Chapter 11

Multiple and Nonlinear Regression

11.1 Introduction

Aim of this chapter:

♠ To extend the techniques to multiple variables / factors.

♠ To check adequacy of a fitted model.

♠ Model building and prediction

Purpose of multiple regression:

— Study association between dependent and independent variables

— Screen irrelevant and select useful variables

— Prediction

Example 11.1 *Hong Kong Environmental Data Set .*

Interest: Study the association between **levels of pollutants** and number of daily total **hospital admissions** for circulatory and respiratory problems.

— Dependent variable (Y) = Daily number of hospital admissions

— Collected covariates = $\{$

level of pollutant Sulphur Dioxide X_1 (in $\mu g/m^3$),

level of pollutant Nitrogen Dioxide X_2 (in $\mu g/m^3$)

level of respirable suspended particles X_3 (in $\mu g/m^3$)

Ozone level X_4

Temperature X_5 (in ^oC)

Humidity (X_6, in percent)

time X_7 (season, confounding factor),

.... }

year	month	day	s_mean	n_mean	tm_mean	o8_mean	tp_mean	h_mean
94	1	1	21.30	74.69	142.82	47.56	15.53	69.00
94	1	2	12.35	64.81	99.00	48.25	16.94	77.14
94	1	3	44.53	90.17	74.00	8.92	19.50	79.43
94	1	4	26.41	78.79	71.67	45.47	18.51	76.00
94	1	5	20.99	74.97	85.33	46.31	18.83	76.00

....

Example 11.2 *Female labor supply in East Germany. (1991)*

Goal: To study factors that affect the female labor supply.

A typical data entry reads like:

working hours	age	hourly earning	Job Pres	Year Edu	Mon Rent	husband earning	Child	Unempl. Rate
21	36	8.269	55	12	1010	2800	1	16.8
40	35	6.059	29	10	268	2500	1	16.8
30	33	11.5	34	12	605	3226	1	16.8
43	30	9.85	44	12	800	1800	1	16.8
45	43	15.16	60	13	280	2040	1	16.8
45	45	7.843	34	12	250	3200	0	16.8

....

Data Format:

	Response	*independent variables*			
case number	Y	X_1	X_2	$\cdots$	X_k
1	y_1	x_{11}	x_{12}	$\cdots$	x_{1k}
2	y_2	x_{22}	x_{23}	$\cdots$	x_{2k}
$\vdots$					
n	y_n	x_{n1}	x_{n2}	$\cdots$	x_{nk}

Multiple regression model:

$$\mathbf{Y} = \beta_\mathbf{0} + \beta_\mathbf{1}\mathbf{X_1} + \cdots + \beta_\mathbf{k}\mathbf{X_k} + \varepsilon,$$

where ε is the random error with $E\varepsilon = 0$ and $\mathrm{var}(\varepsilon) = \sigma^2$.

Group mean: Average response for the group with covariates $\mathbf{x}^* = (x_1^*, \cdots, x_k^*)$ is $E(Y|\mathbf{X} = \mathbf{x}^*) = \beta_0 + \beta_1 x_1^* + \cdots + \beta_k x_k^*$.

Group SD: $\mathrm{var}(Y|\mathbf{X} = \mathbf{x}^*) = \sigma$.

11.2 Parameter Estimation

Data: According to the multiple regression model,

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \varepsilon_i, \quad i = 1, \cdots, n.$$

Least-squares method: Find $\boldsymbol{\beta}$ to minimize

$$\mathrm{SSE}(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_k x_{ik})^2.$$

MLE: This is also the MLE if $\varepsilon_i \sim_{i.i.d.} N(0, \sigma^2)$.

Solution: Easy to obtain by calculus and linear algebra and widely implemented on computers. Let $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_0, \cdots, \widehat{\beta}_k)'$ be the solution.

Important statistical quantities.

- Fitted values: $\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \cdots + \widehat{\beta}_k x_{ik}$.
- Residuals: $\widehat{\varepsilon}_i = y_i - \widehat{y}_i$.
- Residual sum of squares: $\mathrm{SSE} = \sum_{i=1}^n \widehat{\varepsilon}_i^2$.
- Coefficient of determination (multiple R^2): $R^2 = 1 - \frac{\mathrm{SSE}}{S_{yy}}$, which is equal to the sum of squares reduction due to regression (SS_{reg}) divided by total sum of squares (SST$=S_{yy}$).
- Adjusted multiple R^2:

$$R_a^2 = 1 - \frac{n-1}{n-(k+1)}\frac{\mathrm{SSE}}{S_{yy}} = \frac{(n-1)R^2}{n-(k+1)} - \frac{k}{n-(k+1)},$$

adjusting for the number of parameters (that is, variables). **Used in variable selection**.

Est of σ^2: $\widehat{\sigma}^2 = \text{SSE}/(n-k-1)$, which is the MLE with adj.

