Chapter 3: Models for Experimental Outcomes

Introductory Statistics for Engineering Experimentation

Peter R. Nelson, Marie Coffin and Karen A.F. Copeland

Slides by Douglas Bates

Outline

Outline

Outline

Outline

Outline

Mathematical models in science and engineering

- Mathematical models are widely used in science and engineering to express the relationships among several *variables* (quantities that we can measure).
- Typically these models include *parameters*, which are constants related to the process. In some mathematical models the values of the parameters are known. We will consider models with parameters whose values are unknown and must be estimated from the data.
- See the description preceding §3.1 of model used to extrapolate the shelf-life of a compound based on the results of an accelerated life-test experiment. The model uses the Arrhenius relationship from chemical kinetics.

Section 3.1: Single-factor experiments

- In a *single-factor* experiment we measure a numeric *response variable* several times at each of the levels of a categorical *covariate*. The `dhaze` data are an example.
- If the number of measurements at each level of the covariate is constant, we say that the experiment is *balanced*.
- Typically we are interested in the mean response for each group. In the model we write the mean response for population group i as μ_i. The model for the jth measurement in the ith group in a balanced experiment is

$$\mathcal{Y}_{ij} = \mu_i + \epsilon_{ij}, \quad i = 1, \ldots, I;\ j = 1, \ldots, n$$

where

$\mathcal{Y}_{ij}$ is the jth replicate measurement at the ith level
μ_i is the mean response at the ith level
ϵ_{ij} is the individual random error for this observation

Some notation

- We use upper-case Latin letters to designate the value of a response in the model. On the slides these are shown in the script font, like $\mathcal{Y}_{ij}$.
- The observed values are shown as the corresponding lower-case letter, like y_{ij}.
- We use Greek letters for parameters, like μ_i.
- In a single-factor experiment we write the number of levels of the factor as I so the subscript $i = 1, \ldots, I$.
- In a balanced experiment we can write the number of replicate observations in each group as n. For an unbalanced experiment we need to write the number of replicates in the ith group as n_i.

Parameter estimation

- The parameters to be estimated are $\mu_i, i = 1, \ldots, I$ (and a measure of the variability in the ϵ_{ij}).
- Our estimate of μ_i, written

$$\widehat{\mu}_i = \bar{y}_{i\bullet} = \sum_{j=1}^{n} y_{ij}, \quad i = 1, \ldots, I,$$

  is the ith sample mean.
- In general a "hat" over a parameter symbol denotes the estimate of the parameter. A "bar" over a letter indicates an average. We replace the subscript(s) over which we have averaged by a dot.

Estimating the Magnitude of the Error

- The "noise-terms", ϵ_{ij}, in the model characterize the random variability in the measurements.
- We sometimes refer to these as the "error" in that they represent the amount by which the measurement $\mathcal{Y}_{ij}$ deviates from its mean or "expected" value, μ_i. The term "error" should not be interpreted as meaning that something has gone wrong — it simply means that there is random or unexplained variability in the process.
- We characterize the magnitude of the error terms by their standard deviation or, equivalently, their variance. The (theoretical) variance of error in the ith group is written σ_i^2 with estimate

$$\widehat{\sigma^2}_i = s_i^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(y_{ij} - \bar{y}_{i\bullet}\right)^2 .$$

Pooling estimates of error variances; residuals

- If we can reasonably assume that $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_I^2 = \sigma^2$ then we "pool" the estimates from the individual groups as

$$\widehat{\sigma^2} = \frac{s_1^2 + s_2^2 + \cdots + s_I^2}{I}$$

- When estimates of parameters are available, we derive estimates of the noise terms. These estimates are called the *residuals* (in the sense of "the part that is left over after we formulate our best guess"). For this model

$$\widehat{e}_{ij} = y_{ij} - \widehat{\mu}_i = y_{ij} - \bar{y}_{i\bullet},$$

from which we can write

$$\widehat{\sigma^2} = \frac{1}{I(n-1)} \sum_{i=1}^{I} \sum_{j=1}^{n} (\widehat{e}_{ij})^2$$

Expressions like that on the right are called "sums of squared residuals" or "residual sum of squares".

Fitting such models in *R*

- The calculation of $\widehat{\mu}_i, i = 1, \ldots, I$ and $\widehat{\sigma^2}$ is straightforward and could be done with a calculator.
- Instead of showing the individual calculations in *R*, we show the general method of fitting models like this using the `aov` function (these models are sometimes called "analysis of variance" models after one of the statistical techniques applied to the results).
- As in the lattice graphics functions, the first argument to `aov` is a formula. In this case it is a two-sided formula with the response on the left and the covariate(s) on the right.
- We assign the fitted model object to a name and apply various *extractor* functions to it. The assignment operator is the two-character sequence $<-$ (looks like a left-pointing arrow). The `=` can also be used.

