Assignment 3

James Thompson, Sharanya Sivaraman, Neda Zolaktaf

2019-10-04

Question 1 (Chapter 3, #15)

This problem involves the Boston data set, which we saw in the lab for this chapter. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

(a)For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

```
#load packages
library(MASS)
data(Boston)
library(ggplot2)

#view the data
head(Boston)
```

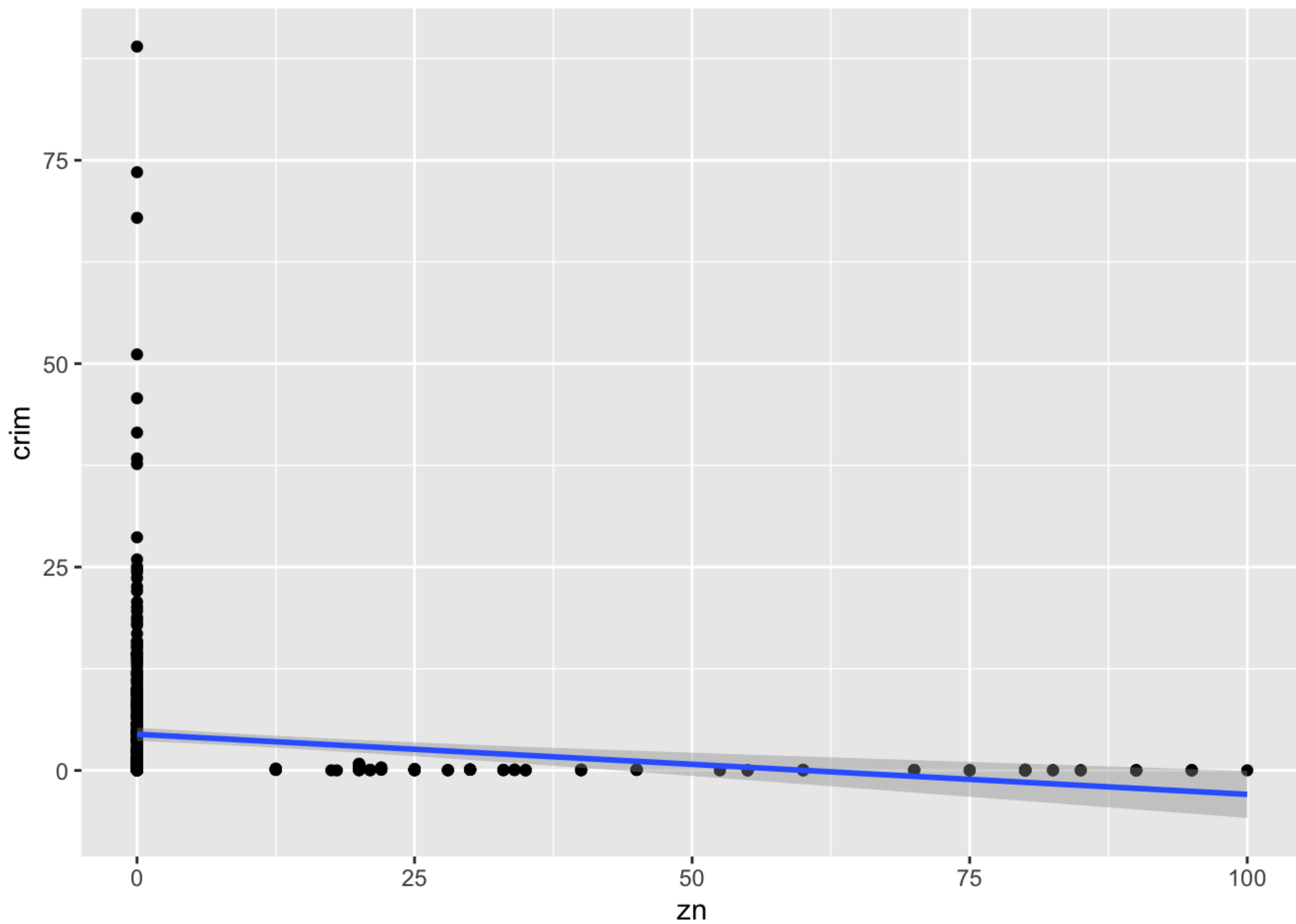
```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12
##   lstat medv
## 1  4.98 24.0
## 2  9.14 21.6
## 3  4.03 34.7
## 4  2.94 33.4
## 5  5.33 36.2
## 6  5.21 28.7
```

```
#Do a bunch at once
# modBoston <- function(x) {
#   form1 <- formula(paste0("crim~",x))
#   fit1 <- lm(form1,data=Boston)
#   summary(fit1)
# }
# nn <- names(Boston)
# for(i in 2:length(nn)) {
#   print(nn[i])
#   print(modBoston(nn[i]))
#   print("-----")
# }
```

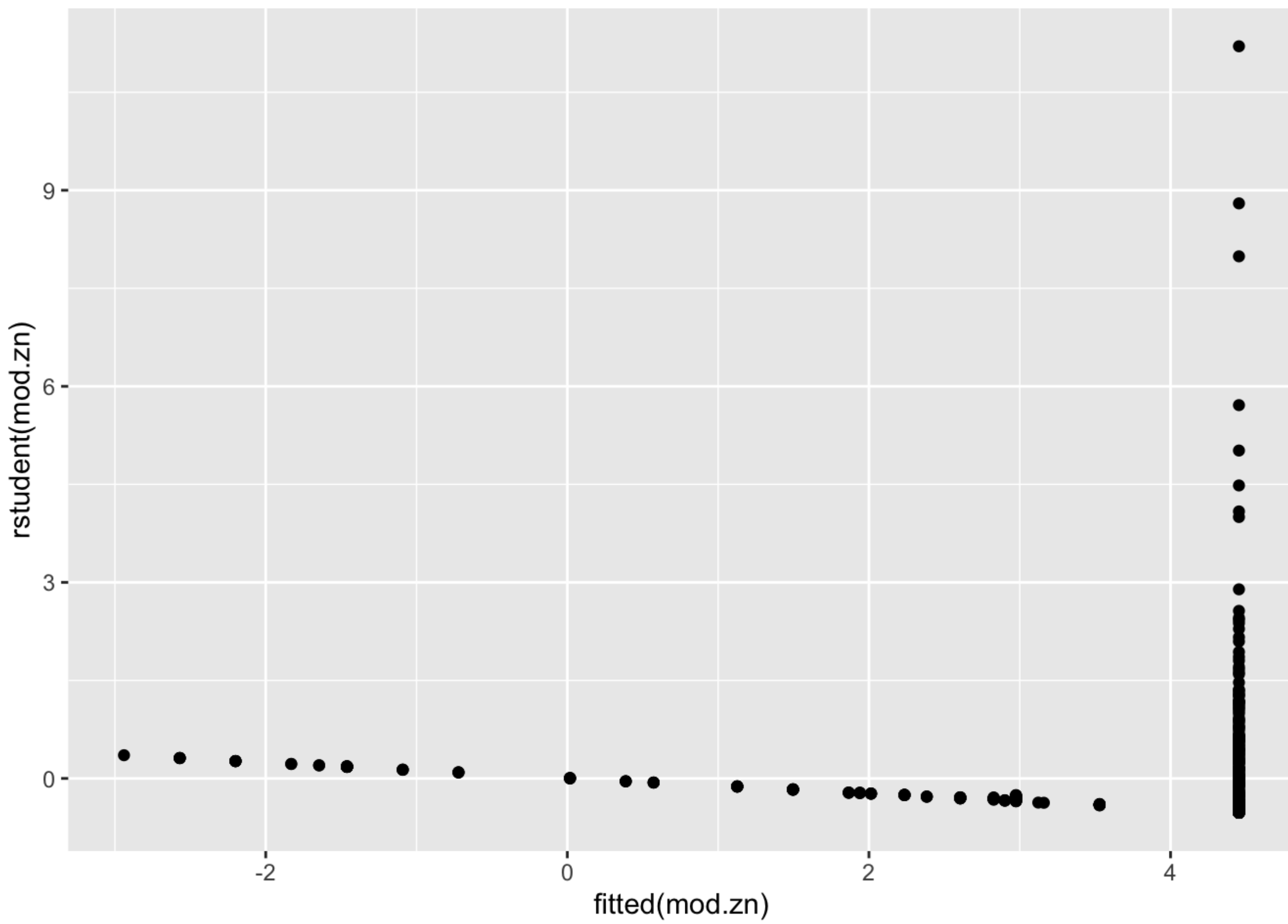
```
#zn
mod.zn = lm(crim~zn,data=Boston)
summary(mod.zn)
```

```
##
## Call:
## lm(formula = crim ~ zn, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```
par(mfrow=c(2,2))
ggplot(Boston,aes(y=crim,x=zn)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.zn),x=fitted(mod.zn))) + geom_point()
```

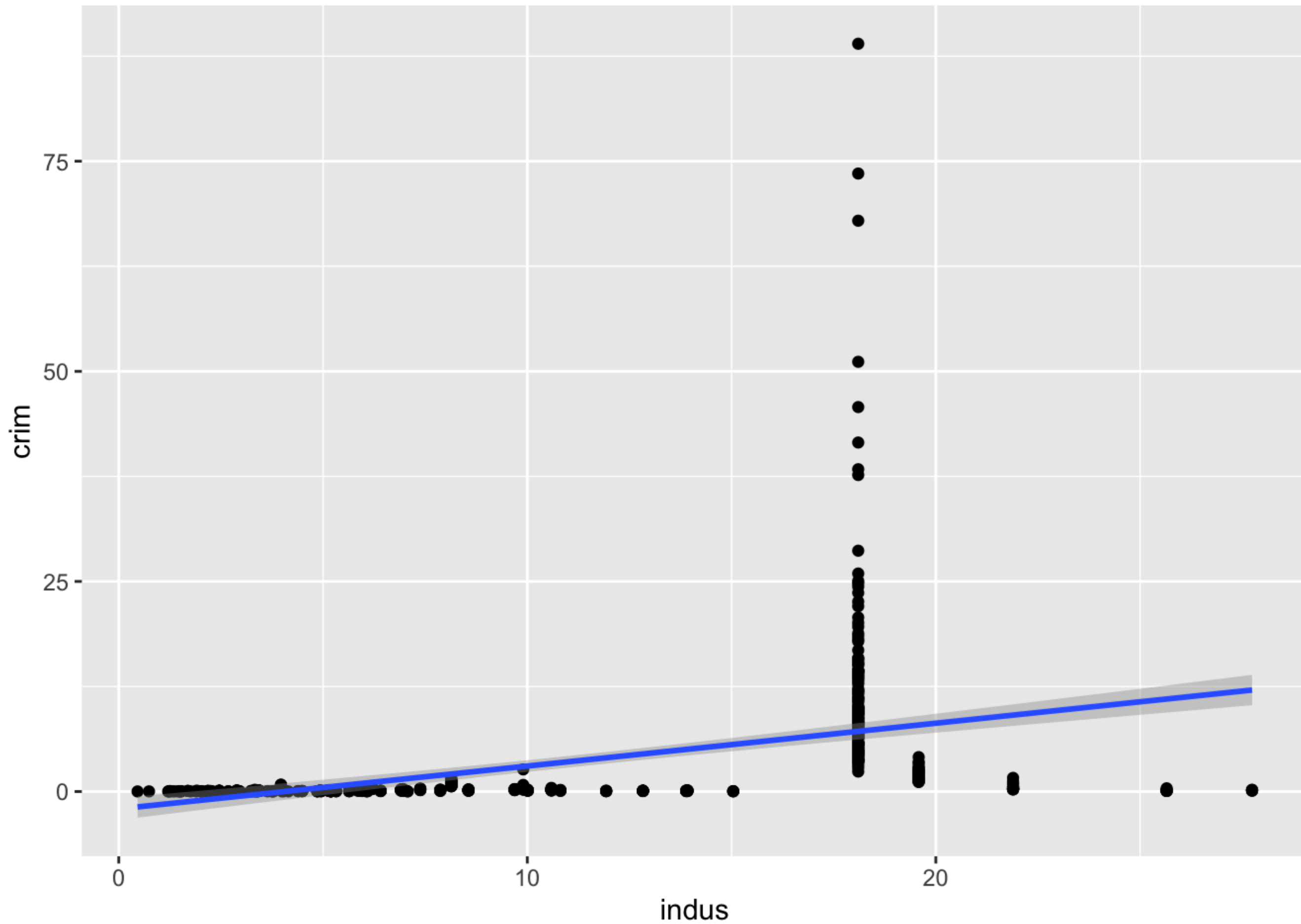


Statistically significant relationship between zn and crim. p-value for slope coefficient of zn = 5.51e-06 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

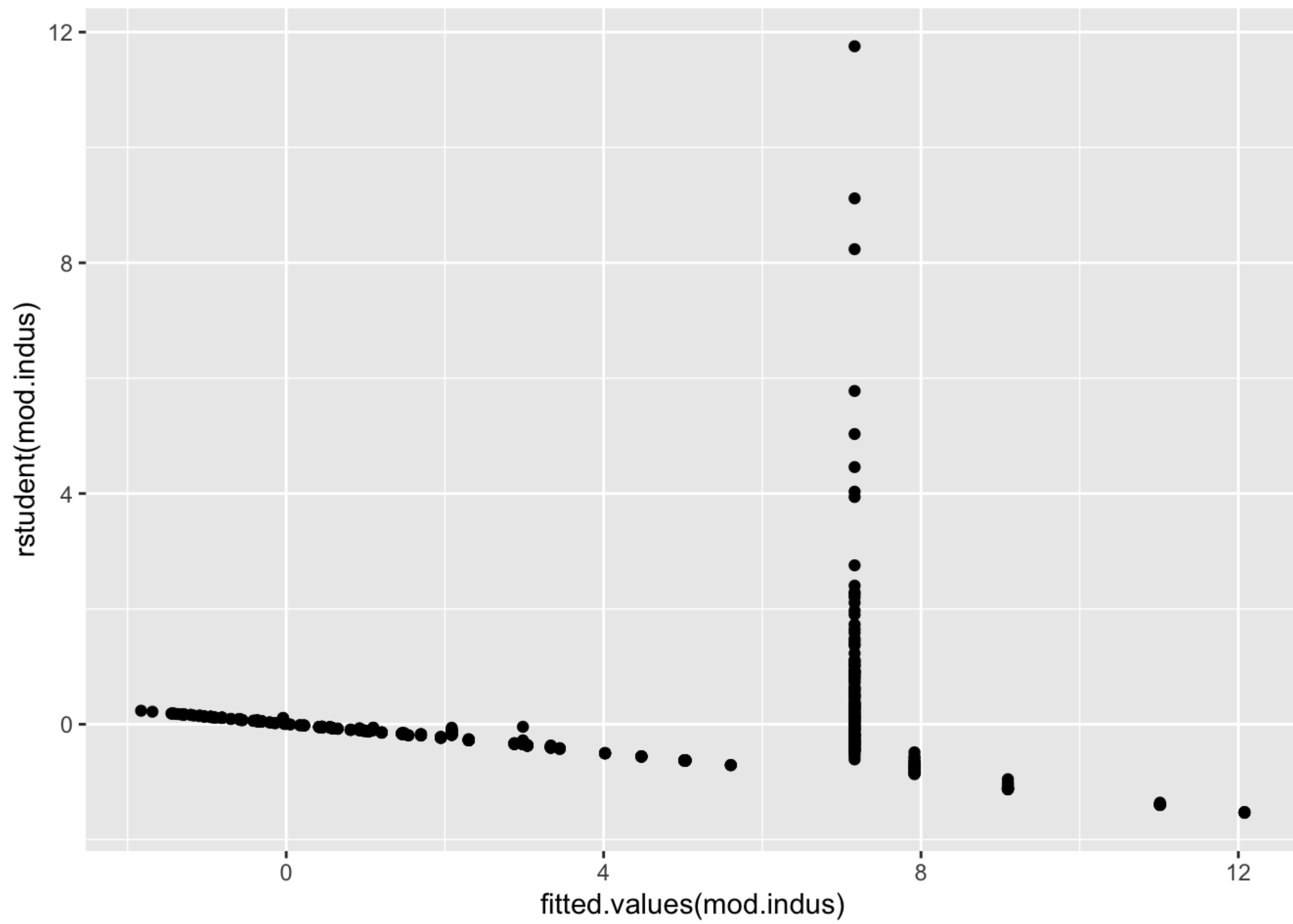
```
#indus
mod.indus = lm(crim~indus,data=Boston)
summary(mod.indus)
```

```
##
## Call:
## lm(formula = crim ~ indus, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
ggplot(Boston,aes(y=crim,x=indus)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.indus),x=fitted.values(mod.indus))) + geom_point()
```

