Likelihood

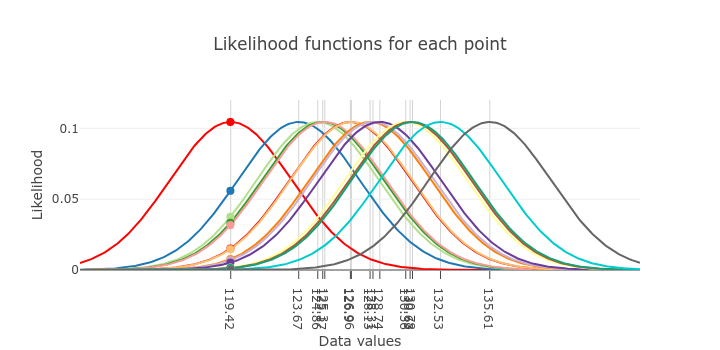
Key

Tue Apr 13 09:39:52 2021

## Maximum likelihood estimates of the mean

Work with the online app, and download images into the working folder as instructed. As long as you download the images with the correct file name they will show up in your knitted output. Answer the questions below based on what you find out in the app.

Calculate the likelihood of an estimate of the mean of 119.4, and download the image of the individual likelihoods (name it “mean\_119\_individual.png”) to your project folder, and it will show up below.

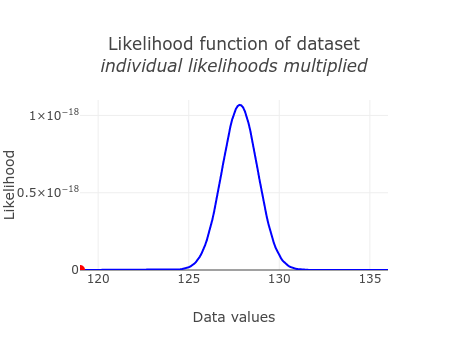


Individual likelihoods for mu = 119.4

**Question: this estimate of the mean very nearly maximizes the likelihod with respect to one of the data values. Which one?**

**Question: for which of these data values is the mean estimate of 119.4 least likely?**

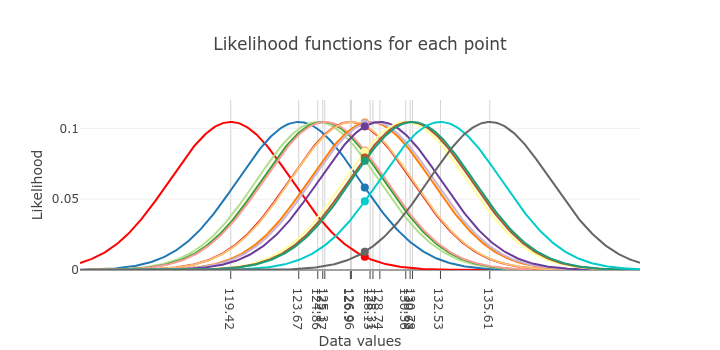
Download the graph on the right, of the likelihood function for the entire dataset (name it “mean\_119\_dataset.png”), and it will appear here:



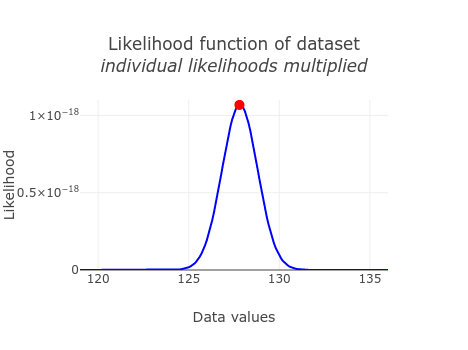
Dataset likelihoods for mu = 119.4

**Question: why is the likelihood of 119.4 so low?**

Now find the maximum likelihood estimate - slide the slider until the red dot on the dataset curve is at the peak. The individual likelihood plot for the maximum likelihood estimate looks like this:

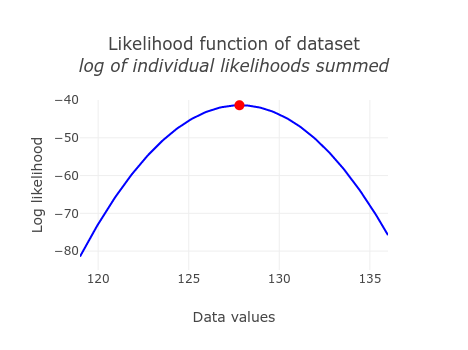
 **Question: is the maximum likelihood estimate for the mean right at the peak of any of the individual likelihoods? Is it way out in the tails for any of them?**

and the dataset plot for the maximum likelihood estimate looks like this:

 **Question: why is this the maximum likelihood estimate? Refer to the individual likelihoods - how do they combine to give this likelihood function? Was it necessary for the estimate of 127.8 to maximize the likelihood for any single point to be the maximum likelihood estimate (why or why not)?**

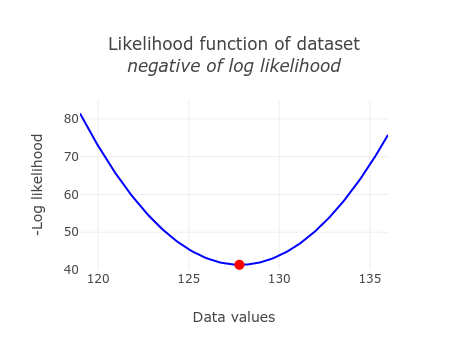
**Question: The likelihood function’s maximum is 1.06 x 10-8, which is a very small number (0.0000000106). Does this mean that the estimate is not very likely? Explain why the numeric value of the likelihood function isn’t interpreted, and is only used in comparison with other likelihoods.**

Change to the log likelihood, and download the dataset graph here:

 **Question: does putting the likelihoods on a log scale change the location of the maximum?**

**Question: quick double check… which is bigger, -10 or -20?**

Change to the negative log likelihood, and download the dataset graph here:

 **Question: to find the maximum likelihood estimate for the mean do you need to maximize the negative log likelihood function or minimize it?**

**Question: how is the negative log likelihood different from the log likelihood function? Is the shape, or the location of the maximum likelihood estimate, any different?**

Now that you know the maximum likelihood estimate of the mean, using your highly non-technical “numerical” method of sliding a slider around until a dot lands on the peak of the likelihood function, let’s see how it compares with the analytical formula.

Import the sheet “EstimateMean” from the likelihood\_data.xlsx spreadsheet you downloaded to your project folder (put it into an object called “mean.data”):

library(readxl)  
  
read\_excel("likelihood\_data.xlsx", sheet = "EstimateMean") -> mean.data

Calculate the mean of the Data column in mean.data:

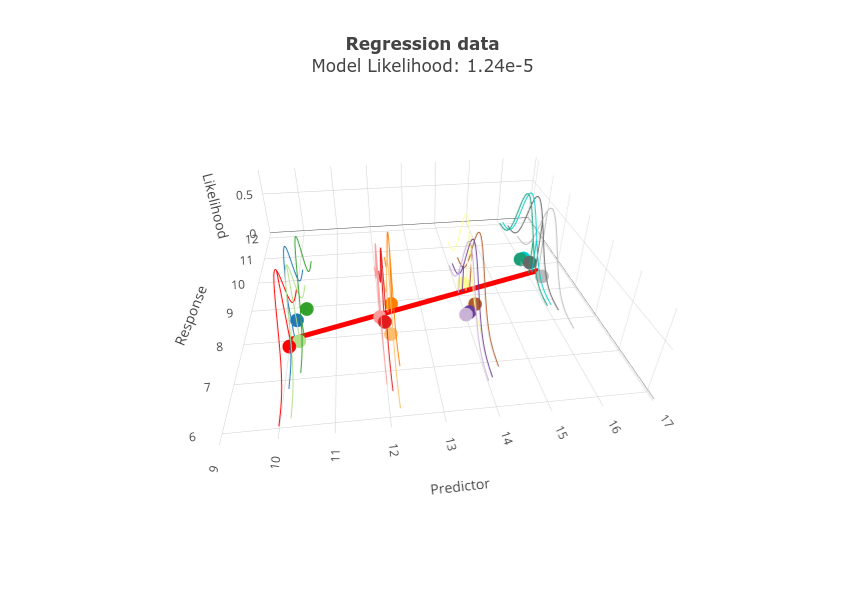
mean(mean.data$Data)

## [1] 127.8247

**Question: we only had the ability to select estimates for the mean down to one decimal place. To this level of precision, does the maximum likelihood estimate for the mean you got from the web app match the mean you calculated in R?**

