Goals For This Week
We’ll do some nonlinear curve fitting to help us understand the affect of an anti cancer drug on WBC count. First we’ll use the loess function to just smooth the data. Then we’ll look at an actual nonlinear model and use nls to fit it to the data.

1 Getting an initial guess

Researchers are testing a new chemotherapy drug. The first phase of clinical trials (in humans) is to just test for side effects. One common effect is a drop in white blood cell count. The researchers want to know not only how much the WBC count will drop, but also how quickly and for how long.

To answer this they gave a small injection of the drug to a willing schmuck and then drew blood every 3 hours to test the WBC count. They then stored the data on my website. You can access it like this.

```r
> con <- url('http://math.utah.edu/~jmoore/1170content/lab11data.RData')
> load(con)
> C
```

The last command should have produced a giant list of numbers that all start with 25.

Now we make our list of T values. This represents samples every 3 hours for a week. Poor guy.

```r
>T <- seq(0, 7, 3/24)
```

Make a plot of this data: white blood cell count as a function of time(days).

2 Smoothing the data

These data look like they follow a definite pattern, but it is nonlinear. We can smooth the data using the `loess` function in R. This function is quite similar to `lm` and we use it in a similar way.

```r
> myfit1 <- loess(C ~ T)
```

We can access the smoothing values that `loess` generates like this

```r
> predict(myfit)
```

Now add a red line to your plot that shows the smoothed data. Where does this curve have a minimum (give a value for T and C). Where does the curve have a point of inflection? Record these values. What is the clinical significance of these two points.
3 Fitting to a model

The smoothed data is useful for visualization but doesn’t give any insight into the underlying process. If we suppose that the drug is cleared by the body at an exponential rate \( r \) and that white blood cells turnover at a rate \( w \) then we would expect

\[
C(G, r, w, K, t) = G + K(e^{-rt} - e^{-wt})
\]  

(1)

\( G \) and \( R \) are additional unknown parameters.

Write this as a function (called Cfunc).

Now we want to fit this model to our data. First we’ll need to make something called a data frame. This is a very useful R object. It’s essentially a matrix with different named columns.

\[
> \text{mydata=as.data.frame(c('C'=C,'T'=T))}
\]

To do a nonlinear fit we use another function with a short name: nls. In the first argument below, we specify the form of the equation. In the second argument is the data we want to fit.

\[
> \text{myfit2<-nls(C~Cfunc(G,r,w,K,T),data=mydata)}
\]

The above should have returned an error along with a suggestion to pick some starting values for our variables. This is because the nls algorithm works similarly to newton’s method. It needs an initial guess. If you don’t provide one, it’ll just make it up. Note that it guessed that all the parameters are set to 1. When this happens it generates something called a singular gradient. This is the equivalent to starting at spot where the derivative is zero. \textbf{Calculate} \( \frac{\partial C(G, r, w, K, T)}{\partial K} \) and then plug in 1 for all the parameters. \textbf{Can you see what the problem is? Choose some parameters to get rid of this problem.}

To input your initial guess into the nls function first write your parameters into a vector. Here is how you would do it if you wanted everything to be 1, which you don’t.

\[
> \text{myguess=c(G=1,r=1,w=1,K=1)}
> \text{myfit2<-nls(C~Cfunc(G,r,w,K,T),data=mydata,start=myguess)}
> \text{summary(myfit2)}
\]

The summary of myfit2 gives an estimate and an error for each parameter. \textbf{Record all these values.}

Now use ‘predict’ and ‘lines’ (just as for the loess fit) to put a blue curve on your plot. Your plot should now have the data points and two curves fitting the data. \textbf{Make sure it has appropriate labels and then save the plot.}

4 Your tasks

In addition to everything bolded above:

1. Where is the minimum point and the inflection point on your blue curve? Provide values for \( T \) and \( C \) for each point.

2. Calculate the derivative and 2nd derivative of \( C \) as a function of \( T \).

3. Plug in all of your parameters and check to make sure that your calculated minimum and inflection point are accurate.

4. What is the half life of the drug? Calculate this from your parameters.

5. What, biologically, does the value \( G \) tell you? Justify your answer using the formula for \( C \) and/or the graph.
6. What is the significance of $R$.

7. The lifespan of a WBC is $1/w$, what is this value?

8. Can you think of a clinical reason why you would want to look at the smoothed data?