CHAPTER 7

Polynomial Regression Models

Introduction

- A second-order polynomial in one variable:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

- A second-order polynomial in two variables:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon$$

Polynomial Models in One Variable

- One-variable form, again:

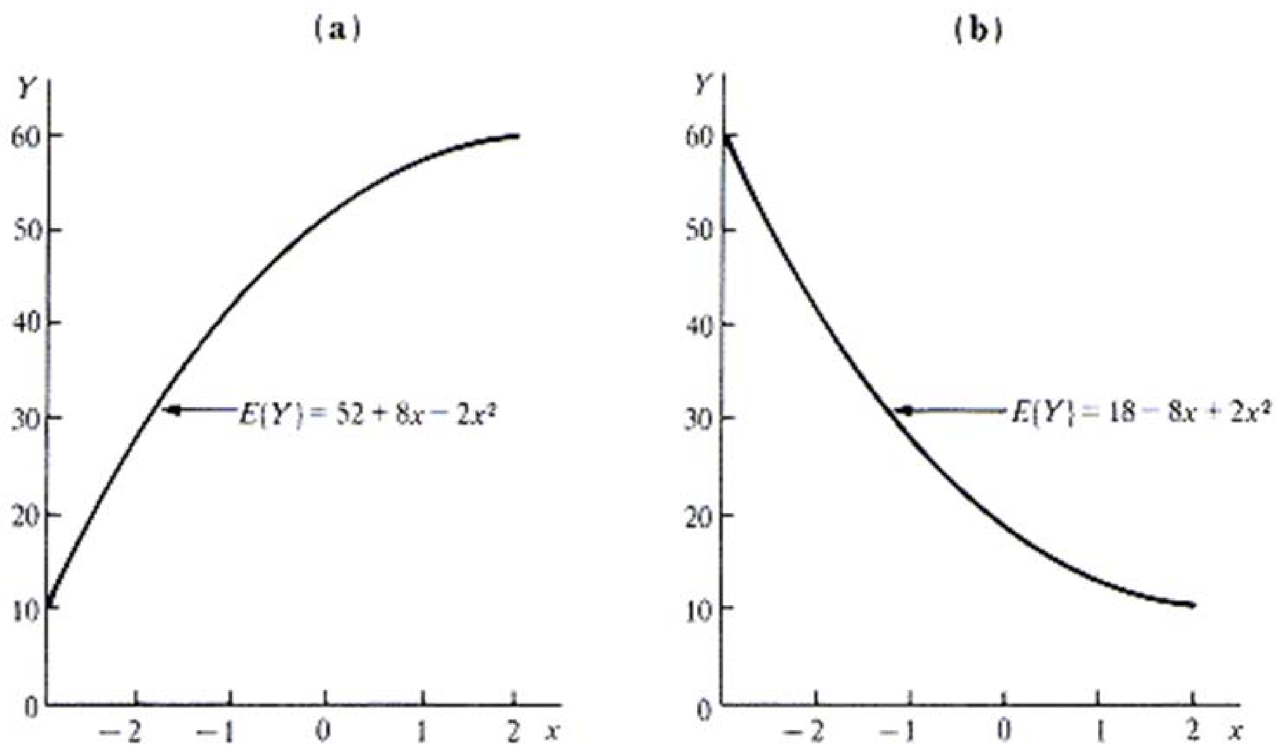
$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

- If $x_1 = x$ and $x_2 = x^2$, – standard linear regression analysis applies.
- The expectation of y for a one-variable polynomial model is

