Simple Linear Regression

Prisma Erika Lopez Jimenez

11/9/2019

```
### Load your packages
library(MASS)
library(ISLR)
```

```
### Make a linear model on select variables of the `Boston` dataset
fix(Boston) # edit the `Boston` dataset in the `MASS` package
attach(Boston) # associate the variables `medv` and `lstat`
# with the dataset `Boston` without incl. in the model code below:
lm.fit=lm(medv~lstat) # linear model of the `Boston` data set where `medv`
# is the response and `lstat` is the predictor

lm.fit # take a look at the linear model basics
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

```
summary(lm.fit) # get a summary
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
names(lm.fit) # what you can extract from the model `lm.fit`
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

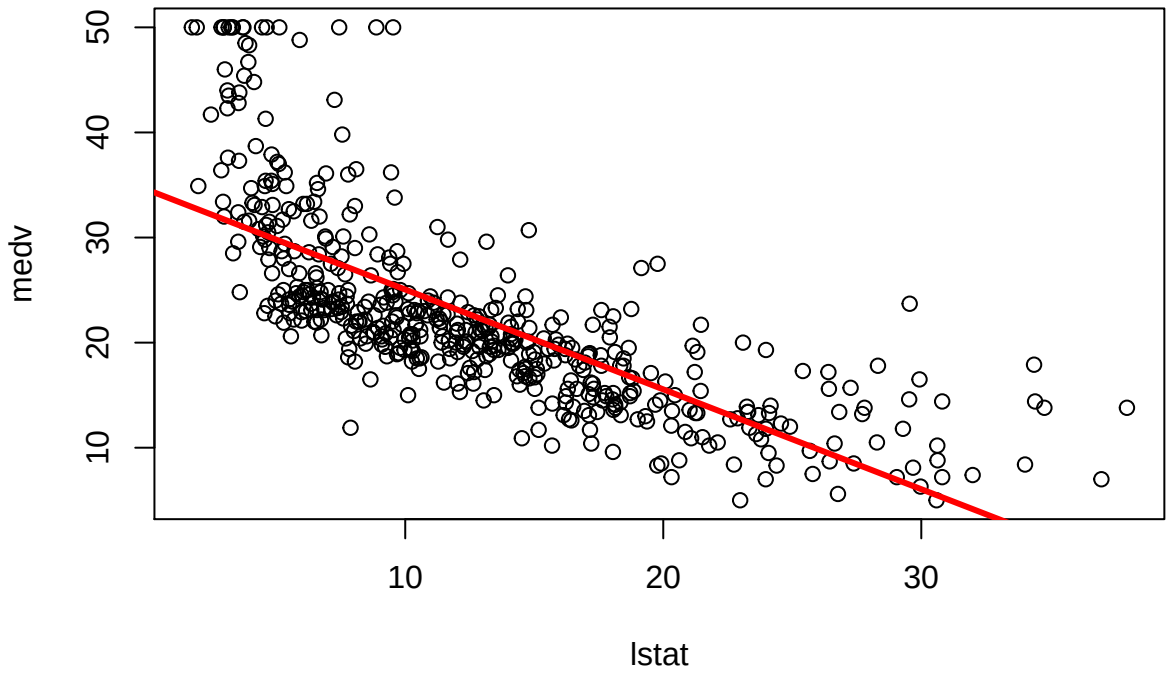
```r
coef(lm.fit) # extract the coefficient values of the linear model
```

```
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

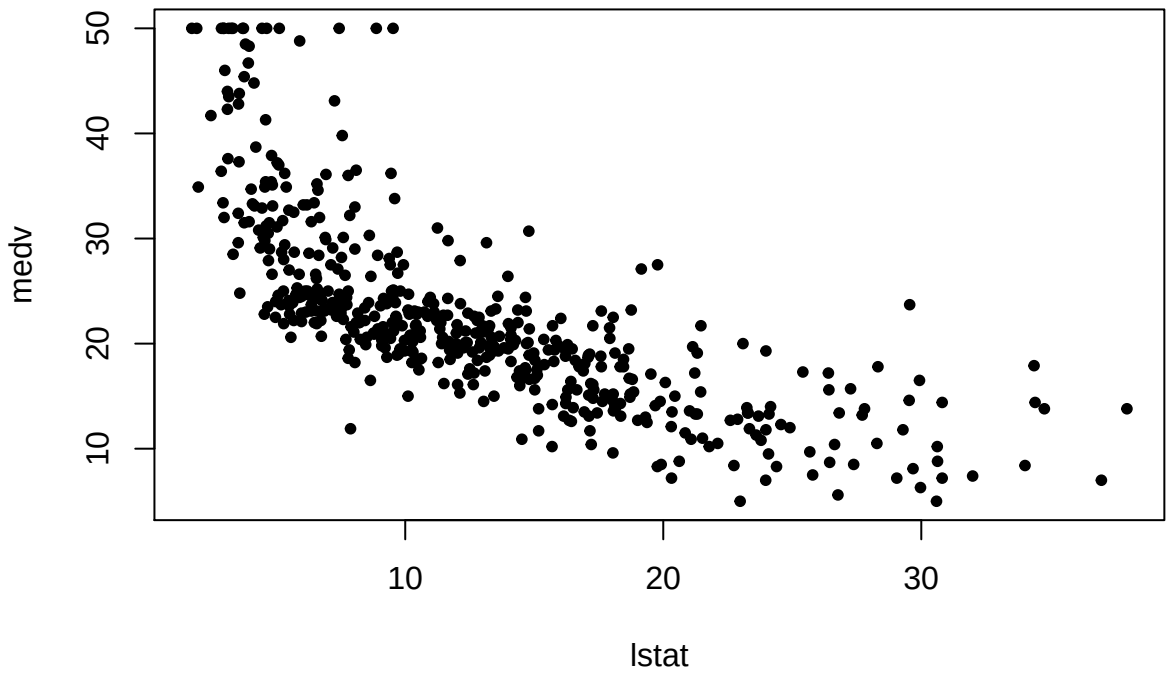
```r
confint(lm.fit) # confidence intervals
```