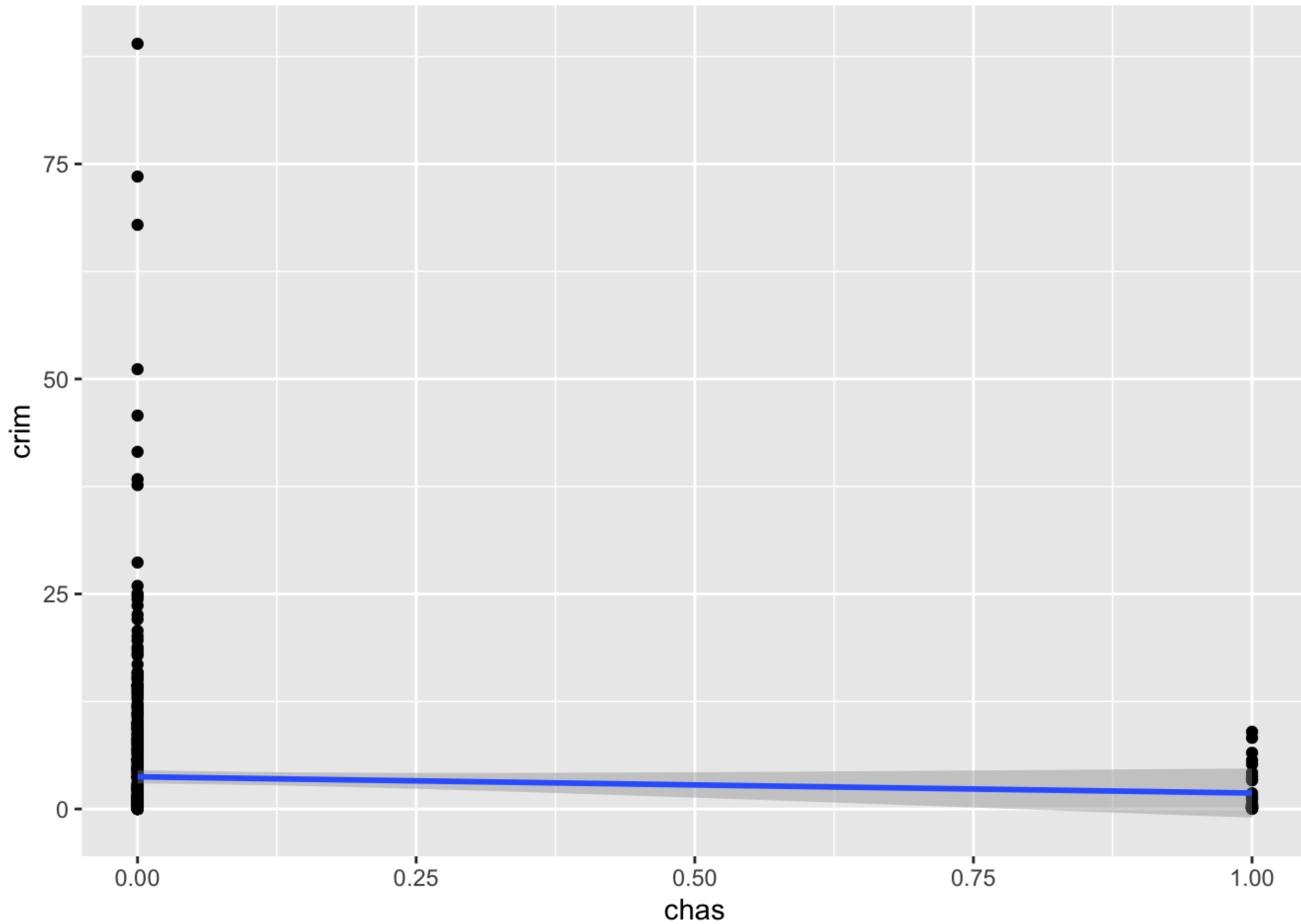


Statistically significant relationship between indus and crim. p-value for slope coefficient of indus < 2e-16 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

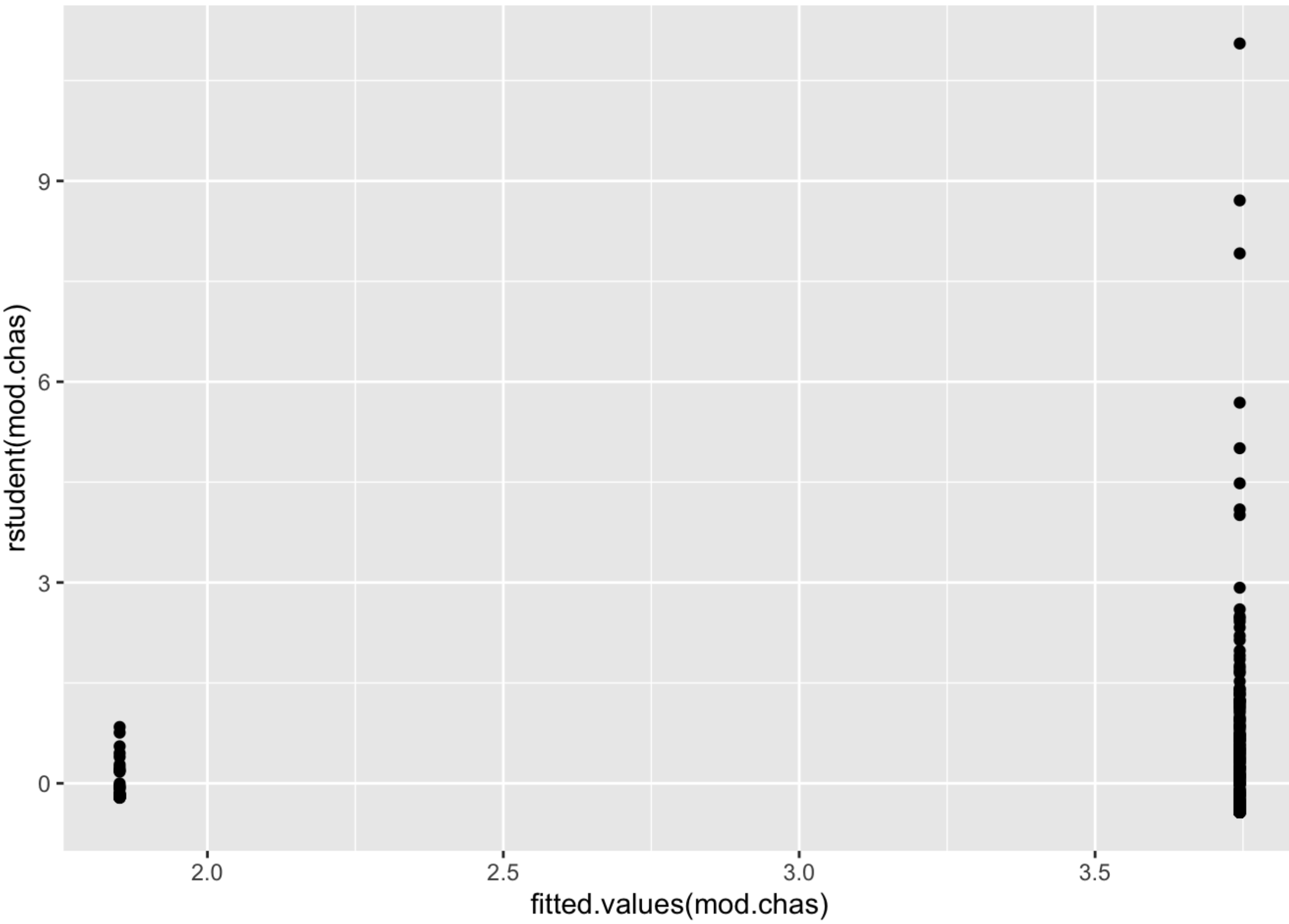
```
#chas
mod.chas = lm(crim~chas,data=Boston)
summary(mod.chas)
```

```
##
## Call:
## lm(formula = crim ~ chas, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

```
ggplot(Boston,aes(y=crim,x=chas)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.chas),x=fitted.values(mod.chas))) + geom_point()
```

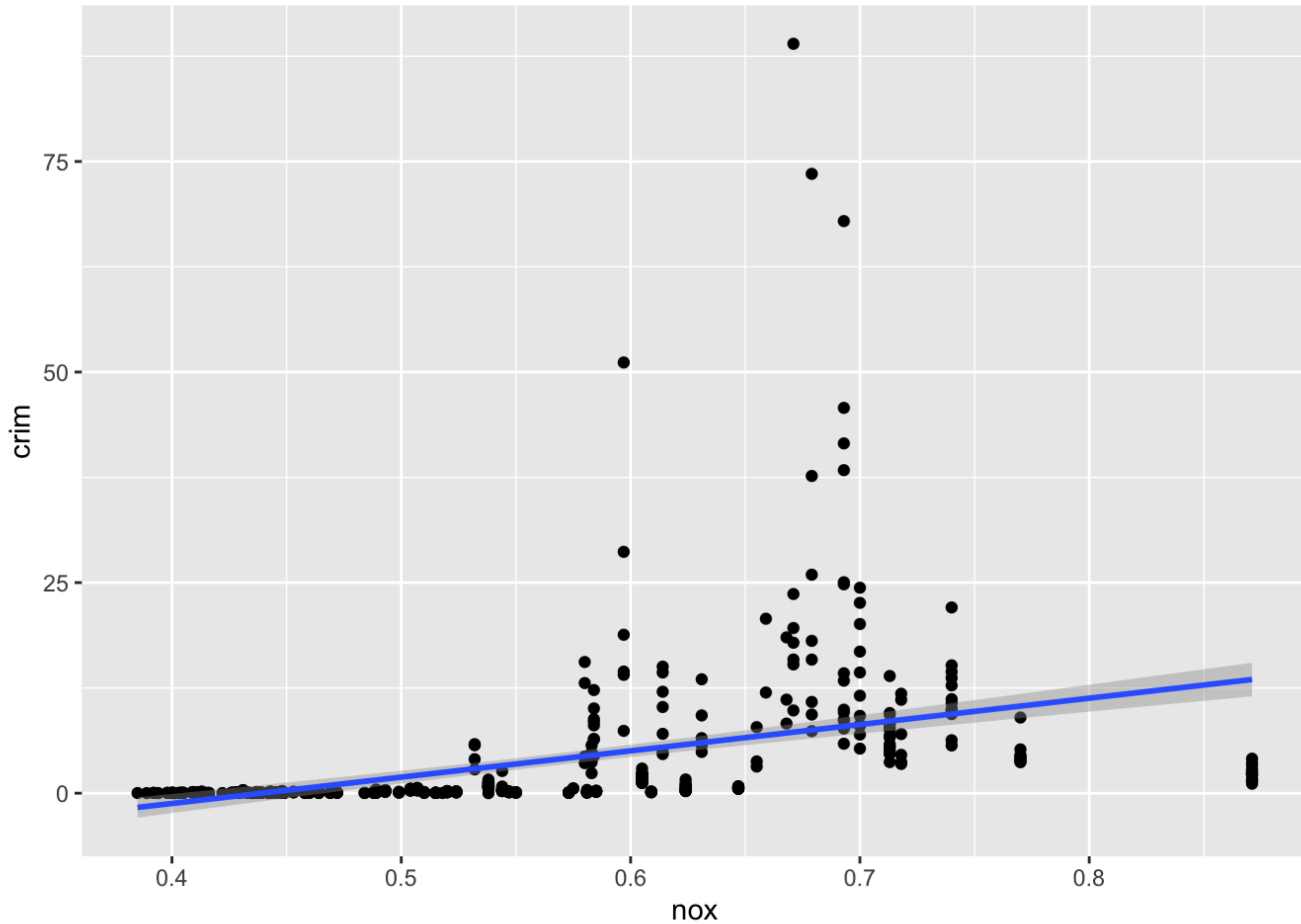


No evidence of linear relationship in dummy variable chas.

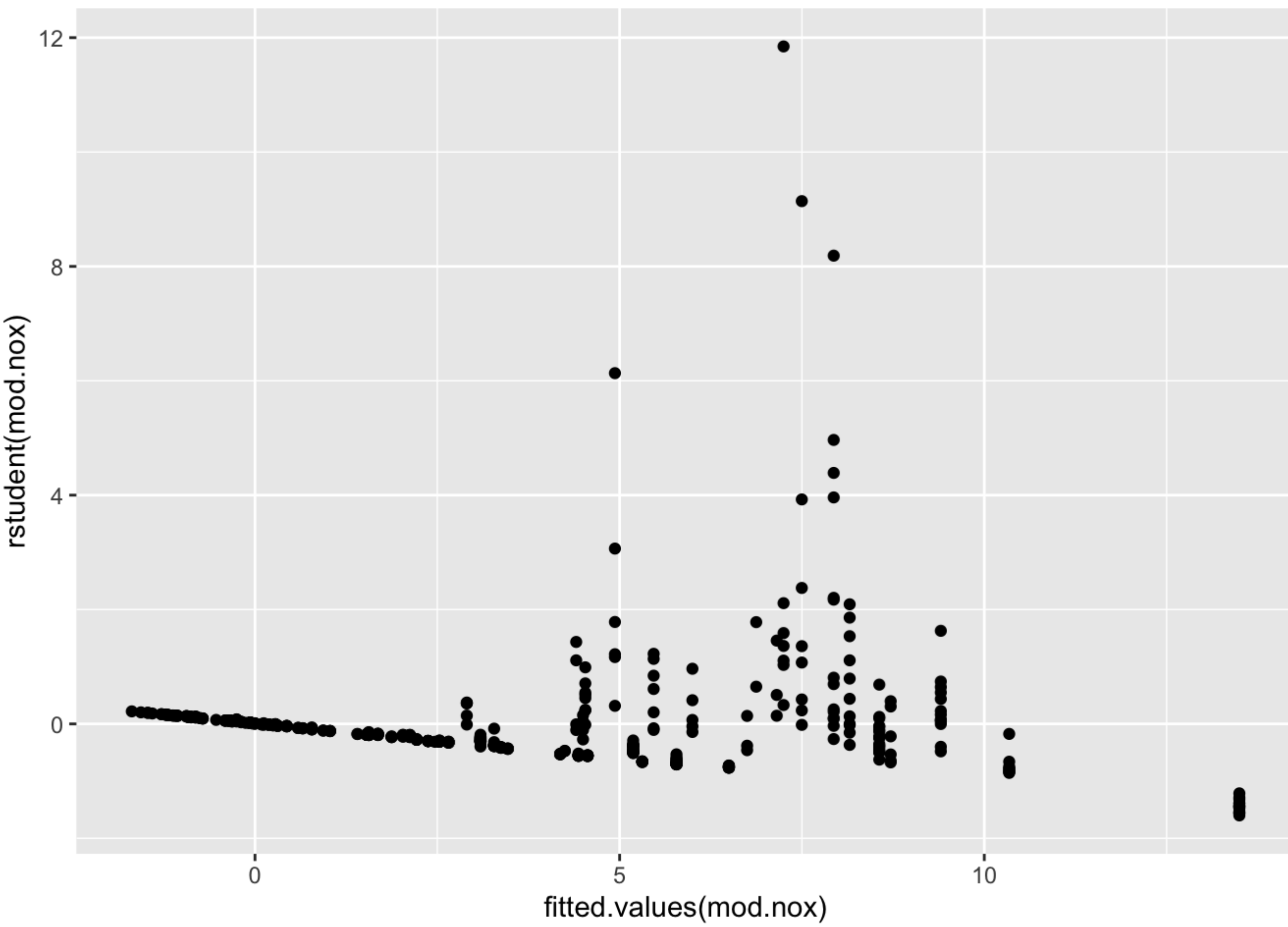
```
#nox
mod.nox = lm(crim~nox,data=Boston)
summary(mod.nox)
```

