Project № 2

Mikhailichenko Anastasia Sergeevna

2022-12-09

Download the dataset and see the beginning

```
data("Boston")

head(Boston)
```

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

Check if there are empty values

```
colSums(is.na(Boston))
```

```
##    crim      zn   indus    chas     nox      rm     age     dis     rad     tax
##       0       0       0       0       0       0       0       0       0       0
## ptratio   black   lstat    medv
##       0       0       0       0
```

They are no NA here. Hooray!

Standardize the predictors and write them into a new variable. Do not touch discrete variables (chas)

```
Bostonst <- data.frame(scale(Boston[1:3], center=TRUE, scale = TRUE),
                       Boston[4],
                       scale(Boston[5:13], center=TRUE, scale = TRUE),
                       Boston[14])
```

Building a complete model

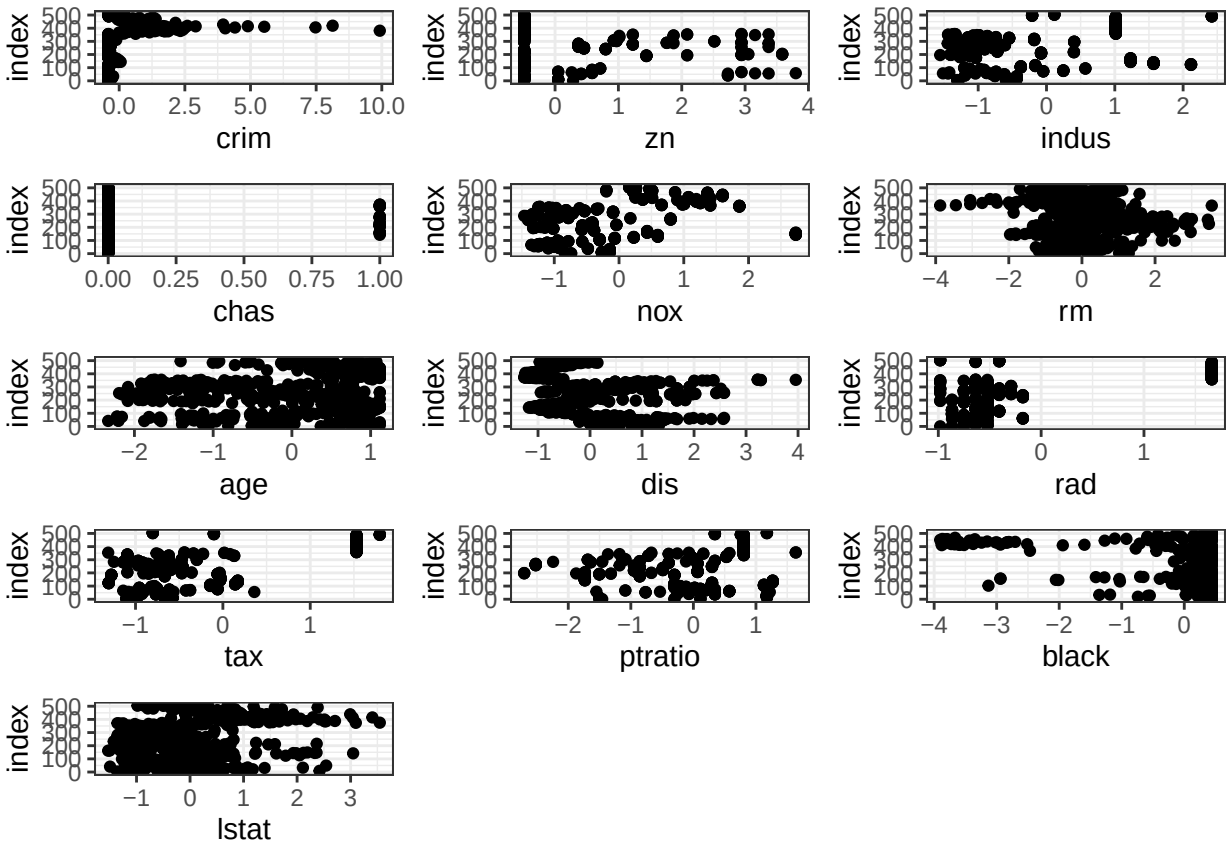
```
mod <- lm(medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat, data = Bostonst)
summary(mod)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + chas + nox + rm + age +
