Linear Models to GLMs

Danielle Quinn

Tuesday, November 03, 2015

The dataset for this tutorial can be found at:

<https://raw.githubusercontent.com/DanielleQuinn/RLessons/master/Models/LinearModels_to_GLMs/Baileyetal2008.txt>

You will also need to source functions from Highland Statistics Ltd., found at:

<https://github.com/DanielleQuinn/RLessons/blob/master/Models/LinearModels_to_GLMs/HighStatLib.R>

And a couple functions that I've created to make life easier:

<https://github.com/DanielleQuinn/RLessons/blob/master/Models/LinearModels_to_GLMs/MoreFunctions.R>

**Load a few required packages**

library(lattice)  
library(MASS)  
library(ggplot2)

**Now, source the Highland Stats Ltd. and the second set of functions and import the dataset**

source("HighStatLib.R")  
source("MoreFunctions.R")  
Fish<-read.delim("Baileyetal2008.txt")

Briefly, the dataset contains information about the abundance and density of a target fish species at multiple sites between 1977 and 2002, including information about site location and mean depth.

head(Fish)

Site TotAbund Dens MeanDepth Year Period Xkm Ykm  
1 1 76 0.002070281 804 1978 1 98.75575 -57.46692  
2 2 161 0.003519799 808 2001 2 76.80388 178.64798  
3 3 39 0.000980515 809 2001 2 103.79283 -50.05184  
4 4 410 0.008039216 848 1979 1 91.53227 146.44797  
5 5 177 0.005933375 853 2002 2 107.14419 -37.07544  
6 6 695 0.021800502 960 1980 1 86.56470 -48.19807  
 SweptArea  
1 36710.00  
2 45741.25  
3 39775.00  
4 51000.00  
5 29831.25  
6 31880.00

Our underlying research question is has the density-depth relationship changed over time?

**But first, some data cleaning.**

Fish<-na.exclude(Fish) # Subset data to omit NAs  
Fish<-Fish[c(-135), ] # Remove a previously identifed spatial outlier  
Fish$MeanDepth<-Fish$MeanDepth/1000 # Express Depth in km

**We'll start with a simple linear model**

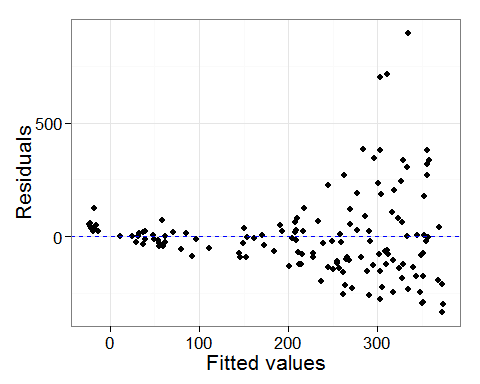
M0<-lm(TotAbund~MeanDepth,data=Fish)  
summary(M0) # Summary

Call:  
lm(formula = TotAbund ~ MeanDepth, data = Fish)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-333.45 -103.40 -13.78 44.46 895.71   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 451.39 33.28 13.562 < 2e-16 \*\*\*  
MeanDepth -97.58 12.32 -7.918 5.92e-13 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 184.9 on 144 degrees of freedom  
Multiple R-squared: 0.3033, Adjusted R-squared: 0.2985   
F-statistic: 62.7 on 1 and 144 DF, p-value: 5.918e-13

E0<-resid(M0) # Residuals  
F0<-fitted(M0) # Fitted Values

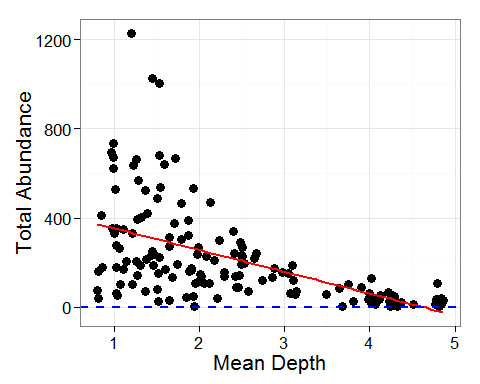
Take a look at the fitted vs residual values

ggplot()+  
 geom\_point(aes(x=F0,y=E0))+  
 geom\_hline(yintercept=0, linetype='dashed', col='blue')+  
 theme\_bw(16)+ylab("Residuals")+xlab("Fitted values")



