Math 131/135/194, Fall 2004

Applied Calculus

Profs. Kaplan & Flath

Macalester College

Lab 2: Functions and Their Parameters

Goal of this lab

To understand some useful functions and combinations of them, and the role of the parameters that appear in each of them.

To hand in

Exponential Functions

The figure shows a few frames from a video of a bouncing ball.

Figure 1: Frames from a movie of a bouncing ball.

We want to figure out out how the bounce height varies from bounce to bounce. To do this, we have measured the height of the bouncing ball, on the first 10 bounces. The heights of the first 10 bounces were found to be.

`176, 131, 100, 70, 51, 34, 25, 19, 14, 8`

The units are "pixels" with respect to the digital video from which the heights were measured.

Make two variables, `h` and `n`, to store the heights and the bounce number (n). As we'll see later, there's an advantage to numbering n from 0 to 9 rather than from 1 to 10.

Plot the height versus the bounce number. You will make the plot in 3 different ways:

- An ordinary plot of `h` versus `n`.
- A semi-log plot of `h` versus `n`. To make this plot, you need to plot the log of `h` rather than `h` directly, as in `plot(n,log10(h))`.
- A semi-log plot where you ask R to use log axes in the conventional way. This is done with `plot(n,h,log='y')`. The named argument `log='y'` instructs R to set the y-axis (the vertical axis) as a log scale. On this scale, the y axis will have tick marks that are unevenly spaced. (To see the tick marks more easily, you can extend them into a grid by giving plot the named argument `tck=1`.)

To facilitate comparing the 3 plots, you may want to copy them to a word processing program where you can easily place them side by side.

The fact that the points lie (approximately) on a straight line in the semi-log plot indicates that the relationship between h and n is (approximately) exponential, of form

$$h_n = a * e^{kn}. \quad (1)$$

- Guess values for a and k. Note that if the first bounce is $n = 0$, then $h_0 = a$, so you can read a directly from the data set. To help in guessing k, recall that it governs the half-life or doubling time of the exponential. (For exponential decay, $k < 0$.)
- Graph the function with your guessed parameters over the data in the ordinary (non-log) h versus n format. If you are not satisfied that you made a good guess, try another guess. Keep modifying your guesses for the parameters until you feel you have a pretty good model for the data. When you feel happy with your values, tell us what they are, and produce a plot containing the points and the function's graph.
- Compare your final value for k to the slope of the data on the semi-log plot. Also, plot out your function and the data on semi-log axis.
- The maximum velocity of the ball, just before it hits the ground on each bounce, will be proportional to the square root of the height the ball started from. Does the ball's maximum velocity, v_n, follow an exponential pattern?

Logistic Functions

Consider the function

$$f(t) = \frac{A}{1 + e^{-(b+mt)}} \tag{2}$$

called a *logistic* function. Figure 2 shows graphs of two different logistic functions, that is, the logistic function with two different sets of parameters A, b, and m.

Both logistic functions have been set so that the values range from 0 to 30. These two values are the *asymptotic values* that the function achieves as $t \to -\infty$ and as $t \to \infty$. That is, $\lim_{t\to-\infty} f(t) = 0$ and $\lim_{t\to\infty} f(t) = 30$ for both of the functions. The parameter A sets the asymptotic values: $A = \lim_{t\to\infty} f(t)$ while $\lim_{t\to-\infty} f(t)$ is always zero.

Both of the plotted logistic functions have parameters so that the time that the function crosses the half-way value at the same time, at $t = 250$. We will denote this half-way time $T_{\text{half-way}}$. In terms of the parameters, $T_{\text{half-way}}$ is related to m and b, but not to A. The relationship is $b + mT_{\text{half-way}} = 0$.

What's different about the two plotted functions is how steep they are when they cross the half-way point. This steepness is described by the parameter m.

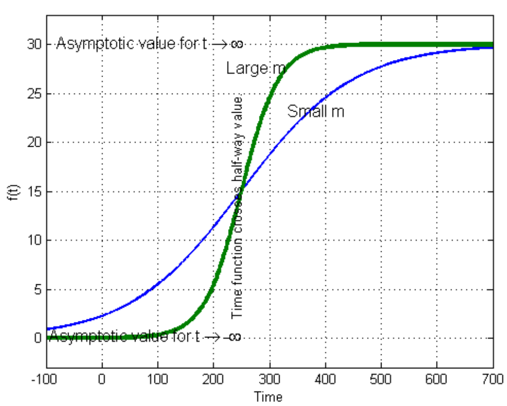


Figure 2: Two members of the family of logistic functions. For both of the functions, the parameter $A = 30$. The parameters b and m have been set so that both functions reach a value of $A/2$ at time $t = 250$. The function with small m crosses this midpoint slowly; the function with large m crosses the midpoint steeply.

