Section 10
Simple Linear Regression

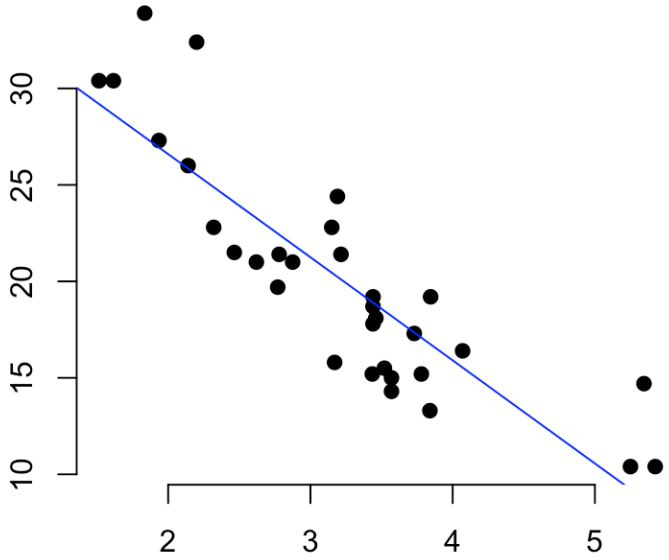
10.1 – The Least Squares Criterion

Introduction

Last section, we learned how to compare two populations in various contexts. Another perspective we could take on this is that we've learned methods analyzing the relationship between two variables in the following contexts:

- Comparing a quantitative variable and a categorical variable with two categories/options
- Comparing two categorical variables, each with two categories/options

In this section, we will investigate how we would analyze the relationship between two quantitative variables. This involves finding a line of best fit to the data, as shown in the picture below:



Finding the "best" line

As a starting point, consider the slope-intercept form of a line:

$$y = mx + b$$

This will be the basis for the model we create – our goal is to find the best such line for our data. How do we determine what is best? One thing we can think about is to minimize the *y*-distance of each point to the line we fit. In our previous class activity, we did this by attempting to fit a line that has the smallest sum of distances to each point.

How could we directly find a line that minimizes the distance? Minimizing a quantity will involve using calculus. One problem with this approach is that these distances are computed with an absolute value – this function has a value which is not differentiable. Instead, we usually find the line that minimizes the squared *y*-distance to the line, which would result in the -

_____.

Sum of squared errors

Of course, rarely (if ever) will we find data with two variables that fit perfectly on a line. As we previously discussed, each point will be some distance along the *y* direction from any line we fit. If we were to write each data point in the y variable, we could write them as a function of our line as so:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

This adds an error term into our model, which is the *y*-distance between a given point and the line. In this way, we typically think of our *y* variable in this model to be determined based on an *x* value and some error value. Thus, when we model the relationship between two variables in this way, we think of our *y* variable as being the _____ in our model, and the *x* variable as the _____.

Based on the motivation at the beginning of the lecture, our goal is to find a slope and intercept that minimize the squared errors, resulting in a least squares line. If we write out our errors as follows:

$$\varepsilon_i = y_i - \beta_0 - \beta_1 x_i$$

Thus, we can write out the sum of squared errors as the following:

$$SSE = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

To minimize this quantity, we can use calculus and take the derivative with respect to each the intercept and slope, and find the value that minimizes this *SSE* quantity. This is beyond the scope of this course, but if you were to minimize this quantity, you would get the following estimates:

$\beta_0 =$ $\beta_1 =$

A model-based approach

We also think of our linear regression model coming from a model-based approach. By assigning a model to the error term, we think of our line giving an expected value of *y* for a given *x* value, but of course values of *y* will differ from this expected value. This difference is the error term we've previously described. In this approach, we can say that each error is independently distributed as follows:

$$\varepsilon_i \sim$$

Therefore, each of our data points can be represented as having the following distribution:

$$y_i \sim$$

Because we think of fitting our least squares line by fitting a normal model to the errors, this gives us a list of assumptions to think about when using linear regression:

- The variables should have a linear relationship.
- Error terms have a normal distribution.
- The errors should have a constant variance.

These will be important to keep in mind as we go through fitting data to a least squares line in the examples to come.

