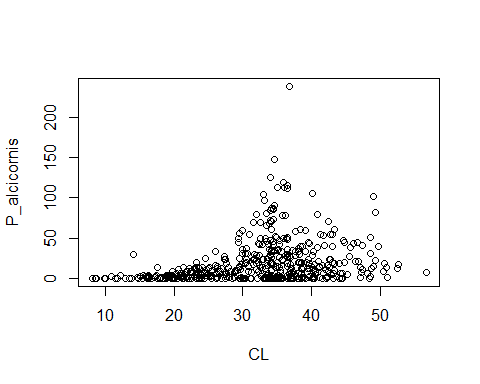
Lab-6

Kent Codding

2023-10-24

# Read in data

setwd("C:/Users/Kent Codding/Desktop/Adv Biostat/Lab 6")  
df <- read.csv("wing\_worm\_data.csv")  
df <- cbind(P\_alcicornis = df$P\_alcicornis, CL = df$CL)  
df <- data.frame(df)  
  
plot(P\_alcicornis ~ CL, data = df)



# Load dependencies

library(MASS)  
library(gam)

## Warning: package 'gam' was built under R version 4.1.3

## Loading required package: splines

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 4.1.3

## Loaded gam 1.22-2

library(car)

## Warning: package 'car' was built under R version 4.1.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.1.3

# Step 1

mod.p <- glm(P\_alcicornis ~  
 CL,   
 data = df, family = "poisson")  
summary(mod.p)

##   
## Call:  
## glm(formula = P\_alcicornis ~ CL, family = "poisson", data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -8.690 -4.464 -2.271 1.358 26.485   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.395756 0.045619 30.60 <2e-16 \*\*\*  
## CL 0.046182 0.001242 37.19 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 13229 on 467 degrees of freedom  
## Residual deviance: 11775 on 466 degrees of freedom  
## AIC: 13428  
##   
## Number of Fisher Scoring iterations: 6

residual deviance divided by degrees of freedom is far greater than 1, so negative binomial may be better

mod.nb <- glm.nb(df$P\_alcicornis ~ df$CL)  
summary(mod.nb)

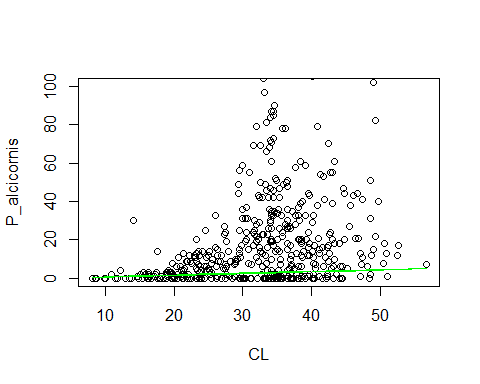
##   
## Call:  
## glm.nb(formula = df$P\_alcicornis ~ df$CL, init.theta = 0.5376400303,   
## link = log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2847 -1.2673 -0.3662 0.2353 2.6084   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.035585 0.243038 -0.146 0.884   
## df$CL 0.088195 0.007207 12.237 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(0.5376) family taken to be 1)  
##   
## Null deviance: 623.63 on 467 degrees of freedom  
## Residual deviance: 544.79 on 466 degrees of freedom  
## AIC: 3513.4  
##   
## Number of Fisher Scoring iterations: 1  
##   
##   
## Theta: 0.5376   
## Std. Err.: 0.0369   
##   
## 2 x log-likelihood: -3507.3720

residual deviance divided by degrees of freedom much closer to 1, so negative binomial is a better choice.