$$E(y) = \beta_0 + \beta_1 x + \beta_2 x^2$$

Polynomial Models in One Variable

- **One Predictor Variable – Second Order**



Polynomial Models in One Variable

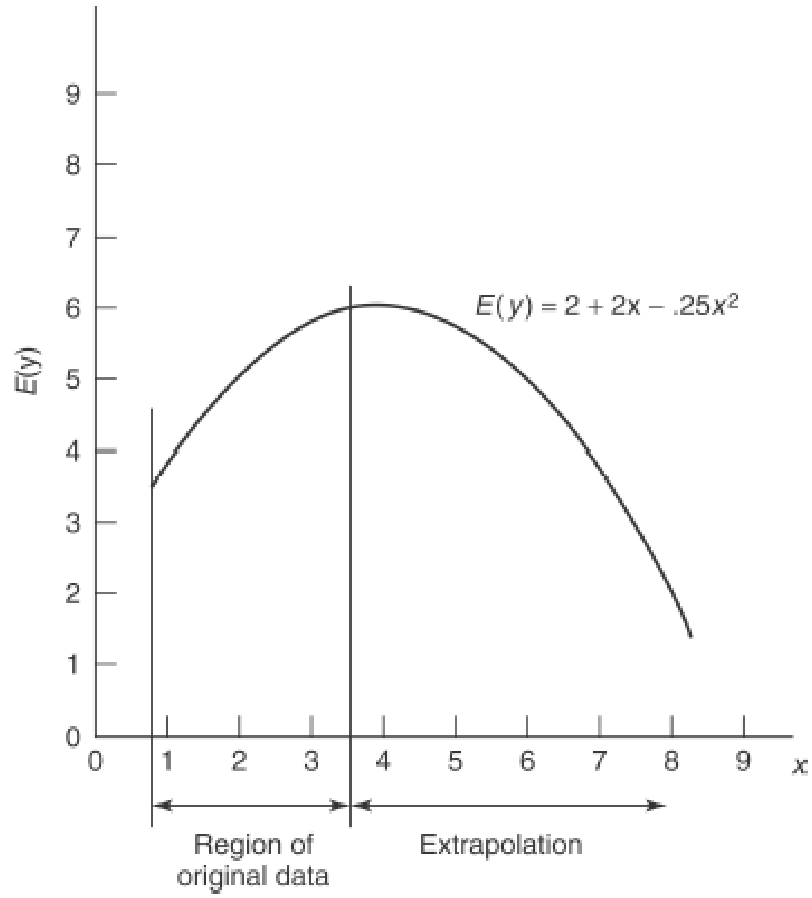


Figure 7.2 Danger of extrapolation.

Polynomial Models in One Variable

- **Cautions in fitting a polynomial in one-variable**
  1. Keep the order of the model as low as possible
     - This is especially true if you are using the model as a predictor
     - Transformations are often preferred over higher-order models
     - Parsimony – try to fit the data using the simplest model possible
     - Remember: An n – 1 order model can be fitted to a set of data with n points

Polynomial Models in One Variable

- **Cautions in fitting a polynomial in one-variable**

2. Model Building Strategy
   - One approach is fitting the lowest order polynomial possible and build up (forward selection).
   - Second approach is fitting the highest order polynomial of interest, and removing terms (backward elimination).
   - Remember: The same result may not be obtained from the two approaches

Polynomial Models in One Variable

- **Cautions in fitting a polynomial in one-variable**
  3. Extrapolation
     - Can be dangerous when the model is a higher-order polynomial
     - The nature of the true underlying relationship may change or be completely different than the system that produced the data used to fit the model

Polynomial Models in One Variable

- **Cautions in fitting a polynomial in one-variable**

4. Ill-conditioning
   - Ill-conditioning refers to the fact that as the order of the model increases, the **X'X** matrix inversion will become inaccurate–error can be introduced into the parameter estimates
   - As the order of the model ↑, multicollinearity ↑
   - Centering the variables first, may remove some ill-conditioning but not all
   - Narrow ranges on the *x* variables can result in significant ill-conditioning and multicollinearity problems.

Polynomial Models in One Variable

- **Cautions in fitting a polynomial in one-variable**

5. Hierarchy
   - Hierarchical model is one which, if it is of order *n*, then it contains all terms with orders of *n* and below:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_{n-1} x^{n-1} + \beta_n x^n \varepsilon$$

   - Two schools of thought: 1) Maintain hierarchy and 2) Maintaining hierarchy is not important.
   - Suggestion: Fit the model with only significant terms and use knowledge and understanding of the process to determine if a hierarchical model is necessary

Polynomial Models in One Variable

- **Centering**
  - Sometimes, centering the regressor variables can minimize or eliminate at least some of the ill-conditioning that may be present in a polynomial model

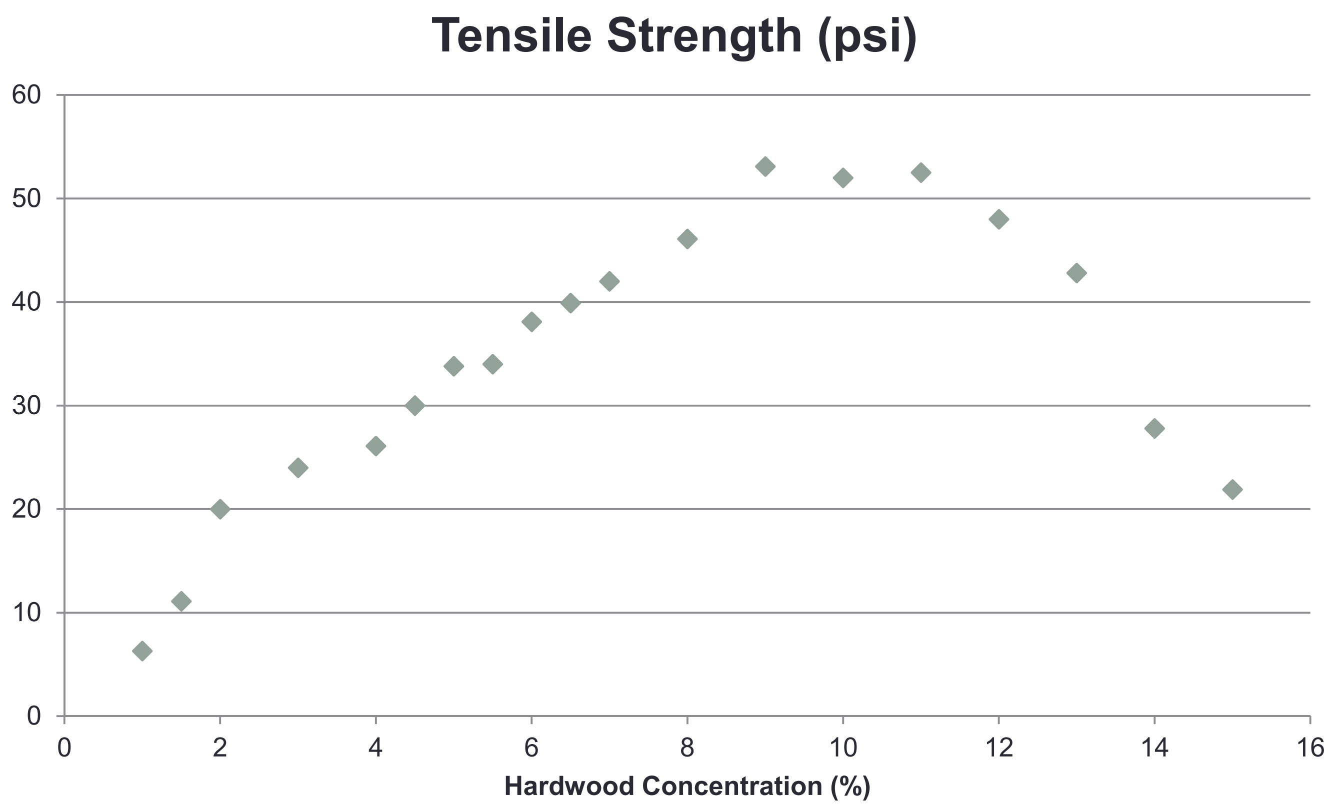
TABLE 7.1 Hardwood Concentration in Pulp and Tensile Strength of Kraft Paper, Example 7.1