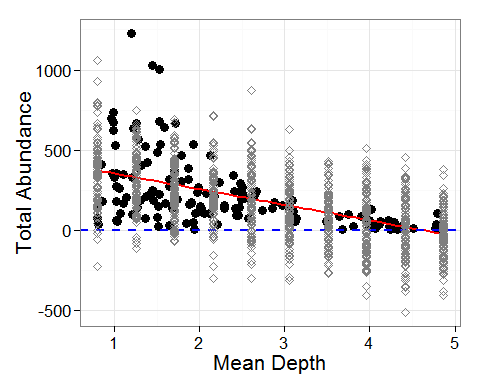
Notice the pattern (heterogeneity) in the residuals? That's the first sign that this isn't a good model. But, let's move ahead anyway and take a look at the predicted values.

bio<-ggplot(Fish)+  
 geom\_point(aes(x=MeanDepth, y=TotAbund), size=3)+  
 stat\_smooth(aes(x=MeanDepth, y=TotAbund), size=1, method='lm', se=FALSE,col='red')+  
 geom\_hline(yintercept=0, linetype='dashed', col='blue', size=1)+  
 theme\_bw(16)+xlab("Mean Depth")+ylab("Total Abundance")  
bio



Hmm.. it appears that at high values of depth, the model predicts *negative* abundances of fish! Let's use the predicted standard error around the model line to randomly generate some potential estimates of abundance.

a<-range(Fish$MeanDepth)  
md<-seq(a[1],a[2],length=10)  
beta<-coef(M0)  
MeanDepth<-c()  
Estimates<-c()  
for (i in 1:10)  
{  
 MeanDepth.in<-rep(md[i],100)  
 MeanDepth<-c(MeanDepth, MeanDepth.in)  
 mu<-beta[1]+beta[2]\*md[i]  
 Estimates.in<-rnorm(100,mean=mu,sd=summary(M0)$sigma)  
 Estimates<-c(Estimates, Estimates.in)  
}  
bio.check<-data.frame(MeanDepth,Estimates)  
  
# Plot Results #  
bio+geom\_point(aes(x=MeanDepth, y=Estimates),col='grey50',pch=5,data=bio.check)



Now we can see that not only does the model predict negative abundances at high values of depth, but some of the simulated predictions fall below zero across all depths! To deal with this, we're going to use a Poisson distribution.

**Let's move on to a simple GLM**

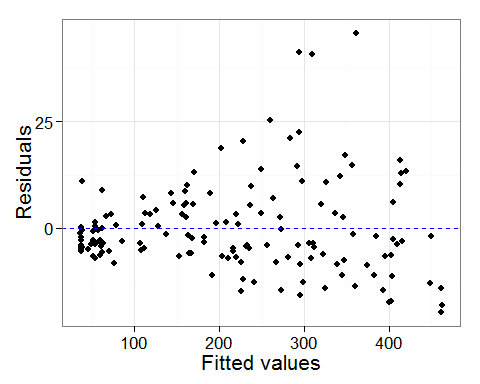
M1<-glm(TotAbund~MeanDepth,data=Fish,family=poisson(link="log"))  
summary(M1)

Call:  
glm(formula = TotAbund ~ MeanDepth, family = poisson(link = "log"),   
 data = Fish)  
  
Deviance Residuals:   
 Min 1Q Median 3Q Max   
-25.544 -6.914 -3.046 3.901 35.744   
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) 6.64334 0.01273 521.70 <2e-16 \*\*\*  
MeanDepth -0.62870 0.00670 -93.84 <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: 27779 on 145 degrees of freedom  
Residual deviance: 15770 on 144 degrees of freedom  
AIC: 16741  
  
Number of Fisher Scoring iterations: 5

E1<-resid(M1,type="pearson")  
F1<-fitted(M1)

Take a look at the fitted vs residual values

ggplot()+  
 geom\_point(aes(x=F1,y=E1))+  
 geom\_hline(yintercept=0, linetype='dashed', col='blue')+  
 theme\_bw(16)+ylab("Residuals")+xlab("Fitted values")



