3.6 Linear Regression

Zongyi Liu

2023-05-17

3.6 Linear Regression

3.6.2: Simple Linear Regression

In this lab we use the Boston data set in MASS library, and try to predict `medv` using other variables

```
library(MASS)
library(ISLR)
```

```
fix(Boston)
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

We can run the regression by using `lm` function, and show the summary of the regression results:

```
lmod=lm(medv~lstat, data=Boston)
attach(Boston)
lmod=lm(medv~lstat)
```

And we can use `names` or `coef` to see details, and `confint` to see the confidence interval.

```
names(lmod)
```

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

```
coef(lmod)
```

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

```
confint(lmod)
```

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

We can also use `predict` to see the prediction interval

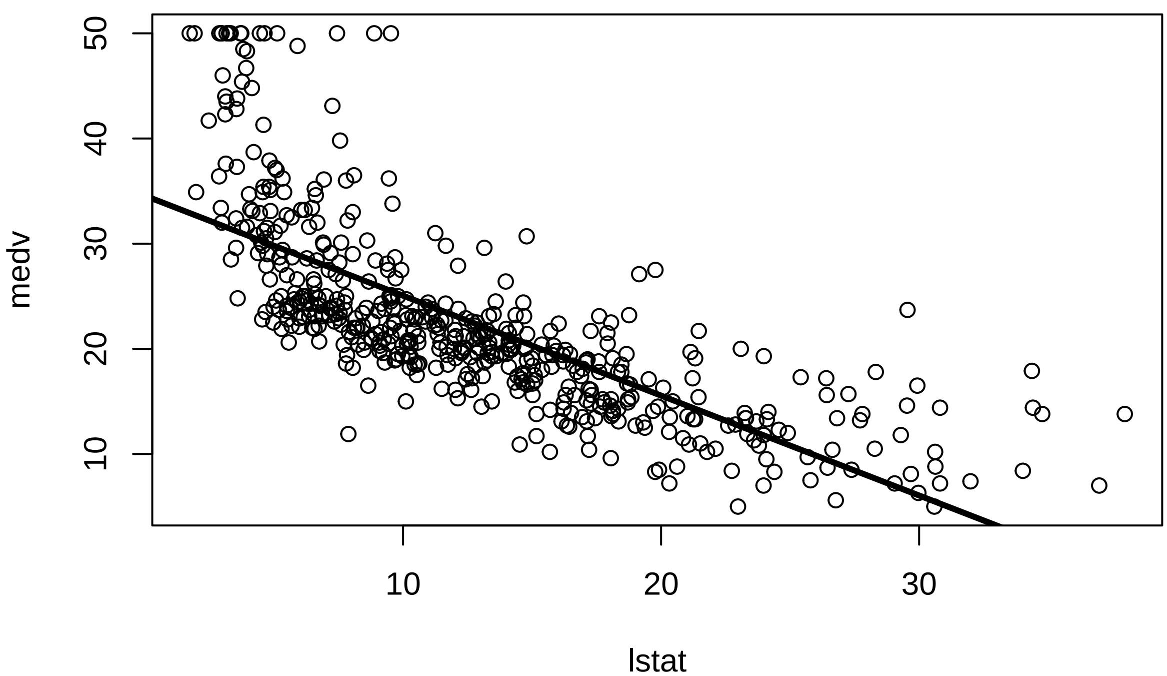
```
predict(lmod, data.frame(lstat=(c(5,10,15))), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

```
predict(lmod, data.frame(lstat=(c(5,10,15))), interval = "prediction")
```

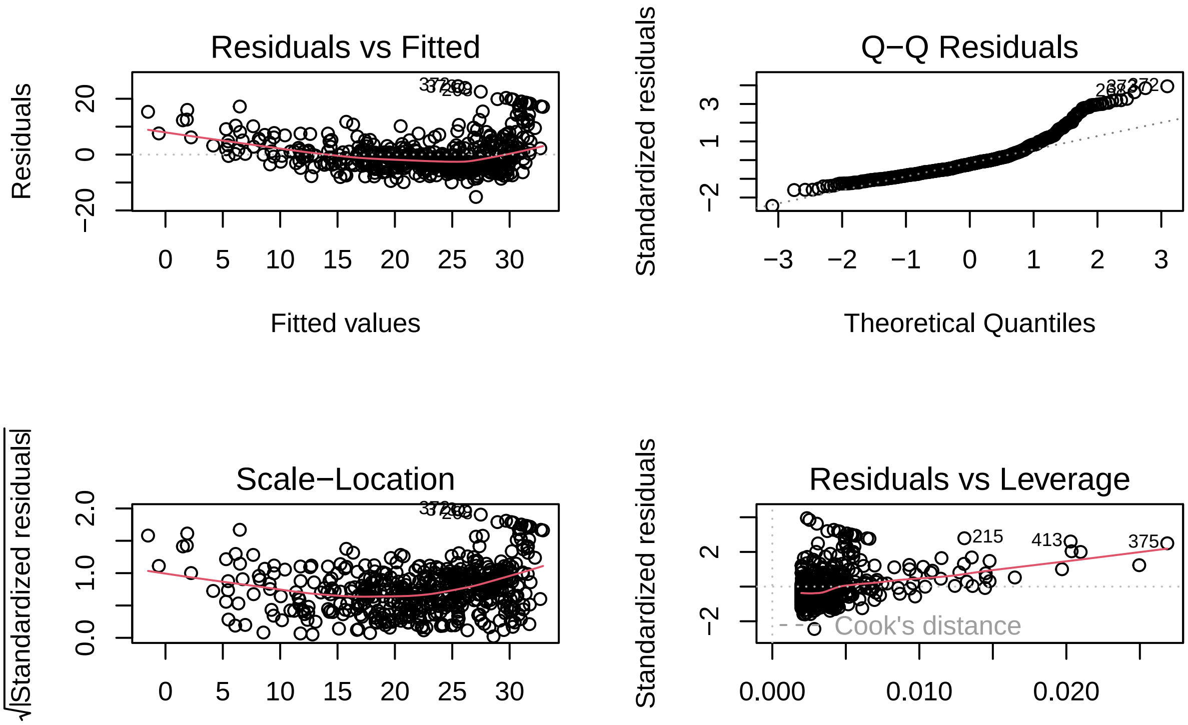
```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