```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

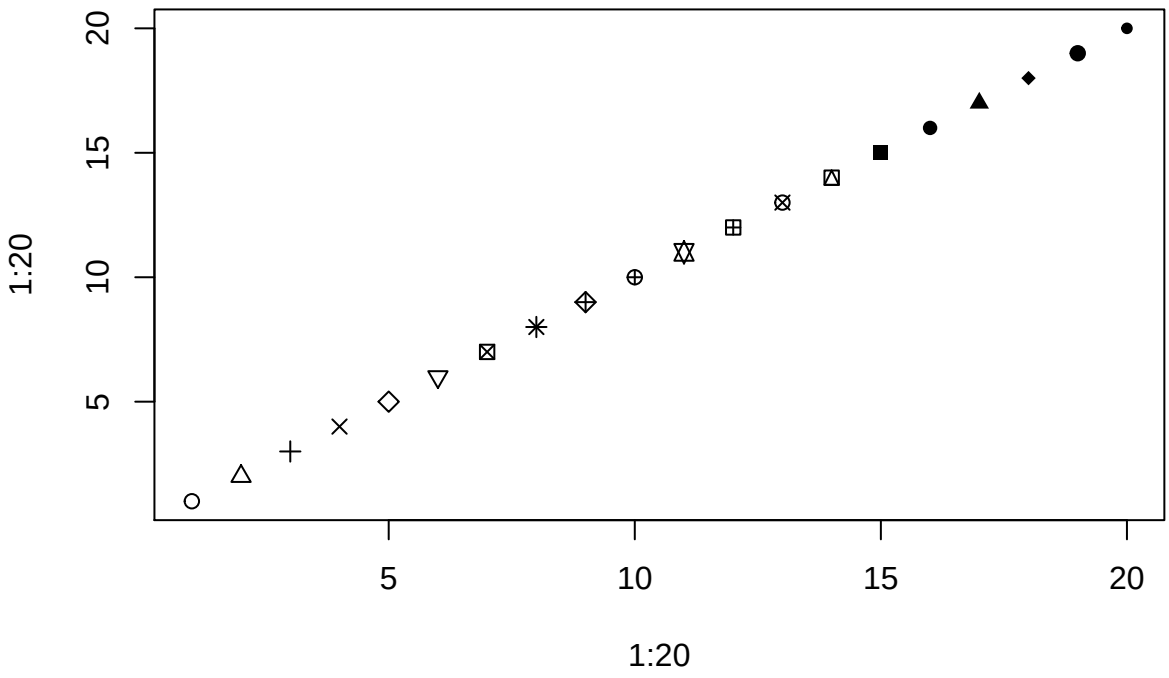
```r
### Plot the data and linear regression line for the model
plot(lstat,medv)  # scatter plot of `lstat` vs. `medv`
abline(lm.fit,lwd=3,col="red") # add linear regression line,
```