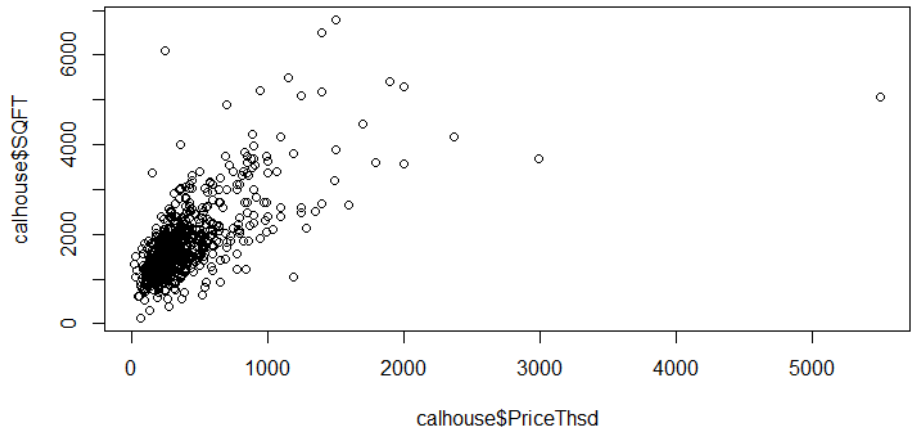
10.2 – Simple Linear Regression

Using R for linear regression

We will now do an example of a linear model based on data. So far, we've been talking about theoretical quantities, specifically, population parameters. When we talk about a least squares or linear regression line based on data, we use the following to represent the model:

$$y_i = b_0 + b_1 x_i + e_i$$

Notably, the e_i term is called a residual, the sample estimate of an error for a particular data point.

> *Example*: The data file "calhouse.csv" has data regarding 781 home sales in San Luis Obispo county, California in 2009. Create a model that estimates the square feet for a home you would expect for a given amount of money in this region.
>
> First we will load the data, and examine a scatterplot in R:
>
> ```
> plot(calhouse$PriceThsd, calhouse$SQFT)
> ```
>
> 
>
>
> Outside of one very expensive home, the scatterplot appears to confirm a linear relationship. Thus, we can use the code to build a linear model in R:
>
> ```
> model = lm(SQFT~PriceThsd, data=calhouse)
> ```
>
> To examine statistics based on this model, we can use the summary function:
>
> ```
> summary(model)
> ```

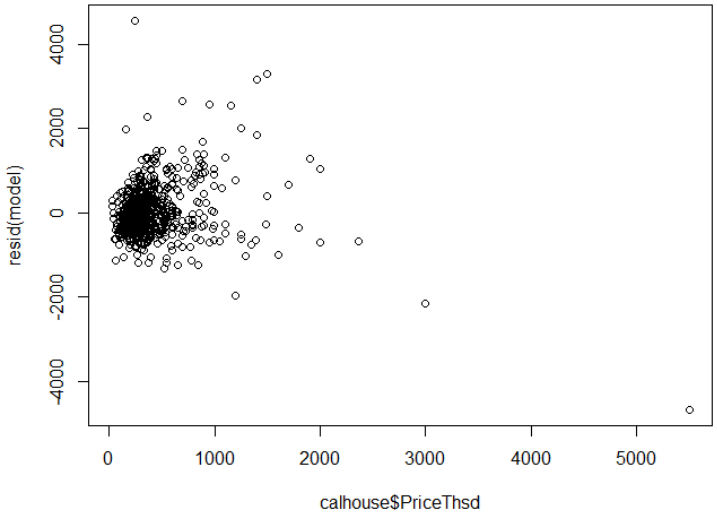
Based on the output, our final regression line is:

In the context of this problem, we would interpret the slope of the line as follows:

Checking assumptions

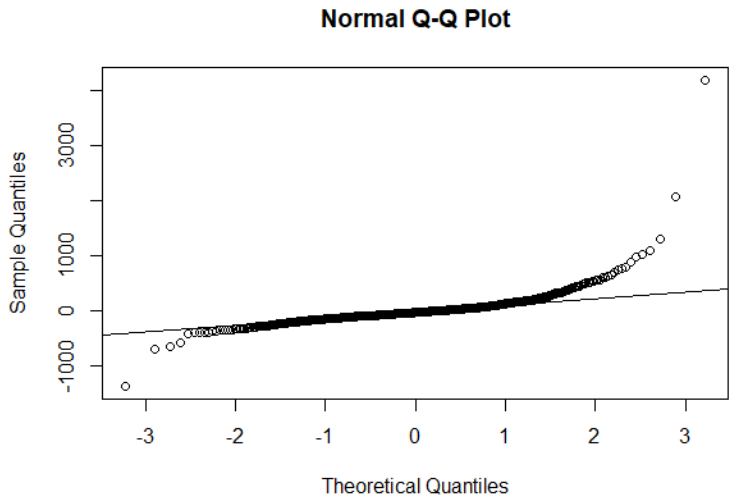
To check our model-based assumptions for this model, we would want to see if the errors have a constant variance, and if the errors are normally distributed. To check for this, we examine the residuals:

```
residuals = resid(model)
plot(calhouse$PriceThsd, residuals)
```



Here, we can see that the residuals mostly seem to have a constant variability regardless of their *x*-value, with the exception of a few outlier values. Now, to check for the normality of the residuals, we use a qq plot:

```
qqnorm(residuals)
qqline(residuals)
```



Here, we see that this assumption of normally distributed residuals may be violated. Creating a histogram of the residuals would confirm they are slightly skewed to the right. Let's keep this in the back of our minds for now and come back to this example later.

10.3 – Assessing and Using Linear Models

Hypothesis testing and confidence intervals for regression parameters

As previously mentioned, the slope and intercept values we get from are sample estimates, as they are based on sample data. Thus, even if there were really no relationship between the two variables we are analyzing (which would represent a slope of 0), we might get a positive or negative slope based on our sample data. Thus, it makes sense to consider a hypothesis test about our slope:

H_0:

H_a:

This test is actually already conducted for us in the summary output previously. Examine the coefficients table:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.157e+03  3.258e+01   35.51   <2e-16 ***
PriceThsd   1.561e+00  6.285e-02   24.83   <2e-16 ***
```

The "`t value`" is the t-test statistic for this test, and the "`Pr(>|t|)`" indicates the associated p-value. Clearly, it seems here that this the slope we have almost surely is not just dumb luck by the sampling we have taken on these houses! This test can be seen as a good diagnostic for if the predictor variable you are examining is providing valuable information about your response variable. When we get to fitting multiple predictor variables in a linear model in section 11, this becomes a useful diagnostic tool for determining which variables are important or unimportant.

From a similar perspective, we could see that the line of best fit we have is just an estimated regression line, and is an estimate of the population regression line.

As we've done in previous sections, we will compute confidence intervals on these values to determine where the true or population regression slope and intercept might be. To do this in R, we would use the same function we used to find confidence intervals for the mean:

```
confint(model, level=0.95)
```

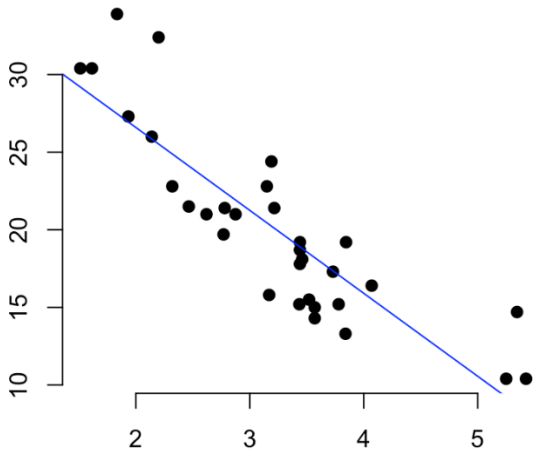
> *Example:* For the model on California houses created in this last example, find a confidence interval for the slope of this model, and interpret what this confidence interval for this slope means in the context of this problem.

Assessing the model

We will now discuss some other tools we can use to analyze a regression model. The first is called the coefficient of determination, or R^2. For a given data point with x-value x_i, let the predicted value of y for that x_i be $\hat{y}_i = b_1 x_i + b_0$. Then the formula for R^2 is given as follows:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

We interpret this value as the percentage of variance in our y variable that is explained by its linear relationship with the x variable. Why is that? Let's examine using a picture. Copied below is the same scatterplot from page 1:



It also turns out that the R^2 value from regression is the square of the sample _____ of the two variables, or R. This value ranges from -1 to 1, and contains information about the relative strength of the relationship and the direction relationship. The further R is from 0, the stronger the correlation, and the sign indicates the direction of the relationship. We can compute the correlation using the following R function:

```
cor(x, y)
```

However, it's important to note that this only measures relative strength – that is, we can compare correlation values across different pairs of variables, but the measurement itself doesn't mean anything itself. The coefficient of determination has an absolute interpretation as a percentage. Let's try interpreting this in our house price example.

> *Example:* For the model on California houses created in the last example, find the coefficient of determination and interpret this value in the context of the problem.

Prediction intervals

One goal of a regression model is to be able to predict a *y* value for a given *x* value. We've already learned how to make predictions with data from a single variable, but now we would like to make predictions for one variable based on values from some predictor or explanatory variable. We would want to use a *prediction interval*, which gives a range of possible values for a single *y* value based on an *x* value. To get this interval in R, we use the following code:

```
predict(model, data.frame(xvar=x), interval="predict", level=0.95)
```

> *Example:* Suppose you have a budget of $400,000 to buy a house in San Luis Obispo county. What would you predict the square footage for a house of that value would be? Give a 90% prediction interval.
>
> ```
> predict(model, data.frame(PriceThsd=400), interval="predict",
> level=0.90)
> ```

An important note about the confidence interval we computed earlier: remember that we found that the residuals were non-normal. This is somewhat problematic, as we used multipliers from the *t*-distribution to get this confidence interval, which assume that the value we are estimating (the slope) is normally distributed with an unknown population variance.

However, because of the CLT, we don't need to worry about this assumption too much in the context of confidence intervals so long as we had a large enough sample size. That being said, the normality assumption is more important with prediction intervals, which are based on a single data point rather than a parameter. Thus, we should be cautious about the prediction interval we created from this model.

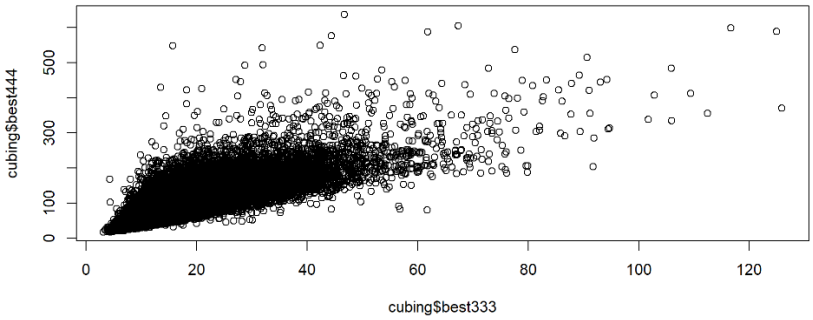
Dealing with heteroskedasticity

One assumption that can be quite tricky to deal with is when your residuals have non-constant variance, that is, the data is *heteroskedastic*. There are many ways to deal with this, but we will examine just one case in the next example.

Example: Personal best times in World Cube Association competitions for all competitors that have competed in the 3x3x3 and 4x4x4 events are given in the **cubing.csv** data file. Create a model that estimates the 4x4x4 solving time based on the 3x3x3 solving time.

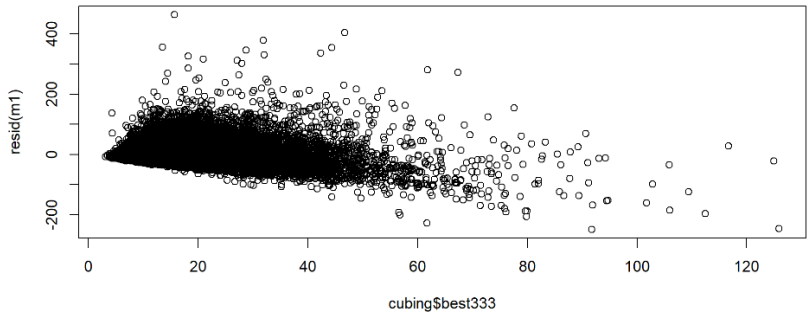
When creating such a model, we run into a problem -- the variability of 4x4x4 times increases as 3x3x3 times increase. This makes some intuitive sense: world class competitors will have very small differences in times among them, where the competitors who take a longer time to solve will have more differences among them. The scatterplot below shows this increasing variance.

