Not all relationships are linear. There are two common ways to deal with nonlinear relationships:

1. Transform the data so that there is a linear relationship between the transformed variables.
2. Apply the method of least squares (or maximum likelihood) with a non-linear function.

### 8.1 Transformations in Linear Regression

The utility of linear models is greatly enhanced through the use of various transformations of the data. There are several reasons why one might consider a transformation of the predictor or response (or both).

- To correspond to a theoretical model.

Sometimes we have a priori information that tells us what kind of non-linear relationship we should anticipate. As a simple example, if we have an apparatus that can accelerate objects by applying a constant (but unknown) force $F$, then since $F = ma$, we would expect the relationship between the mass of the objects tested and the acceleration measured to satisfy

$$a = \frac{F}{m}.$$  

This might lead us to fit a model with no intercept (see Exercise 8.3) after applying an inverse transformation to the predictor $m$:

$$a = \beta_1 \cdot \frac{1}{m}.$$  

The parameter $\beta_1$ corresponds to the (unknown) force being applied and could be estimated by fitting this model.

Alternatively, we could use a logarithmic transformation

$$\log(a) = \beta_0 + \beta_1 \log(m),$$  

$$a = e^{\beta_0} m^{\beta_1}.$$  

In this model, $e^{\beta_0}$ corresponds to the unknown force and we can test whether $\beta_1 = -1$ is consistent with our data.

Many non-linear relationships can be transformed to linearity. Exercise 8.3 presents several examples and asks you to determine a suitable transformation.
• To obtain a better fit.

If a scatterplot or residual plot shows a clearly non-linear pattern to the data, then there is no reason to use the linear fit. In the absence of a clear theoretical model, we may select transformations based on the shape of the relationship as revealed in a scatterplot. Section [8.1.3] provides some guidance for selecting transformations in this situation.

• To obtain better residual behavior.

Some transformations are used to improve the agreement between the data and the assumptions about the error terms in the model. For example, if the variance in the response appears to increases as the predictor increases, a logarithmic or square root transformation of the response will decrease the disparity in variance.

In practice, all three of these issues are intertwined. A transformation that improves the fit, for example, may or may not have a good theoretical interpretation. Similarly, a transformation performed to achieve homoskedasticity (equal variance; the opposite is called heteroskedasticity) may result in a fit that does not match the overall shape of the data very well. Despite these potential problems, there are many situations where a relatively simple transformation is all that is needed to greatly improve the model.

### 8.1.1 Three Important “Laws”

In the sciences, relationships between variables based on some scientific theory are often referred to as laws. Many of these fall into one of three categories that are easily handled by transforming the data and fitting a linear regression model.

**Linear Laws**

We’ve already talked about linear relationships, but it is worth mentioning them again because there are so many situations in which a linear relationship arises.

**Power Laws**

Relationships of the form

\[ y = Ax^p \]

are often called power laws. The two parameters are the exponent \( p \) and a constant of proportionality \( A \). Power laws can be linearized by taking logarithms:

\[ \log(y) = \log(Ax^p) = \log(A) + p \log(x) \]

So if we fit a model of the form

```
lm(log(y) ~ x)
```

Then \( \hat{\beta}_0 = \log(A) \) and \( \hat{\beta}_1 = p \). If a power law is a good fit for the data then

```
xyplot(log(y) ~ log(x))
```

will produce a roughly linear plot.

Fitting a power law results in estimates for the parameters \( \hat{\beta}_0 = \log(A) \) and \( \hat{\beta}_1 = p \). Note that we can use logarithms with any base for this transformation. Typically natural logarithms are used (that’s what \( \log() \)
Non-Linear Relationships

In some specific applications, base 10 logarithms (\texttt{log10}() in R) or base 2 logarithms (\texttt{log2}() in R) this yields the commonly used scale for $\beta_0 = \log(A)$, the constant of proportionality.

Some common situations that are modeled with power laws include drag force vs. speed, velocity vs. force, and frequency vs. force.

**Exponential Laws**

Relationships of the form

\[
y = AB^x = Ae^{Cx}
\]

are often called exponential laws. The two parameters are the base $B = e^C$ and a constant of proportionality $A$. Exponential laws can also be linearized by taking logarithms:

\[
\log(y) = \log(AB^x) = \log(A) + x \log(B)
\]

So if we fit a model of the form

\[
\text{lm}(\log(y) \sim x)
\]

Then $\beta_0 = \log(A)$ and $\beta_1 = \log(B) = C$. If an exponential law is a good fit for the data then

\[
\text{xyplot}(\log(y) \sim x)
\]

will produce a roughly linear plot.