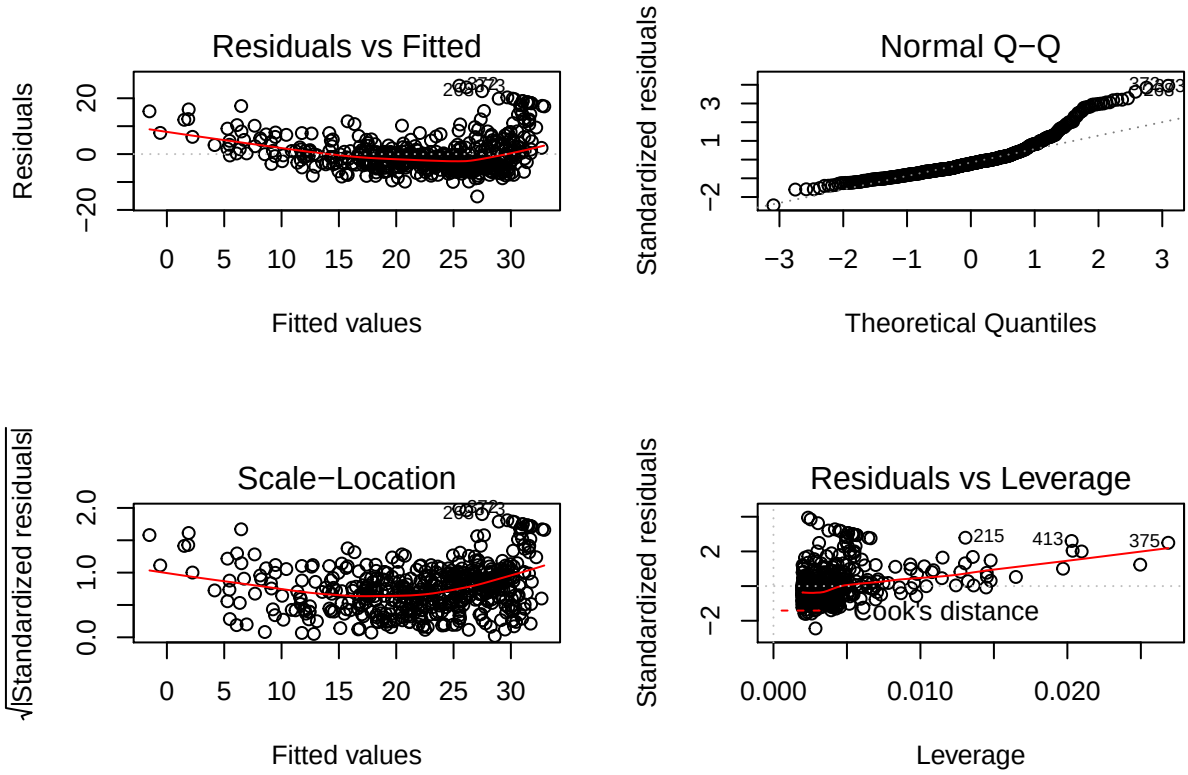

```r
# color it red, and bold it 3x
plot(lstat,medv,pch=20) # modify the dots on the scatter plot
```