Unfortunately, there's still a pattern in the residuals of our model. However, let's continue by looking at the predicted values. To generate the predicted values, you need to set up a data frame that contains all the combinations of any factors that are included in the model. In this case, it's simple because we only need to consider a single factor: MeanDepth. First, let's only look at a few values of MeanDepth.

newdata<-data.frame(MeanDepth=c(1,2,3,4))  
newdata

MeanDepth  
1 1  
2 2  
3 3  
4 4

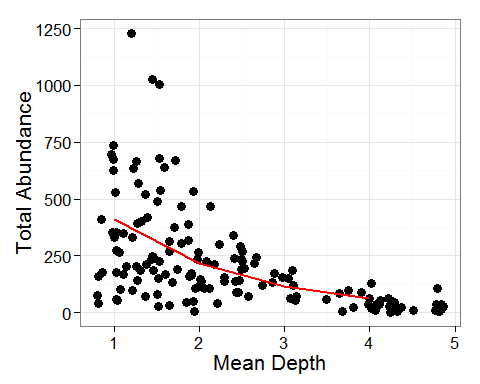
Now we'll use predict() to generate predicted values of TotAbund given the model M1 and the MeanDepth found in newdata.

predict(M1,newdata,type='response')

1 2 3 4   
409.3790 218.3156 116.4244 62.0874

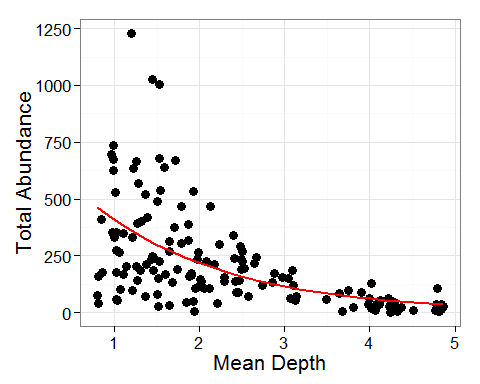
Add the predicted values to newdata and plot the results

newdata$TotAbund<-predict(M1,newdata,type='response')  
  
ggplot()+  
 geom\_point(aes(x=MeanDepth, y=TotAbund), data=Fish, size=3)+  
 geom\_path(aes(x=MeanDepth, y=TotAbund), data=newdata, col="red", size=1)+  
 theme\_bw(16)+ylab("Total Abundance")+xlab("Mean Depth")



Notice how the predicted line doesn't fall below zero anymore! But the line itself is a little jagged because it's only based on four points; the predicted values of TotAbund at depths of 1, 2, 3, and 4. Let's improve on this by predicting TotAbund across the range of depths that the data actually includes.

newdata<-data.frame(MeanDepth=seq(from=range(Fish$MeanDepth)[1],  
 to=range(Fish$MeanDepth)[2],  
 length=25))  
newdata$TotAbund<-predict(M1,newdata,type='response')  
  
predictions<-ggplot()+  
 geom\_point(aes(x=MeanDepth, y=TotAbund), data=Fish, size=3)+  
 geom\_path(aes(x=MeanDepth, y=TotAbund), data=newdata, col="red", size=1)+  
 theme\_bw(16)+ylab("Total Abundance")+xlab("Mean Depth")  
predictions



It's time that we check on dispersion. Essentially, we want to look at the distribution of the data and discribe it as a non-negative number.

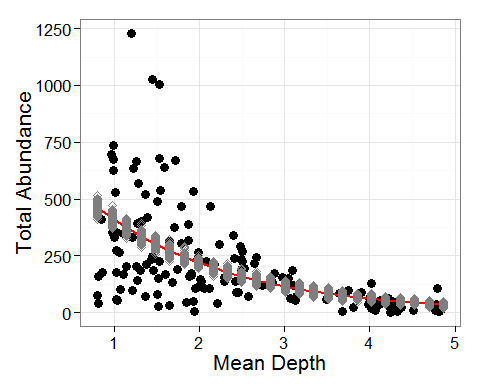
dispersion(M1)

[1] 115.5856

Our model has a dispersion of 115.6! When modelling, you'll want to aim for a dispersion of 1. In this case we have *overdispersion*. The model assumes that the variability of counts within a covariate group is equal to the mean. So, if the *variance is greater than the mean*, this will lead to underestimated standard errors, and overestimated significance of regression parameters.

To better understand, let's simulate dispersion around our predicted values.