Fitting an exponential law results in estimates for the parameters $\beta_0 = \log(A)$ and $\beta_1 = \log(B) = C$. Again, we are assuming natural logarithms.

Some common situations that are modeled with exponential laws include population growth and radioactive decay. Note that exponential growth models are typically only good approximations over a limited range since exponential functions eventually grow quickly, and often some external constraints will limit this growth. For example, a culture of bacteria may grow roughly exponentially for a while, but eventually, limits on space and nourishment will make it impossible for exponential growth to continue.

**Log-log and semi-log plots**

Graphs of $\log(y)$ vs. $\log(x)$ (log-log) or $\log(y)$ vs $x$ (semi-log) can be used to assess whether the power law or exponential law appears to apply in a given situation. If the law were a perfect description of the situation, all the points on the log-log or semi-log plot would fall along a straight line. In practice, the fit won’t be perfect, but the plot is a useful diagnostic.

In the old days, before computers could readily transform the data, special graph paper was produced with semi-log or log-log scales to facilitate this sort of plot. R can also create plots with transformed scales.

```r
x <- 1:10
y <- 3 * x^1.5
xyplot(log(y) ~ log(x))
xyplot(log(y) ~ x)
xyplot(y ~ x, scales = list(x = list(log = TRUE), y = list(log = TRUE)))
```

Math 241 : Spring 2014 : Pruim

Last Modified: April 24, 2014
8. Non-Linear Relationships

8.1.2 Other Models That Can Be Transformed to Linear

The three laws above are not the only kinds of relationships that can be transformed to linear.

Example 8.1.1. A chemical engineering text book suggest a law of the form

\[ \log\left(-\frac{dC}{dt}\right) = \log(k) + \alpha \log(C) \]

where \(C\) is concentration and \(t\) is time.

This is equivalent to

\[ -\frac{dC}{dt} = k \cdot C^\alpha \]
\[ -\int C^{-\alpha} dC = \int k \, dt \]
\[ -\frac{1}{1-\alpha} C^{1-\alpha} = kt + d \]
\[ \frac{1}{\beta} C^{-\beta} = kt + d \]
\[ C^{-\beta} = \beta kt + \beta d \]

If we know \(\beta = \alpha - 1\) (i.e., if we know \(\alpha\)), then we can fit a linear model using

\[ \text{lm}(C^{-\beta(1/beta)} ~ t) \]

The intercept of such a model will be \(\beta d\) and the slope will be \(\beta k\), from which we can easily recover \(d\) and \(k\).
Alternatively, if we know \( d = 0 \) (i.e., if we know that \( C = 0 \) when \( t = 0 \)), then we can use

\[
\log(C^{-\beta}) = -\beta \log(C) = \log(\beta k t) = \log(\beta k) + \log t
\]

\[
\log(C) = -\frac{\log(\beta k)}{\beta} - \frac{1}{\beta} \log t
\]

Now if we fit a model of the form

\[
\text{lm}(C \sim \log(t))
\]

the intercept will be \(-\frac{\log(\beta k)}{\beta}\) and the slope will be \(-\frac{1}{\beta}\). From this we can solve for \( k \) and \( \beta \).

**Example 8.1.2.** Continuing the previous example, we will fit the following data

```r
Concentration <- data.frame(
  time=c(0, 50, 100, 150, 200, 250, 300), # minutes
  concentration=c(50, 38, 30.6, 25.6, 22.2, 19.5, 17.4) # mol/dm^3 * 10^3
)
xyplot(concentration ~ time, data=Concentration)
```

under the assumption that \( \alpha = 2 \), so \( \beta = 1 \). In this case, our relationship becomes

\[
\frac{1}{C} = -kt - d.
\]

We can now fit a model and see how well it does.

```r
conc.model <- lm(1/concentration ~ time, data = Concentration)
summary(conc.model)
```

```r
## Call:
## lm(formula = 1/concentration ~ time, data = Concentration)
## ## Residuals:
## 1 2 3 4 5 6 7
## ## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.01e-02 8.76e-05 230 3.0e-11
## time 1.25e-04 4.86e-07 257 1.7e-11
```
This provides estimates for the intercept $-\beta_d$ and the slope $-\beta_k$ of our model. We can divide by $-\beta$ to obtain estimates for $d$ and $k$.

Of course, we should always look to see whether the fit is a good fit.

Notice that these residuals are very small relative to the values for concentration. (We can see this from the vertical scale of the plot and also from the small value for residual standard error in the summary output.) The shape of the residual plot would be more disturbing if the magnitudes were larger and if there were more data. As is, even if there is some systematic problem, it is roughly five orders of magnitude smaller than our concentration measuremments, which likely can’t be measured to that degree of accuracy.

