Chapter 15, part 2: The simple linear model

S520

These notes are written to accompany Trosset section 15.5.

The simple linear regression model

Let $x_1, \ldots, x_n$ be a list of real numbers for which $s_x > 0$.

Under the **simple linear regression model**:

- Associated with each x_i is an (independent) random variable

$$Y_i \sim \text{Normal}(\mu_i, \sigma^2)$$

- The population means μ_i satisfy the linear relation

$$\mu_i = E(Y_i) = \beta_0 + \beta_1 x_i$$

  for some real numbers β_0 and β_1.

The simple linear regression model is sufficient to do all the thing we usually do in regression – estimate parameters using least squares; find confidence intervals; and use the normal to find probabilities.

(Note: This equation is not used to calculate μ_i given x_i, since in real life we never know the population parameters, β_0 and β_1. This is the assumption in the simple regression model. It means that the expected value of Y_i for a give value of x_i is linearly related to x_i. The least square regression line (or fitted line, or $\hat{y}$) you learned last week is to estimate the parameter μ_i.)

It may be easier to understand when you can use the simple linear model by rewriting as the FOUR ASSUMPTIONS below, which are basically in order of importance.

1. **Linearity:** $E(Y_i) = \beta_0 + \beta_1 x_i$.

Even this assumption is often not literally true – in the social sciences, a linear relationship between X and the expected value of Y is usually an approximation rather than a true law of nature. However, the approximation should be good. If the relationship is curved, then there would seem to be no point in fitting a linear model, and using such a model might be highly misleading. With modern statistical software like R, there's little difficulty in fitting a nonlinear function: using the `loess()` function to fit a curve is (almost) as easy as using the `lm()` function to fit a straight line. Alternatively, a transformation such as taking the log of the Y-variable may make the relationship more linear while fixing some of the problems below as well.

Strong nonlinearity is often apparently just by looking at the scatterplot,

though a **residual plot** (see below) might give a clearer picture.

2. **Independence of errors.** Let the **errors** be

$$\epsilon_i = Y_i - (\beta_0 + \beta_1 x_i)$$

These ϵ_i's should be independent. (Technically errors only need to be uncorrelated, but it's hard to distinguish between "independent" and "uncorrelated" in practice.)

Usually, independence will be determined by how your data was collected. If your data set is a simple random sample, it should be close enough to independent.

If it's a sequence of observations in time (a **time series**,) it'll generally be dependent.

(Note: This assumption of the simple linear model requires the errors (ϵ_i's) be independent.

Two random variables, X and Y, are "Uncorrelated" if their covariance is 0, or $E(XY) = E(X)E(Y)$. But the "independence" between X and Y is defined from the perspective of probability distributions, which makes it a stronger assumption. All independent R.V.s are uncorrelated, but not all uncorrelated R.V.s are independent. If you are interested in the details of these concepts, you many want to read other textbooks on probability. The difference between these two is beyond the scope of this course.)

Independence matters for a couple of reasons:

- You might get slightly better parameter estimates if you take the dependence structure of your observations into accout.
- Your predictions might be *much* better if you take the dependence into account. The latter is a particular issue with time series. Suppose the x_i's are points in time, and the Y_i's are an economic series like unemployment. Then simple linear regression will give you bad predictions for the future values of Y_i, because in some sense simple regression treats all the data points the same, whereas when predicting unemployment, the most recent observations are much, much more important than, say, the observations from 1946.

If you want to make predictions for strongly dependent data, you have to learn how to model the dependence. So if you have time series data, take a time series course; if you have spatial data, take a spatial course; and so on.

(Note: "To model the dependence" is beyond the scope of the course. In this introductory course, we only focus on the simple regression model, where we assume independence between y observations or error terms. But in reality, data might be much more complicated than what the simple regression can model. In those cases, we should account for the dependence when we estimate parameter or make prediction.

For examples, in time series, the independent variable is "time" t, and dependent variable is the subject you are interested in. Time series analysis is useful when you want to know how something evolves with time. In this kind of data analysis, the error terms are usually dependent on time, the independent variable t. Therefore, in time series, it is reasonable to make prediction not only based on independent variable "time t", but also based on the y observations for the previous recent time points. How related those "y" observations are is determined by some dependence structure, which is usually specified in the model.

"more recent" means closer time points. For example, if you want to predict next year's unemployment rate, this year's data should be quite related to your prediction, and so is last year's. And these recent years' data can help you make better prediction than the data 20 or 30 years ago.)

3. **Homoskedasticity, i.e. equal variance of errors.** This means that σ^2 is the same for each observation x_i. If this is not the case, we call the data **heteroskedastic**, and:

- There will be more efficient ways to estimate the regression line, e.g. using **weighted** least squares rather than ordinary least squares.
- The standard errors (and hence confidence intervals and tests) will be off.

Again, there are ways of dealing with heteroskedastic data if it arises. For example, instead of using **parametric** models like the t- and F-distributions, inference can be done using the **bootstrap**, which you can learn about in a good modern nonparametric statistics course. Once again, log transforming Y often magically solves (or at least strongly alleviates) heteroskedasticity as well as nonlinearity.

(Note: It might not be easy for you to understand the "equal variance of errors". In the population data or the model, for each x_i, there could be many different values for y_i, and the expected value of those y_i's is μ_i which depends on x_i, but the variance of those y_i's (same as the variance of the errors) is σ which is common

for differnt x_i value. However, this is an assumption of the simple regression model, and we cannot check this using the population data. We will check it using residuals that are calculated from the sample data.

Errors are the difference between the y observation and the average (expected value) y in the population data, and residuals are the difference between the y observation and the estimated average y value (or fitted y value) in the sample. They have similar meanings.

Standard error is to measure the variation of the y observations around the regression line. So it makes sense to calculate the standard deviation of the residuals as the measure if residuals have similar spread/variance for different x values. If not, using the standard deviation of residuals as a measure of variation for the whole line is not reasonable. "Off" means that standard error will not be an accurate measure in the case where there is heteroskedasticity. Since the calculation of confidence intervals and hypothesis tests both depend on the value of the standard error, those results won't be valid /reliable either.

These above two bullets are consequences of heteroskedasticity (or unequall variances), and how to deal with it when you encounter a case like that. You will need take more advanced statistical courses to learn these.

