Chapter 4. Regression Analysis

- Simple linear regression
- Multiple linear regression
- Understanding regression results
- Nonlinear effects in linear regression
- Regression trees
- Bagging, random forests and boosting
- From global modelling to local fitting
- Regression analysis in R

Linear regression is one of the oldest and also the most frequently used statistical or data-mining methods

A useful tool for predicting a quantitative response based on some observable features/predictors/variables

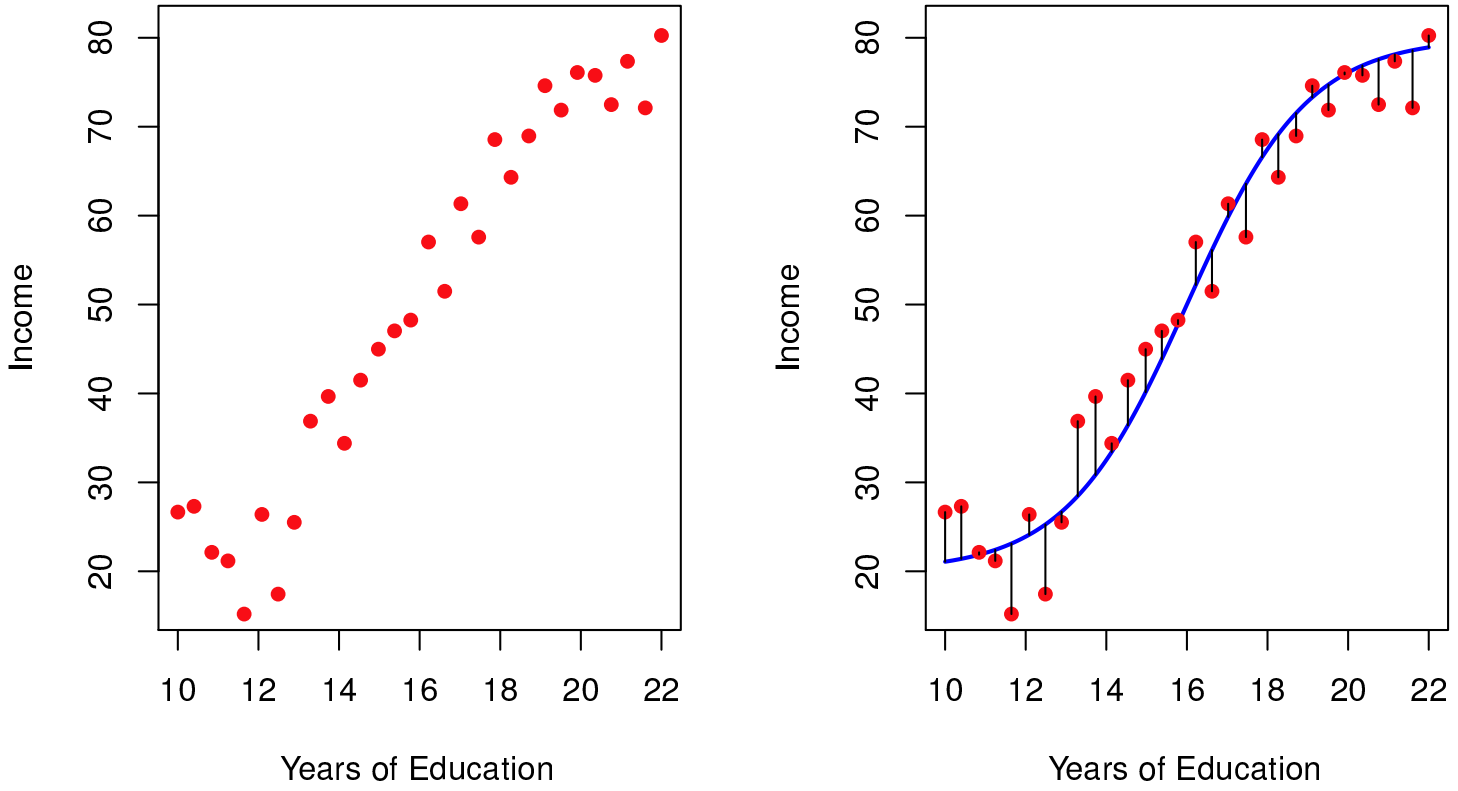
Many fancy data-mining methods can be viewed as the extensions of linear regression

Further reading:

James et al. (2013) Chapter 3 & Section 4.6,

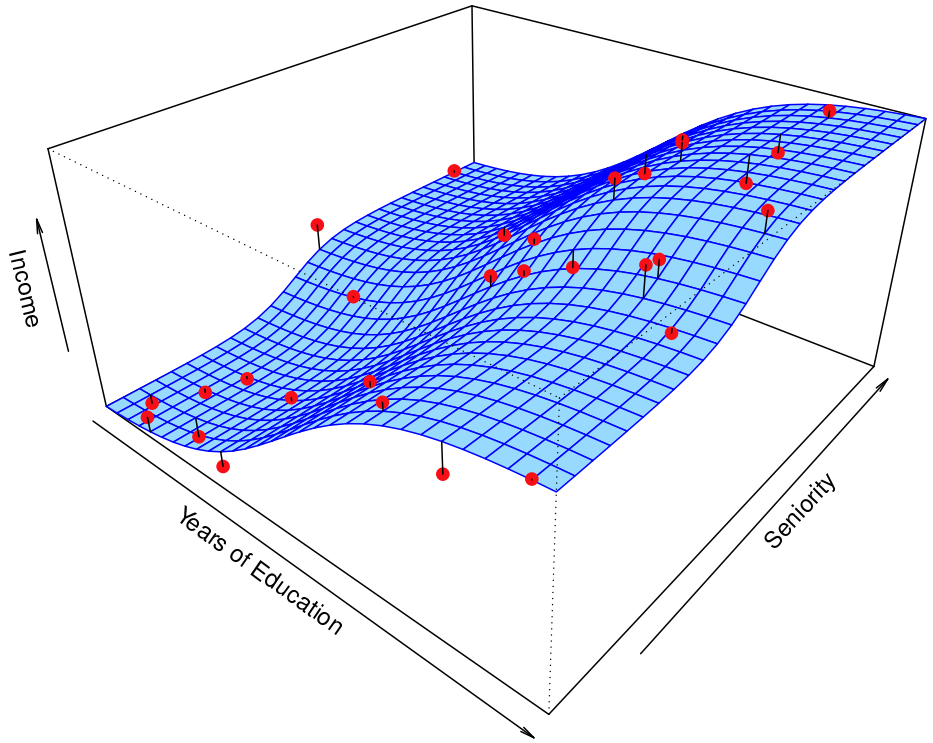
Provost and Fawcett (2013) Chapter 4.

Task of regression analysis: estimate the true curve from data points.



Left panel: Incomes of 30 individuals are plotted against their years of education.

Right panel: The curve (or regression curve) represents the true underlying relationship between income and years of education



Plot of income as a function of years of education and seniority. The blue surface represents the true relationship.

Task: to estimate the surface from the data.

Much harder! Data points are sparse: curse-of-dimensionality

Way-out: impose some parametric forms for the unknown surface, such as linear regression models.

Mincer Equation: How is one's earning related to human capital?

$$\log(Y) = \beta_0 + \beta_1 X + \beta_2 U + \beta_3 U^2 + \varepsilon,$$

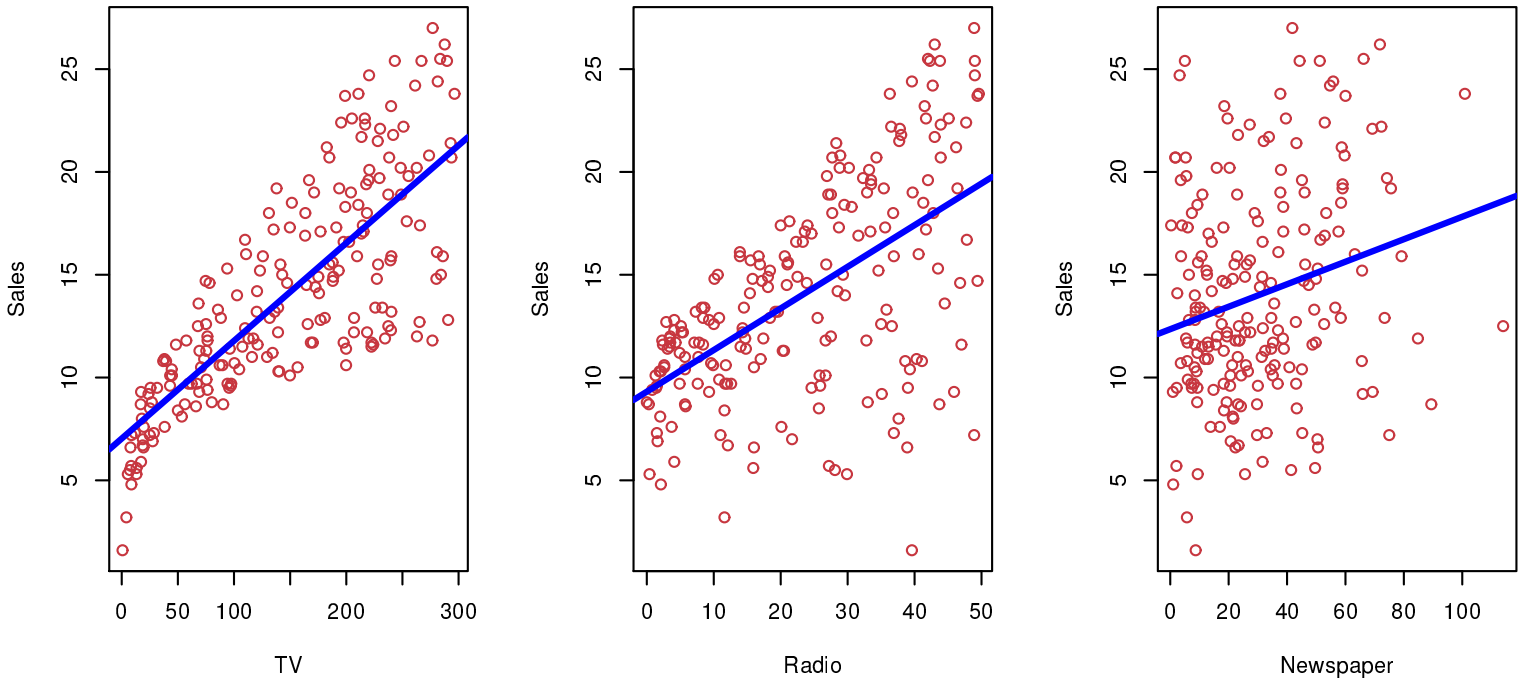
Y — earning
X — education capital: No. of years in school/university
U — experience capital: No. of years in employment

Rate of return to education: β_1

β_1, β_2 are positive, and β_3 tends to negative and small

This is a simple linear regression model.

We may even add an interaction term: $X \cdot U$



Sales are plotted against the ad budget in, respectively, TV, radio and newspaper. In each plot, the blue straight line is the regression estimator $Y = \widehat{\beta}_0 + \widehat{\beta}_1 X$.

Advertising in TV is most effective. Is there any added value to advertise in addition in radio and newspaper?

What can we learn from regression?

- is there a relationship between ad budget and sales?
- how strong is the relationship if there is?
- is the relationship linear?
- which media contribute to sales?
- how accurately can we estimate the effect of each medium on sales?
- how accurately can we predict the future sales?
- how should we distribute ad budget over different media? (synergy effect or interaction effect)

Simple line regression: $Y = \beta_0 + \beta_1 X + \varepsilon$

With n data points $(x_1, y_1), \cdots, (x_n, y_n)$, we calculate the LSE:

$$(\widehat{\beta}_0, \widehat{\beta}_1) = \arg\min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - x_i \beta_1)^2.$$

It can be shown that

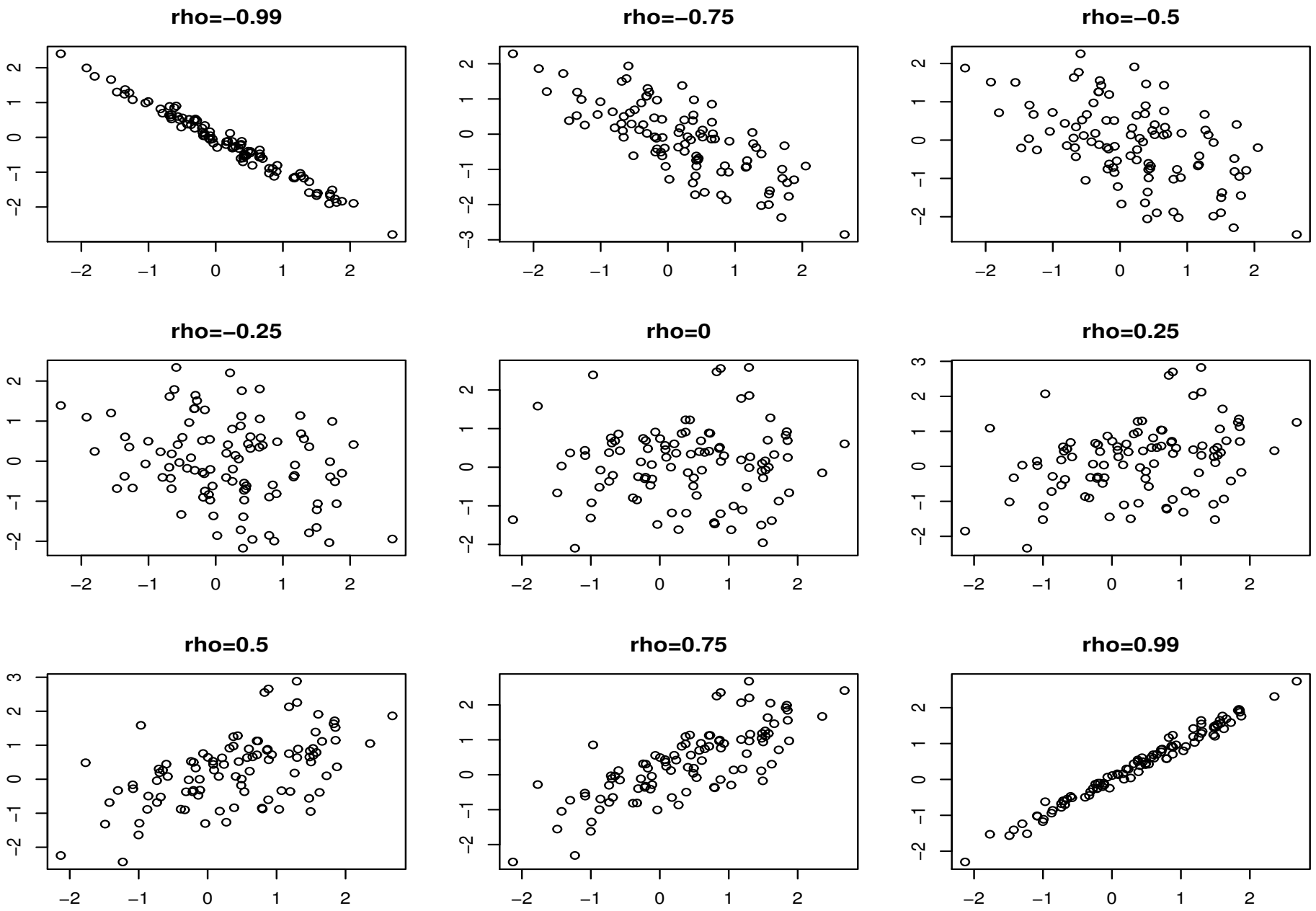
$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \qquad \widehat{\beta}_0 = \bar{y} - \bar{x}\widehat{\beta}_1,$$