The logistic function was introduced by the Belgian mathematician and social scientist Pierre François Verhulst around 1846 to describe the growth of populations with limited food resources. A particular example he used was the data from the US Census from 1790 to 1840. Total US population, in millions, is shown in the following table for every federal census that has been carried out since the US Constitution was established:

Year	Pop.	Year	Pop.	Year	Pop.
1790	3.929	1800	5.308	1810	7.240
1820	9.638	1830	12.866	1840	17.069
Data available to Verhulst in 1846					
1850	23.192	1860	31.443	1870	38.558
1880	50.156	1890	62.948	1900	75.996
1910	91.972	1920	105.711	1930	122.775
1940	131.669				
Additional data for the problem.					
1950	150.697	1960	179.323	1970	203.185
1980	226.546	1990	248.710	2000	281.421
Remaining data to the present.					

- Make a plot of the above data, population versus time, for the years 1790 through 1940. It will help if you define two variables, `pop` and `t`, with `t` containing the year.

- Whether or not the data through 1940 indicate a leveling off is a matter of judgement. Make your best guess of what the asymptotic value of the population is, judging just from the data through 1940. This value will be the parameter A.

- Given your value for A, you should be able to say what year $T_{\text{half-way}}$ the population graph crosses the half-way point. This value doesn't fix any single parameter, but it does impose the relationship between b and m that was described above: $b + mT_{\text{half-way}} = 0$.

- Pick a trial value of m, let's say $m = 0.01$. Given your value of $T_{\text{half-way}}$, calculate the corresponding value of b. Then plot the logistic function with your values of A, m, and b on top of the census data from 1790 to 1940.

- If you aren't satisfied with the quality of fit of the function to the data, modify m, recalculate b, and replot the function. In modifying m, remember that m controls the steepness of the function at the half-way point. Please note that $f(t)$ is just a model, and it may not be a very close fit to the census data, but do the best you can.

- Compare the data to the fitted $f(t)$ through the year 2000. How does the model based on data up to 1940 do for later data?

Sine Functions

Sparing no expense to aid in your education, your professors have procured the services of an experienced whistler, who has recorded a scale for us. This scale is recorded in the file `whistle-scale.wav`, which you can play with any suitable software. (Since there are no speakers on the lab computers, we will play the sound in class.)

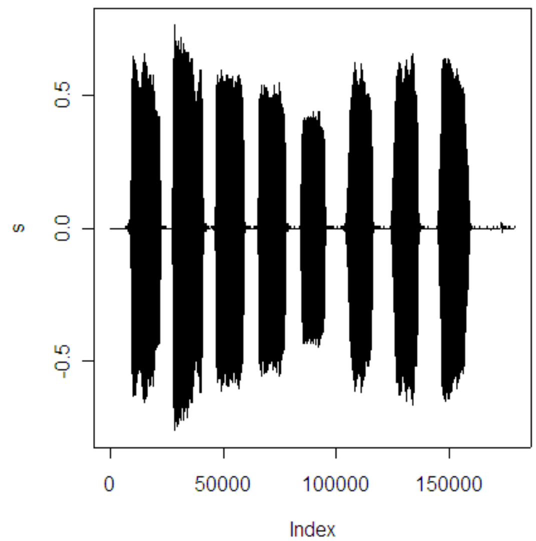


Figure 3: Sound amplitude $s(t)$ versus time (in samples) for a whistled musical scale.

Physically, sound corresponds to fluctuations in air pressure. We can think of a sound as a function of time, $s(t)$. Technologically, sound is often recorded digitally, which means that the function has been *sampled* at many discrete moments in time. In the case of `whistle-scale.wav`, there are 22,050 samples every second; each sample is a number giving the instantaneous value of $s(t)$ at the time the sample was taken. Figure 3 shows a plot of the sound function $s(t)$:

To make this plot, we read the sound file into R and plotted it in the normal way, instructing `plot` to draw lines between the samples. Before you can do this, however, you need to tell R which directory on the computer to find the files in. To do this, use the FILE/CHANGE DIR menu item, then select browse and navigate the file system in the familiar way. Make sure to press the "OK" button when you have located the right directory.