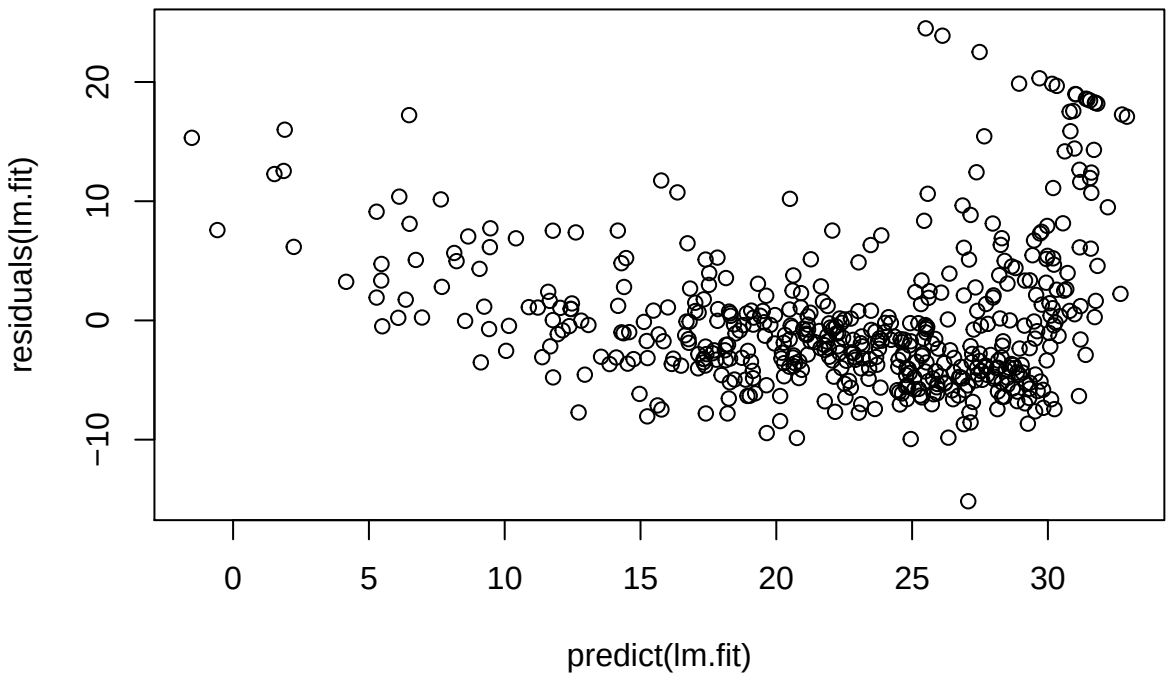

```
plot(1:20,1:20,pch=1:20)# example of plot() function implementation
```



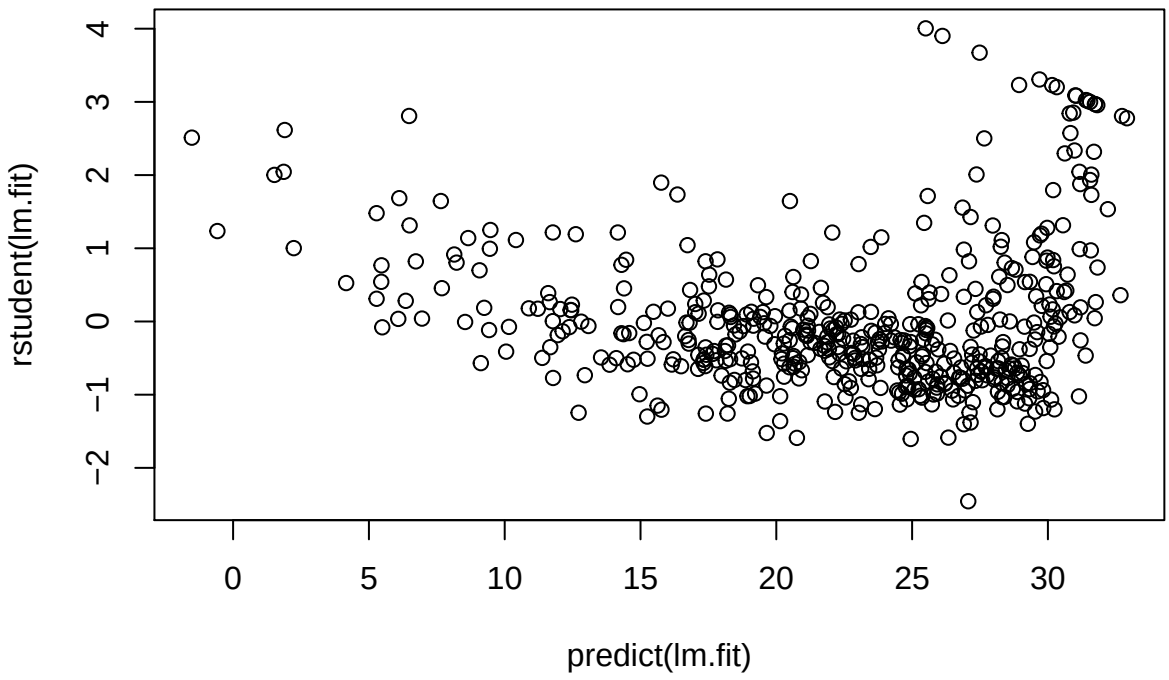

```
### Create Diagnostic plots on the complete model
par(mfrow=c(2,2)) # separate the grid as a 2 x 2 fit
plot(lm.fit) # plot 4 diagnostic plots on the `lm.fit` model
```