If we want to show the fit on top of the original data, we must remember to untransform the response, since the model is a model for $1/C$:
8.1.3 The Ladder of Re-expression

Sometimes we have data for which there is no theory (yet) to suggest the form of a model. In such a case, we may let the data help suggest a model. If we find a model that fits well, we can return to the question of whether there is an explanation for that type of model.

In the 1970s, Mosteller and Tukey introduced what they called the ladder of re-expression and bulge rules [Tuk77, MT77] that can be used to suggest an appropriate transformation to improve the fit when the relationship between two variables \( (x \text{ and } y \text{ in our examples}) \) is monotonic and has a single bend. Their idea was to apply a power transformation to \( x \) or \( y \) or both – that is, to work with \( x^a \) and \( y^b \) for an appropriate choice of \( a \) and \( b \). Tukey called this ordered list of transformations the ladder of re-expression. The identity transformation has power 1. The logarithmic transformation is a special case and is included in the list associated with a power of 0. The direction of the required transformation can be obtained from Figure 8.1, which shows four bulge types, represented by the curves in each of the four quadrants. A bulge can potentially be straightened by applying a transformation to one or both variables, moving up or down the ladder as indicated by the arrows. More severe bulges require moving farther up or down the ladder. A curve bulging in the same direction as the one in the first quadrant of Figure 8.1, for example, might be straightened by moving up the ladder of transformations for \( x \) or \( y \) (or both), while a curve like the one in the second quadrant, might be straightened by moving up the ladder for \( y \) or down the ladder for \( x \).

This method focuses primarily on transformations designed to improve the overall fit. The resulting models may or may not have a natural interpretation. These transformations also affect the shape of the distributions of the explanatory and response variables and, more importantly, of the residuals from the linear model (see Exercise 8.5). When several transformations lead to reasonable linear fits, these other factors may lead us to prefer one over another.

Example 8.1.3. Q. The scatterplot in Figure 8.2 shows a curved relationship between \( x \) and \( y \). What transformations of \( x \) and \( y \) improve the linear fit?

A. This type of bulge appears in quadrant IV of Figure 8.1 so we can hope to improve the fit by moving up the ladder for \( x \) or down the ladder for \( y \). As we see in Figure 8.3, the fit generally improves as we move down and to the right – but not too far, lest we over-correct. A log-transformation of the response \((a = 1, b = 0)\) seems to be especially good in this case. Not only is the resulting relationship quite linear, but the residuals appear to have a better distribution as well.

Example 8.1.4. Some physics students conducted an experiment in which they dropped steel balls from various heights and recorded the time until the ball hit the floor. We begin by fitting a linear model to this data.

```r
ball.model <- lm(time ~ height, balldrop)
summary(ball.model)
```
8. Non-Linear Relationships

[Diagram of bulge rules and ladder of re-expression]

**ladder of re-expression**

<table>
<thead>
<tr>
<th>power</th>
<th>transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$x \mapsto x^3$</td>
</tr>
<tr>
<td>2</td>
<td>$x \mapsto x^2$</td>
</tr>
<tr>
<td>1</td>
<td>$x \mapsto x$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>$x \mapsto \sqrt{x}$</td>
</tr>
<tr>
<td>0</td>
<td>$x \mapsto \log(x)$</td>
</tr>
<tr>
<td>$-1$</td>
<td>$x \mapsto \frac{1}{x}$</td>
</tr>
<tr>
<td>$-2$</td>
<td>$x \mapsto \frac{1}{x^2}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Figure 8.1: Bulge rules and ladder of re-expression.

[Scatterplot diagram]

Figure 8.2: A scatterplot illustrating a non-linear relationship between $x$ and $y$. 
Figure 8.3: Using the ladder of re-expression to find a better fit.
At first glance, the large value of $r^2$ and the reasonably good fit in the scatterplot might leave us satisfied that we have found a good model. But a look at the residual plot reveals a clear curvilinear pattern in this data. A knowledgeable physics student knows that (ignoring air resistance) the time should be proportional to the square root of the height. This transformation agrees with Tukey’s ladder of re-expression, which suggests moving down the ladder for height or up the ladder for time.

```r
ball.modelT <- lm(time ~ sqrt(height), balldrop)
summary(ball.modelT)
```

```r
## Call: ## lm(formula = time ~ sqrt(height), data = balldrop) ## ## Residuals: ## Min 1Q Median 3Q Max ## -0.008777 -0.003885 0.000057 0.003056 0.012555 ## ## Coefficients: ## Estimate Std. Error t value Pr(>|t|) ## (Intercept) 0.01608 0.00408 3.94 5e-04
```
8. Non-Linear Relationships

```
## sqrt(height) 0.43080 0.00486 88.58 <2e-16
##
## Residual standard error: 0.00523 on 28 degrees of freedom
## Multiple R-squared: 0.996, Adjusted R-squared: 0.996
## F-statistic: 7.85e+03 on 1 and 28 DF, p-value: <2e-16
```

```
xypol(time ~ height, balldrop, panel = panel.lm, model = balldrop
xplot(balldrop, w = 1)
```