In this introductory course, what you need to know is how to diagnose–check whether residuals for different x values have same/similar variance. If not, there are methods to deal with it. Some approaches are mentioned here. To derive why "log" transformation or sometimes "square root" transformation might solve this problem, involves the transformation in probability distribution which is not a focus of this course.)

The best way to diagnose heteroskedsticity is the **residual plot**. A **residual** is the difference between the observed Y and its **fitted value**; that is, the value that would be predicted by the regression line:

$$z_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$$

There is one residual for each observation. Note that residuals are not quite the same as the true errors:

$$\epsilon_i = y_i - (\beta_0 + \beta_1 x_i)$$

because the residuals use the regression estimates of the β coefficients rather than their true values. However, if your sample size is large then the differences should be small.

(Note: $\hat{\beta}$ is the estimate of β. The β's are parameters in the model, and the $\hat{\beta}$'s are estimates of those parameters and can be calculated from the sample data.

Compare the definitions of residuals and errors from the two equations. Since beta and beta-hat are different, the errors and residuals are not exactly the same.

In the simple linear regression model, the errors are assumed to have same variance at different x values. If the model is correctly, then after we fit the regression line, the residuals, which represent similar things, should have about the same variances for different x's. So in the residual plot (a scatterplot of residuals vs x), we would like to see no change in the spread of residuals for different x values.)

> **If the linear model is correctly specified, the residuals plotted against the x-variable should be scattered around the x-axis with no noticable change in their typical distance from zero.**

This is a bit hard to understand in words, so see the section below with simulated examples of residual plots.

4. **Normality of errors.** This is only really important if you want to make probabilistic predictions for individuals. For example, if you want to know the probability that a 5'10" kid weighs over 150 pounds, then in addition to the regression prediction, you need to know whether the weights of 5'10" kids follow a Normal distribution or some other distribution. For all other purposes (such as estimating β_0 and β_1), then a reasonably large sample size is an adequate substitute for the normality assumption. So if you have hundreds of observations (with no ridiculous outliers) and linearity, independence, and

homoskedasticity are satisfied, you can fit a regression line and do inference on the slope parameter even if the errors aren't normal.

The best check for approximate normality of errors is a normal QQ plot of the residuals.

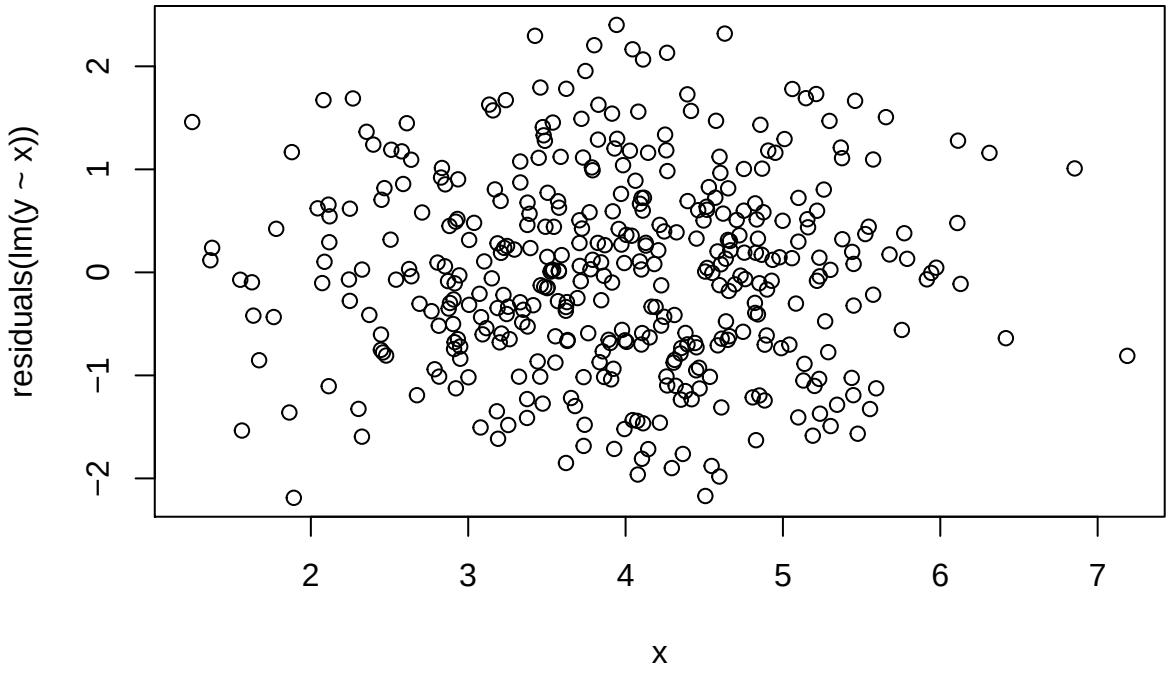
Even if you don't want to make probabilistic predictions, extreme skewness or outliers in the residuals may give you ideas for improving your model.

(Note: In this assumption of the simple regression model, we assume the errors (or y observations for a given x value) are normally distributed. If this assumption is violated, then when we make a probability statement using normal distribution, it won't be accurate. However, this normality assumption only matters to the prediction for an individual y value. For the other inferences, such as the fitted regression line, the estimation of the slope, it will be ok if the sample size is large enough, as the CLT will apply.)

Examples of residual plots

Here's a residual plot for linear, homoskedastic data:

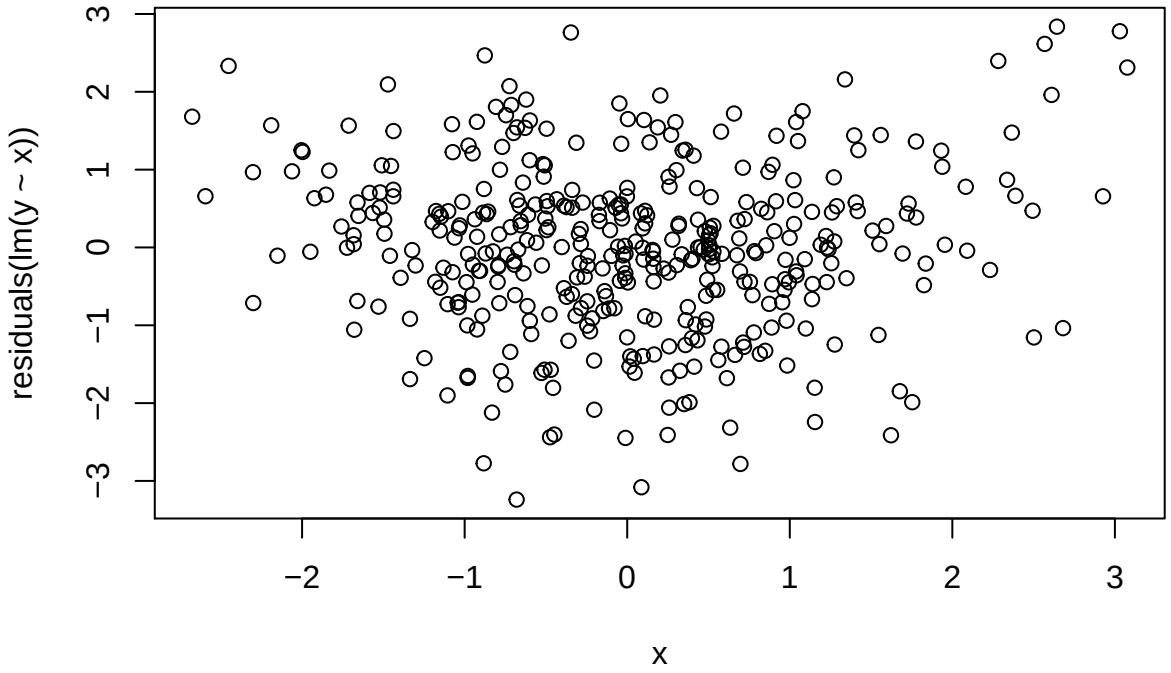
```
x = rnorm(400, mean = 4)
y = x + rnorm(400)
plot(x, residuals(lm(y ~ x)))
```



There seems to be nothing going on here, which is what we want to see. Note that the residuals on the right-hand side might *seem* less spread out, but this is an illusion due to the lack of data on the extreme right.

Next, we draw a residual plot for homoskedastic data that has moderate nonlinearity:

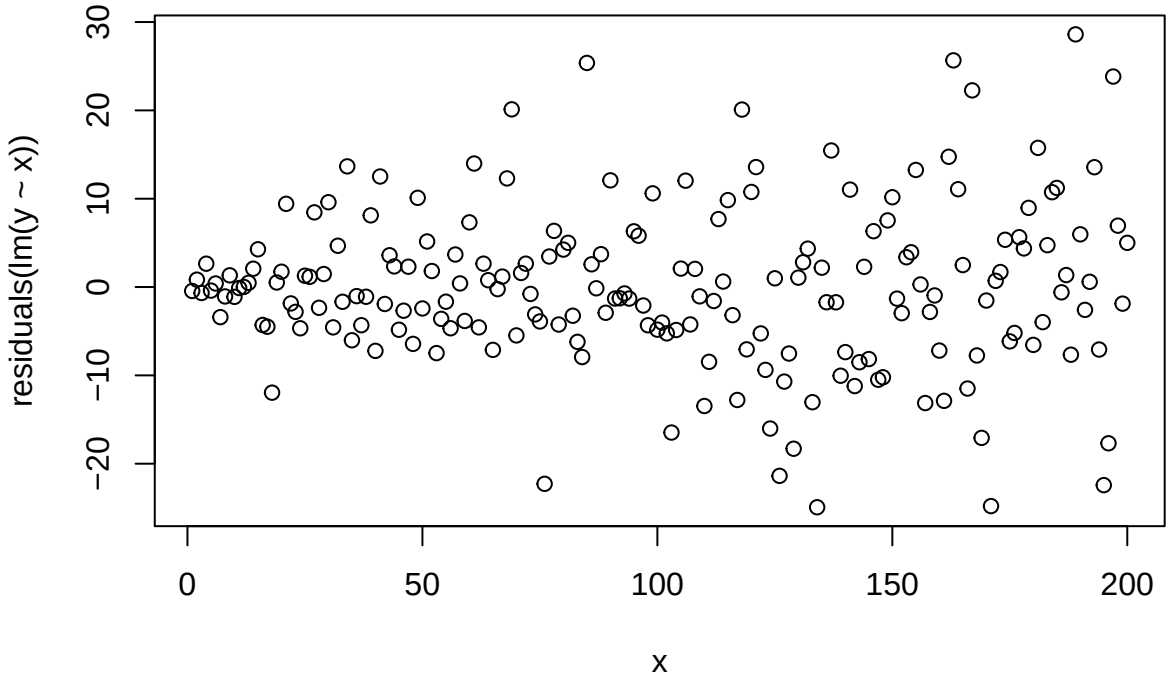
```
x = rnorm(400)
y = 0.2 * x^2 + rnorm(400)
plot(x, residuals(lm(y ~ x)))
```



The nonlinearity takes a little effort to see. The tell is that on the left edge, the residuals are almost all above zero; a similar thing happens at the right edge. So a U-shaped curves looks to fit the residual plot better than a horizontal line. That tells us there's nonlinearity, and we should fit the data using a nonparametric or other nonlinear method.

Next, we draw a residual plot for data that's linear but heteroskedastic.

```
x = 1:200
y = x + rnorm(200, sd = sqrt(x))
plot(x, residuals(lm(y ~ x)))
```