Optical lens data, Example 3.1.2

```
> fm1 <- aov(dhaze ~ treatment, dhaze)
> summary(fm1)

            Df Sum Sq Mean Sq F value    Pr(>F)
treatment    3 51.067  17.022  10.124 0.0001696
Residuals   24 40.352   1.681

> model.tables(fm1, type = "means")

Tables of means
Grand mean
11.075

 treatment
treatment
     A      B      C      D
 9.447 12.611 12.189 10.053

> str(resid(fm1))

 Named num [1:28] -0.927 -0.237 1.003 0.783 -0.697 ...
 - attr(*, "names")= chr [1:28] "1" "2" "3" "4" ...
```

Factorial experiments

- In a factorial experiment each level of every factor occurs in combination with each level of every other factor.
- If the number of times each combination occurs is constant, we say it is a balanced factorial experiment.
- If combinations of factor levels occur more than once, we say it is a replicated factorial experiment.
- For a balanced, two-factor factorial with replications we write the responses as $y_{ijk}, i = 1, \ldots, I; j = 1, \ldots, J; k = 1, \ldots, n$ (if it is unbalanced then $k = 1, \ldots, n_{ij}$).
- "Cell means" and "cell variances" (the names come from considering cells in a two-way table, like tables 3.1 and 3.2) are written $\bar{y}_{ij\bullet}$ and s^2_{ij}. Row and column means are written $\bar{y}_{i\bullet\bullet}, i = 1, \ldots, I$ and $\bar{y}_{\bullet j\bullet}, j = 1, \ldots, J$. The "grand mean" is $\bar{y}_{\bullet\bullet\bullet}$

A Model with No Interaction

- With multiple factors we write the model in terms of "effects" of the levels of each factor (written α_i and β_j) and possible interactions.
- An "additive" model, meaning one without interactions, is

$$\mathcal{Y}_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk}, \quad i = 1, \ldots, I;\ j = 1, \ldots, J;\ k = 1, \ldots, n$$

- Parameter estimates from a balanced two-factor factorial are based on averages; $\widehat{\mu} = \bar{y}_{\cdots}$, $\widehat{\alpha}_i = \bar{y}_{i\cdot\cdot} - \bar{y}_{\cdots}$, etc.
- Things get much more complicated with unbalanced data. I tend to use the computer, even for balanced designs, so that I can also plot the data to check assumptions.

Paint formulation data, Example 3.2.2

- In the text the factors are written so that component 2 is the first factor (factor A) and component 1 is the second (factor B). We list them in that order in the formula so our results are consistent with those in the text.

```
> fm2 <- aov(lw ~ comp2 + comp1, lw)
```

- The "Mean square for residuals" in the summary table is $\widehat{\sigma^2} = s^2$

```
> summary(fm2)
```

```
            Df  Sum Sq Mean Sq F value    Pr(>F)
comp2        2 104.553  52.276  37.485 3.823e-07
comp1        3  47.791  15.930  11.423 0.0001993
Residuals   18  25.103   1.395
```

Estimated effects and residuals from example 3.2.2

```
> model.tables(fm2)

Tables of effects
 comp2
comp2
      A       B       C
-0.0125 -2.5500  2.5625

 comp1
comp1
      A       B       C       D
 1.4458 -1.9375  1.2458 -0.7542

> str(resid(fm2))

 Named num [1:24] 0.979 -0.721 -0.383 1.417 -0.496 ...
 - attr(*, "names")= chr [1:24] "1" "2" "3" "4" ...
```

A Model Accounting for Interaction

- As described in the text, a two-factor model allowing for interactions of the factors provides a separately estimated mean for each cell. That is

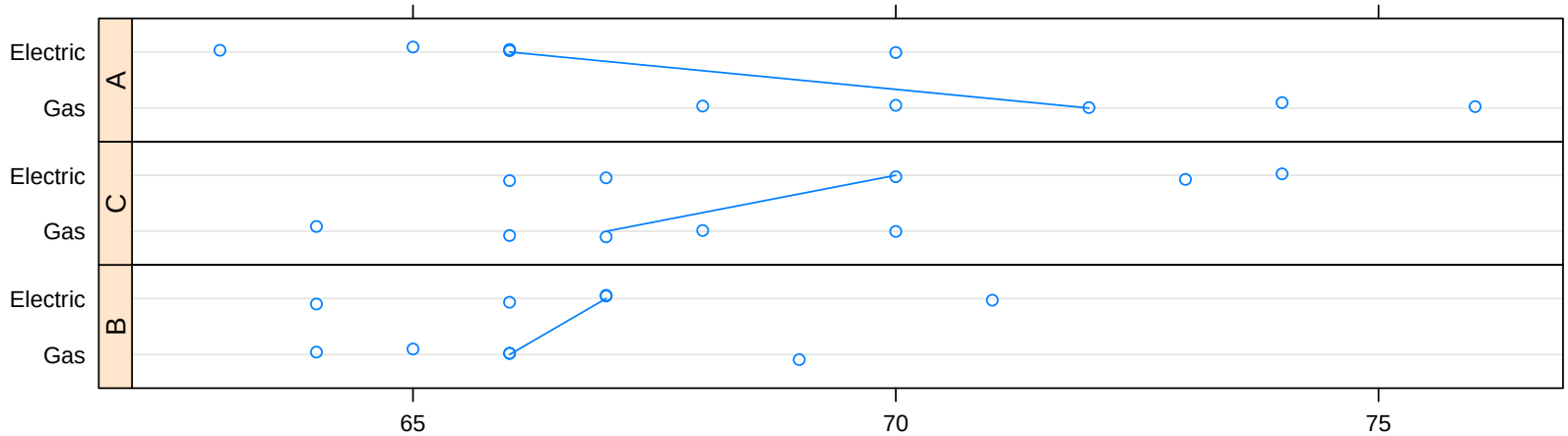
$$\mathcal{Y}_{ijk} = \mu_{ij} + \epsilon_{ijk}, \quad i = 1, \ldots, I;\ j = 1, \ldots, J;\ k = 1, \ldots, n$$

- The estimates of the cell mean parameters are the cell sample means.

$$\widehat{\mu}_{ij} = \bar{y}_{ij\bullet}$$

- As before the estimate of σ^2 is the mean squared residual.
- In the formula for the `aov` function we use an `*` instead of a `+` between the factors to indicate a model with interactions.

