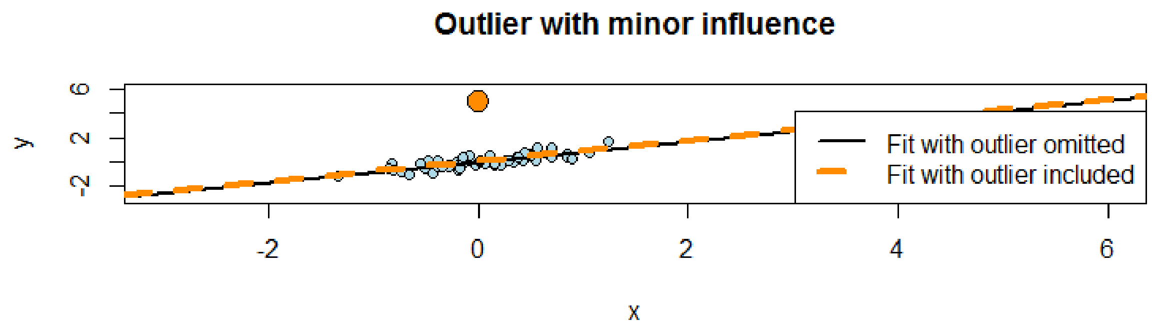
Exploration librairie swirl : Residual Diagnostics and variations

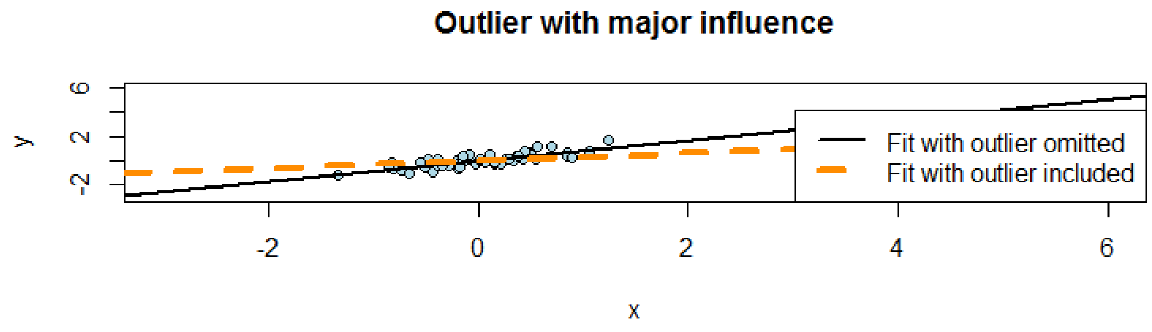
```
> swirl()

 Welcome to swirl! Please sign in. If you've been here before, use the same name as you did then. If you are new, call yourself something unique.

What shall I call you? jlbellier

 Would you like to continue with one of these lessons?

1: Regression Models Least Squares Estimation
2: No. Let me start something new.

Selection: 2

 Please choose a course, or type 0 to exit swirl.

1: Regression Models
2: Statistical Inference
3: Take me to the swirl course repository!

Selection: 1

 Please choose a lesson, or type 0 to return to course menu.

 1: Introduction                              2: Residuals
 3: Least Squares Estimation                  4: Residual Variation
 5: Introduction to Multivariable Regression  6: MultiVar Examples
 7: MultiVar Examples2                        8: MultiVar Examples3
 9: Residuals Diagnostics and Variation      10: Variance Inflation Factors
11: Overfitting and Underfitting             12: Binary Outcomes
13: Count Outcomes

Selection: 9
                                                                    0%

 Residuals, Diagnostics, and Variation. (Slides for this and other Data Science courses may be found  at github https://github.com/DataScienceSpecialization/courses. If you care to use them, they must  be downloaded as a zip file and viewed locally. This lesson corresponds to
 Regression_Models/02_04_residuals_variation_diagnostics.)
                                ===                                 3%
```

In the accompanying figure there is a fairly obvious outlier. However obvious, it does not affect the fit very much as can be seen by comparing the orange line with the black. The orange line represents a fit in which the outlier is included in the data set, and the black line represents a fit in which the outlier is excluded. Including this outlier does not change the fit very much, so it is said to lack influence.