```
plot(cubing$best333, cubing$best444)
```



We can also see that when we fit a linear regression model and look at a residual plot, we see that the variability of the residuals increases.

```
m1 = lm(best444~best333, data=cubing)
plot(cubing$best333, resid(m1))
```

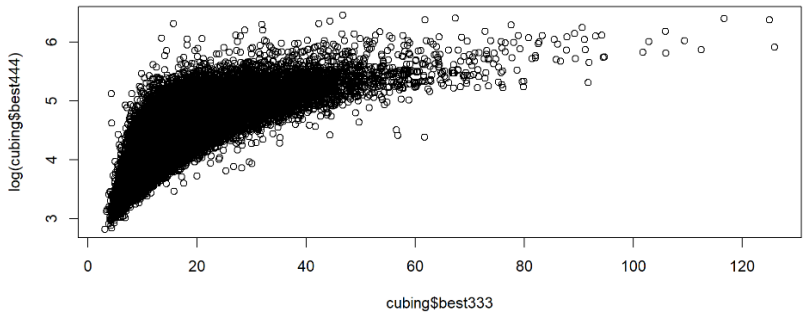


One way we can address this is to apply some sort of function on our response variable. Some good functions to try on data with increasing variability:

- $f(x) = \sqrt{x}$ (or other powers of x less than 1)
- $f(x) = \log(x)$
- $f(x) = 1/x$

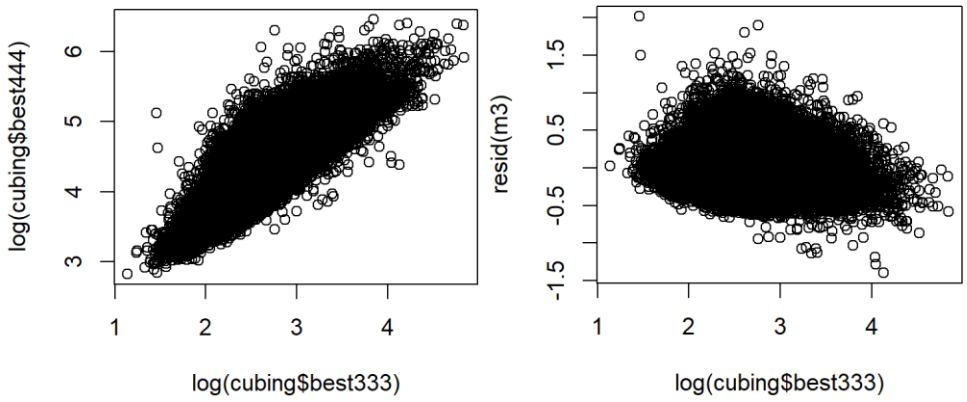
One issue with applying these transformations to only the response variable is that they affect the linearity. See this attempt to use a log transformation on the 4x4x4 times:

```
m2 = lm(log(best444)~best333, data=cubing)
plot(cubing$best333, log(cubing$best444))
```



Transforming the explanatory variable as well can help to fix the linearity issue. For this data, using logs on both variables seems to help fix the heteroskedasticity and preserve the overall linear relationship.

