Homework 5

Skylar Liu

2024-02-04

Homework #5, Stat 660, Spring 2024, Due Class #7, February 7, 2024

1. Do a similar analysis about heteroscedasticity for each choice of k as was done in Lecture 5. Plot the fitted absolute residuals against either the predicted values or against X, your choice. Put them on one graph, and display.

```
# clear workspace
rm(list = ls())
# set the seed
set.seed(382957)
library(HRW)
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

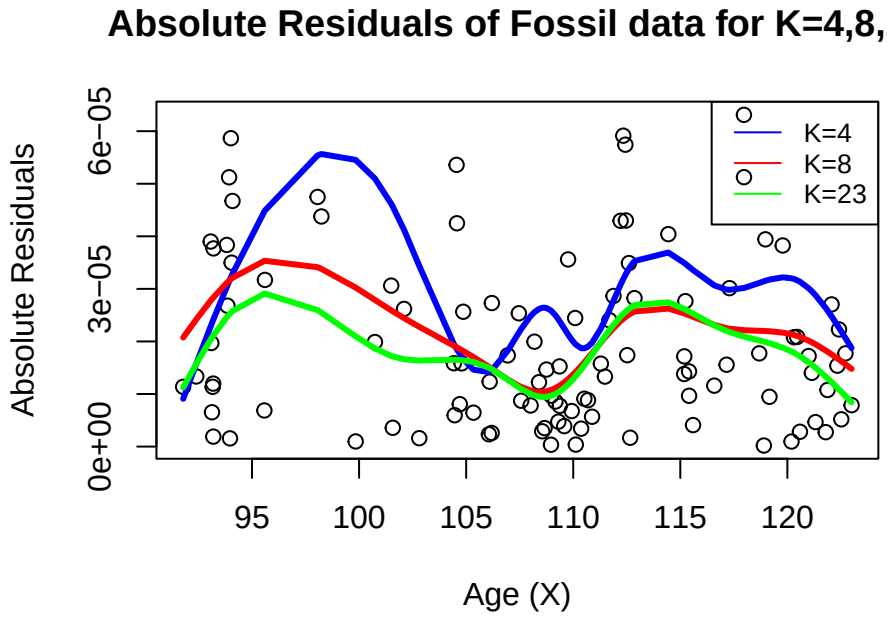
```
# import data
fossil = read.csv("~/660 - Flexible Regression/Homework/Homework3/fossil.csv")
X = fossil$Age
Y = fossil$Strontium.Ratio

# Fit a default gam to the data
gam_default = gam(Y~s(X,bs="cr"))

# mgcv fit with K = 4 knots
gam4 = gam(Y~s(X,bs="cr",k=4))
# mgcv fit with K = 8 knots
gam8 = gam(Y~s(X,bs="cr",k=8))
# mgcv fit with K = 23 knots
gam23 = gam(Y~s(X,bs="cr",k=23))

# absolute residual plot against predicted values
ord = sort(X, index.return = T)$ix
plot(X, abs(gam_default$residuals),
     ylab="Absolute Residuals",
     main="Absolute Residuals of Fossil data for K=4,8,23",
     xlab="Age (X)")
y4    = abs(gam4$residuals)
y8    = abs(gam8$residuals)
```

```
y23    = abs(gam23$residuals)
gam4_abs = gam(y4~s(X,bs="cr"))
gam8_abs = gam(y8~s(X,bs="cr"))
gam23_abs = gam(y23~s(X,bs="cr"))
lines(X[ord],fitted(gam4_abs)[ord],lwd=3,col="blue")
lines(X[ord],fitted(gam8_abs)[ord],lwd=3,col="red")
lines(X[ord],fitted(gam23_abs)[ord],lwd=3,col="green")
legend("topright", legend = c("K=4", "K=8", "K=23"), col=c("blue", "red", "green"),lty=1, cex=0.8)
```



2. Display the ratio of the maximum fitted absolute residual to the minimum fitted absolute residual, for each case.