The following commands will read in the sound file, storing the samples in a variable `s`. Next, to satisfy our curiousity, we ask how many samples there are. Finally, we plot out `s`, using the `type='l'` argument to `plot` in order to instruct `plot` to connect the data points with lines rather than plotting out a symbol at each data point.

```
> s = scan('whistle-scale.dat')
> length(s)
[1] 179080
> plot(s,type='l')
```

It's hard to tell typographically, but the `'l'` is a lower-case L in quotes. Note that since only one variable, namely `s`, was given to plot, the `plot` function plots each of the values on the y-axis, arranging the x-axis to show the sample number, or "index" of each data point.

The eight notes in the musical scale are the visible large "chunks." At this size of plot, it's impossible to see that this is a single-value function of time. To see the details of the function, we'll zoom in on part of the function, plotting out 200 samples starting at sample 50,000. This is a segment of the sound function within the 3rd note of the musical scale.

```
> plot(s[50000:50200],type='l')
```

The stickler for accuracy will note that there are 201 samples being plotted. The R-language notation `s[50000:50200]` means, "Pick samples 50000 through 50200 from the vector `s`." Note the use of the square brackets, `[` and `]`.

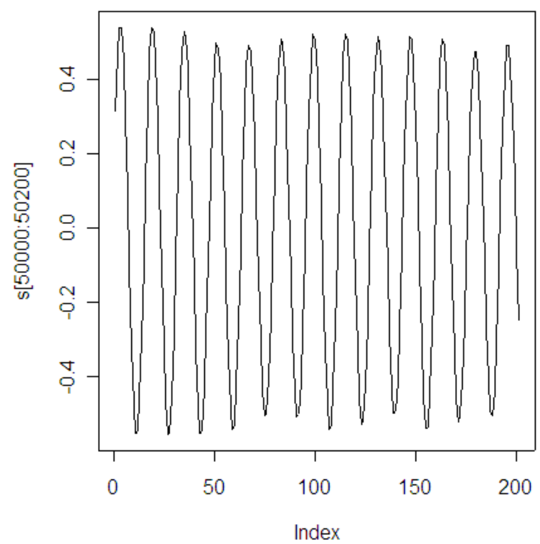


Figure 4: Sound amplitude $s(t)$ versus time for 200 consecutive samples of the third note in the scale shown in Fig. 3.

From the graph, it may seem reasonable to model the 3rd note as a sine wave. You can read the amplitude of the sine wave straight off of the graph. The period of the sine wave can be estimated by counting how many cycles of the wave there are in the 201 samples. Just divide 200 into the number of cycles to get the length of a typical cycle in units of samples. (Here's where the stickler for accuracy gets what's deserved: even though there are 201 samples, there are only 200 intervals between samples, so it's legitimate to divide the number of cycles in the graph by 200, not 201.)

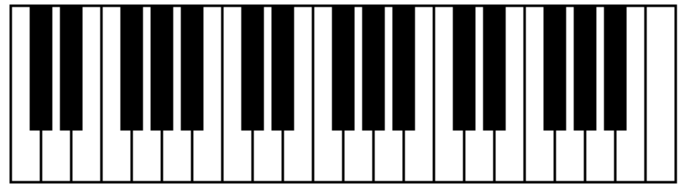
We aren't usually too interested in measuring periods in units of samples. You will want to convert the cycle length to seconds. To perform the conversion, recall that in this recording there are 22050 samples per second.

[1] OK, 201 samples. You stickler!

- Make a 200-sample plot[1] for each of the 8 notes in the scale. To make each plot, you will have to read off from the first plot above a reasonable starting point within each note. It doesn't matter much exactly where the starting point is, so long as it is somewhere within the note you want to plot. For example, sample 50,000 is within the 3rd note. From each detailed plot, estimate the period of the sine wave that models the note.

- Once you have recorded the period of each note, construct two vectors, `period` to contain the period in seconds and `num` containing the numbers 1 to 8. Plot out `period` versus `num`. Use semi-log and log-log plots to try to figure out what is the functional relationship between `period` and `num`. Explain what you found.

If you know something about music, you may realize that the notes in a scale are not evenly spaced. Note that one simple scale is C,D,E,F,G,A,B,C. Numbering the C key on a piano as 0 and counting all the keys — black and white — note that D is key 2, E is 4, F is 5, and so on.



- Create a variable `notenum` that contains the note numbers with proper piano spacing. What model best describes the observed relationship between period and `notenum`: linear, exponential or power-law?

It will be a bit hard to see the true musical structure of the scale because the whistler is not very good. If there is a student with better whistling abilities who will volunteer, we will record a proper scale and see the proper relationship.