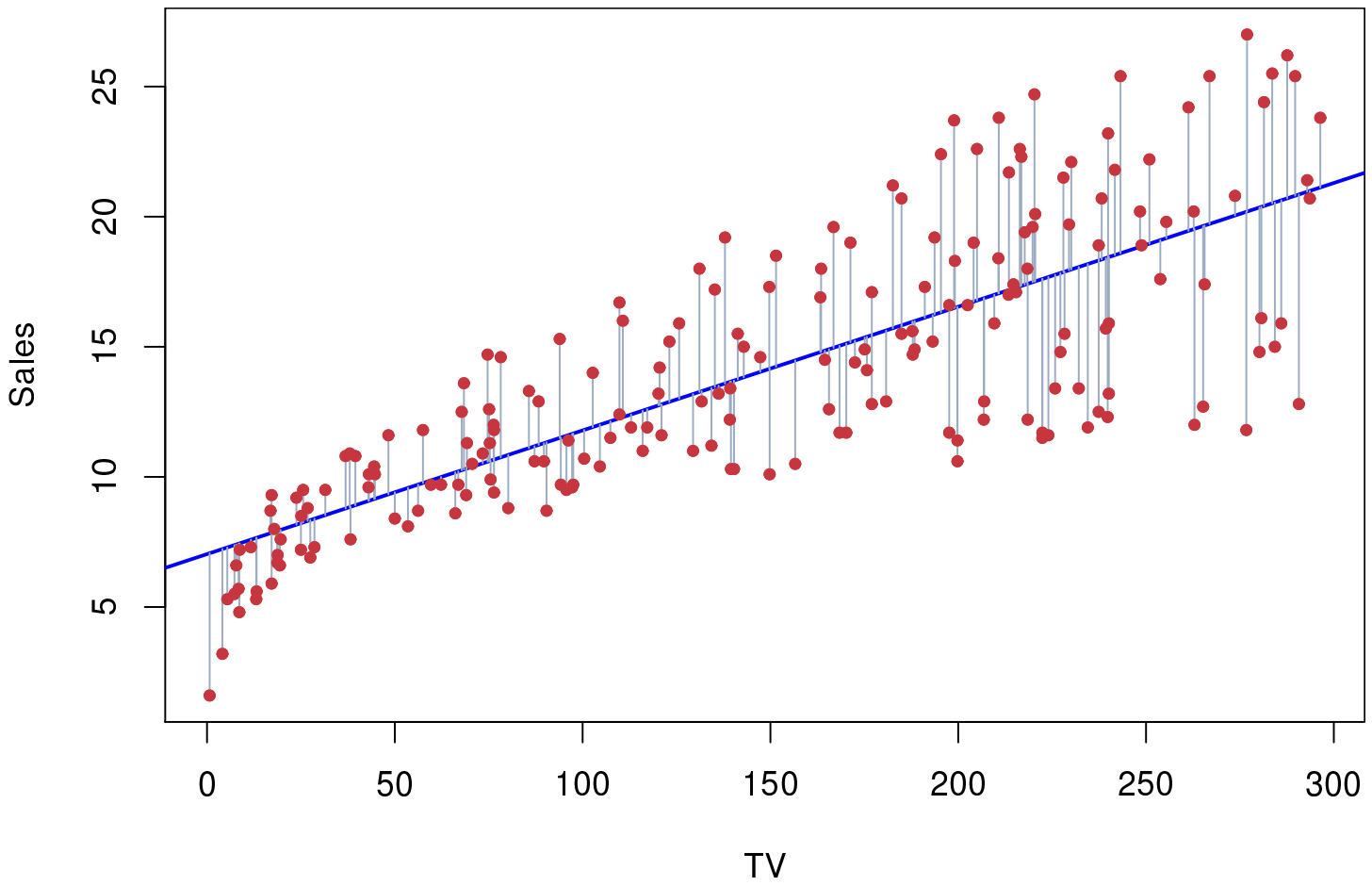
where $\bar{y} = n^{-1} \sum_i y_i$, $\bar{x} = n^{-1} \sum_i x_i$.

Interpretation: make RSS $= \sum_i \varepsilon_i^2$ as small as possible, where

$$\varepsilon_i = y_i - \beta_0 - x_i \beta_1, \qquad i = 1, \cdots, n.$$

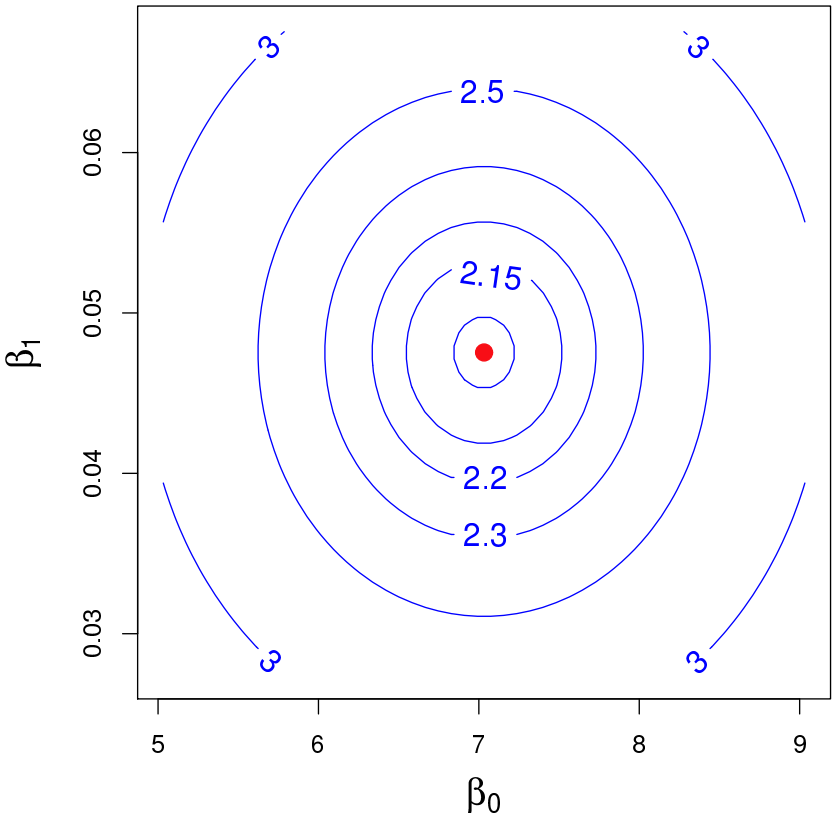
Sample correlation: $\widehat{\rho}_{x,y} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$. Note $\widehat{\beta}_1 = \widehat{\rho}_{x,y} \sqrt{\frac{\sum_i (y_i - \bar{y})^2}{\sum_i (x_i - \bar{x})^2}}$.

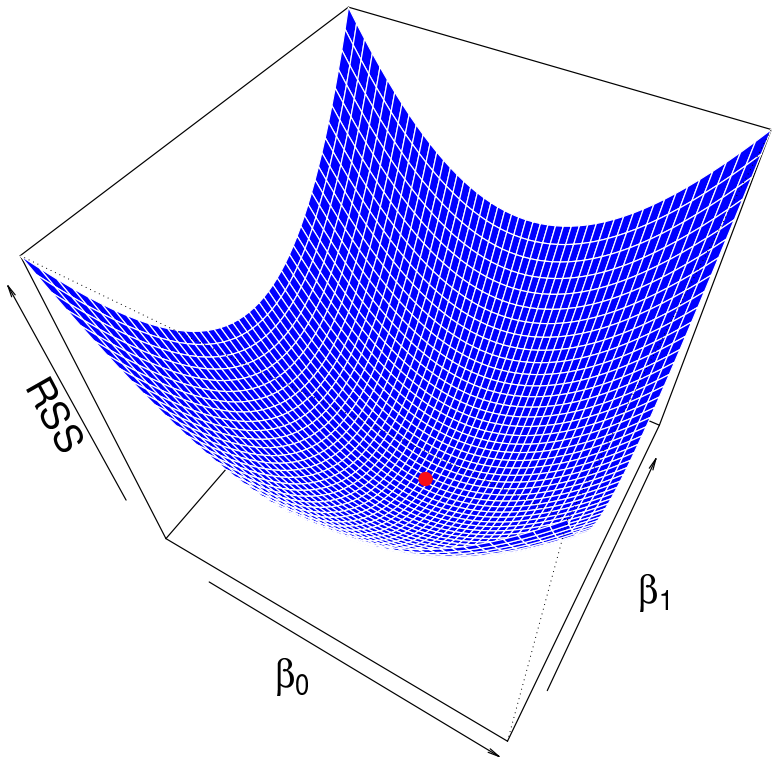

rho=−0.99
rho=−0.75
rho=−0.5
rho=−0.25
rho=0
rho=0.25
rho=0.5
rho=0.75
rho=0.99



Fitting: Sales $= \beta_0 + \beta_1 \text{TVad} + \varepsilon$. LSE: $\widehat{\beta}_0 = 7.03$, $\widehat{\beta}_1 = 0.0475$

Is it right to conclude that increasing one unit of `TV` budget leads to an increase in `sales` by 0.0475 unit?


3
2.5
2.15
2.2
2.3
3
3
3
0.03
0.04
0.05
0.06
β_1
5
6
7
8
9
β_0


RSS
β_0
β_1

How accurate are $\widehat{\beta}_0$, $\widehat{\beta}_1$?

Assumption: $E(\varepsilon) = 0, \quad \text{Var}(\varepsilon) = \sigma^2.$ Then

$$\widehat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2.$$

Standard errors:

$$\text{SE}(\widehat{\beta}_0) = \widehat{\sigma}\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^{1/2}, \qquad \text{SE}(\widehat{\beta}_1) = \left(\frac{\widehat{\sigma}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^{1/2}$$

Note. $\text{SE}(\widehat{\beta}_1)$ is an approximation for $\{\text{Var}(\widehat{\beta}_1)\}^{1/2}$.

95% Confidence intervals (error bars): $\widehat{\beta}_j \pm 1.96 \cdot \text{SE}(\widehat{\beta}_j)$, or simply,

$$\widehat{\beta}_j \pm 2 \cdot \text{SE}(\widehat{\beta}_j), \qquad j = 1, 2$$

For the model Sales $= \beta_0 + \beta_1 \text{TVad} + \varepsilon$, the 95% Confidence interval is $[6.130, 7.935]$ for β_0, and $[0.042, 0.053]$ for β_1.

Interpretation: In the absence of any advertising, sales will on average fall between 6.130 and 7.935. Furthermore, an increase of 1000 units in TV advertising is likely to increase sales between 42 and 53 units.

Since $\widehat{\beta}_1 = 0.0475$ is so small, is it possible that $\beta_1 = 0$ in the sense that there is no relationship between sales and TV advertising.

Hypothesis tests. To test the null hypothesis

H_0 : there is no relationship between sales and TV advertising

which is equivalently to $H_0 : \beta_1 = 0$

Since the 95% confidence interval for β_1 does not contain 0, we reject H_0.

How much is the sales fluctuation due to TV advertising?

Total SS: $\sum_{i=1}^{n}(y_i - \bar{y})^2$

Regression SS: $\sum_{i=1}^{n}(\widehat{y}_i - \bar{y})^2$, where $\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$

Residual SS: $\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$

It can be shown that

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\widehat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2.$$

Regression correlation coefficient:

$$R = \Big(\frac{\text{Regression SS}}{\text{Total SS}}\Big)^{1/2} = \Big(1 - \frac{\text{Residual SS}}{\text{Total SS}}\Big)^{1/2}.$$

Then $R \in [0, 1]$.

Interpretation: $100R^2$ is the percentage of the total variation of Y explained by the regressor X.

Adjusted regression correlation coefficient:

$$R_{adj} = \Big(1 - \frac{(\text{Residual SS})/(n-2)}{(\text{Total SS})/(n-1)}\Big)^{1/2}.$$

For the model $\text{Sales} = \beta_0 + \beta_1 \text{TVad} + \varepsilon$,

$$\widehat{\sigma} = 3.26, \qquad R^2 = 61.2\%$$

Multiple Linear Regression: $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$.

Available data: $(y_i, x_{i1}, \cdots, x_{ip})$, $i = 1, \cdots, n$.

LSE: $\widehat{\beta}_0, \widehat{\beta}_1, \cdots, \widehat{\beta}_p$ are obtained by minimizing

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2.$$

Sum of squared residuals:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2, \quad \text{where } \widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \cdots + \widehat{\beta}_p x_{ip}.$$

Square of regression correlation coefficient:

$$R^2 = 1 - \frac{\text{RSS}}{\sum_i (y_i - \bar{y})^2}, \qquad R^2_{adj} = 1 - \frac{\text{RSS}/(n-p-1)}{\sum_i (y_i - \bar{y})^2/(n-1)}$$

Fitting: sales $= \beta_0 + \beta_1 \cdot \text{TV} + \beta_2 \cdot \text{radio} + \beta_3 \cdot \text{newspaper} + \varepsilon$

	Coefficient	Std error	t-statistic	P-value
Intercept	2.939	0.3119	9.24	$<$0.0001
TV	0.046	0.0014	32.81	$<$0.0001
ratio	0.189	0.0086	21.89	$<$0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

1. Given the budget for the two other media, an increase of one unit in TV budget will bring in an increase of 0.046 unit in sales, an increase of one unit in radio budget will bring in an increase of 0.189 unit in sales.

   Is it more effective to advertise in ratio than in TV?

   Not necessarily, the fitted model is valid only within the range of the learning data (i.e. observations).

   Caution should be exercised when extrapolating a fitted model outside of observed range

2. $\widehat{\beta}_3 = -0.001$ with the 95% confidence interval $-0.001 \pm 2 \times 0.0059 = [-0.0128, 0.0108]$. The interval contains the value 0. Hence we cannot reject the hypothesis $H_0 : \beta_3 = 0$.

   Having advertising on TV and radio, the effect of advertising on newspaper is not significant.

   However the cross correlations are

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

   Note the correlation between ratio and newspaper budgets is 0.3541!

The fitted univariate models are:

	Coefficient	Std error	t-statistic	P-value
Intercept	7.0325	0.4758	15.36	$<$0.0001
TV	0.0475	0.0027	17.67	$<$0.0001
Intercept	9.312	0.563	16.54	$<$0.0001
radio	0.203	0.020	9.92	$<$0.0001
Intercept	12.351	0.621	19.88	$<$0.0001
newspaper	0.055	0.017	3.30	0.00115

Thus, the estimated coefficients for TV and ratio are about the same in both the multiple regression and the simple univariate regression.

The coefficients for newspaper in the multiple regression and the univariate regression are significantly different.

Two possible interpretations:

(a) The effect from advertising in newspaper is encapsulated in that from TV or radio, i.e. the sales will increase by advertising in newspaper even if no advertising in both TV and radio

(b) Advertising on newspaper has no effect on sales. The significance in the univariate regression is due to the significant correlation between newspaper budget and radio budget, i.e. newspaper advertising acts as a surrogate for ratio advertising.

Common sense would suggest that (a) is more likely the case for this example, or not? (as only fewer people read news papers those days)

3. Since newspaper is not significant, we can refine the model using TV and radio only.

Variable selection: How many variables should be selected in the model?

A general principle: choose the model which minimizes a certain criterion defined as, for example,

(Goodness of fit of model) + (Penalty for model complexity)

- *Forward selection*. Starting with the model with a intercept only, add one variable each time such that the added variable leads to the maximum reduction in residual sum of squares (RSS). Stop according to a certain criterion.