```
##
## Call:
## lm(formula = crim ~ nox, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
ggplot(Boston,aes(y=crim,x=nox)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```r
ggplot(Boston,aes(y=rstudent(mod.nox),x=fitted.values(mod.nox))) + geom_point()
```

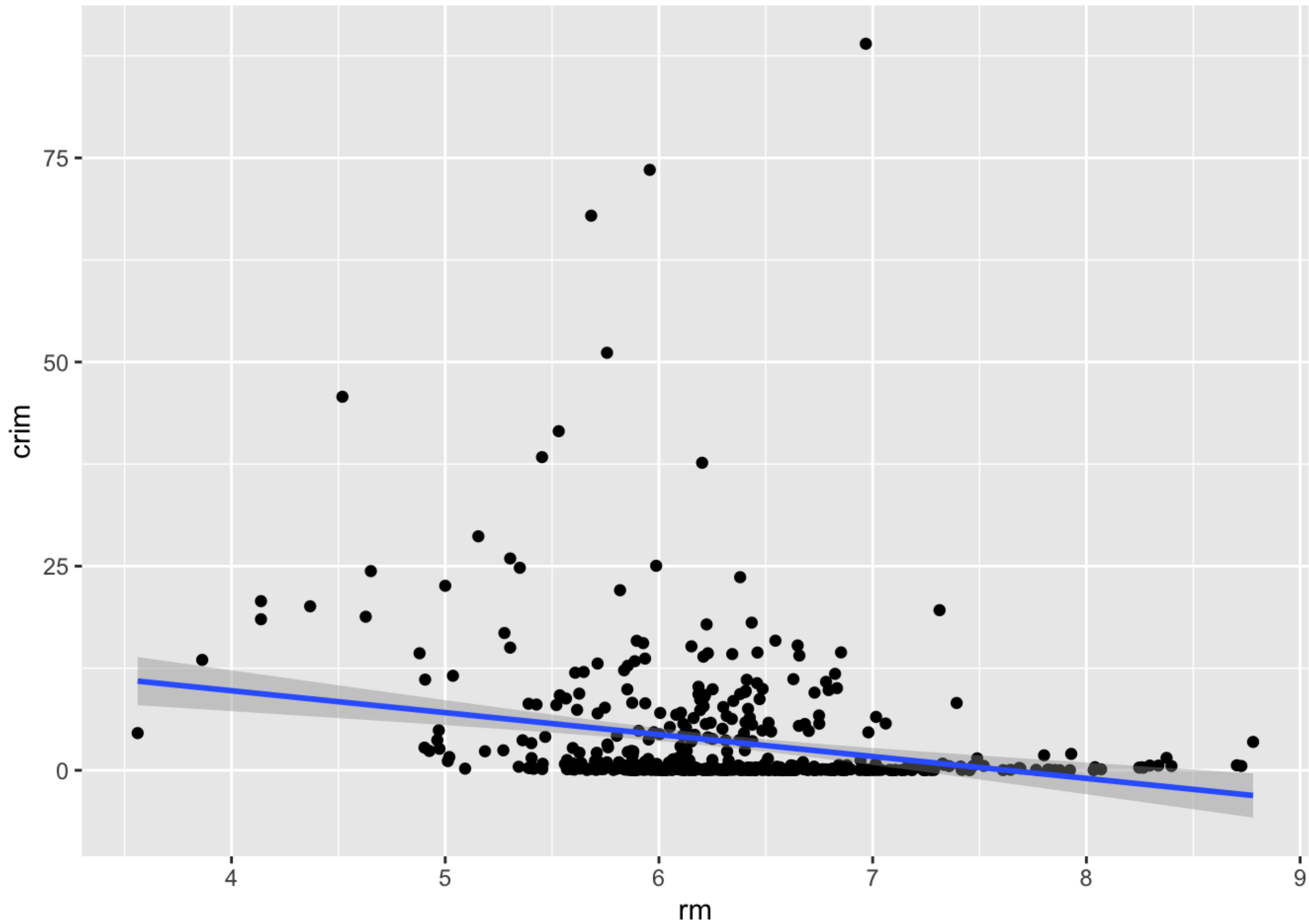


Statistically significant relationship between indus and crim p-value for slope coefficient of nox < 2e-16 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

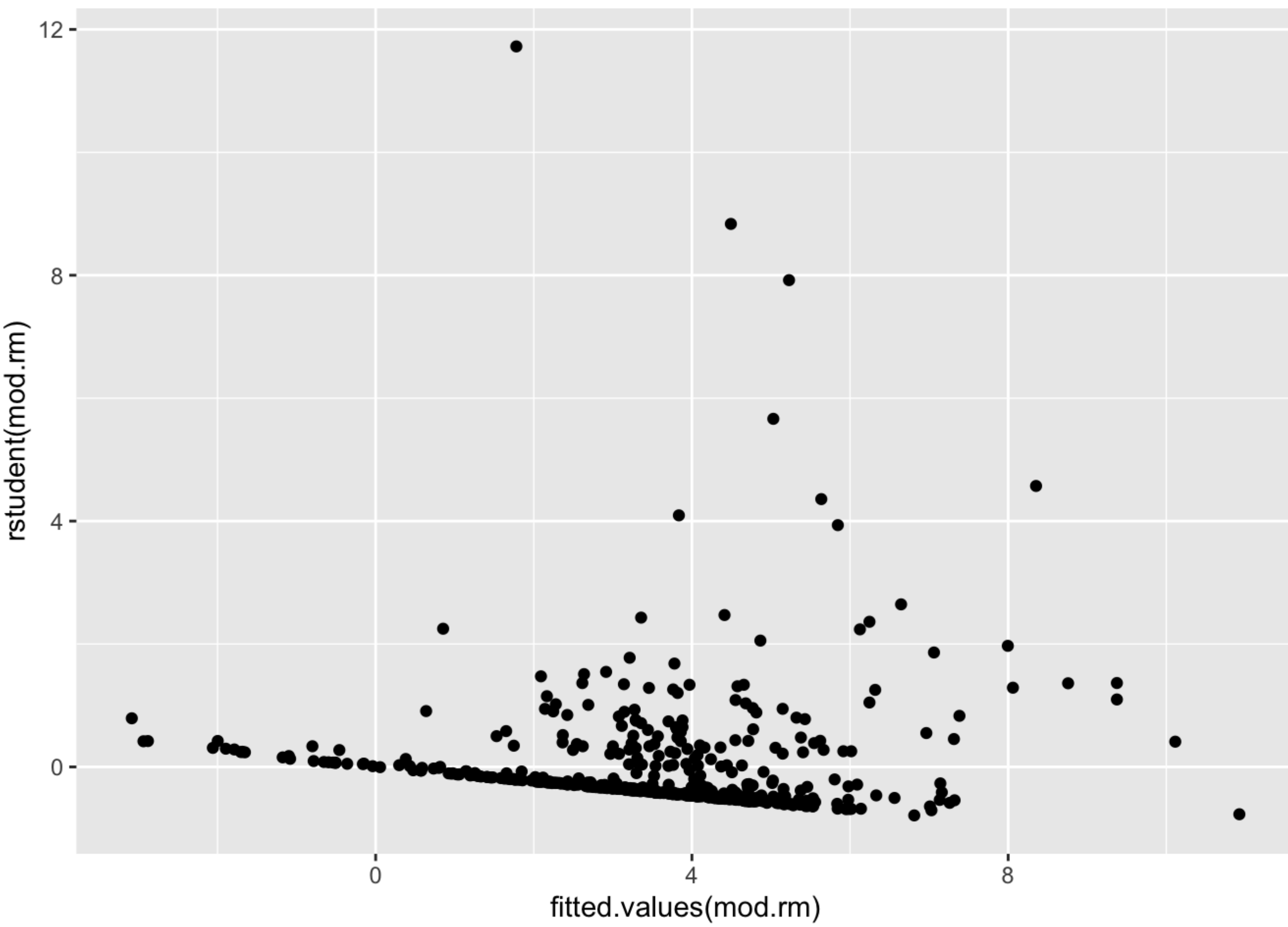
```
#rm
mod.rm = lm(crim~rm,data=Boston)
summary(mod.rm)
```

```
##
## Call:
## lm(formula = crim ~ rm, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```
ggplot(Boston,aes(y=crim,x=rm)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.rm),x=fitted.values(mod.rm))) + geom_point()
```

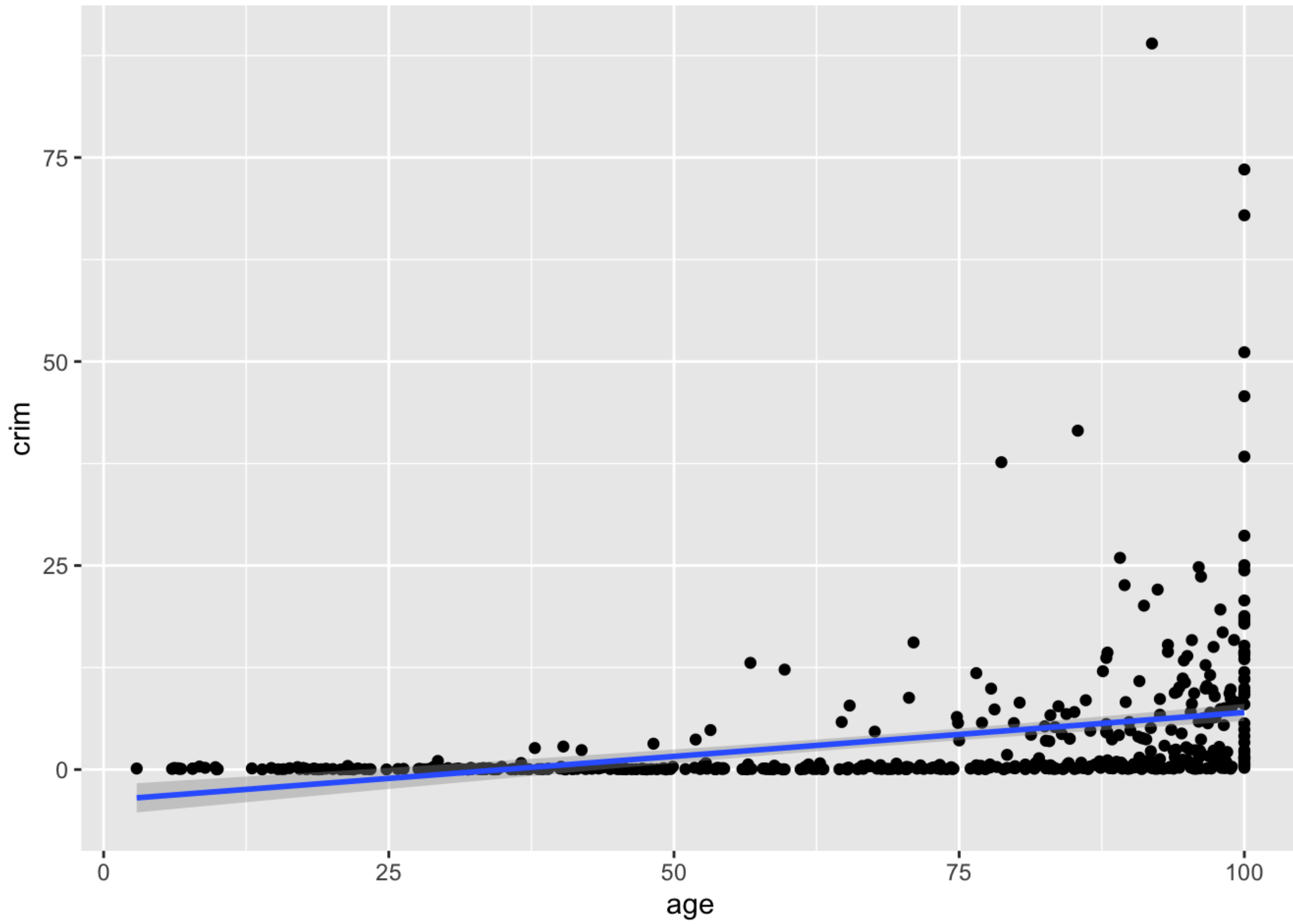


Statistically significant relationship between crim and rm p-value for slope coefficient of rm = 6.35e-07 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

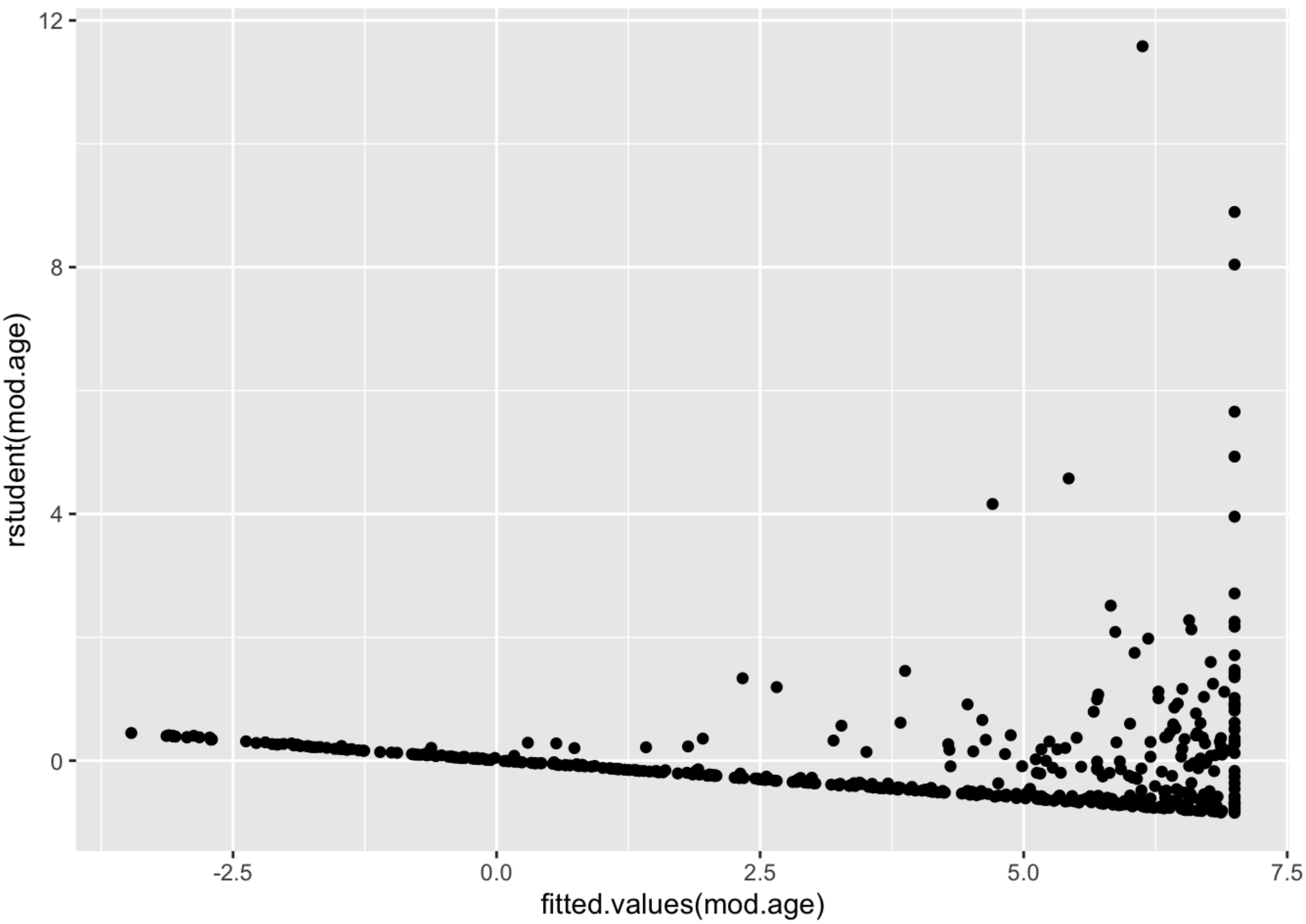
```
#age
mod.age = lm(crim~age,data=Boston)
summary(mod.age)
```