Oven data, example 3.2.4




```
> summary(fm3 <- aov(moisture ~ type * brand, oven))

             Df  Sum Sq Mean Sq F value   Pr(>F)
type          1   3.333   3.333  0.4545 0.506627
brand         2  35.000  17.500  2.3864 0.113439
type:brand    2 111.667  55.833  7.6136 0.002751
Residuals    24 176.000   7.333

> str(fitted(fm3))

 Named num [1:30] 72 72 72 72 72 66 66 66 66 66 ...
 - attr(*, "names")= chr [1:30] "1" "2" "3" "4" ...
```

Bivariate data

- If we have measured two numeric characteristics and can regard one as an *independent* or *predictor* variable while the other is a *dependent* or *response* variable, we fit an appropriate response function.
- For many phenomena observed over a restricted range, it is appropriate to use a linear model of the form

$$\mathcal{Y}_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \ldots, n$$

where

$\mathcal{Y}_i$ is the response on the ith trial
x_i is the value of the covariate on the ith trial
n is the number of observations

- The best way to decide if this is an appropriate model is to **plot the data**. Never fit a statistical model without first plotting the data.

Parameter estimates for linear models

- If a linear model seems appropriate and the variability seems reasonably constant across the range of the data, we use the *least squares* parameter estimates which minimize the sum of squared residuals.
- That is, we determine $\widehat{\beta}_0$ and $\widehat{\beta}_1$ as

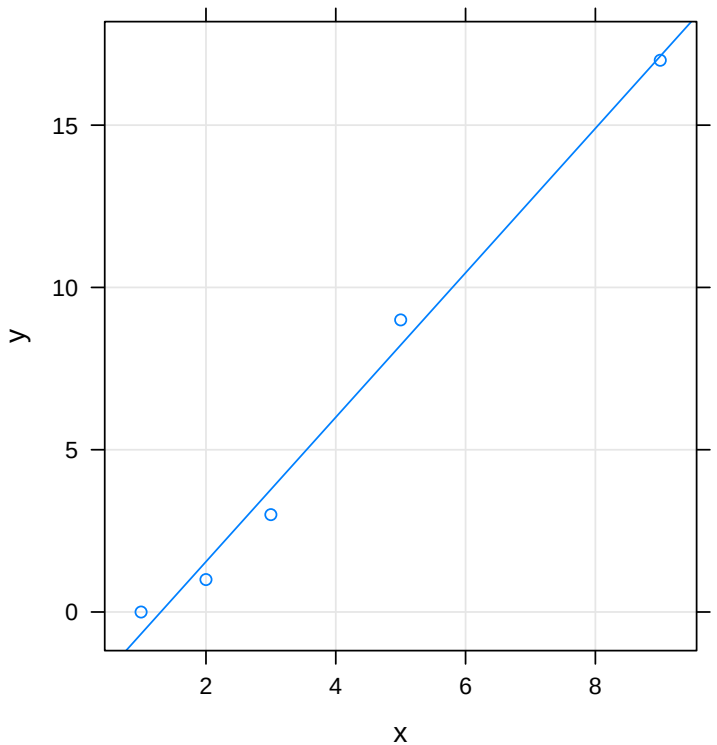
$$\arg\min_{b_0,b_1} \sum_{i=1}^{n} [y_i - (b_0 + b_1 x_i)]^2$$

- As before, the estimate of the variance of the ϵ_i is the mean squared residual.
- We use the `lm` function in *R* to fit such models. As for `aov` the first two arguments are the formula and the name of the data set.

Example 3.3.1

This example uses a toy data set to show the calculations

```
> ex331 <- data.frame(x = c(2,9,3,5,1), y = c(1,17,3,9,0))
```



Example 3.3.1 (cont'd)