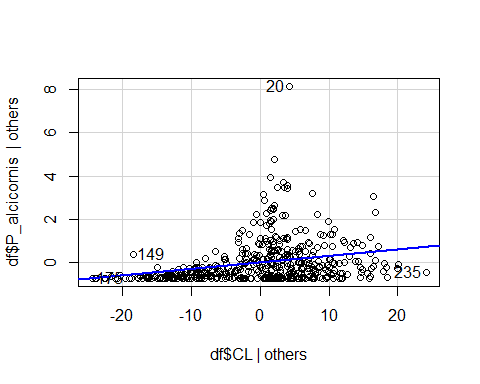
# Step 2

## Plot of raw data

#show best GLM (negative binomial) over raw data by subsetting 3 prediction lines over the range of CL  
df$P\_alcicornis\_pred <- predict(mod.nb, df)  
  
plot(P\_alcicornis ~ CL,   
 data = df,  
 ylim = c(0,100))  
lines(P\_alcicornis\_pred ~ CL, data = subset(df, CL < 20,  
 lwd = 2), col = "green")  
lines(P\_alcicornis\_pred ~ CL, data = subset(df, (CL > 20 & CL < 40),  
 lwd = 2), col = "green")  
lines(P\_alcicornis\_pred ~ CL, data = subset(df, (CL > 40),  
 lwd = 2), col = "green")

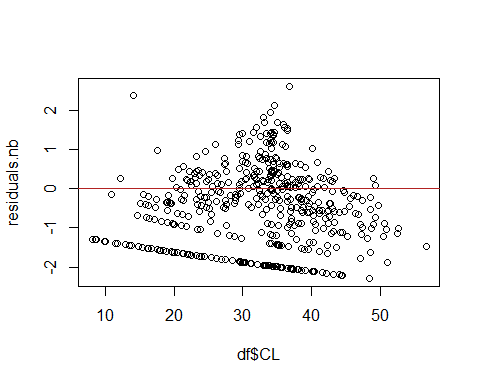


avPlots(mod.nb) #show best GLM (negative binomial) over raw data



## plot of GLM residuals

residuals.nb <- residuals.glm(mod.nb) # create object for deviance residuals  
plot(residuals.nb ~ df$CL)  
abline(h = 0, col = "firebrick")



Yes, the residuals show some patterns, and a non-shotgun blast looking plot of the deviance residuals suggests that the data violate the assumption of nonindependence. Thus, we may want to use a nonlinear model for these data.

# Step 3

## fit 3 negative binomial GAMS

mod.gam1 <- gam::gam(df$P\_alcicornis ~ lo(df$CL, span = .1), family = "gaussian")  
summary(mod.gam1)

##   
## Call: gam::gam(formula = df$P\_alcicornis ~ lo(df$CL, span = 0.1), family = "gaussian")  
## Deviance Residuals:  
## Min 1Q Median 3Q Max   
## -41.557 -14.366 -2.402 5.313 203.677   
##   
## (Dispersion Parameter for gaussian family taken to be 580.2135)  
##   
## Null Deviance: 330768.5 on 467 degrees of freedom  
## Residual Deviance: 259565.2 on 447.3614 degrees of freedom  
## AIC: 4328.365   
##   
## Number of Local Scoring Iterations: NA   
##   
## Anova for Parametric Effects  
## Df Sum Sq Mean Sq F value Pr(>F)   
## lo(df$CL, span = 0.1) 1.00 27022 27022.0 46.572 2.882e-11 \*\*\*  
## Residuals 447.36 259565 580.2   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Anova for Nonparametric Effects  
## Npar Df Npar F Pr(F)   
## (Intercept)   
## lo(df$CL, span = 0.1) 18.6 4.0854 3.839e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#NOTE: "nb" and negative.binomial(theta = 2) did not work and yielded odd results for predicted values respectively, so I stuck with family = gaussian

mod.gam2 <- gam::gam(df$P\_alcicornis ~ lo(df$CL, span = .5), family = "gaussian")   
summary(mod.gam2)

