1. Plot

plot(recruits~spawners,data=yellowfin2,

xlab=c("Spawners"),

ylab=c("Recruits"),

xlim=c(0,max(spawners)),

ylim=c(0,max(recruits)),pch=16,col=4)

title("Yellow Fin Spawner-Recruit Relationship")

2. Line from a linear model

lm1 = lm(recruits~spawners,data=yellowfin2)

summary(lm1)

abline(coef(lm1),col=2,lwd=2)

3. Line from a non linear model

bh.nls = nls(recruits~alpha\*spawners/(1+beta\*spawners),

data=yellowfin2,start=list(alpha=1,beta=0.00001))

summary(bh.nls)

bh.new = list(spawners=seq(0,30))

lines(bh.new$spawners,predict(bh.nls,newdata=bh.new),col=3,lwd=2)

4. Access coeficients of a regression

a = coef(bh.lm)[1]

b = coef(bh.lm)[2]

5. calculating p\_values

beta.pvalue = 2\*(1-pt(beta.tvalue,23))

6. Plotting legend inside a graph

b = coef(head.e.lm)[1]

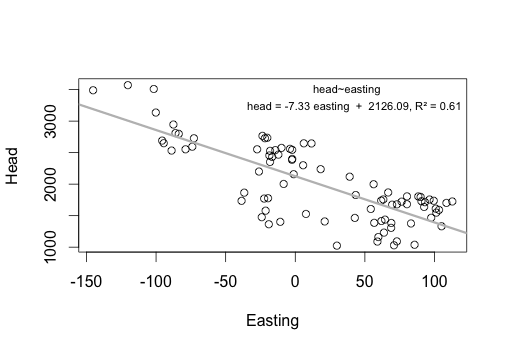
a = coef(head.e.lm)[2]

r2 = summary(head.e.lm)$r.squared

if( b<0) {my\_sign = ' - '; b = -b} else { my\_sign= ' + '}

legend('topright', title='head~easting', legend=sprintf("head = %3.2f easting %s %3.2f, R\UB2 = %3.2f", a, my\_sign, b, r2), bty='n', cex=0.7)

Look like this.



Example 2 of a Legend

legend("topright",paste("Area:",c("Smoking","Nonsmoking")),lwd=2,col=c('red','blue'),bty="n",cex = 0.75)

7. Making a vector of n parameters

B = rep(0,n)

8. multiple plots in a graph

par(mfrow=c(2,2))

plot(x1,y)

plot(x2,y)

plot(x3,y)

plot(x4,y)

par(mfrow=c(1,1))

9. Dropping rows in your dataset

pollock[-c(22, 23), ] #dropping last two rows

10. Manual data entry

sparrow = data.frame(

age.days=c(3,4,5,6,8,9,10,11,12,14,15,16,17),

wing.length=

c(1.4,1.5,2.2,2.4,3.1,3.2,3.2,

3.9,4.1,4.7,4.5,5.2,5.0))

11. Plotting smoother between residual and variables

Look at the residuals from your best model (thus far) relative to number of beds. Perhaps put a smooth curve through the residuals of this plot to guide the eye. Something like:

plot(APPENC01$beds,resid(senic1.lm))

lines(lowess(APPENC01$beds,resid(senic1.lm)))

12. Plotting correlations and pairs

the numbers are the column # of the variables.

round(cor(APPENC01[,c(7,10,11,12)]),2)

pairs(APPENC01[,c(7,10,11,12)])

13. Confidence intervals

yh + c(-1, 1) \*2\* Yh.se

[1] 2.458987 2.750408

14. Title smaller font

title("Bearing Sea Pollock Spawner-Recruit Relationship", cex.main = 0.9, font = 2)

Sensitivity Analysis : Holding one variable constant and varying the other.

#Fixing beta and varying alpha

#boot.coef has a list of pairs of alpha and beta

b = boot.coef[1,2]

S = seq(0,350)

for(i in seq(1000))

{

a = boot.coef[i,1]

lines(S,a\*S\*(exp(b\*S)),col=2,lwd=2)

}

15. Make a new column - variable

AlcUse = rep(1,dim(Surgical)[1])

AlcUse[Surgical$AlcUse.Mod==1] = 2

AlcUse[Surgical$AlcUse.Hvy==1] = 3

16. Doing the super smoother line

plot(log(length)~age,data=SpinnerShark,

xlab="Age",ylab="log(Length)")

lines(supsmu(SpinnerShark$age,log(SpinnerShark$length)),

lwd=2,col=2)

17. Creating a new variable in a column

logistic <- read.delim("~/Desktop/SPRING2015/BioStats II/Lab8/logistic.txt")

logistic$p = logistic$dead / logistic$n //p is the new variable

View(logistic)

18. renaming the variable name in a dataset prognostic

library(plyr)

prognostic <-rename(prognostic, c("patient"="days", "days..index"="index"))

19. Making a nonlinear objective function

# Step2: make you own objective function

my.regress = function(param,data=list(x=enzyme$X,y=enzyme$Y)){

b0 = param[1]

b1 = param[2]

x = data$x

y = data$y

y.pred = (b0 \* x)/(b1 + x)

RSS = sum((y-y.pred)^2)

nn = length(x)

conc = (nn/2)\*log(2\*pi\*RSS/nn)+nn/2

}

my.regress.optim = optim(c(0,1),my.regress,hessian=T)

cbind(my.regress.optim$par,

sqrt(diag(solve(my.regress.optim$hessian))))

Subset Data set to get only Genotype A and not the others

Genotype P Height

1 A 0.0 11.91119

2 A 0.1 16.46677

3 A 0.2 21.73446

4 A 0.3 26.23806

5 A 0.5 29.95274

6 A 0.9 43.57491

# using subset function

newdata <- subset(phosphorus, Genotype =="A" ,

select=c(P, Height))

to get:

P Height

1 0.0 11.91119

2 0.1 16.46677

3 0.2 21.73446

4 0.3 26.23806

5 0.5 29.95274

6 0.9 43.57491

7 1.5 60.66681

8 2.0 68.57008

9 3.0 80.48361

10 5.0 93.47072

20. Abline intersecting X-AXIS- verticle line at x=5.

abline(v=5,lwd=3)

Y-Axis abline(h=5,lwd=3)

21. Making an array

M = array(0, dim=c(14, 5))

Creating a data base and plotting values to show variations

vessel = data.frame(

fishermen.ID=c(1,2,3,4,5,6,7,8,9,10),

revenue=

c(25000,15000,10000, 6000,50000, 40000,10000, 14000,7000, 30000))

plot(vessel$fishermen.ID, vessel$revenue,

xlab=c("Fishermen ID"),

ylab=c("Annual Revenue"),

col=ifelse(vessel$revenue < mean(vessel$revenue), "red", "black"),

pch=ifelse(vessel$revenue < mean(vessel$revenue), 19, 1))

abline(h=mean(vessel$revenue))

cbind(date,time)

test <- data.frame(cbind(date,time))

Manipulating Dates

bDates <- as.Date(date,

format = "%m/%d/%Y")

bDates <- as.Date(bouy$Date,

format = "%m/%d/%Y")

bouy$Date<-strftime(strptime(bouy$Date,"%d/%m/%Y"),"%d/%m/%y")

Date conversion //this one works!

bouy$Date <- format(as.Date(bouy$Date,format = "%d/%m/%Y"), "%d/%m/%y")