Regression analysis and non-linear model fit of periphyton inorganic carbon content vs. water depth and TP, across regions

Kristin Briggs

4/23/2020

data set up

main= read.csv(file.choose())

tag = mainï$..TAG_IDdepth = main$ WATER_DEPTH_CM region = main$REGIONafdm = main$ PERI_AFDM_G_PER_M2 orgp = main$PERI_PROP_ORGANICinorgp = 1 - main$ PERI_PROP_ORGANIC temp = main$SURFACE_WATER_TEMP_Cph = main$ SURFACE_WATER_PH cond = main$SURFACE_WATER_COND_US_PER_CMcond = as.numeric(factor(am$ cond)) soil = main$SOIL_DEPTH_CMsoil = as.numeric(factor(am$ soil)) tp = main`PERI_TP_UG_PER_G_DRY ash = main` PERI_DRY_G_PER_M2 - main$PERI_AFDM_G_PER_M2 ashl = log(1+ash) am = data.frame(tag,region,depth,afdm,orgp,inorgp,temp,ph,cond,soil,tp, ash,ashl,ash1) ash1=(ash+1)

plots, visualizing relationships and data distributions

plot(am$depth, am$ ash) plot(am$depth + am$ tp, am$ash)

plot(am$depth, am$ ashl) plot(am$depth + am$ tp, am$ashl)

plot(am$depth, am$ ashl)

plot(an$depth, log(an$ afdm), ylab="AFDM", xlab="water depth") #log transformed y

par(mfrow=c(1,1))

library(ggplot2) ggplot(am, aes(x = depth, y = ash, color = region)) + geom_point() + scale_y_continuous(trans = 'log10')

ggplot(am, aes(x = depth+tp, y = ash, color = region)) + geom_point()

ggplot(am, aes(x = tp, y = ash, color = region)) + geom_point() ## linear models, looking for explanatory variables that fit

m1 = lm(am$ash~am$ depth) m2 = lm(am$ash~am$ depth + am$tp)m3 = lm(am$ ash~am$depth + am$ region) m4 = lm(am$ash~am$ depth +am$soil)m5 = lm(am$ ash~am$depth + am$ cond) m6 = lm(sqrt(am$ash)~am$ depth + am$tp)

summary(m1) summary(m2) summary(lm1) summary(lm2)

linear models with log transformed response variable (ash weight)

lm1 = lm(am$ashl~am$ depth) lm2 = lm(am$ashl~am$ depth + am$tp)lm3 = lm(am$ ashl~am$depth + am$ region) lm4 = lm(am$ashl~am$ depth +am$soil)

plotting residuals to evaluate assumptions, shapiro tests for normality

library(ggplot2) par(mfrow=c(2,2)) par(mfrow=c(1,1))

plot(m1) plot(m2) plot(m3)

plot(lm1) plot(lm2) plot(lm3) plot(ashl)

shapiro.test(ashl) shapiro.test(ash) shapiro.test(depth) shapiro.test(tp)

testing model fit with ANOVA

anova(m1) aov(m2) aov(m3) summary(aov(lm1)) summary(aov(lm2)) summary(aov(lm3))

vif(m3) anova(m1,m2) anova(m1,m5) anova(lm1,lm2) anova(lm1,lm3)

correlation coefficients

cor(depth,ash) cor(depth+tp,ash) cor(tp,ash) cor(cond,ash)

cor(depth,ashl) cor(tp,ashl) cor(depth+tp,ashl) cor(cond,ashl)

colinearity test

cor (depth, tp) summary(lm(depth~tp))

vif(m3) vif(lm2) vif(m2) vif(lm3)

evaluating region and water depth. There is an effect, to keep in mind

par(mfrow=c(2,2)) plot(am$region, am$ depth,ylab="water level", xlab="region") summary(aov(am$depth\ am$ region)) m1rg= lm(am$depth\ am$ region)

library(ggplot2)

plot(m1rg) shapiro.test(am$depth)

attempting to fit Weibull distribution

plot(pweibull(ash, 3, scale = 1000, log = FALSE)) ### attempting to use Weibull distribution to model the ash weight against depth and TP. visually a weibull distribution seems reasonable plot(SSweibull(am$depth, 3000, 2900, -5, 1.5))

plot(SSweibull(am`depth, 2500, 2900, -5, 1.5) ) local({ Asym <- 3000; Drop <- 2900; lrc <- -5; pwr <- 1.5 SSweibull(am` depth, Asym, Drop, lrc, pwr) # response *and* gradient }) getInitial(am$depth\ SSweibull(am$ depth, Asym, Drop, lrc, pwr), data = am) y=am$ashsummary(aov(am$ depth~am$region))

nls models

Initial values are in fact the converged values

fm1 <- nls(y ~ SSweibull(am$depth, 3000, 2900, -5, 1.5), data = am) summary(fm1)

y = nls(am$ash SSweibull(am$ depth, 3250,2250, -5.5, 2)) y = nls(am$ash SSweibull(am$ depth, 2250,250,-5.5,2))

bootstrap

how to “nest” this by region?

I’m not sure how to generate the t statistics that make sense.. this doesn’t seem to work. Mainly I wanted to bootstrap by region to see how different the conditions were between regions, since the shape of the data seemed to differ between regions. But I don’t think my approach made sense.

ash_estimate = function(data, indices){ d = data[indices,] dlm= lm(am$depth am$ region, data = am) dlms = summary(dlm)$r. square plm = lm(am$ ash~am$region, data = am) plms = summary(plm)$ r.square clm = lm(am$ash am$ depth+am$tp, data = am) clms = summary(clm)$ r.square rel = c(dlms,plms,clms) return(rel) } library(boot)

results = boot(data = am, statistic = ash_estimate, R = 1000) print(results) plot(results, index = 1)

confidence_interval_H = boot.ci(results, index = 1, conf = 0.95) print(confidence_interval_H)