Tutorial 3: Regression I: Linear Models

ECO3080: Machine Learning in Business

Instructor: Prof. Qihui Chen
Teaching Assistant: Long Ma

September 22, 2022

- True model (Population model):

$$y_i = \beta_0^{true} + \beta_1^{true} x_{1,i} + \beta_2^{true} x_{2,i} + \cdots + \beta_k^{true} x_{k,i} + \varepsilon_i^{true} \tag{1}$$

- OLS is just a tool targeting at parameters β_0 to β_k:

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1,i} + \hat{\beta}_2 x_{2,i} + \cdots + \hat{\beta}_k x_{k,i} + \hat{\varepsilon}_i \tag{2}$$

- Impose assumptions to make sure that the estimation is "good".

- How to measure this "good"?
  - Unbiasedness: "many shots" $\rightarrow$ average level;
  - Consistency: "many shots" $\rightarrow$ under large sample;
  - Efficiency: "one-shot" $\rightarrow$ variance (range).
- Is this "good" always important in machine learning?
  - Trade-off between "Unbiasedness" and "Variance";
  - Goals: Causal Inference v.s. Out-of-sample Prediction.

- What assumptions should we impose?
  1. No perfect multicollinearity: $rank(X) = k + 1$;
  2. Linearity: true model is true;
  3. Strong exogeneity: $\mathbb{E}[\varepsilon_i^{true}|X] = \mathbb{E}[\varepsilon_i^{true}] = 0$
  4. Spherical variance: $VCov(\varepsilon^{true}|X) = \sigma^2 I$;
  5. Gaussianity: $\varepsilon^{true}|X$ is jointly normal.
- What problems are solved under each asusmption?

- Further discussions:
  - What are the differences between "i.i.d." data and time series data?
  - Will the OLS results be affected by those differences?
  - How to cope with this kind of problems?
- We might talk about this in the last few weeks.
- What we care about in machine learning are two things:
  1. Prediction Accuracy (the most important issue);
  2. Interpretability (minor issue but also noteworthy in business/economic/financial analysis).

First Look on Data Set

- Let's look at the data set first: 506 Obs. and 14 variables

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
1	0.00632	18.0	2.31	0	0.5380	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
2	0.02731	0.0	7.07	0	0.4690	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
3	0.02729	0.0	7.07	0	0.4690	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
4	0.03237	0.0	2.18	0	0.4580	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	0.06905	0.0	2.18	0	0.4580	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
6	0.02985	0.0	2.18	0	0.4580	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
7	0.08829	12.5	7.87	0	0.5240	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
8	0.14455	12.5	7.87	0	0.5240	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	27.1
9	0.21124	12.5	7.87	0	0.5240	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	16.5
10	0.17004	12.5	7.87	0	0.5240	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	18.9
11	0.22489	12.5	7.87	0	0.5240	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15.0
12	0.11747	12.5	7.87	0	0.5240	6.009	82.9	6.2267	5	311	15.2	396.90	13.27	18.9
13	0.09378	12.5	7.87	0	0.5240	5.889	39.0	5.4509	5	311	15.2	390.50	15.71	21.7
14	0.62976	0.0	8.14	0	0.5380	5.949	61.8	4.7075	4	307	21.0	396.90	8.26	20.4
15	0.63796	0.0	8.14	0	0.5380	6.096	84.5	4.4619	4	307	21.0	380.02	10.26	18.2

Variables

1. crim: per capita crime rate by town.
2. zn: proportion of residential land zoned for lots over 25,000 sq.ft.
3. indus: proportion of non-retail business acres per town.
4. chas: Charles River dummy variable (= 1 if tract bounds river).
5. nox: nitrogen oxides concentration (parts per 10 million).
6. rm: average number of rooms per dwelling.
7. age: proportion of owner-occupied units built prior to 1940.
8. dis: weighted mean of distances to five Boston employment centres.
9. rad: index of accessibility to radial highways.
10. tax: full-value property-tax rate per 10,000 dollar.
11. ptratio: pupil-teacher ratio by town.
12. black: $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town.
13. lstat: lower status of the population (percent).
14. medv: median value of owner-occupied homes in 1000 dollar.