Example 11.3 *Predicting macroeconomic variables*

129 macroecnomic time series, updated by Michael McCraken of Fed. St. Louis, is available on the class web. We focus on the variables:

```
macro = read.csv("macro2016-10.csv",header=T)  #read data
month = macro[,1]  #Months of Data
Month = strptime(month, "%m/%d/%Y") #convert to POSIXlt (a date class)
Unrate = macro[,25]       #Unemploy rates
IndPro = macro[,7]       #Industrial Production Index
HouSta = macro[,49]       #House start
PCE    = macro[,4]       #Real Personal Consumption
M2Real = macro[,67]               #Real M2 Money Stock
FedFund= macro[,79]               #Fed Funds Rate
CPI   =  macro[,107]              #Consumer Price Index
SPY   =  macro[,75]       # S&P 500 index
```

These 8 time series are depicted in Fig. 11.1.

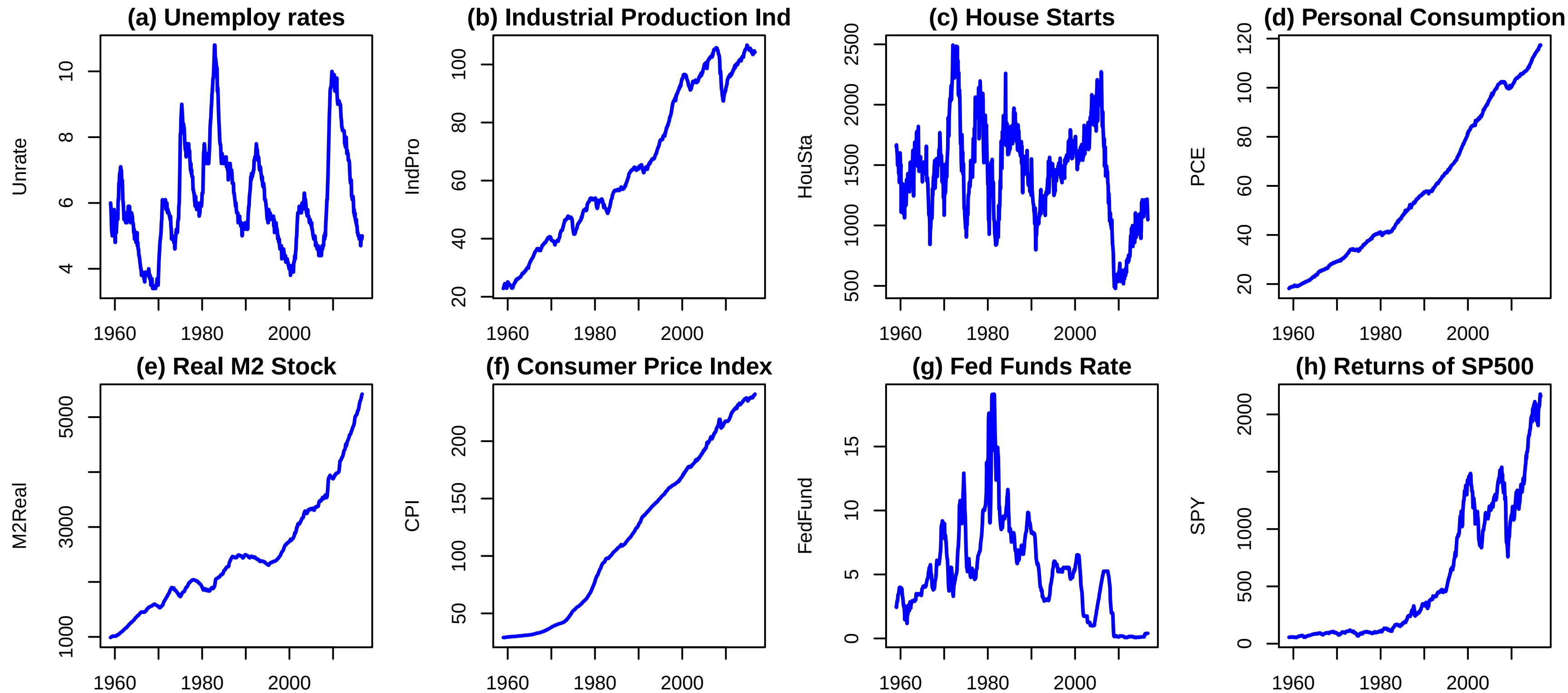

Figure 11.1: Macroeconomics time series from 1959–2016.