Hardwood Concentration, x_i (%)	Tensile Strength, (psi) y, (psi)
1	6.3
1.5	11.1
2	20.0
3	24.0
4	26.1
4.5	30.0
5	33.8
5.5	34.0
6	38.1
6.5	39.9
7	42.0
8	46.1
9	53.1
10	52.0
11	52.5
12	48.0
13	42.8
14	27.8
15	21.9

R code

- rm(list=ls())
- Paper <- read.csv("data-ex-7-1-(Hardwood).csv",h=T)
- plot(Paper$HwdCon,Paper$TsStr)
- # fit polynomial regression with order 2 note the function I()
- model1 <- lm(Paper$TsStr ~ Paper$HwdCon+ I(Paper$HwdCon^2))
- summary(model1)
- library(car)
- vif(model1)
- cor(Paper$HwdCon, Paper$HwdCon^2)
- # plot data
- plot(Paper$HwdCon,Paper$TsStr)
- # plot the fitted values using dots
- points(Paper$HwdCon,model1$fitted.values,pch=20)
- # plot the fitted values using a line
- points(Paper$HwdCon,model1$fitted.values,type="l")

Example 7.1


Tensile Strength (psi)
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
Hardwood Concentration (%)

Example 7.1

- Consider the regression analysis provided below:

```
The regression equation is
y = - 6.67 + 11.8 x - 0.635 x2

Predictor        Coef     SE Coef        T       P      VIF
Constant       -6.674       3.400    -1.96   0.067
x              11.764       1.003    11.73   0.000     17.1
x2           -0.63455     0.06179   -10.27   0.000     17.1

S = 4.420       R-Sq = 90.9%     R-Sq(adj) = 89.7%

Analysis of Variance
Source            DF          SS          MS         F        P
Regression         2      3104.2      1552.1     79.43    0.000
Residual Error    16       312.6        19.5
Total             18      3416.9
```

- Note that the variance inflation factors indicate that multicollinearity may be a problem

Example 7.1

	Hardwood Concentration, x_i (%)	*Hardwood Concentration Squared, x_i^2 (%)*
Hardwood Concentration, x_i (%)	100.00%	97.04%
Hardwood Concentration Squared, x_i^2 (%)	97.04%	100.00%

Example 7.1

- Center the data using the mean of the regressor variable:

$x_i - 7.2632$	$(x_i - 7.2632)^2$	y
-6.2632	39.2277	6.3
-5.7632	33.2145	11.1
-5.2632	27.7013	20.0
-4.2632	18.1749	24.0
-3.2632	10.6485	26.1
-2.7632	7.6353	30.0
-2.2632	5.1221	33.8
-1.7632	3.1089	34.0
-1.2632	1.5957	38.1
-0.7632	0.5825	39.9
-0.2632	0.0693	42.0
0.7368	0.5429	46.1
1.7368	3.0165	53.1
2.7368	7.4901	52.0
3.7368	13.9637	52.5
4.7368	22.4373	48.0
5.7368	32.9109	42.8
6.7368	45.3845	27.8
7.7368	59.8581	21.9

R code

- # center the observations
- Paper$HwdCon_Centered=Paper$HwdCon-mean(Paper$HwdCon)
- # refit polynomial regression using centered observations
- model2 <- lm(Paper$TsStr ~ Paper$HwdCon_Centered+ I(Paper$HwdCon_Centered^2))
- summary(model2)
- library(car)
- # check VIF
- vif(model2)
- # check correlation
- cor(Paper$HwdCon_Centered, Paper$HwdCon_Centered^2)