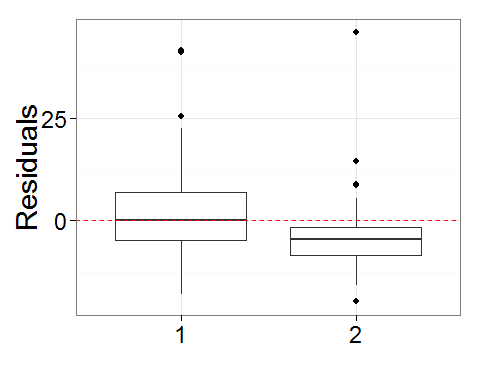
md<-seq(from=range(Fish$MeanDepth)[1],  
 to=range(Fish$MeanDepth)[2],  
 length=25)  
beta<-coef(M1)  
MeanDepth<-c()  
Estimates<-c()  
for(i in 1:25)  
{  
 MeanDepth.in<-rep(md[i],100)  
 MeanDepth<-c(MeanDepth, MeanDepth.in)  
 mu<-exp(beta[1]+beta[2]\*md[i])  
 Estimates.in<-rpois(100,lambda=mu)  
 Estimates<-c(Estimates, Estimates.in)  
}  
bio.check<-data.frame(MeanDepth,Estimates)  
  
# Plot #  
predictions+  
 geom\_point(aes(x=MeanDepth, y=Estimates),col='grey50',pch=5,data=bio.check)



You can see that essentially, the distribution of our predicted values (grey points) don't meaningfully describe the actual data (black points).

It's possible that all of these issues could be a result of a factor that is influencing the data but not being included in the model. Let's take a look at how Period might influence the data. Plot the residuals against a variable not included in the model (Period).

pr.fac(M1, as.factor(Fish$Period))



What we want to see here is that the residuals associated with each period are normally distributed around 0. Looking at this plot, we see that this isn't true for residuals associated with period 2! This suggests to us that we should include Period as a factor in subsequent models.

**Adding a factor to a model**

Fish$fPeriod<-as.factor(Fish$Period)  
M2<-glm(TotAbund~MeanDepth\*fPeriod, data=Fish, family='poisson')  
summary(M2)

Call:  
glm(formula = TotAbund ~ MeanDepth \* fPeriod, family = "poisson",   
 data = Fish)  
  
Deviance Residuals:   
 Min 1Q Median 3Q Max   
-25.298 -6.375 -1.721 3.323 44.621   
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) 6.832036 0.014837 460.473 < 2e-16 \*\*\*  
MeanDepth -0.658858 0.007935 -83.031 < 2e-16 \*\*\*  
fPeriod2 -0.674857 0.029189 -23.120 < 2e-16 \*\*\*  
MeanDepth:fPeriod2 0.115712 0.014908 7.762 8.39e-15 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for poisson family taken to be 1)  
  
 Null deviance: 27779 on 145 degrees of freedom  
Residual deviance: 14293 on 142 degrees of freedom  
AIC: 15268  
  
Number of Fisher Scoring iterations: 5

dispersion(M2)

[1] 111.1873

It turns out that this model still suffers from overdispersion. Rather than explore further, let's just move on.

One thing that we haven't considered yet is that the swept area of different sites varies, meaning that the effort put into finding abundance also varies. This could be the source of our problems! Let's include SweptArea as an offset in our next model.

**Adding an offest to a model**

Fish$logSweptArea<-log(Fish$SweptArea)  
M3<-glm(TotAbund~MeanDepth\*fPeriod+offset(logSweptArea),   
 data=Fish, family='poisson')  
summary(M3)

Call:  
glm(formula = TotAbund ~ MeanDepth \* fPeriod + offset(logSweptArea),   
 family = "poisson", data = Fish)  
  
Deviance Residuals:   
 Min 1Q Median 3Q Max   
-26.286 -5.989 -1.444 3.239 47.137   
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -3.426658 0.014987 -228.644 <2e-16 \*\*\*  
MeanDepth -0.968647 0.008034 -120.572 <2e-16 \*\*\*  
fPeriod2 -0.767278 0.029772 -25.771 <2e-16 \*\*\*  
MeanDepth:fPeriod2 0.129206 0.015275 8.459 <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: 45669 on 145 degrees of freedom  
Residual deviance: 14976 on 142 degrees of freedom  
AIC: 15951  
  
Number of Fisher Scoring iterations: 5

E3<-resid(M3,type='pearson')  
dispersion(M3)

[1] 122.4857

We still have overdispersion. It's time to move away from the Poisson distribution and handle this overdispersion directly.