Since several variables are increasing, we take their log-differences:

$Y_t = \text{Unrate}_t, \quad X_{t,1} = \text{Unrate}_{t-1}, \; X_{t,2} = \Delta \log(\text{IndPro}_t), \; X_{t,3} = \Delta \log(\text{PCE})_t = \log(\text{PCE})_t - \log(\text{PCE})_{t-1},$

$X_{t,4} = \Delta \log(\text{M2Real}_t), \quad X_{t,5} = \Delta \log(\text{CPI}_t), \quad X_{t,6} = \Delta\log(\text{SPY})_t, \quad X_{t,7} = \text{HouSta}_t, \quad X_{t,8} = \text{FedFund}_t$

```
#### creating variables ################
DIndPro = diff(log(IndPro)) # changes of IndPro
```

```
DPCE = diff(log(PCE)) # changes of PCE
DM2  = diff(log(M2Real)) # chances of M2 stock
DCPI = diff(log(CPI)) # changes of CPI
DSPY = diff(log(SPY))            # log-returns of SP500

n = length(Unrate)
Y = Unrate[3:n]    #future unemployrate
X1 = cbind(DIndPro,DPCE, DM2, DCPI, DSPY) #present data
X2 = cbind(HouSta,FedFund)
X  = cbind(Unrate[2:(n-1)], X1[1:(n-2),], X2[2:(n-1),])
colnames(X) = list("lag1", "DIndPro", "DPCE", "DM2",
"DCPI", "DSPY", "HouSta", "FedFund")
           #give covariates names
```

Learning/training and testing sets: Take the last 10 years data as testing set and remaining as traing set.

```
n = length(Y)
Y.L = Y[1:(n-120)]        #learning set
Y.T = Y[(n-119):n]        #testing set
X.L = X[1:(n-120),]       #learning set
X.T = X[(n-119):n,]       #testing set
```

```
#Putting them as  data frames
data_train = data.frame(Unrate=Y.L, X.L) #give Y.L the name Unrate.
data_test  = data.frame(X.T)
```

Least-squares fit: We now use the training set to fit the model

```
> fitted=lm(Unrate ~ .,  data=data_train)   #fit model using learning data
   ### the short hand for
  lm(Unrate~lag1 + DIndPro + DPCE + DM2+ DCPI + DSPY + HouSta  + FedFund,
    data=data_train)
> summary(fitted)
Call:
lm(formula = Unrate ~ ., data = data_train)

Residuals:
    Min      1Q  Median      3Q     Max
-0.5755 -0.1018 -0.0068  0.1023  0.5771

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.2261704  0.0483233   4.680 3.59e-06 ***
lag1         0.9835370  0.0051298 191.731  < 2e-16 ***
```

```
DIndPro     -6.3738324  0.8835707  -7.214 1.77e-12 ***
DPCE        -3.2168829  1.2747416  -2.524   0.0119 *
DM2          3.1805548  2.0666216   1.539   0.1244
DCPI         5.4460126  3.5150404   1.549   0.1219
DSPY        -0.1432069  0.2025529  -0.707   0.4799
HouSta      -0.0001025  0.0000230  -4.456 1.01e-05 ***
FedFund      0.0047555  0.0026699   1.781   0.0754 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1611 on 562 degrees of freedom
Multiple R-squared:  0.9876,    Adjusted R-squared:  0.9874
F-statistic:  5579 on 8 and 562 DF,  p-value: < 2.2e-16
```

Estimated reg. equation: $\widehat{y} = .2262 + .9835 x_1 - 6.3738 x_2 + \cdots$

RSS: $\mathrm{SSE} = (n-k-1) * \widehat{\sigma}^2 = (571-8-1) * 0.1611^2 = 14.5857$

with d.f. $= 571 - 8 - 1 = 562.$

Multiple R^2: $= 1 - \mathrm{SSE}\,/(\mathrm{var}(Y.L) * (571-1)) = .9876$

SE: e.g., $\widehat{\beta}_1 = 0.9835$ and $\widehat{\text{SE}}(\widehat{\beta}_1) = 0.005130$.

Inferences about coefficients: For testing $H_0 : \beta_1 = 0$, the test statistic is $t = \frac{0.9835-0}{0.005130} = 191.731$. With alternative $H_1 : \beta_1 \neq 0$, we have the

$$\text{P-value} = P(|T_{562}| > 191.731) = 0\%.$$

The 95% confidence interval for β_1 is $0.9835 \pm 1.96 * 0.005130$.