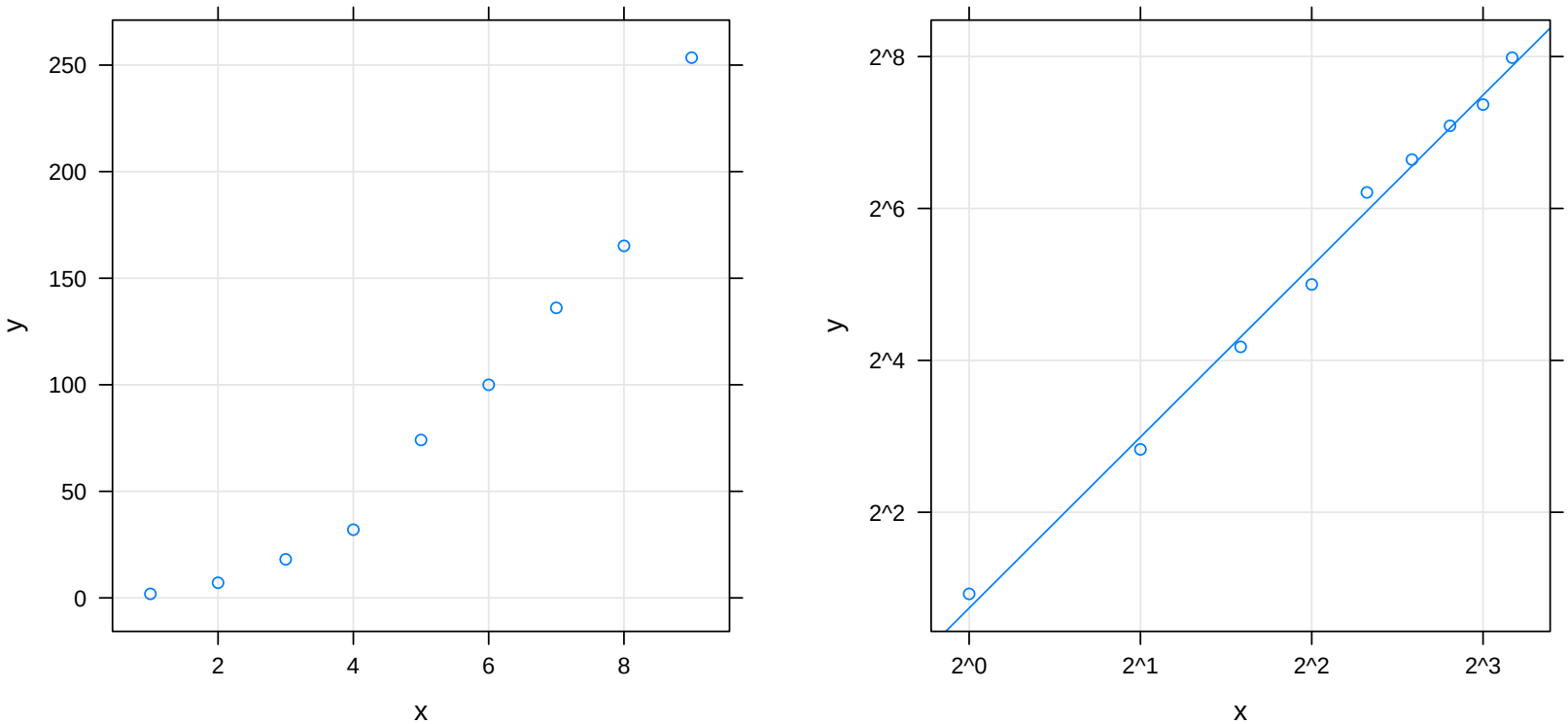
```
> summary(fm4 <- lm(y ~ x, ex331))

Call:
lm(formula = y ~ x, data = ex331)
Residuals:
     1      2      3      4      5
-0.550 -0.125 -0.775  0.775  0.675

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.9000     0.6285  -4.614 0.019152
x             2.2250     0.1283  17.344 0.000418

Residual standard error: 0.8114 on 3 degrees of freedom
Multiple R-squared: 0.9901, Adjusted R-squared: 0.9868
F-statistic: 300.8 on 1 and 3 DF,  p-value: 0.0004177
```

The quantity labeled "Residual standard error" is s, the square root of the variance estimate.

Fitting exponential curves

- There are several forms of models used in engineering that can be converted to a linear model by taking logarithms of the response or of the covariate or both.
- Naturally the variable to be transformed by taking the logarithm must take on positive values only.
- Always check the plot of the transformed data to ensure that it exhibits a linear relationship and approximately constant variability after transformation.

Example 3.3.3



There is noticeable curvature in the original plot, which is dramatically reduced in the log-log plot.

Example 3.3.3 (cont'd)

```
> summary(fm5 <- lm(log(y) ~ log(x), ex333))

Call:
lm(formula = log(y) ~ log(x), data = ex333)
Residuals:
     Min       1Q   Median       3Q      Max
-0.16738 -0.08958  0.02047  0.07669  0.16996

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.51231    0.09783   5.237  0.00120
log(x)       2.25118    0.06208  36.260 3.15e-09

Residual standard error: 0.1263 on 7 degrees of freedom
Multiple R-squared: 0.9947, Adjusted R-squared: 0.9939
F-statistic:  1315 on 1 and 7 DF,  p-value: 3.152e-09
```