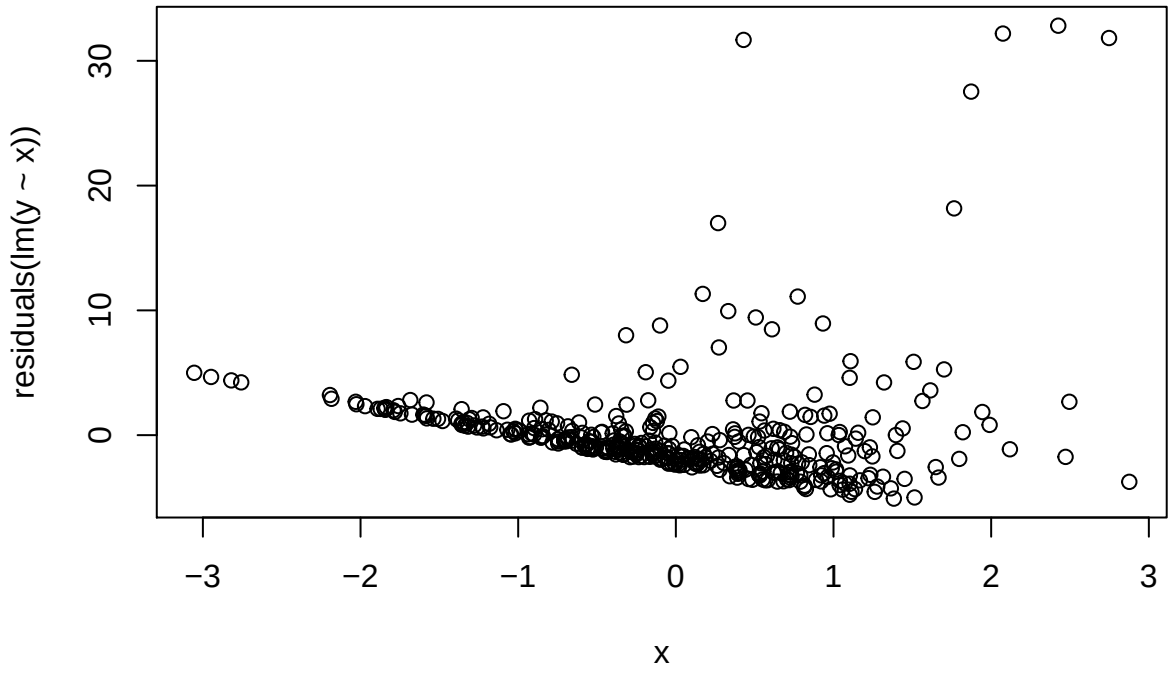


A common type of heteroskedastic data has the residuals close together on one edge, while they get more

and more spread out as you go across the graph. If the heteroskedasticity is accompanied by nonlinearity, consider a transformation. However, in this graph, the residuals still look scattered around zero; it's just that the amount of scatter increases as we go from left to right. While there are better methods (e.g. weighted regression), it might not be the worst thing to just use simple linear regression here for description – after all, as the name suggests, it's simple. However, since the heteroskedasticity is pretty bad here, confidence intervals and tests for the slope parameter will generally be a bit biased here regardless of sample size. (What you should do instead is surprisingly controversial, but most statisticians would be okay with **bootstrapping**, which is however a computationally intensive method. Whether you can get away with much simpler one-line of R code methods like sandwich estimators is a heated topic, but that's way beyond the scope of this course.)

Finally, we do an example where the residuals are both nonlinear and heteroskedastic.

```
x = rnorm(400)
y = exp(x + rnorm(400))
plot(x, residuals(lm(y ~ x)))
```



When your residual plot is this bad, it's best to start over again with a transformation (log being the first choice) or a completely different method.

Example: Real(er) heights and weights

The `NHANES` data set in the R package of the same name contains survey data collected by the National Center for Health Statistics. We wish to look at the relationship of height and weight. If you've never used the package before, you'll have to install it first (you only have to do this once):

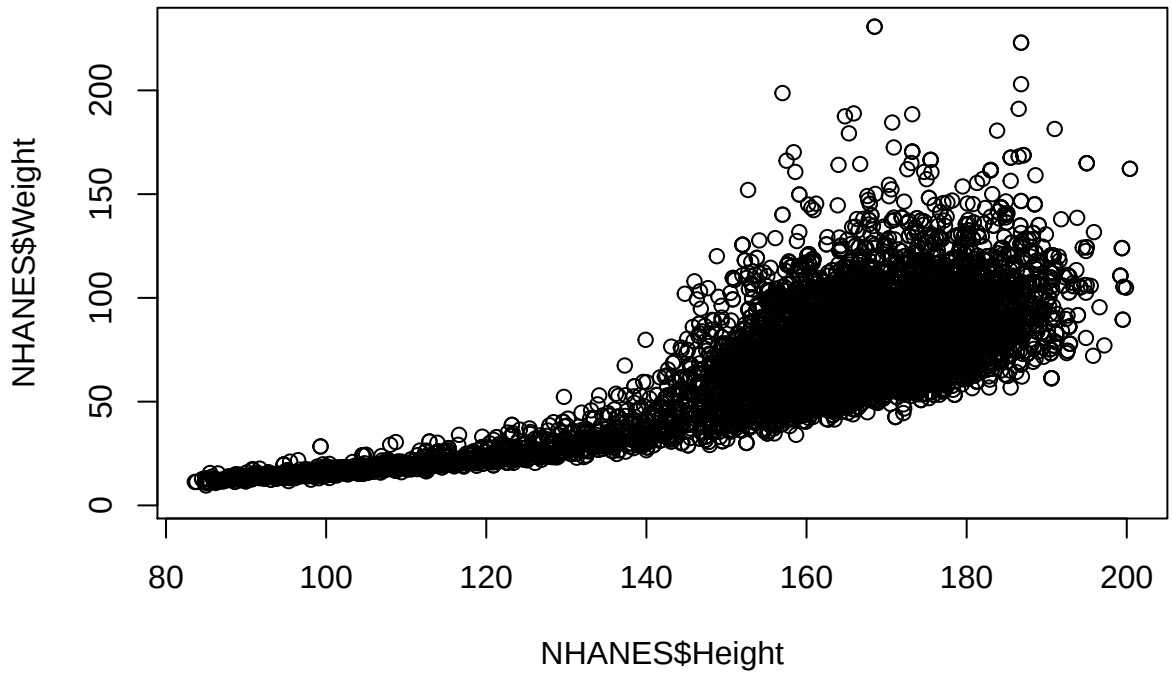
```
install.packages("NHANES")
```

Load the package (you have to do this every time you want to use it):

```
library(NHANES)
```