This model does indeed fit better, but the residual plot indicates that there may be some inaccuracy in the
measurement of the height. In this experiment, the apparatus was set up once for each height and then several
observations were made. So any error in this set-up affected all time measurements for that height in the same
way. This could explain why the residuals for each height are clustered the way they are since it violates the
assumption that the errors are independent. (See Example 8.1.5 for a simple attempt to deal with this problem.)

**Example 8.1.5.** One simple way to deal with the lack of independence in the previous example is to average
all the readings made at each height. (This works reasonably well in our example because we have nearly equal
numbers of observations at each height.) We pay for this data reduction in a loss of degrees of freedom, but it
may be easier to justify that the errors in average times at each height are independent (if we believe that the
errors in the height set-up are independent and not systematic).

```
require(plyr) # to access ddply
balldropavg <- ddply(balldrop, .(height), summarise, time = mean(time))
balldropavg
```

```
##   height time
## 1 0.281 0.2484
## 2 0.453 0.3020
## 3 0.600 0.3526
## 4 0.701 0.3716
## 5 0.868 0.4176
## 6 1.328 0.5148
```

```
ball.modelA <- lm(time ~ sqrt(height), balldropavg)
summary(ball.modelA)
```

```
##
## Call:
## lm(formula = time ~ sqrt(height), data = balldropavg)
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.004944   0.004840   1.019    0.316
## sqrt(height) 0.400500   0.004630  86.617   <2e-16
```

Math 241 : Spring 2014 : Pruim

Last Modified: April 24, 2014
Using a square root transformation on averaged height measurements in the balldrop data gives a similar fit but a very different residual plot. The interpretation of this model is also different.

Notice that the parameter estimates are essentially the same as in the preceding example. The estimate for σ has decreased some. This makes sense since we are now estimating the variability in averaged measurements rather than in individual measurements.

Of course, we’ve lost a lot of degrees of freedom, and as a result, the standard error for our parameter estimate is about twice as large as before. This might have been different; had the mean values fit especially well, our standard error might have been smaller despite the reduced degrees of freedom.

One disadvantage of the data reduction is that it is hard to interpret the residuals (because there are fewer of them). At first glance there appears to be a downward trend in the residuals, but this is largely driven by the fact that the largest residual happened to be for the smallest fit.

Example 8.1.6. Q. Rex Boggs of Glenmore State High School in Rockhampton, Queensland, had an interesting hypothesis about the rate at which bar soap is used in the shower. He writes:

I had a hypothesis that the daily weight of my bar of soap [in grams] in my shower wasn’t a linear function, the reason being that the tiny little bar of soap at the end of its life seemed to hang around for just about ever. I wanted to throw it out, but I felt I shouldn’t do so until it became unusable. And that seemed to take weeks.

Also I had recently bought some digital kitchen scales and felt I needed to use them to justify the cost. I hypothesized that the daily weight of a bar of soap might be dependent upon surface area, and hence would be a quadratic function ....
The data end at day 22. On day 23 the soap broke into two pieces and one piece went down the plughole.

The data indicate that although Rex showered daily, he failed to record the weight for some of the days.

What do the data say in regard to Rex’s hypothesis?

A. Rex’s assumption that weight should be a (quadratic) function of time does not actually fit his intuition. His intuition corresponds roughly to the differential equation

\[
\frac{\partial t}{\partial W} = kW^{2/3},
\]

for some negative constant \(k\) since the rate of change should be proportional to the surface area remaining. (We are assuming that the bar shrinks in such a way that its shape remains proportionally unaltered.) Solving this equation (by separation of variables) gives

\[
W^{1/3} = kt + C.
\]