Example 7.1(Centered)

	Hardwood Concentration, x_i (%)	*Hardwood Concentration Squared, x_i^2 (%)*
Hardwood Concentration, x_i (%)	100.00%	29.74%
Hardwood Concentration Squared, x_i^2 (%)	29.74%	100.00%

Example 7.1(Centered)

$$y = \beta_0 + \beta_1(x - 7.2632) + \beta_2(x - 7.2632)^2 + \varepsilon$$

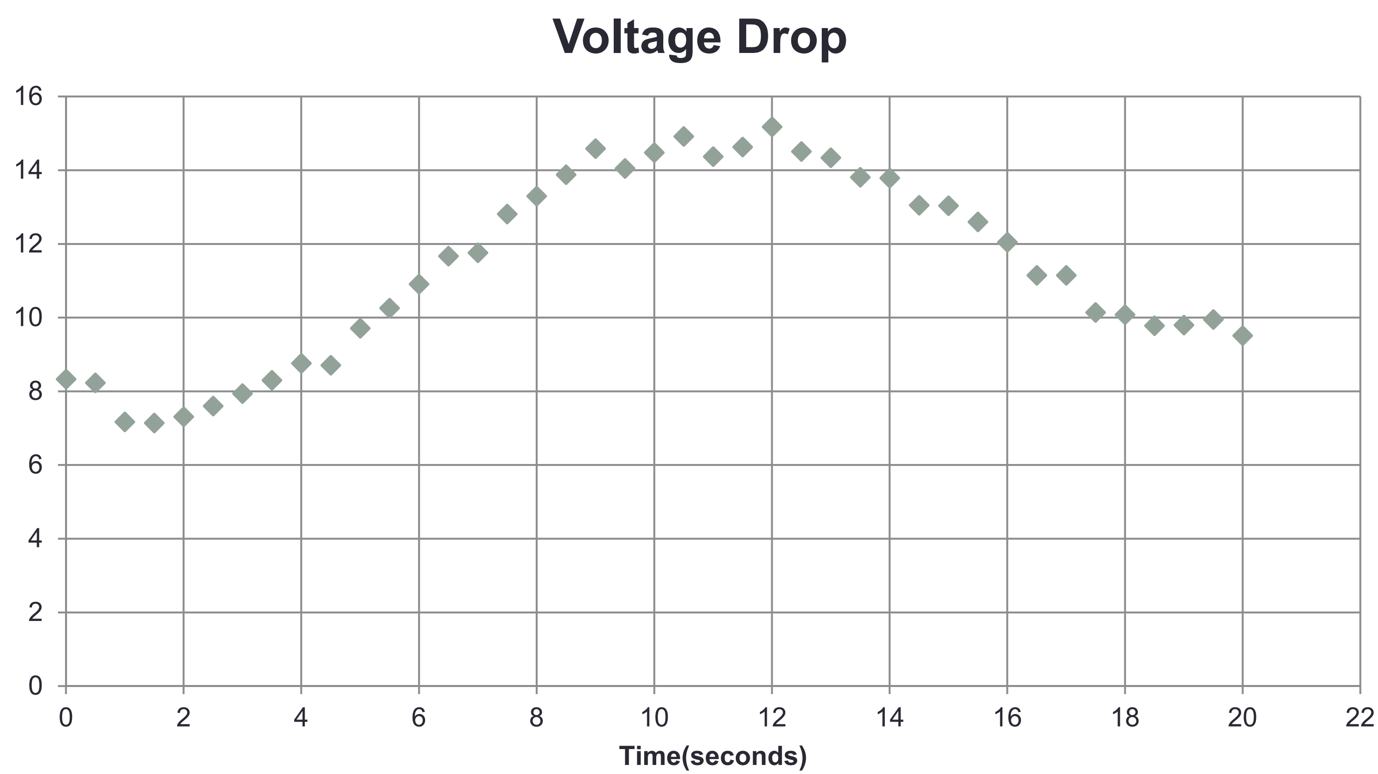
```
The regression equation is
y = 45.3 + 2.55 xcent - 0.635 x2cent

Predictor        Coef     SE Coef         T       P       VIF
Constant       45.295       1.483     30.55   0.000
xcent          2.5463      0.2538     10.03   0.000       1.1
x2cent       -0.63455     0.06179    -10.27   0.000       1.1

S = 4.420      R-Sq = 90.9%     R-Sq(adj) = 89.7%

Analysis of Variance
Source           DF          SS          MS        F        P
Regression        2      3104.2      1552.1    79.43    0.000
Residual Error   16       312.6        19.5
Total            18      3416.9
```