## Likelihood of a linear regression model

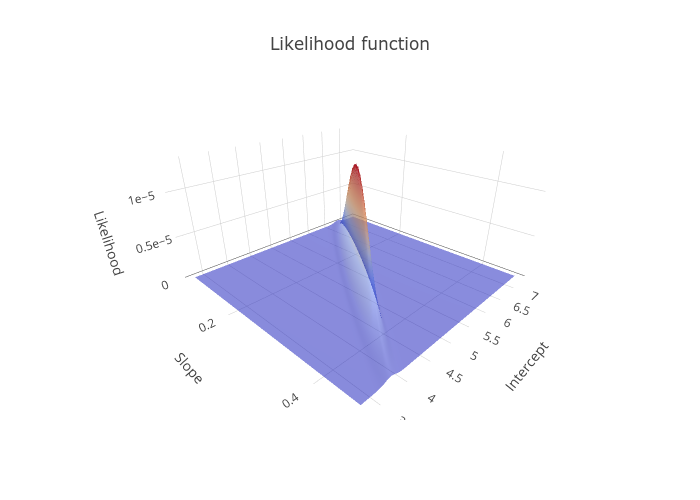
Download the graph of the best line you could get by changing the slope and intercept (call it regression\_ml.png)



Regression ml line

**Question: what slope and intercept did you use to get this maximum likelihood model? These are the maximum likelihood estimates of the slope and intercept.**

Download the graph of the likelihood function, and call it like\_surface.png.



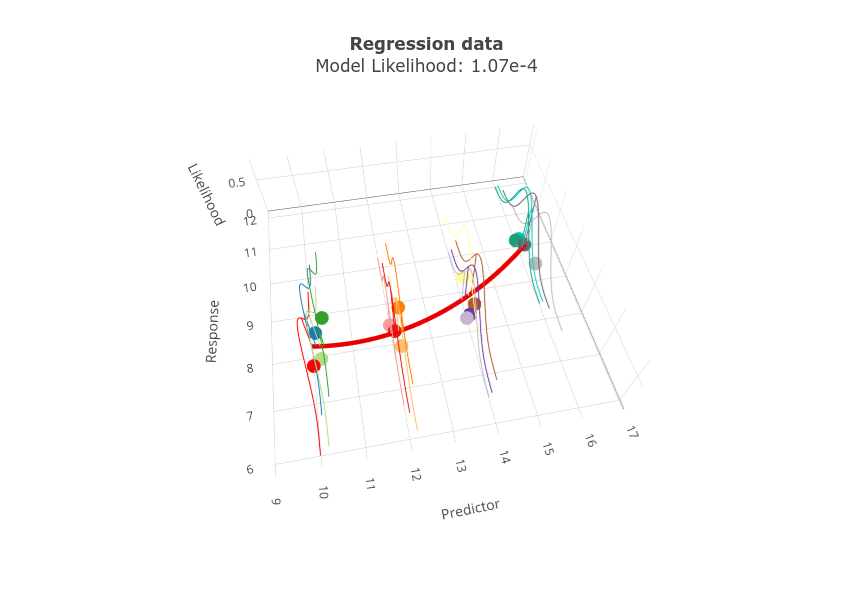
Regression likelihood surface

\*\*Question: is there still only one maximum for this likelihood function, now that \*we need to use a surface to display it?\*\*

**Question: as you were working your way to your maximum likelihood solutions you probably found what seemed like a pretty good estimate for one of the parameters (one that made the regression line thicker or redder), but it became bad when you changed the other one. How does the fact that this likelihood surface is shaped like a shark fin, instead of like a sombrero, explain this?**

## Likelihood of a regression with a quadratic term

Now find the maximum likelihood model that includes a quadratic term. Download the graph here (call it regression\_quad\_ml.png).



Regression quad ml line

**Question: is the likelihood for this quadratic model higher or lower than your simple linear regression model?**

**Question: given the definition of likelihood, what does a higher likelihood of a model tell you about which is better supported by the data (that is, which is more likely to have given rise to the data)?**

Since models are hypotheses about the structure of the data, we now have two hypotheses about the structure of this same set of response data values. We can compare these hypotheses with a likelihood ratio test.

**Question: to compare the models we need the log likelihoods for each, and the difference in degrees of freedom for each model. Report the log likelihoods for each model here. What is the difference in degrees of freedom between the models?**

**Question: when you enter the log likelihoods into the instruction page you will get a likelihood ratio test comparing them. Report the G statistic, degrees of freedom, and the p-value for the test. Did adding the quadratic term significantly increase the fit of the model to the data?**

## Using an intercept-only model as a null hypothesis

You can use the likelihood of an intercept-only model to represent the null hypothesis, and compare the log likelihood of the regression model (with both slope and intercept) against the intercept-only model (with only an intercept). Do this in the app in the instructions, and report the results here.