Fitting polynomial curves

- A polynomial curve is, strangely enough, regarded as a linear model in statistics because the coefficients in the model, which are the parameters to be estimated, occur linearly.
- That is, we can fit curves of the form

$$\mathcal{Y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

or

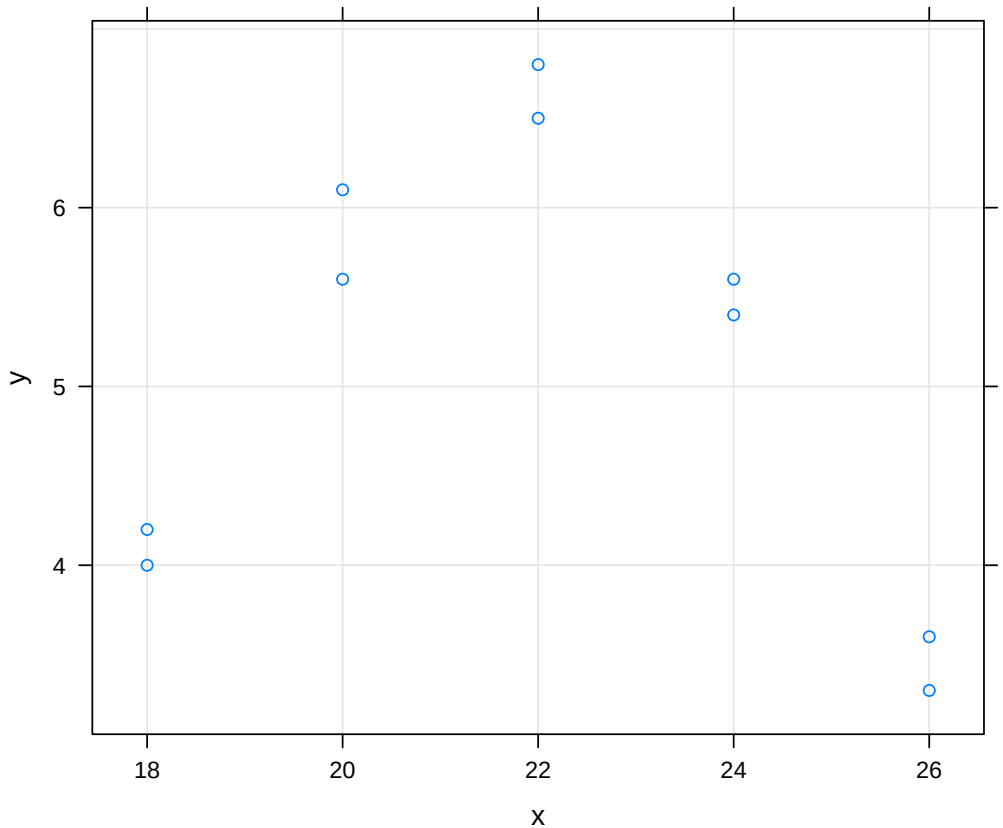
$$\mathcal{Y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \epsilon$$

with the `lm` function in *R*.

- Generally it is not a good idea to go beyond a cubic polynomial. Predictions from higher-order polynomials are too sensitive to small perturbations in the data.
- In the formula for such a model we must use the `I` function (an identity operator) to protect the expressions for the powers of `x`.

Example 3.3.6

```
> ex336 <- data.frame(x = c(18,18,20,20,22,22,24,24,26,26),
+                     y = c(4.0,4.2,5.6,6.1,6.5,6.8,5.4,5.6,3.3,3.6))
```



Example 3.3.6 (cont'd)

```
> summary(fm6 <- lm(y ~ x + I(x^2), ex336))

Call:
lm(formula = y ~ x + I(x^2), data = ex336)
Residuals:
       Min          1Q      Median          3Q         Max
-0.3571429 -0.1057143 -0.0007143  0.1382143  0.3257143

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -74.25000    5.24071  -14.17 2.07e-06
x             7.42107    0.48221   15.39 1.18e-06
I(x^2)       -0.17054    0.01094  -15.58 1.08e-06

Residual standard error: 0.2316 on 7 degrees of freedom
Multiple R-squared: 0.9731, Adjusted R-squared: 0.9654
F-statistic: 126.5 on 2 and 7 DF,  p-value: 3.203e-06
```

Models for multivariate data

- In general we can fit linear models (i.e. linear in the parameters, not necessarily linear in the covariate values) with many different types of terms based on either numeric covariates or categorical covariates (factors).
- A model of the form

$$\mathcal{Y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

with linear terms in two or more numeric covariates is called a multiple linear model.

- In example 3.4.2 a model is fit to chemical process yield data

```
> str(yield)

'data.frame': 20 obs. of  3 variables:
 $ temp : num  20.9 21.2 20.8 20.1 20.3 22.7 20.4 22 20.5 21.5 ...
 $ pH   : num  6.8 6.3 6.8 6.4 6.3 6.6 6.4 6.7 6.8 6.5 ...
 $ yield: num  32.5 32.1 32.2 31.6 30.8 33 31.5 32.9 32.4 32.5 ...
```

Example 3.4.2

```
> summary(fm7 <- lm(yield ~ temp + pH, yield))

Call:
lm(formula = yield ~ temp + pH, data = yield)
Residuals:
      Min        1Q    Median        3Q       Max
-0.683608 -0.210688  0.007775  0.239225  0.734784

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.2751     3.5726   3.996 0.000936
temp          0.4723     0.1162   4.065 0.000805
pH            1.2027     0.4729   2.543 0.021007

Residual standard error: 0.4067 on 17 degrees of freedom
Multiple R-squared: 0.6223, Adjusted R-squared: 0.5779
F-statistic: 14.01 on 2 and 17 DF,  p-value: 0.0002543
```