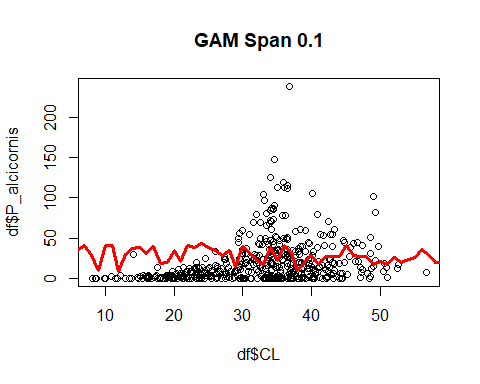
##   
## Call: gam::gam(formula = df$P\_alcicornis ~ lo(df$CL, span = 0.5), family = "gaussian")  
## Deviance Residuals:  
## Min 1Q Median 3Q Max   
## -30.867 -14.234 -3.232 4.541 210.219   
##   
## (Dispersion Parameter for gaussian family taken to be 598.2801)  
##   
## Null Deviance: 330768.5 on 467 degrees of freedom  
## Residual Deviance: 276860.1 on 462.76 degrees of freedom  
## AIC: 4327.757   
##   
## Number of Local Scoring Iterations: NA   
##   
## Anova for Parametric Effects  
## Df Sum Sq Mean Sq F value Pr(>F)   
## lo(df$CL, span = 0.5) 1.00 27022 27022.0 45.166 5.337e-11 \*\*\*  
## Residuals 462.76 276860 598.3   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Anova for Nonparametric Effects  
## Npar Df Npar F Pr(F)   
## (Intercept)   
## lo(df$CL, span = 0.5) 3.2 13.87 3.631e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

mod.gam3 <- gam::gam(df$P\_alcicornis ~ lo(df$CL, span = 1), family = "gaussian")   
summary(mod.gam3)

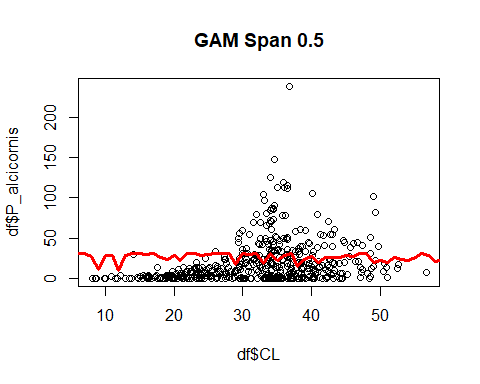
##   
## Call: gam::gam(formula = df$P\_alcicornis ~ lo(df$CL, span = 1), family = "gaussian")  
## Deviance Residuals:  
## Min 1Q Median 3Q Max   
## -28.940 -14.606 -5.337 5.410 213.783   
##   
## (Dispersion Parameter for gaussian family taken to be 636.3566)  
##   
## Null Deviance: 330768.5 on 467 degrees of freedom  
## Residual Deviance: 296296.6 on 465.614 degrees of freedom  
## AIC: 4353.802   
##   
## Number of Local Scoring Iterations: NA   
##   
## Anova for Parametric Effects  
## Df Sum Sq Mean Sq F value Pr(>F)   
## lo(df$CL, span = 1) 1.00 27022 27022.0 42.464 1.874e-10 \*\*\*  
## Residuals 465.61 296297 636.4   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Anova for Nonparametric Effects  
## Npar Df Npar F Pr(F)   
## (Intercept)   
## lo(df$CL, span = 1) 0.4 30.333 0.0001415 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Plot each GAM model

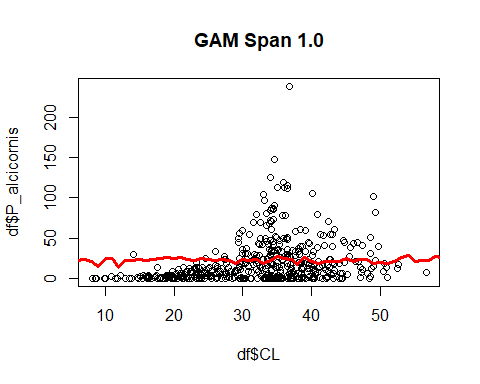
plot(df$P\_alcicornis ~ df$CL,  
 main = "GAM Span 0.1")  
lines(predict(mod.gam1), col = "red", lwd = 3)



plot(df$P\_alcicornis ~ df$CL,  
 main = "GAM Span 0.5")  
lines(predict(mod.gam2), col = "red", lwd = 3)