```
##
## Call:
## lm(formula = crim ~ age, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```
ggplot(Boston,aes(y=crim,x=age)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.age),x=fitted.values(mod.age))) + geom_point()
```

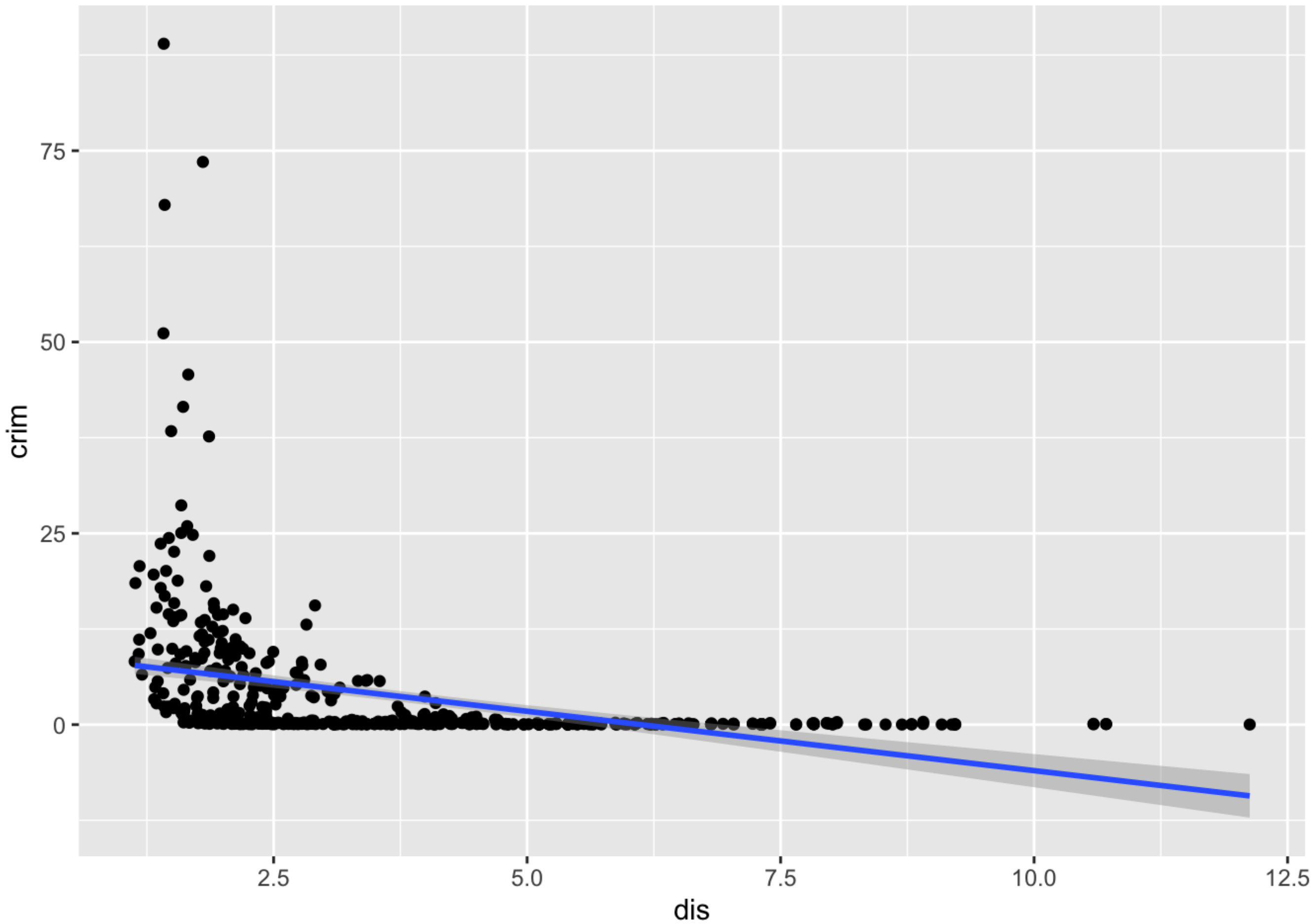


Statistically significant relationship between age and crim p-value for slope coefficient of age = 2.85e-16 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

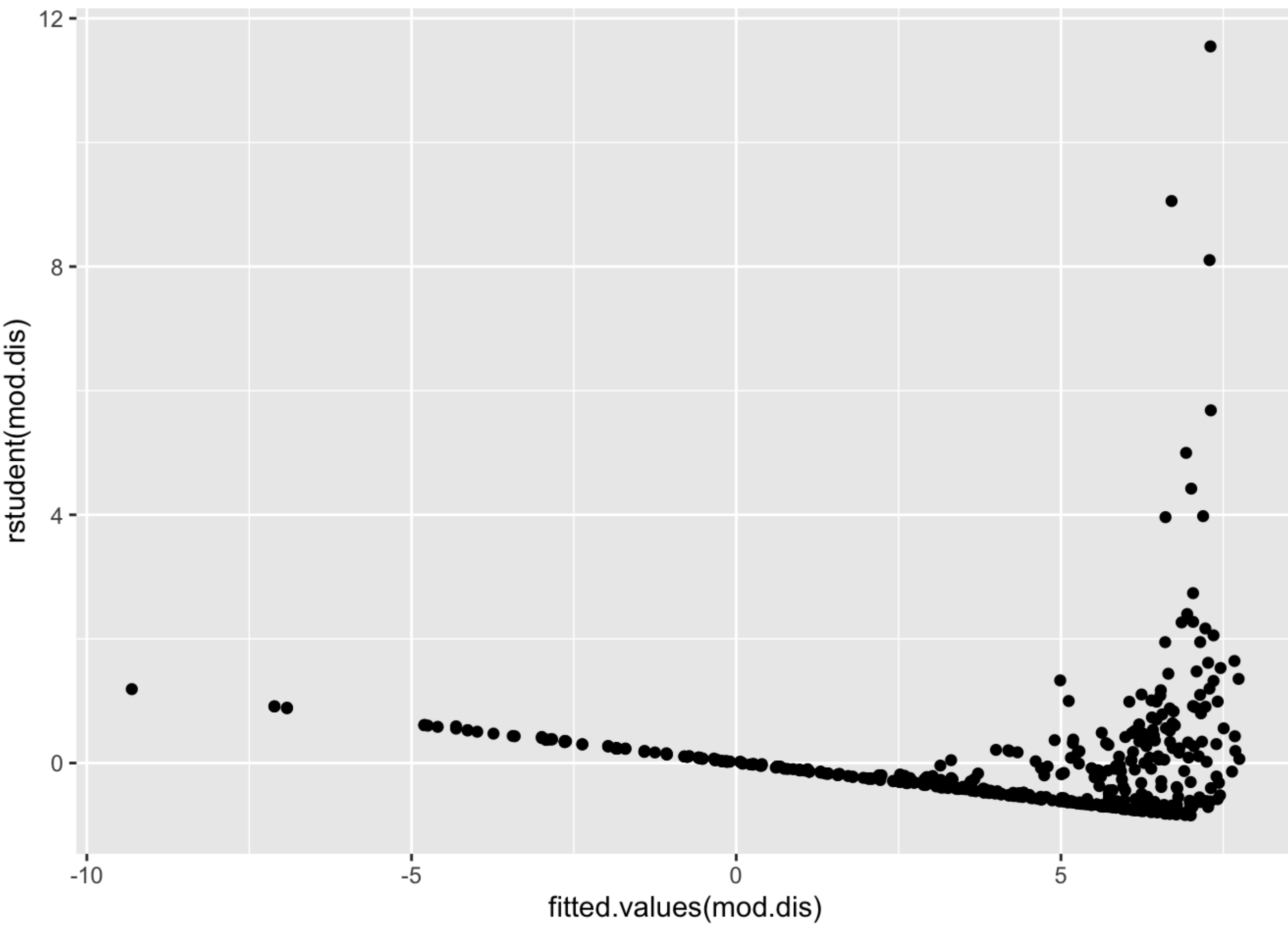
```
#dis
mod.dis = lm(crim~dis,data=Boston)
summary(mod.dis)
```

```
##
## Call:
## lm(formula = crim ~ dis, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
ggplot(Boston,aes(y=crim,x=dis)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```r
ggplot(Boston,aes(y=rstudent(mod.dis),x=fitted.values(mod.dis))) + geom_point()
```

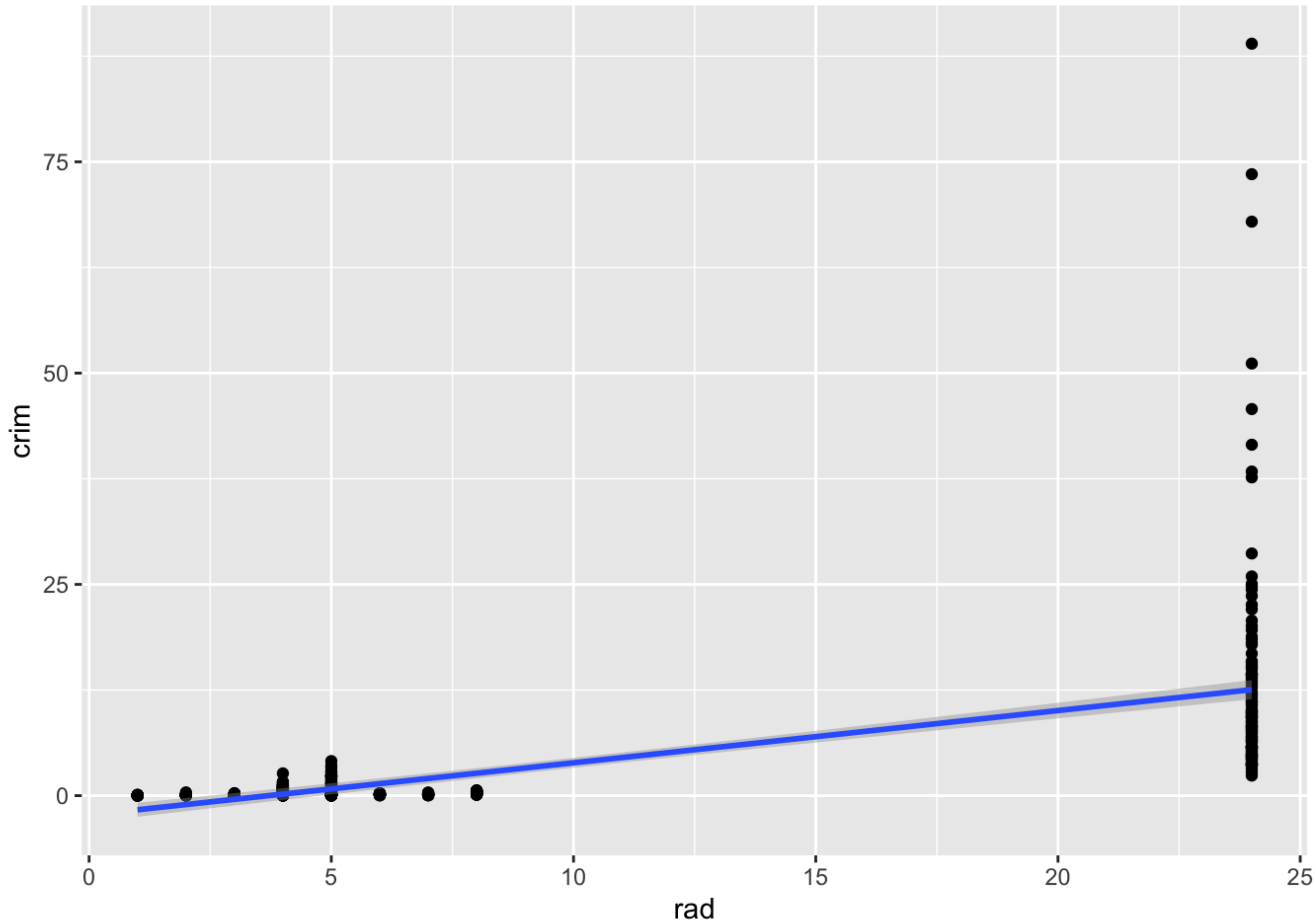


Statistically significant relationship between dis and crim p-value for slope coefficient of dis < 2e-16 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

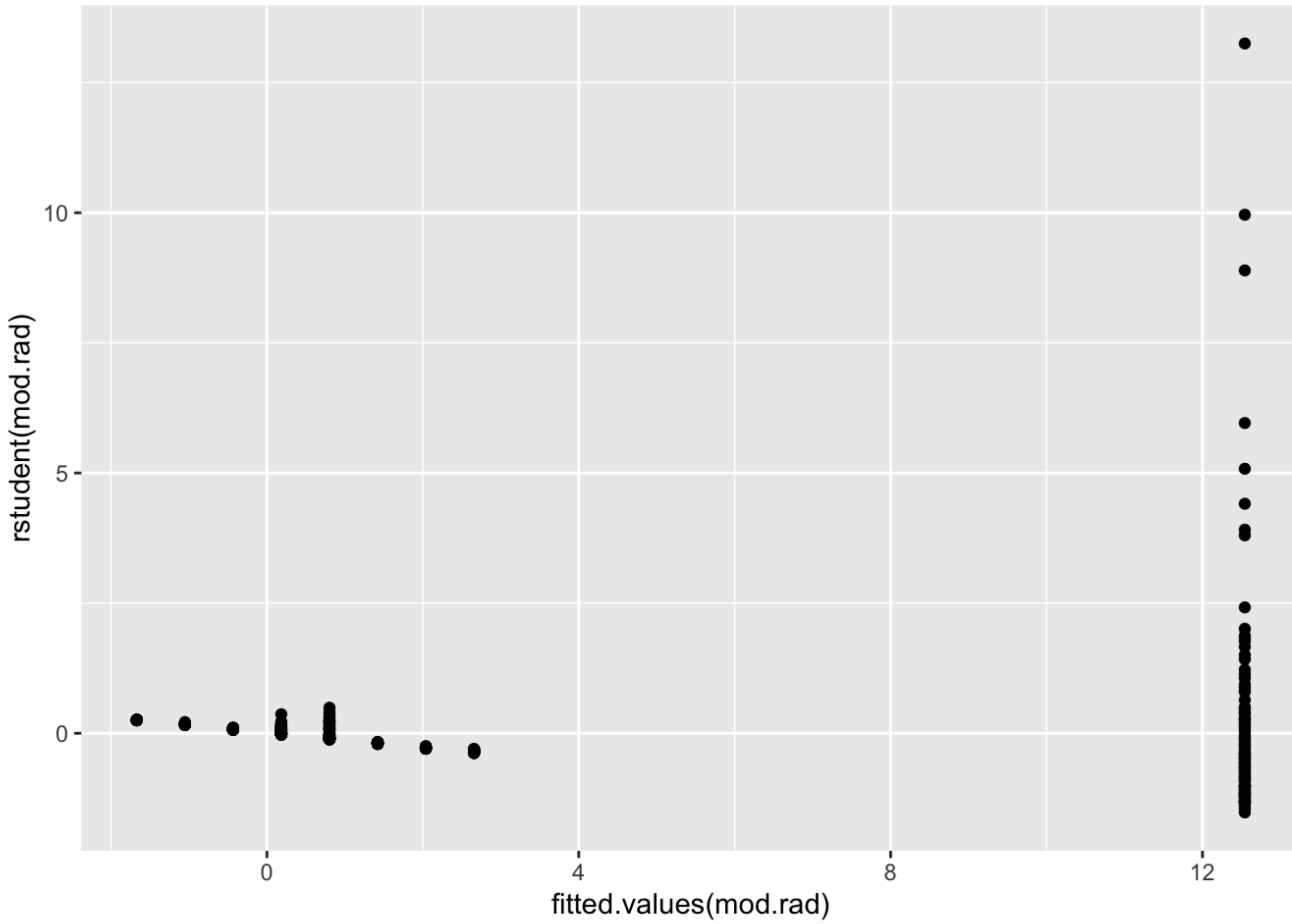
```
#rad
mod.rad = lm(crim~rad,data=Boston)
summary(mod.rad)
```