```
m3 = lm(log(best444)~log(best333), data=cubing)
plot(log(cubing$best333), log(cubing$best444))
plot(log(cubing$best333), resid(m3))
```



While not perfect, the range of *y* values for a particular *x* value is more constant than before. Examining the coefficients of our new model:

```
summary(m3)
```

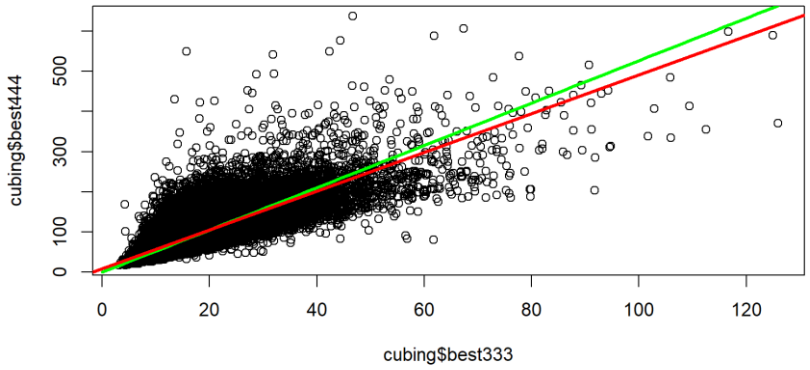
```
   Coefficients:
               Estimate Std. Error t value Pr(>|t|)
  (Intercept)  1.648017   0.006503   253.4   <2e-16 ***
  log(best333) 1.002963   0.002406   416.8   <2e-16 ***
```

We would write out our model for this relationship in the following non-linear form:

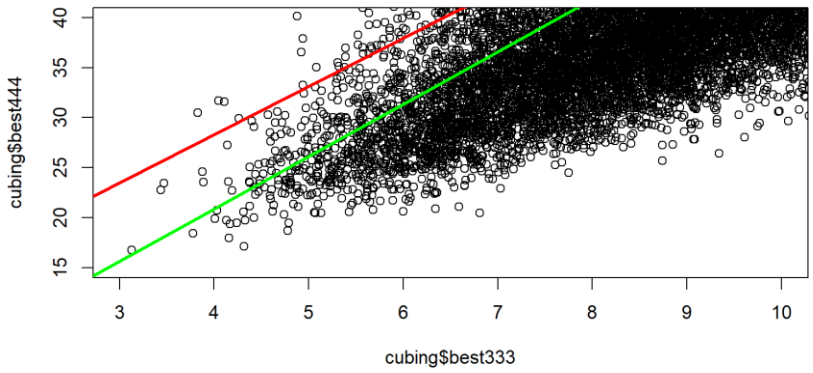
$$\log(y) = 1.648 + 1.003 \log(x)$$

It is worth noting that using log transformations on both the explanatory and response variables turns our intercept value into a slope, and the slope becomes the power of *x*. This means that our line will always go through the origin now, which you should confirm makes sense with your data.

Here is a comparison of the original model to the new model on the scatterplot of the original data. The red line is the ordinary least squares model, and the green line is the new model based on log data.



These two lines may not seem too different, but if we zoom in on the bottom left of the plot, we can see that this is actually a vast difference.



For these world class competitors, the predicted 4x4x4 solve time is about 10-15 seconds slower than with this model that reduces heteroskedasticity. Since the residuals are so small for the model with the red line, it doesn't try too hard to minimize these residuals, resulting in a bad prediction. By creating a model that evens out the residual size across all 3x3x3 times, the predictions become much better, and don't miss the data like the red line does!

Depending on the data you are working with, you will need to try out different combinations of functions on the different variables. Statistical modeling is like an art form, and there's so much more to talk about that cannot be packed into one section of an introductory statistics class. Another method that you may learn about in future statistics courses that helps to deal with heteroskedasticity is *weighted least squares*, which can help reduce the impact of larger residuals on your model.

10.4 – Additional Practice

Example: A random sample of 95 sprinters was taken, and the following information was collected on them:

- **time**: The athlete's best 100m sprint time.
- **cal**: The athlete's average daily caloric intake.
- **height**: The athlete's height in inches.
- **hr**: The athlete's resting heart rate in beats per minute.
- **Train**: The athlete's average number of hours spent training weekly.

This data can be found in **runners.csv**. In the following questions, we will use linear models to predict the time of sprinters.

Fit a simple linear regression model that predicts time based on the average number of hours spent training weekly. Write out the model for this below.

Evaluate the assumptions for this model.

Interpret the slope term of this model.

What is the R^2 for this model? Interpret what this value means.

Create a 90% prediction interval for an athlete's 100m time that spends 30 hours a week training.