- *Backward selection*. Start with the full model, delete each time one variable such that the deleted variable causes the minimum increase in RSS. Stop according to a certain criterion.

- *Stepwise selection*. After adding each variable, delete all *redundant* variables according to an appropriate criterion before

adding a new variable. Stop when no variables can be added or deleted.

Note. None of the above procedures will give you the overall optimal model. They are tradeoff between searching for a good model and computation efficiency. The stepwise selection procedure proves to be more effective.

4. How well the various models fit the data?

   Two simple measures: R^2 and $\hat{\sigma}$

   Note. Neither R^2 nor $\hat{\sigma}$ can be used as a sole measure for variable selection!

   For regression models for `sales`,

regressors	(T, r, n)	(T, r)	T	r	n
R^2	0.8972	0.89719	0.6119	0.332	0.05212
$\hat{\sigma}$	1.686	1.681	3.259	4.275	5.092

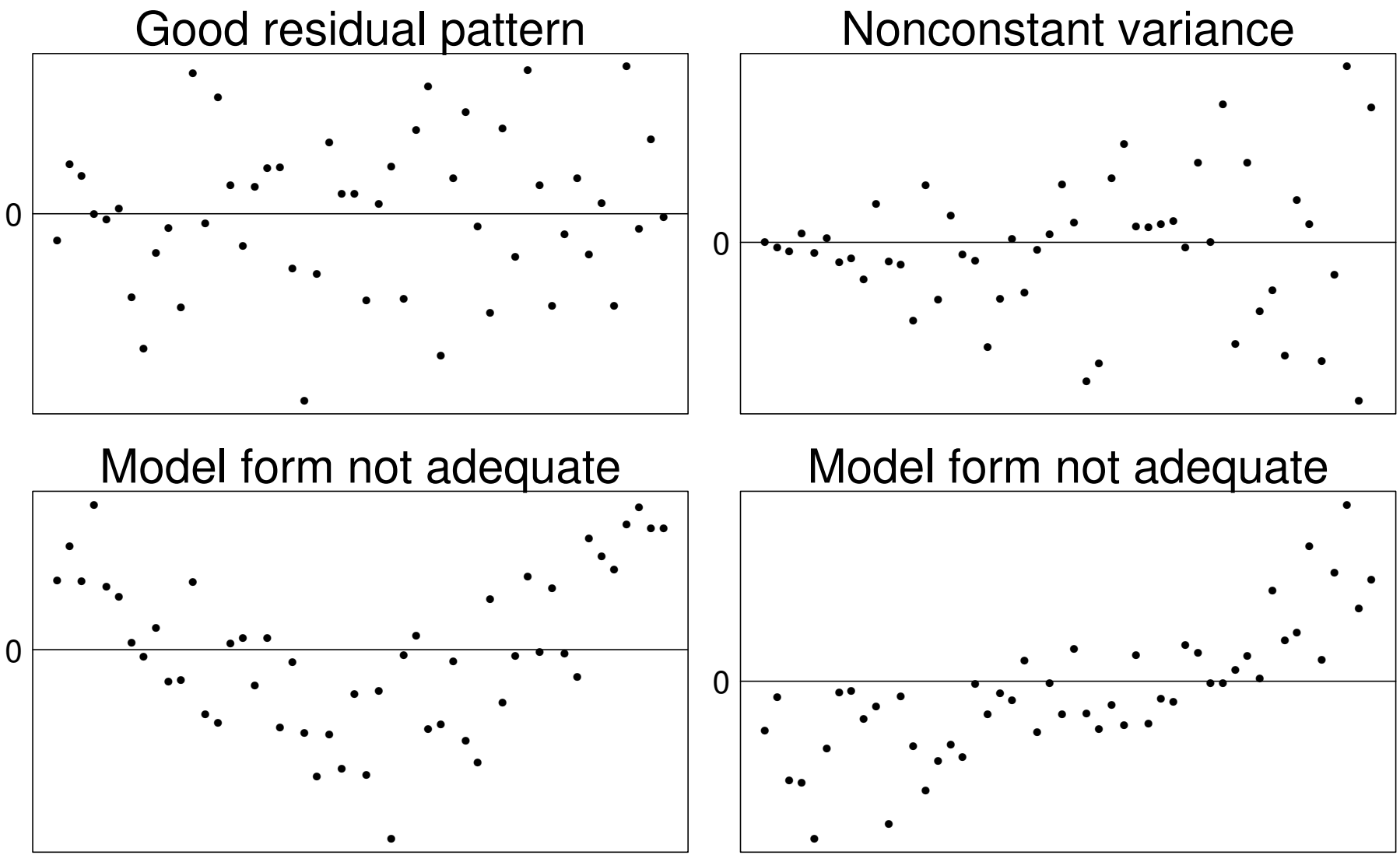
The model with two regressors (`TV`, `ratio`) fits the data better.

5. Goodness-of-fit: checking residuals $\widehat{\varepsilon}_i = y_i - \widehat{y}_i,\ i = 1, \cdots, n$.

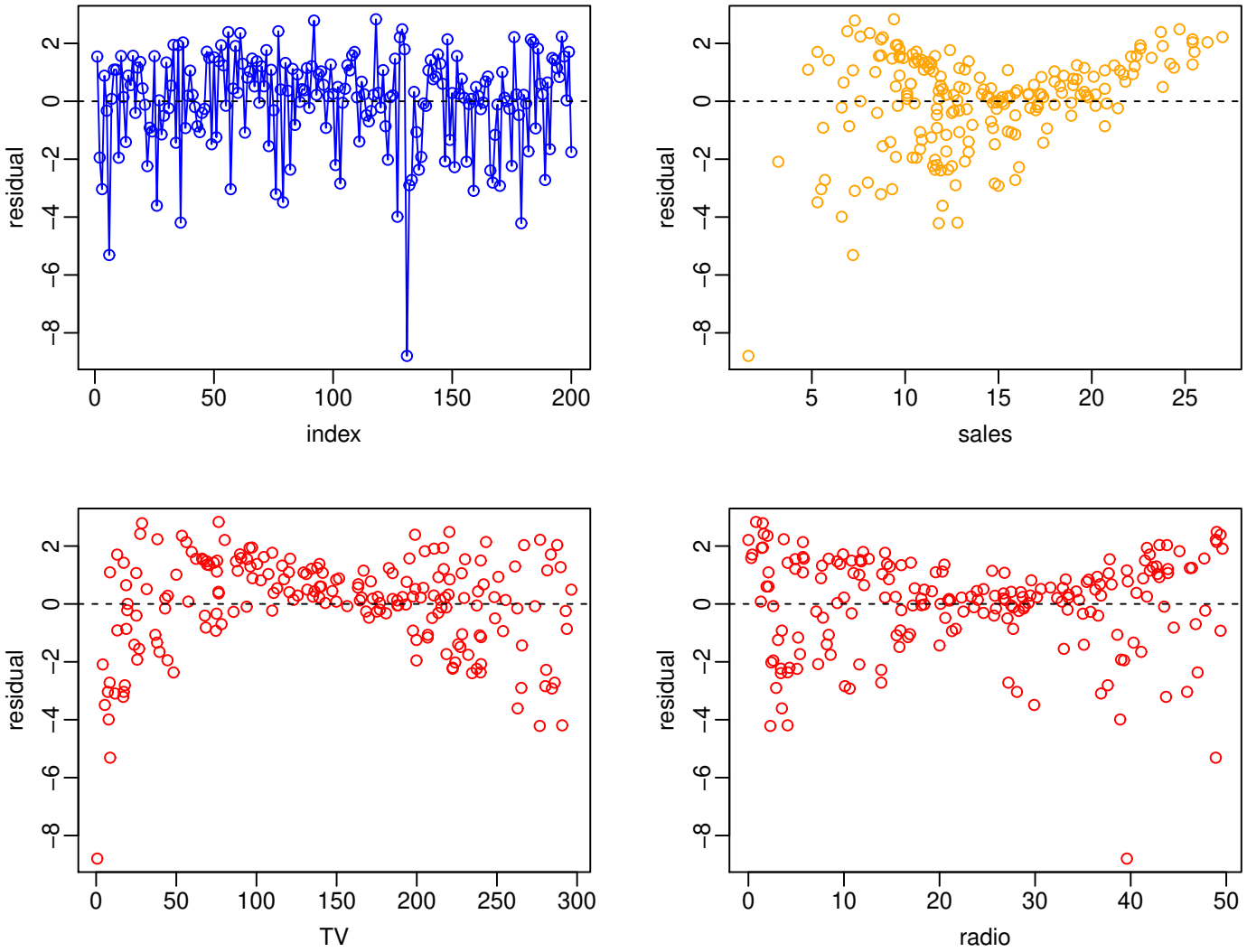
   A good fitting leads to patternless residuals.

   Plot residuals against index, y_i or $x_{i1}, \cdots, x_{ip}$.

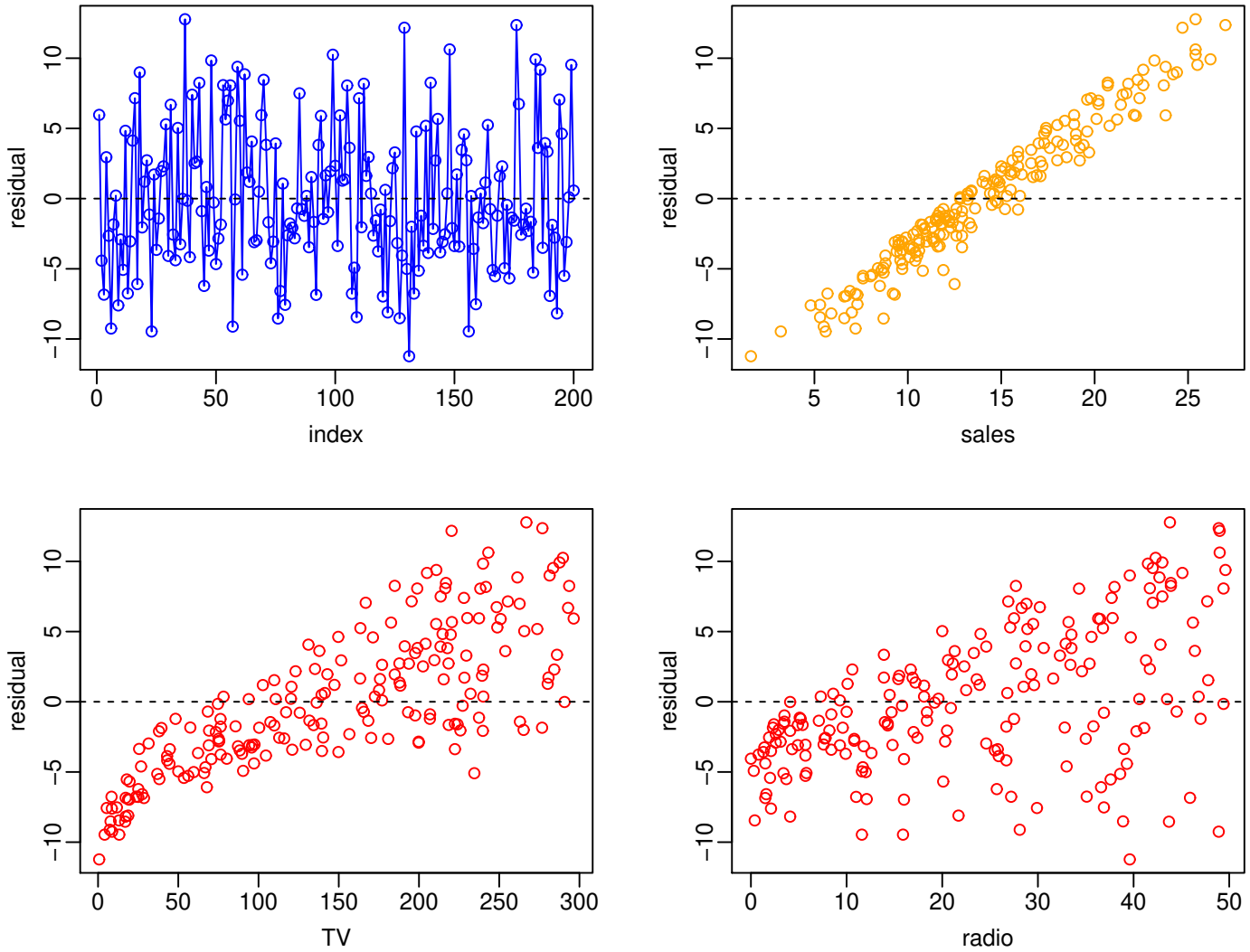
   Powerless in detecting overfitting!


Good residual pattern
0
Nonconstant variance
0
Model form not adequate
0
Model form not adequate
0

Residuals: $\texttt{sales} - (2.9211 + 0.0458\texttt{TV} + 0.1880\texttt{radio})$



Residuals: sales − (12.3514 + 0.0547newspapge)



6. Prediction

Given new values $x_1, \cdots, x_p$, we can predict the corresponding y by

$$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \cdots + \widehat{\beta}_p x_p.$$

The error in predicting y by $\widehat{y}$ may be caused by 3 sources:

(a) Estimation errors in $\widehat{\beta}_0, \widehat{\beta}_1, \cdots, \widehat{\beta}_p$.

This type of errors can be quantified by constructing a confidence interval for

$$E(y) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p.$$

(b) Unobserved error $\varepsilon = y - E(y)$.

By estimating $\sigma^2 = \mathrm{Var}(\varepsilon)$, we can enlarge the confidence interval above to a predictive interval.

(c) Model bias, i.e. $E(y)$ may not be linear in $x_1, \cdots, x_p$.

This is more difficult to quantify. Typically a linear model is regarded as an approximation.

With the model $\texttt{sales} = 2.9211 + 0.0458\texttt{TV} + 0.1880\texttt{radio}$, the predicted sales for spending 100 on TV and 20 on radio is 11.26, with the 95% confidence interval $[10.99, 11.53]$ and the 95% predictive interval $[7.93, 14.58]$.

Note. Confidence interval is for $E(y)$, i.e. spending 100 on TV and 20 on radio over many cities, the average sales over those cities will fall between 10.99 and 11.53 (with the probability 95%).

Predictive interval is for y, i.e. spending 100 on TV and 20 on radio in one city, the sales will fall between 7.93 and 14.58 (with the probability 95%).

7. Nonlinear regressors.

Linear regression models are linear in coefficients $\beta_0, \beta_1, \cdots, \beta_p$, while regressors $X_1, \cdots, X_p$ can be replaced by any known functions of them.

The capacity is far beyond linear relationship, as the regressors $x_1, \cdots, x_p$ may be

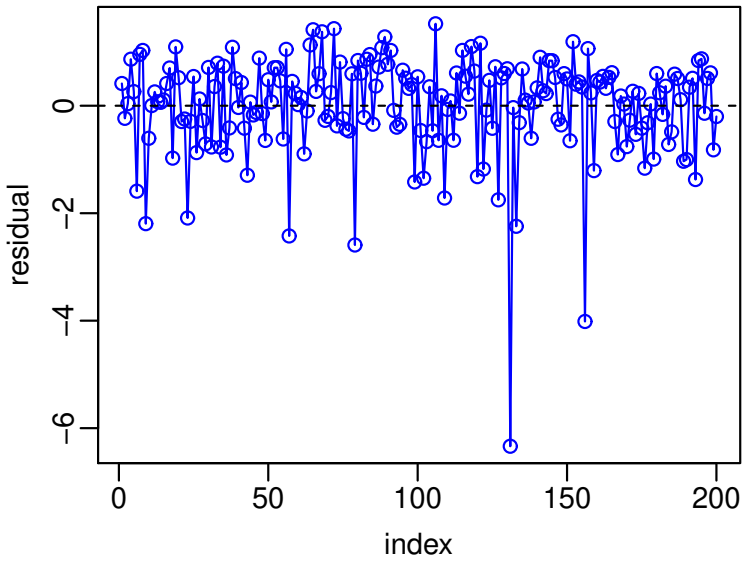
- quantitative inputs
- transformations of quantitative inputs, such as log, square-root etc
- interactions between variables, e.g. $x_3 = x_1 x_2$
- basis expansions, such as $x_2 = x_1^2, x_3 = x_1^3, \cdots$
- numeric or "dummy" coding of the levels if qualitative variables.