```
##
## Call:
## lm(formula = crim ~ rad, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
ggplot(Boston,aes(y=crim,x=rad)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.rad),x=fitted.values(mod.rad))) + geom_point()
```

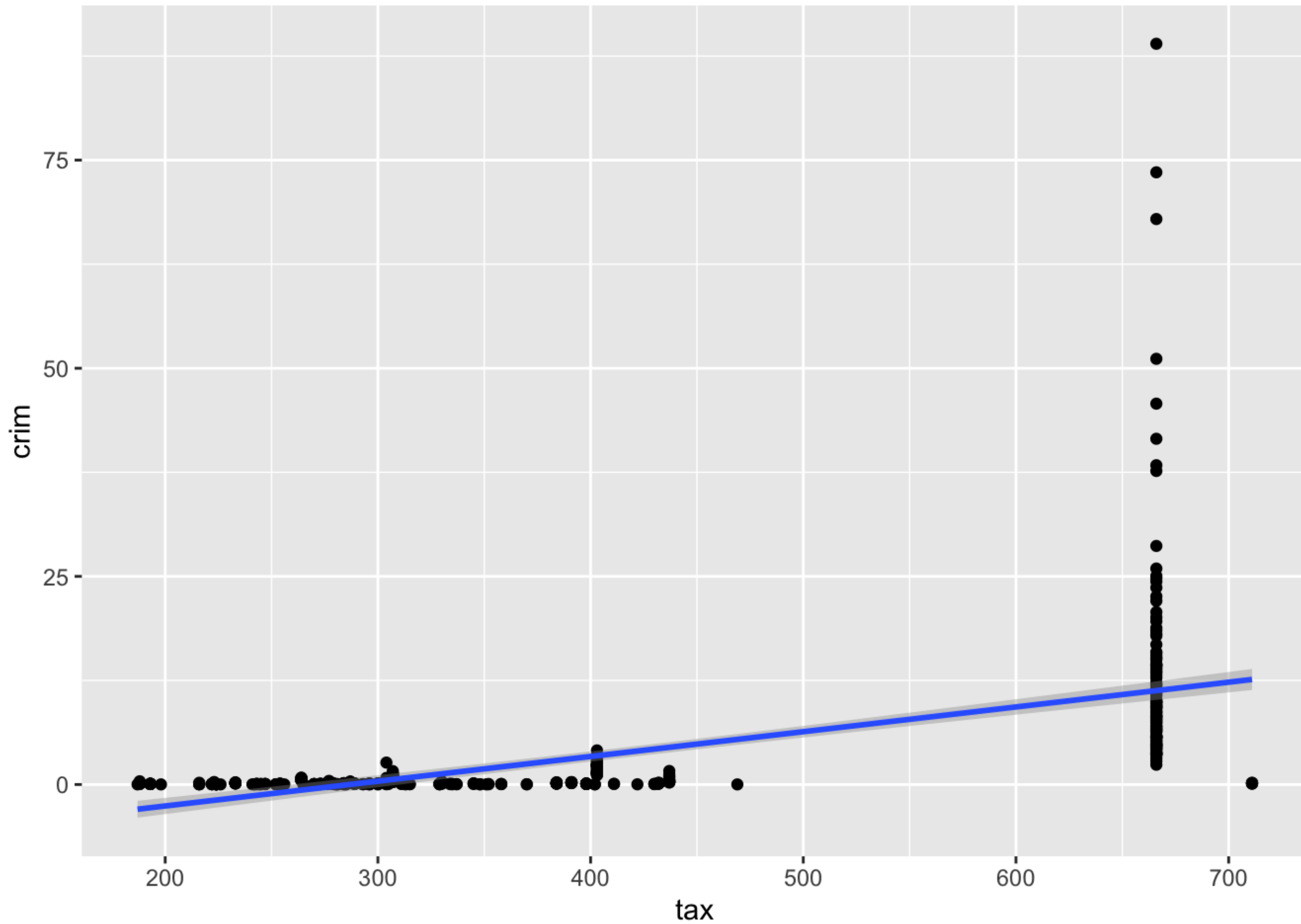


Statistically significant relationship between rad and crim p-value for slope coefficient of rad < 2e-16 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

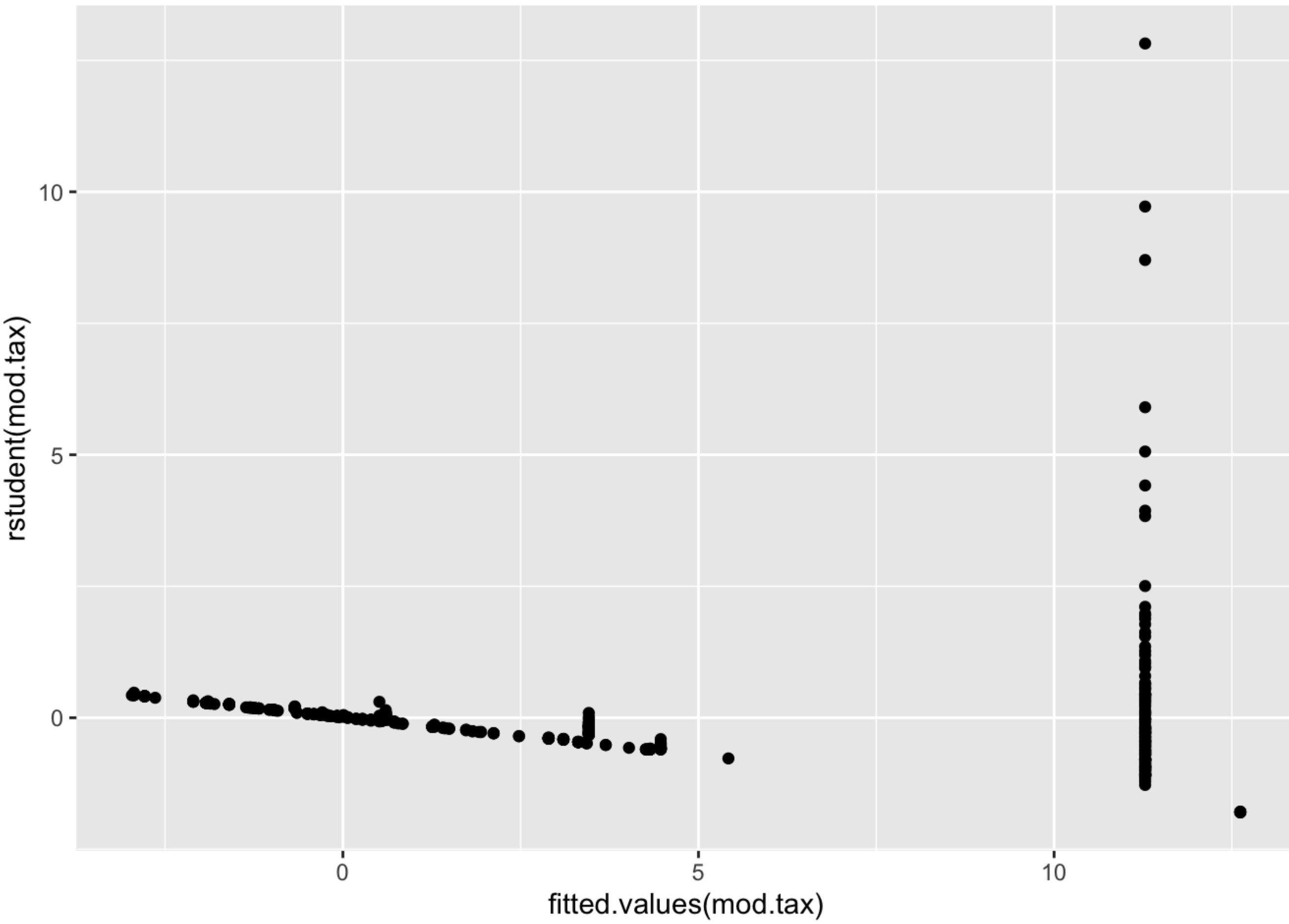
```
#tax
mod.tax = lm(crim~tax,data=Boston)
summary(mod.tax)
```

```
##
## Call:
## lm(formula = crim ~ tax, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
ggplot(Boston,aes(y=crim,x=tax)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.tax),x=fitted.values(mod.tax))) + geom_point()
```

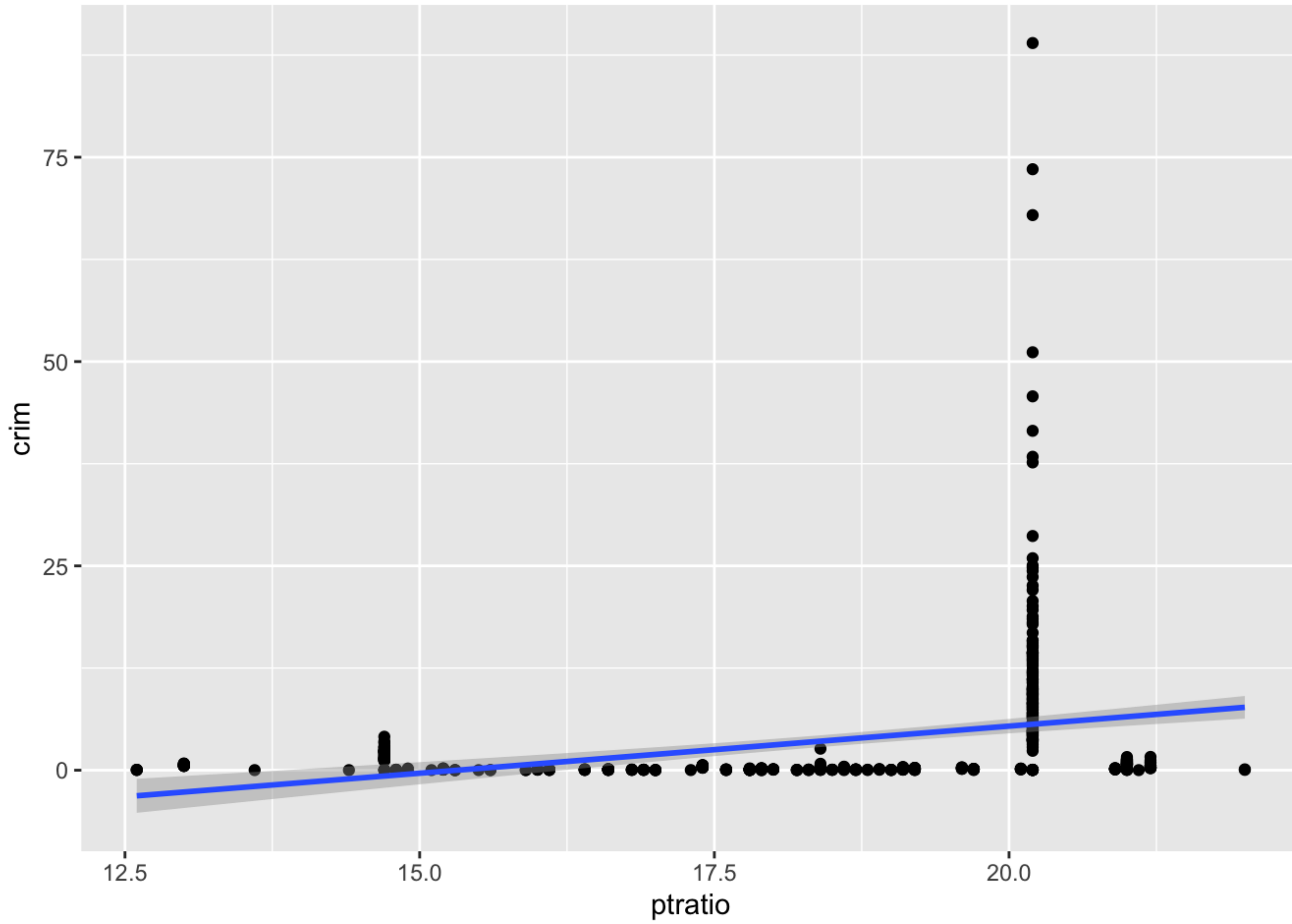


Statistically significant relationship between tax and crim p-value for slope coefficient of tax < 2e-16 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

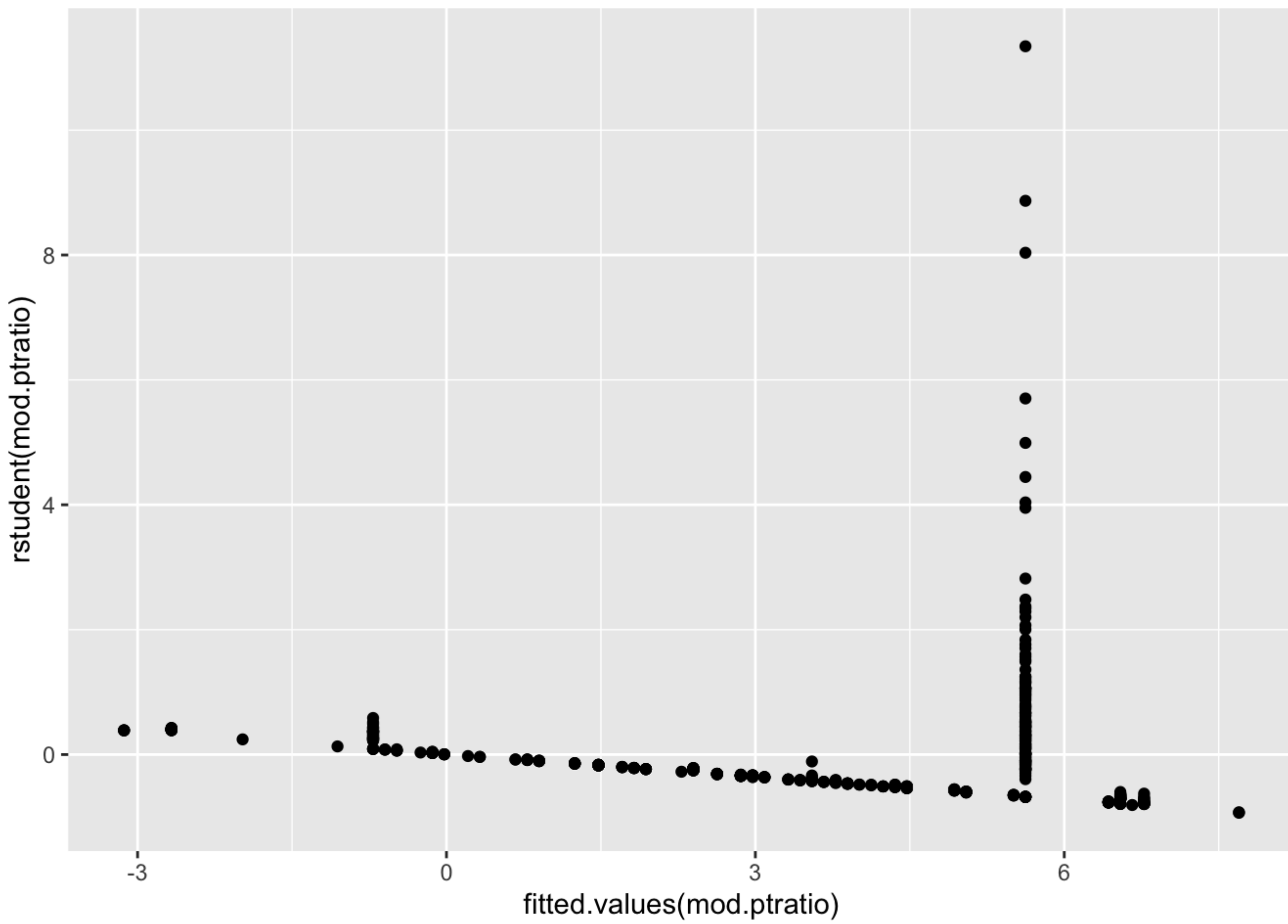
```
#ptratio
mod.ptratio = lm(crim~ptratio,data=Boston)
summary(mod.ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```
ggplot(Boston,aes(y=crim,x=ptratio)) + geom_point() + geom_smooth(method = "lm",se = TRU
E)
```