##     dis + rad + tax + ptratio + black + lstat, data = Bostonst)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.34697    0.21921 101.943  < 2e-16 ***
## crim        -0.92906    0.28269  -3.287 0.001087 **
## zn           1.08264    0.32016   3.382 0.000778 ***
## indus        0.14104    0.42188   0.334 0.738288
## chas         2.68673    0.86158   3.118 0.001925 **
## nox         -2.05875    0.44262  -4.651 4.25e-06 ***
## rm           2.67688    0.29364   9.116  < 2e-16 ***
## age          0.01949    0.37184   0.052 0.958229
## dis         -3.10712    0.41999  -7.398 6.01e-13 ***
## rad          2.66485    0.57770   4.613 5.07e-06 ***
## tax         -2.07884    0.63379  -3.280 0.001112 **
## ptratio     -2.06265    0.28323  -7.283 1.31e-12 ***
## black        0.85011    0.24521   3.467 0.000573 ***
## lstat       -3.74733    0.36216 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Build Cleveland charts to search for pop-up values

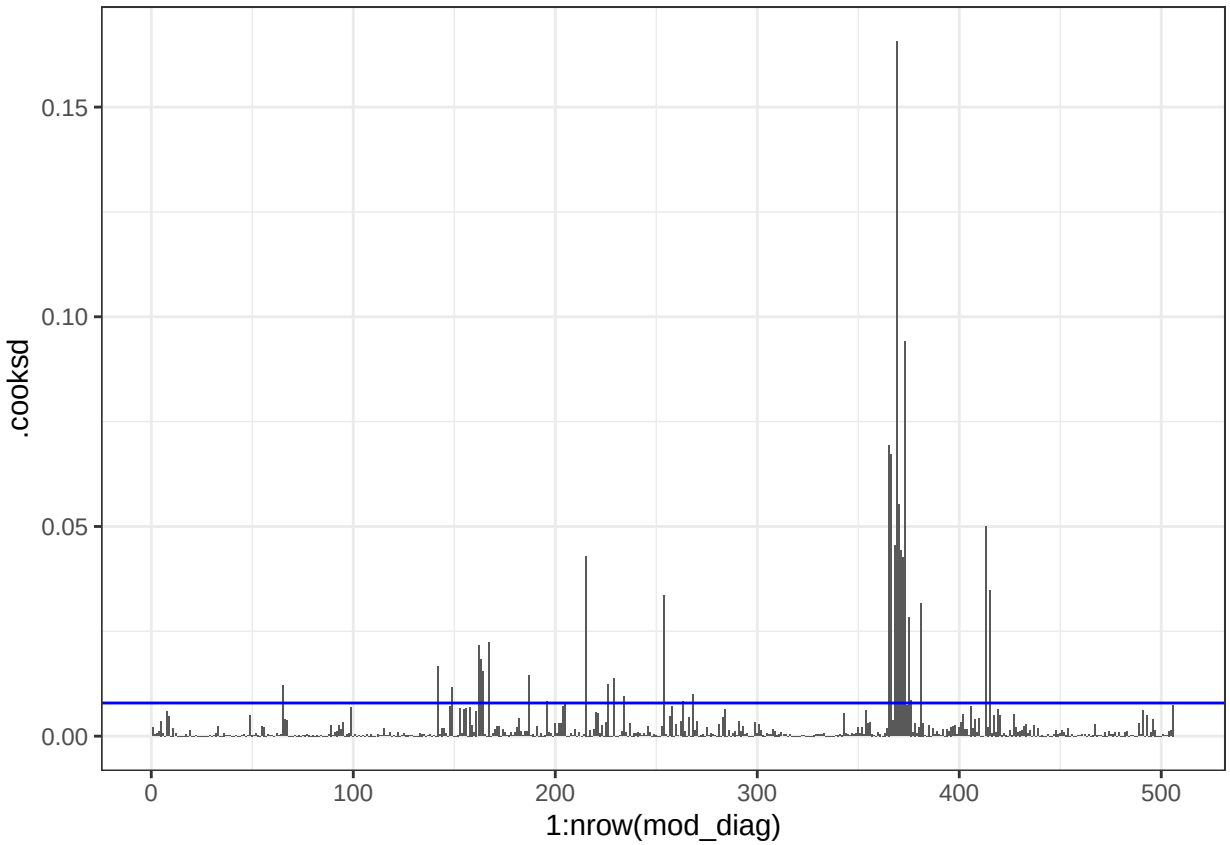
```
gg_dot <- ggplot(Bostonst, aes(y = 1:nrow(Bostonst))) + geom_point() + ylab('index')
Pl1 <- gg_dot + aes(x = crim)
Pl2 <- gg_dot + aes(x = zn)
Pl3 <- gg_dot + aes(x = indus)
Pl4 <- gg_dot + aes(x = chas)
Pl5 <- gg_dot + aes(x = nox)
Pl6 <- gg_dot + aes(x = rm)
Pl7 <- gg_dot + aes(x = age)
Pl8 <- gg_dot + aes(x = dis)
Pl9 <- gg_dot + aes(x = rad)
Pl10 <- gg_dot + aes(x = tax)
Pl11 <- gg_dot + aes(x = ptratio)
Pl12 <- gg_dot + aes(x = black)
Pl13 <- gg_dot + aes(x = lstat)
```