**Question: report the log likelihood of the intercept-only model, the G statistic, degrees of freedom, and the p-value for the test. Does the slope significantly improve the fit of the model to the data?**

## Repeat the analysis in R

We can do all of the analyses we did by hand using the apps in the instructions with the lm() functions instead. Import the Regression sheet from likelihood\_data.xlsx into a dataset called regression.

read\_excel("likelihood\_data.xlsx",sheet="Regression") -> regression

Fit a linear model of y (response) and x (predictor), and put it in an object called linear.lm. Print the linear.lm object so you can see the coefficients (just type the name linear.lm on the line after you fit the model):

lm(y ~ x, data = regression) -> linear.lm  
linear.lm

##   
## Call:  
## lm(formula = y ~ x, data = regression)  
##   
## Coefficients:  
## (Intercept) x   
## 5.3094 0.2668

**Question: did the coefficients from lm() match the estimates from the online app (to a decimal place or two)?**

Get the log likelihood of the linear.lm model:

logLik(linear.lm)

## 'log Lik.' -11.24156 (df=3)

**Question: are the degrees of freedom for the log likelihood the same as for the regression model itself? Where do the degrees of freedom for the log likelihood come from (that is, what are the three parameters that are estimated)? Note that one of the parameters estimated in R was not part of your work in the app, because I estimated it separately to simplify the example (read the explanation in the instructions to see what it is).**

**Question: log likelihoods are measures of model fit, like model R2. The model R2 for this model is 0.5992, which tells us that 59.92% of the variation in y is explained by the model. Does the numeric value of the log likelihood of -11.24 have an interpretable meaning the way that model R2 does?**

Fit a model with both a linear and a quadratic term, and put it into an object called quad.lm. Print the object so you can see the coefficients:

lm(y ~ x + I(x^2), data = regression) -> quad.lm  
quad.lm

##   
## Call:  
## lm(formula = y ~ x + I(x^2), data = regression)  
##   
## Coefficients:  
## (Intercept) x I(x^2)   
## 16.77218 -1.52861 0.06827

**Question: do these coefficients match the ones you got from the app (x is the slope, and I(x^2) is the quad term)?**

Get the log likelihood of the quad.lm model:

logLik(quad.lm)

## 'log Lik.' -8.187772 (df=4)

**Question: why does the quad model have one more degrees of freedom than the linear regression model?**

Now fit an intercept-only model, store it in an object called intercept.lm, and print it to display the coefficient:

lm(y ~ 1, data = regression) -> intercept.lm  
intercept.lm

##   
## Call:  
## lm(formula = y ~ 1, data = regression)  
##   
## Coefficients:  
## (Intercept)   
## 8.818

**Question: is the intercept the same as the mean of the y data?**

Get the log likelihood for the intercept only model:

logLik(intercept.lm)

## 'log Lik.' -18.55673 (df=2)

**Question: why would we want to fit a model with no predictors? What purpose will it serve?**

Now to do a likelihood ratio test of each of these models against each other we use lrtest() from the lmtest library, or anova() to compare the three models:

anova(intercept.lm, linear.lm, quad.lm, test = "Chisq")

## Analysis of Variance Table  
##   
## Model 1: y ~ 1  
## Model 2: y ~ x  
## Model 3: y ~ x + I(x^2)  
## Res.Df RSS Df Sum of Sq Pr(>Chi)   
## 1 15 9.5286   
## 2 14 3.8187 1 5.7100 9.498e-08 \*\*\*  
## 3 13 2.6070 1 1.2117 0.01397 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

lrtest(intercept.lm, linear.lm, quad.lm)

## Likelihood ratio test  
##   
## Model 1: y ~ 1  
## Model 2: y ~ x  
## Model 3: y ~ x + I(x^2)  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 2 -18.5567   
## 2 3 -11.2416 1 14.6303 0.0001308 \*\*\*  
## 3 4 -8.1878 1 6.1076 0.0134604 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Question: does the likelihood ratio test of the intercept only vs. linear regression, and linear regression vs. quadratic regression, match what we got with the app? Which is closer to the results we got from the app, the version from lrtest() or from anova()?**

**Question: likelihood ratio tests have to use the same response data, and have to have nested predictors, meaning that one model has a subset of the predictors that the other model has. This was true for both of our comparisons, of linear vs. quadratic regression, and of linear vs. intercept only models. Does this mean that the same estimates for the parameters have to be used? In other words, when you compare the linear regression to the intercept only model both have an intercept term, but did the intercept have to be the same for both models?**