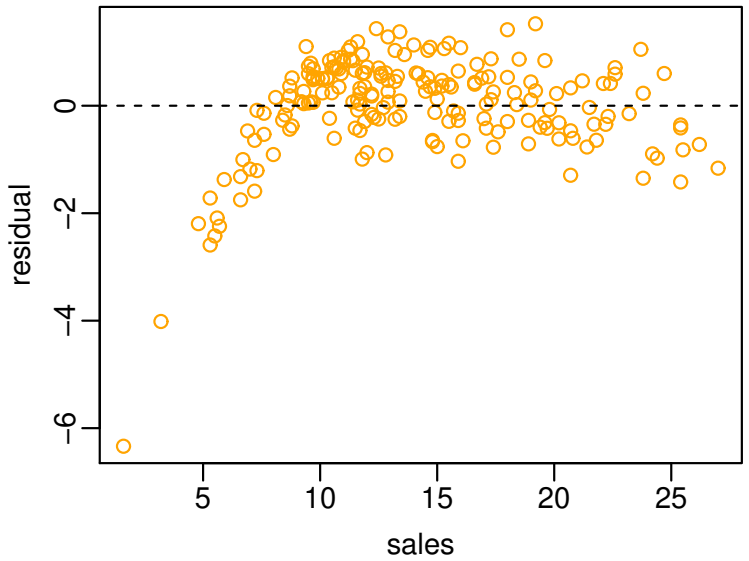
Fitting: $\texttt{sales} = \beta_0 + \beta_1 \texttt{TV} + \beta_2 \texttt{radio} + \beta_3 \texttt{TV} \cdot \texttt{radio} + \varepsilon$

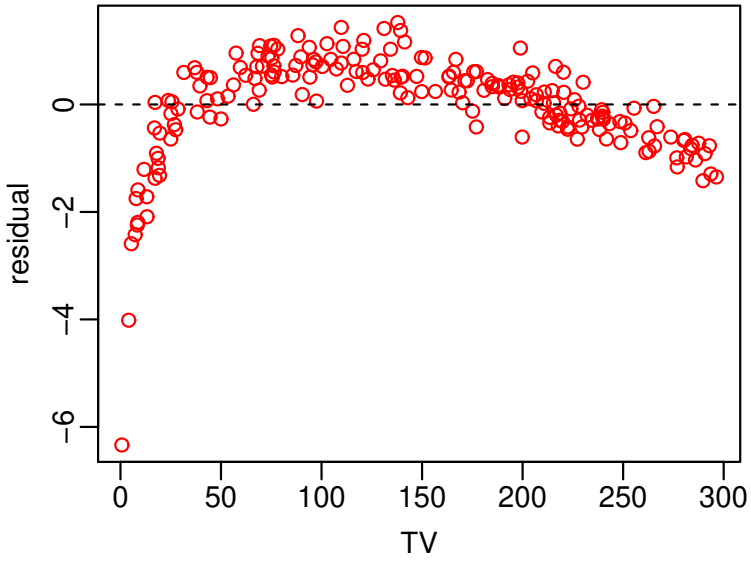
	Coefficient	Std error	t-statistic	P-value
Intercept	6.7502	0.248	27.23	$<$0.0001
TV	0.0191	0.002	12.70	$<$0.0001
radio	0.0289	0.009	3.24	0.0014
TV·radio	0.0011	0.000	20.73	$<$0.0001

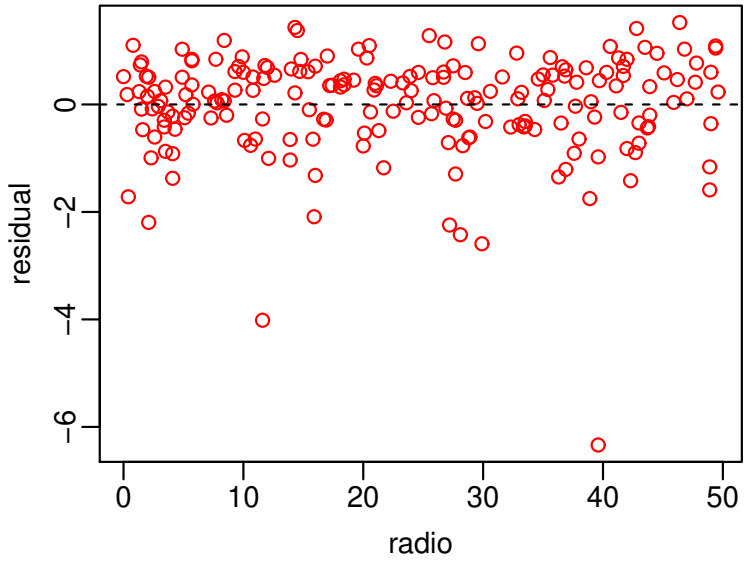
$R^2 = 0.9678,$ $\widehat{\sigma} = 0.9435.$

A better fitting???


residual
0
−2
−4
−6
0
50
100
150
200
index


residual
0
−2
−4
−6
5
10
15
20
25
sales


residual
0
−2
−4
−6
0
50
100
150
200
250
300
TV


residual
0
−2
−4
−6
0
10
20
30
40
50
radio

A market plan for sales

1. There is a clear relationship between sales and advertising budget, as the hypotheses $\beta_1 = 0$ (TV) and $\beta_2 = 0$ (radio) is overwhelmingly rejected. But there is little impact on sales by advertising on newspaper.

2. The strength of the relationships can be measured by either R^2 and $\widehat{\sigma}$ from the judiciously selected models. The about 90% of variation in sales is due to the advertising on TV and radio. The recommended model for this data set is

$$\texttt{sales} = 2.9211 + 0.0458\texttt{TV} + 0.1880\texttt{radio}.$$

3. The effect on sales from the advertising can be reflected by the estimates for β_1 (TV) and β_2 (radio), or more precisely their confidence intervals (0.043, 0.049) and (0.172, 0.206). For example,

an increase of 1 unit budget in TV advertising would lead to an increase of sales between 0.043 and 0.049 unit. But one should be cautious in extrapolating the results out of the observed range.

4. We can predict the future sales based on the above model. There are two types intervals for gauging the prediction errors: confidence interval for predicting sales over many cities, and predictive interval for predicting sales for one city.

Polynomial regression: an illustration by example

The data set `Auto.txt` contains various indices for 387 cars. Let us consider the relationship between `mpg` (gas mileage in miles per gallon) versus `horsepower`.

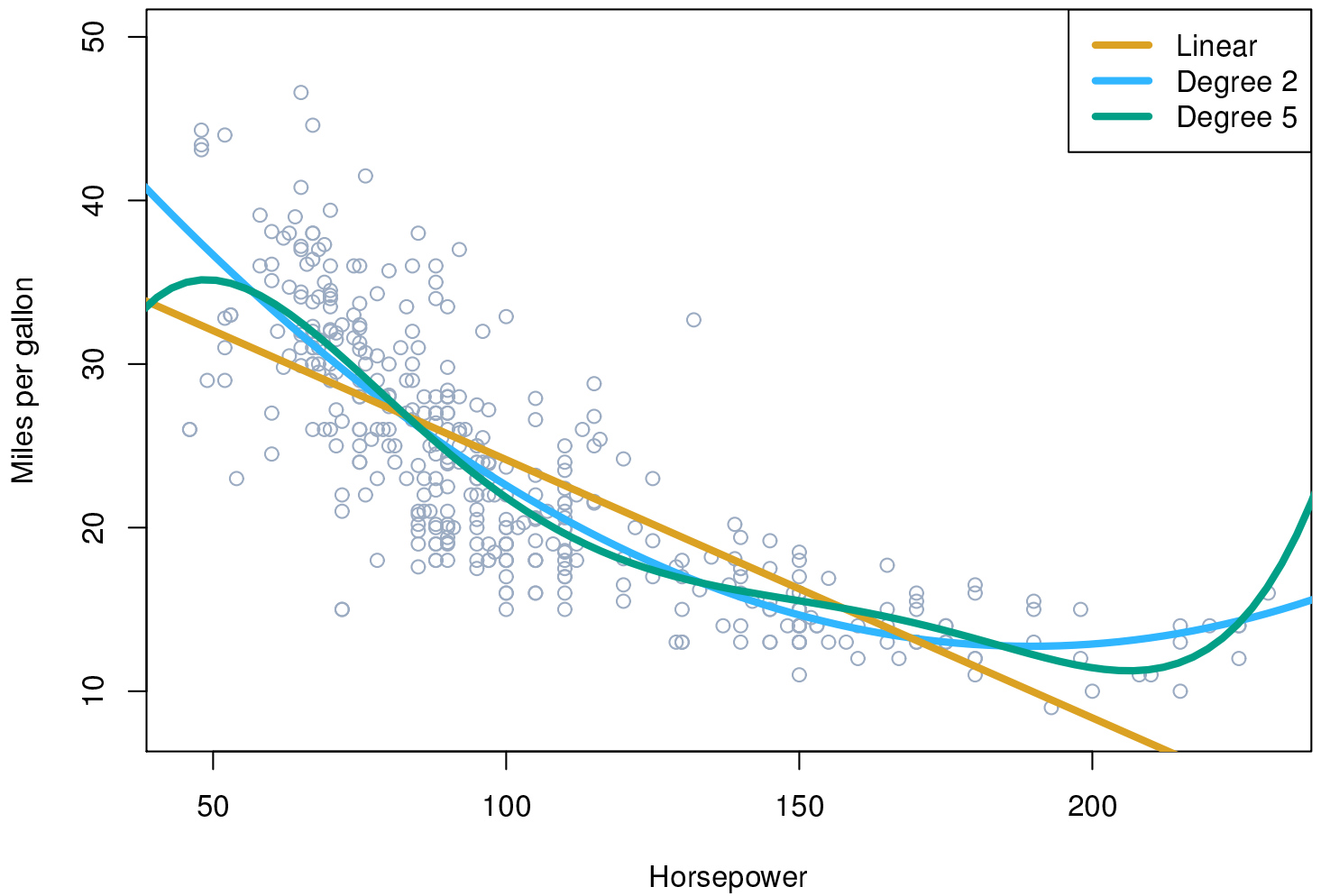
Looking at scatter plots, the relationship does not appear to be linear. We fit polynomial regression:

$$\texttt{mpg} = \beta_0 + \beta_1 \texttt{hpower} + \cdots + \beta_p \, \texttt{hpower}^p + \varepsilon$$

for $p = 1, 2, \cdots$.

The results for $p = 2$ are listed below

	Coefficient	Std error	t-statistic	P-value
Intercept	56.9001	1.8004	31.6	$<$0.0001
horsepower	-0.4662	0.0311	-15.0	$<$0.0001
horsepower^2	0.0012	0.0001	10.1	$<$0.0001



Extrapolation of polynomial regression can be explosive!

Linear Regression in R

We use dataset `Boston` in package `MASS` as an illustration.

```
> install.packages("MASS")
> library(MASS)
> names(Boston)
 [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"     "dis"
 [9] "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
> View(Boston)
```

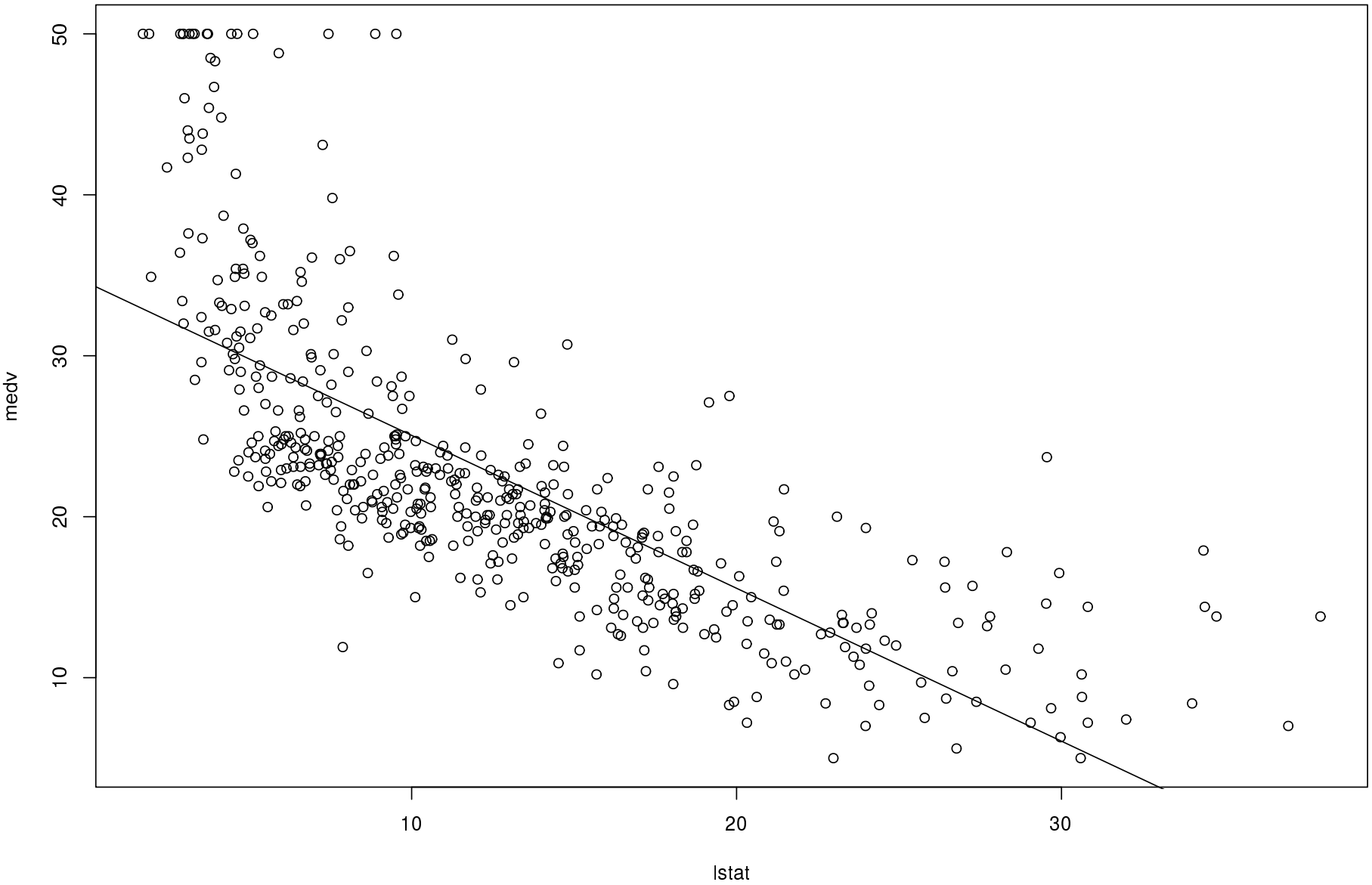
The data record `medv` (median house value) for 506 neighbourhoods around Boston, together with other 13 variables including `rm` (average number of rooms), `age` (average age of houses), `lstat` (percent of households with low socioeconomic status). More info is available from `?Boston`.