```
plot_grid(Pl1, Pl2, Pl3, Pl4, Pl5, Pl6,
          Pl7,Pl8, Pl9, Pl10, Pl11, Pl12, Pl13,
          ncol = 3, nrow = 5)
```



And plot Cook's distance chart

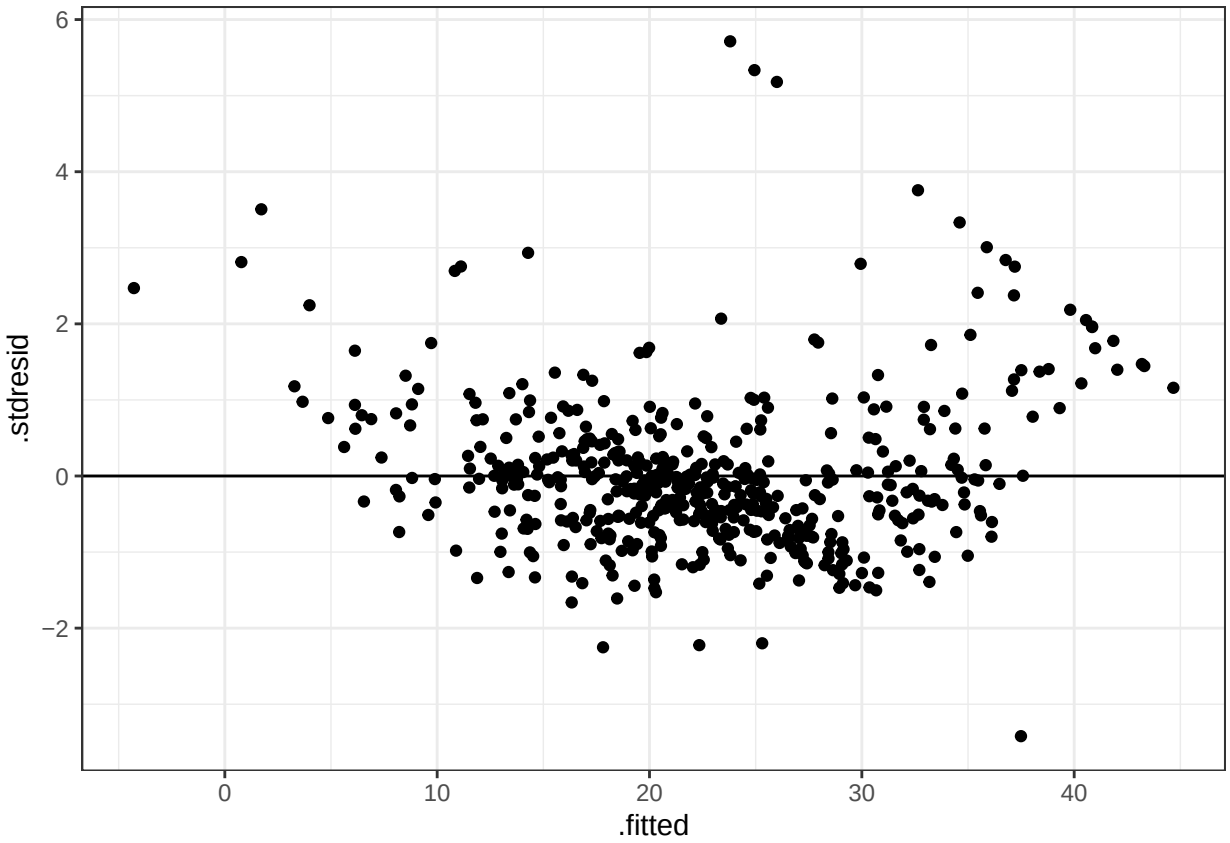
```
mod_diag <- data.frame(fortify(mod))