Table: Summary statistics of Boston data

Statistic	N	Mean	St. Dev.	Min	Max
crim	506	3.614	8.602	0.006	88.976
zn	506	11.364	23.322	0.000	100.000
indus	506	11.137	6.860	0.460	27.740
chas	506	0.069	0.254	0	1
nox	506	0.555	0.116	0.385	0.871
rm	506	6.285	0.703	3.561	8.780
age	506	68.575	28.149	2.900	100.000
dis	506	3.795	2.106	1.130	12.126
rad	506	9.549	8.707	1	24
tax	506	408.237	168.537	187	711
ptratio	506	18.456	2.165	12.600	22.000
black	506	356.674	91.295	0.320	396.900
lstat	506	12.653	7.141	1.730	37.970
medv	506	22.533	9.197	5.000	50.000

Easy example: simple LR

- Split the data set into training set and validation set.
- Run simple regression on the training set and the result is:

```
> reg1 <- lm(medv ~ lstat, data = Boston_train) # main regression
>
> summary(reg1) # summary results of reg1

Call:
lm(formula = medv ~ lstat, data = Boston_train)

Residuals:
    Min      1Q  Median      3Q     Max
-15.017  -3.781  -1.173   1.639  24.067

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.66826    0.67161   51.62   <2e-16 ***
lstat       -0.98364    0.04741  -20.75   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.058 on 352 degrees of freedom
Multiple R-squared:  0.5501,    Adjusted R-squared:  0.5489
F-statistic: 430.5 on 1 and 352 DF,  p-value: < 2.2e-16
```

Easy example: simple LR

- Some other things you can do after regression:

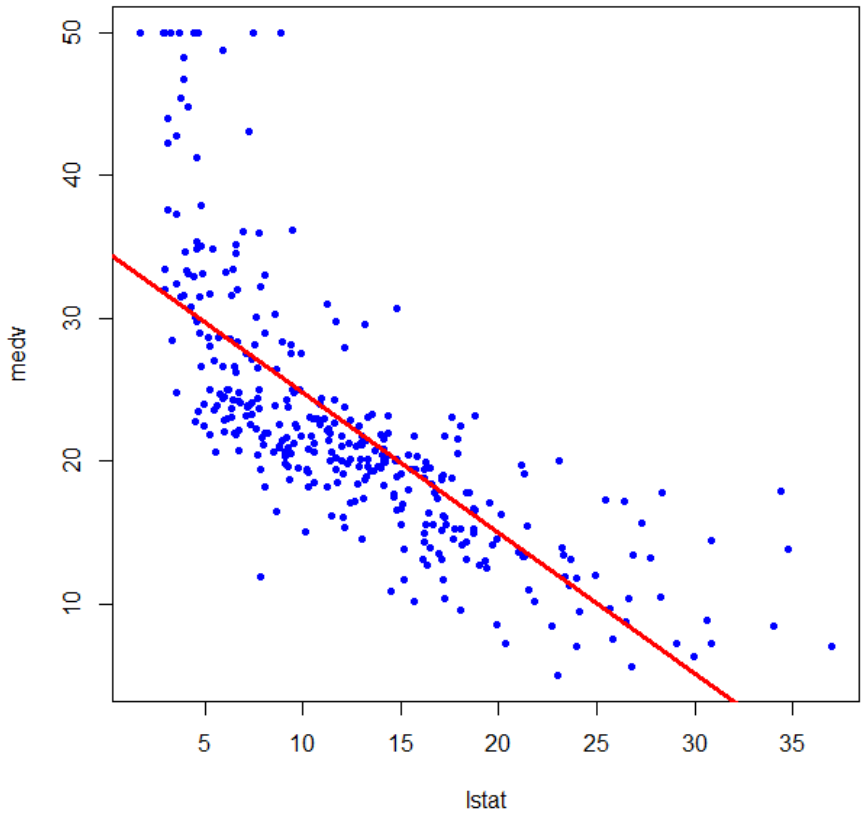
```
> names(reg1) # all information name of reg1
 [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values"
 "assign"        "qr"            "df.residual"   "xlevels"
[10] "call"          "terms"         "model"
> coef(reg1) # extract coefficients of reg1 (a vector)
(Intercept)       lstat
 34.6682590  -0.9836389
> confint(reg1) # get the interval estimation of parameters
                2.5 %     97.5 %
(Intercept) 33.347386 35.9891317
lstat       -1.076879 -0.8903987
```