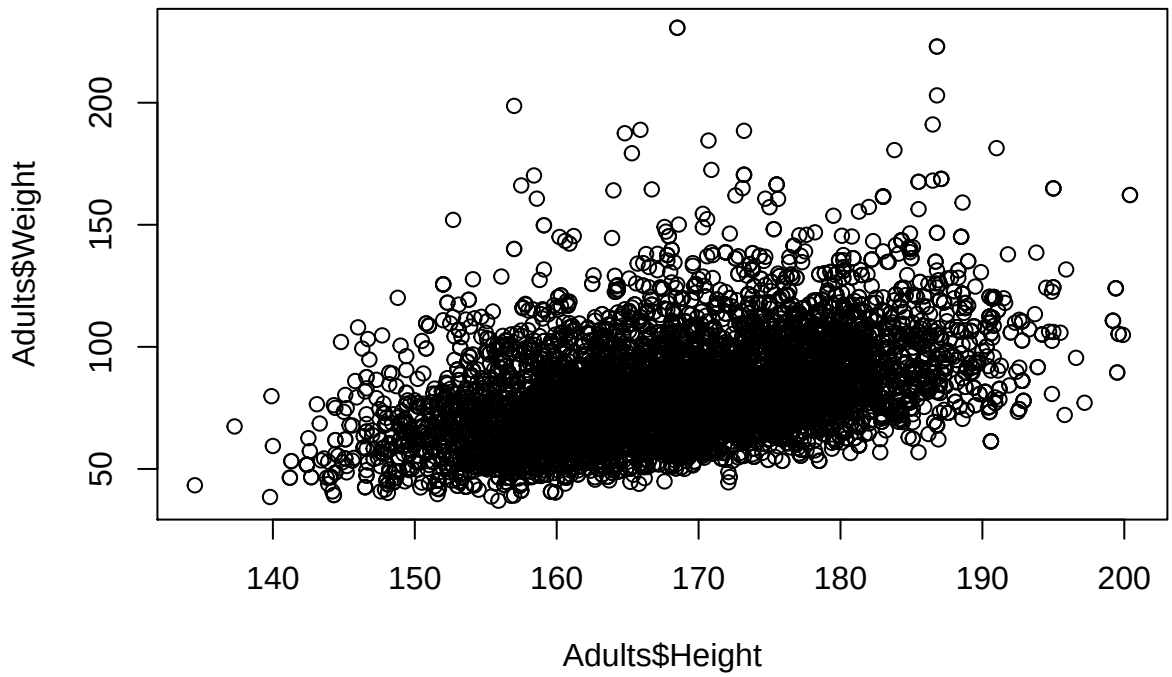
The data set contains `Height` and `Weight` variables in cm and kg respectively. First, we plot the data:

```
plot(NHANES$Height, NHANES$Weight)
```



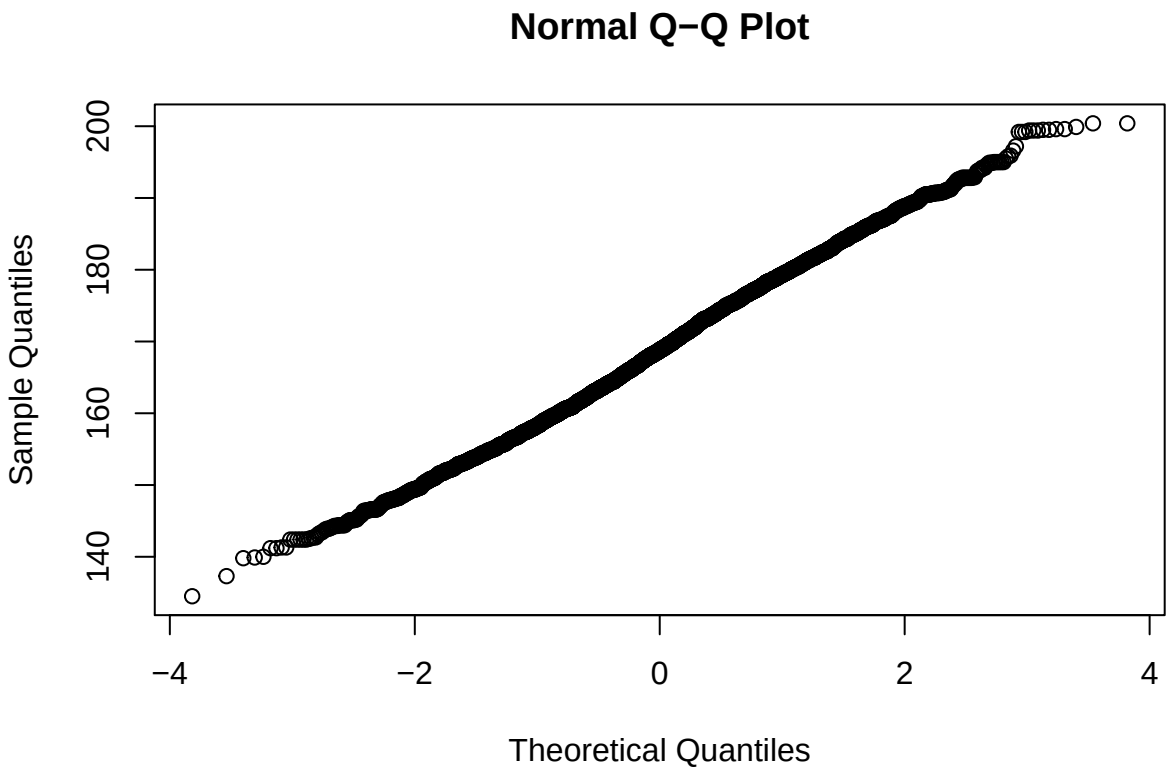
This looks weird, but from the heights we can quickly guess that the data set includes both adults and children, which doesn't seem like it would be useful. Let's limit the data to just adults and plot again.

```
Adults = NHANES[NHANES$Age >= 18, ]
plot(Adults$Height, Adults$Weight)
```