Significant variables: Y_{t-1}, $\Delta\text{IndPro}_{t-1}$, ΔPCE_{t-1}, HouSta_{t-1}, FedFund_{t-1}.

$$\widehat{\sigma} = 0.1611, \qquad \text{adjusted Multiple } R^2 = 0.9874.$$

Prediction: Now use hold-out data for testing. For each given $\mathbf{x}_i^*$ in the testing set, compute

★ **Predicted value**: $\widehat{y}_i^* = \widehat{\beta}_0 + \widehat{\beta}_1 x_{i1}^* + \cdots + \widehat{\beta}_k x_{ik}^*.$

★ **Prediction error**: $\widehat{\varepsilon}_i = y_i - \widehat{y}_i^*.$

★ MSE $= n_*^{-1} \sum_i (y_i - y_i^*)^2$ and MADE $= n_*^{-1} \sum_i |y_i - y_i^*|$,

where n^* is the number of test cases.

```
Y.P = predict(fitted, newdata=data_test)  #predicted values at testing set

pdf("Fig112.pdf", width=8, height=2,  pointsize=10)
par(mfrow = c(1,2), mar=c(2, 4, 1.5,1)+0.1, cex=0.8)

plot(Month[(n-119):n], Y.T, type="l", col="red", lwd=2) #actual values
lines(Month[(n-119):n], Y.P, lty=2, col="blue")   #predicted values

rMSE = sqrt(mean((Y.T-Y.P)^2))     ### root mean-square prediction error
MADE = mean(abs(Y.T-Y.P))          ### mean absolute deviation error
> c(rMSE, MADE)
[1] 0.1685880 0.1335712
```

Fig. 11.2 depicts the results of prediction (quite well).

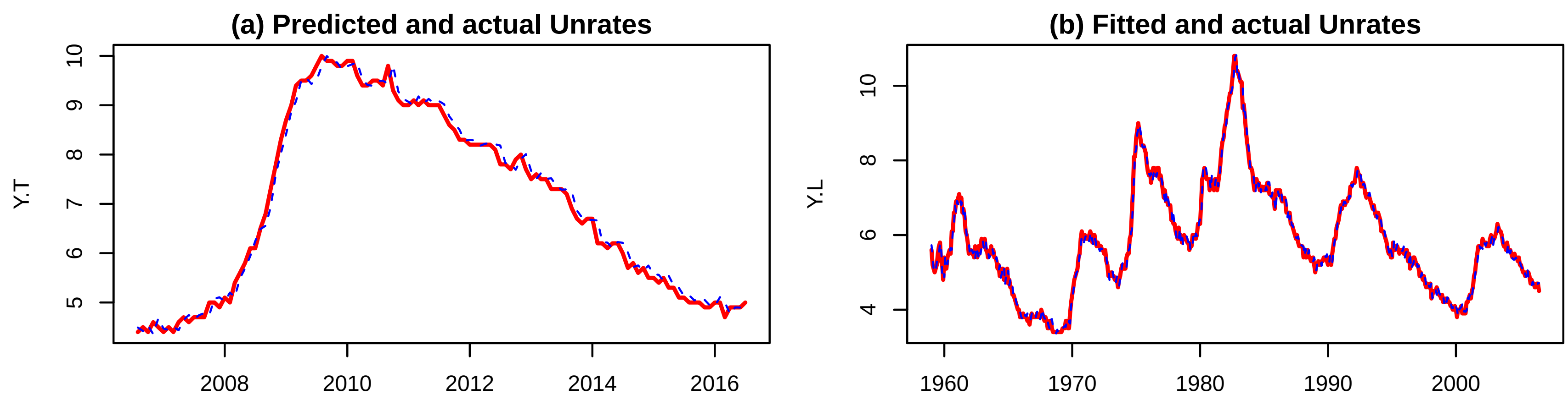

Figure 11.2: (a) Real and Predicted unemployment rate; (b) Observed and fitted unemployment rate.

Fitted values and Residuals:

```
fitted.values = fitted$fitted.values      #extract fitted values
residuals = fitted$residuals             #extract residuals

plot(Month[1:(n-120)], Y.L, type="l", col="red", lwd=2) #actual
lines(Month[1:(n-120)], fitted.values, lty=2, col="blue") #fitted
title("(b) Fitted and actual Unrates")
dev.off()
```

Residuals and Model Diagonistics: Plot residuals against time, covariates, and fitted values to see if there are any patterns. Standardized residuals $\widehat{\varepsilon}_i^* = \frac{\widehat{\varepsilon}_i}{\mathrm{SE}(\widehat{\varepsilon}_i)}$ are often used (better). See Fig. 11.3.

Diagnostic plots:

- Standardized residuals vs index or fitted or predictor values (i or $\widehat{y}_i$ or x_i vs $\widehat{\varepsilon}_i^*$). Ideal: No pattern of the plots.
- Fitted vs original values ($\widehat{y}_i$ vs y_i)
- Normal Q-Q plot for the standardized residuals.

```
pdf("Fig113.pdf", width=8, height=4,  pointsize=10)
par(mfrow = c(2,2), mar=c(2, 4, 1.5,1)+0.1, cex=0.8)

plot(Month[1:(n-120)], residuals, type="l", col="red", lwd=2) #residuals
title("(a) Time series plot of residuals")
plot(fitted.values, residuals, pch="*", col="red")
title("(b) Fitted versus residuals")

std.res = ls.diag(fitted)$std.res  #standardized residuals
```

