

3.6 Lab: Linear Regression

Simple Linear Regression

```
#####data
```

```
install.packages("ISLR")
```

```
library(MASS)
```

```
library(ISLR)
```

```
fix(Boston)
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
6	0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
8	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
9	0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5
10	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9
11	0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15
12	0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9
13	0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7
14	0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4
15	0.63796	0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21	380.02	10.26	18.2
16	0.62739	0	8.14	0	0.538	5.834	56.5	4.4986	4	307	21	395.62	8.47	19.9
17	1.05393	0	8.14	0	0.538	5.935	29.3	4.4986	4	307	21	386.85	6.58	23.1
18	0.7842	0	8.14	0	0.538	5.99	81.7	4.2579	4	307	21	386.75	14.67	17.5
19	0.80271	0	8.14	0	0.538	5.456	36.6	3.7965	4	307	21	288.99	11.69	20.2
20	0.7258	0	8.14	0	0.538	5.727	69.5	3.7965	4	307	21	390.95	11.28	18.2

```
#Call:
lm(formula = medv ~ lstat, data = Boston)
```

```
Coefficients:
```

```
(Intercept)      34.55
lstat        -0.95
```

```
names(lm.fit)
```

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

```
> confint(lm.fit)
```

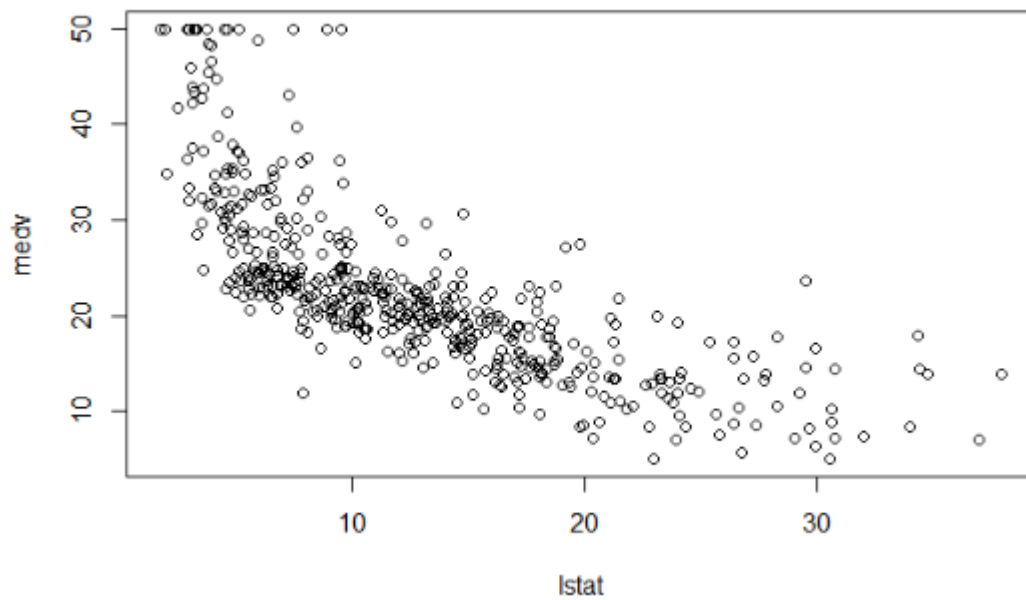
```
                2.5 %      97.5 %
(Intercept) 33.448457 35.6592247
lstat      -1.026148 -0.8739505
```

The predict() function can be used to produce confidence intervals and predict() prediction intervals for the prediction