ggplot(data = mod_diag, aes(x = 1:nrow(mod_diag), y = .cooksd)) +
  geom_bar(stat = "identity") +
  geom_hline(yintercept = 4/nrow(mod_diag), colour ='blue')
```



Outliers are already visible on the Cleveland charts, and the Cooke distance plot only confirms this, as there are a number of observations above the threshold. They are too powerful in this model, and when refining the model, it is better to exclude them from the analysis.

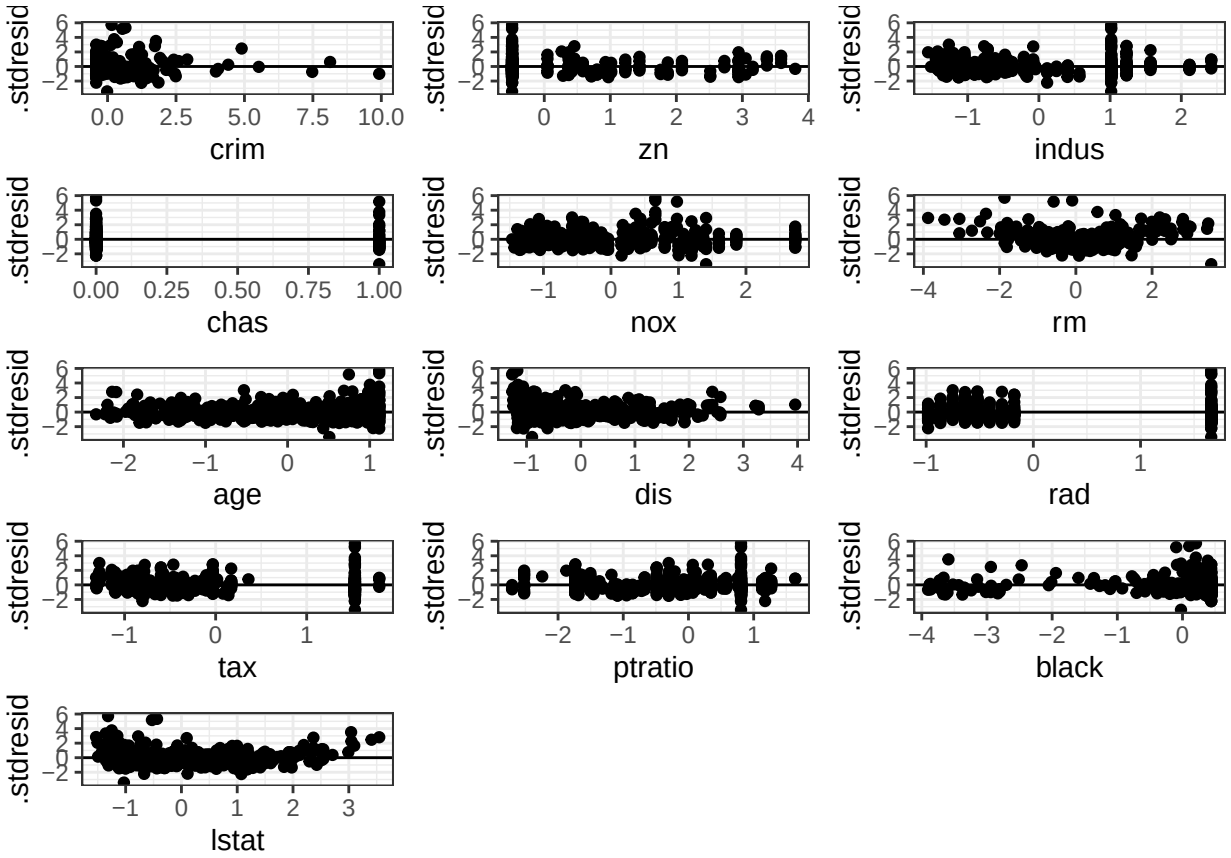
Build a graph of the residuals. There is a suspicious straight line, it may have arisen due to outlayers