Example 7.2 – Voltage Drop Data



R code

```
# example 7.2
Voltage <- read.csv("data-ex-7-2-(Voltage-Drop).csv",h=T)
# visualize data
plot(Voltage$Time,Voltage$VoltageDrop)
# build polynomial regression
model3=lm(Voltage$VoltageDrop~Voltage$Time+I(Voltage$Time^2))
# plot fitted values
par(mfrow=c(1,2))
plot(Voltage$Time,Voltage$VoltageDrop)
points(Voltage$Time,model3$fitted.values,pch=20)
points(Voltage$Time,model3$fitted.values,type="l")
plot(Voltage$Time,model3$residuals)
abline(h=0,col="grey")
# build polynomial regression
model3=lm(Voltage$VoltageDrop~Voltage$Time+I(Voltage$Time^2)+I(Voltage$Time^3))
par(mfrow=c(1,2))
plot(Voltage$Time,Voltage$VoltageDrop)
points(Voltage$Time,model3$fitted.values,pch=20)
points(Voltage$Time,model3$fitted.values,type="l")
plot(Voltage$Time,model3$residuals)
abline(h=0,col="grey")
# build polynomial regression
model3=lm(Voltage$VoltageDrop~Voltage$Time+I(Voltage$Time^2)+I(Voltage$Time^3)+I(Voltage$Time^4))
par(mfrow=c(1,2))
plot(Voltage$Time,Voltage$VoltageDrop)
points(Voltage$Time,model3$fitted.values,pch=20)
points(Voltage$Time,model3$fitted.values,type="l")
plot(Voltage$Time,model3$residuals)
abline(h=0,col="grey")
```

Optional reading

- Piecewise polynomial regression

Piecewise Polynomial Fitting (Splines)

- This is a technique that can be used if a particular function behaves differently for different ranges of x
- Generally, divide the range of x into “homogeneous” segments and fit an appropriate function in each section

Piecewise Polynomial Fitting (Splines)

Splines:

- Splines are piecewise polynomials of order *k*
- Splines have **knots** – the points at which the segments are joined
- Too many knots can result in “overfitting” and will not necessarily provide more insight into the system.
- Usually, a cubic spline is sufficient – polynomial of order 3

Piecewise Polynomial Fitting (Splines)

- Cubic Spline with *continuous* first and second derivatives
- There are *h* knots, $t_1 < t_2 < \ldots < t_h$. This cubic spline is given by:

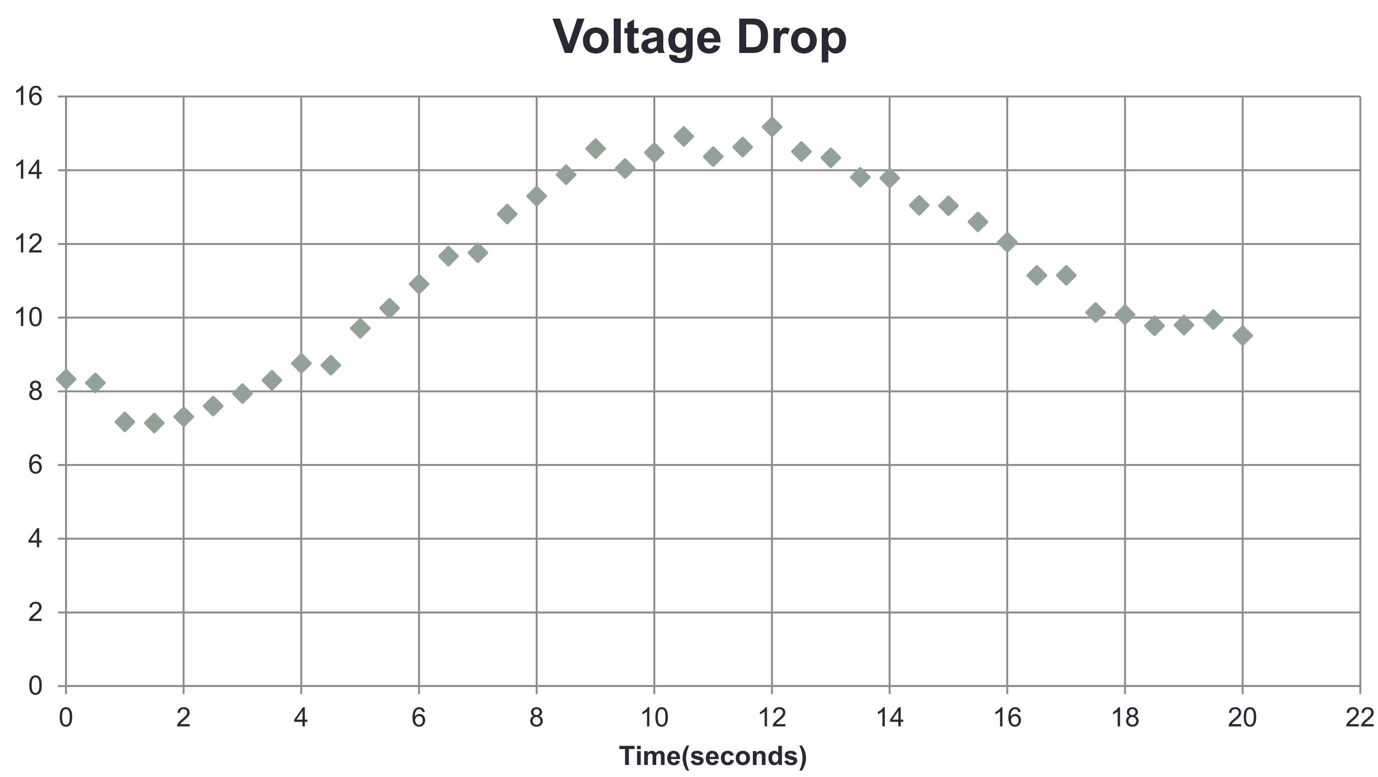
$$E(y) = S(x) = \sum_{j=0}^{3} \beta_{0j} x^j + \sum_{i=1}^{h} \beta_i (x - t_i)_+^3$$