We can fit untransformed and transformed models \((\text{Weight}^{1/3} \sim \text{Day})\) to this data and compare.

```r
soap.model1 <- lm(Weight ~ Day, soap)
summary(soap.model1)
```

```markdown
## Call:
## lm(formula = Weight ~ Day, data = soap)
##
## Residuals:
##    Min  1Q Median  3Q Max
## -6.244 -1.295  0.308  1.394  5.504
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 123.141     1.382   89.1   <2e-16
## Day          -5.575      0.107  -52.2   <2e-16
##
## Residual standard error: 2.95 on 13 degrees of freedom
## Multiple R-squared: 0.995, Adjusted R-squared: 0.995
## F-statistic: 2.72e+03 on 1 and 13 DF, p-value: <2e-16
```

The high value of \(r^2\) and the scatterplot in Figure 8.4 (darker line) indicate that the untransformed model is already a good fit.

```r
soap.model2 <- lm(I(Weight^(1/3)) ~ Day, soap)
summary(soap.model2)
```

```markdown
## Call:
## lm(formula = I(Weight^(1/3)) ~ Day, data = soap)
##
## Residuals:
##    Min  1Q Median  3Q Max
## -0.311 -0.137  0.016  0.150  0.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 123.141     1.382   89.1   <2e-16
## Day          -5.575      0.107  -52.2   <2e-16
##
## Residual standard error: 0.137 on 13 degrees of freedom
## Multiple R-squared: 0.995, Adjusted R-squared: 0.995
## F-statistic: 2.72e+03 on 1 and 13 DF, p-value: <2e-16
```

For now, it suffices to know that larger values of \(r^2\) generally indicate a better fit. We will discuss \(r^2\) and what it measures in Section ??.
The transformed model in this case actually fits worse. The higher value of $r^2$ for the untransformed model is an indication that it performs better. Figure 8.4 shows a scatterplot with both fits. The data do not support Rex’s assumption that a transformation is necessary. We can also fit a quadratic model of the form $\text{Weight} \sim I(\text{Day}^2)$, but this model is worse still. Fitting a full quadratic model requires two predictors ($\text{Day}$ and $\text{Day}^2$) and so will have to wait until our discussion of multiple linear regression. The scatterplot and especially the residual plot both show that the residuals are mostly positive near the ends of the data and negative near the center. Part of this is driven by a flattening of the pattern of dots near the end of the measurement period. Perhaps as the soap became very small, Rex used slightly less soap than when the soap was larger. Exercise 8.2 asks you to remove the last few observations and see how that affects the models.

Finally, since a linear model appears to fit at least reasonably well (but see Exercise 8.2), we can give a confidence interval for $\beta_1$, the mean amount of soap Rex uses each shower.

```
confint(soap.model1)
```

```
## 2.5 % 97.5 %
## (Intercept) 120.155 126.127
## Day -5.806 -5.344
```

### 8.2 Nonlinear Least Squares

Another approach to non-linear relationships is called **nonlinear least squares** or **nonlinear regression**. In this approach, instead of attempting to transform the relationship until it becomes linear, we fit a nonlinear function by minimizing the the sum of the squared residuals relative to that (parameterized) nonlinear function (form). That is, our model now becomes

$$y = f(x) + \varepsilon$$
where $f$ may be any parameterized function.

The R function for fitting these models is `nls()`. This functions works much like `lm()`, but there are some important differences:

1. Because the model does not have to be linear, we have to use a more verbose description of the model.

2. Numerical optimization is used to fit the model, and the algorithm used needs to be given a reasonable starting point for its search. Specifying this starting point simultaneously lets R know what the parameters of the model are. (Each quantity with a starting value is considered a parameter, and the algorithm will adjust all the parameters looking for the best fit – i.e., the smallest MSE (and hence also the smallest SSE and RMSE).

Let’s illustrate with an example.

**Example 8.2.1.** Returning to the ball dropping experiment, let’s fit

$$\text{time} = \alpha_0 + \alpha_1 \sqrt{\text{height}}$$ (8.1)

using nonlinear least squares.

```r
nls.model <- nls( time ~ alpha0 + alpha1 * sqrt(height),
                  data=balldrop,
                  start=list(alpha0=0, alpha1=1) )
```

Notice how the model formula compares with the formula in (8.1). The starting point for the algorithm is specified with `start=list(alpha0=0, alpha1=1)`, which also declares that the parameters to be fit.