====== ***6%***

This next figure also has a fairly obvious outlier, but in this case including the outlier changes the fit a great deal. The slope and the residuals of the orange line are very different than those of the black line. This outlier is said to be influential.



======== ***9%***

Outliers may or may not belong in the data. They may represent real events or they may be spurious. In any case, they should be examined. In order to spot them, R provides various diagnostic plots and measures of influence. In this lesson we'll illustrate their meanings and use. The basic technique is to examine the effects of leaving one sample out, as we did in comparing the black and orange lines above. We'll use the influential outlier to illustrate, since leaving it out has clear effects.

=========== ***12%***

The influential outlier is in a data frame named out2. It has two columns, labeled y and x, respectively. To begin, create a model named fit using fit <- lm(y ~ x, out2) or an equivalent expression.

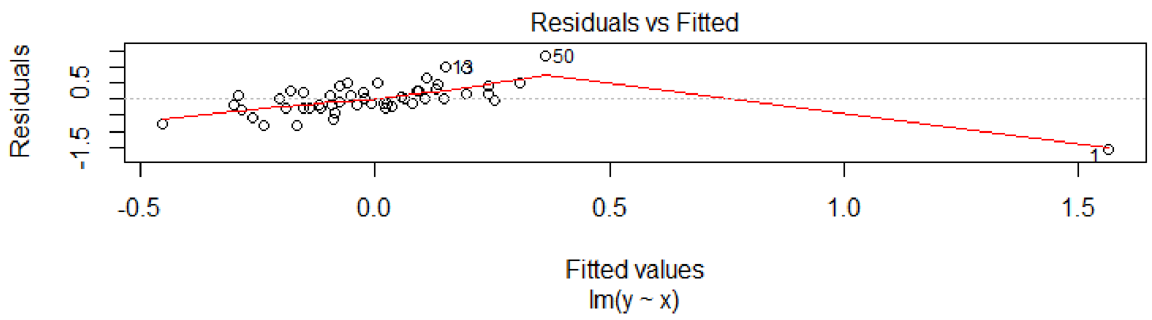
```
> fit <- lm(y~x,out2)
```

That's a job well done!

============== ***15%***

The simplest diagnostic plot displays residuals versus fitted values. Residuals should be *uncorrelated with the fit, independent and (almost) identically distributed with mean zero. Enter plot(fit, which=1) at the R prompt to see if this is the case.*

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



All that hard work is paying off!

================= **18%**

Do the residuals appear uncorrelated with the fit?

1: Yes
2: No. There is a linear pattern involving all but one residual and the fit.

Selection: ***2***

You are amazing!

==================== **21%**

The Residuals vs Fitted plot labels certain points with their row names or numbers, numbers in our case. Which of the three labeled points would you guess is our influential outlier?

1: 13
2: 50
3: 1

Selection: ***3***

You got it!

====================== **24%**

Our influential outlier is in row 1 of the data. To exclude it is just a matter using out2[-1,] rather than out2 as data. Create a second model, named fitno for 'fit with no outlier', which excludes the outlier.

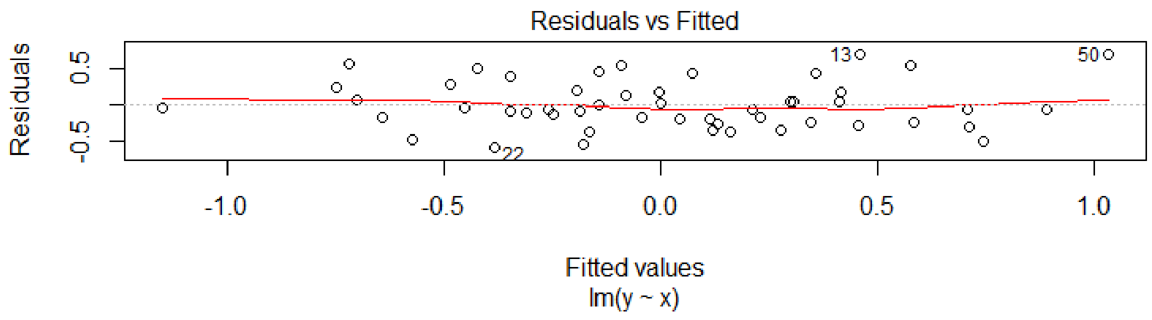
```
> fitno <- lm(y~x,out2[-1,])
```

You're the best!