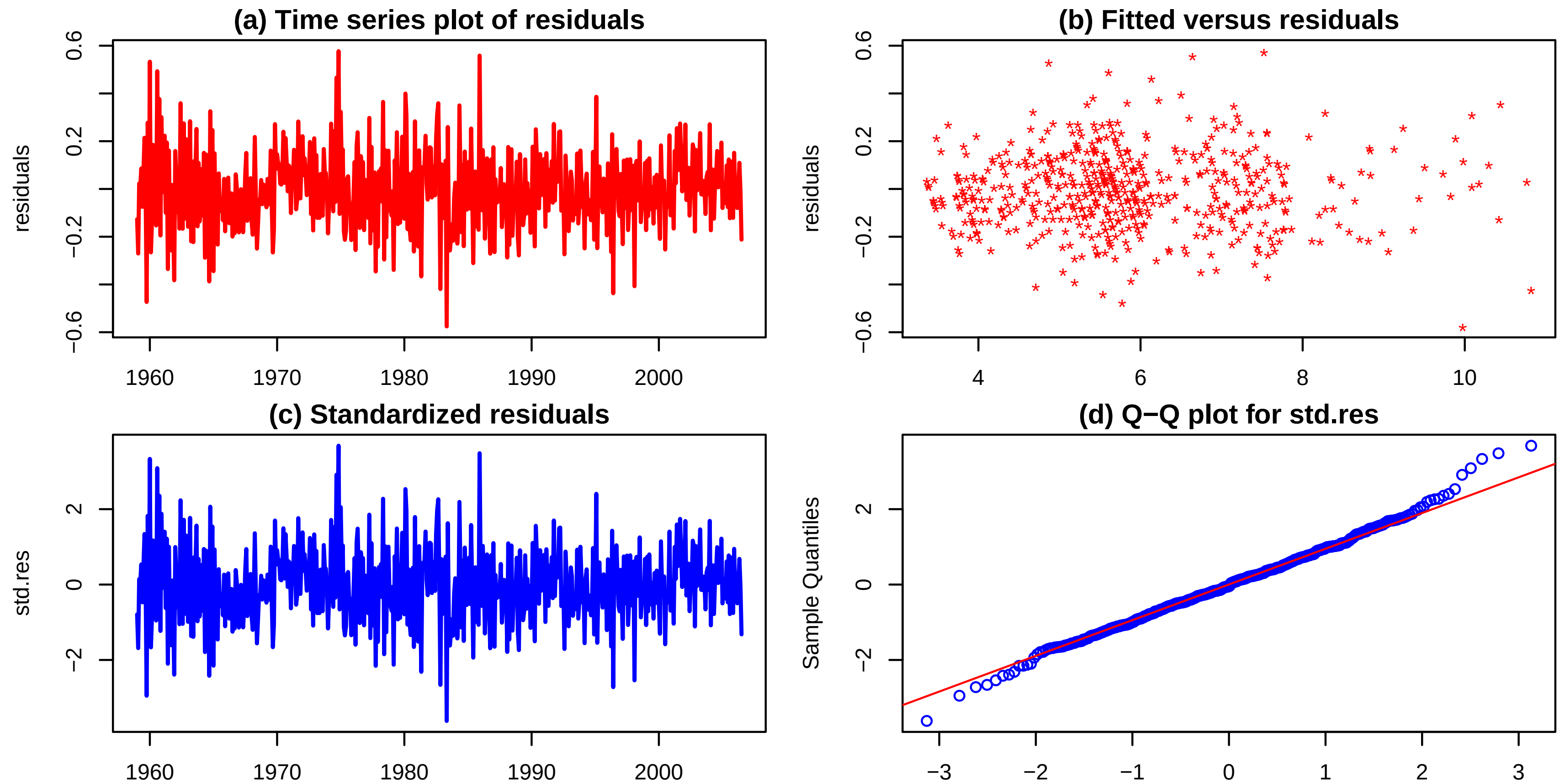

Figure 11.3: Model diagnostics using residuals and standardized residuals. Top panel using residuals and bottom panel using standardized residuals

```
plot(Month[1:(n-120)], std.res, type="l", col="blue", lwd=2) #residuals
title("(c) Standardized residuals")
qqnorm(std.res,col="blue", main="(d) Q-Q plot for std.res")
qqline(std.res, col="red")
dev.off()
```

Comparison: We use only lag1 alone to fit

```
fitted1 = lm(Unrate~lag1,data=data_train)
summary(fitted1)
Y.P1 = predict(fitted1, newdata=data_test)
rMSE1 = sqrt(mean((Y.T-Y.P1)^2))     ### root mean-square errors
MADE1 = mean(abs(Y.T-Y.P1))          ### mean absolute deviation error
c(rMSE1, MADE1)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.038911   0.031700   1.227     0.22
lag1        0.992958   0.005243 189.381   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1794 on 569 degrees of freedom
Multiple R-squared:  0.9844,    Adjusted R-squared:  0.9844
F-statistic: 3.587e+04 on 1 and 569 DF,  p-value: < 2.2e-16
> c(rMSE1, MADE1)
[1] 0.1885837 0.1383447
```

In terms of adjusted R^2, the fitting is worse. So are the test errors.