```
> predict(lm.fit, data.frame(lstat=c(5, 10, 15)),
+         interval = "confidence")
```

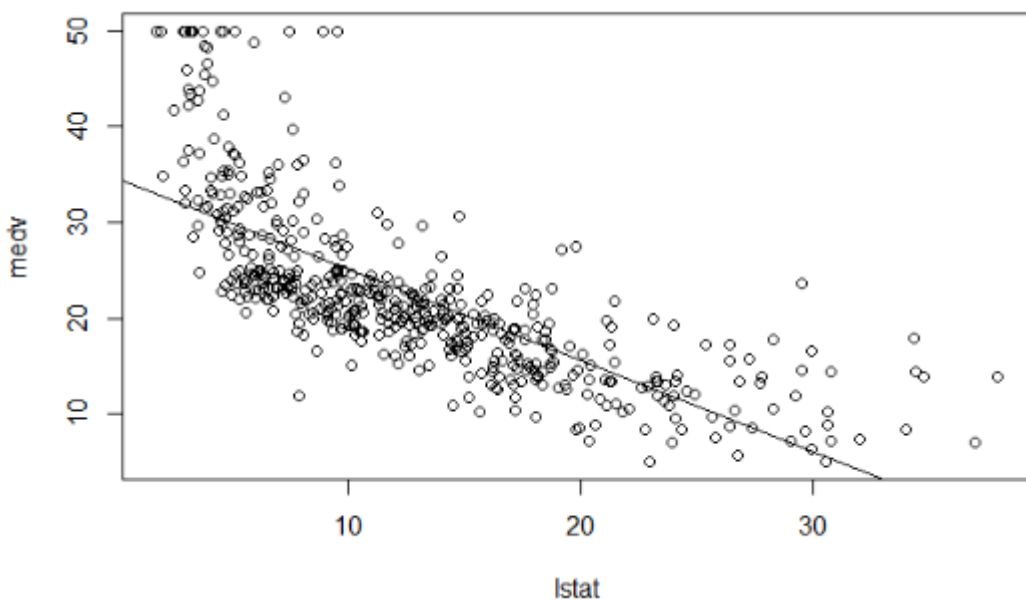
```
      fit      lwr      upr
1 29.80359 29.00741 30.59978
2 25.05335 24.47413 25.63256
3 20.30310 19.73159 20.87461
```

```
> plot(lstat ,medv)
```



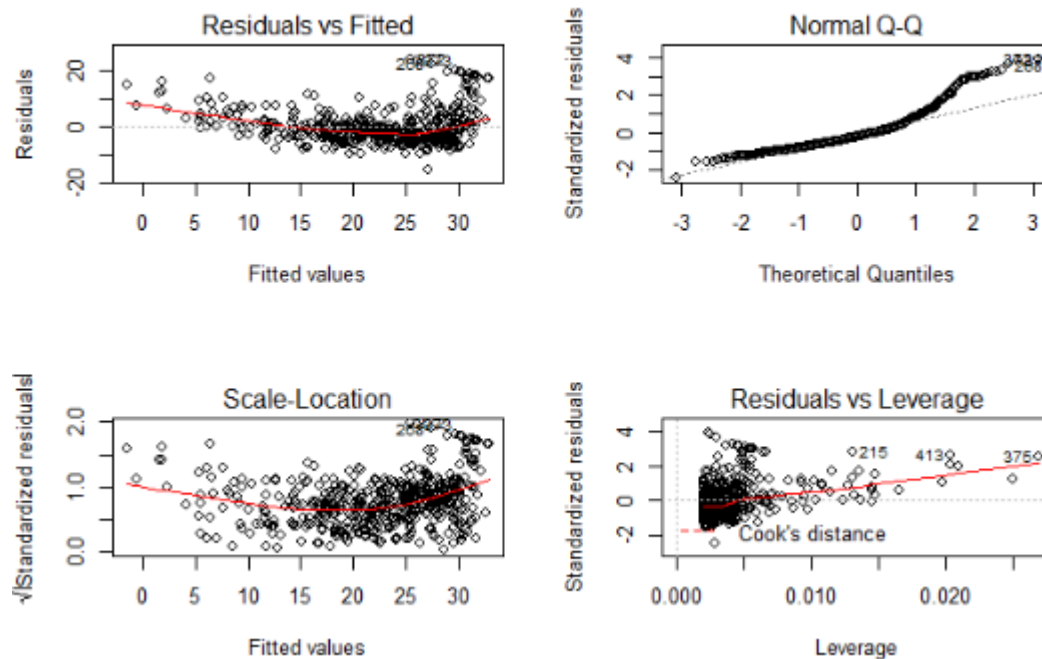
##To draw a line with intercept a and slope b

