CHAPTER 4

Model Adequacy Checking

Introduction

- Assumptions
  1. Relationship between response and regressors is linear (at least approximately).
  2. Error term, ε has zero mean
  3. Error term, ε has constant variance
  4. Errors are uncorrelated
  5. Errors are normally distributed (required for tests and intervals)

Residual Analysis

- Definition of Residual (= data – fit):

$$e_i = y_i - \hat{y}_i$$

- Approximate average variance:

$$\frac{\sum (e_i - \bar{e})^2}{n-p} = \frac{\sum e_i^2}{n-p} = \frac{SS_{\text{Res}}}{n-p} = MS_{\text{Res}}$$

- This is important quantity because it is the estimate of variance of residuals, or σ^2.

Methods for Scaling Residuals

- Scaling helps in identifying **outliers** or extreme values
- Four Methods
  1. Standardized Residuals
  2. Studentized Residuals
  3. PRESS Residuals
  4. *R*-student Residuals

Methods for Scaling Residuals

1. Standardized Residuals

$$d_i = \frac{e_i}{\sqrt{MS_{\text{Res}}}}$$

- d_i's have mean zero and variance *approximately* equal to 1
- Large values of d_i ($d_i > 3$) may indicate an outlier

Methods for Scaling Residuals

2. Studentized Residuals
   - MS_{Res} is only an approximation of the variance of the *i*th residual.
   - Improve scaling by dividing e_i by the exact standard deviation:

$$Var(e_i) = \sigma^2(1 - h_{ii})$$

What is h_{ii}?

- Elements in the Hat matrix
- In simple linear regression case:
  - $h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
- The second part of right hand side of the above equation represents how far the data point is away from the center of the data set.
- If h_{ii} is large, a high leverage point (potential to affect to linear regression).
- If h_{ii} is small, a low leverage point.

Methods for Scaling Residuals

2. Studentized Residuals

The studentized residuals are then:

$$r_i = \frac{e_i}{\sqrt{\hat{\sigma}^2(1-h_{ii})}} = \frac{e_i}{\sqrt{MS_{\mathrm{Res}}(1-h_{ii})}}$$

- r_i's have mean zero and unit variance.
- Studentized residuals are generally larger than the corresponding standardized residuals.

Methods for Scaling Residuals

3. PRESS Residuals (predicted residual sum of squares)

   Examine the differences: $y_i - \hat{y}_{(i)}$

   - These are the differences between the actual response for the *i*th data point and the fitted value of the response for the *i*th data point, using all observations except the *i*th one.

Methods for Scaling Residuals

3. PRESS Residuals

Logic:

- If the *i*th point is unusual, then it can "overly" influence the regression model.
- If the *i*th point is used in fitting the model, then the residual for the *i*th point will be impacted by the *i*th point.
- If the *i*th point is not used in fitting the model, then the residual will better reflect how unusual that point is.

Methods for Scaling Residuals

3. PRESS Residuals

Prediction error: $e_{(i)} = y_i - \hat{y}_{(i)}$

- Calculate the PRESS residuals using

$$e_{(i)} = \frac{e_i}{1 - h_{ii}}$$

Methods for Scaling Residuals

3. PRESS Residuals

$$Var\left[e_{(i)}\right] = Var\left[\frac{e_i}{1-h_{ii}}\right]$$

$$= \frac{1}{(1-h_{ii})^2}\sigma^2(1-h_{ii})$$

$$= \frac{\sigma^2}{(1-h_{ii})}$$

Methods for Scaling Residuals

3. PRESS Residuals
   - The **standardized** PRESS residuals are

$$\frac{e_{(i)}}{\sqrt{Var(e_{(i)})}} = \frac{e_i/(1-h_{ii})}{\sqrt{\sigma^2/(1-h_{ii})}} = \frac{e_i}{\sqrt{\sigma^2(1-h_{ii})}}$$

   - **Note:** these are the studentized residuals when MS_{Res} is used as the estimate of the variance.

Methods for Scaling Residuals

4. R-Student
   - MS_{Res} is an “internal” estimate of variance
   - Use a variance estimate that is based on all observations except the *i*th observation:

$$S_{(i)}^{2} = \frac{(n - p)MS_{\text{Res}} - \frac{e_i^2}{(1 - h_{ii})}}{n - p - 1}$$

Methods for Scaling Residuals

4. R-Student

- The *R*-student residual is

$$t_i = \frac{e_i}{\sqrt{S_{(i)}^2(1 - h_{ii})}}$$

- This is an *externally studentized* residual.