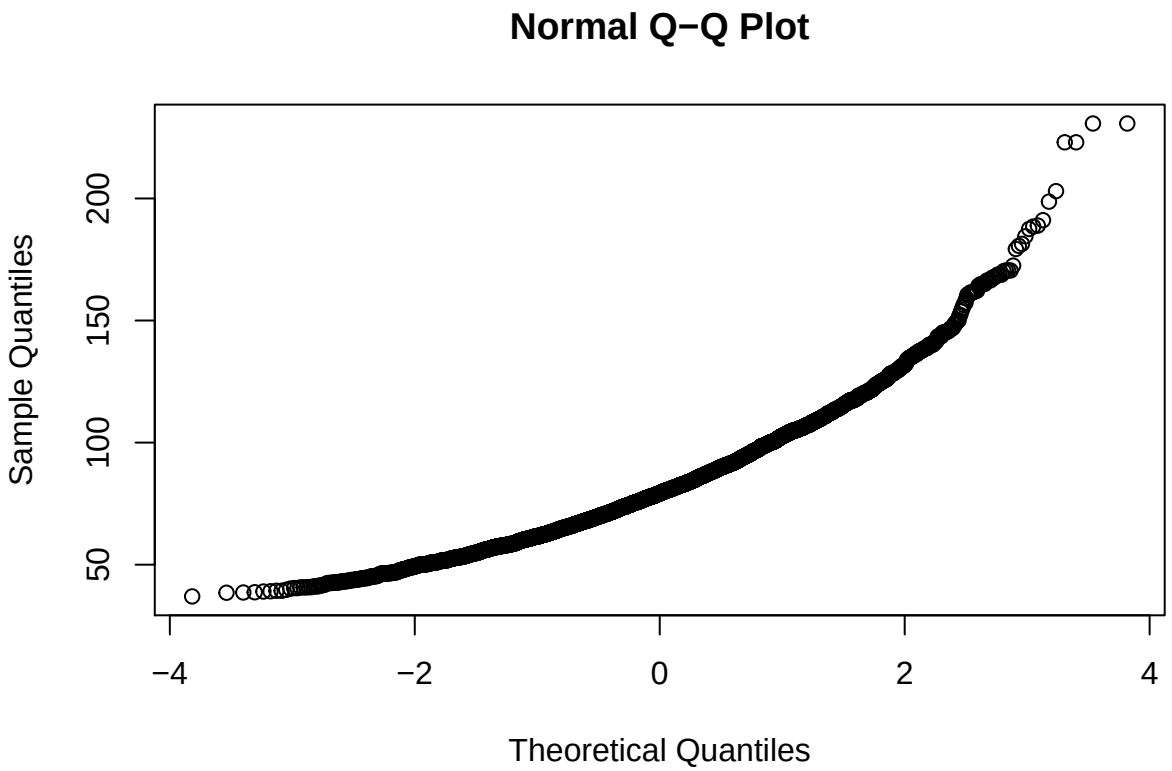


It's a bit hard to tell whether the relationship is a straight line, but it doesn't look like our nice bivariate normal ellipse. We can probe a bit more deeply by studying the distributions of height and weight separately. Draw QQ plots:

```
qqnorm(Adults$Height)
```

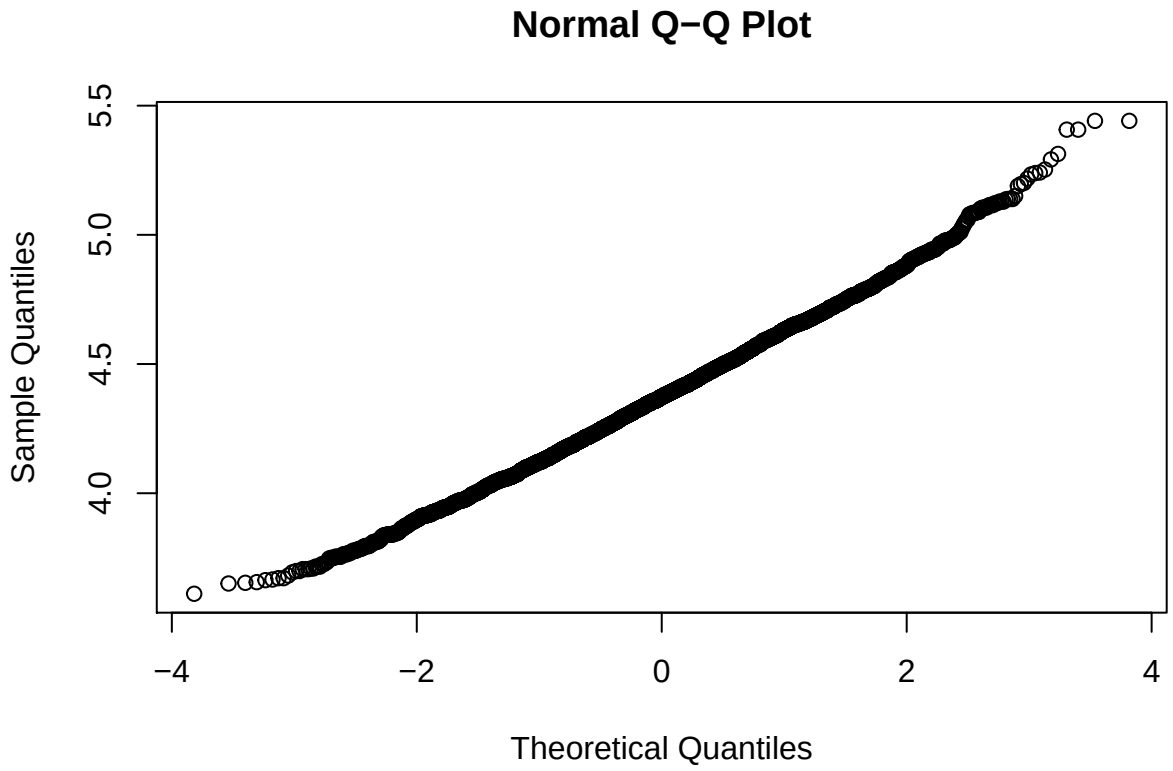



```
qqnorm(Adults$Weight)
```



Height is reasonably close to normal, but the normal QQ plot for weight curves upward, indicating right-skew. Any time you have right-skewed positive data, a log transformation is worth considering.

