    geom_segment(aes(x = 0, y = 0.5, xend =a50sex , yend = 0.5))+ #to add L50 lines if wanted
    geom_segment(aes(x = a50sex, y = -0.01, xend =a50sex , yend = 0.5))+ #to add L50 lines if wanted
    theme_bw() +
    theme(
      panel.border = element_blank(),
      panel.grid = element_blank(),
      axis.title = element_text(size = 12),
      axis.title.y = element_text(margin= margin(r= 15)), # move title away from axis,
      axis.title.x = element_text(margin= margin(t= 15)),
      legend.position = "none",
      legend.text = element_blank(),
      legend.title = element_blank(),
      axis.text = element_text(colour = "black", size = 12),
      axis.line = element_line(colour = "black")) +
    xlab('Age (Years)') +
    ylab("Proportion Male")+
    scale_x_continuous(name = "Age (years)",breaks=seq(0,18,2))


  a50sexbubble<-a50sexbubble +
  scale_y_continuous(expand = c(0, 0), limits = c(-0.01, 1.02))+
    scale_x_continuous(expand = c(0, 0), limits = c(0, 18),breaks=seq(0,18,2))


  scale_y_continuous( limits = c(0.00, 1))
```

```
## <ScaleContinuousPosition>
##  Range:
##  Limits:    0 --    1
```

```
#l50bubble<-l50bubble+theme(legend.position = c(.8,.25),
#                            legend.text = element_text(colour="black", size=10),
#                            legend.title = element_text(colour="black", size=10))

a50sexbubble<- a50sexbubble+theme(legend.position = c(.87,.35),
                              legend.text = element_text(colour="black", size=12),
                              legend.title = element_text(colour="black", size=12))



figure <- ggpubr::ggarrange(l50bubble, a50bubble,l50sexbubble,  a50sexbubble,
                    labels = c("", "", "",""),
                    ncol = 2, nrow = 2,
                    hjust=-8.2,
                    vjust=0.6,
                    font.label = list(size = 12, color = "black", face = "plain", family = NULL))

figure
```

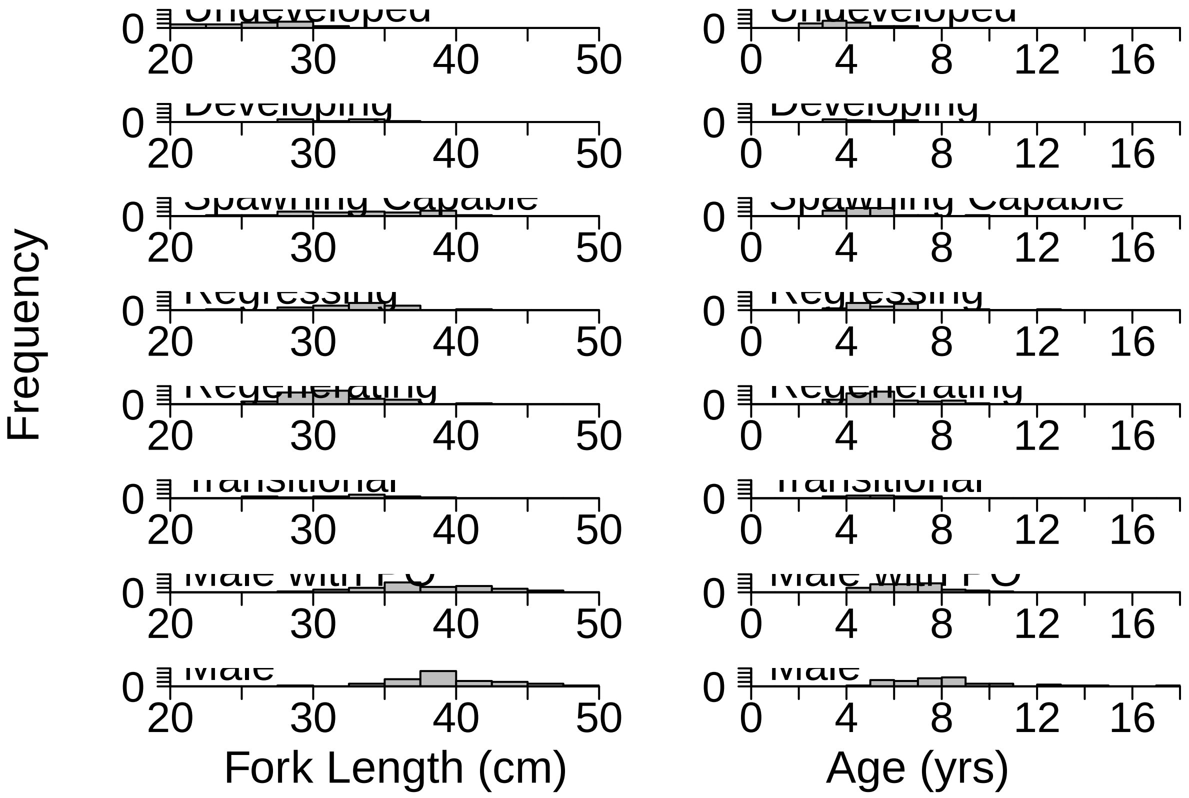


Length/Age Frequency Distribution