Coefficient of determination

- A common numeric measure of the quality of the fit of a linear model is the R^2 statistic which is the proportion of the variability in the response that has been incorporated into the model.
- This is shown in the summary of an `lm` fit labeled "Multiple R-squared". As a proportion it satisfies $0 \leq R^2 \leq 1$. Larger is better.
- If SS_e is the sum of squared residuals and $\mathrm{SS}_{\text{total}}$ is the total sum of squares

$$\mathrm{SS}_{\text{total}} = \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2$$

then

$$R^2 = \frac{\mathrm{SS}_{\text{total}} - \mathrm{SS}_e}{\mathrm{SS}_{\text{total}}} = 1 - \frac{\mathrm{SS}_e}{\mathrm{SS}_{\text{total}}}$$

Residual plots

- Creating a statistical model should not be regarded as a "one-shot" process. Instead we should consider it as an iterative process where we examine the data and form a preliminary model, fit this model and then re-examine the fit to see if it satisfies the assumptions on the model, changing the model and re-fitting if necessary.
- Graphical methods are best for the preliminary investigation and for the model criticism. When assessing a model fit we are particularly interested in properties of the residuals. We plot the residuals versus the fitted values and versus covariates that are not yet incorporated in the model.
- In the case of a simple linear regression model the plot of the residuals versus the fitted values is equivalent to plotting the residuals versus the covariate.
- The pattern we are seeking is "no pattern". In particular, we want the residuals to lie in what looks like a horizontal band of constant height centered around the zero line.

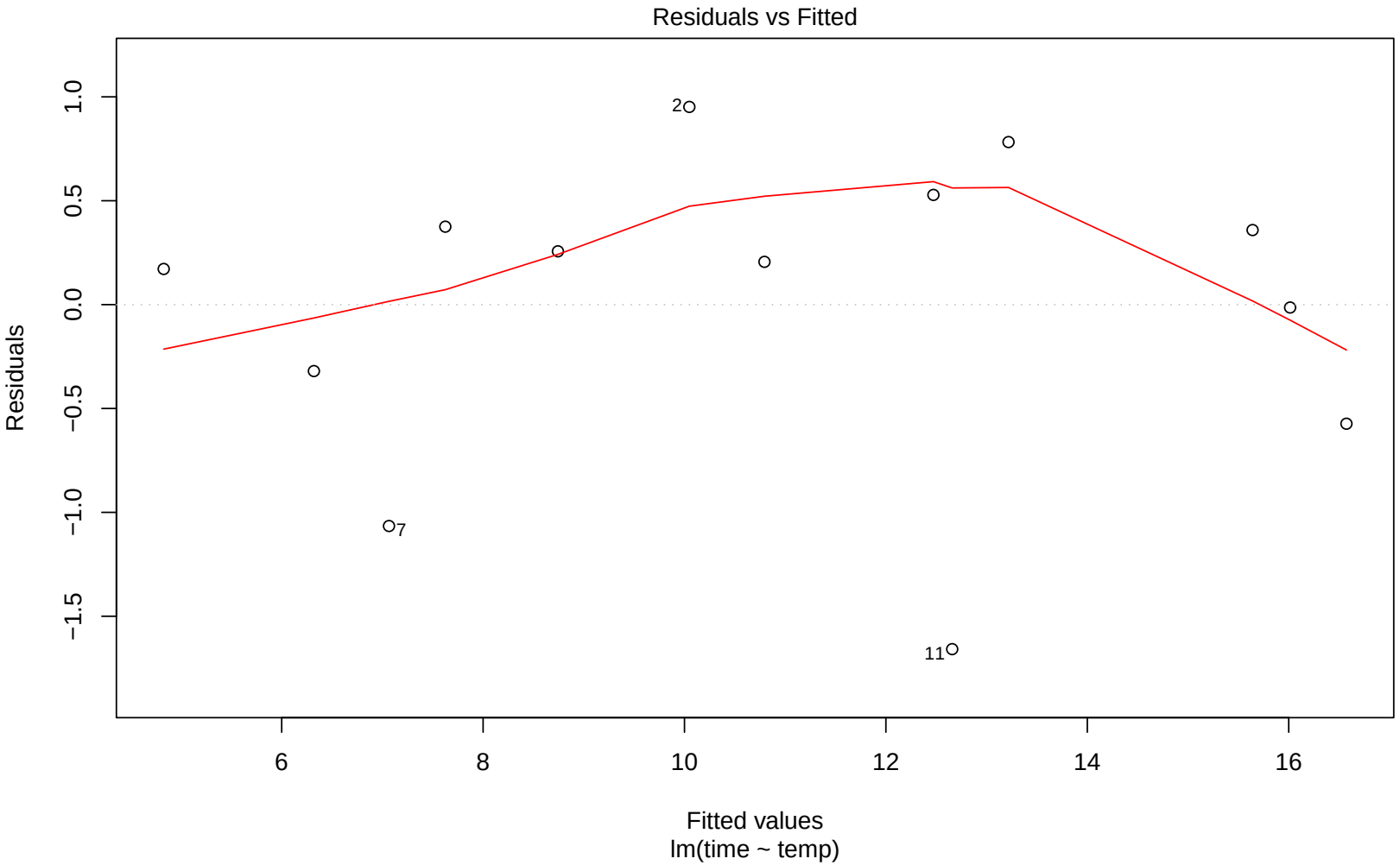
Residual plots in *R*

- Residuals can be plotted by extracting them from the fitted model and using `xyplot`. For the special case of the residuals versus the fitted values a direct call to `plot` can be used.
- In example 3.5.3 the residuals from a simple linear model fit to the repaired panels in the `timetemp` data are plot versus the temperature. We fit the model as

```
> fm8 <- lm(time ~ temp, timetemp, subset = type == "Repaired")
```