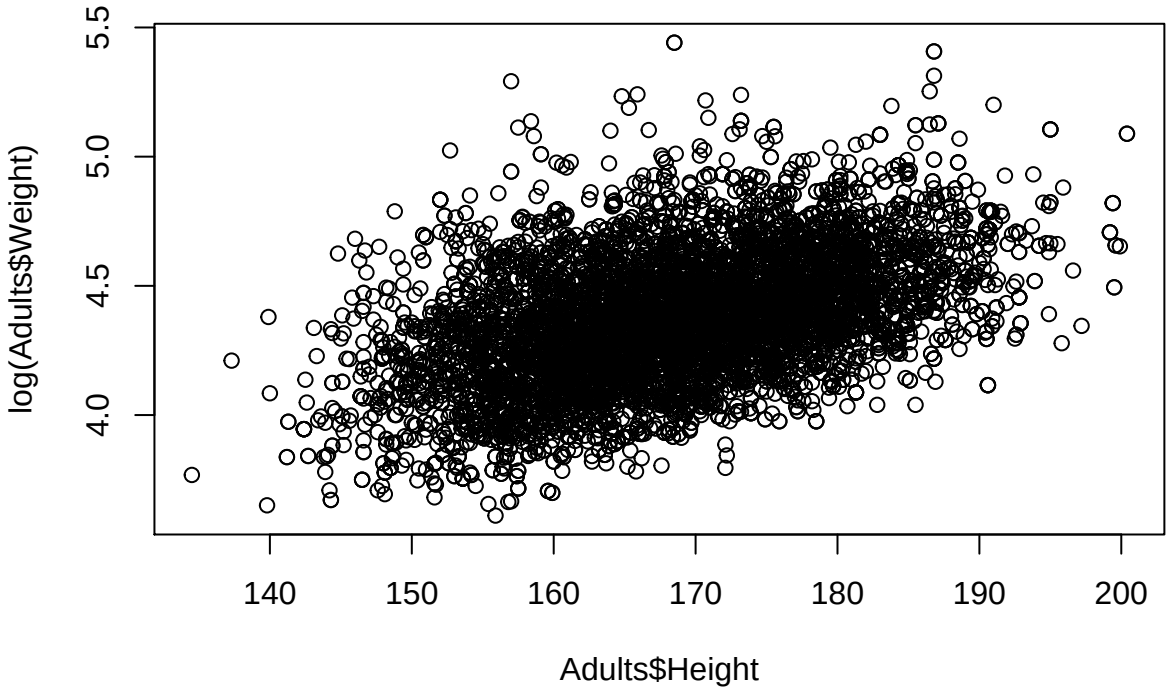
```
qqnorm(log(Adults$Weight))
```



While the log transformation makes the normal QQ plot better, we can still see there's a little bit of upward curve in the plot. So the log of weights isn't perfectly normal. That's okay – nothing we do requires our Y-variable to be perfectly normal – but we'll need to check our model carefully after fitting.

We proceed to look at the relationship between height and log weight. Draw the scatterplot:

```
plot(Adults$Height, log(Adults$Weight))
```



As far as we can tell from the scatterplot, a straight line looks like a reasonable fit. We could find the regression line from scratch, but let's just use the built-in `lm()` command:

```
log.model = lm(log(Weight) ~ Height, data = Adults)
log.model
```

```
##
## Call:
## lm(formula = log(Weight) ~ Height, data = Adults)
##
## Coefficients:
## (Intercept)       Height
##     2.37612      0.01184
```

Our prediction for log weight increases by 0.01184 for every extra inch of height. Remember that anything additive on the log scale is multiplicative on the original scale. So back-transforming:

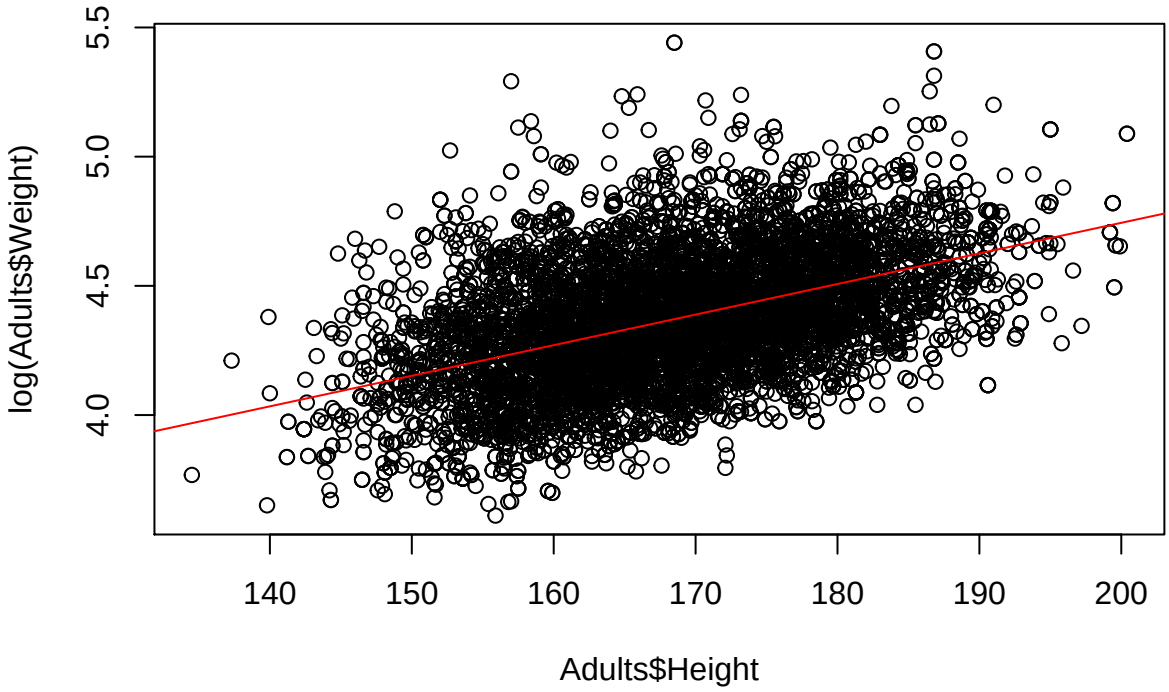
```
exp(0.01184)
```

```
## [1] 1.01191
```

tells us that an extra inch of height means we multiply the prediction for weight by 1.012. That is, an extra inch of height means we increase our prediction for weight by 1.2%. (This of course is only meaningful if our predictions are sensible.)