========================= **27%**

Display a Residuals vs Fitted plot for fitno. Remember to use which=1.

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



That's the answer I was looking for.

============================== ***30%***

This plot has none of the patterned appearance of the first. It looks as we would expect if residuals were independently and (almost) identically distributed with zero mean, and were uncorrelated with the fit.

================================= ***33%***

The change which inclusion or exclusion of a sample induces in coefficents is a simple measure of its influence. Subtract coef(fitno) from coef(fit) to see the change induced by including the influential first sample.

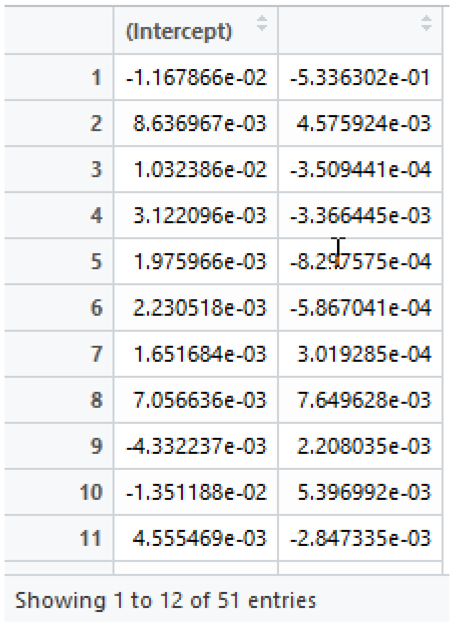
```
> coef(fit)-coef(fitno)
(Intercept)          x
-0.01167866 -0.53363019
```

Excellent work!

=================================== ***36%***

dfbeta: The function, dfbeta, does the equivalent calculation for every sample in the data. The first row of dfbeta(fit) should match the difference we've just calculated. The second row is a similar calculation for the second sample, and so on. Since dfbeta returns a large matrix, use either head(dfbeta(fit)) or View(dfbeta(fit)) to examine the result.

```
> View(dfbeta(fit))
```



	(Intercept)	
1	-1.167866e-02	-5.336302e-01
2	8.636967e-03	4.575924e-03
3	1.032386e-02	-3.509441e-04
4	3.122096e-03	-3.366445e-03
5	1.975966e-03	-8.297575e-04
6	2.230518e-03	-5.867041e-04
7	1.651684e-03	3.019285e-04
8	7.056636e-03	7.649628e-03
9	-4.332237e-03	2.208035e-03
10	-1.351188e-02	5.396992e-03
11	4.555469e-03	-2.847335e-03

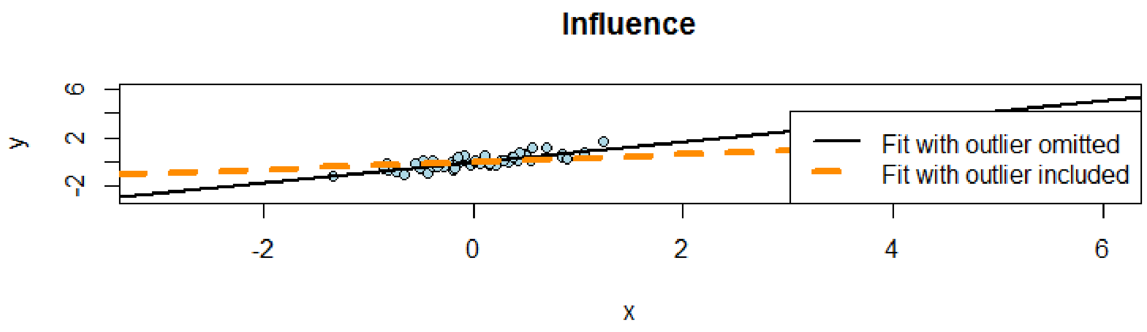
Showing 1 to 12 of 51 entries

You got it right!