```
gg_resid <- ggplot(data = mod_diag, aes(x = .fitted, y = .stdresid)) +
  geom_point() + geom_hline(yintercept = 0)
gg_resid
```



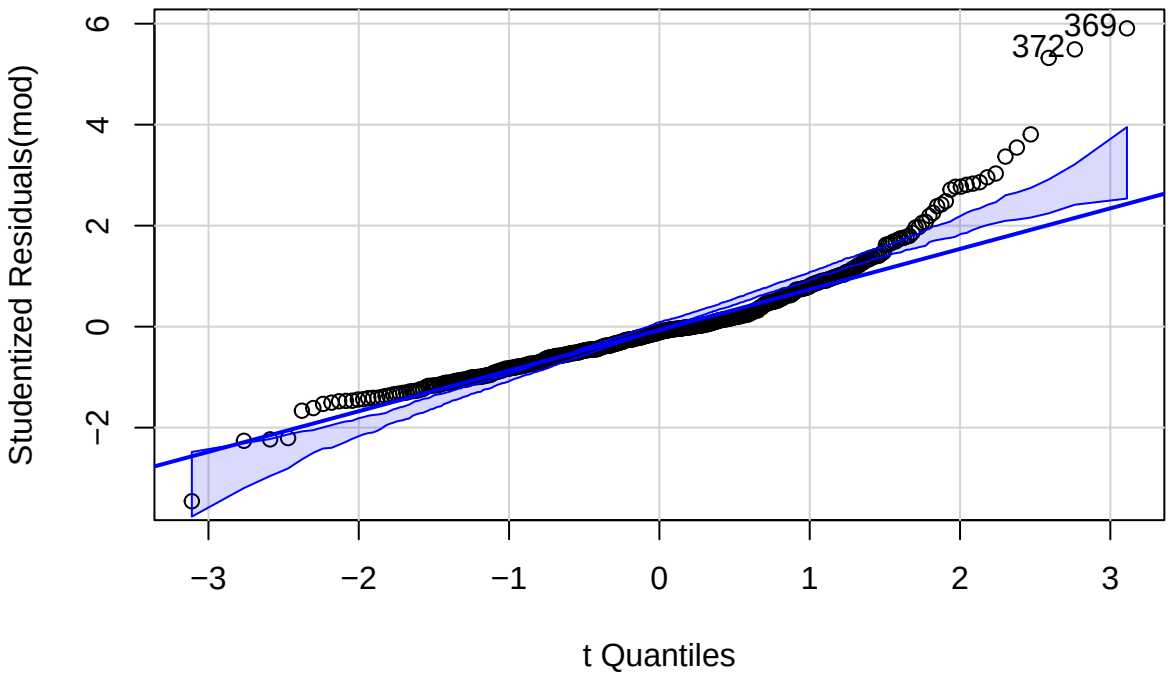
In the case of some graphs of the dependence of residuals on predictors, non-random patterns are visible, hinting at the nonlinearity of relationships (pm, lstat) and heterogeneity (age, dis)

```
res_1 <- gg_resid + aes(x = crim)
res_2 <- gg_resid + aes(x = zn)
res_3 <- gg_resid + aes(x = indus)
res_4 <- gg_resid + aes(x = chas)
res_5 <- gg_resid + aes(x = nox)
res_6 <- gg_resid + aes(x = rm)
res_7 <- gg_resid + aes(x = age)
res_8 <- gg_resid + aes(x = dis)
res_9 <- gg_resid + aes(x = rad)
res_10 <- gg_resid + aes(x = tax)
res_11 <- gg_resid + aes(x = ptratio)
res_12 <- gg_resid + aes(x = black)
res_13 <- gg_resid + aes(x = lstat)