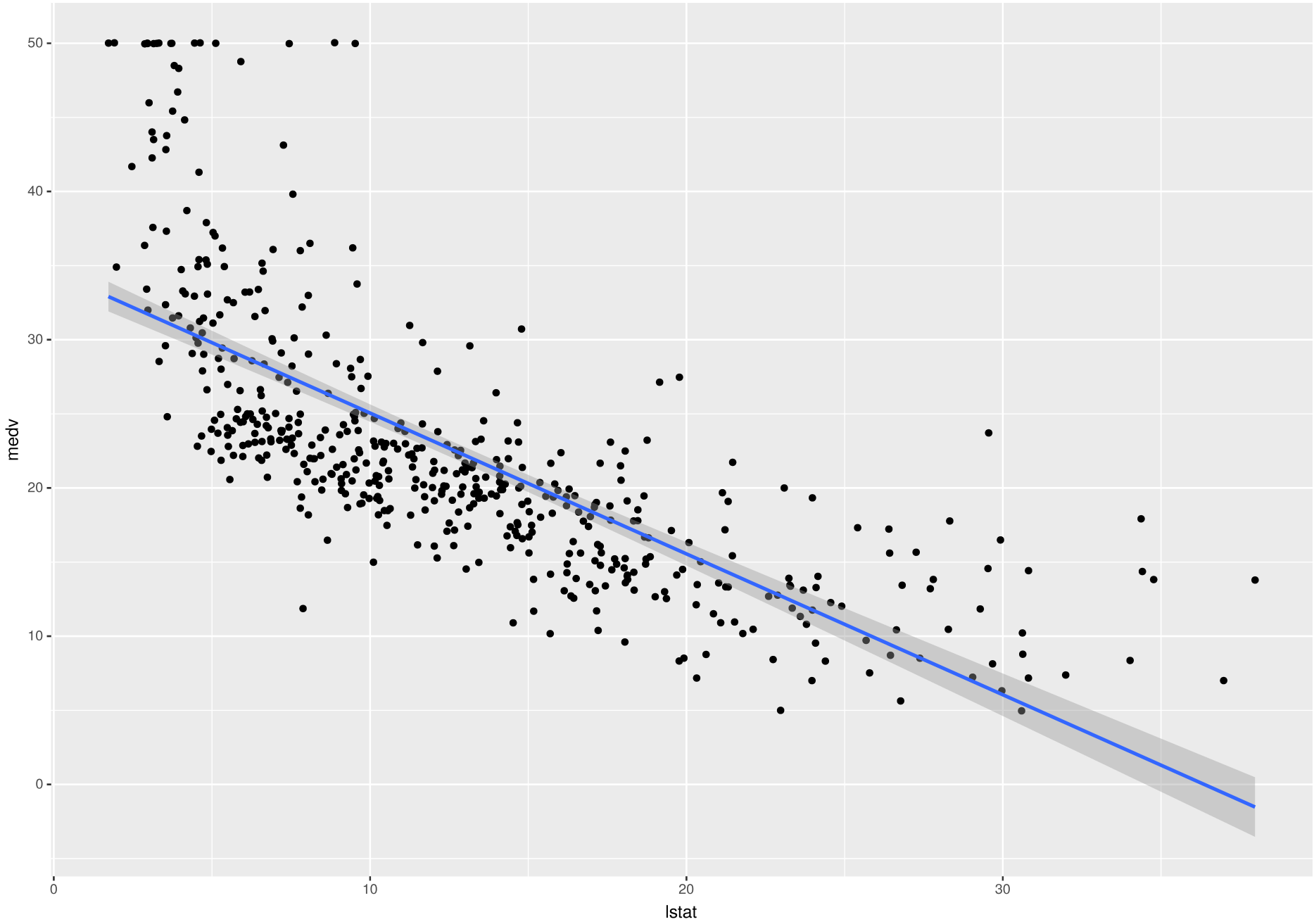
```
> attach(Boston)
> lm1.Boston=lm(medv~lstat)
> summary(lm1.Boston)

Call:
```

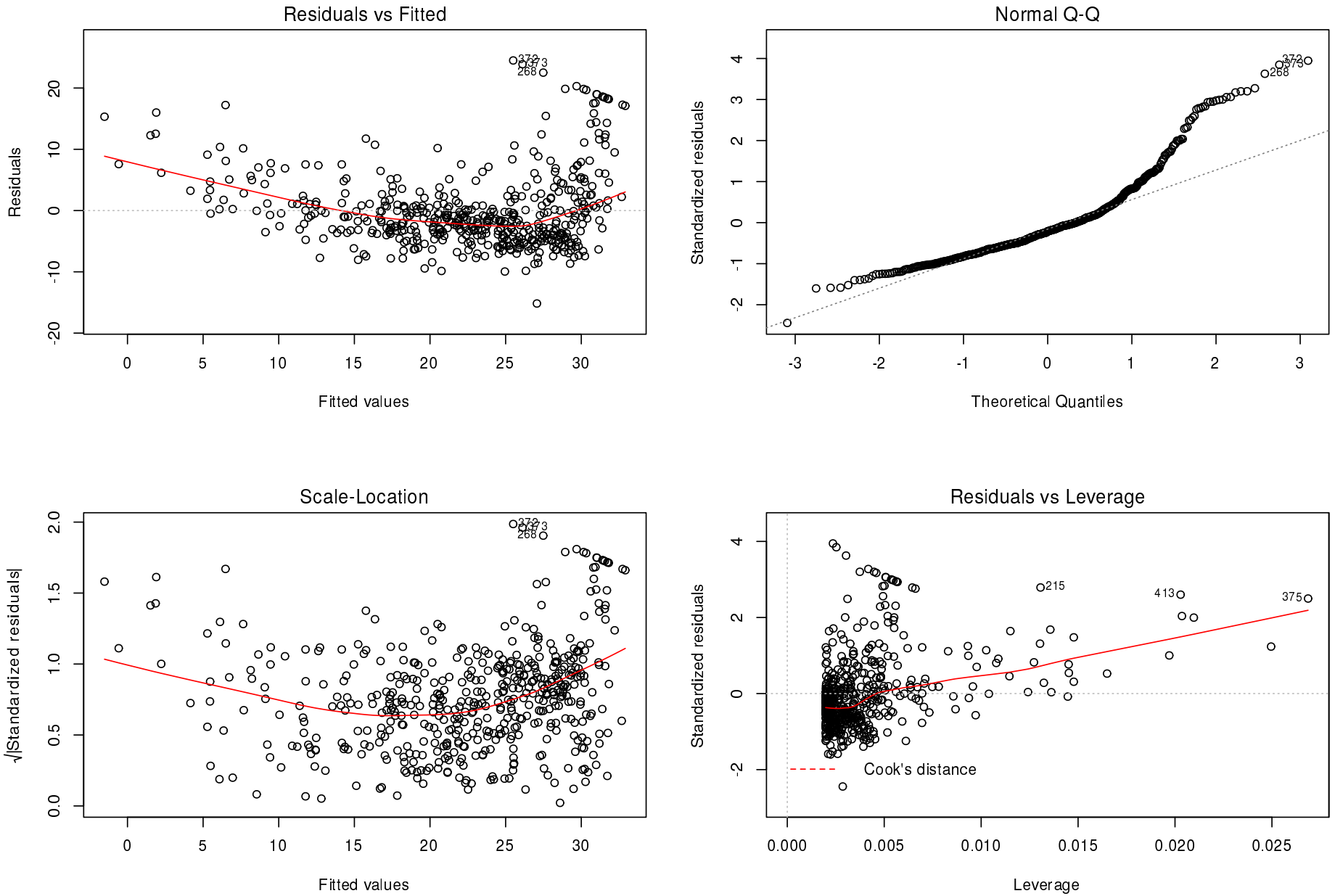
```
lm(formula = medv ~ lstat)
Residuals:
    Min      1Q  Median      3Q     Max
-15.168  -3.990  -1.318   2.034  24.500
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.55384    0.56263   61.41   <2e-16 ***
lstat       -0.95005    0.03873  -24.53   <2e-16 ***
Signif. codes:  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(lm1.Boston)  # additional info/components in output
 [1] "coefficients"  "residuals"     "effects"    "rank"  "fitted.values" "assign"
 [7] "qr"            "df.residual"   "xlevels"    "call"  "terms"         "model"
> lm1.Boston$rank   # to call for components in output
  # rank of X-matrix in matrix representation of regression model
[1] 2

> confint(lm1.Boston)  # Confidence intervals for coefficients
                2.5 %     97.5 %
(Intercept) 33.448457 35.6592247
lstat       -1.026148 -0.8739505
```

```
> predict(lm1.Boston, 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(lm1.Boston, 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(lm1.Boston)
> library(ggplot2) # re-do the plot using ggplot2
> ggplot(Boston, aes(lstat, medv))+geom_jitter()+geom_smooth(method=lm)

> par(mfrow=c(2,2)) # put 4 plots in one panel of 2x2 format
> plot(lm1.Boston)  # 4 plots for model diagnostic checking
```

Residual plots are not patternless! The Q-Q plot indicates that residuals are not normal-like.


medv
lstat
10
20
30
40
50
10
20
30


50
40
30
20
10
0
medv
0
10
20
30
lstat


Residuals vs Fitted
Residuals
Fitted values
372
373
268
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Residuals vs Leverage
215
413
375
Cook's distance
Leverage

`plot(lm1.Boston)` produces 4 diagnostic plots for the fitted model.

Top-left: Residuals vs Fitted — a plot of $\widehat{\varepsilon}_i$ against $\widehat{y}_i$. If the model is adequate, this plot should be patternless, as $\widehat{\varepsilon}_i$ should behave like random noise. The plot is helpful in detecting outliers, i.e. those y_i far away from $\widehat{y}_i$.

Top-right: Normal Q-Q — a plot of the quantiles of standardized residuals against the $N(0,1)$ quantiles. If residuals are normally distributed, the points should be on the straight line. It is particularly effective in highlighting *heavy tails: the points near the left-end are below the straight line, and the points close to the right-end are above the straight line*.

Bottom-left: Scale-Location — a plot of $\sqrt{|\widetilde{\varepsilon}_i|}$ against $\widehat{y}_i$, where $\widetilde{\varepsilon}_i$ denotes the standardized residual. This plot should be patternless too if the fitting is adequate. It is powerful in detecting inhomogeneous

variances among different observations. Note that for $Z \sim N(0, \sigma^2)$, $|Z|$ is heavily skewed to the left, and $\sqrt{|Z|}$ is much less skewed.

Bottom-right: Residuals vs Leverage – a plot of $\widetilde{\varepsilon}_i$ against leverage h_{ii}, where h_{ii} is the (i,i)-th element of the hat matrix $\mathbf{P}_x$ which defines the fitted value $\widehat{\mathbf{y}} = \mathbf{P}_x\mathbf{y}$, $\mathbf{y} = (y_1, \cdots, y_n)'$.

A **leverage point** is an observation which has a great influence on the analysis. The amount of the leverage of the i-th observation is reflected by h_{ii}. It can be shown that the total leverage is

$$\sum_{i=1}^{n} h_{ii} = \text{trace}(\mathbf{P}_x) = p + 1.$$

Therefore the average leverage for each observations is $\frac{p+1}{n}$.

A rule of thumb: if $h_{ii} > \frac{2(p+1)}{n}$, the i-th observation is a leverage point.

Note that as $\mathbf{P}_x$ only depends on $\mathbf{x}_1, \cdots, \mathbf{x}_n$, so is the leverage.

A leverage point is called a **good leverage point** if the corresponding y is close to $\widehat{y}$. It is called a **bad leverage point** if the corresponding y is an outlier.

For the food data set, $\frac{2(p+1)}{n} = \frac{4}{506} = 0.0079$. The figure shows the 215th, 413th and 375th observations are bad leverage points. We may consider to remove them from the analysis, since they have great influence on the fitted model.

We may try

```
lm(medv ~ lstat + age) # Use 2 regressors: lstat and age
lm(medv ~ lstat*age) # Use 3 regressors: lstat, age, and their product
lm(medv ~ lstat + I(lstat^2)) # I(lstat^2) represent lstat^2
lm(medv ~ poly(lstat, 5)) # using polynomial function of order 5
lm(medv ~ ., data=Boston)   # Using all variables in Boston as regressors
lm(medv ~.-age, data=Boston) # Using all but age
```

Let us try a few.

```
> lm2.Boston=lm(medv~., Boston)
> summary(lm2.Boston)
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
```

Since age has the largest P-value (i.e. 0.958), we remove it from the model

```
> lm3.Boston=lm(medv~.-age, Boston)
> summary(lm3.Boston)
Call:
lm(formula = medv ~ . - age, data = Boston)
Residuals:
     Min       1Q   Median       3Q      Max
-15.6054  -2.7313  -0.5188   1.7601  26.2243
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
crim         -0.108006   0.032832  -3.290 0.001075 **
```

```
zn            0.046334   0.013613   3.404 0.000719 ***
indus         0.020562   0.061433   0.335 0.737989
chas          2.689026   0.859598   3.128 0.001863 **
nox         -17.713540   3.679308  -4.814 1.97e-06 ***
rm            3.814394   0.408480   9.338  < 2e-16 ***
dis          -1.478612   0.190611  -7.757 5.03e-14 ***
rad           0.305786   0.066089   4.627 4.75e-06 ***
tax          -0.012329   0.003755  -3.283 0.001099 **
ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
black         0.009321   0.002678   3.481 0.000544 ***
lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4.74 on 493 degrees of freedom
Multiple R-squared:  0.7406,Adjusted R-squared:  0.7343
F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

As `indus` is not significant, we remove it now

```
> lm4.Boston = update(lm3.Boston, ~.-indus)
> summary(lm4.Boston)
Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
    tax + ptratio + black + lstat, data = Boston)
Residuals:
```

```
     Min       1Q   Median       3Q      Max
-15.5984  -2.7386  -0.5046   1.7273  26.2373
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
crim         -0.108413   0.032779  -3.307 0.001010 **
zn            0.045845   0.013523   3.390 0.000754 ***
chas          2.718716   0.854240   3.183 0.001551 **
nox         -17.376023   3.535243  -4.915 1.21e-06 ***
rm            3.801579   0.406316   9.356  < 2e-16 ***
dis          -1.492711   0.185731  -8.037 6.84e-15 ***
rad           0.299608   0.063402   4.726 3.00e-06 ***
tax          -0.011778   0.003372  -3.493 0.000521 ***
ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
black         0.009291   0.002674   3.475 0.000557 ***
lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4.736 on 494 degrees of freedom
Multiple R-squared:  0.7406,Adjusted R-squared:  0.7348
F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

Although we have removed two variables `age`, `indus`, the regression correlation coefficient is unchanged, as $R^2 = 0.7406$ always. But the adjusted correlation coefficient increases slightly.

We can also fit the data using the stepwise procedure to select variables. R function `step` implements this selection method using the AIC criterion. To use `step`, we need to specify a maximum model, which is `ls2.Boston` including all the variables, and a minimum model which may include intercept term only:

```
> lm0.Boston=lm(medv~1)
> lm0.Boston
Call:
lm(formula = medv ~ 1)

Coefficients:
(Intercept)
      22.53
```

The selected model will be between `ls0.Boston` and `ls2.Boston`.