We can obtain the coefficients of the fitted model with

```r
nls.model
```

```r
## Nonlinear regression model
## model: time ~ alpha0 + alpha1 * sqrt(height)
## data: balldrop
## alpha0 alpha1
## 0.0161 0.4308
## residual sum-of-squares: 0.000765
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 2.11e-07
```

or

```r
coeff(nls.model)
```

```r
## alpha0 alpha1
## 0.01608 0.43080
```

A more complete summary can be obtained by
summary(nls.model)

##
## Formula: time ~ alpha0 + alpha1 * sqrt(height)
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## alpha0  0.01608   0.00408   3.94   5e-04
## alpha1  0.43080   0.00486  88.58  <2e-16
##
## Residual standard error: 0.00523 on 28 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 2.11e-07

We can restrict our attention to the coefficients table with

coef(summary(nls.model))

##     Estimate Std. Error t value Pr(>|t|)
## alpha0  0.01608   0.00408   3.937 4.976e-04
## alpha1  0.43080   0.00486  88.580 7.732e-36

f <- makeFun(nls.model)
xyplot(time ~ height, data = balldrop)
plotFun(f(height) ~ height, add = TRUE, col = "gray40")
xyplot(resid(nls.model) ~ fitted(nls.model))
We can compare this to the ordinary least squares model by plotting both together on the same plot.

```r
lm.model <- lm(time ~ sqrt(height), data = balldrop)
g <- makeFun(lm.model)
xyplot(time ~ height, data = balldrop)
plotFun(f(height) ~ height, add = TRUE, col = "gray40", lwd = 3)
plotFun(g(height) ~ height, add = TRUE, col = "red", lwd = 1, lty = 2)
```

In this particular case, there is very little difference between the two models, but this is not always the case.

```r
c coef(nls.model)
## alpha0  alpha1
## 0.01608 0.43080
c coef(lm.model)
## (Intercept) sqrt(height)
## 0.01608 0.43080
```

**Example 8.2.2.** Here is an example where we fit a different model to the `balldrop` data, namely

\[ \text{time} = \alpha \times \text{height}^p \]

```r
power.model <- nls(time ~ alpha * height^power, data=balldrop,
                   start=c(alpha=1, power=.5) )
c coef(summary(power.model))
## Estimate Std. Error t value Pr(>|t|)
## alpha  0.4472 0.001343 333.09 6.333e-52
## power  0.4797 0.005805 82.63 5.388e-35
```

A power law can also be fit using `lm()` by using a log-log transformation.

```r
power.model2 <- lm(log(time) ~ log(height), data = balldrop)
c coef(summary(power.model2))
```
Again, the parameter estimates (and uncertainties) are very similar. Recall that to compare our intercept in the second model to the $\alpha$ value in the first model, we must untransform:

$$\exp(\text{coef(power.model2)[1]})$$

## (Intercept)  
## 0.4459

We can use the delta method to estimate the uncertainty. Since \(\frac{d}{dx}e^x = e^x\) the uncertainty is approximately

$$0.4459 \cdot 0.0043 = 0.0019$$

**Example 8.2.3.** In addition to comparing estimated parameters and their uncertainties, we should always look at the residuals of our model. For both the linear regression and the nonlinear least squares models, the assumption is that the error terms are independent, normally distributed, and have a common standard deviation. From the plots below we see

1. The nonlinear least squares model is a better match for these assumptions than the linear regression model.

2. Both models reveal a lack of independence – at a given height, the residuals move up or down as a cluster as was discussed in the previous section. Neither model is designed to handle this flaw in the design of the experiment.
Now let’s take a look at an example where we need the extra flexibility of the nonlinear least squares approach.

**Example 8.2.4.** A professor at Macalester College put hot water in a mug and recorded the temperature as it cooled. Let’s see if we can fit a reasonable model to this data

```r
xyplot(temp ~ time, data = CoolingWater, ylab = "temp (C)", xlab = "time (sec)"
```

Our first guess might be some sort of exponential decay

```r
cooling.model1 <- nls( temp ~ A * exp( -k * time), data=CoolingWater,
                        start=list(A=100, k=0.1) )
f1 <- makeFun(cooling.model1)
xyplot( temp ~ time, data=CoolingWater, xlim=c(-50,300), ylim=c(0,110),
        ylab="temp (C)", xlab="time (sec)"
)
plotFun( f1(time) ~ time, add=TRUE)
```
That doesn’t fit very well, and there is a good reason. The model says that eventually the water will freeze because

\[ \lim_{t \to \infty} Ae^{-kt} = 0 \]

when \( k > 0 \). But clearly our water isn’t going to freeze sitting on a lab table. We can fix this by adding in an offset to account for the ambient temperature:

```r
cooling.model2 <- nls( temp ~ ambient + A * exp( k * (1+time) ), data=CoolingWater,
                       start=list(ambient=20, A=80, k=-1) )
f2 <- makeFun(cooling.model2)
xyplot( temp ~ time, data=CoolingWater, xlim=c(-50,300), ylim=c(0,110),
       ylab="temp (C)", xlab="time (sec)"
)
plotFun( f1(time) ~ time, add=TRUE, lty=2, col="gray80"
plotFun( f2(time) ~ time, add=TRUE, col = "steelblue"
```

![Graph showing the fit of the model](image)

This fits much better. Furthermore, this model can be derived from a differential equation

\[ \frac{dT}{dt} = -k(T_0 - T_{ambient}) , \]

known as Newton’s Law of Cooling.

