Homework #9

Mike Lehman

October 31, 2016

#1  
boston <- read.csv("Boston.csv")

#2  
set.seed(12345)  
train\_sample <- sample(506, 354) #70% = 354.2  
boston\_train <- boston[train\_sample,]  
boston\_test <- boston[-train\_sample,]

#3  
library(mgcv)

## Loading required package: nlme

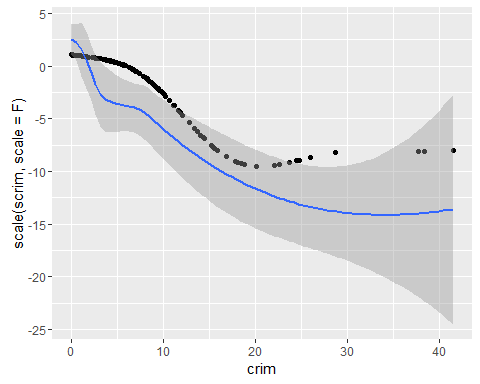
## This is mgcv 1.8-12. For overview type 'help("mgcv-package")'.

formula.glin <- as.formula("medv ~ s(crim) + s(nox) + s(rm) + s(age) + s(dis) + s(tax) + s(ptratio)")  
glin.model <- gam(formula.glin, data = boston\_train)  
summary(glin.model)

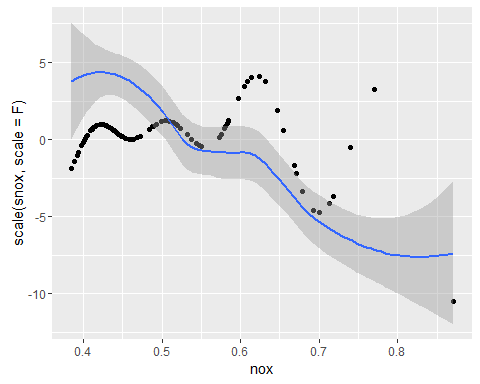
##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## medv ~ s(crim) + s(nox) + s(rm) + s(age) + s(dis) + s(tax) +   
## s(ptratio)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 22.7037 0.2031 111.8 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(crim) 4.071 5.030 10.442 2.67e-09 \*\*\*  
## s(nox) 8.599 8.937 13.954 < 2e-16 \*\*\*  
## s(rm) 8.249 8.841 57.931 < 2e-16 \*\*\*  
## s(age) 1.000 1.000 22.193 3.66e-06 \*\*\*  
## s(dis) 8.811 8.987 8.923 3.66e-12 \*\*\*  
## s(tax) 6.756 7.784 3.223 0.00177 \*\*   
## s(ptratio) 1.000 1.000 26.040 5.68e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.826 Deviance explained = 84.5%  
## GCV = 16.432 Scale est. = 14.599 n = 354

#4  
sVars <- predict(glin.model, type = "terms")  
boston\_train <- cbind(boston\_train, scrim = sVars[,1], snox = sVars[,2], srm = sVars[,3], sage = sVars[,4], sdis = sVars[,5], stax = sVars[,6], sptratio = sVars[,7])

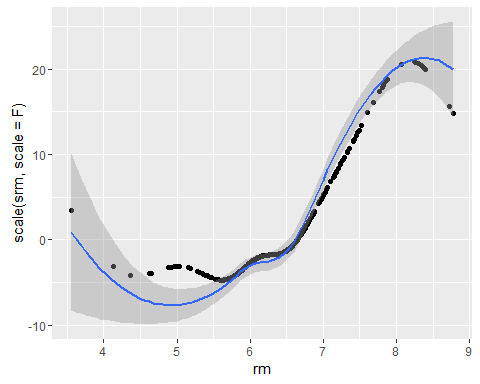
#5  
library(ggplot2)  
ggplot(boston\_train, aes(x = crim)) + geom\_point(aes(y = scale(scrim, scale = F))) + geom\_smooth(aes(y = scale(medv, scale = F)))



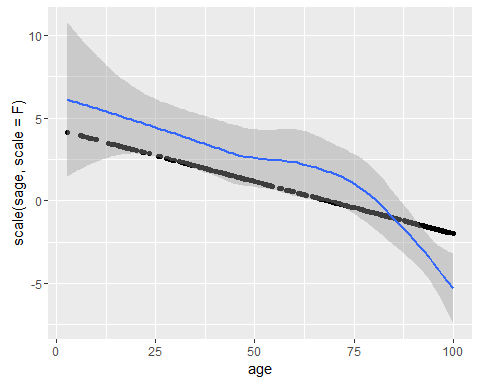
ggplot(boston\_train, aes(x = nox)) + geom\_point(aes(y = scale(snox, scale = F))) + geom\_smooth(aes(y = scale(medv, scale = F)))