Now we are ready to call for stepwise selection:

```
> step.Boston=step(lm0.Boston, scope=list(upper=lm2.Boston))
> summary(step.Boston)
```

```
Call:
lm(formula = medv ~ lstat + rm + ptratio + dis + nox + chas +
    black + zn + crim + rad + tax)

Residuals:
     Min       1Q   Median       3Q      Max
-15.5984  -2.7386  -0.5046   1.7273  26.2373

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
rm            3.801579   0.406316   9.356  < 2e-16 ***
ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
dis          -1.492711   0.185731  -8.037 6.84e-15 ***
nox         -17.376023   3.535243  -4.915 1.21e-06 ***
chas          2.718716   0.854240   3.183 0.001551 **
black         0.009291   0.002674   3.475 0.000557 ***
zn            0.045845   0.013523   3.390 0.000754 ***
crim         -0.108413   0.032779  -3.307 0.001010 **
rad           0.299608   0.063402   4.726 3.00e-06 ***
tax          -0.011778   0.003372  -3.493 0.000521 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 4.736 on 494 degrees of freedom
Multiple R-squared:  0.7406,Adjusted R-squared:  0.7348
F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

The final selected model is actually `ls4.Boston`.

Note. R prints out on the screen the whole process of how this model was derived from the initial `ls0.Boston` by adding and deleting variables step by step.

Regression with qualitative Predictors

We use the data `Carseats` in ISLR as an illustration.

```
> summary(Carseats)
     Sales          CompPrice       Income        Advertising       Population
 Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000   Min.   : 10.0
 1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000   1st Qu.:139.0
 Median : 7.490   Median :125   Median : 69.00   Median : 5.000   Median :272.0
 Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635   Mean   :264.8
 3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000   3rd Qu.:398.5
 Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000   Max.   :509.0
    Price       ShelveLoc        Age          Education    Urban       US
Min.   : 24.0  Bad   : 96   Min.   :25.00   Min.   :10.0   No :118   No :142
1st Qu.:100.0  Good  : 85   1st Qu.:39.75   1st Qu.:12.0   Yes:282   Yes:258
Median :117.0  Medium:219   Median :54.50   Median :14.0
Mean   :115.8               Mean   :53.32   Mean   :13.9
3rd Qu.:131.0               3rd Qu.:66.00   3rd Qu.:16.0
Max.   :191.0               Max.   :80.00   Max.   :18.0
```

The 3 qualitative variables: `Urban` and `US` are binary, and `ShelveLoc` takes 3 values.

We add interaction terms `Income:Advertising` and `Price:Age` in the model.

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

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
```

For binary variables `Urban` and `US`, R creates dummy variables `UrbanYes`, `USYes`.

For `ShelveLoc` with 3 values, two dummies `ShelveLocGood` and `ShelveLocMedium` are created. To check their definition,

```
> attach(Carseats}
> contrasts(ShelveLoc)
       Good Medium
Bad       0      0
Good      1      0
Medium    0      1
```

i.e. `ShelveLocGood` takes value 1 if `ShelveLoc=Good`, and 0 otherwise, and `ShelveLocMedium` takes value 1 if `ShelveLoc=Medium`, and 0 otherwise.

Both `ShelveLocGood` and `ShelveLocMedium` are significant in the fitted model with coefficient 4.849 and 1.953 respectively, indicating that a medium shelving location leads to higher sales than a bad shelving location but lower sales than a good shelving location.

Regression trees

Linear regression specifies an explicit model, which is *linear* in coefficients, for the regression function. It works well when the model is about correct.

When the true function is highly nonlinear, a tree model may provide a valid alternative.

A regression free:

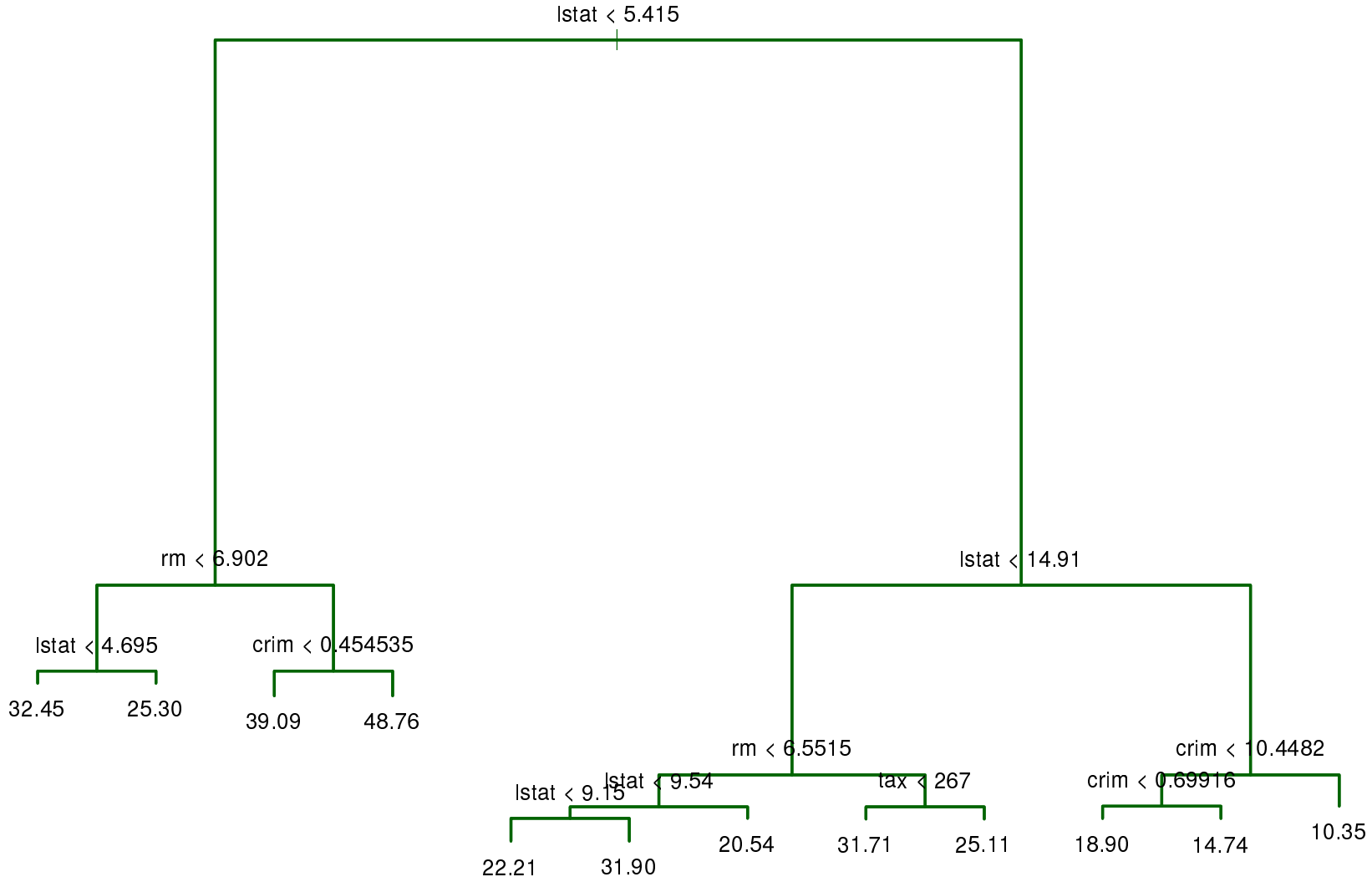
$$\widehat{Y} = \sum_{i=1}^{M} c_i I(\mathbf{X} \in R_i),$$

where $R_1, \cdots, R_M$ form a partition of the feature space (i.e. the $\mathbf{X}$-space) and

$$c_i = \sum_{j=1}^{M} y_j I(\mathbf{x}_j \in R_i) \Big/ \sum_{j=1}^{M} I(\mathbf{x}_j \in R_i).$$

Similar to growing a decision tree, we use recursive binary splitting to grow a regression tree. But each time we search for the splitting which maximises the reduction of RSS= $\sum_i (y_i - \widehat{y}_i)^2$. We stop when, for example, each terminal node has fewer than k_0 observations, where k_0 is a prespecified integer.

```
> library(MASS); library(tree)
> train=sample(1:nrow(Boston), nrow(Boston)/2)
> tree.Boston=tree(medv~., data=Boston, subset=train)
> summary(tree.Boston)

Regression tree:
tree(formula = medv ~ ., data = Boston, subset = train)
Variables actually used in tree construction:
[1] "lstat" "rm"    "crim"  "tax"
Number of terminal nodes:  12
Residual mean deviance:  12.79 = 3082 / 241 # Mean RSS for regression
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-11.0000  -1.8060  -0.3394   0.0000   1.8860  18.1000
> plot(tree.Boston, col="blue")
> text(tree.Boston, pretty=0)
```


lstat < 5.415
rm < 6.902
lstat < 14.91
lstat < 4.695
crim < 0.454535
rm < 6.5515
crim < 10.4482
lstat < 9.54
tax < 267
crim < 0.69916
lstat < 9.15
32.45
25.30
39.09
48.76
20.54
31.71
25.11
18.90
14.74
10.35
22.21
31.90

The fitting entails the mean RSS 12.79 for the training data. Let us check how it performs on testing data

```
> medv.test=Boston[-train, "medv"]
> medv.predict=predict(tree.Boston, newdata=Boston[-train,])
> mean((medv.predict-medv.test)^2)
[1] 30.14707
```

The mean squares of predictive errors is 30.15 for the testing sample, which is greater than 12.79 for the training sample.

Bagging: a bootstrap aggregation method

A fitted tree suffers from *high variance*, i.e. the tree depends on training data sensitively.

Basic idea to reduce variance: average a set of observations.

For example, the variance of a sample mean, from a sample of size n, is reduced from σ^2 to σ^2/n.

If we had many fitted trees, the "mean" of those trees would have a smaller variance. But in practice we only have one sample.

Create B sets of training data by bootstrapping from the original data set. For each bootstrap sample, fitting a tree. Those trees are grown deep, and are not pruned. Hence each individual tree has high variance but low bias.

'Averaging' those B trees reduces the variance.

For decision trees, 'averaging' means taking the majority votes of the B trees.

Bootstrap in R: Let `X` be a vector containing n observation. Then a bootstrap sample is obtained as follows

```
> Xstar = sample(X, n, replace=F)
```

Note. Bagging improves prediction accuracy at the expense of interpretability, as the final result is an average of many different trees.

Importance: The importance of each predictor can be measured by the average reduction of RSS (for regression trees), or average information gain (for decision trees) over different trees – The larger the better!

Random forests: an improvement of Bagging by decorrelating the trees.

Similar to Bagging, we build a tree for each bootstrap sample. Differently from Bagging, at each step we split the feature space by searching the split from m, instead of all p, predictors, where $m \leq p$. Furthermore, we use a different and randomly selected subsets of m predictors for each split.

Typical choice: $m = \sqrt{p}$.

When $m = p$, it reduces to Bagging.

Why is better: the trees in the forest are less correlated than those in the bagging.

For example, suppose there is a dominate predictor which is likely to enter all or most trees if $m = p$. This would make the fitted tree highly correlated.

```
> install.packages("randomForest")
> library(randomForest)
> bag.Boston=randomForest(medv~., data=Boston, subset=train,
    mtry=13, importance=T)  # mtry=13 sets m=p
> bag.Boston
Call:
 randomForest(formula = medv ~ ., data = Boston, mtry = 13,
             importance = T, subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 13
          Mean of squared residuals: 14.48475
                    % Var explained: 82.31
```

The option `mtry=13` demands to search over all $p = 13$ variables in each split, and therefore, it is a Bagging fitting. The mean RSS is 14.48, and $R^2 = 82.31\%$.

```
> medv.predict=predict(bag.Boston, newdata=Boston[-train,])
> mean((medv.predict-medv.test)^2)
[1] 12.83074
```

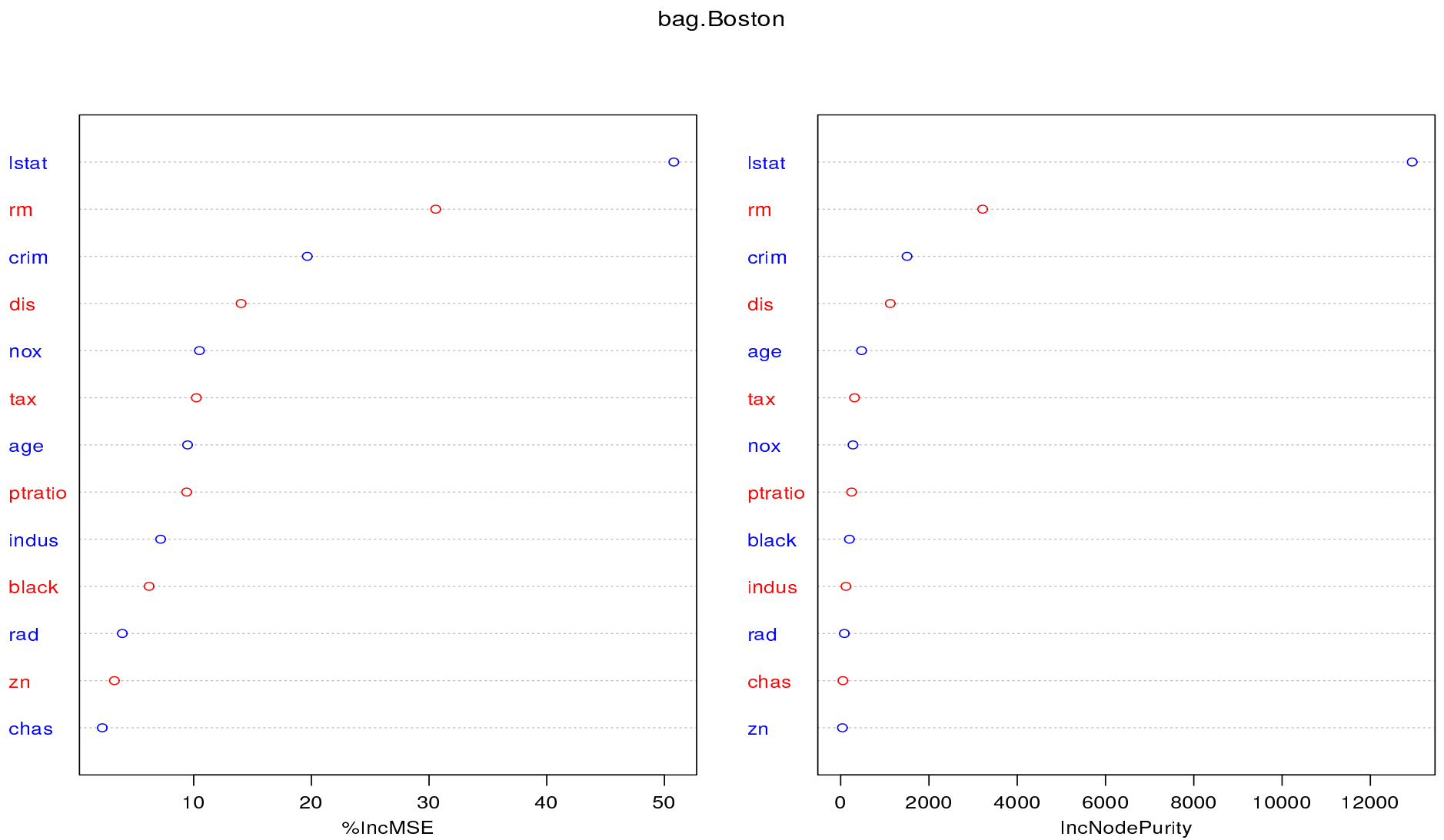
The mean squares of predictive errors for the test sample from this Bagging fitting is 12.83.

```
> importance(bag.Boston)
          %IncMSE IncNodePurity
crim    19.651982    1503.30977
zn       3.258542      39.40467
indus    7.189510     119.63707
chas     2.234998      50.37624
nox     10.487697     278.67558
rm      30.564156    3216.50360
age      9.478481     473.88776
dis     14.028146    1122.75188
rad      3.936966      82.59776
tax     10.225864     313.85307
ptratio  9.402331     248.13892
black    6.207921     197.53684
lstat   50.795822   12946.76696
```

Two measures of importance are reported: The former is based on the mean decrease of accuracy in prediction on the 'out of bag' sample when the predictor is excluded from the model. The latter is measure of the total decrease in node impurity that results from splits over the variable, averaged over all trees.