grid.arrange(res_1, res_2, res_3, res_4,
             res_5, res_6, res_7, res_8, res_9, res_10,
             res_11, res_12, res_13, nrow = 5)
```



On the qq-plot, we see significant deviations from the normal distribution

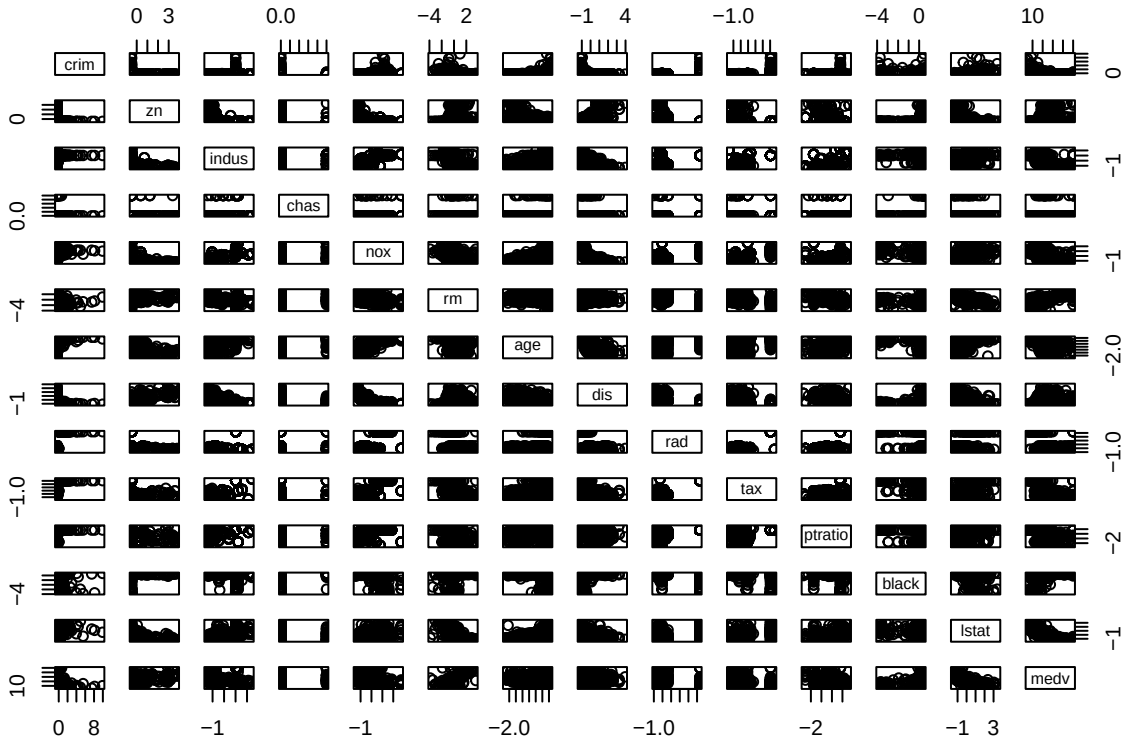
```
qqPlot(mod)
```




```
## [1] 369 372
```

Pair plots show notable correlations between predictors

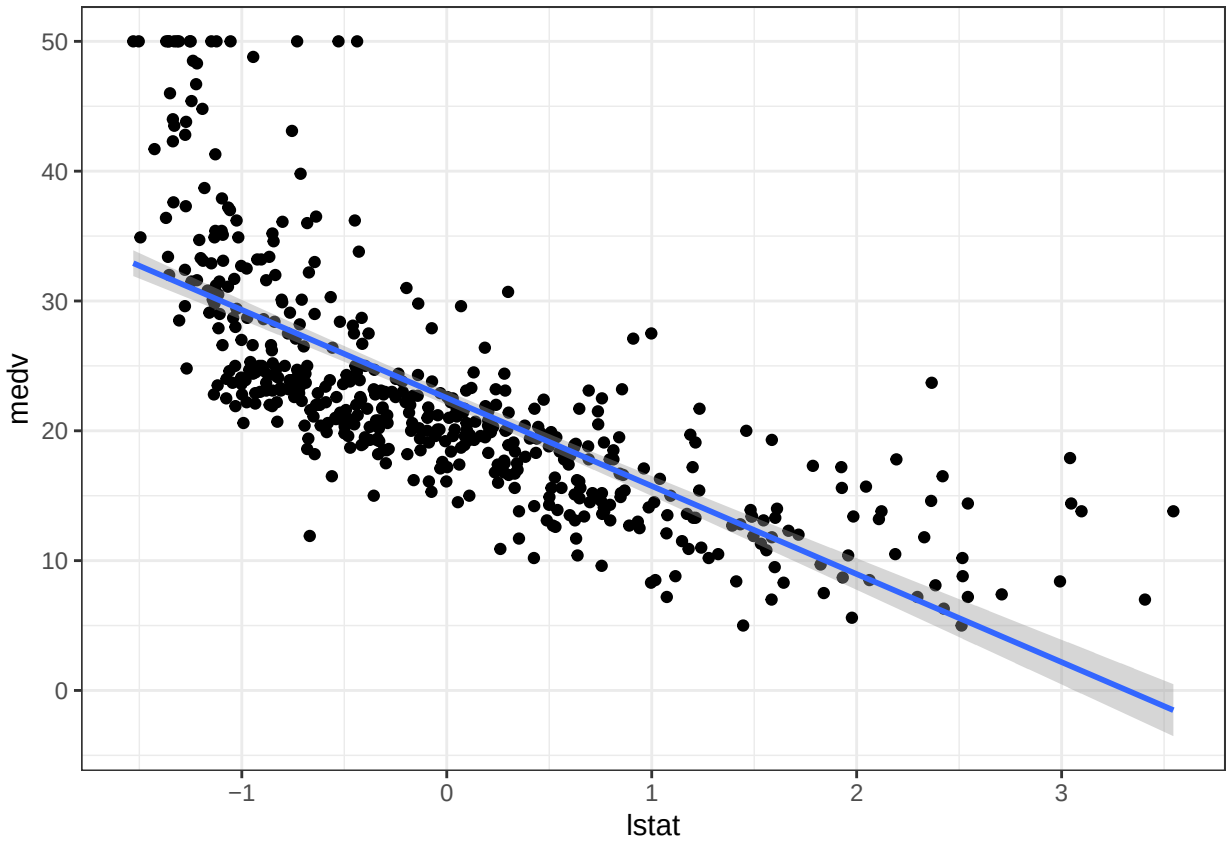
```
pairs(Bostonst)
```



The full linear model does not satisfy any conditions for the applicability of a linear model, so dividing predictions by it is not a good idea.

Still, let's build a graph of the dependence of the variable response on the variable that has the largest coefficient (lstat) in absolute value.