- Prediction:

```
> predict(reg1, data.frame(lstat = (c(5, 10, 15))), interval = "confidence")
       fit      lwr      upr
1 29.75006 28.81134 30.68879
2 24.83187 24.15926 25.50448
3 19.91368 19.23671 20.59064
> predict(reg1, data.frame(lstat = (c(5, 10, 15))), interval = "prediction")
       fit       lwr      upr
1 29.75006 17.799263 41.70087
2 24.83187 12.899022 36.76472
3 19.91368  7.980582 31.84677
>
```

Easy example: simple LR

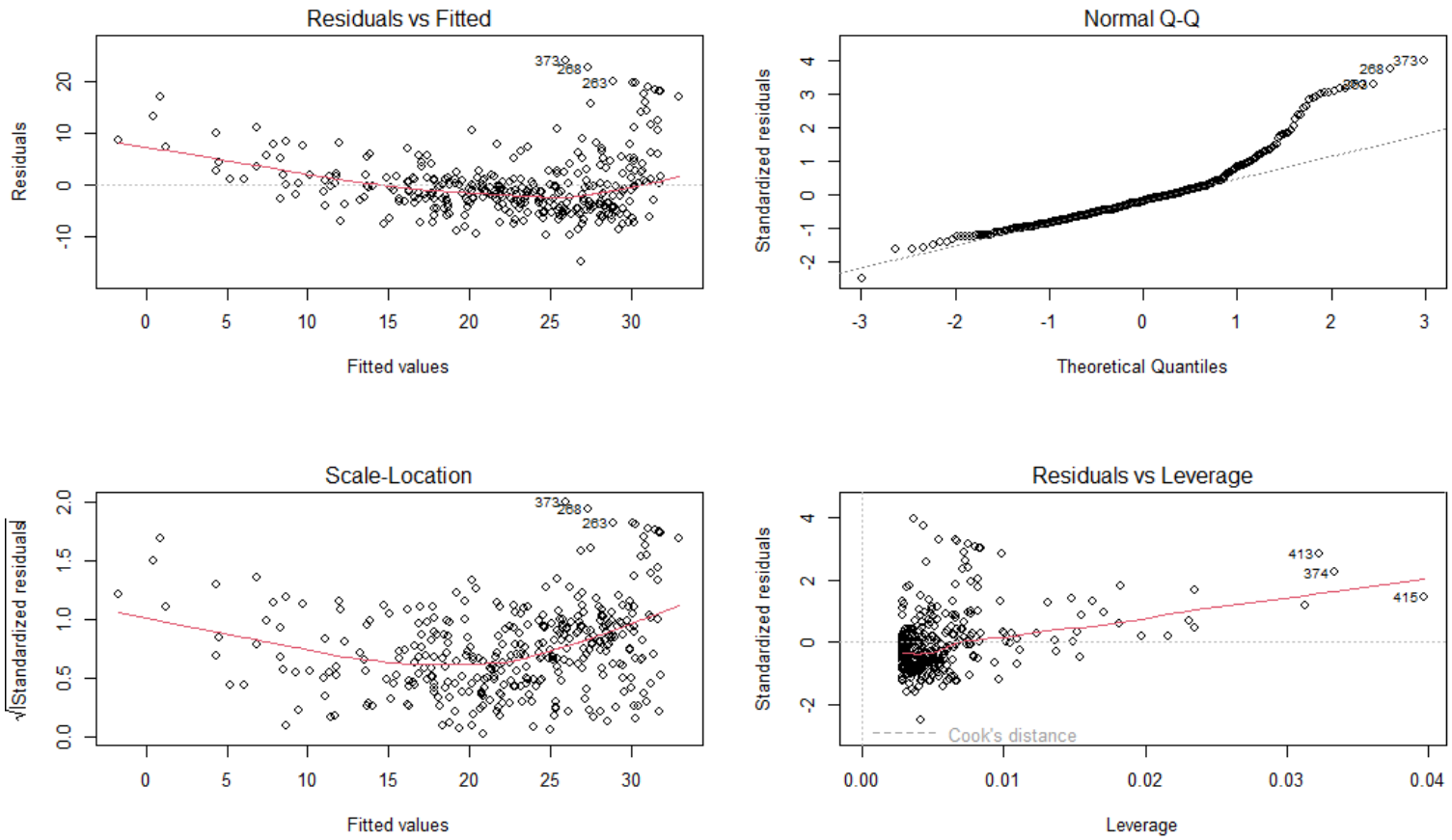
- Plot:



Easy example: simple LR

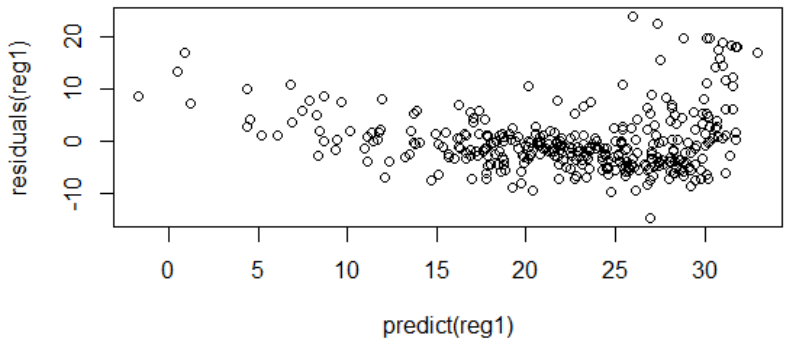
- Potential Problems:
  1. Nonlinearity
  2. Correlation of error term
  3. Non-constant variance of error term
  4. Outliers
  5. High-leverage point
  6. Collinearity (Don't worry in simple regression)
  7. Not normal

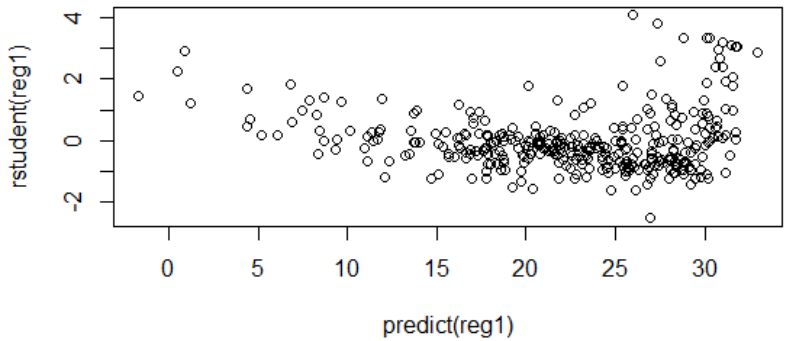
- Regression diagnostics (Read R in action Chapter 8):



Easy example: simple LR

- Regression diagnostics:



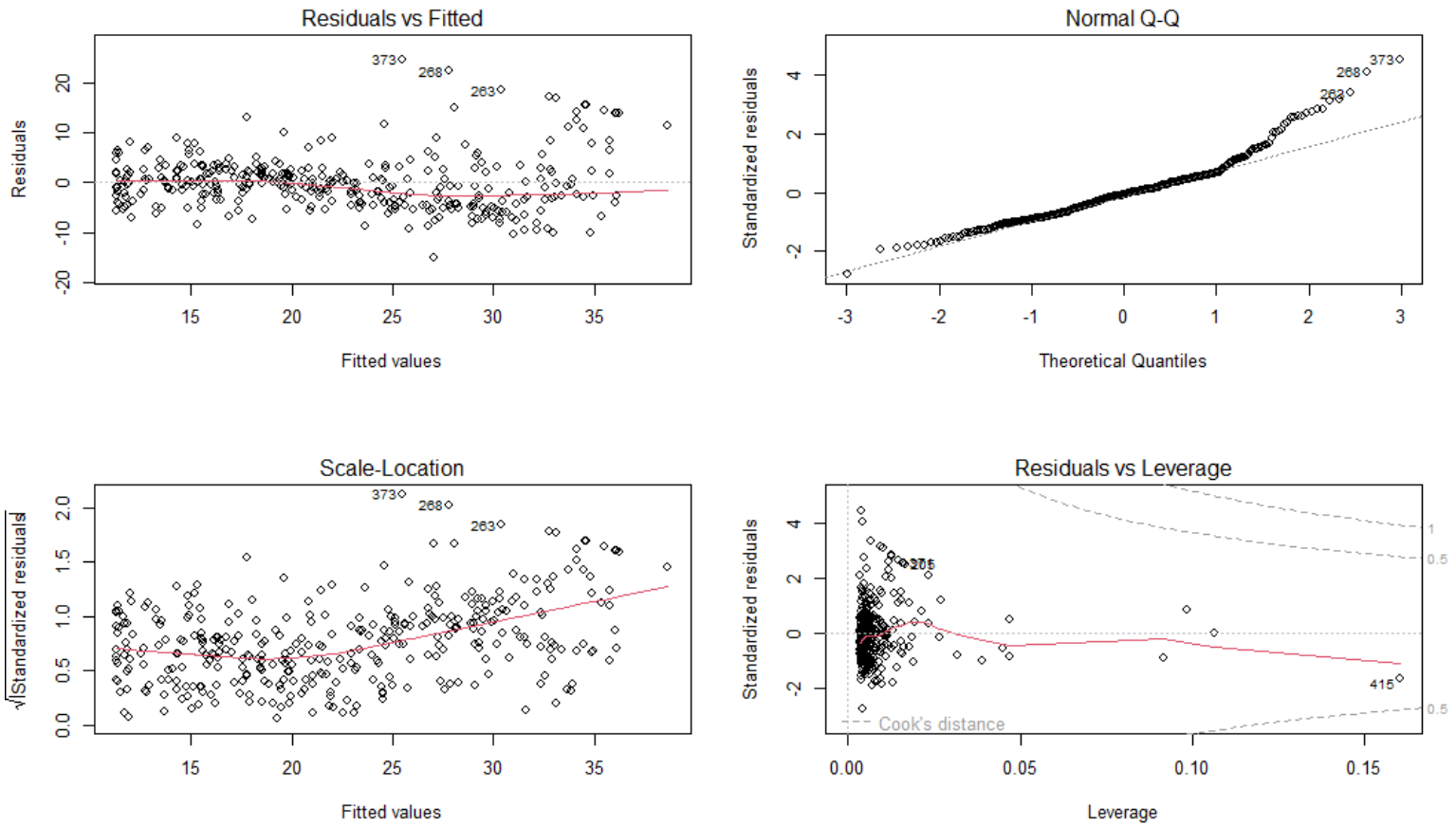


Easy example: simple LR

- Include the nonlinear part (but the model is still linear in parameters)

```
> reg2 <- lm(medv ~ lstat + I(lstat^2), data = Boston_train) # main regression
> summary(reg2)

Call:
lm(formula = medv ~ lstat + I(lstat^2), data = Boston_train)

Residuals:
     Min       1Q   Median       3Q      Max
-15.1361  -3.8298  -0.4067   2.3962  24.5507

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 42.490434   1.061563  40.026   <2e-16 ***
lstat       -2.293491   0.152044 -15.084   <2e-16 ***
I(lstat^2)   0.042168   0.004697   8.978   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.471 on 351 degrees of freedom
Multiple R-squared:  0.6342,    Adjusted R-squared:  0.6321
F-statistic: 304.2 on 2 and 351 DF,  p-value: < 2.2e-16
```