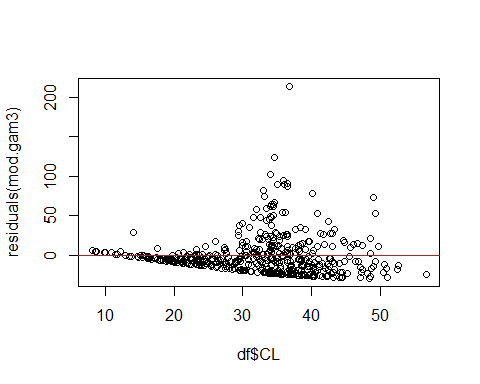
plot(df$P\_alcicornis ~ df$CL,  
 main = "GAM Span 1.0")  
lines(predict(mod.gam3), col = "red", lwd = 3)



the highest value for Span, 1.0, seems to be the most appropriate because the lower values for Span overfit the model to the data and are sensitive to noise within the data.

##plot residuals from best GAM

plot(residuals(mod.gam3) ~ df$CL)  
abline(h = 0, col = "firebrick")



To be honest, this chart looks worse because the trend is more clear, suggesting nonindependence. However, I bet with an apt value for theta with family = nb instead of Gaussian, the GAM residuals would look more independent.

# Step 4

library(mgcv)

## Loading required package: nlme

## This is mgcv 1.8-36. For overview type 'help("mgcv-package")'.

##   
## Attaching package: 'mgcv'

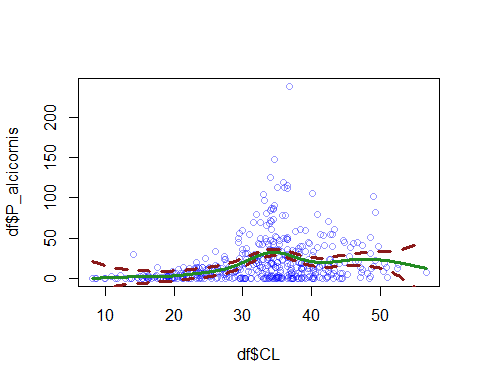
## The following objects are masked from 'package:gam':  
##   
## gam, gam.control, gam.fit, s

mod.gam4 <- mgcv::gam(P\_alcicornis ~ s(CL), data = df)  
summary(mod.gam4)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## P\_alcicornis ~ s(CL)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 19.43 1.13 17.2 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(CL) 6.884 8.011 11.18 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.157 Deviance explained = 16.9%  
## GCV = 607.44 Scale est. = 597.21 n = 468

The GAM shows the same significance value and a low R-squared value, so the summary suggests that the GLM may have been just as good. The pseudo R-squared calculated from the negative binomial GLM is around the same as the R-squared from this GAM.

pred <- predict(mod.gam4, se = T)  
plot(df$P\_alcicornis ~ df$CL,  
 col = rgb(0, 0, 1, alpha = 0.4))  
lines(df$CL[order(df$CL)], pred$fit[order(df$CL)], col = "forestgreen", lwd = 3)  
lines(df$CL[order(df$CL)], pred$fit[order(df$CL)] + 1.96\*pred$se.fit[order(df$CL)], lwd = 3, lty = 2, col = "firebrick4")  
lines(df$CL[order(df$CL)], pred$fit[order(df$CL)] - 1.96\*pred$se.fit[order(df$CL)], lwd = 3, lty = 2, col = "firebrick4")

 This model looks much less sensitive to noise. Overall, I would have come to a far different conclusion using the mgcv package; this GAM prediction shows support of the hypothesis. There is a clear increase in wing worms as age increases until Carapace length of about 30-40. At this point, the number of wing worms slightly decreases, suggesting that the wing worms become out competed towards the end of the crayfish’s life cycle.