```
abline (lm.fit)
```



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

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



3.6.3 Multiple Linear Regression

```
#Call:
lm(formula = medv ~ lstat + age, data = Boston)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-15.981  -3.978  -1.283   1.968   23.158
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 33.22276    0.73085   45.458 < 2e-16 ***
lstat      -1.03207    0.04819  -21.416 < 2e-16 ***
age         0.03454    0.01223   2.826  0.00491 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 6.173 on 503 degrees of freedom
Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
F-statistic: 309 on 2 and 503 DF, p-value: < 2.2e-16
```

```
> library(car)
> vif(lm.fit)
      lstat      age 
1.569395 1.569395
```

3.6.5 Non-linear Transformations of the Predictors

```
> 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

```
> anova(lm.fit ,lm.fit2)
```

Analysis of Variance Table

Model 1: medv ~ lstat

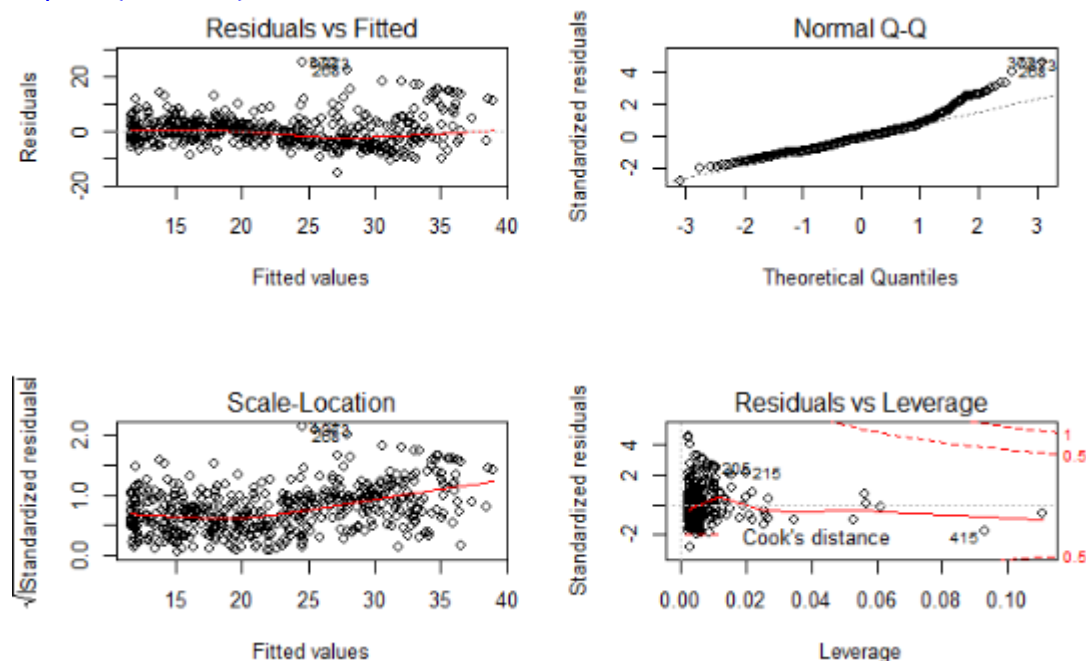
Model 2: medv ~ lstat + I(lstat^2)

	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

```
> par(mfrow=c(2,2))
```

```
> plot(lm.fit2)
```



##poly() function to create the polynomial within lm()

3.6.7 Writing Functions

Example:

```
> LoadLibraries=function (){  
+   library (ISLR)  
+   library (MASS)  
+   print (" The libraries have been loaded .")  
+ }
```