```
plot(lstat,medv)
abline(lmod)
abline(lmod, lwd=3)
```

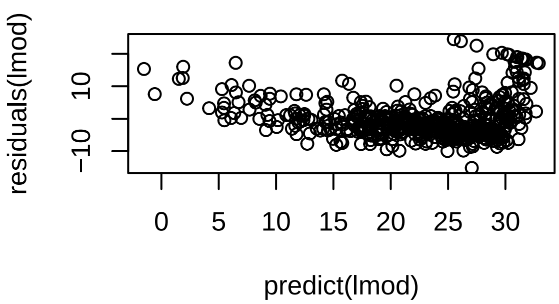


We can also use the `par` function to let pictures be splited, and use `rstudent` or `residuals` to see detailed plots.

```
par(mfrow=c(2,2))
plot(lmod)
```




```
plot(predict(lmod), residuals(lmod))
```



3.6.3 Multiple Regression

We basically need to use plus sign to perform multiple regression in R

```
lmod2 = lm(medv~lstat+age, Boston)
lmod2
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Coefficients:
## (Intercept)        lstat          age
##    33.22276     -1.03207      0.03454
```

Or use `.` to regress on all variables

```
lmod3 = lm(medv~., Boston)
lmod3
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Coefficients:
## (Intercept)         crim           zn        indus         chas          nox
##   3.646e+01   -1.080e-01    4.642e-02    2.056e-02    2.687e+00   -1.777e+01
##          rm          age          dis          rad          tax      ptratio
##   3.810e+00    6.922e-04   -1.476e+00    3.060e-01   -1.233e-02   -9.527e-01
##       black        lstat
##   9.312e-03   -5.248e-01
```

We can use `?summar$lmod` to see details. `summary(lomd)$sigma` gives RSE, and r.sq gives R^2

3.6.4 Interaction Terms

For interactions terms, we can use `*` to show it. `lstat*age` equals to `lstat+age+lstat:age`:

```
summary(lm(medv~lstat*age, data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ lstat * age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