TABLE 4.1 Scaled Residuals for Example 4.1

Observation Number, i	$e_i = y_i - \hat{y}_i$ (1)	$d_i = e_i/\sqrt{MS_{Res}}$ (2)	$r_i = e_i/\sqrt{MS_{Res}(1-h_{ii})}$ (3)	h_{ii} (4)	$e_{(i)} = e_i/(1-h_{ii})$ (5)	$t_i = e_i/\sqrt{S_{(i)}^2(1-h_{ii})}$ (6)	$[e_i/(1-h_{ii})]^2$ (7)
1	−5.0281	−1.5426	−1.6277	0.10180	−5.5980	−1.6956	31.3373
2	1.1464	0.3517	0.349	0.07070	1.2336	0.3575	1.5218
3	−0.0498	−0.0153	−0.0161	0.09874	−0.0557	−0.0157	0.0031
4	4.9244	1.5108	1.5798	0.05838	5.2297	1.6392	27.3499
5	−0.4444	−0.1363	−0.1418	0.07501	−0.4804	−0.1386	0.2308
6	−0.2896	−0.0888	−0.0908	0.04287	−0.3025	−0.0887	0.0915
7	0.8446	0.2501	0.2704	0.08180	0.9198	0.2646	0.8461
8	1.1566	0.3548	0.3667	0.06373	1.2353	0.3594	1.5260
9	7.4197	2.2763	3.2138	0.49829	14.7888	4.3108	218.7093
10	2.3764	0.7291	0.8133	0.19630	2.9568	0.8068	8.728
11	2.2375	0.6865	0.7181	0.08613	2.4484	0.7099	5.9946
12	−0.5930	−0.1819	−0.1932	0.11366	−0.6690	−0.1890	0.4476
13	1.0270	0.3151	0.3252	0.06113	1.0938	0.3185	1.1965
14	1.0675	0.3275	0.3411	0.07824	1.1581	0.3342	1.3412
15	0.6712	0.2059	0.2103	0.04111	0.7000	0.2057	0.4900
16	−0.6629	−0.2034	−0.2227	0.16594	−0.7948	−0.2178	0.6317
17	0.4364	0.1339	0.1381	0.05943	0.4640	0.1349	0.2153
18	3.4486	1.0580	1.1130	0.09626	3.8159	1.1193	14.5612
19	1.7932	0.5502	0.5787	0.09645	1.9846	0.5698	3.9387
20	−57880	−1.7758	−1.8736	0.10169	−6.4432	−1.9967	41.5150
21	−2.6142	−0.8020	−0.8779	0.16528	−3.1318	−0.8731	9.8084
22	−3.6865	−1.1310	−1.4500	0.39158	−6.0591	−1.4896	36.7131
23	−4.6076	−1.4136	−1.4437	0.04126	−4.8059	−1.4825	23.0966
24	−4.5728	−1.4029	−1.4961	0.12061	−5.2000	−1.5422	27.0397
25	−0.2126	−0.0652	−0.0675	0.06664	−0.2278	−0.0660	0.0519
							PRESS = 457.4000

R code

```
# clean memory
rm(list=ls())
# read data
delivery <- read.csv("eg3.1.delivery.csv",h=T)
# obtain sample size
n=dim(delivery)[1]
names(delivery)
# visualize data
pairs (delivery,pch=20)
# build linear regression
model1 <- lm(DeliveryTime ~ NumberofCases+Distance, data=delivery)

# obtain residual
model1$residuals

# obtain SSRes and MSRes, note the MSRes is also our estimate of sigma square.
SSRes=sum((model1$residuals-mean(model1$residuals))^2)
MSRes=SSRes/(n-3)
# obtain standardized residuals
standardized_res=model1$residuals/sqrt(MSRes)

# obtain studentized residuals we first obtain leverage hii
lm.influence(model1)$hat
# obtain studentized residuals
studentized_res=model1$residuals/sqrt(MSRes)/sqrt(1 - lm.influence(model1)$hat)
# a manual way to obtain studentized_res.  We first need to construct X, which is matrix with the first column of 1s.
X=as.matrix(cbind(1,delivery[,c("NumberofCases","Distance")]))
# obtain hat matrix H
H=X%*%solve(t(X)%*%X)%*%t(X)
# obtain leverage
diag(H)
# manually obtain studentized residuals
studentized_res2=model1$residuals/sqrt(MSRes)/sqrt(1 - diag(H))

# PRESS residuals
PRESS_res=model1$residuals/(1 - lm.influence(model1)$hat)

# R student
R_Student=rstudent(model1)
# a manual way to obtain the R student
S2_i=( (n-3)*MSRes-model1$residuals^2/(1 - diag(H)) )/(n-3-1)
R_Student2=model1$residuals/sqrt(S2_i)/sqrt(1 - diag(H))
```

R code

- # plot all residual and leverage
- # partition the canvas into 6 columns.
- par(mfrow=c(1,6))
- plot(model1$fitted.values,model1$residuals,pch=20,ylab="residual",xlab="fitted value")
- abline(h=0,col="grey")
- plot(model1$fitted.values,standardized_res,pch=20,ylab="standardized residual",xlab="fitted value")
- abline(h=0,col="grey")
- plot(model1$fitted.values,studentized_res,pch=20,ylab="studentized residual",xlab="fitted value")
- abline(h=0,col="grey")
- plot(model1$fitted.values,PRESS_res,pch=20,ylab="PRESS residual",xlab="fitted value")
- abline(h=0,col="grey")
- plot(model1$fitted.values,R_Student,pch=20,ylab="R student",xlab="fitted value")
- abline(h=0,col="grey")
- plot(model1$fitted.values,lm.influence(model1)$hat,pch=20,ylab="leverage",xlab= "fitted value")

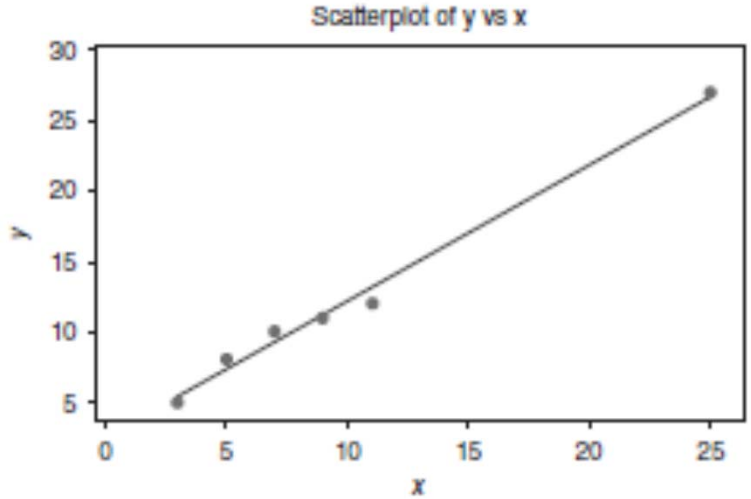


Figure 4.1 Example of a pure leverage point.