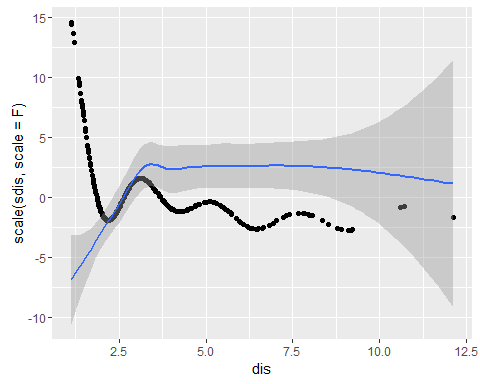
ggplot(boston\_train, aes(x = rm)) + geom\_point(aes(y = scale(srm, scale = F))) + geom\_smooth(aes(y = scale(medv, scale = F)))



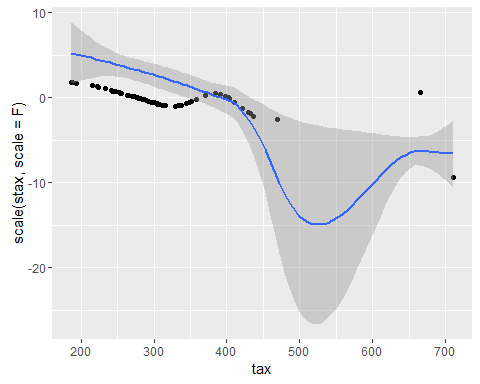
ggplot(boston\_train, aes(x = age)) + geom\_point(aes(y = scale(sage, scale = F))) + geom\_smooth(aes(y = scale(medv, scale = F)))



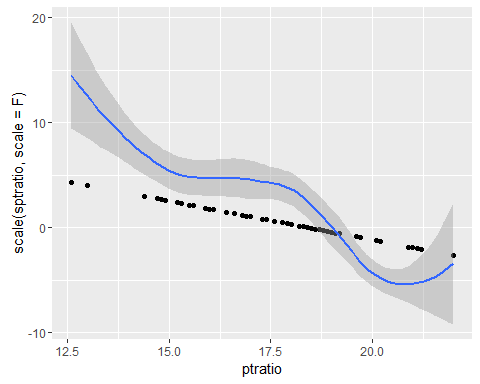
ggplot(boston\_train, aes(x = dis)) + geom\_point(aes(y = scale(sdis, scale = F))) + geom\_smooth(aes(y = scale(medv, scale = F)))



ggplot(boston\_train, aes(x = tax)) + geom\_point(aes(y = scale(stax, scale = F))) + geom\_smooth(aes(y = scale(medv, scale = F)))



ggplot(boston\_train, aes(x = ptratio)) + geom\_point(aes(y = scale(sptratio, scale = F))) + geom\_smooth(aes(y = scale(medv, scale = F)))



#6  
residual.glin.train <- boston\_train$medv - predict(glin.model, data = boston\_train)  
residual.glin.test <- boston\_test$medv - predict(glin.model, newdata = boston\_test)  
sqrt(mean(residual.glin.test^2)) #4.24814

## [1] 4.24814

sqrt(mean(residual.glin.train^2)) #3.601483

## [1] 3.601483

The R-Squared values are close, which tells us that the model is not overfitted to the training data and would likely translate well to unseen data.

#7  
summary(glin.model)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## medv ~ s(crim) + s(nox) + s(rm) + s(age) + s(dis) + s(tax) +   
## s(ptratio)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 22.7037 0.2031 111.8 <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(crim) 4.071 5.030 10.442 2.67e-09 \*\*\*  
## s(nox) 8.599 8.937 13.954 < 2e-16 \*\*\*  
## s(rm) 8.249 8.841 57.931 < 2e-16 \*\*\*  
## s(age) 1.000 1.000 22.193 3.66e-06 \*\*\*  
## s(dis) 8.811 8.987 8.923 3.66e-12 \*\*\*  
## s(tax) 6.756 7.784 3.223 0.00177 \*\*   
## s(ptratio) 1.000 1.000 26.040 5.68e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.826 Deviance explained = 84.5%  
## GCV = 16.432 Scale est. = 14.599 n = 354

The model did quite well in predicting the target variable. The overall deviance explained is high (84.5%), which explains a great deal of the deviance of predictions in the dependent variable. There are a few smoothed features with a low P value (nox, rm, and tax) which indicate those independent features have a noticable impact on the outcome variable, P value below 0.05.

#8

If we remove crim, age, dis, and ptratio from the model, there would be minimal impact on the preditive power of the model. The smoothed versions of these features all have a higher P value.