```
###age freq distribution#########
valo_f_u<-valo_ar %>%
  subset(H_Sex=="F" & Phase=="U")
valo_f_d<-valo_ar %>%
  subset(H_Sex=="F" & Phase=="D")
valo_f_sc<-valo_ar %>%
  subset(H_Sex=="F" & Phase=="SC")
valo_f_rg<-valo_ar %>%
  subset(H_Sex=="F" & Phase=="RG")
valo_f_re<-valo_ar %>%
  subset(H_Sex=="F" & Phase=="RE")
valo_t<-valo_ar %>%
  subset(H_Sex=="T")
valo_t_po<-valo_ar %>%
  subset(H_Sex=="M_PO")
valo_m<-valo_ar %>%
  subset(H_Sex=="M")

#Plot it. I save it width=2900, height=3800, res=250 t see all the details in the figure that you cant
{par(mfcol=c(8,2), mar=c(1,5,2,1), oma=c(4,3,.4,.4))
hist(valo_f_u$Length.cm., right=FALSE,main="" ,xaxs="i",yaxs="i", breaks=seq(20,50,2.5), ylab="", xlab=
axis(1,at= seq(20,50,by=5) , labels = seq(20,50,by=5) ,cex=2, cex.lab=2, cex.axis=2)
```

```r
box(bty = "l")
text(20,18.5, "Undeveloped",cex=2, cex.lab=2, cex.axis=2, pos = 4)
hist(valo_f_d$Length.cm., right=FALSE,main="" ,xaxs="i",yaxs="i",breaks=seq(20,50,2.5), ylab="",xlab=""
axis(1,at= seq(20,50,by=5) , labels = seq(20,50,by=5),cex=2, cex.lab=2, cex.axis=2 )
box(bty = "l")
text(20,18.5, "Developing",cex=2, cex.lab=2, cex.axis=2, pos = 4)
hist(valo_f_sc$Length.cm., right=FALSE,main="" ,xaxs="i",yaxs="i",breaks=seq(20,50,2.5), ylab="",xlab="
axis(1,at= seq(20,50,by=5) , labels = seq(20,50,by=5),cex=2, cex.lab=2, cex.axis=2  )
text(20,18.5, "Spawning Capable",cex=2, cex.lab=2, cex.axis=2, pos = 4)
box(bty = "l")
hist(valo_f_rg$Length.cm.,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(20,50,2.5), ylab="", xlab=
axis(1,at= seq(20,50,by=5) , labels = seq(20,50,by=5),cex=2, cex.lab=2, cex.axis=2  )
text(20,18.5, "Regressing",cex=2, cex.lab=2, cex.axis=2, pos = 4)
box(bty = "l")
hist(valo_f_re$Length.cm.,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(20,50,2.5), ylab="", xlab=
axis(1,at= seq(20,50,by=5) , labels = seq(20,50,by=5),cex=2, cex.lab=2, cex.axis=2  )
text(20,18.5, "Regenerating",cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
hist(valo_t$Length.cm.,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(20,50,2.5), ylab="", xlab=""
axis(1,at= seq(20,50,by=5) , labels = seq(20,50,by=5) ,cex=2, cex.lab=2, cex.axis=2 )
text(20,18.5, "Transitional",cex=2, cex.lab=2, cex.axis=2, pos = 4)
box(bty = "l")
hist(valo_t_po$Length.cm.,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(20,50,2.5), ylab="", xlab=
axis(1,at= seq(20,50,by=5) , labels = seq(20,50,by=5),cex=2, cex.lab=2, cex.axis=2  )
text(20,18.5, "Male with PO",cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
hist(valo_m$Length.cm.,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(20,50,2.5), ylab="", xlab="Fo
axis(1,at= seq(20,50,by=5) , labels = seq(20,50,by=5), cex=2, cex.lab=2, cex.axis=2  )
text(20,18.5, "Male", cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
mtext(text="Fork Length (cm)",side=1,line=2,adj=0.22,outer=TRUE, cex=1.4)
mtext(text="Frequency",side=2,line=0,outer=TRUE, cex=1.4)