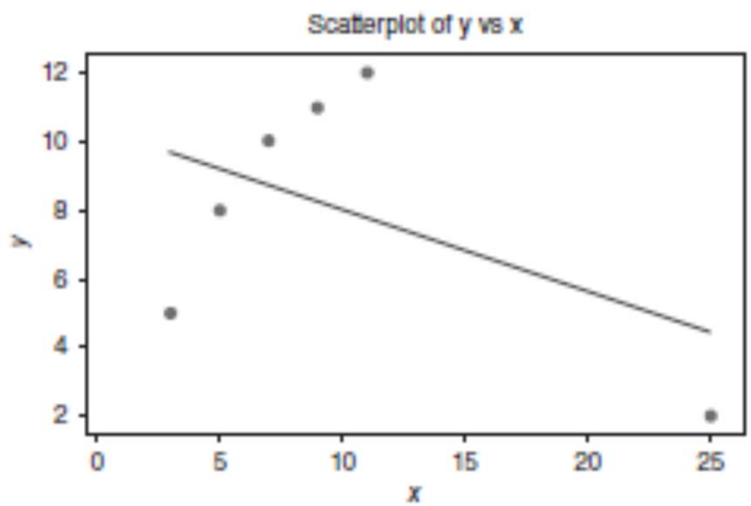


Figure 4.2 Example of an influential point.

Residual Plots

- Normal Probability Plot of Residuals
  - Checks the normality assumption
- Residuals against Fitted values, $\hat{y}_i$
  - Checks for nonconstant variance
  - Checks for nonlinearity
  - Look for potential outliers

Residual Plots

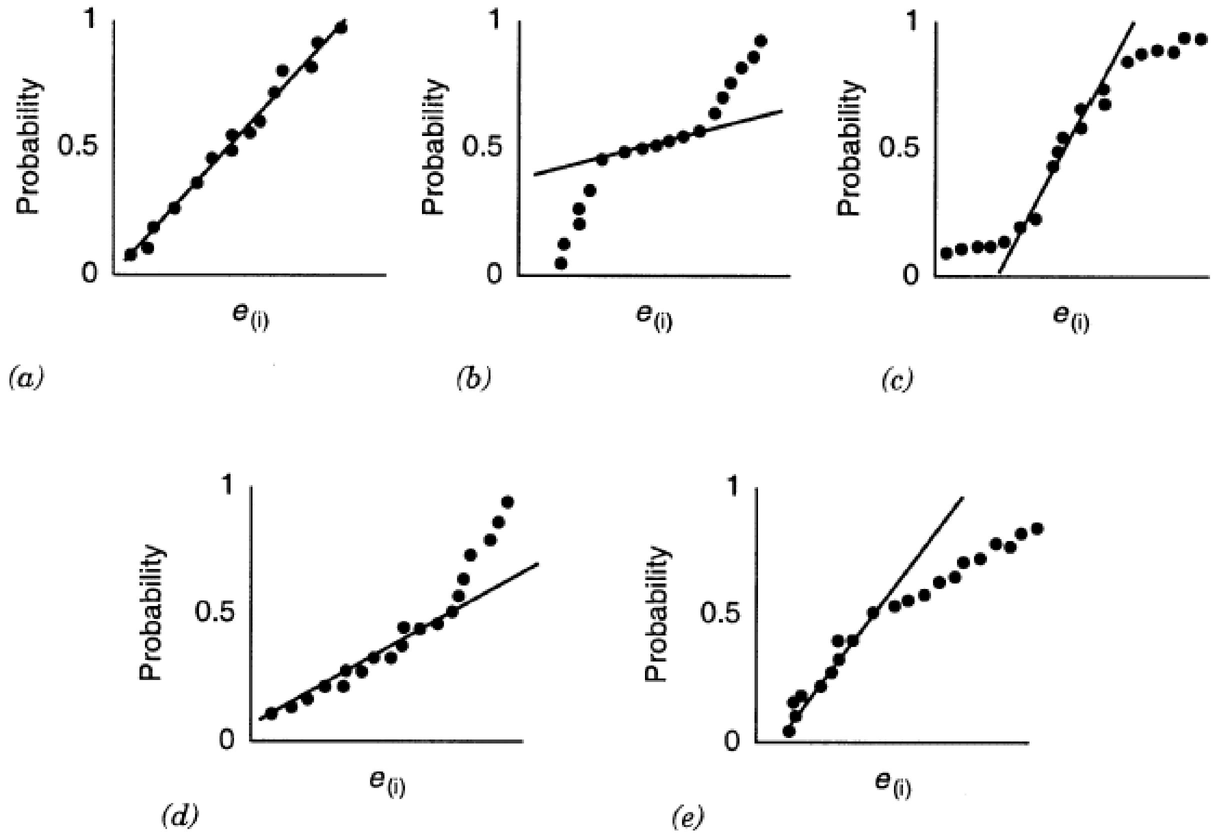


Figure 4.1 Normal probability plots: (a) ideal; (b) heavy-tailed distribution; (c) light-tailed distribution; (d) positive skew; (e) negative skew.