with

$$(x - t_i)_+ = \begin{cases} x - t_i, & x > t_i \\ 0, & x \le t_i \end{cases}$$

- Consider $(x - t_i)_+$ as an "indicator variable" – that is, "on" or "off"

Example 7.2 – Voltage Drop Data



R code

```
# example 7.2
Voltage <- read.csv("data-ex-7-2-(Voltage-Drop).csv",h=T)
# visualize data
plot(Voltage$Time,Voltage$VoltageDrop)
# build polynomial regression
model3=lm(Voltage$VoltageDrop~Voltage$Time+I(Voltage$Time^2))
# plot fitted values
points(Voltage$Time,model3$fitted.values,pch=20)
points(Voltage$Time,model3$fitted.values,type="l")
# residual plot
plot(model3$fitted.values,model3$residuals)
abline(h=0)
```

Example 7.2 – Voltage Drop Data

The regression equation is
y = 5.27 + 1.49 x - 0.0652 x2

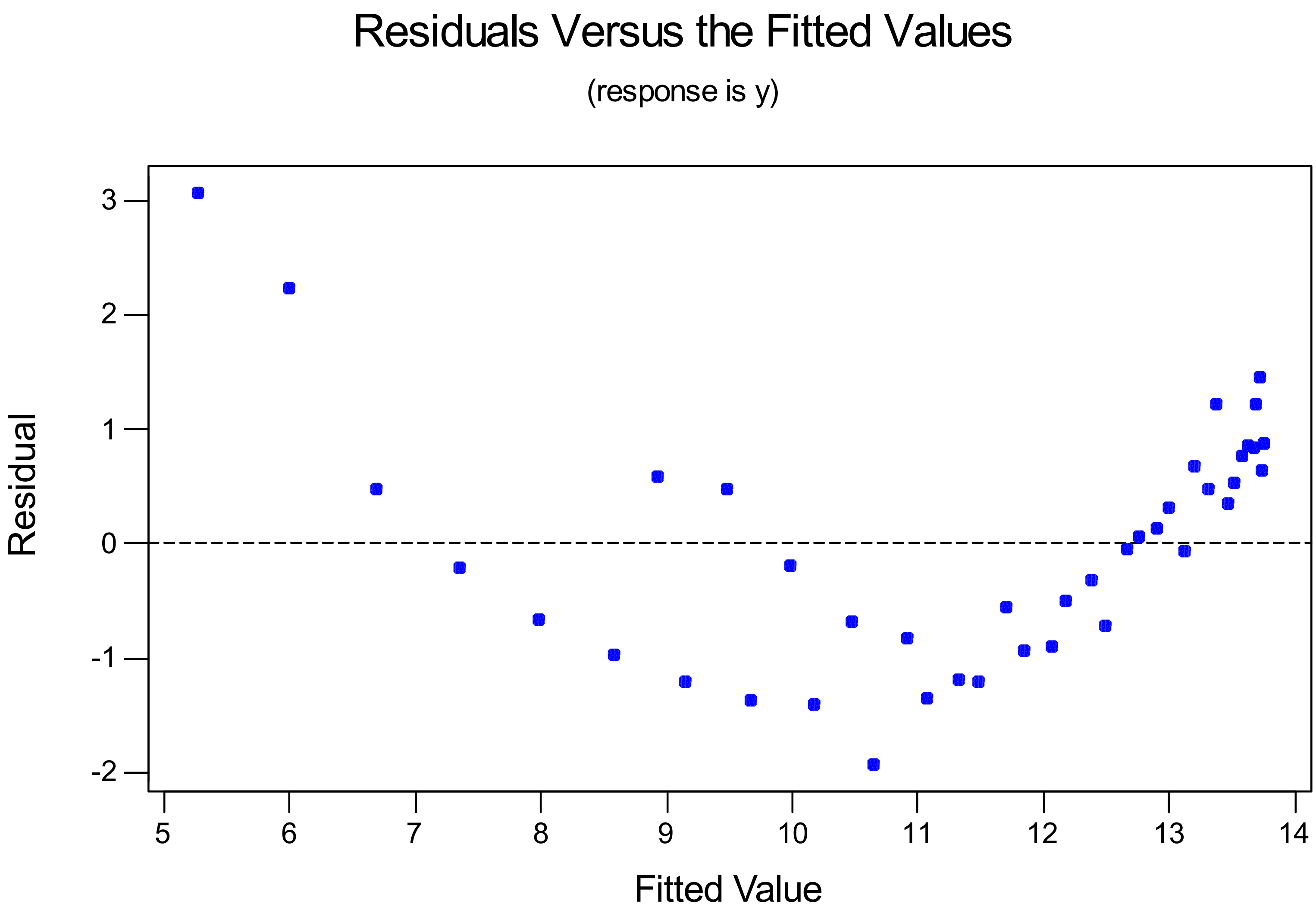
Predictor	Coef	SE Coef	T	P	VIF
Constant	5.2657	0.4807	10.95	0.000	
x	1.4872	0.1112	13.37	0.000	15.3
x2	-0.065198	0.005375	-12.13	0.000	15.3