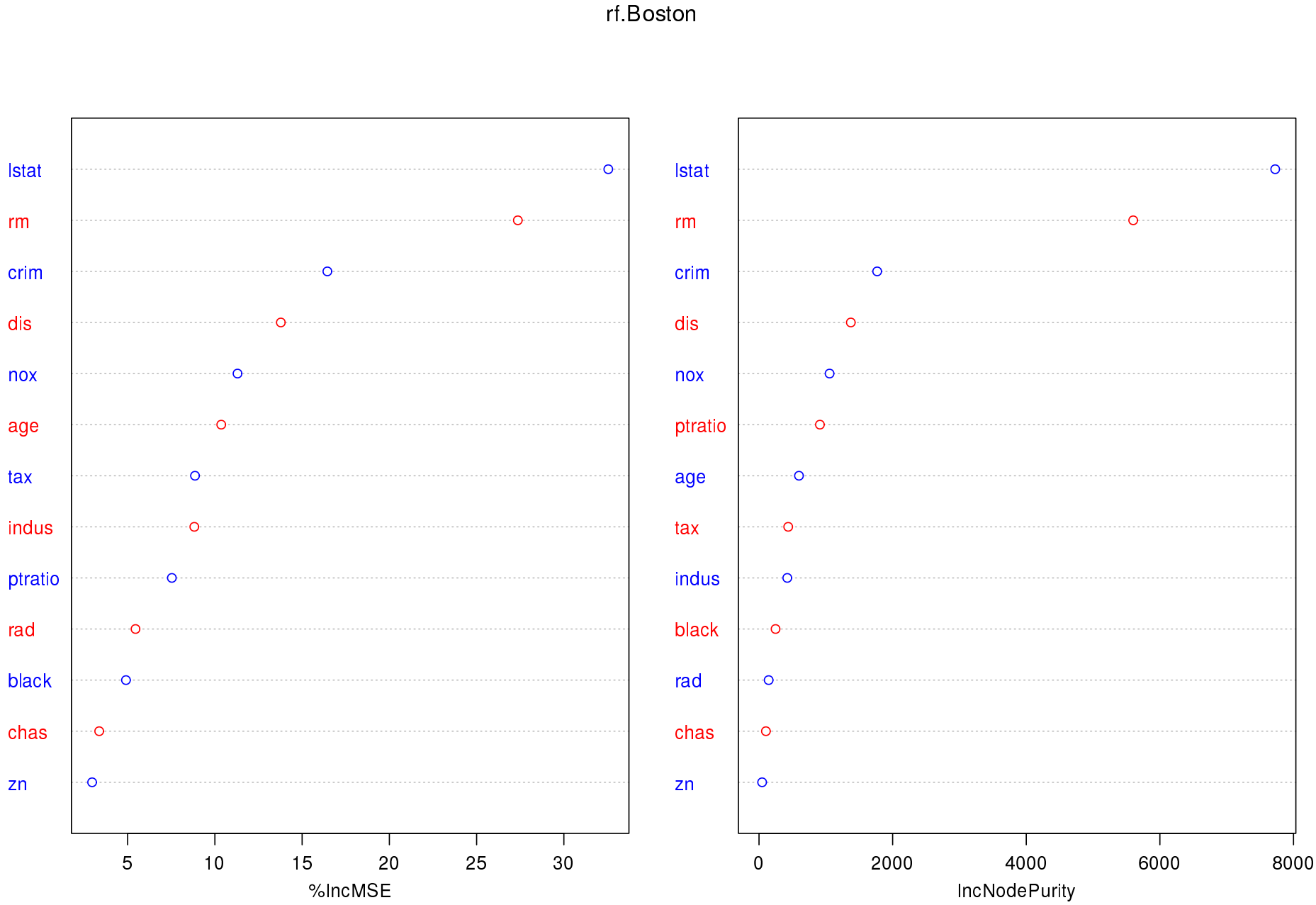
For this example, `lstat` (the wealth level of the community) is the most important predictor, followed by rm (house size).

```
> varImpPlot(bag.Boston, col=c("blue","red")) # Importance measure plot
```



```
> rf.Boston=randomForest(medv~., data=Boston, subset=train,
                    mtry=6, importance=T)  # 'mtry=6' sets m=6<p
> rf.Boston
Call:
 randomForest(formula = medv ~ ., data = Boston, mtry = 6,
                importance = T, subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 6

          Mean of squared residuals: 13.69494
                    % Var explained: 83.27
> medv.predict=predict(rf.Boston, newdata=Boston[-train,])
> mean((medv.predict-medv.test)^2)
[1] 12.18501
> varImpPlot(rf.Boston, col=c("blue","red"))
```

Mean squares of predictive errors for the testing sample from this random forest model is 12.18501, smaller than that from the Bagging model.


rf.Boston
lstat
rm
crim
dis
nox
age
tax
indus
ptratio
rad
black
chas
zn
5
10
15
20
25
30
%IncMSE
lstat
rm
crim
dis
nox
ptratio
age
tax
indus
black
rad
chas
zn
0
2000
4000
6000
8000
IncNodePurity

Boosting

Like Bagging, boosting can be applied to many learning methods for regression and classification. We use regression trees as an illustration.

Boosting algorithm for regression trees

1. Set $\widehat{f}(x) = 0$ and $r_i = y_i$ for all i.
2. For $b = 1, \cdots, B$, repeat:
   (a) Fit a tree $\widehat{f}^b$ with d split ($d+1$ terminal nodes) to the training data $\{x_i, r_i\}$.
   (b) Update $\widehat{f} = \widehat{f} + \lambda \widehat{f}^b$.
   (c) Update residuals $r_i = r_i - \lambda \widehat{f}^b(x_i)$.
3. Output the boosted model $\widehat{f}(x) = \sum_{b=1}^{B} \widehat{f}^b(x)$.

3 tuning parameters:

B is a large integer, too large B leads to overfitting. It can be selected by cross-validation.

d is a small integer, typically taking values 1 or 2.

$\lambda \in (0, 1)$, is typically small such as 0.01 or 0.001. Smaller λ requires larger B.

It is known that *fitting the data hard* may lead to overfitting. Boosting is in the spirit of *learning slowly*: Given the current model, we fit a small tree (as d is small) to the residuals from the model. By fitting small trees to the residuals, we slowly improve $\widehat{f}$ in areas where it does not perform well. The shrinkage parameter λ slows the process down even further, allowing more and different shaped trees to attack the residuals.

Note that in boosting, unlike in Bagging, the construction of each tree depends strongly on the trees that have already been grown.

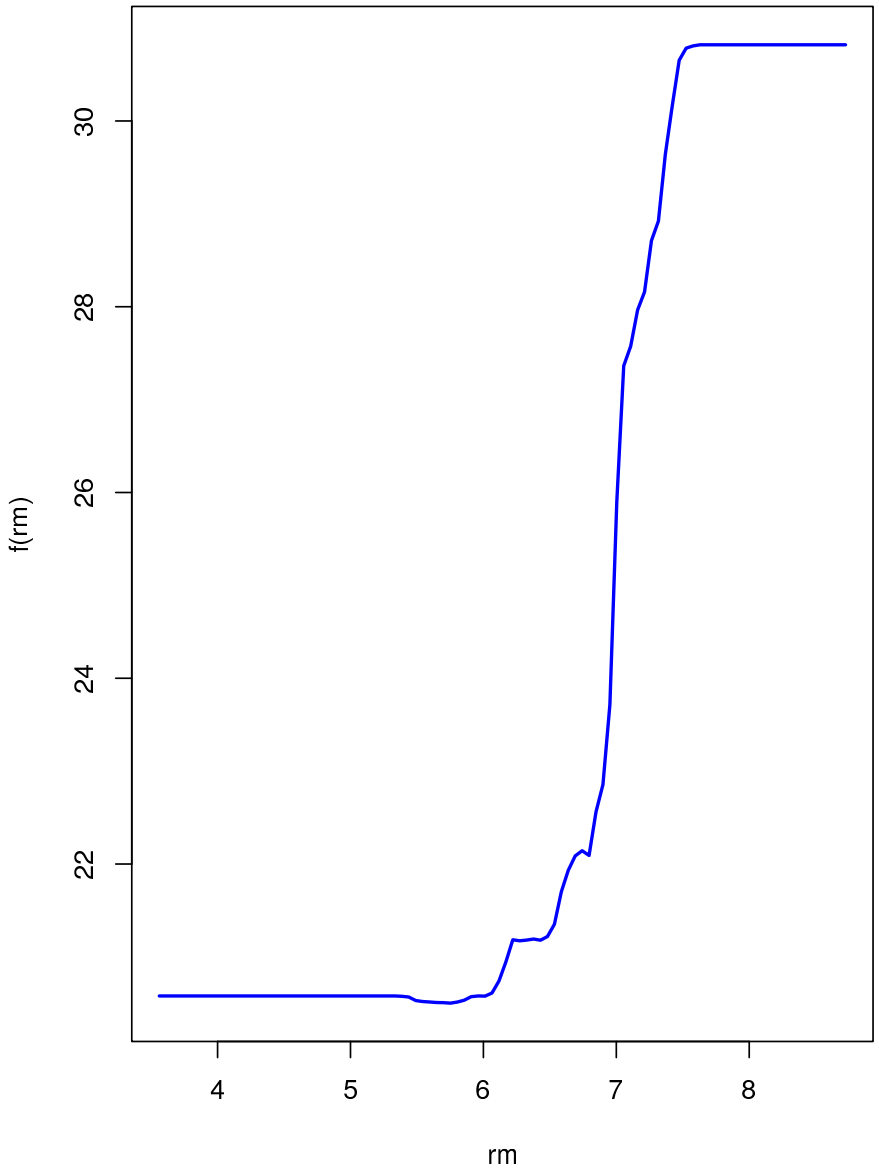
```
> install.packages("gbm") # gbm: Generalized boosted model
> library(gbm)
> boost.Boston=gbm(medv~., data=Boston[train,], n.trees=5000,
      interaction.depth = 4) # B=n.trees, default value lambda=0.001
     # d=interaction.depth
> summary(boost.Boston)
            var    rel.inf
lstat     lstat 51.6849642
rm           rm 23.1720615
crim       crim  7.9381089
```

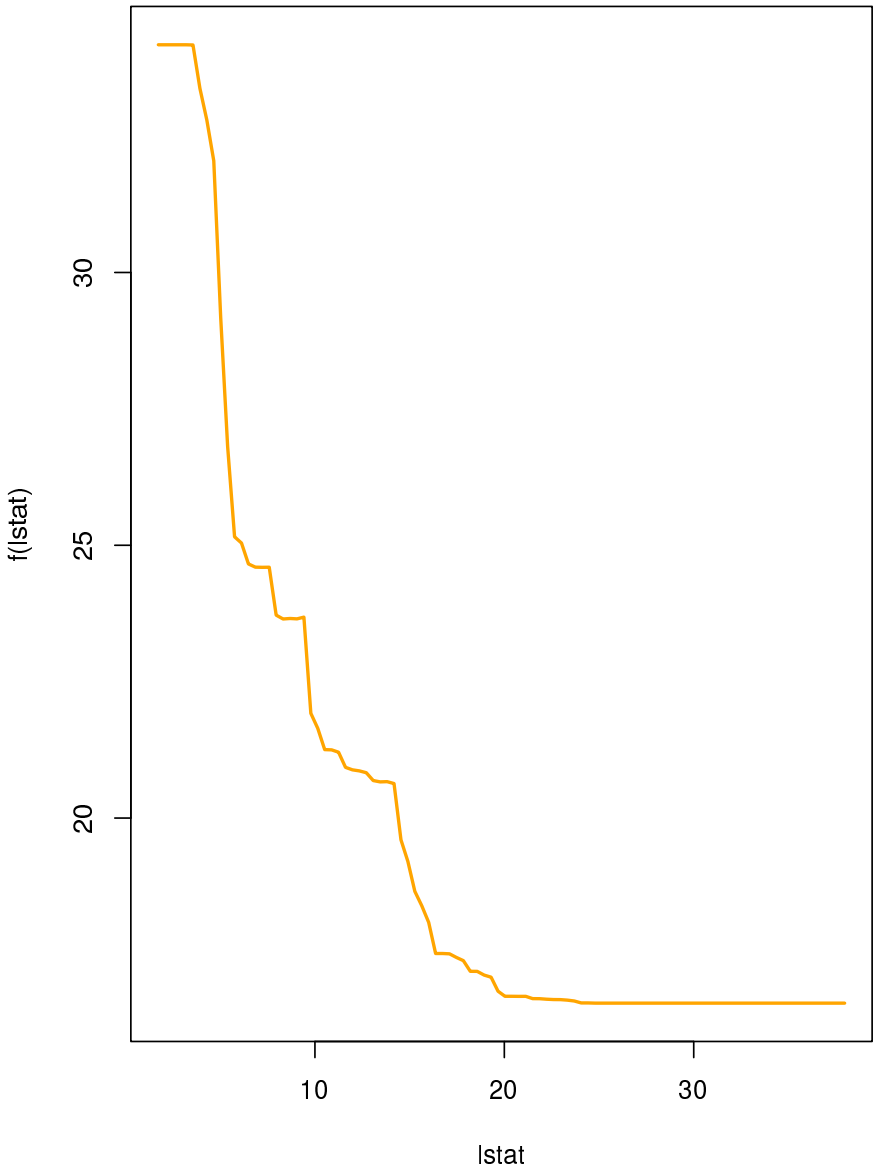
```
dis          dis  6.4563409
nox          nox  2.4229547
age          age  2.2533567
ptratio ptratio  2.2168609
black      black  1.0393828
tax          tax  0.8387594
indus      indus  0.6450746
chas        chas  0.6439905
rad          rad  0.5473836
zn            zn  0.1407613
```

The summary provides the relative inference statistics: `lstat` and `rm` are by far the most important variables.

We can also produce partial dependence plots for these two variables. These plots illustrate the marginal effect of the selected variables on the response after integrating out the other variables. In this case, as we might expect, median house prices are increasing with `rm` and decreasing with `lstat`.

```
> par(mfrow=c(1,2))
> plot(boost.Boston, i="rm", lwd=2, col="blue")
> plot(boost.Boston, i="lstat", lwd=2, col="orange")
```


f(rm)
22
24
26
28
30
4
5
6
7
8
rm


f(lstat)
20
25
30
10
20
30
lstat

To compute the mean squares of predictive errors for the test sample:

```
> medv.predict=predict(boost.Boston, newdata=Boston[-train,],
          n.trees=5000)
> mean((medv.predict-medv.test)^2)
[1] 13.896
```

One can, for example, set $\lambda = 0.2$:

```
> boost.Boston=gbm(medv~., data=Boston[train,], n.trees=5000,
                interaction.depth =4, shrinkage =0.2)
```

From global fitting to local fitting — an illustration by example

Consider linear regression model

$$Y = X_1\beta_1 + \cdots + X_d\beta_d + \varepsilon, \tag{1}$$

where $\varepsilon \sim (0, \sigma^2)$.