11.3 Cross-validation and Prediction errors

Learning & Testing: Divide data into two sets: S_L and S_T. Use S_L to fit a model, predict values in S_T and compare w/ actual values.

***k*-fold cross-validation**: Divide data randomly into k pieces (about the same size). Use any $k-1$ subsets of the data as training set and the remaining subset as the test set. Average all testing errors.

```
############Pseudo-code in R.  ##########
S = sample(1:n)                 #random permutation of index set
size = round(n/k)               #size of each testing set
for (j in 1:k)                  #loop through j, need to deal last block more carefully
{
t.start = (j-1)*size+1          #starting point of $j$ testing
t.end   =  j*size               #ending point of $j$ testing
S.T = S[t.start:t.end]          #index for testing set
S.L = S[-(t.start:t.end)]       #index for training set
data.L = data[S.L, ]            #test data
data.T = data[S.T,]             #training data
...... }
```

CV: When $k = n$, we use $n - 1$ data as learning and 1 as testing.

Bootstrap est of PE: sampling n_1 as training and the remaining as testing. Repeat B times and average PEs.

11.4 Analsis of Variance

Is a set of variables $\{x_i : i \in S\}$ **significant given others**? Formally,

$$H_0: \beta_i = 0, \text{ for all } i \in S \quad \longleftrightarrow \quad H_1: \beta_i \neq 0 \text{ for some } i \in S.$$

E.g. $S = \{2\} \Longrightarrow$ has *DIndPro* any significant contribution to *Unrate* given those of all others?

E.g. $S = \{1, ..., 8\} \Longrightarrow$ are all covariates related to *Unrate*?

Test statistic: Compare SSE using all variables with that without using variables in S, namely using $\{X_i : i \in S^c\}$. Clearly $\mathrm{SSE}(S^c) - \mathrm{SSE}(all)$ is the **additional contribution** (SSE reduction) of variables $\{X_i : i \in S\}$, after accounting for the contributions by $\{x_i : i \in S^c\}$.

$$\mathbf{F} = \frac{(\mathrm{SSE}(\mathbf{S^c}) - \mathrm{SSE}(\mathbf{all}))/\mathbf{p}}{\mathrm{SSE}(\mathbf{all})/(\mathbf{n} - \mathbf{k} - \mathbf{1})} \overset{\mathbf{H_0}}{\sim} \mathbf{F_{p,n-k-1}},$$

where p is the number of covariates involved in S. Thus,

$$\text{P-value} = P\{F_{p,n-k-1} \geq F_{obs}\}.$$

The results are often summarized in

```
> fitted2 = lm(Unrate~lag1 + DIndPro  + DPCE + HouSta  + FedFund, data=data_train)
> summary(fitted2)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.185e-01  4.805e-02   4.546 6.68e-06 ***
lag1         9.846e-01  4.948e-03 198.990  < 2e-16 ***
DIndPro     -6.406e+00  8.825e-01  -7.259 1.29e-12 ***
DPCE        -3.524e+00  1.235e+00  -2.855  0.00446 **
HouSta      -9.098e-05  2.206e-05  -4.125 4.26e-05 ***
FedFund      6.288e-03  2.188e-03   2.874  0.00421 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1612 on 565 degrees of freedom
Multiple R-squared:  0.9875,    Adjusted R-squared:  0.9874
F-statistic:  8917 on 5 and 565 DF,  p-value: < 2.2e-16

> anova(fitted, fitted2)
Analysis of Variance Table
```

```
Model 1: Unrate ~ lag1 + DIndPro + DPCE + DM2 + DCPI + DSPY + HouSta + FedFund
Model 2: Unrate ~ lag1 + DIndPro + DPCE + HouSta + FedFund
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    562 14.586
2    565 14.678 -3 -0.091976 1.1813 0.3161
```

11.5 Nonlinear regression §13.3

Polynomial regression of order k:

$$Y = \beta_0 + \beta_1 X + \cdots + \beta_k X^k + \varepsilon$$

is a multiple regression problem by setting $X_1 = X, \cdots, X_k = X^k$.

```
motor = read.table("motordata.txt", header=T, skip=3) #read data
x = motor[,1]; y = motor[,2]
pdf("Fig114.pdf", width=5, height=2,  pointsize=10)
par(mfrow = c(1,1), mar=c(2, 4, 1.5,1)+0.1, cex=0.8)
plot(motor,pch="*")     #scatter plot
###############polynomial fit  ###################
```

```
X = cbind(x, x^2, x^3)                    #cubic polynomials
fitted4 = lm(y~X)$fitted.values            #fitted
lines(x, fitted4, lwd=2, col="red")
```