```
ggplot(Bostonst, aes(x = lstat, y = medv)) +
  geom_point() +
  geom_smooth(method="lm")
```



Let's try to improve the model a bit. We remove the multicollinearity of predictors by evaluating vif. One by one, we will exclude predictors in the largest vif from the model until all vif are less than 3.

```
vif(mod)
```

```
##     crim       zn    indus     chas      nox       rm      age      dis
## 1.792192 2.298758 3.991596 1.073995 4.393720 1.933744 3.100826 3.955945
##      rad      tax  ptratio    black    lstat
## 7.484496 9.008554 1.799084 1.348521 2.941491
```

```
mod <- update(mod, . ~ . - tax)
vif(mod)
```

```
##     crim       zn    indus     chas      nox       rm      age      dis
## 1.791940 2.184240 3.226015 1.058220 4.369271 1.923075 3.098044 3.954446
##      rad  ptratio    black    lstat
## 2.837494 1.788839 1.347564 2.940800
```

```
mod <- update(mod, . ~ . - nox)
vif(mod)
```

```
##     crim       zn    indus     chas       rm      age      dis      rad
## 1.785343 2.183394 2.872809 1.057571 1.904013 2.875130 3.641492 2.533616
##  ptratio    black    lstat
## 1.598944 1.339554 2.927273
```

```
mod <- update(mod, . ~ . - dis)
vif(mod)
```

```
##     crim       zn    indus     chas       rm      age      rad  ptratio
## 1.765881 1.758636 2.517520 1.056840 1.879925 2.423551 2.507024 1.530992
##    black    lstat
## 1.339553 2.926111
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + chas + rm + age + rad +
##     ptratio + black + lstat, data = Bostonst)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.1321  -3.0552  -0.7419   1.6972  28.3811
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.320608   0.234481  95.192  < 2e-16 ***
## crim        -0.647164   0.300331  -2.155 0.031656 *
## zn           0.006162   0.299715   0.021 0.983606
## indus       -0.283374   0.358597  -0.790 0.429771
## chas         3.067788   0.914747   3.354 0.000859 ***
## rm           3.099272   0.309878  10.002  < 2e-16 ***
## age          0.392857   0.351840   1.117 0.264717
## rad          0.818534   0.357848   2.287 0.022595 *
## ptratio     -2.029855   0.279644  -7.259 1.52e-12 ***
## black        0.965629   0.261577   3.692 0.000248 ***
## lstat       -3.895343   0.386603 -10.076  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.079 on 495 degrees of freedom
## Multiple R-squared:  0.7011, Adjusted R-squared:  0.695
## F-statistic: 116.1 on 10 and 495 DF,  p-value: < 2.2e-16
```

Now we will get rid of insignificant predictors one by one and we will exclude the least significant ones until only those for whom the effect on the response variable is statistically significant remain.