Put $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d)^\tau$

With observations $\{(y_i, \mathbf{x}_i),\ 1 \le i \le n\}$, where $\mathbf{x}_i = (x_{i1}, \cdots, x_{id})^\tau$, the LSE minimizes

$$\sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^\tau \boldsymbol{\beta}\right)^2, \tag{2}$$

resulting to

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\tau\mathbf{X})^{-1}\mathbf{X}^\tau\mathbf{y},$$

where $\mathbf{y} = (y_1, \cdots, y_n)^\tau$, and $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^\tau$ is an $n \times d$ matrix.

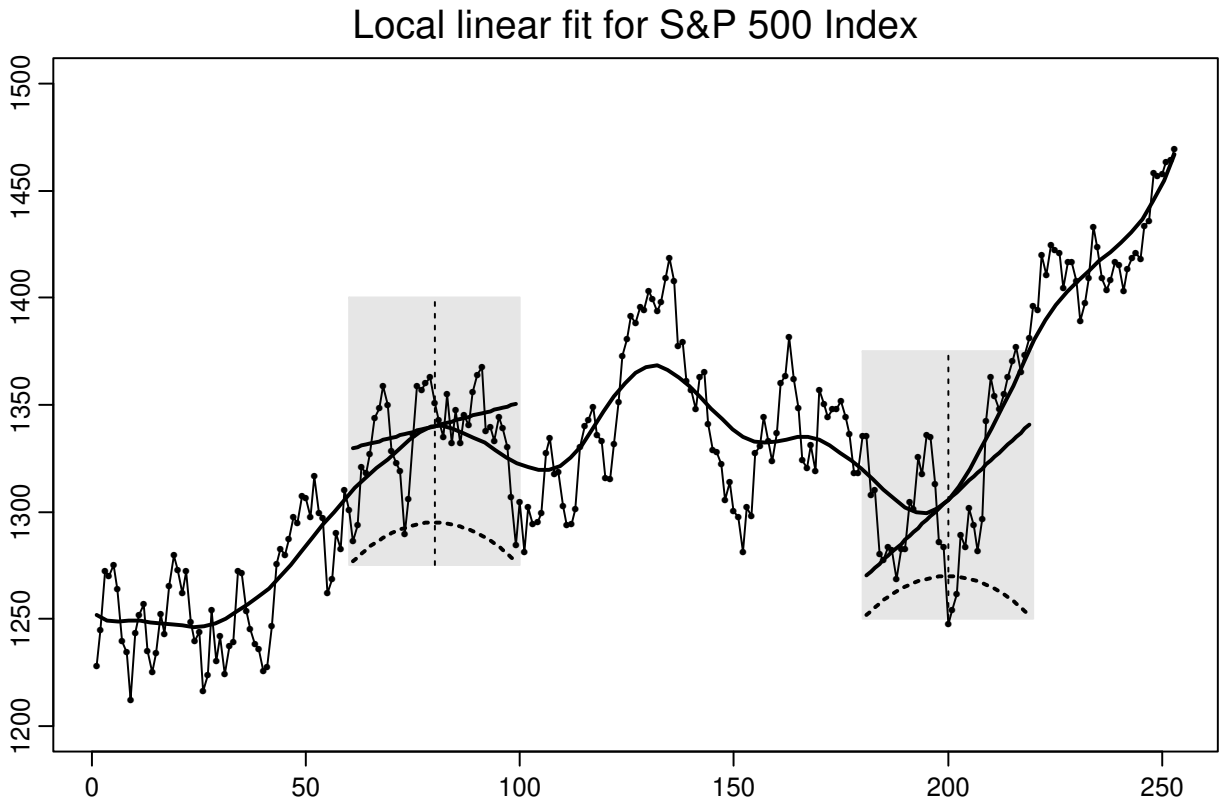
The fitted model is

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}.$$

This is a **global** fitting, since the model is assumed to be true everywhere in the sample space and the estimator $\widehat{\boldsymbol{\beta}}$ is obtained using all the available data.

Such a global fitting is efficient **if** the assumed form of the regression function (1) is correct.

In general (1) may be incorrect globally. But it may provide a reasonable approximation at any small area in the sample space. We fit for each given small area a different linear model — This is the basic idea of **local** fitting.



Local linear fit for the S&P 500 Index from January 4, 1999 to December 31, 1999, using the Epanechnikov kernel and bandwidth $h = 20$. *The dashed parabola in each window indicates the weight that each local data point receives.*

Local fitting – Non-parametric models: little assumption on the underlying model, but estimation is less efficient.

Serious drawback for multivariate model: curse of dimensionality

Semi-parametric models: mitigate the curse of dimensionality

- partial linear models
- index models
- additive models (R package `gam`)
- varying-coefficient linear models

Does a tree model provide a local fitting?

Mincer Equation: How is one's earning related to human capital?

$$\log(Y) = \beta_0 + \beta_1 X + \beta_2 U + \beta_3 U^2 + \varepsilon,$$

Y — earning

X — education capital: No. of years in school/university

U — experience capital: No. of years in employment

Rate of return to education: β_1

This is a simple linear regression model.

Drawbacks: no interaction between X and U

Extended Mincer Equation:

$$\begin{aligned}
\log(Y) &= \beta_0 + \beta_{11}X + \beta_{12}UX + \beta_{13}U^2X + \beta_2U + \beta_2U^2 + \varepsilon \\
&= (\beta_0 + \beta_2U + \beta_2U^2) + (\beta_{11} + \beta_{12}U + \beta_{13}U^2)X + \varepsilon \\
&= g_0(U) + g_1(U)X + \varepsilon.
\end{aligned}$$

If we see $g_0(\cdot)$ and $g_1(\cdot)$ as coefficient functions, this is varying-coefficient linear model.

In general, we do not restrict the coefficients as quadratic functions: local fitting.

Rate function of return to education: $g_1(\cdot)$

Example. (Wang and Yue 2008). Survey data on annual incomes and human capitals of Chinese citizen in 1991, 1995, 2000 and 2004.

Fitting a linear model:

$$\log(Y) = \beta_0 + \beta_1 X + \beta_2 U + \beta_3 U^2 + \eta_1 Z_1 + \eta_2 Z_2 + \varepsilon$$

where

Y: total annual income
X: no. of years in school/university
U: no. of years in employment
$Z_1 = 1$ – female, $Z_1 = 0$ – male
$Z_2 = 1$ – eastern China, $Z_2 = 0$ – central/western China

Hence

η_1: difference in log(income) between female and male
η_2: difference in log(income) between the Eastern and the rest of China

	1991	1995	2000	2004
$\widehat{\beta}_0$	6.823	7.647	7.410	7.747
$\widehat{\beta}_1$	0.028	0.045	0.086	0.105
$\widehat{\beta}_2$	0.054	0.026	0.034	0.024
$\widehat{\beta}_3$	-0.0001	-0.0002	-0.0004	-0.0002
$\widehat{\gamma}_1$	-0.095		-0.171	-0.272
$\widehat{\gamma}_2$	0.195	0.293	0.260	0.362

Note. The quality of the data for 1995 is poor: no gender, censored to the minimum 1003 yuans etc.

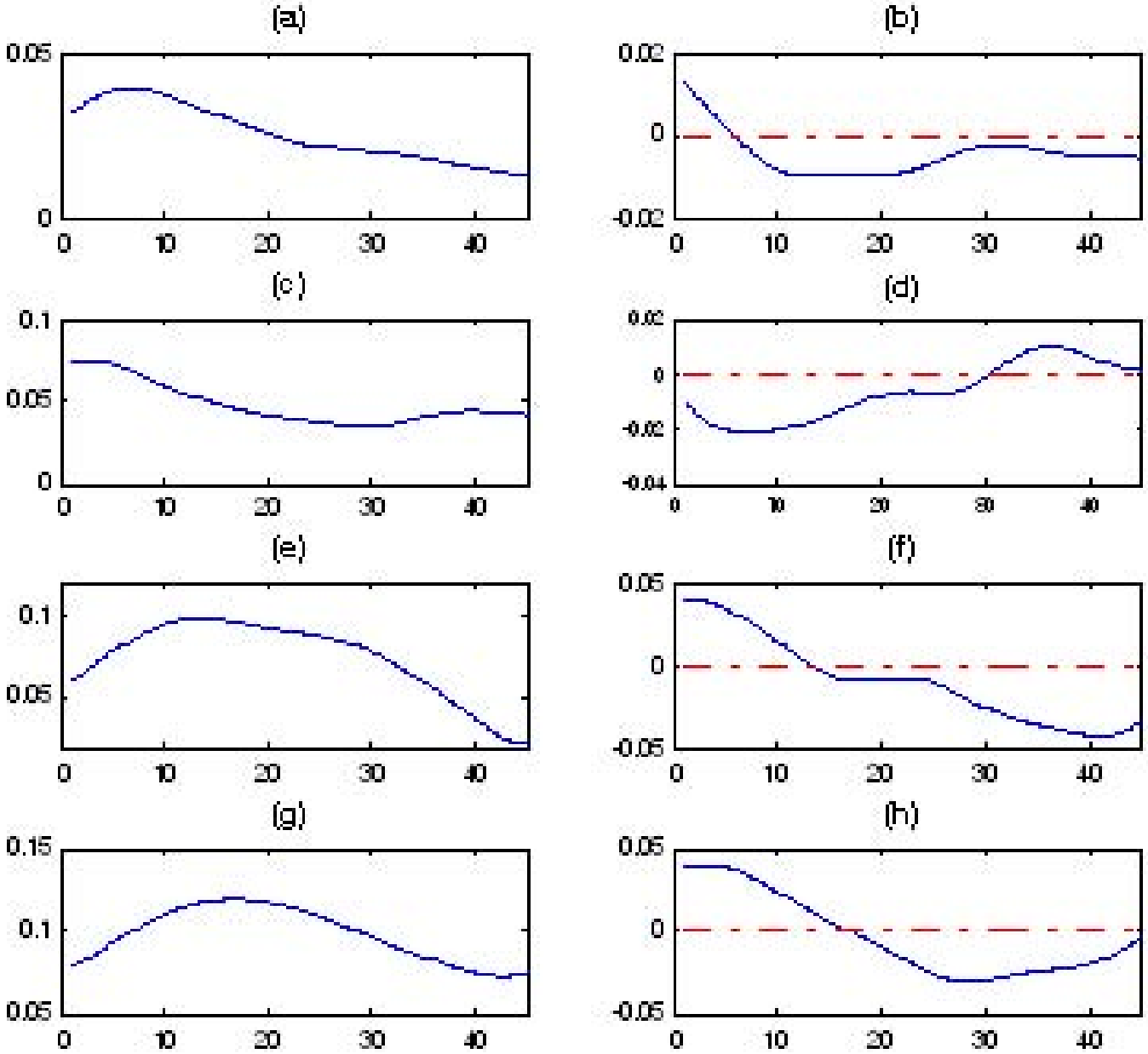
Fitting a varying-coefficient linear model:

$$\log(Y) = g_0(U) + g_1(U)X + \eta_1 Z_1 + \eta_2 Z_2 + \varepsilon$$

where

Y: total annual income
X: no. of years in school/university
U: no. of years in employment
$Z_1 = 1$ – female, $Z_1 = 0$ – male
$Z_2 = 1$ – eastern China, $Z_2 = 0$ – central/western China

The estimated values for η_1 and η_2 hardly changed.



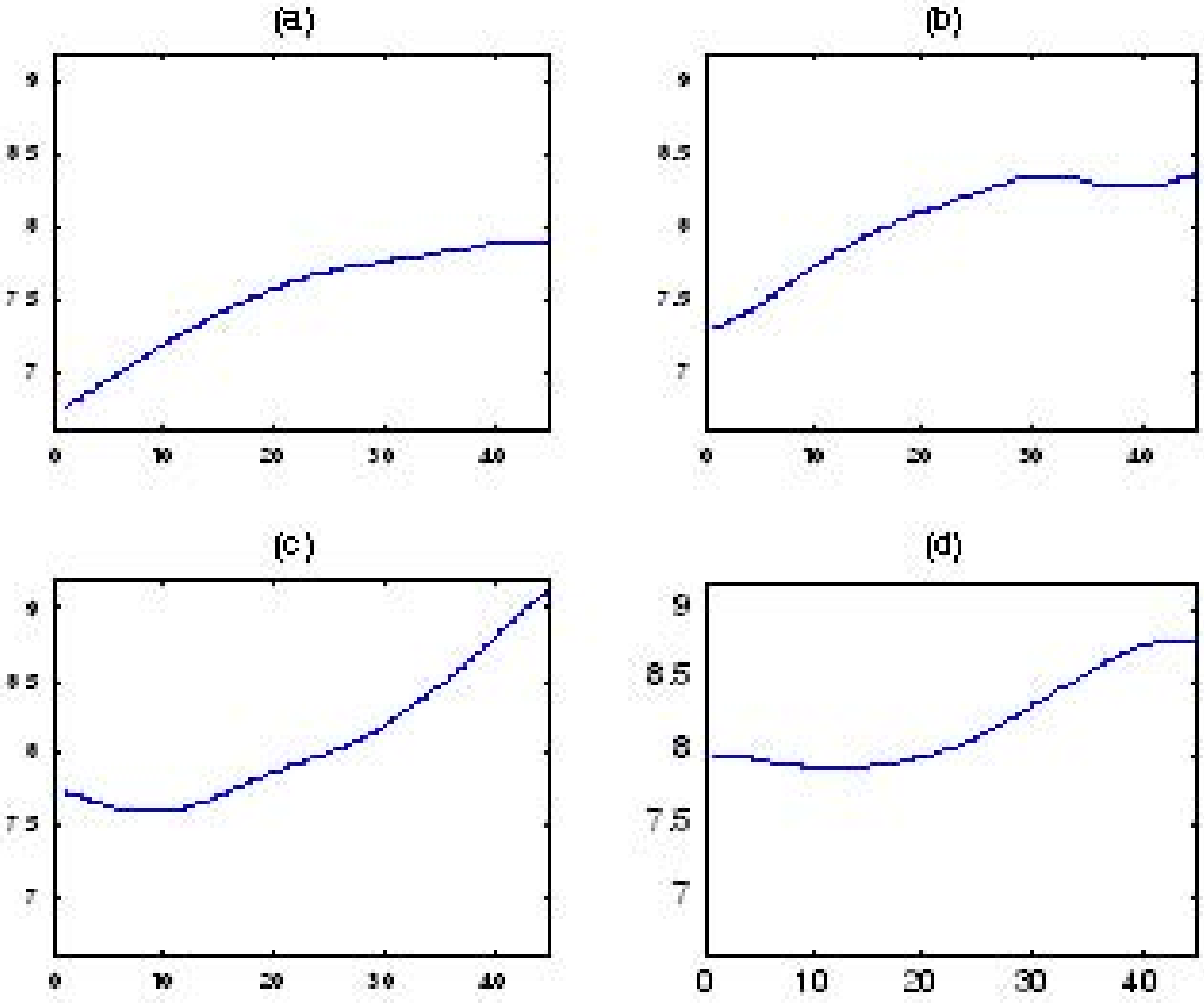
Estimated $g_1(U)$ for (a) 1991, (c) 1995, (e) 2000 & (g) 2004, and their derivatives.

In 1991, g_1 = max when $U = 5.8$

In 2000, g_1 = max when $U = 13.5$

In 2004, g_1 = max when $U = 16.9$

Those started to work around 1985 – 1987 always have the maximum returns out of education.



Estimated $g_0(U)$ for (a) 1991, (b) 1995, (c) 2000 & (d) 2004.

$g_0(U)$ increases initially, then gradually flat out after about 20-30 years when experience saturated.