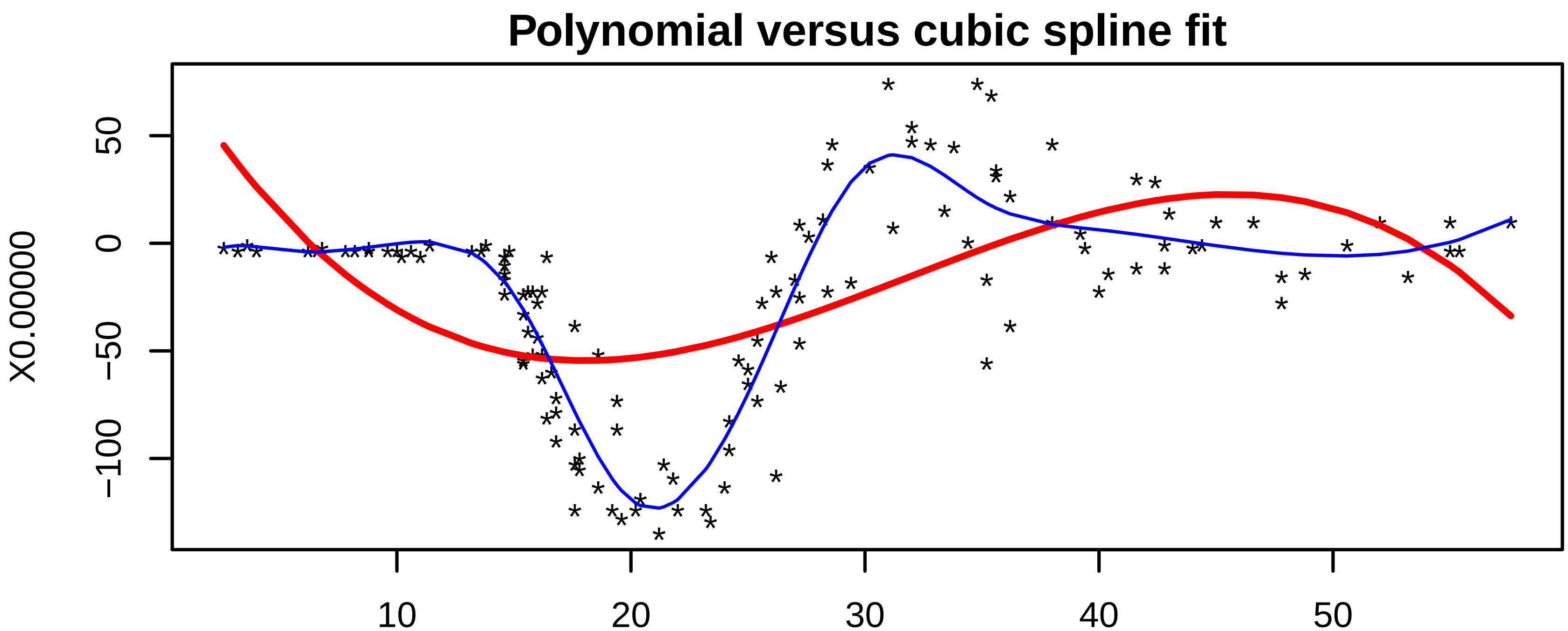

Figure 11.4: Scatter plot of time (in milliseconds) after a simulated impact on motorcycles against the head acceleration (in a) of a test object. Red = cubic polynomial fit, blue = cubic spline fit.

Cubic spline basis: For given knots $\{t_1, \cdots, t_m\}$,

$$B_1(x) = x, B_2(x) = x^2, B_3(x) = x^3, \quad B_{3+i}(x) = \begin{cases} (x - t_i)^3 & \text{if } x \geq t_i \\ 0 & \text{otherwise} \end{cases}$$

This is a much more very flexible basis.

Spline regression:

$$Y = \beta_0 + \beta_1 \underbrace{B_1(X)}_{X_1} + \cdots + \beta_{m+3} \underbrace{B_{m+3}(X)}_{X_{m+3}} + \varepsilon$$

```
############### cubic spline basis ##############
knots = seq(5,40,by=5)                  #creating knots
k = length(knots)                       #length of knots
X = matrix(rep(x, k),ncol=k) #repeating x, k times
X = t(t(X)- knots)     #col i = x - knot[i]
X = X^3
X[X < 0 ] = 0                             #cubic spline basis w knots
X = cbind(x, x^2, x^3, X) #cubic spline basis
```

```
fitted5 = lm(y~X)$fitted.values       #cubic spline fitted
lines(x, fitted5, col="blue")
title("Polynomial versus cubic spline fit")
dev.off()
```

11.6 Polynomials with several predictors§13.4

Quadratic regression: For bivariate

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{12} X_1 X_2 + \text{error}$$

The term $\beta_{12} X_1 X_2$ is the **interactions** between X_1 and X_2.