```
ggplot(Boston,aes(y=rstudent(mod.ptratio),x=fitted.values(mod.ptratio))) + geom_point()
```

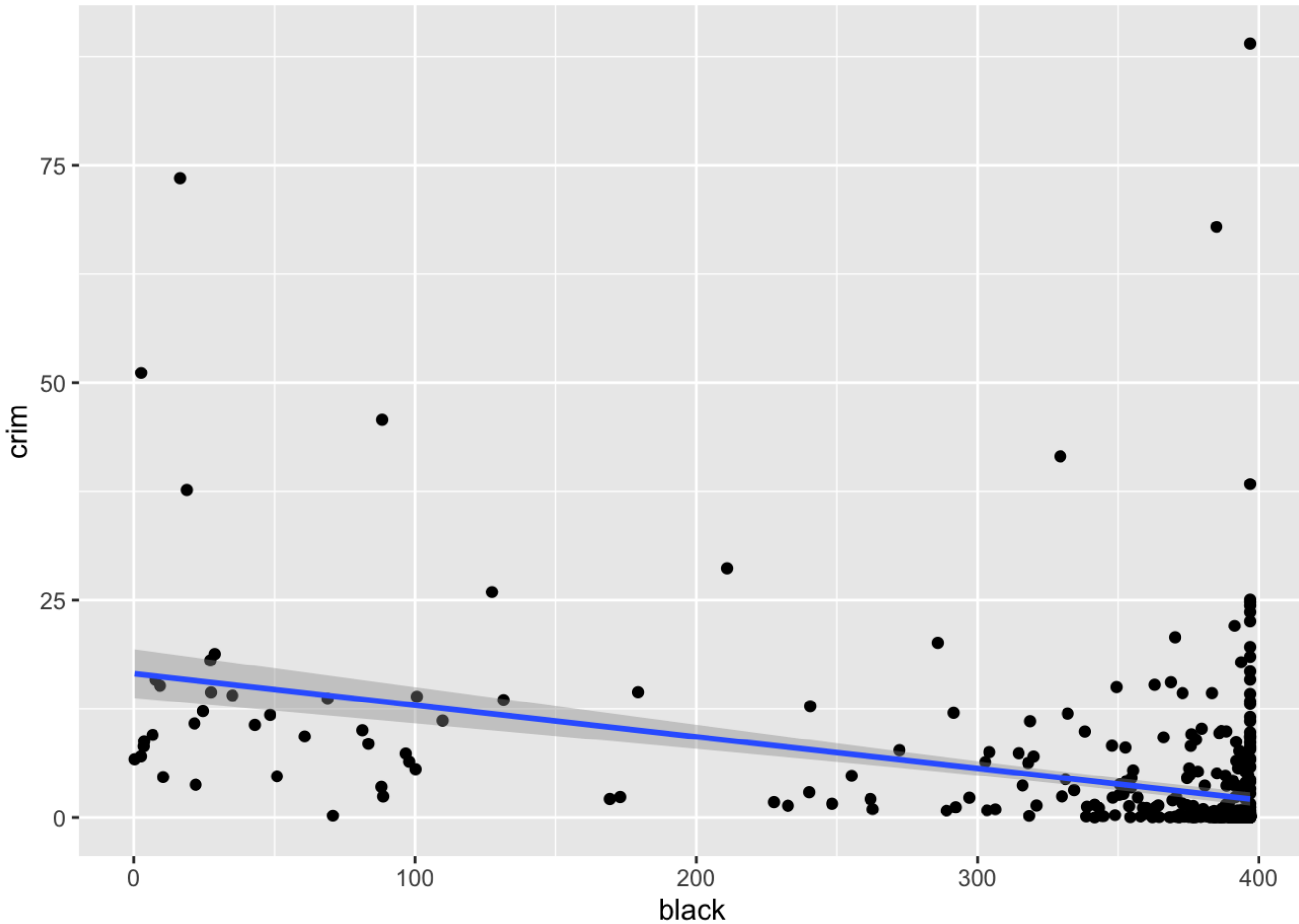


Statistically significant relationship between ptratio and crim p-value for slope coefficient of ptratio = 2.94e-11 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

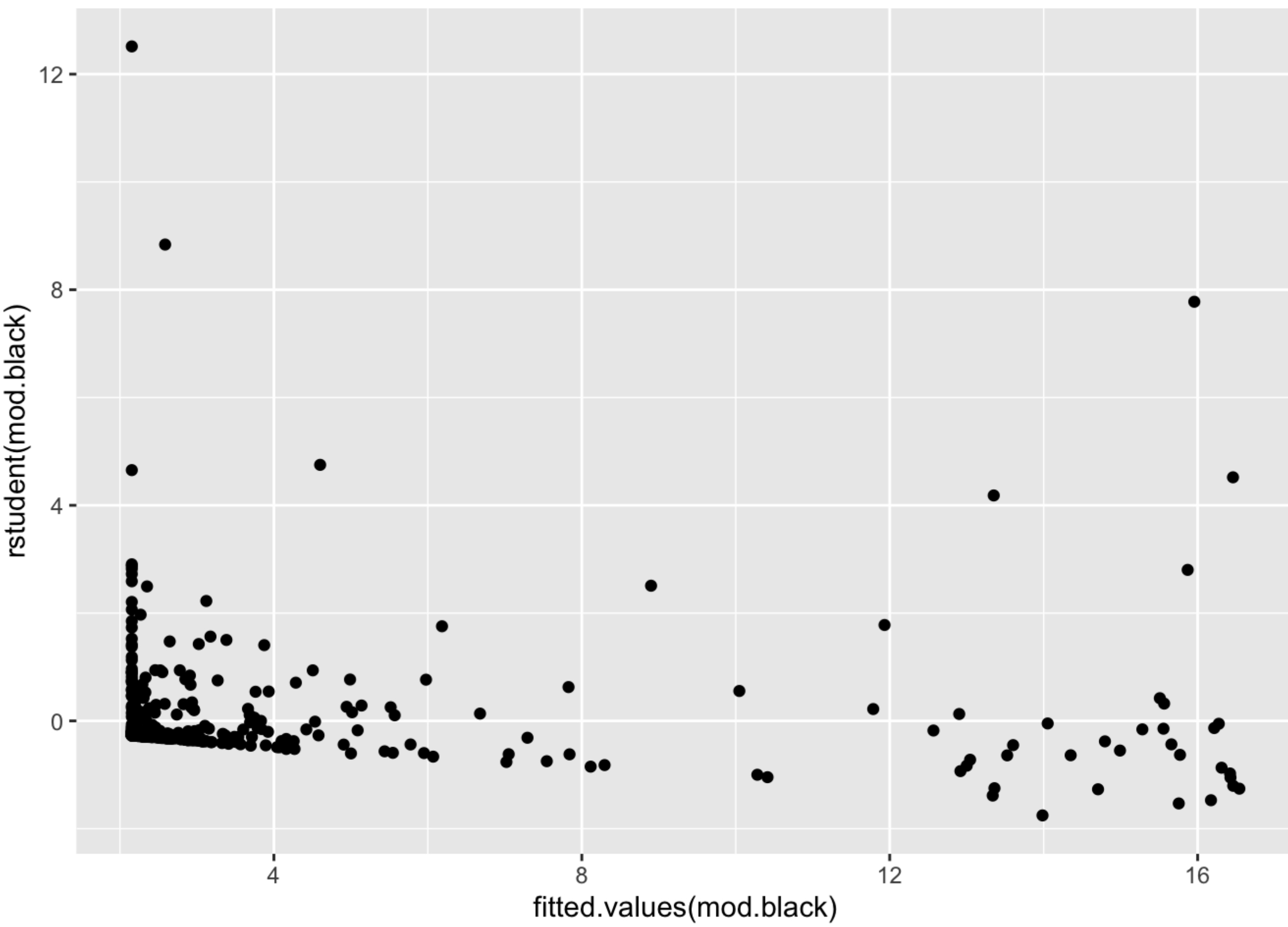
```
#black
mod.black = lm(crim~black,data=Boston)
summary(mod.black)
```