Residual Plots

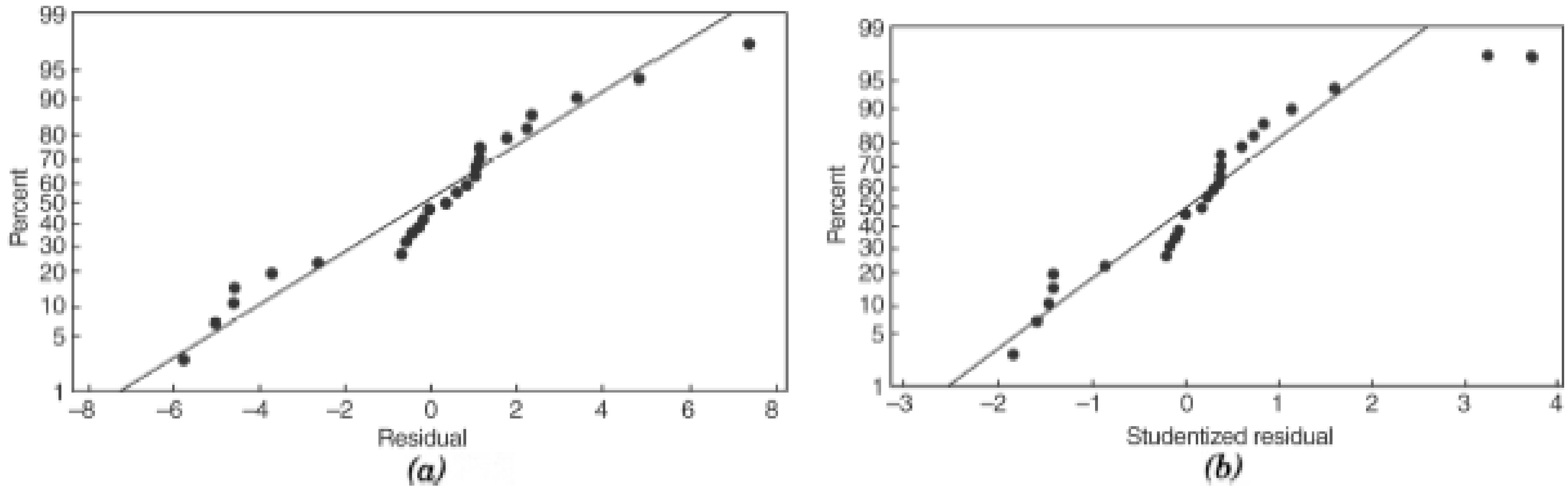


Figure 4.2 Normal probability plots of the residuals for the delivery time data: (*a*) ordinary least-squares residuals; (*b*) studentized residuals.

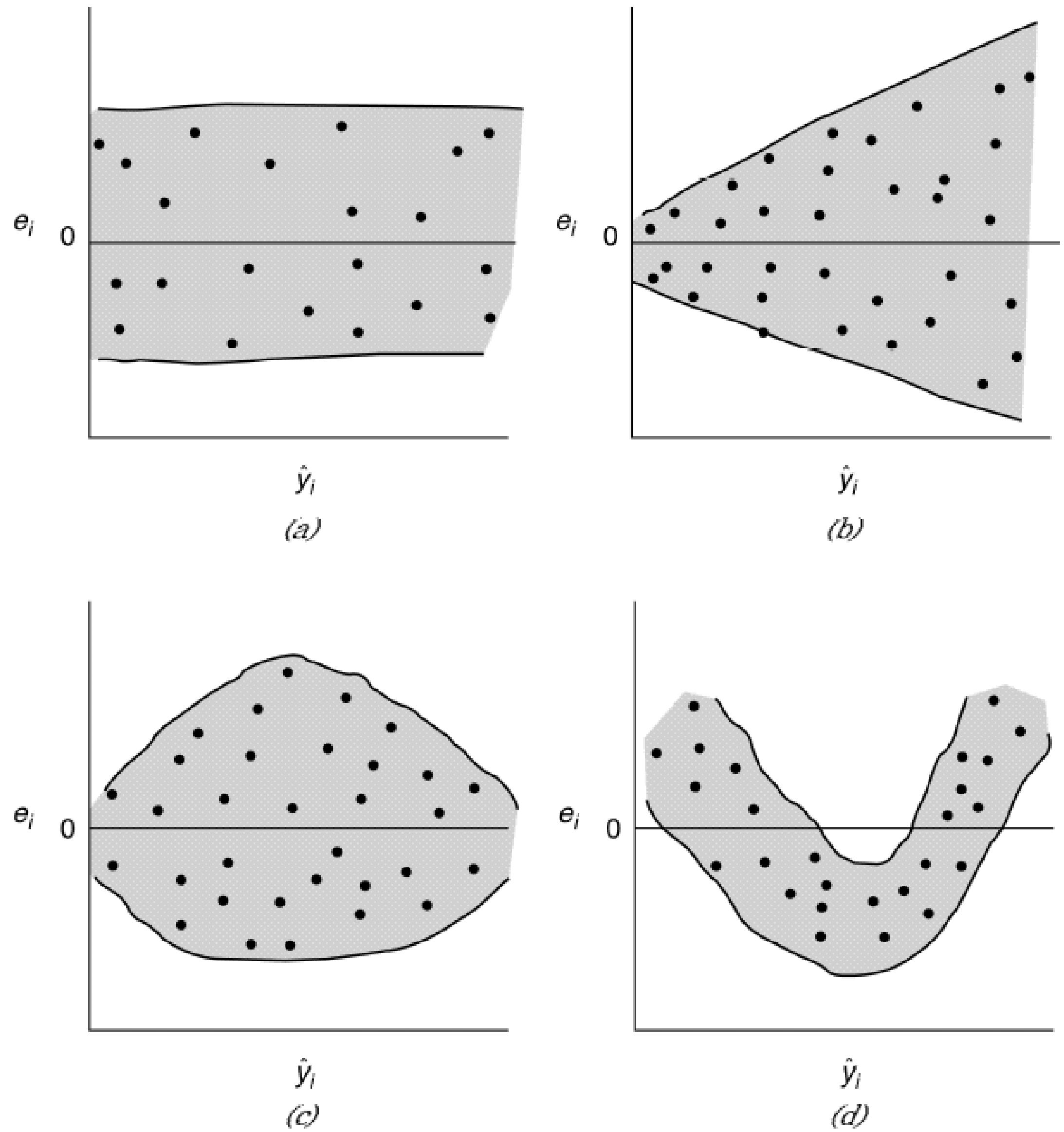
R code

- # generate QQ plot
- qqnorm(model1$residuals)
- qqline(model1$residuals)

Residual Plots

- Residuals against Regressors in the model
  - Checks for nonconstant variance
  - Look for nonlinearity
- Residuals against Regressors *not* in the model
  - If a pattern appears, could indicate that adding that regressor might improve the model fit.
- Residuals against time order
  - Check for Correlated errors

Figure 4.3 Patterns for residual plots: a) satisfactory; b) funnel; c) double bow; d) nonlinear.