===================================== ***39%***

Comparing the first row with those below it, we see that the first sample has a much larger effect on the slope (the x column) than other samples. In fact, the magnitude of its effect is about 100 times that of any other point. Its effect on the intercept is not very distinctive essentially because its y coordinate is 0, the mean of the other samples.

== ***42%***



When a sample is included in a model, it pulls the regression line closer to itself (orange line) than that of the model which excludes it (black line.) Its residual, the difference between its actual y value and that of a regression line, is thus smaller in magnitude when it is included (orange dots) than when it is omitted (black dots.) The ratio of these two residuals, orange to black, is therefore small in magnitude for an influential sample. For a sample which is not influential the ratio would be close to 1. Hence, 1 minus the ratio is a measure of influence, near 0 for points which are not influential, and near 1 for points which are.

=== ***45%***

This measure is sometimes called influence, sometimes leverage, and sometimes hat value. Since it is 1 minus the ratio of two residuals, to calculate it from scratch we must first obtain the two residuals. The ratio's numerator (orange dots) is the residual of the first sample of the model we called fit. The model fitno, which excludes this sample, also excludes its residual, so we will have to calculate its value. This is easily done. We use R's predict function *to calculate fitno's predicted value of y and subtract it from the actual value. Use the expression resno <- out2[1, "y"] - predict(fitno, out2[1,]) to do the calculation.*

*> **resno <- out2[1,"y"]-predict(fitno,out2[1,])***

Perseverance, that's the answer.

=== ***48%***

Now calculate the influence of our outlier using 1-resid(fit)[1]/resno or an equivalent expression.

*> **1-resid(fit)[1]/resno***

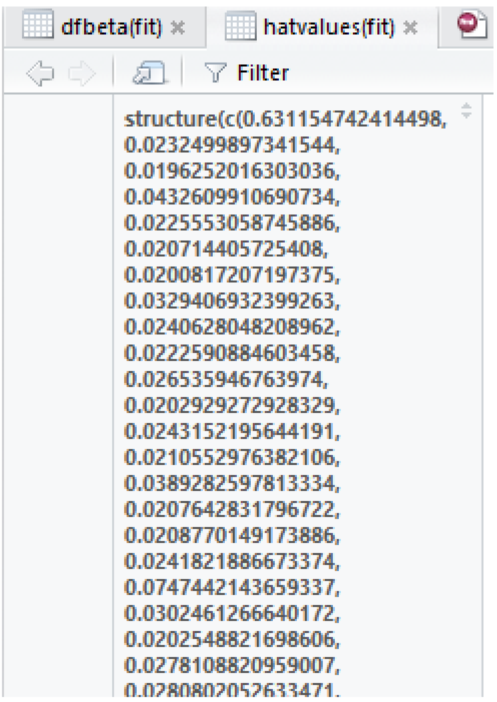
```
    1
0.6311547
```

Perseverance, that's the answer.

=== ***52%***

hatvalues: The function, hatvalues, performs for every sample a calculation equivalent to the one you've just done. Thus the first entry of hatvalues(fit) should match the value which you have just calculated. Since there are quite a few samples, use head(hatvalues(fit)) or View(hatvalues(fit)) to compare the influence measure of our outlier to that of some other samples.

*> **View(hatvalues(fit))***



```
structure(c(0.631154742414498,
0.0232499897341544,
0.0196252016303036,
0.0432609910690734,
0.0225553058745886,
0.020714405725408,
0.0200817207197375,
0.0329406932399263,
0.0240628048208962,
0.0222590884603458,
0.026535946763974,
0.0202929272928329,
0.0243152195644191,
0.0210552976382106,
0.0389282597813334,
0.0207642831796722,
0.0208770149173886,
0.0241821886673374,
0.0747442143659337,
0.0302461266640172,
0.0202548821698606,
0.0278108820959007,
0.0280802052633471,
```

Excellent work!