#age
hist(valo_f_u$final_age, right=FALSE, main="" ,xaxs="i",yaxs="i", breaks=seq(0,18, 1), ylab="", xlab=""
axis(1,at= seq(0,18,by=2) , labels = seq(0,18,by=2) ,cex=2, cex.lab=2, cex.axis=2)
text(0.2,18.5, "Undeveloped",cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
hist(valo_f_d$final_age,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(0,18, 1), ylab="",xlab="" ,y
axis(1,at= seq(0,18,by=2) , labels = seq(0,18,by=2),cex=2, cex.lab=2, cex.axis=2 )
text(0.2,18.5, "Developing",cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
hist(valo_f_sc$final_age, right=FALSE,main="" ,xaxs="i",yaxs="i",breaks=seq(0,18, 1), ylab="",xlab="" ,
axis(1,at= seq(0,18,by=2) , labels = seq(0,18,by=2),cex=2, cex.lab=2, cex.axis=2  )
text(0.2,18.5, "Spawning Capable",cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
hist(valo_f_rg$final_age,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(0,18, 1), ylab="", xlab=""
axis(1,at= seq(0,18,by=2) , labels = seq(0,18,by=2),cex=2, cex.lab=2, cex.axis=2  )
text(0.2,18.5, "Regressing",cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
hist(valo_f_re$final_age,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(0,18, 1), ylab="", xlab=""
axis(1,at= seq(0,18,by=2) , labels = seq(0,18,by=2),cex=2, cex.lab=2, cex.axis=2  )
text(0.2,18.5, "Regenerating",cex=2, cex.lab=2, cex.axis=2,pos = 4)
```

```
box(bty = "l")
hist(valo_t$final_age,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(0,18, 1), ylab="", xlab="" ,yl
axis(1,at= seq(0,18,by=2) , labels = seq(0,18,by=2) ,cex=2, cex.lab=2, cex.axis=2 )
text(0.2,18.5, "Transitional",cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
hist(valo_t_po$final_age,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(0,18, 1), ylab="", xlab=""
axis(1,at= seq(0,18,by=2) , labels = seq(0,18,by=2),cex=2, cex.lab=2, cex.axis=2  )
text(0.2,18.5, "Male with PO",cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
hist(valo_m$final_age,right=FALSE, main="" ,xaxs="i",yaxs="i",breaks=seq(0,18,by=1), ylab="", xlab="Age
axis(1,at= seq(0,18,by=2) , labels = seq(0,18,by=2), cex=2, cex.lab=2, cex.axis=2  )
text(0.2,18.5, "Male", cex=2, cex.lab=2, cex.axis=2,pos = 4)
box(bty = "l")
mtext(text="Age (yrs)",side=1,line=2, adj=.8,outer=TRUE, cex=1.4)}
```



Male and female size and age differences

```
males<-rbind(valo_m, valo_t_po)
ks.test(valo_ar_f$Length.cm.,males$Length.cm.,   alternative = "greater")
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
```

```
## data:  valo_ar_f$Length.cm. and males$Length.cm.
## D^+ = 0.64188, p-value < 2.2e-16
## alternative hypothesis: the CDF of x lies above that of y
```

```
ks.test(valo_ar_f$final_age,males$final_age,   alternative = "greater")
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  valo_ar_f$final_age and males$final_age
## D^+ = 0.49613, p-value = 1.101e-11
## alternative hypothesis: the CDF of x lies above that of y
```

```
t.test(valo_ar_f$Length.cm., males$Length.cm.)
```

```
##
##  Welch Two Sample t-test
##
## data:  valo_ar_f$Length.cm. and males$Length.cm.
## t = -12.563, df = 179.35, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.488346 -6.183763
## sample estimates:
## mean of x mean of y
##  31.15672  38.49277
```

```
t.test(valo_ar_f$final_age, males$final_age)
```

```
##
##  Welch Two Sample t-test
##
## data:  valo_ar_f$final_age and males$final_age
## t = -8.2563, df = 130.34, p-value = 1.457e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.949576 -1.809281
## sample estimates:
## mean of x mean of y
##  4.656716  7.036145
```

```
max(males$final_age, na.rm=TRUE)
```

```
## [1] 17
```

```
mean(males$final_age, na.rm=TRUE)
```

```
## [1] 7.036145
```

```
mean(valo_ar_f$final_age, na.rm=TRUE)
```

```
## [1] 4.656716
```