**Negative Binomial (NB) GLMs** Basically, NB GLMs use an additional parameter theta that accounts for the variance being greater than the mean (overdispersion). In our NB GLM we're going to include MeanDepth and Period as factors, use logSweptArea as an offset, and also include the interaction between MeanDepth and Period.

M4<-glm.nb(TotAbund~MeanDepth\*fPeriod+offset(logSweptArea),data=Fish)  
summary(M4)

Call:  
glm.nb(formula = TotAbund ~ MeanDepth \* fPeriod + offset(logSweptArea),   
 data = Fish, init.theta = 1.950355727, link = log)  
  
Deviance Residuals:   
 Min 1Q Median 3Q Max   
-3.3575 -0.8384 -0.1985 0.3658 2.8850   
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -3.25515 0.16064 -20.263 <2e-16 \*\*\*  
MeanDepth -1.03764 0.06032 -17.202 <2e-16 \*\*\*  
fPeriod2 -0.61216 0.27395 -2.235 0.0254 \*   
MeanDepth:fPeriod2 0.07571 0.10209 0.742 0.4583   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for Negative Binomial(1.9504) family taken to be 1)  
  
 Null deviance: 461.62 on 145 degrees of freedom  
Residual deviance: 158.32 on 142 degrees of freedom  
AIC: 1754.7  
  
Number of Fisher Scoring iterations: 1  
  
 Theta: 1.950   
 Std. Err.: 0.219   
  
 2 x log-likelihood: -1744.729

dispersion(M4,modeltype='nb')

[1] 1.001618

Notice that our dispersion is now close to 1!

From here, we're going to decide if including all of the factors and interactions is meaningful to the model. To look at which, if any, terms should be dropped from the model:

drop1(M4,test="Chi")

Single term deletions  
  
Model:  
TotAbund ~ MeanDepth \* fPeriod + offset(logSweptArea)  
 Df Deviance AIC LRT Pr(>Chi)  
<none> 158.32 1752.7   
MeanDepth:fPeriod 1 158.79 1751.2 0.47576 0.4904

This suggests that the interaction term between MeanDepth and Period isn't necessary to the model (p > 0.05) and can be dropped.

**Dropping a level from the model**

M5<-glm.nb(TotAbund~MeanDepth+fPeriod+offset(logSweptArea),data=Fish)  
summary(M5)

Call:  
glm.nb(formula = TotAbund ~ MeanDepth + fPeriod + offset(logSweptArea),   
 data = Fish, init.theta = 1.943752383, link = log)  
  
Deviance Residuals:   
 Min 1Q Median 3Q Max   
-3.3467 -0.8318 -0.2305 0.3692 2.7250   
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -3.31134 0.13703 -24.164 < 2e-16 \*\*\*  
MeanDepth -1.01372 0.04876 -20.788 < 2e-16 \*\*\*  
fPeriod2 -0.43027 0.12650 -3.401 0.000671 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for Negative Binomial(1.9438) family taken to be 1)  
  
 Null deviance: 460.09 on 145 degrees of freedom  
Residual deviance: 158.28 on 143 degrees of freedom  
AIC: 1753.2  
  
Number of Fisher Scoring iterations: 1  
  
 Theta: 1.944   
 Std. Err.: 0.218   
  
 2 x log-likelihood: -1745.204

E5<-resid(M5,type='pearson')  
F5<-predict(M5,type="link")  
dispersion(M5,modeltype="nb")

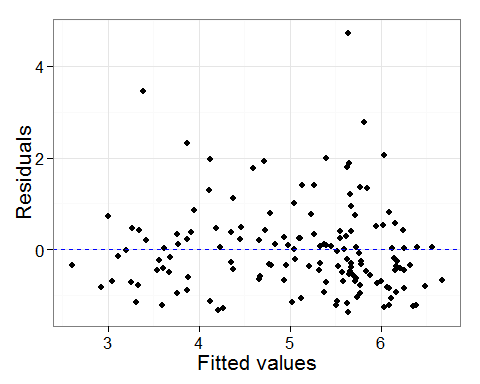
[1] 0.9718338

drop1(M5, test="Chi")