Let’s take a look at the residual plot

```r
xyplot(resid(cooling.model2) ~ time, data = CoolingWater)
plot(cooling.model2, which = 1)
```

![Residual plots](image)

Hmm. These plots show a clear pattern and very little noise. The fit doesn’t look as good when viewed this way. It suggests that Newton’s Law of Cooling does not take into account all that is going on here. In particular, there is a considerable amount of evaporation (at least at the beginning when the water is warmer). More
complicated models that take this into account can fit even better. For a discussion of a model that includes evaporation, see http://stanwagon.com/public/EvaporationPortmannWagonMiER.pdf

8.2.1 Choosing Between Linear and Non-linear Models

So how do we choose between linear and non-linear models? Let’s enumerate some of the differences between them:

1. Some models cannot be expressed as linear models, even after transformations.
   In this case we only have one option, the non-linear model.

2. Linear models can be fit quickly and accurately without numerical optimization algorithms because they satisfy nice linear algebra properties.
   The use of numerical optimizers in non-linear least squares models makes them subject to potential problems with the optimizers. They may not converge, may converge to the wrong thing, or convergence may depend on choosing an appropriate starting point for the search.

3. The two types of models make different assumptions about the error terms.
   In particular, when we apply transformations to achieve a linear model, those transformations often affect the distribution of the error terms as well. For example, if we apply a log-log transformation to fit a power law, then the model is

\[
\log(y) = \beta_0 + \beta_1 \log(x) + \varepsilon \\
y = e^{\beta_0} x^{\beta_1} e^\varepsilon \\
y = \alpha x^{\beta_1} e^\varepsilon
\]

So the errors are multiplicative rather than additive and they have a normal distribution after applying the logarithmic transformation. This implies that the relative errors should be about the same magnitude rather than the absolute errors.
   This is potentially very different from the nonlinear model where the errors are additive:

\[
y = \alpha x^\beta + \varepsilon
\]

Plots of residuals vs. fits and qq-plots for residuals can help us diagnose whether the assumptions of a model are reasonable for a particular data set.

4. Linear models provide an easy way to produce confidence intervals for a mean response or an individual response.
   The models fit using nls() do not have this capability.

---

2 The model with evaporation adds another complication in that the resulting differential equation cannot be solved algebraically, so there is no algebraic formula to fit with nls(). But the method of least squares can still be used by creating a parameterized numerical function that computes the sum of squares and using a numerical minimizer to find the optimal parameter values. Since the use of numerical differential equation solvers is a bit beyond the scope of this course, we’ll leave that discussion for another day.
Exercises

8.1 In Example 8.1.4 we applied a square root transformation to the height. Is there another transformation that yields an even better fit?

8.2 Remove the last few days from the soap data set and refit the models in Example 8.1.6. How much do things change? Do the residuals look better, or is there still some cause for concern?

8.3 For each of the following relationships between a response $y$ and an explanatory variable $x$, if possible find a pair of transformations $f$ and $g$ so that $g(y)$ is a linear function of $f(x)$:

\[ g(y) = \beta_0 + \beta_1 f(x). \]

For example, if $y = ae^{bx}$, then $\log(y) = \log(a) + bx$, so $g(y) = \log(y)$, $f(x) = x$, $\beta_0 = \log(a)$, and $\beta_1 = b$.

a) $y = ab^x$.

b) $y = ax^b$.

c) $y = \frac{1}{a+bx}$.

d) $y = \frac{x}{a+bx}$.

e) $y = ax^2 + bx + c$.

f) $y = \frac{1}{1 + e^{a+bx}}$.

g) $y = \frac{100}{1 + e^{a+bx}}$.

8.4 What happens to the role of the error terms ($\varepsilon$) when we transform the data? For each transformation from Exercise 8.3 start with the form

\[ g(y) = \beta_0 + \beta_1 f(x) + \varepsilon \]

and transform back into a form involving the untransformed $y$ and $x$ to see how the error terms are involved in these transformed linear regression models.

It is important to remember that when we fit a linear model to transformed data, the usual assumptions of the model are that the errors in the (transformed) linear form are additive and normally distributed. The errors may appear differently in the untransformed relationship.