R code

- # generate residual plot, NumberofCases vs residuals
- plot(delivery$NumberofCases,model1$residuals)
- # generate residual plot, Distance vs residuals
- plot(delivery$Distance,model1$residuals)
- # generate residual plot, fitted values vs residual
- plot(model1$fitted.values,model1$residuals)

Example 4.4 The Delivery Time Data

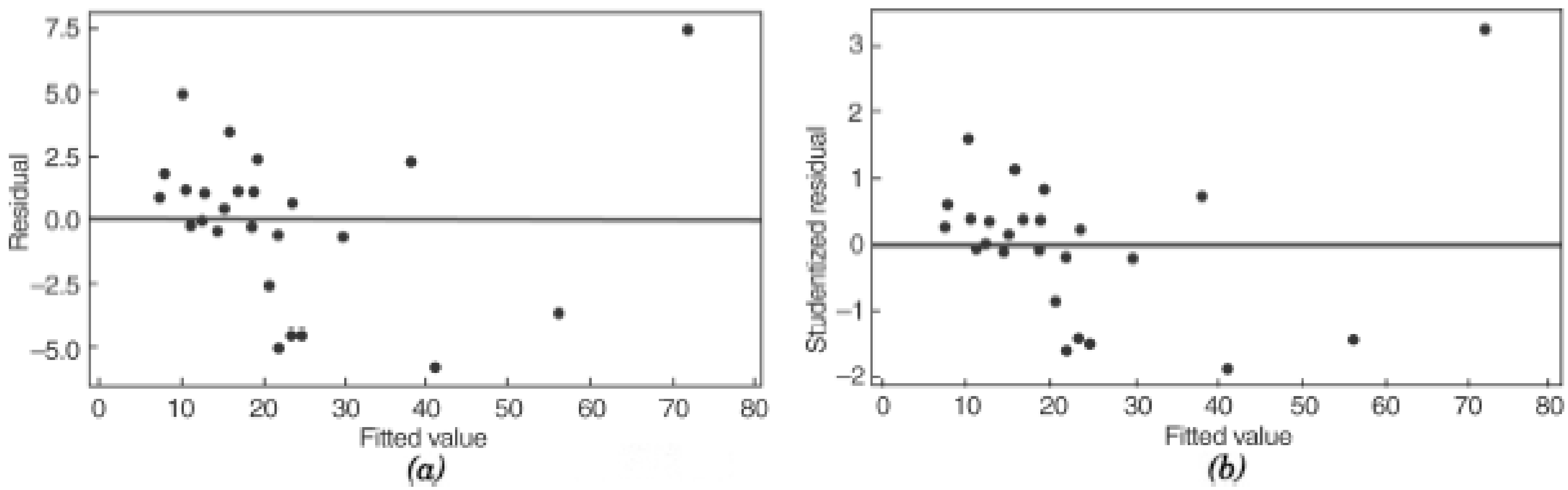


Figure 4.4 Plot of residuals versus predicted for the delivery time data: (*a*) original residuals; (*b*) studentized residuals.

Example 4.4 The Delivery Time Data

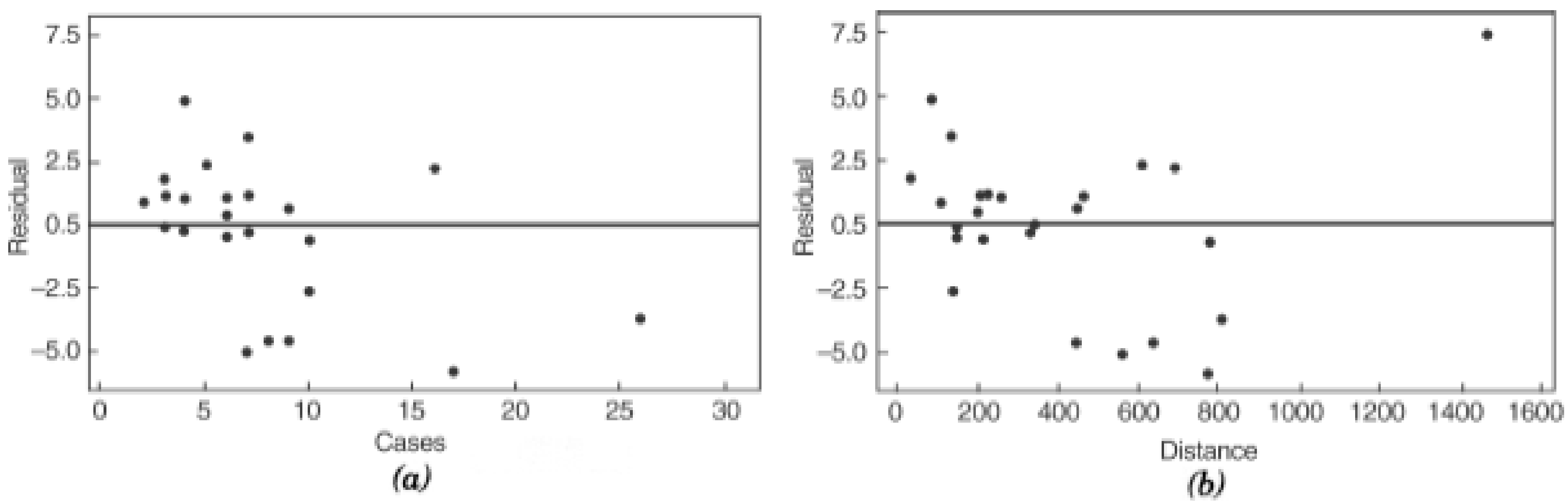


Figure 4.5 Plot of residuals versus the regressors for the delivery time data: (*a*) residuals versus cases; (*b*) residuals versus distance.