```
# ratio of the maximum fitted absolute residual to the minimum fitted absolute residual
ratio_k4 = max(abs(fitted(gam4_abs))) / min(abs(fitted(gam4_abs)))
ratio_k8 = max(abs(fitted(gam8_abs))) / min(abs(fitted(gam8_abs)))
ratio_k23 = max(abs(fitted(gam23_abs))) / min(abs(fitted(gam23_abs)))
cat('k = 4 fit, maximum to minimum ratio = ',ratio_k4,"\n")
```

```
## k = 4 fit, maximum to minimum ratio =  6.127354
```

```
cat('k = 8 fit, maximum to minimum ratio = ',ratio_k8,"\n")
```

```
## k = 8 fit, maximum to minimum ratio =  3.392537
```

```
cat('k = 23 fit, maximum to minimum ratio = ',ratio_k23,"\n")
```

```
## k = 23 fit, maximum to minimum ratio =  3.452144
```

All of these fits fail the rule of thumb, since the ratios are all above 3. K=8 and K=23 are just over the line while K=4 is well over 3.

4. If any choice of k fails my rule of thumb, describe in words what that means. Remember, it is possible that the heteroscedasticity can occur in the middle of the X values, and not merely at the end.

   Since all of these values fail the rule of thumb, there is heteroscedasticity somewhere in the model and we cannot trust the confidnce intervals.

5. Give a verbal description of what k is.

   K is the number of knots, which are points where the slope appears to change on a continuous fitted line. Increased knots increase flexibility and wiggles in the line.

6. Load the trees data set: install.packages("trees", dependencies=TRUE). Then library(trees). Ignore any messages. The response Y is volume of the tree, and the predictors are girth and height of the tree. Run mgcv::gam with volume as the response, height entering the model linearly, and girth entering as a cubic Bspline with k=5 (there are only 32 or so data points, so too many basis functions are silly). Show the summary output and plot the model object (just guess).

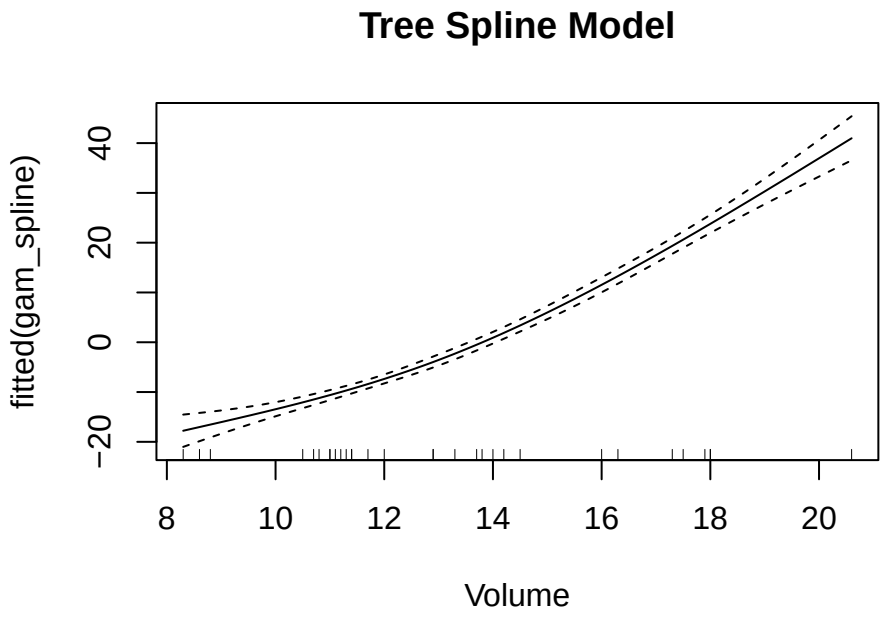
```
# import the tree package
library(tree)

# define variables
volume = trees$Volume
height = trees$Height
girth  = trees$Girth

# fit the tree data
gam_spline = gam(volume ~ height + s(girth,bs="cr", k=5))
summary(gam_spline)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## volume ~ height + s(girth, bs = "cr", k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.46708    7.16334   0.205 0.839293
## height       0.37768    0.09404   4.016 0.000437 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F p-value
## s(girth) 2.537  3.059 228.8  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.974   Deviance explained = 97.7%
## GCV = 8.3565  Scale est. = 7.1334    n = 31
```

```r
# plot the model
plot(gam_spline,
     main="Tree Spline Model",
     xlab="Volume", ylab="fitted(gam_spline)", cex=2)
```



7. Rerun the model, except this time making girth a linear predictor. Compare the two models using an anova statement. Try anova('Linear Fit Name","Spline fit Name",type="?"), and display the p-value. With such a small sample size, do not be surprised if the p-value is not < 0.05.

```r
# fit using girth as a linear predictor
gam_linear = gam(volume ~ height + girth)

# anova comparison
anova(gam_linear, gam_spline, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: volume ~ height + girth
## Model 2: volume ~ height + s(girth, bs = "cr", k = 5)
##   Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
## 1    28.000     421.92
## 2    25.941     188.77 2.0591   233.15 8.839e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
cat('p-value = ',anova(gam_linear, gam_spline, test = "Chisq")$"Pr(>Chi)"[2],"\n")
```

```
## p-value =  8.83872e-08
```