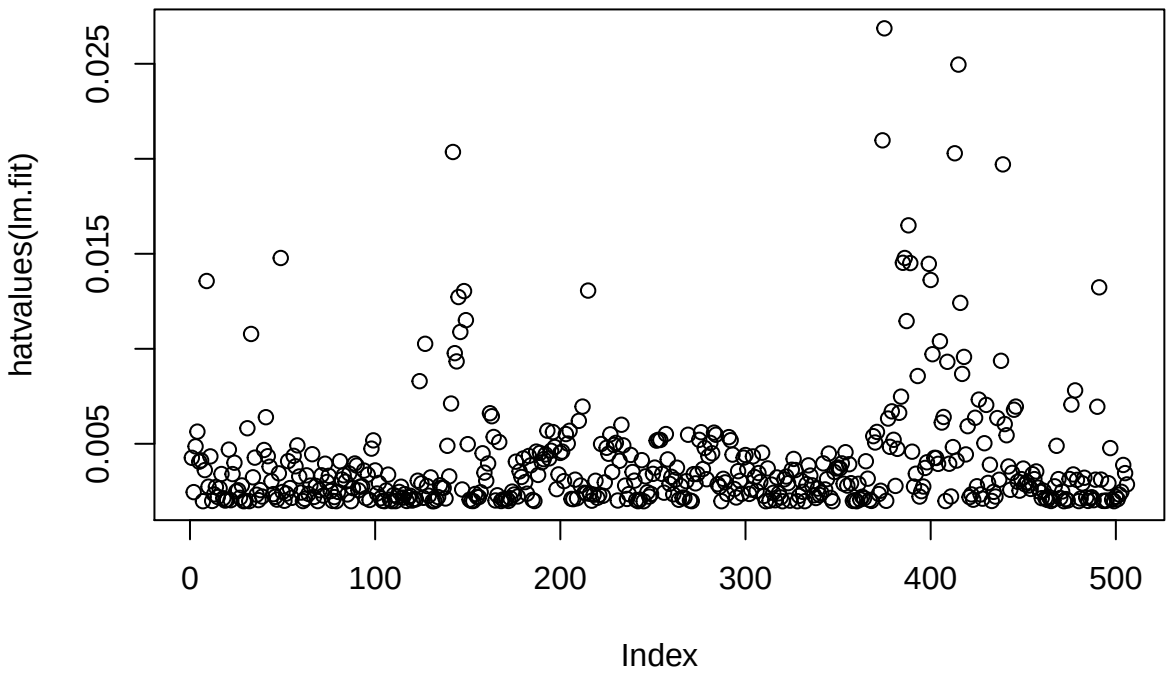

```
### Compute the residuals from a linear regression fit
plot(predict(lm.fit),residuals(lm.fit)) # compute residual method 1 note: residuals()
```




```
plot(predict(lm.fit),rstudent(lm.fit)) # compute residual method 2 note: rstudent( )
```




```
plot(hatvalues(lm.fit))
```




```
which.max(hatvalues(lm.fit))
```

```
## 375
## 375
```