Plot of Residuals in Time Sequence

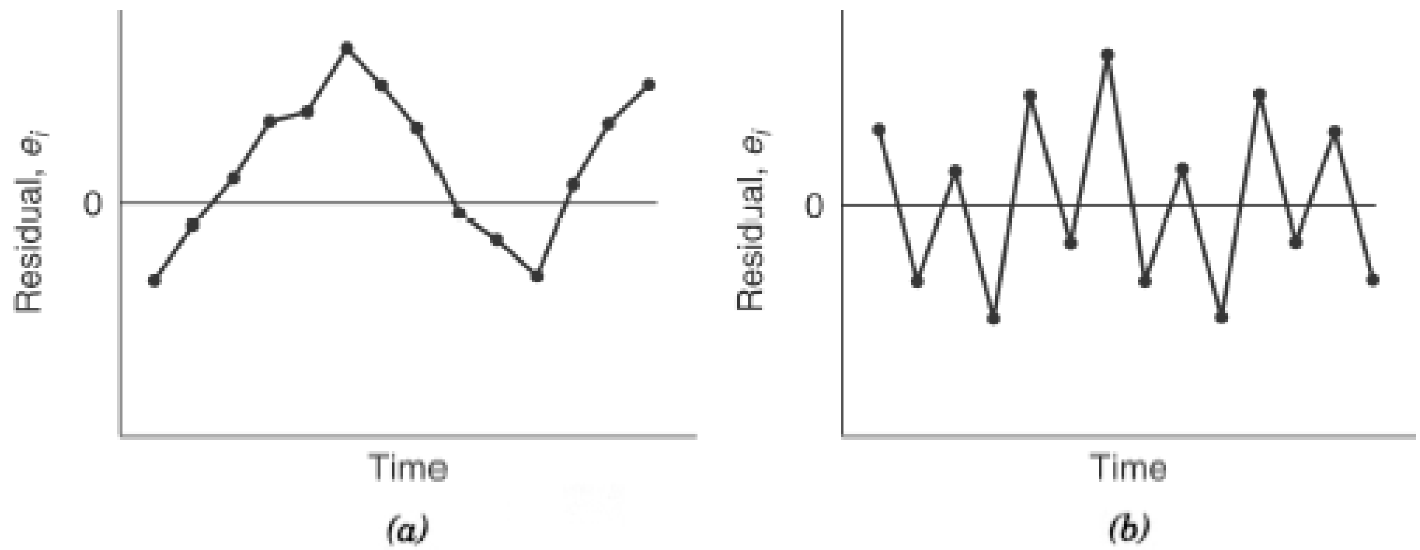


Figure 4.6 Prototype residual plots against time displaying autocorrelation in the errors: (*a*) positive autocorrelation; (*b*) negative autocorrelation.

R code (rocket propellant example)

```
# example
rm(list=ls())
rocket <- read.delim("Data-ex-2-1 (Rocket Prop).txt",h=T)
n=dim(rocket)[1]
plot(rocket$x,rocket$y,pch=20)
model1 <- lm(y ~ x, data=rocket)
abline(model1,col="blue")
# residual
model1$residuals
#standardized residuals
SSRes=sum((model1$residuals-mean(model1$residuals))^2)
MSRes=SSRes/(n-3)
standardized_res=model1$residuals/sqrt(MSRes)
# studentized residuals
studentized_res=model1$residuals/sqrt(MSRes)/sqrt(1 - lm.influence(model1)$hat)
# PRESS residuals
PRESS_res=model1$residuals/(1 - lm.influence(model1)$hat)
# R student
R_Student=rstudent(model1)

# plot all residual and leverage
# partition the canvas into 6 columns.
par(mfrow=c(1,6))
plot(model1$fitted.values,model1$residuals,pch=20,ylab="residual",xlab="fitted value")
abline(h=0,col="grey")
plot(model1$fitted.values,standardized_res,pch=20,ylab="standardized residual",xlab="fitted value")
abline(h=0,col="grey")
plot(model1$fitted.values,studentized_res,pch=20,ylab="studentized residual",xlab="fitted value")
abline(h=0,col="grey")
plot(model1$fitted.values,PRESS_res,pch=20,ylab="PRESS residual",xlab="fitted value")
abline(h=0,col="grey")
plot(model1$fitted.values,R_Student,pch=20,ylab="R student",xlab="fitted value")
abline(h=0,col="grey")
plot(model1$fitted.values,lm.influence(model1)$hat,pch=20,ylab="leverage",xlab="fitted value")
```

Partial Regression and Partial Residual Plots

- Partial Regression Plots

  Purpose:

  - To determine if the correct relationship between y and x_i has been identified
  - To determine the marginal contribution of a variable, given all other variables are in the model.

Partial Regression and Partial Residual Plots

- Partial Regression Plots

  Method:
  - Regress y against all variables except x_i and calculate residuals
  - Regress x_i against all other regressor variables and calculate residuals
  - Plot these two sets of residuals against each other.

Partial Regression and Partial Residual Plots

- Partial Regression Plots

  Interpretation:
  - If the plot appears to be linear, then a linear relationship between y and x_i seems reasonable
  - If plot is curvilinear, may need x_i^2 or $1/x_i$ instead
  - If x_i is a **candidate variable**, and a horizontal "band" appears, then that variable adds no new information.