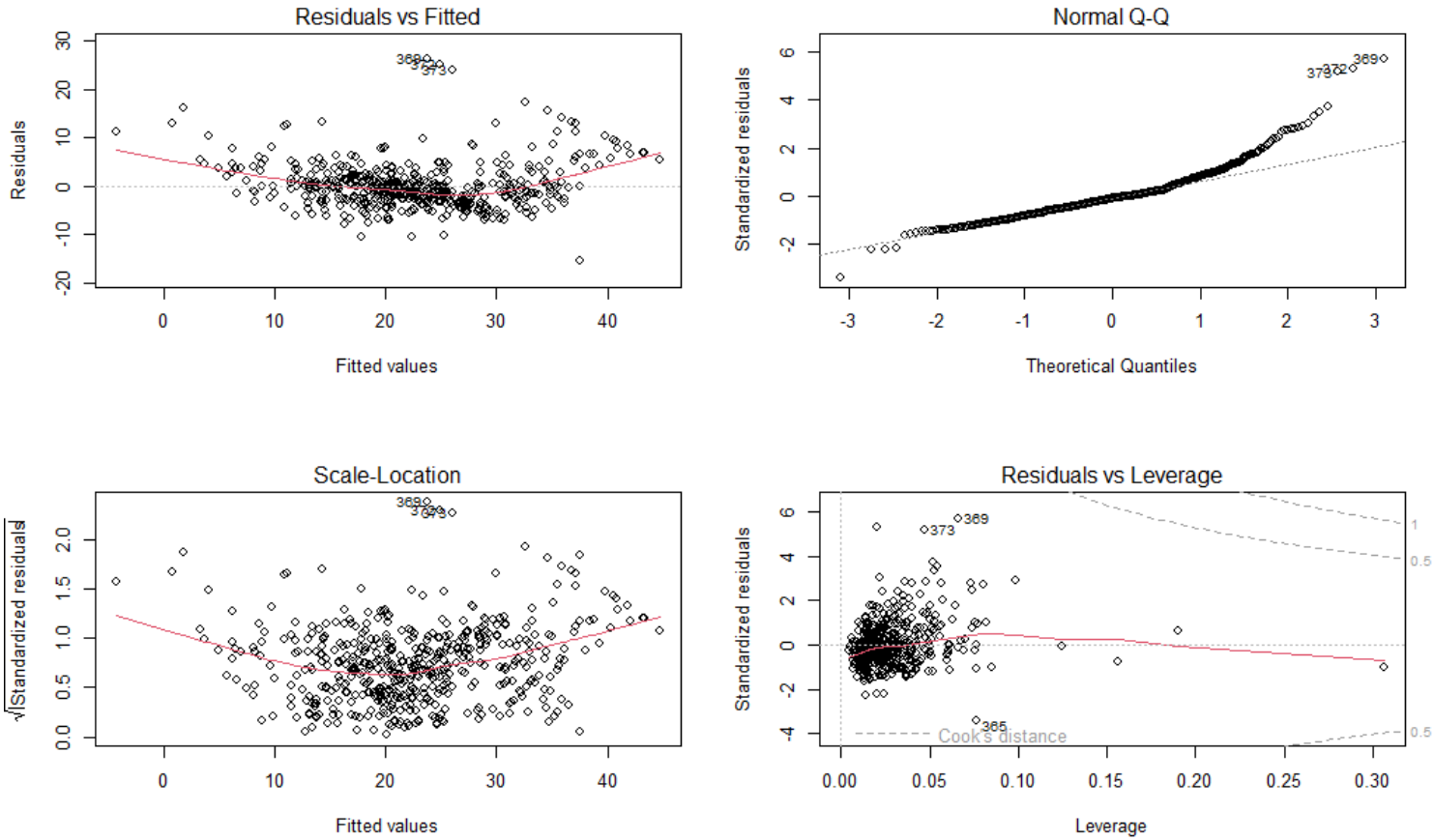
- Regression diagnostics:



- Then, let's run multiple linear regression:

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

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)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925 **
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -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
```

- Regression diagnostics:



- Regression diagnostics:

```
> vif(reg3)
    crim       zn    indus     chas      nox       rm      age
1.792192 2.298758 3.991596 1.073995 4.393720 1.933744 3.100826
     dis      rad      tax  ptratio    black    lstat
3.955945 7.484496 9.008554 1.799084 1.348521 2.941491
```

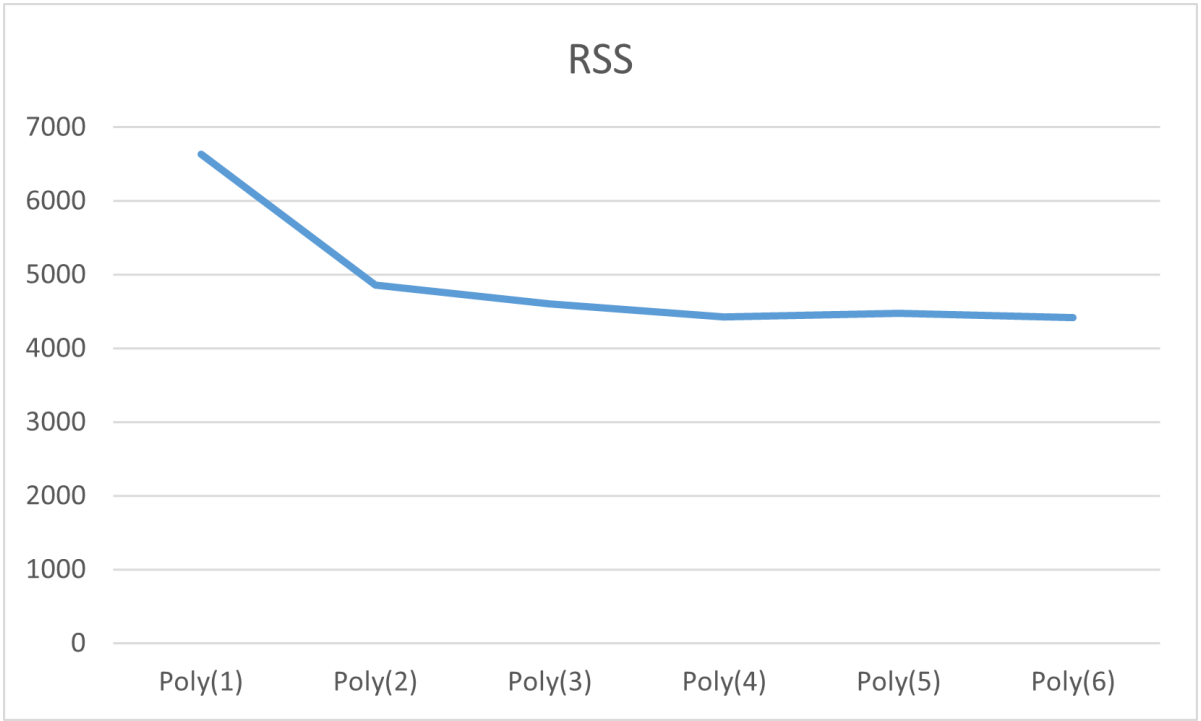
- If you want to include interaction terms of variables or nonlinear part:

```
# interactions
reg4 <- lm(medv ~ lstat * age, data = Boston_train)
reg5 <- lm(medv ~ lstat : age, data = Boston_train)

# nonlinear part
reg6 <- lm(medv ~ poly(lstat, 5), data = Boston_train)
reg7 <- lm(medv ~ log(rm), data = Boston_train)
```

- What we want to do:

```
comp_reg1 <- lm(medv ~ lstat, data = Boston_train)
comp_reg2 <- lm(medv ~ lstat + I(lstat^2), data = Boston_train)
comp_reg3 <- lm(medv ~ lstat + I(lstat^2) + I(lstat^3), data = Boston_train)
comp_reg4 <- lm(medv ~ lstat + I(lstat^2) + I(lstat^3) + I(lstat^4),
                data = Boston_train)
comp_reg5 <- lm(medv ~ lstat + I(lstat^2) + I(lstat^3) + I(lstat^4) +
                  I(lstat^5), data = Boston_train)
comp_reg6 <- lm(medv ~ lstat + I(lstat^2) + I(lstat^3) + I(lstat^4) +
                  I(lstat^5) + I(lstat^6), data = Boston_train)
stargazer(comp_reg1, comp_reg2, comp_reg3, comp_reg4, comp_reg5, comp_reg6,
          type = "html", title = "Example")
```

- regression results: see ".html" file.


RSS
7000
6000
5000
4000
3000
2000
1000
0
Poly(1)
Poly(2)
Poly(3)
Poly(4)
Poly(5)
Poly(6)