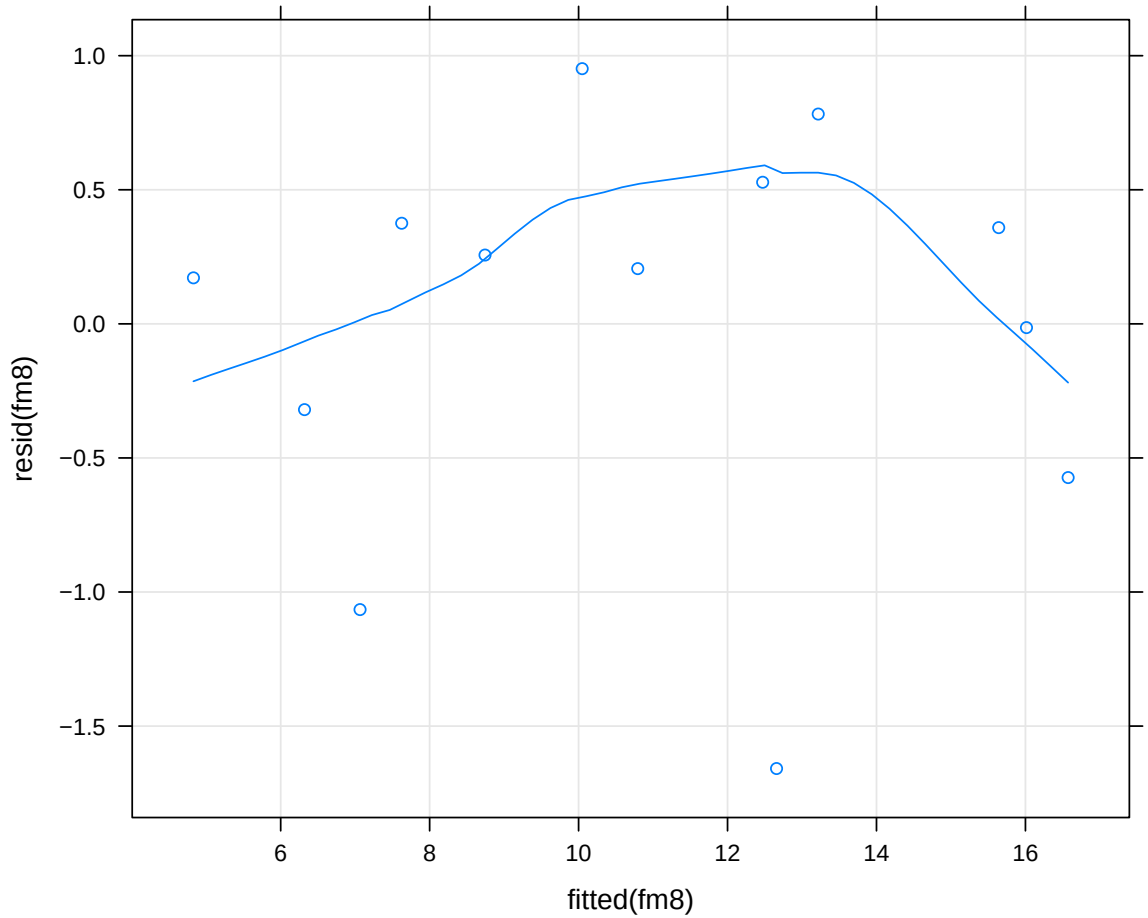
"Pre-packaged" plot of residuals vs. fitted

```
> plot(fm8, which = 1)
```