8.5 The transformations in the ladder of re-expression also affects the shape of a distribution.

a) If a distribution is symmetric, how does the shape change as we move up the ladder?

b) If a distribution is symmetric, how does the shape change as we move down the ladder?

c) If a distribution is left skewed, in what direction should we move to make the distribution more symmetric?

d) If a distribution is right skewed, in what direction should we move to make the distribution more symmetric?

8.6 By attaching a heavy object to the end of a string, it is easy to construct pendulums of different lengths. Some physics students did this to see how the period (time in seconds until a pendulum returns to the same location) depends on the length (in meters) of the pendulum. The students constructed pendulums of lengths varying from 10 cm to 16 m and recorded the period length (averaged over several swings of the pendulum). The resulting data are in the pendulum data set.
8. Non-Linear Relationships

a) Fit a power law to this data using a transformation and a linear model. How well does the power law fit? What is the estimated power in the power law based on this model?

b) Fit a power law to this data using a nonlinear model. How well does the power law fit? What is the estimated power in the power law based on this model?

c) Compare residual plots and normal-quantile plots for the residuals for the two models. How do the models compare in this regard?

8.7 The pressure data set contains data on the relation between temperature in degrees Celsius and vapor pressure in millimeters (of mercury). With temperature as the predictor and pressure as the response, use transformations or nonlinear models as needed to obtain a good fit. Make a list of all the models you considered and explain how you chose your best model. What does your model say about the relationship between pressure and temperature?

8.8 The cornnit data set in the package faraway contains data from a study investigating the relationship between corn yield (bushels per acre) and nitrogen (pounds per acre) fertilizer application in Wisconsin. Using nitrogen as the predictor and corn yield as the response, use transformations (if necessary) to obtain a good fit. Make a list of all the models you considered and explain how you chose your best model.

8.9 The data set actgpa contains the ACT composite scores and GPAs of some randomly selected seniors at a Midwest liberal arts college.

a) Give a 95% confidence interval for the mean ACT score of seniors at this school.

b) Give a 95% confidence interval for the mean GPA of seniors at this school.

c) Use the data to estimate with 95% confidence the average GPA for all students who score 25 on the ACT.

d) Suppose you know a high school student who scored 30 on the ACT. Estimate with 95% confidence his GPA as a senior in college.

e) Are there any reasons to be concerned about the analyses you have just done? Explain.

8.10 In the absence of air resistance, a dropped object will continue to accelerate as it falls. But if there is air resistance, the situation is different. The drag force due to air resistance depends on the velocity of an object and operates in the opposite direction of motion. Thus as the object’s velocity increases, so does the drag force until it eventually equals the force due to gravity. At this point the net force is 0 and the object ceases to accelerate, remaining at a constant velocity called the terminal velocity.

Now consider the following experiment to determine how terminal velocity depends on the mass (and therefore on the downward force of gravity) of the falling object. A helium balloon is rigged with a small basket and just the right ballast to make it neutrally buoyant. Mass is then added and the terminal velocity is calculated by measuring the time it takes to fall between two sensors once terminal velocity has been reached.

The drag data set contains the results of such an experiment conducted by some undergraduate physics students. Mass is measured in grams and velocity in meters per second. (The distance between the two sensors used for determining terminal velocity is given in the height variable.)

By fitting models to this data, determine which of the following “drag laws” matches the data best:
• Drag is proportional to velocity.
• Drag is proportional to the square of velocity.
• Drag is proportional to the square root of velocity.
• Drag is proportional to the logarithm of velocity.

8.11 Construct a plot that reveals a likely systematic problem with the drag (see Exercise 8.10) data set. Speculate about a potential cause for this.

8.12 Exercise 8.11 suggests that some of the data should be removed before analyzing the drag data set. Redo Exercise 8.10 after removing this data.

8.13 The spheres data set contains measurements of the diameter (in meters) and mass (in kilograms) of a set of steel ball bearings. We would expect the mass to be proportional to the cube of the diameter. Fit a model and see if the data reflect this.

8.14 The spheres data set contains measurements of the diameter (in meters) and mass (in kilograms) of a set of steel ball bearings. We would expect the mass to be proportional to the cube of the diameter. Using appropriate transformations fit two models: one that predicts mass from diameter and one that predicts diameter from mass. How do the two models compare?

8.15 The utilities data set has information from utilities bills at a Minnesota residence. Fit a linear model that predicts thermsPerDay from temp.

a) What observations should you remove from the data before doing the analysis? Why?
b) Are any transformations needed?
c) How happy are you with the fit of your model? Are there any reasons for concern?
d) Interpret your final model (even if it is with some reservations listed in part c)). What does it say about the relationship between average monthly temperature and the amount of gas used at this residence? What do the parameters represent?