Single term deletions  
  
Model:  
TotAbund ~ MeanDepth + fPeriod + offset(logSweptArea)  
 Df Deviance AIC LRT Pr(>Chi)   
<none> 158.28 1751.2   
MeanDepth 1 442.78 2033.7 284.504 < 2.2e-16 \*\*\*  
fPeriod 1 169.27 1760.2 10.989 0.0009165 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

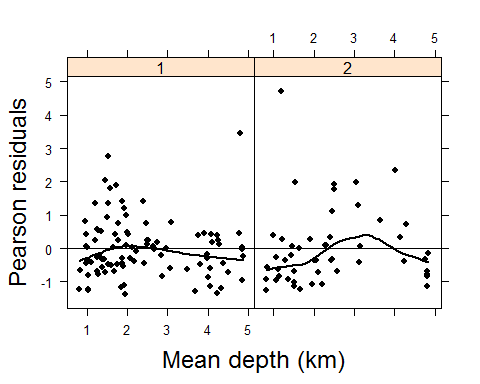
Now we can see that all of our factors are meaningful to the model, and we no longer have overdispersion. Check the residuals versus fitted values for heterogeneity.

ggplot()+  
 geom\_point(aes(x=F5,y=E5))+  
 geom\_hline(yintercept=0, linetype='dashed', col='blue')+  
 theme\_bw(16)+ylab("Residuals")+xlab("Fitted values")



Great, there is no pattern! Now plot the residuals against each factor included in the model.

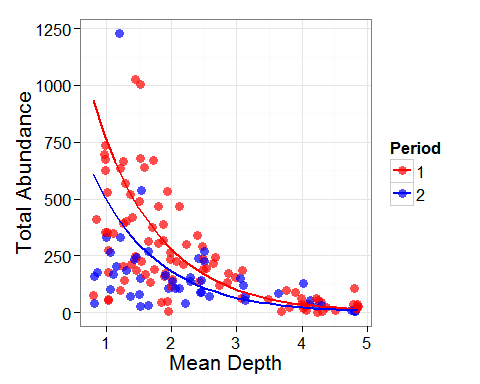
xyplot(E5 ~ MeanDepth | factor(Period),   
 data = Fish,  
 xlab = list(label = "Mean depth (km)", cex = 1.5),  
 ylab = list(label = "Pearson residuals", cex = 1.5),  
 panel = function(x,y)  
 {  
 panel.points(x,y, col = 1, pch = 16, cex = 0.7)  
 panel.loess(x,y, col = 1, lwd = 2)  
 panel.abline(h=0)  
 }  
 )



Also no patterns here!

Time to generate our predicted values of the model by creating newdata. Use the function expand.grid to easily generate a dataframe that contains all combinations of the values you require.

newdata<-expand.grid(MeanDepth=seq(from=range(Fish$MeanDepth)[1], to=range(Fish$MeanDepth)[2], length=25),  
 fPeriod=as.factor(c(1,2)),  
 logSweptArea=mean(log(Fish$SweptArea)))  
newdata$TotAbund<-predict(M5,newdata,type="response")  
  
# Plot #  
plotM5<-ggplot()+  
 geom\_point(aes(x=MeanDepth, y=TotAbund, col=fPeriod), data=Fish, size=3, alpha=0.7)+  
 geom\_path(aes(x=MeanDepth, y=TotAbund, col=fPeriod), data=newdata, size=1)+  
 theme\_bw(16)+ylab("Total Abundance")+xlab("Mean Depth")+  
 scale\_colour\_manual(values=c("red","blue"),name="Period")  
plotM5

 You can see that the model predictions vary slightly based on the Period that is being considered.

Finally, let's add confidence limits around the predicted model lines.

newdata$fit<-predict(M5,newdata,type="link",se=TRUE)$fit  
newdata$se<-predict(M5,newdata,type="link",se=TRUE)$se  
  
plotM5+  
 geom\_path(aes(x=MeanDepth, y=exp(fit-1.96\*se), col=fPeriod),  
 alpha=0.7, linetype='dashed', data=newdata)+  
 geom\_path(aes(x=MeanDepth, y=exp(fit+1.96\*se), col=fPeriod),  
 alpha=0.7, linetype='dashed', data=newdata)