S = 1.076 R-Sq = 83.2% R-Sq(adj) = 82.4%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	2	218.66	109.33	94.35	0.000
Residual Error	38	44.03	1.16		
Total	40	262.69			

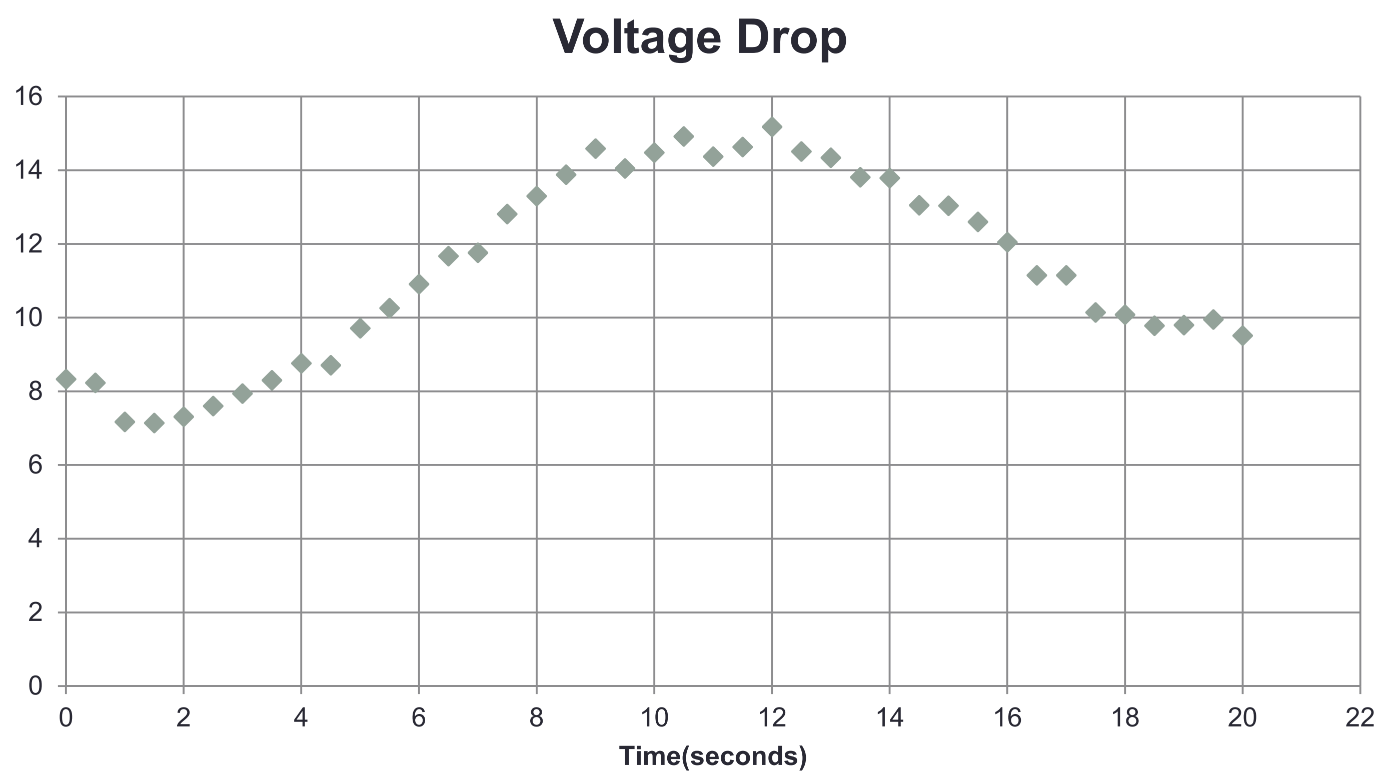
Example 7.2 – Voltage Drop Data



Example 7.2 – Voltage Drop Data

- A cubic spline is now investigated. Based on the plot of the original data and knowledge of the process, two knots are chosen.
- It appears that voltage behaves different between time 0 and 6.5 seconds than it does between 6.5 and 13 seconds.
- It appears to behave differently yet again after 13 seconds.
- Therefore, h = 2 knots are chosen to be $t_1 = 6.5$ and $t_2 = 13$.
- Deciding the number and locations of knots is difficult. At least a few data points in each interval. No more than one extreme value in each interval. Prior knowledge.