```
##
## Call:
## lm(formula = crim ~ black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
ggplot(Boston,aes(y=crim,x=black)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.black),x=fitted.values(mod.black))) + geom_point()
```

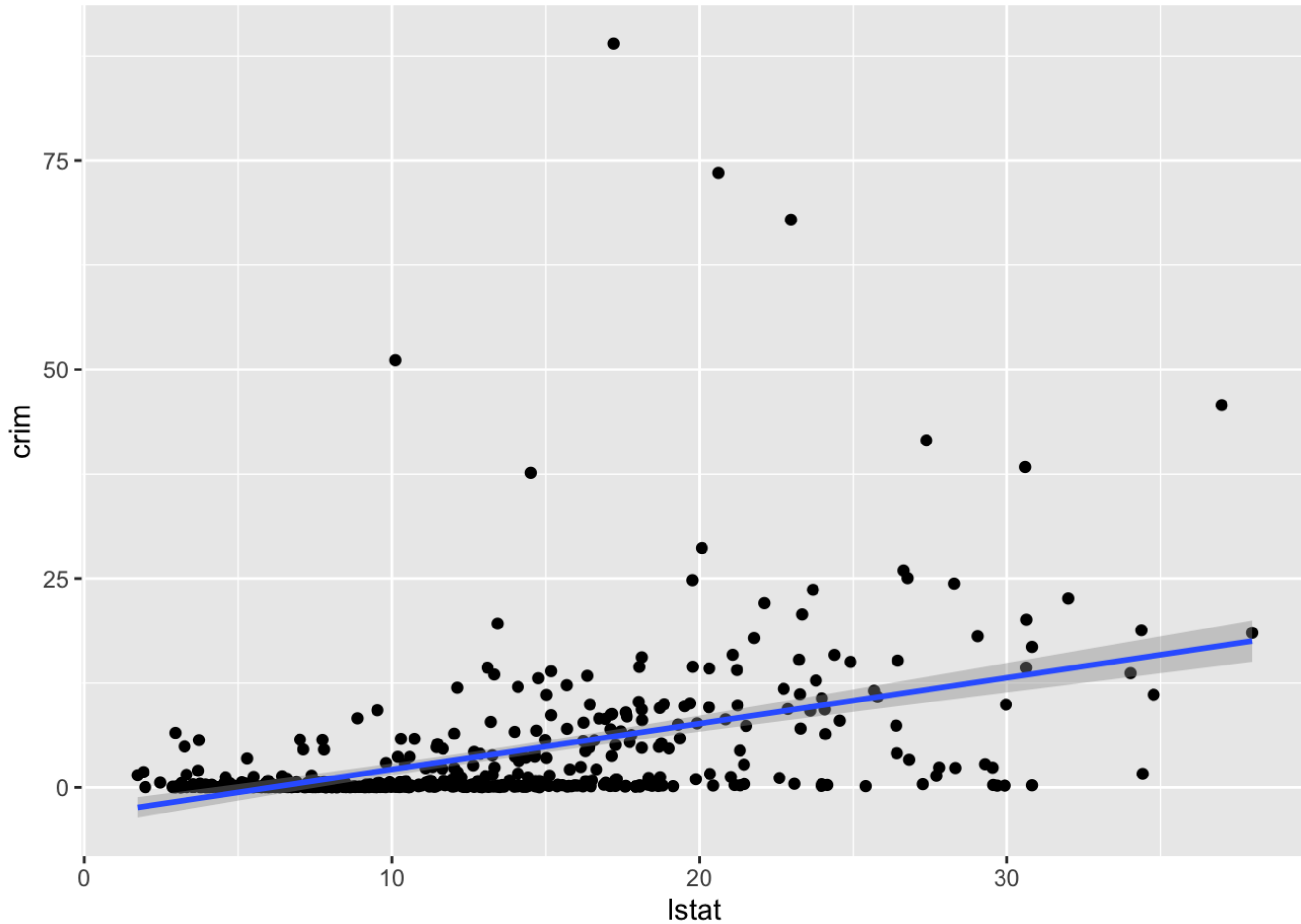


Statistically significant relationship between crim and black p-value for slope coefficient of last < 2e-16 - strong evidence of non-zero slope coefficient coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

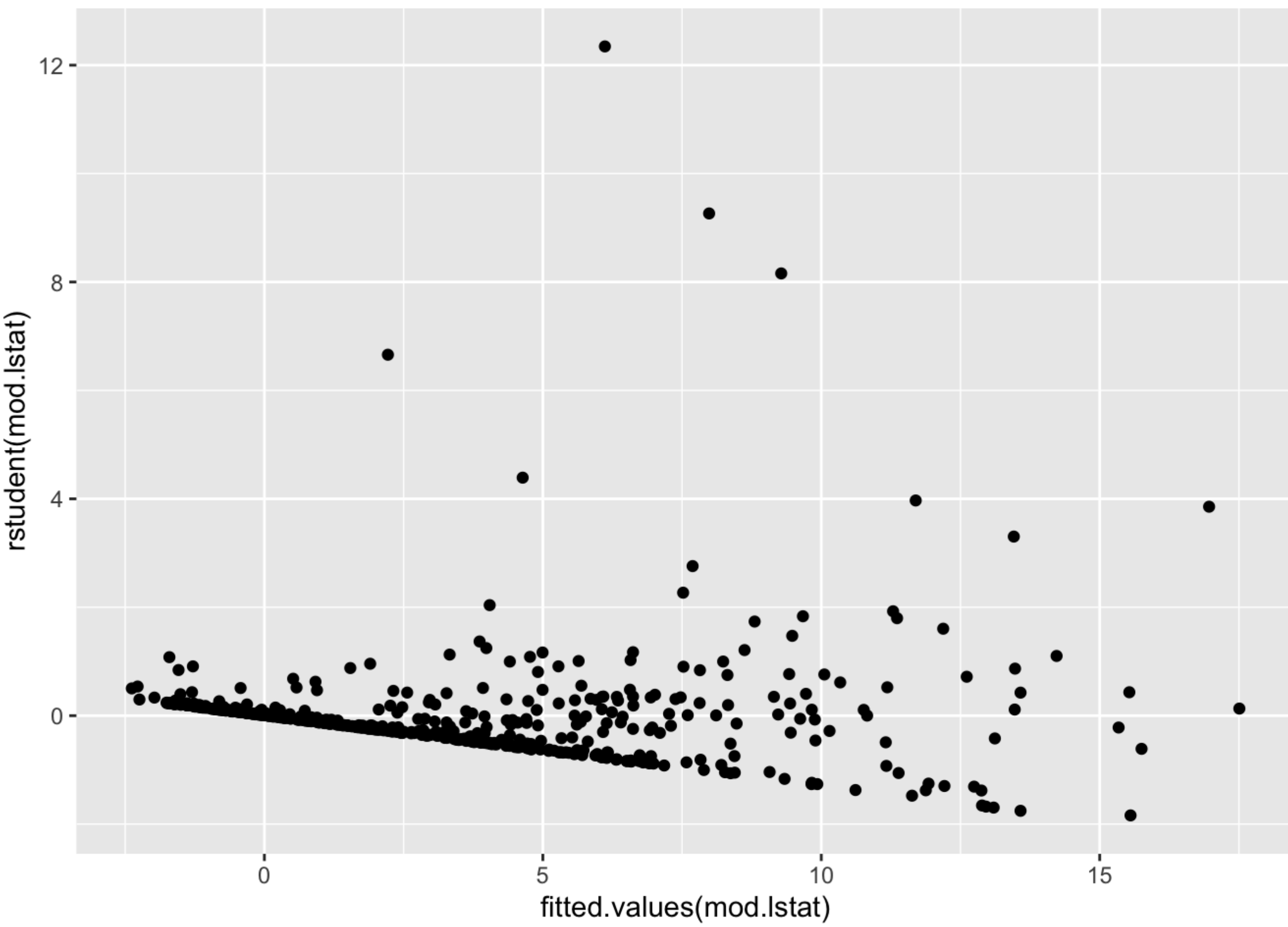
```
#lstat
mod.lstat = lm(crim~lstat,data=Boston)
summary(mod.lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
ggplot(Boston,aes(y=crim,x=lstat)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.lstat),x=fitted.values(mod.lstat))) + geom_point()
```

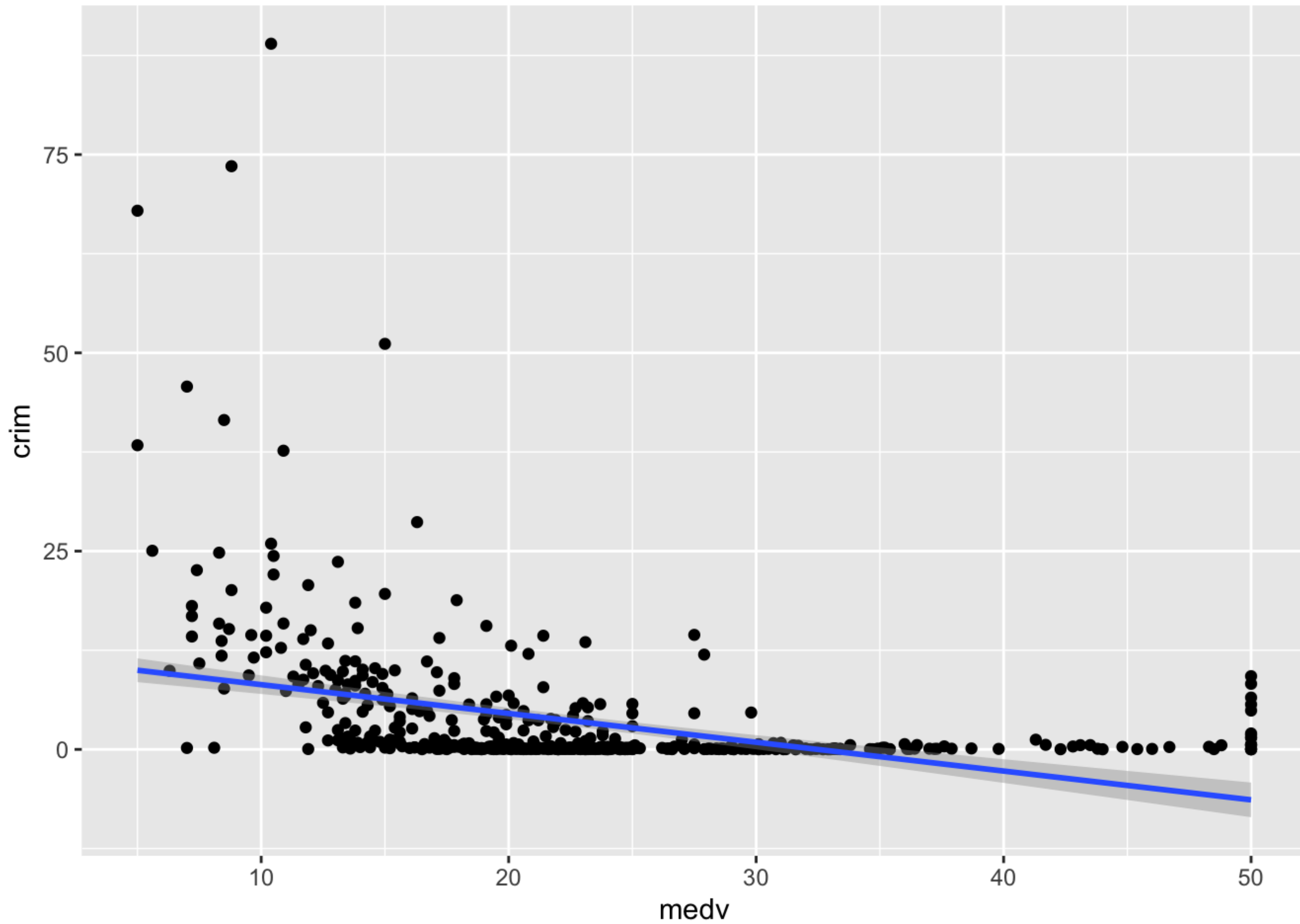


Statistically significant relationship between crim and lstat p-value for slope coefficient of lstat < 2e-16 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

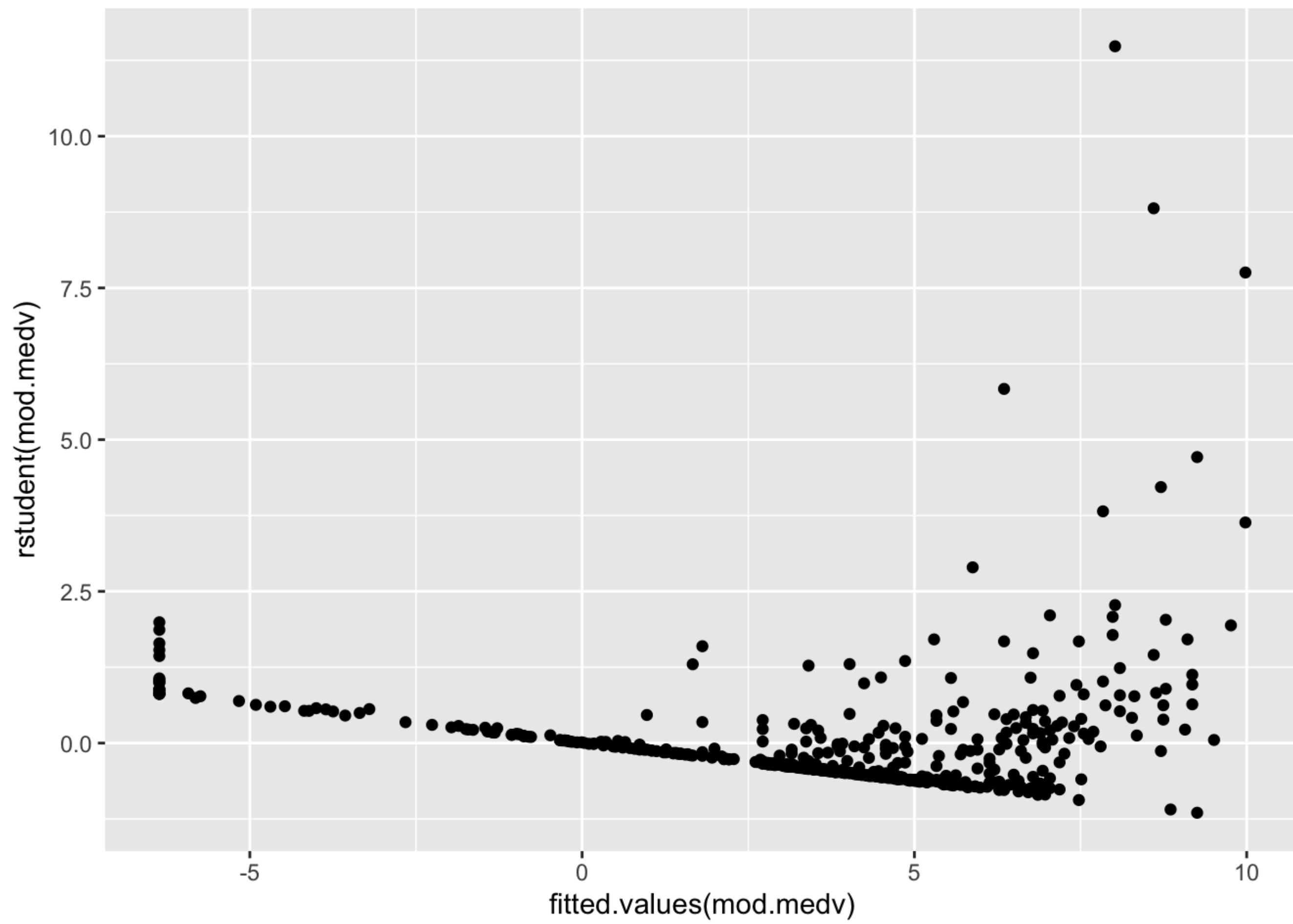
```
#medv
mod.medv = lm(crim~medv,data=Boston)
summary(mod.medv)
```

```
##
## Call:
## lm(formula = crim ~ medv, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
ggplot(Boston,aes(y=crim,x=medv)) + geom_point() + geom_smooth(method = "lm",se = TRUE)
```




```
ggplot(Boston,aes(y=rstudent(mod.medv),x=fitted.values(mod.medv))) + geom_point()
```



Statistically significant relationship between medv and lstat p-value for slope coefficient of mdev < 2e-16 - strong evidence of non-zero slope coefficient. Residual v Fitted plot shows strong evidence of non-linearity.

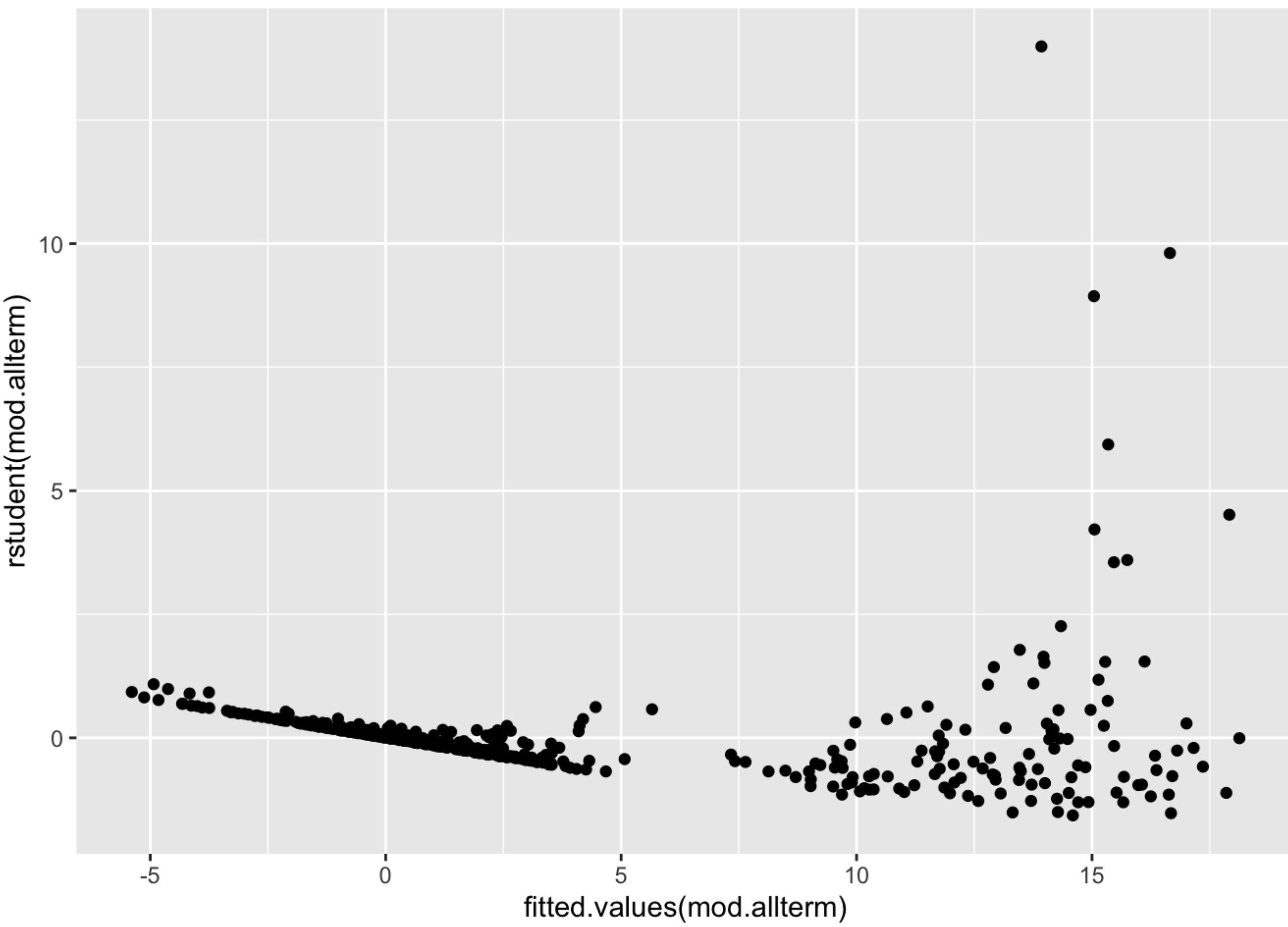
All explanatory factors besides chas have statistically significance (of non-zero slope coefficient coefficient). We note that these factors have non-linear issues in the residual v fitted plots - possibly due to many crim values around 0.

(b)Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis H0 : βj = 0?

```
mod.allterm <- lm(crim ~ . , data = Boston)
summary(mod.allterm)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
ggplot(Boston,aes(y=rstudent(mod.allterm),x=fitted.values(mod.allterm))) + geom_point()
```



At alpha = .05 we reject the null hypothesis (βj = 0) for dis, rad zn, black, and medv factors. With a p-value of < 2.2e-16 for the model (from f-statistic) we note that the model has strong statistical evidence of providing value vrs the NULL model. This is confirmed with the R^2 of .454 which means all factors explain 45.4% of the variation in crim. The residual vs fitted plot does indicate a non-linear relationship because there appears to be highly non-random scattering.

(c)How do your results from (a) compare to your results from (b)? Create a plot displaying the uni-variate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regres- sion model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

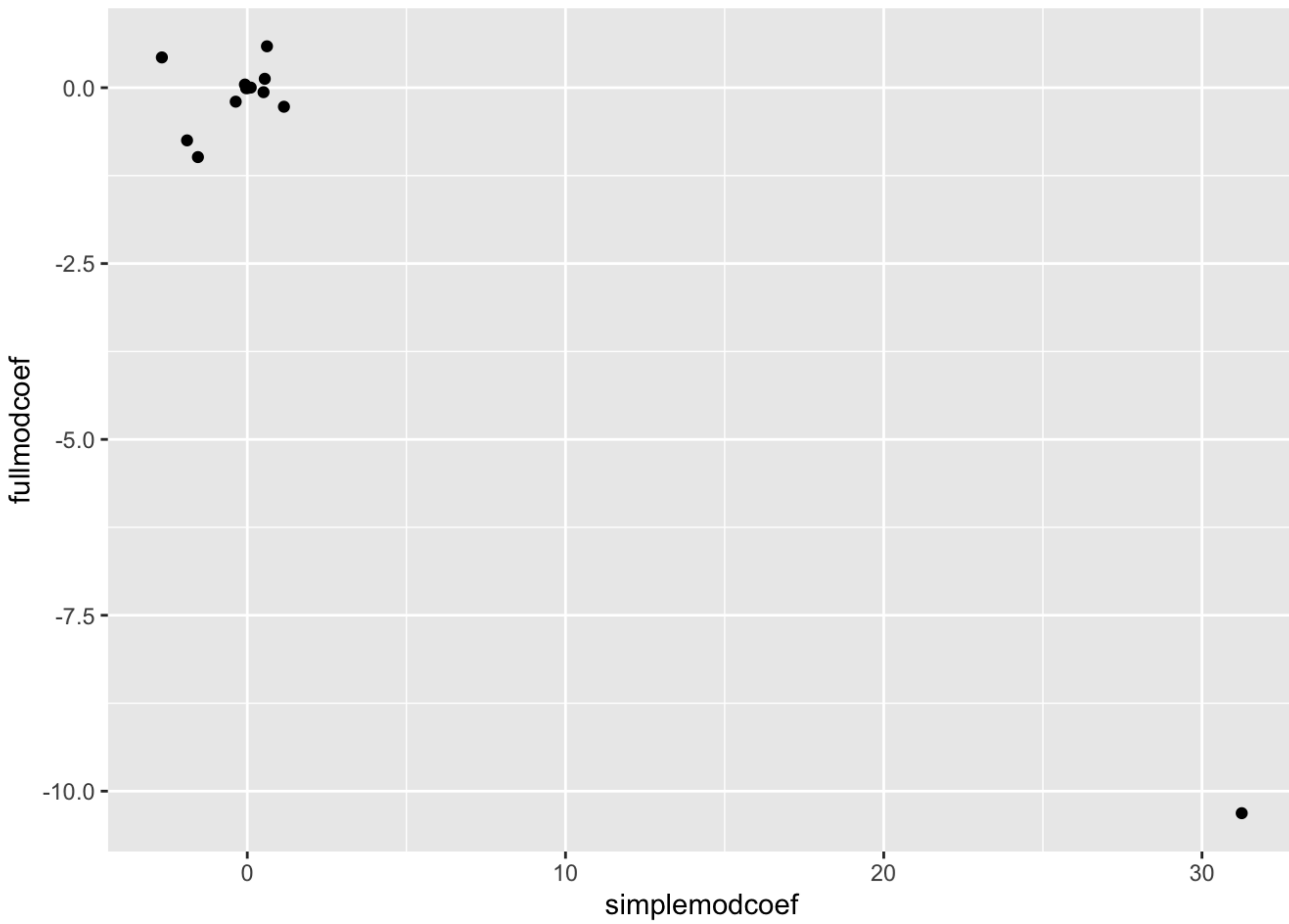
```
simplemodcoef = c(coef(mod.zn)[2],
      coef(mod.indus)[2],
      coef(mod.chas)[2],
      coef(mod.nox)[2],
      coef(mod.rm)[2],
      coef(mod.age)[2],
      coef(mod.dis)[2],
      coef(mod.rad)[2],
      coef(mod.tax)[2],
      coef(mod.ptratio)[2],
      coef(mod.black)[2],
      coef(mod.lstat)[2],
      coef(mod.medv)[2])
fullmodcoef = c(coef(mod.allterm)[2:14])