=== ***55%***

Residuals of individual samples are sometimes treated as having the same variance, which is estimated as the sample variance of the entire set of residuals. Theoretically, however, residuals of individual samples have different variances and these differences can become large in the presence of outliers. Standardized and Studentized residuals attempt to compensate for this effect in two slightly different ways. Both use hat values.

== **58%**
We'll consider standardized residuals first. To begin, calculate the sample standard deviation of fit's residual by dividing fit's deviance, i.e., its residual sum of squares, by the residual degrees of freedom and taking the square root. Store the result in a variable called sigma.

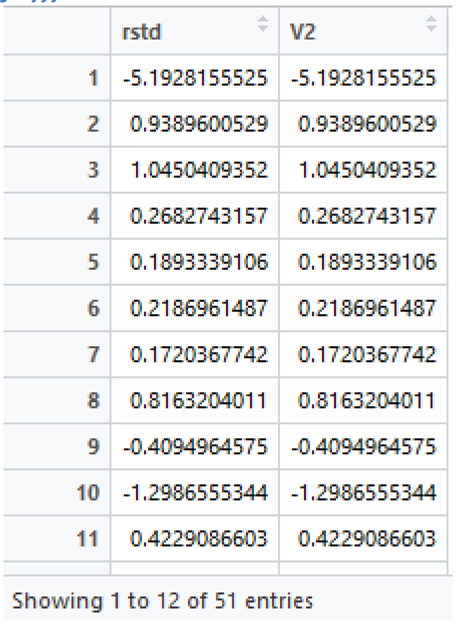
> **sigma <- sqrt(sum(resid(fit)^2)/fit$df)**

Great job!
=== **61%**
*Ordinarily we would just divide fit's residual (which has mean 0) by sigma. In the present case we multiply sigma times sqrt(1-hatvalues(fit)) to estimate standard deviations of individual samples. Thus, instead of dividing resid(fit) by sigma, we divide by sigma*sqrt(1-hatvalues(fit)). The result is called the standardized residual. Compute fit's standardized residual and store it in a variable named rstd.*

> **rstd <- resid(fit)/(sigma*sqrt(1-hatvalues(fit)))**

All that practice is paying off!
=== **64%**
rstandard: The function, rstandard, computes the standardized residual which we have just computed step by step. Use head(cbind(rstd, rstandard(fit))) or View(cbind(rstd, rstandard(fit))) to compare the two calculations.

> **View(cbind(rstd,rstandard(fit)))**

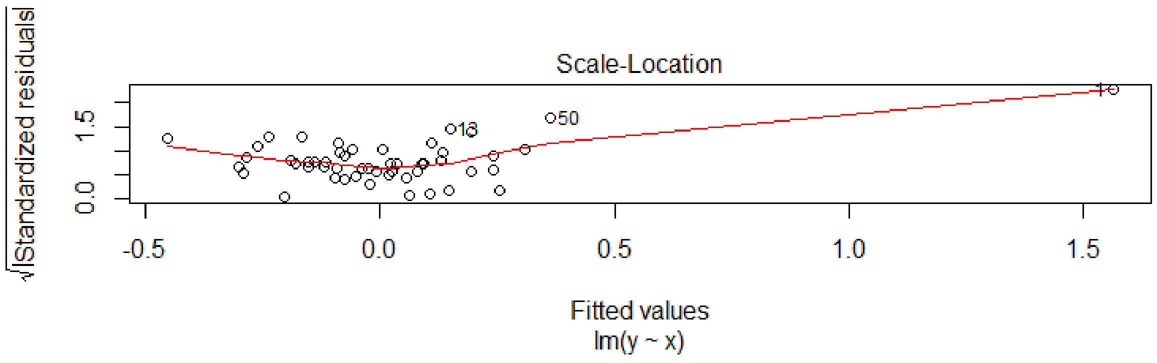


	rstd	V2
1	-5.1928155525	-5.1928155525
2	0.9389600529	0.9389600529
3	1.0450409352	1.0450409352
4	0.2682743157	0.2682743157
5	0.1893339106	0.1893339106
6	0.2186961487	0.2186961487
7	0.1720367742	0.1720367742
8	0.8163204011	0.8163204011
9	-0.4094964575	-0.4094964575
10	-1.2986555344	-1.2986555344
11	0.4229086603	0.4229086603

Showing 1 to 12 of 51 entries

All that hard work is paying off!
=== **67%**
A Scale-Location plot shows the square root of standardized residuals against fitted values. Use plot(fit, which=3) to display it.