Example 4.5

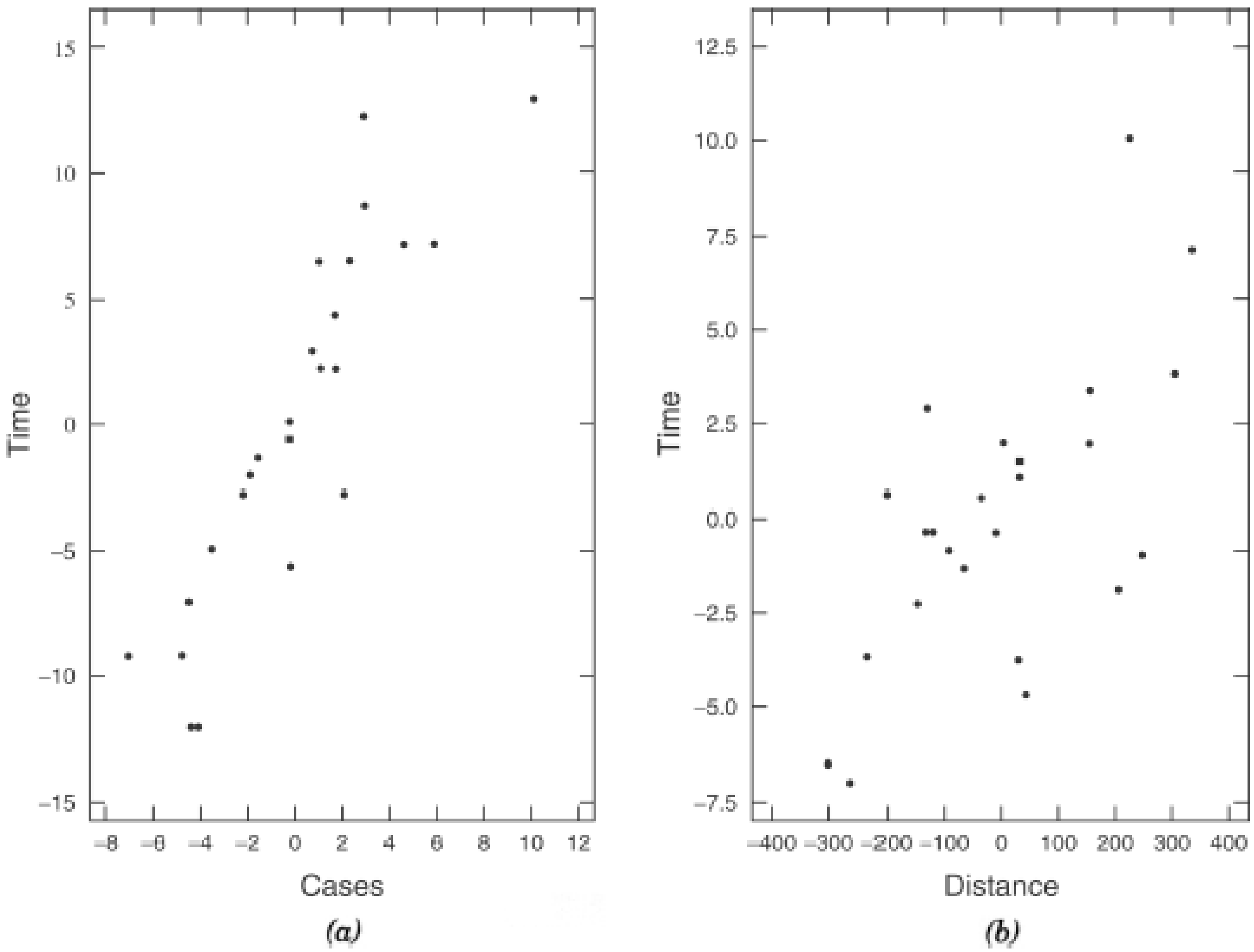
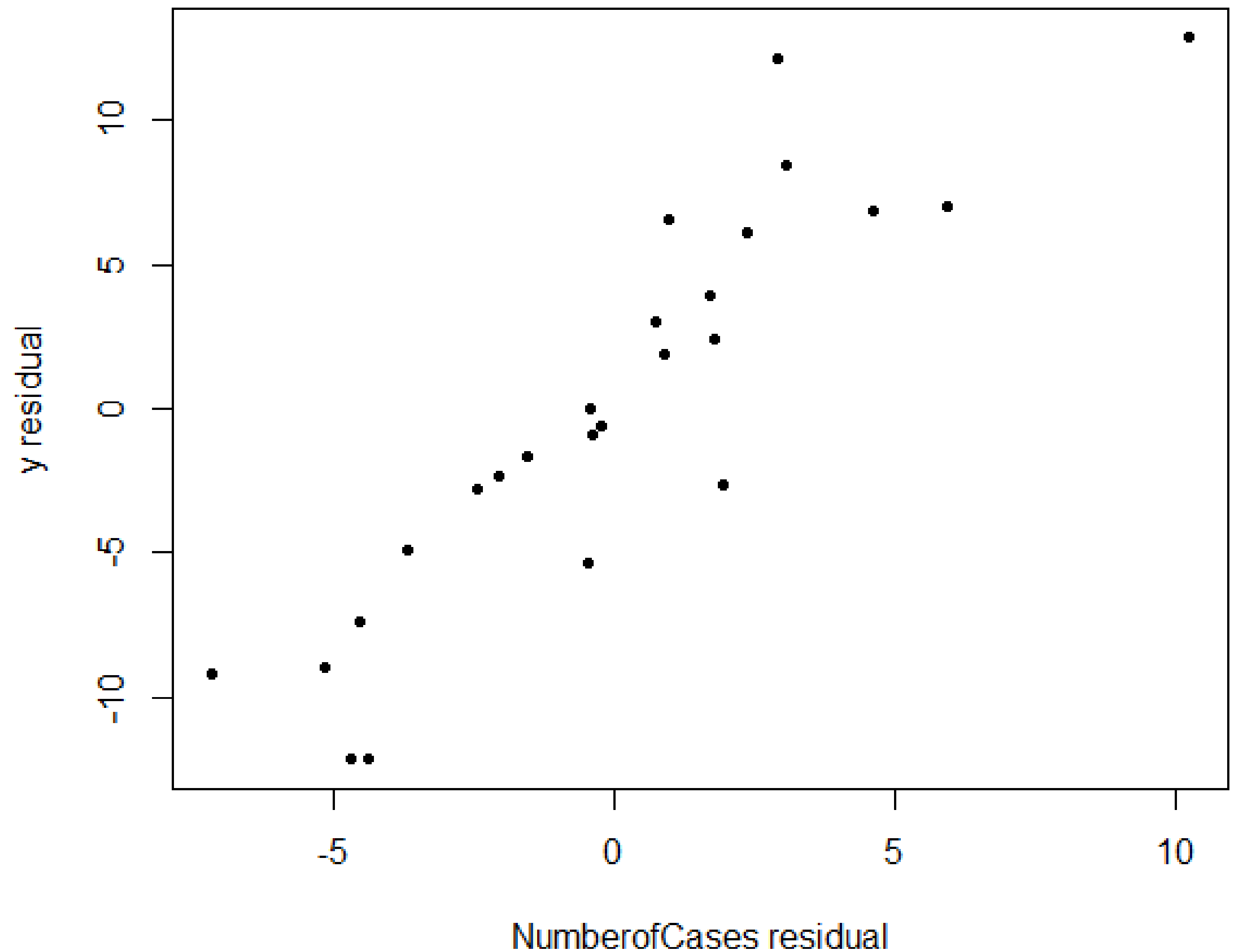