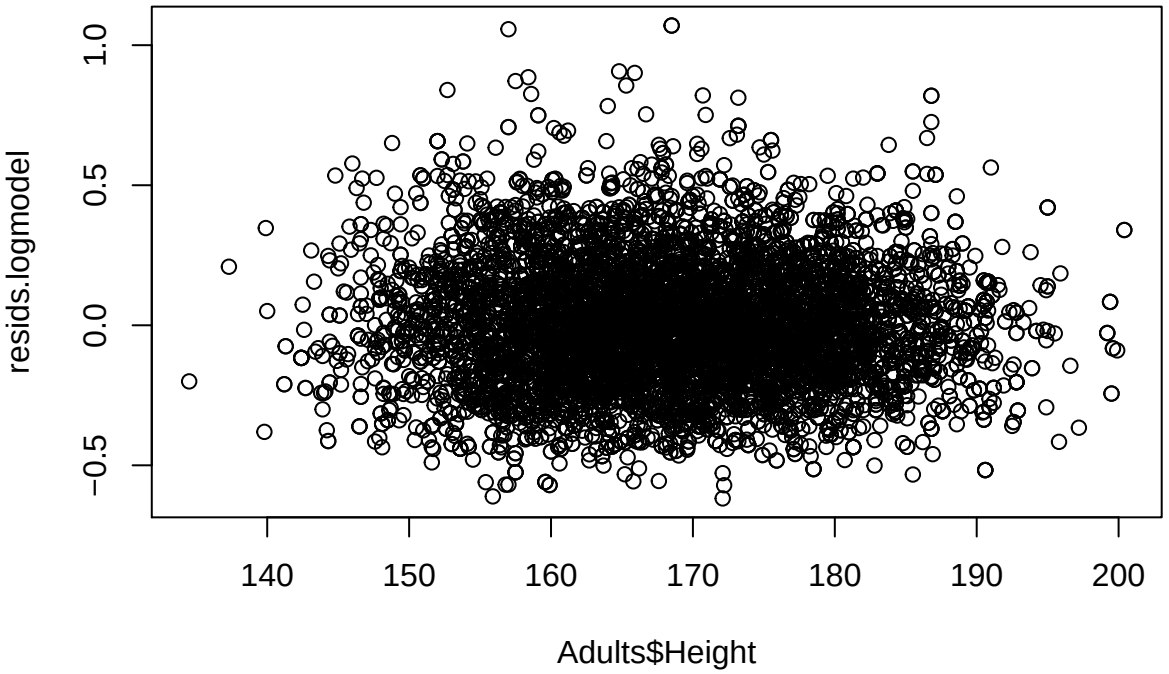
The `abline()` command makes it easy to add the regression line to the transformed data:

```
plot(Adults$Height, log(Adults$Weight))
abline(log.model, col = "red")
```



This looks okay, but looking at residuals gives a better idea of whether the model is a good fit. We calculate and plot the residuals:

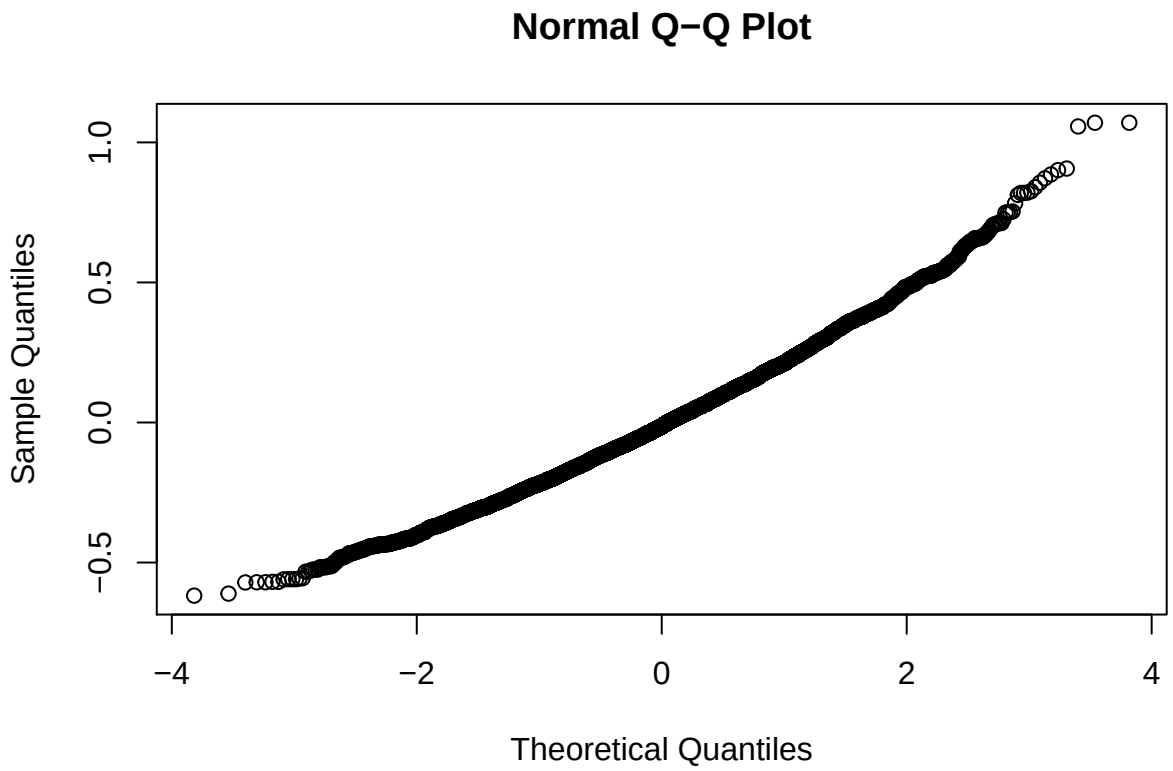
```
coefs = coefficients(log.model)
fitted = coefs[1] + coefs[2] * Adults$Height
resids.logmodel = log(Adults$Weight) - fitted
plot(Adults$Height, resids.logmodel)
```



There's no obvious trend in the residuals. Furthermore, the spread of the residuals doesn't seem to change much with height (though if you squint, the residuals on the extreme right look a little less spread out.) So any heteroskeasticity is mild.

We should also draw a normal QQ plot of the residuals:

```
qqnorm(resids.logmodel)
```



It's not horrible, but there's a little more curvature here than we would like.

So to go through our linear model assumptions:

- *Linearity:* consistent with this assumption
- *Independence of errors:* presumably, if the data was generated/collected sensibly
- *Equal variance of errors:* close to true
- *Normality of errors:* not horrible but not normal

So what does all this imply as to what we can do with this data?

- *Is the regression line a good description of the data?* After the log transformation then yes, it seems so.
- *Can we do classical intervals, tests, etc.?* This is probably okay, though there might be better approaches if we knew more statistics.
- *Can we do probabilistic prediction under the assumption of normal errors?* This is somewhat dangerous – the non-normality would introduce some bias. It would be better to fit a more sophisticated model, or else derive probabilities directly from the data.