> **plot(fit,which=3)**

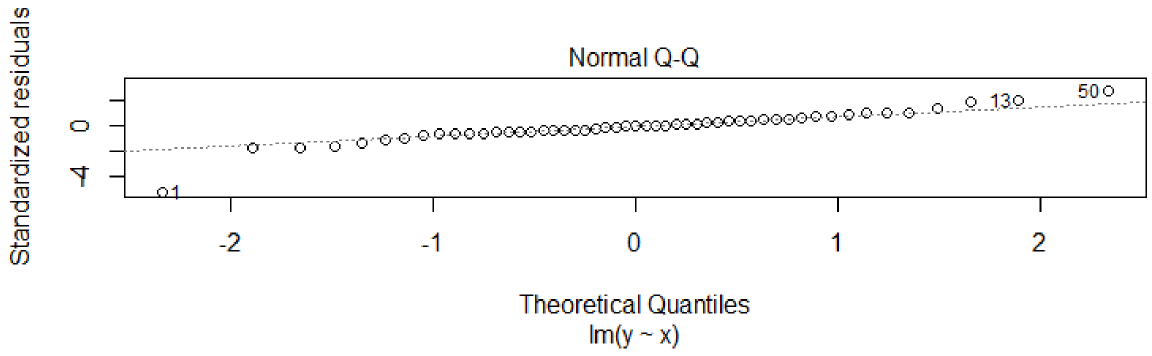


You are doing so well!

== 70%

Most of the diagnostic statistics under discussion were developed because of perceived shortcomings of other diagnostics and because their distributions under a null hypothesis could be characterized. The assumption that residuals are approximately normal is implicit in such characterizations. Since standardized residuals adjust for individual residual variances, a QQ plot of standardized residuals against normal with constant variance is of interest. Use plot(fit, which=2) to display this diagnostic plot.

> plot(fit,which=2)



Keep working like that and you'll get there!

=== 73%

Look at the outlier's standardized residual, labeled 1 on the Normal QQ plot. About how many standard deviations from the mean is it?

1: About -2
2: About -5

Selection: 2

All that hard work is paying off!

== 76%

Studentized residuals, (sometimes called externally Studentized residuals,) estimate the standard deviations of individual residuals using, in addition to individual hat values, the deviance of a model which leaves the associated sample out. We'll illustrate using the outlier. Recalling that the model we called fitno omits the outlier sample, calculate the *sample standard deviation of fitno's residual by dividing its deviance, by its residual degrees of freedom and taking the square root. Store the result in a variable called sigma1.*

```
> sigma1 <- sqrt(sum(resid(fitno)^2)/fitno$df)
```

Excellent job!

=== 79%

Calculate the Studentized residual for the outlier sample by dividing resid(fit)[1] by the product of sigma1 and sqrt(1-hatvalues(fit)[1]). There is no need to store this in a variable.

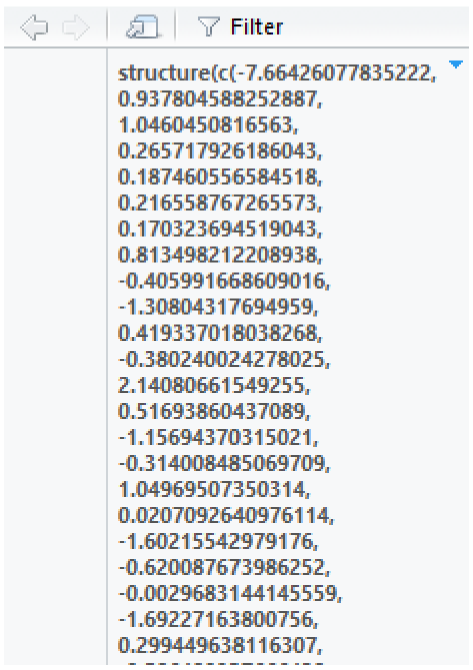
```
> resid(fit)[1]/(sigma1*sqrt(1-hatvalues(fit)[1]))
        1
-7.664261
```

Great job!