3.6.5 Non-linear Transformations of the Predictors

We can use `I()` to transform the predictor into special forms.

```r
lm.fit2=lm(medv~lstat+I(lstat^2))
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

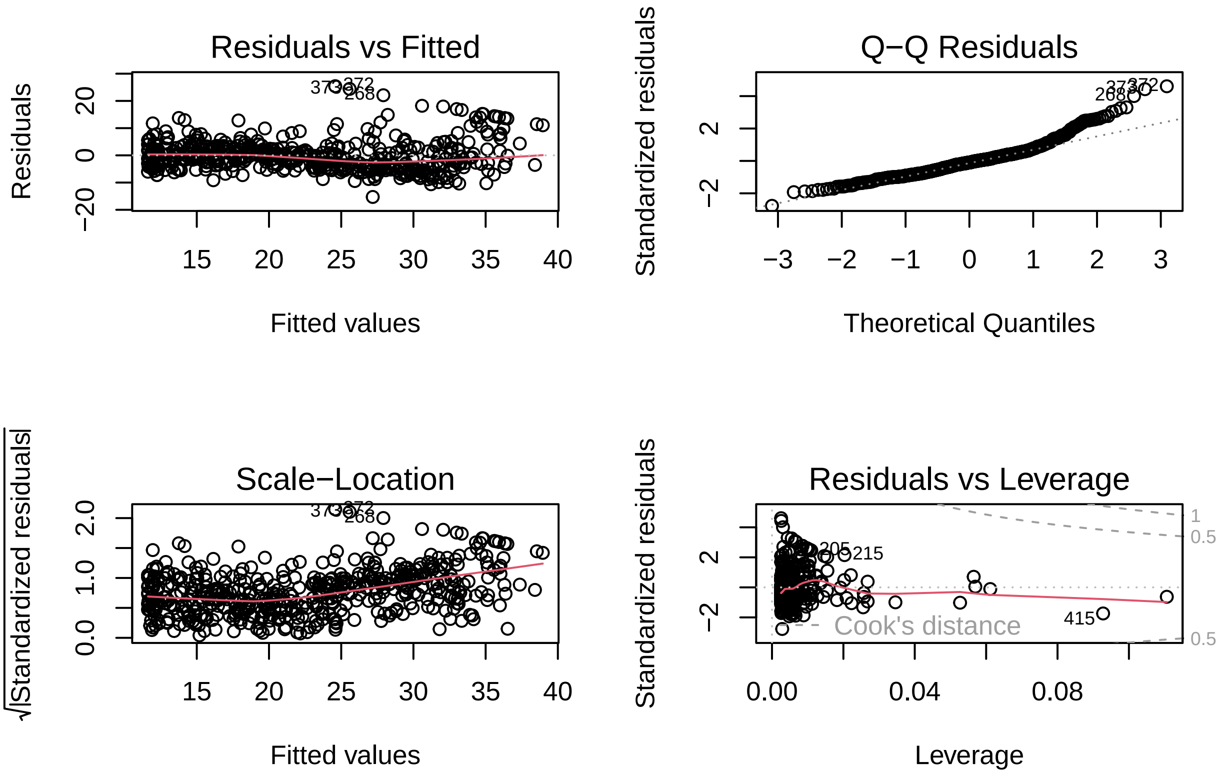
To test which model is better, we can use the `anova()` function:

```r
lm.fit=lm(medv~lstat)
anova(lm.fit,lm.fit2)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here the p-value is near to zero, meaning that the the model containing the quadratic form is much more superior to the original model.

```r
par(mfrow=c(2,2))
plot(lm.fit2)
```



There is also a shortcut to add terms in `poly()` function:

```
lm.fit5=lm(medv~poly(lstat,5))
summary(lm.fit5)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 5))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5433  -3.1039  -0.7052   2.0844  27.1153
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       22.5328     0.2318  97.197  < 2e-16 ***
## poly(lstat, 5)1 -152.4595     5.2148 -29.236  < 2e-16 ***
## poly(lstat, 5)2   64.2272     5.2148  12.316  < 2e-16 ***
## poly(lstat, 5)3  -27.0511     5.2148  -5.187 3.10e-07 ***
## poly(lstat, 5)4   25.4517     5.2148   4.881 1.42e-06 ***
## poly(lstat, 5)5  -19.2524     5.2148  -3.692 0.000247 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.215 on 500 degrees of freedom
## Multiple R-squared:  0.6817, Adjusted R-squared:  0.6785
## F-statistic: 214.2 on 5 and 500 DF,  p-value: < 2.2e-16
```

3.6.6 Qualitative Predictors

Then we will use the `Carseara` data set

```
fix(Carseats)
names(Carseats)
```

```
##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"
##  [6] "Price"       "ShelveLoc"   "Age"         "Education"   "Urban"
## [11] "US"
```

Do a regression on it with interaction terms:

```
lm.fit=lm(Sales~.+Income:Advertising+Price:Age,data=Carseats)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Sales ~ . + Income:Advertising + Price:Age, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9208 -0.7503  0.0177  0.6754  3.3413
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.5755654  1.0087470   6.519 2.22e-10 ***
## CompPrice           0.0929371  0.0041183  22.567  < 2e-16 ***
## Income              0.0108940  0.0026044   4.183 3.57e-05 ***
## Advertising         0.0702462  0.0226091   3.107 0.002030 **
## Population          0.0001592  0.0003679   0.433 0.665330
## Price              -0.1008064  0.0074399 -13.549  < 2e-16 ***
## ShelveLocGood       4.8486762  0.1528378  31.724  < 2e-16 ***
## ShelveLocMedium     1.9532620  0.1257682  15.531  < 2e-16 ***
## Age                -0.0579466  0.0159506  -3.633 0.000318 ***
## Education          -0.0208525  0.0196131  -1.063 0.288361
## UrbanYes            0.1401597  0.1124019   1.247 0.213171
## USYes              -0.1575571  0.1489234  -1.058 0.290729
## Income:Advertising  0.0007510  0.0002784   2.698 0.007290 **
## Price:Age           0.0001068  0.0001333   0.801 0.423812
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.011 on 386 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8719
## F-statistic:   210 on 13 and 386 DF,  p-value: < 2.2e-16
```

To see the dummy variables, we can use the `cotrast()` function

```
attach(Carseats)
contrasts(ShelveLoc)
```

```
##         Good Medium
## Bad       0      0
## Good      1      0
## Medium    0      1
```