Example 7.2 – Voltage Drop Data



Example 7.2 – Voltage Drop Data

- The cubic spline model is

$$y = \beta_{00} + \beta_{01}x + \beta_{02}x^2 + \beta_{03}x^3 + \beta_1(x - 6.5)_+^3 + \beta_2(x - 13)_+^3 + \varepsilon$$

R code

- library(splines)
- # quadratic term
- Voltage$x2 <- Voltage$Time^2
- # qubic term
- Voltage$x3 <- Voltage$Time^3
- # knot at 6.5
- Voltage$x65 <- ifelse ((Voltage$Time>6.5), (Voltage$Time-6.5)^3, 0)
- # knot at 13
- Voltage$x13 <- ifelse ((Voltage$Time>13), (Voltage$Time-13)^3, 0)
- # fit spline regression
- model4 <- lm(Voltage$VoltageDrop ~ Voltage$Time+Voltage$x2+Voltage$x3+Voltage$x65+Voltage$x13)
- summary(model4)

- # check the fit
- par(mfrow=c(1,2))
- plot(Voltage$Time,Voltage$VoltageDrop)
- points(Voltage$Time,model4$fitted.values,pch=20)
- points(Voltage$Time,model4$fitted.values,type="l")
- # residual plot
- plot(model4$fitted.values,model4$residuals)
- abline(h=0)

Example 7.2 – Voltage Drop Data

y = 8.47 - 1.45 x + 0.490 x2 - 0.0295 x3 + 0.0247 x65 + 0.0271 x13

Predictor	Coef	SE Coef	T	P
Constant	8.4657	0.2005	42.22	0.000
x	-1.4531	0.1816	-8.00	0.000
x2	0.48989	0.04302	11.39	0.000
x3	-0.029467	0.002848	-10.35	0.000
x65	0.024706	0.004039	6.12	0.000
x13	0.027112	0.003578	7.58	0.000

Example 7.2 – Voltage Drop Data

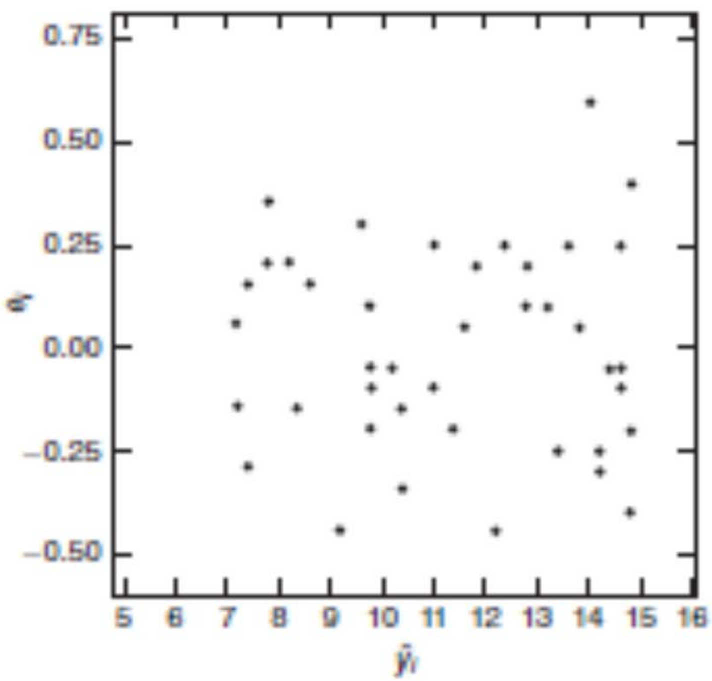


Figure 7.7 Plot of residuals e_i, versus fitted values $\hat{y}_i$ for the cubic spline model.

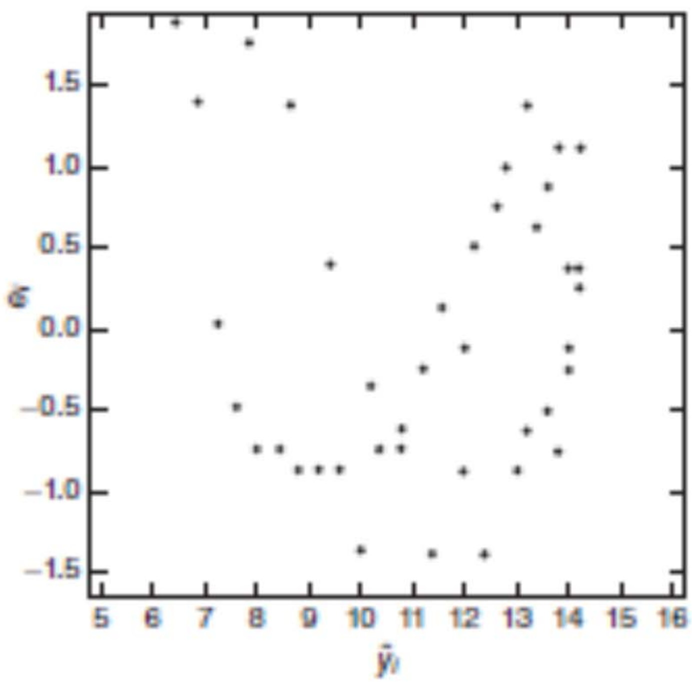


Figure 7.8 Plot of residuals e_i, versus fitted values $\hat{y}_i$ for the cubic polynomial model.