simplemodcoef
```

```
##          zn       indus        chas         nox          rm         age
## -0.07393498  0.50977633 -1.89277655 31.24853120 -2.68405122  0.10778623
##         dis         rad         tax     ptratio       black       lstat
## -1.55090168  0.61791093  0.02974225  1.15198279 -0.03627964  0.54880478
##        medv
## -0.36315992
```

```
fullmodcoef
```

```
##            zn         indus          chas           nox            rm
##   0.044855215  -0.063854824  -0.749133611 -10.313534912   0.430130506
##           age           dis           rad           tax       ptratio
##   0.001451643  -0.987175726   0.588208591  -0.003780016  -0.271080558
##         black         lstat          medv
##  -0.007537505   0.126211376  -0.198886821
```

```
ggplot(NULL,aes(x=simplemodcoef,y=fullmodcoef)) + geom_point()
```



The most significant difference is the nox coefficients which was about 30 in the simple coefficient and -10 in the full model.

(d)Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form (3rd degree polynomial)

```
nonlinBoston <- function(x) {
  form1 <- formula(paste0("crim~",x))
  fit1 <- lm(form1,data=Boston)
  form3 <- formula(paste0("crim~poly(",x,",3)"))
  fit3 <- lm(form3,data=Boston)
  print(summary(fit3))
  anova(fit1,fit3)$"Pr(>F)"[2]
}
nn <- names(Boston)
nn <- nn[-4] # remove chas
for(i in 2:length(nn)) {
  print(nn[i])
  print(nonlinBoston(nn[i]))
  print("-----")
}
```

```
## [1] "zn"
##
## Call:
## lm(formula = form3, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
##
## [1] 0.008511995
## [1] "-----"
## [1] "indus"
##
## Call:
## lm(formula = form3, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1    78.591      7.423  10.587  < 2e-16 ***
## poly(indus, 3)2   -24.395      7.423  -3.286  0.00109 **
## poly(indus, 3)3   -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] 8.408754e-14
## [1] "-----"
## [1] "nox"
##
## Call:
## lm(formula = form3, data = Boston)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max 
## -9.110 -2.068 -0.255  0.739 78.302 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928 
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
## 
## [1] 7.122383e-18
## [1] "-----"
## [1] "rm"
## 
## Call:
## lm(formula = form3, data = Boston)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -18.485  -3.468  -2.221  -0.015  87.219 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 ** 
## poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222 
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
## 
## [1] 0.005229427
## [1] "-----"
## [1] "age"
## 
## Call:
## lm(formula = form3, data = Boston)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -9.762 -2.673 -0.516  0.019 82.842 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)     3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] 4.125056e-07
## [1] "-----"
## [1] "dis"
##
## Call:
## lm(formula = form3, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] 3.071837e-19
## [1] "-----"
## [1] "rad"
##
## Call:
## lm(formula = form3, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] 0.02607832
## [1] "-----"
## [1] "tax"
##
## Call:
## lm(formula = form3, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] 1.144238e-05
## [1] "-----"
## [1] "ptratio"
##
## Call:
## lm(formula = form3, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2   24.775      8.122   3.050  0.00241 **
## poly(ptratio, 3)3  -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
##
## [1] 0.0002541647
```

```
## [1] "-----"
## [1] "black"
##
## Call:
## lm(formula = form3, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3536  10.218   <2e-16 ***
## poly(black, 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
## poly(black, 3)2   5.9264     7.9546   0.745    0.457
## poly(black, 3)3  -4.8346     7.9546  -0.608    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] 0.6301501
## [1] "-----"
## [1] "lstat"
##
## Call:
## lm(formula = form3, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] 0.03698322
## [1] "-----"
## [1] "medv"
##
## Call:
## lm(formula = form3, data = Boston)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -24.427  -1.976  -0.437   0.439  73.655 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167 
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
## 
## [1] 2.504778e-42
## [1] "-----"
```

It appears that all predictors besides black have a non-linear trend. We note that all models except black have values from f-test <.05, this means there is evidence that the additional polynomial terms have non-zero slope and add ‘value’ to the model. We also see that in all models besides crim ~ black have statistically significant slope coefficients for some or all polynomial terms.

Question 2 (Chapter 4, #4)

When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that non-parametric approaches often perform poorly when p is large. We will now investigate this curse.

(a)Suppose that we have a set of observations, each with measurements on p = 1 feature, X. We assume that X is uniformly (evenly) distributed on [0,1]. Associated with each observation is a response value. Suppose that we wish to predict a test obser- vation’s response using only observations that are within 10 % of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with X = 0.6, we will use observations in the range [0.55,0.65]. On average, what fraction of the available observations will we use to make the prediction?

We would expect about 10% or .1 of available overvaluations to be used for prediction.

b. Now suppose that we have a set of observations, each with measurements on p = 2 features, X1 and X2. We assume that (X1,X2) are uniformly distributed on [0,1]×[0,1]. We wish to predict a test observation’s response using only observations that are within 10 % of the range of X1 and within 10 % of the range of X2 closest to that test observation. For instance, in order to predict the response for a test observation with X1 = 0.6 and X2 = 0.35, we will use observations in the range [0.55, 0.65] for X1 and in the range [0.3, 0.4] for X2. On average, what fraction of the available observations will we use to make the prediction?

We would expect 10% or .1 of both <x1,x2> available observations to be used. If we consider <x1,x2> as a 10X10 grid with only 1 square that fits both .1 of x1 and .1 of x2 observations we see that only .1*.1 = .01 of total available observations will be used for prediction

c. Now suppose that we have a set of observations on p = 100 features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations within the 10 % of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?

Like part (b) we now imagine a hyper-cube with .1 of each axis available for observations. This means, on average, that only .1^100 of the total observation space would be used for predictions.

d. Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations "near" any given test observation.

We see that as p features increase the nearest observations available decrease exponentially at a fixed range for a given feature P(i).

(e)Now suppose that we wish to make a prediction for a test observation by creating a p-dimensional hyper-cube centered around the test observation that contains, on average, 10 % of the train- ING observations. For p = 1,2, and 100, what is the length of each side of the hyper-cube? Comment on your answer.

for p = 1 the length of the side is .1 for p = 2 the area associated with the square is a*a = .1 (where a is the length of each side) solving for the length of each side a

```
a <- .1^(1/2)
a
```

```
## [1] 0.3162278
```

for p = 100 the area associated with the hyper-cube b^100 = .1 Solving for the length of each side (b) of the hyper-cube:

```
b <- .1^(1/100)
b
```

```
## [1] 0.9772372
```

This leads to quite a large space in larger dimensions which means the nearest observations may not be very good for prediction because in reality they are not all that near.

Question 3 (Chapter 4, #10 parts (a)-(h), 9 marks)

```
library(ISLR)
data(Weekly)
head(Weekly)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
```

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a)Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

```
summary(Weekly)
```

```
##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202
##  Median :  0.2380   Median :  0.2340   Median :1.00268
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
##      Today          Direction
##  Min.   :-18.1950   Down:484
##  1st Qu.: -1.1540   Up  :605
##  Median :  0.2410
##  Mean   :  0.1499
##  3rd Qu.:  1.4050
##  Max.   : 12.0260
```

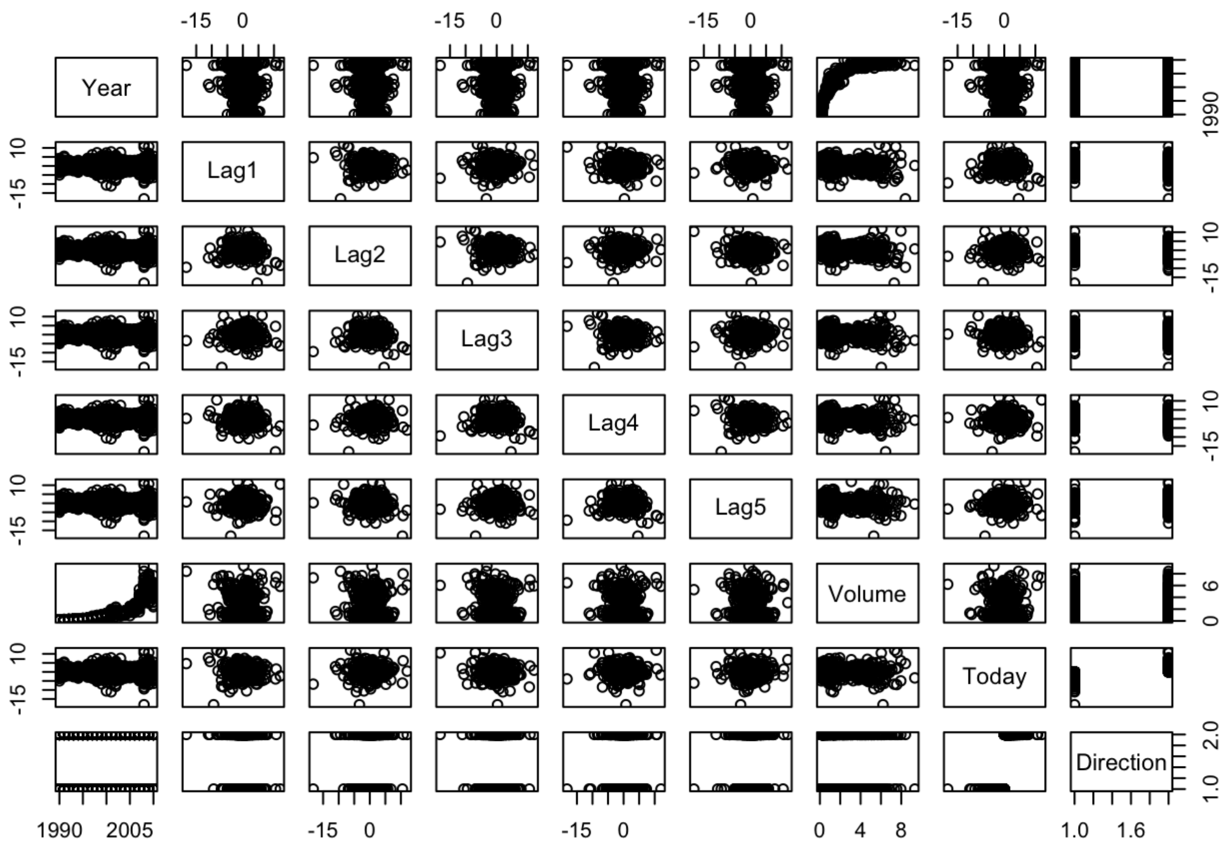
```
summary(Weekly$Direction)
```

```
## Down   Up
##  484  605
```

```
#factor dirction not numerical
cor(Weekly[,-9])
```

```
##                Year         Lag1         Lag2         Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5       Volume         Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```

```
plot(Weekly)
```



Volume and Year have a correlation of 0.84194162. We notice that Direction is the only Boolean variable. Nothing other patters are easily detected.

(b)Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
logmod = glm(Direction ~ . - Today -Year, family = binomial, data = Weekly)
summary(logmod)
```

```
##
## Call:
## glm(formula = Direction ~ . - Today - Year, family = binomial,
##     data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Only Lag2 appears to be statistically significant with p-value = 0.0296 < alpha = .05.

(c)Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
logmod.preds <- predict(logmod)
modpredict=rep("Down",1089)
modpredict[logmod.preds >.5]="Up"

table(modpredict,Weekly$Direction)
```

```
##
## modpredict Down  Up
##       Down  465 563
##       Up     19  42
```

```
mean(modpredict == Weekly$Direction)
```

```
## [1] 0.4655647
```

```
specificity <- 54/(54+430)
sensitivity <- 557/(557+48)
specificity
```

```
## [1] 0.1115702
```

```
sensitivity
```

```
## [1] 0.9206612
```

The confusion matrix seems to indicate the model correctly predicts the weekly tend in the market 56.1% of the time. However it seems that model had many false positives (when the model predicted up but the true result was down). This results in a poor specificity of .11157 compared to a good sensitivity of .9207.

d. Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
d.training <- subset.data.frame(Weekly,Year<2009)
d.test <- subset.data.frame(Weekly,Year > 2008)

d.mod <- glm(Direction ~ Lag2, data = d.training,family = binomial)

d.probs <-  predict.glm(d.mod,newdata = d.test,type = "response")
d.preds <- rep("Down",length(d.probs))
d.preds[d.probs>.5] = "Up"
table(d.preds,d.test$Direction)
```

```
##
## d.preds Down Up
##    Down    9  5
##    Up     34 56
```

```
mean(d.preds== d.test$Direction)
```

```
## [1] 0.625
```

e. Repeat (d) using LDA.

```r
library(MASS)

e.mod <- lda(Direction ~ Lag2, data = d.training,family = binomial)

e.preds <-  predict(e.mod,newdata = d.test,type = "response")

table(e.preds$class,d.test$Direction)
```

```
##
##         Down Up
##   Down     9  5
##   Up      34 56
```

```r
mean(e.preds$class== d.test$Direction)
```

```
## [1] 0.625
```

f. Repeat (d) using QDA.

```r
f.mod <- qda(Direction ~ Lag2, data = d.training,family = binomial)

f.preds <-  predict(f.mod,newdata = d.test,type = "response")

table(f.preds$class,d.test$Direction)
```

```
##
##         Down Up
##   Down     0  0
##   Up      43 61
```

```r
mean(f.preds$class== d.test$Direction)
```

```
## [1] 0.5865385
```

g. Repeat (d) using KNN with K = 1.

```r
library(class)
set.seed(1)
g.train <- as.matrix(d.training$Lag2)
g.test <- as.matrix(d.test$Lag2)
g.pred <- knn(g.train,g.test,d.training$Direction,k=1)

table(g.pred,d.test$Direction)
```

```
##
## g.pred Down Up
##   Down   21 30
##   Up     22 31
```

```
mean(g.pred==d.test$Direction)
```

```
## [1] 0.5
```

h. Which of these methods appears to provide the best results on this data?

Both LDA and logistic regression methods produce the highest proportion of correctly classified test set responses with 62.5% correctly identified.