=== 82%

rstudent: The function, rstudent, calculates Studentized residuals for each sample using a procedure equivalent to that which we just used for the outlier. Thus rstudent(fit)[1] should match the value we calculated in the previous question. Use head(rstudent(fit)) or View(rstudent(fit)) to verify this and to compare the Studentized residual of the outlier with those of other samples.

```
> View(rstudent(fit))
```



```
structure(c(-7.66426077835222,
0.937804588252887,
1.0460450816563,
0.265717926186043,
0.187460556584518,
0.216558767265573,
0.170323694519043,
0.813498212208938,
-0.405991668609016,
-1.30804317694959,
0.419337018038268,
-0.380240024278025,
2.14080661549255,
0.51693860437089,
-1.15694370315021,
-0.314008485069709,
1.04969507350314,
0.0207092640976114,
-1.60215542979176,
-0.620087673986252,
-0.0029683144145559,
-1.69227163800756,
0.299449638116307,
```

Your dedication is inspiring!

=== 85%

Cook's distance is the last influence measure we will consider. It is essentially the sum of squared differences between values fitted with and without a particular sample. It is normalized (divided by) residual sample variance times the number of predictors which is 2 in our case (the intercept and x.) It essentially tells how much a given sample changes a model. We'll illustrate once again by calculating Cook's distance for the outlier.

=== 88%
We'll begin by calculating the difference in predicted values between fit and fitno, the models which respectively include and omit the outlier. This is most easily done by subtracting predict(fit, out2) from predict(fitno, out2). Store the difference in a variable named dy.

```
> dy <- predict(fitno,out2)-predict(fit,out2)
```

You are really on a roll!

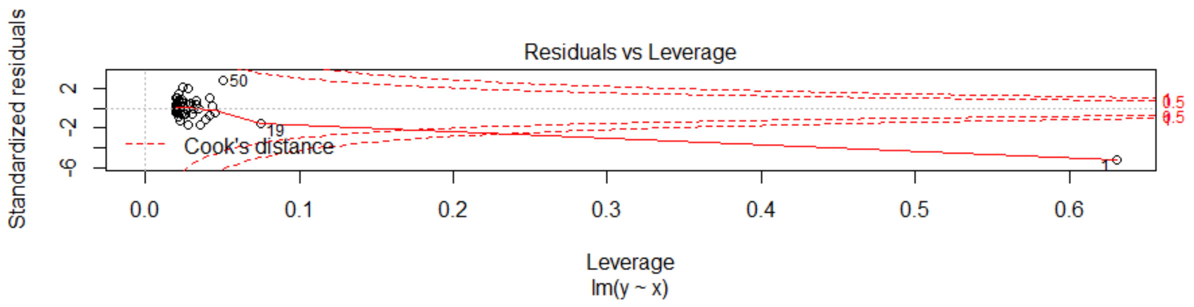
===91%
*Recall that we calculated the sample standard deviation of fit's residual, sigma, earlier. Divide the summed squares of dy by 2*sigma^2 to calculate the outlier's Cook's distance. There is no need to store the result in a variable.*

```
> sum(dy^2)/(2*sigma^2)
[1] 23.07105
```

Keep working like that and you'll get there!

== 94%
cooks.distance: The function, cooks.distance, will calculate Cook's distance for each sample. Rather than verify that cooks.distance(fit)[1] is equal to the value just calculated, because that sort of thing must be getting tedious by now, display a diagnostic plot which uses Cook's distance using plot(fit, which=5).

```
> plot(fit,which=5)
```



You are amazing!

== 97%
That concludes swirl's coverage of Residuals, Diagnostics, and Variation. The HTML5 slides for this as well as other units in the Johns Hopkins Data Science Specialization can be found here: https://github.com/DataScienceSpecialization/courses. They must be downloaded and viewed locally.

=== 100%