Figure 4.7 Partial regression plots for the delivery time data.

R code

- # generate partial regression plot
- # suppose we first have one regression with only Distance as covariate.
- model2 <- lm(DeliveryTime ~ Distance, data=delivery)
- # now we are considering if we want to add a new covariate NumberofCases. So we first regress NumberofCases on Distance and obtain residuals.
- model3 <- lm(NumberofCases ~ Distance, data=delivery)
- # we plot the residual from model3 against the residuals from model2 and see if there is a linear relationship. If yes, then we should include this new variable NumberofCases.
- plot(model3$residuals,model2$residuals,pch=20,ylab="y residual", xlab="NumberofCases residual")

Partial Regression Plot



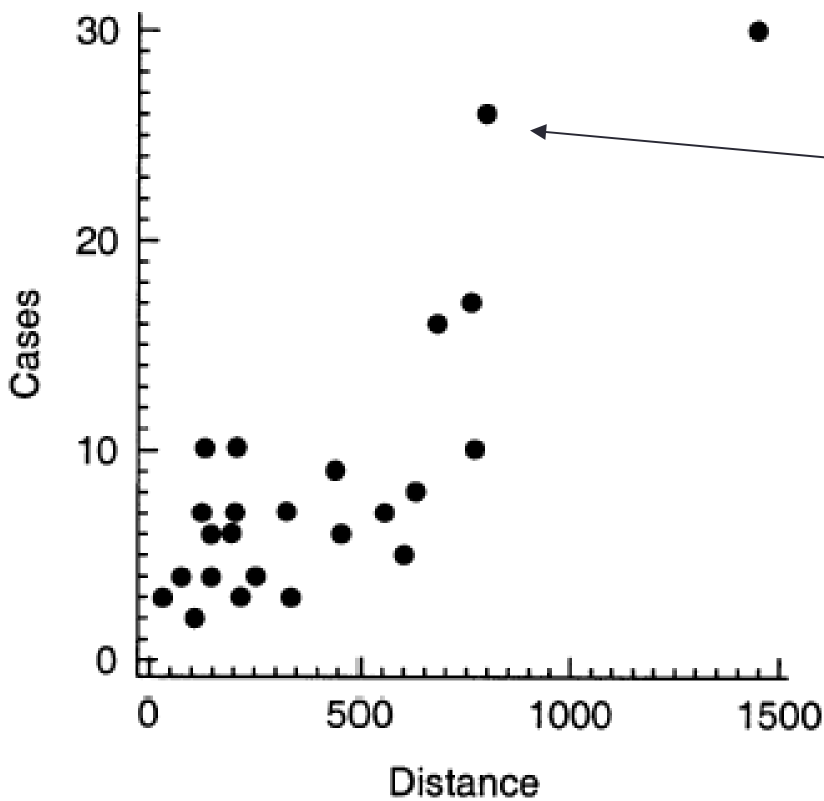
Partial Regression and Partial Residual Plots

- Partial Regression Plots

  Comments:
  - Use with caution, they only suggest possible relationships
  - Do not generally detect interaction effects
  - If multicollinearity is present, regression plots could give incorrect information
  - The slope of the partial regression plot is the regression coefficient for the variable of interest

Other Residual Plotting and Analysis Methods

- Plotting regressors against each other can give information about the relationship between the two:
  - may indicate correlation between the regressors.
  - may uncover remote points



Note location of these two point in the x - space

Figure 4.9 Plot of regressor x_1 (cases) versus regressor x_2 (distance) for the delivery time data in Table 3.2.

The PRESS Statistic

- PRESS Residual:
  - Prediction Error Sum of Squares (PRESS) Statistic:

$$e_{(i)} = y_i - \hat{y}_{(i)}$$

$$PRESS = \sum \left(y_i - \hat{y}_{(i)}\right)^2$$
$$= \sum \left(\frac{e_i}{1 - h_{ii}}\right)^2$$

  - A small value of the PRESS Statistic is desired.

The PRESS Statistic

- R^2 for Prediction Based on PRESS

$$R^2_{prediction} = 1 - \frac{PRESS}{SS_T}$$

- Interpretation:
  - We expect the model to explain about R^2% of the variability in prediction of a new observation.
- PRESS is a valuable statistic for comparison of models.

R code

- # To obtain R_square_prediction, we first calculate PRESS (prediction error sum of square)
- PRESS=sum(PRESS_res^2)
- # then obtain SST
- SST=sum((delivery$DeliveryTime-mean(delivery$DeliveryTime))^2)
- # finally, we obtain R square prediction
- R_square_pred=1-PRESS/SST

Outliers

- An outlier is an observation that is considerably different from the others
- Formal tests for outliers
- Points with large residuals may be outliers
- Impact can be assessed by removing the points and refitting