Interaction: commonly used form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2 + \text{error}.$$

Multiple regression: by defining

$$Z_1 = X_1,\ Z_2 = X_2,\ Z_3 = X_1^2,\ Z_4 = X_2^2,\ Z_5 = X_1X_2$$

we can see the multiple regression technique.

11.7 Model building using dummies §13.4

Dummy variables, also called **indicator variables**, are used to include categorical predictors in a regression analysis.

Example: Here are a few simple cases (Dichotomous):

$$\text{Gender}\begin{cases}\text{male}\\ \text{female}\end{cases} \qquad \text{Smoking}\begin{cases}\text{Yes}\\ \text{No}\end{cases} \qquad \text{Disease}\begin{cases}\text{present}\\ \text{not present}\end{cases}$$

For a dichotomous variable, we define $X = \begin{cases} 1, & \text{if treatment} \\ 0, & \text{if control} \end{cases}$

Example: Gender difference in income:

$$Y = \text{salary},\ X_1 = \text{age},\ X_2 = \text{year of exp.},\ X_3 = \begin{cases} 1, & \text{if male} \\ 0, & \text{if female} \end{cases}$$

The dummy variables can be used quite differently.

Possible models:

a) No-interaction model:

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \text{error} \\ &= \begin{cases} \beta_0 + \beta_1 X_1 + \beta_2 X_2 + e, & \text{for female} \\ (\beta_0 + \beta_3) + \beta_1 X_1 + \beta_2 X_2 + e, & \text{for male} \end{cases} \end{aligned}$$

■ β_3 is the gender difference after adjusting for X_1, X_2.

b) Complete interaction model.

$$
\begin{aligned}
Y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1 X_3 + \beta_5 X_2 X_3 + \text{error} \\
&= \begin{cases} \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error}, & \text{for female} \\ (\beta_{\mathbf{0}} + \beta_{\mathbf{3}}) + (\beta_{\mathbf{1}} + \beta_{\mathbf{4}}) X_1 + (\beta_{\mathbf{2}} + \beta_{\mathbf{5}}) X_5 + \text{error}, & \text{male} \end{cases}
\end{aligned}
$$

$\beta_3, \beta_4, \beta_5$ reflect the **gender diff.** wrt salary, age and exp.

c) Partial interaction model,

$$
\begin{aligned}
Y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{23} X_2 X_3 + \text{error} \\
&= \begin{cases} \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error}, & \text{for female} \\ (\beta_{\mathbf{0}} + \beta_{\mathbf{3}}) + \beta_1 X_1 + (\beta_{\mathbf{2}} + \beta_{\mathbf{23}}) X_2 + \text{error}, & \text{for male} \end{cases}
\end{aligned}
$$

Difference in intercept and experience, but fair in age.

More than two categories (polytomous): When a categorical predictor contains more than two categories, e.g. Race = { Black, white, Asian }. One way is to define

$$X_3 = \begin{cases} 0, & \text{if Black,} \\ 1, & \text{if White,} \\ 2, & \text{if Asian.} \end{cases}$$

but often **not** useful. For example,

$$\begin{aligned} \text{Salary} &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + e \\ &= \begin{cases} \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \mathbf{0} & \text{for black} \\ \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 & \text{for white} \\ \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \mathbf{2}\beta_3 & \text{for asian} \end{cases} \end{aligned}$$

Remedy: More than one dummy is needed. Define

$$X_3 = \begin{cases} 1 & \text{black} \\ 0 & \text{not black} \end{cases}, \qquad X_4 = \begin{cases} 1 & \text{white} \\ 0 & \text{not white} \end{cases}, \qquad X_5 = \begin{cases} 1 & \text{asian} \\ 0 & \text{not asian} \end{cases}$$

Note that $X_3 + X_4 + X_5 = 1 =$ intercept term, so only two of them can be used. Now, assume the linear model

$$\begin{aligned} \text{eg. Salary} &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + e \\ &= \begin{cases} \beta_0 + \beta_1 X_1 + \beta_2 X_2 & \text{for asian} \\ (\boldsymbol{\beta_0} + \boldsymbol{\beta_3}) + \beta_1 X_1 + \beta_2 X_2 & \text{for black} \\ (\boldsymbol{\beta_0} + \boldsymbol{\beta_4}) + \beta_1 X_1 + \beta_2 X_2 & \text{for white} \end{cases} \end{aligned}$$

Nolinear fits using dummies. We can divide DCPI into low, middle, and high inflations and use indicators to fit the unemployment.