```
drop1(mod, test = "F")
```

```
## Single term deletions
##
## Model:
## medv ~ crim + zn + indus + chas + rm + age + rad + ptratio +
##     black + lstat
##         Df Sum of Sq   RSS    AIC  F value    Pr(>F)
## <none>               12768 1655.5
## crim     1    119.77 12888 1658.2   4.6433 0.0316565 *
```

```
## zn       1      0.01 12768 1653.5   0.0004 0.9836056
## indus    1     16.11 12784 1654.1   0.6245 0.4297709
## chas     1    290.12 13058 1664.8  11.2473 0.0008585 ***
## rm       1   2580.30 15349 1746.6 100.0321 < 2.2e-16 ***
## age      1     32.16 12800 1654.7   1.2467 0.2647169
## rad      1    134.96 12903 1658.8   5.2321 0.0225949 *
## ptratio  1   1359.09 14128 1704.7  52.6887 1.523e-12 ***
## black    1    351.52 13120 1667.2  13.6277 0.0002476 ***
## lstat    1   2618.74 15387 1747.9 101.5223 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod <- update(mod, . ~ . - zn)
drop1(mod, test = "F")
```

```
## Single term deletions
##
## Model:
## medv ~ crim + indus + chas + rm + age + rad + ptratio + black +
##     lstat
##         Df Sum of Sq   RSS    AIC  F value    Pr(>F)
## <none>               12768 1653.5
## crim     1    119.93 12888 1656.2   4.6588 0.0313732 *
## indus    1     17.00 12785 1652.1   0.6602 0.4168682
## chas     1    290.24 13059 1662.8  11.2748 0.0008461 ***
## rm       1   2611.92 15380 1745.6 101.4623 < 2.2e-16 ***
## age      1     36.64 12805 1652.9   1.4233 0.2334255
## rad      1    136.03 12904 1656.8   5.2841 0.0219354 *
## ptratio  1   1454.53 14223 1706.0  56.5024 2.645e-13 ***
## black    1    351.57 13120 1665.2  13.6571 0.0002438 ***
## lstat    1   2629.25 15398 1746.2 102.1355 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod <- update(mod, . ~ . - indus)
drop1(mod, test = "F")
```

```
## Single term deletions
##
## Model:
## medv ~ crim + chas + rm + age + rad + ptratio + black + lstat
##         Df Sum of Sq   RSS    AIC  F value    Pr(>F)
## <none>               12785 1652.1
## crim     1    115.92 12901 1654.7   4.5059 0.0342715 *
## chas     1    279.99 13065 1661.1  10.8841 0.0010395 **
## rm       1   2780.11 15566 1749.7 108.0698 < 2.2e-16 ***
## age      1     22.93 12808 1651.0   0.8914 0.3455507
## rad      1    119.38 12905 1654.8   4.6406 0.0317033 *
## ptratio  1   1482.31 14268 1705.6  57.6210 1.583e-13 ***
## black    1    368.35 13154 1664.5  14.3187 0.0001731 ***
## lstat    1   2693.41 15479 1746.9 104.6996 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod <- update(mod, . ~ . - age)
drop1(mod, test = "F")
```

```
## Single term deletions
##
## Model:
## medv ~ crim + chas + rm + rad + ptratio + black + lstat
##         Df Sum of Sq   RSS    AIC  F value    Pr(>F)
## <none>               12808 1651.0
## crim     1     116.6 12925 1653.6   4.5316 0.0337644 *
## chas     1     308.5 13117 1661.1  11.9938 0.0005798 ***
## rm       1    2959.3 15768 1754.2 115.0604 < 2.2e-16 ***
## rad      1     141.3 12950 1654.6   5.4943 0.0194711 *
## ptratio  1    1474.8 14283 1704.2  57.3427 1.792e-13 ***
## black    1     371.0 13179 1663.5  14.4251 0.0001639 ***
## lstat    1    3216.9 16025 1762.4 125.0770 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = medv ~ crim + chas + rm + rad + ptratio + black +
##     lstat, data = Bostonst)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3932  -3.0253  -0.8268   1.5912  28.9649
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.3171     0.2339  95.412  < 2e-16 ***
## crim         -0.6369     0.2992  -2.129 0.033764 *
## chas          3.1190     0.9006   3.463 0.000580 ***
## rm            3.1952     0.2979  10.727  < 2e-16 ***
## rad           0.7840     0.3345   2.344 0.019471 *
## ptratio      -2.0403     0.2694  -7.572 1.79e-13 ***
## black         0.9880     0.2601   3.798 0.000164 ***
## lstat        -3.7485     0.3352 -11.184  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.071 on 498 degrees of freedom
## Multiple R-squared:  0.7002, Adjusted R-squared:  0.6959
## F-statistic: 166.1 on 7 and 498 DF,  p-value: < 2.2e-16
```

The improved model is not perfect, but if you start from it, the developer should look at areas near the Charles River, with a high index of accessibility to radial highways, with a high percentage of black residents, and with a high average number of rooms per dwelling. It is worth avoiding areas with a high lower status of the population, with a high crime rate and a high student-to-teacher ratio in the city.