"Manual" plot of residuals vs. fitted

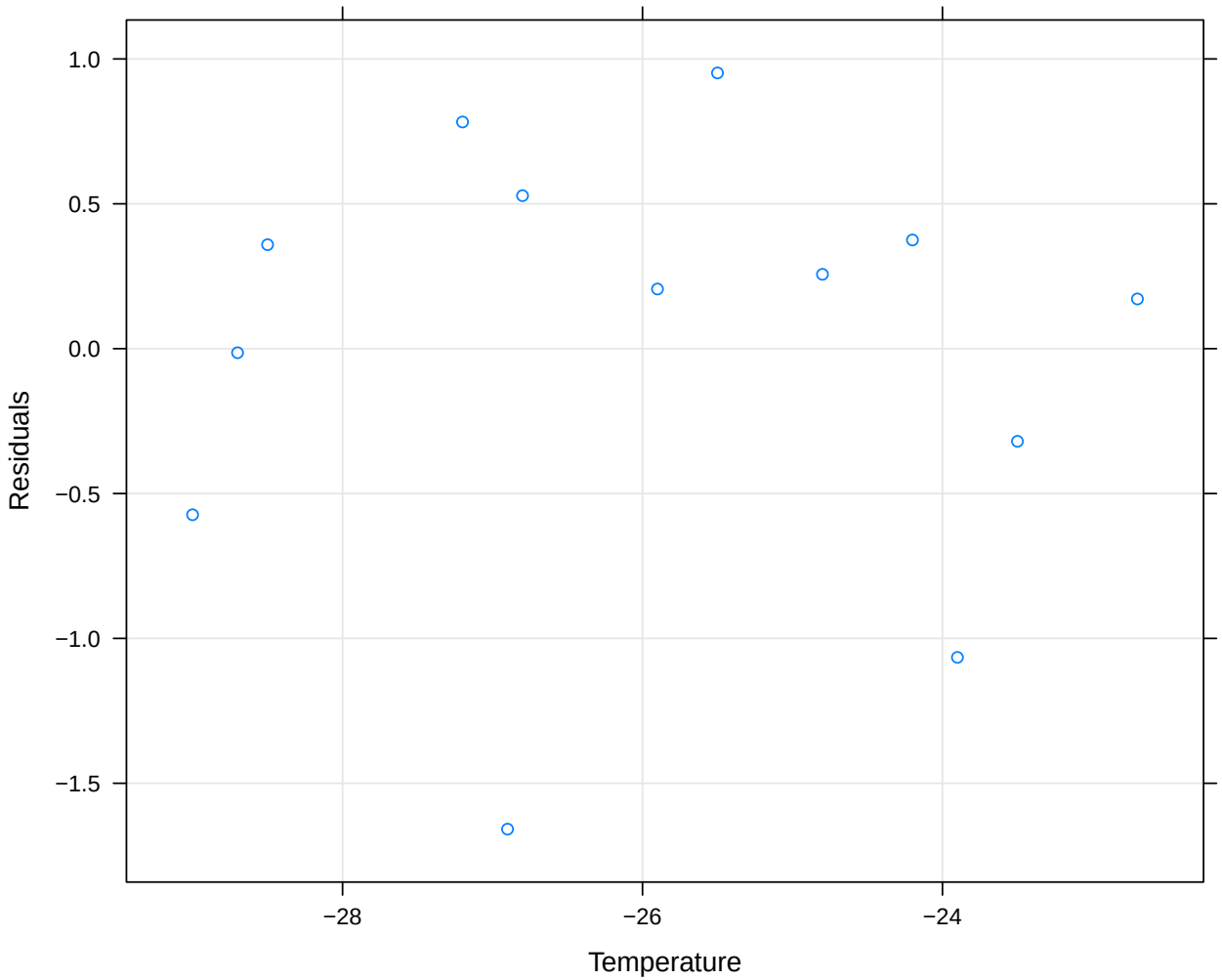
```
> xyplot(resid(fm8) ~ fitted(fm8), type = c("g","p","smooth"))
```



This is a mirror image of Fig. 3.10 because $\widehat{\beta}_1 < 0$.

Reproducing Figure 3.10, page 72

```
> xyplot(resid(fm8) ~ temp, timetemp, subset = type == "Repaired
```



Example 3.5.5

- Example 3.5.5 shows model building for a response (leftover) as a function of two covariates, flow rate and vacuum.
- We will show a slightly different approach. First create the data frame.

```
> shaker <-
+     data.frame(leftover = c(6.3,6.1,5.8,5.9,5.6,5.3,6.1,5.8,5.
+                flowrate = rep(c(85,90,95), 3),
+                vacuum = rep(c(20,22,24), each = 3))
```

Example 3.5.5 (cont'd)

```
> print(xyplot(leftover ~ vacuum, shaker,
+              type = c("g","b"), groups = flowrate,
+              auto.key = list(columns = 3, lines = TRUE)),
+       split = c(1,1,2,1), more = TRUE)
> print(xyplot(leftover ~ flowrate, shaker,
+              type = c("g","b"), groups = vacuum,
+              auto.key = list(columns = 3, lines = TRUE)),
+       split = c(2,1,2,1))
```

Example 3.5.5 (cont'd)

- The plots indicate a quadratic in `vacuum` but a linear term in `flowrate`. There is little evidence of interaction.

```
> summary(fm9 <- lm(leftover ~ flowrate + vacuum + I(vacuum^2),
+                   shaker))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 52.500000   2.704502   19.41 6.69e-06
flowrate    -0.056667   0.002582  -21.95 3.65e-06
vacuum      -3.733333   0.246052  -15.17 2.25e-05
I(vacuum^2)  0.083333   0.005590   14.91 2.46e-05
Residual standard error: 0.03162 on 5 degrees of freedom
Multiple R-squared: 0.9939, Adjusted R-squared: 0.9902
F-statistic: 270.2 on 3 and 5 DF,  p-value: 5.982e-06
```

(Some output has been truncated)

With an R^2 of 99.4% further improvements are unlikely (these are probably constructed data, not an actual experiment).

The correlation coefficient

- For the special case of a simple linear model (i.e. a linear term in only one covariate) the correlation coefficient

$$r = (\text{sign of } \widehat{\beta}_1)\sqrt{R^2}$$

  measures the linear correlation of the response and the covariate.
- Generally correlation measures the extent to which two variables vary together. In observational studies (as opposed to experimental studies) we must be careful not to confuse correlation with causation.
- The *R* function `cor` evaluates r directly.

```
> with(subset(timetemp, type == "Repaired"), cor(time, temp))
[1] -0.9824036
```