

# Computational Physics - Problem Set 8

Frederik Holst Knudsen

November 1, 2024

Github URL: <https://github.com/frederikholst/phys-ga2000>

## 1 Maximization Likelihood Problem

We want to maximize the Likelihood function:

$$L = \prod_{i=1}^N p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

where the  $y_i$ 's are the answers of the survey and  $p(x_i)$  is the PDF given in the problem:

$$p(x) = \frac{1}{1 + \exp[-(\beta_0 - \beta_1 x)]}$$

. To avoid large numbers, we minimize the negative log likelihood instead of Maximization:

$$\mathcal{L} = - \sum_{i=1}^N y_i \log(p(x_i)) - (1 - y_i) \log(1 - p(x_i))$$

. This way we get the fit shown in Figure 1, with  $\beta_0 = -5$  and  $\beta_1 = 0.09830074$ .

The covariance matrix is found to be:

$$\begin{bmatrix} 8.6736393 \times 10^{-1} & -1.6442016 \times 10^{-2} \\ -1.6442016 \times 10^{-2} & 3.4502562 \times 10^{-4} \end{bmatrix}$$

and the standard deviation is:

$$STD_{\beta_0} = 0.931323766708374$$

$$STD_{\beta_1} = 0.01857486553490162$$

.

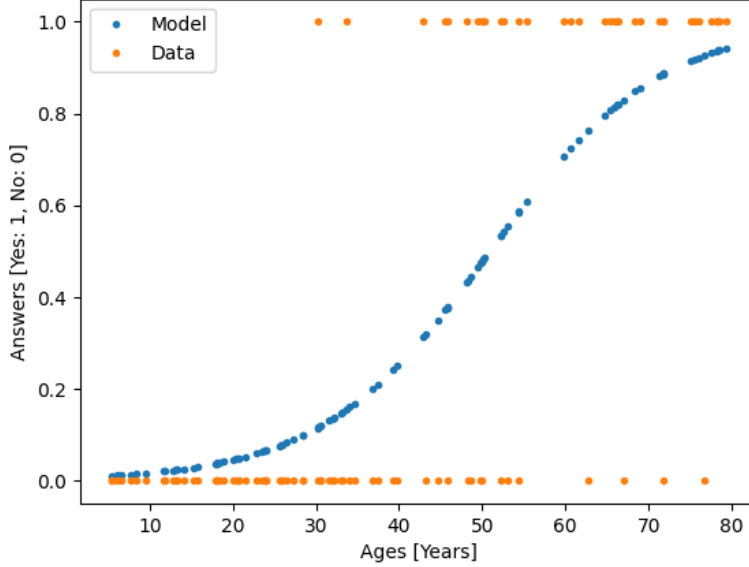


Figure 1: The logistic regression function  $p(x)$  fitted to the data of the survey with the parameters,  $\beta_0 = -5$  and  $\beta_1 = 0.09830074$ , which was found using Likelihood Maximization.

## 2 Newman 7.3: FFT of Trumpet and Piano

In Figure 2 we see the spectrum comparison of the piano and the trumpet playing the same note. The overtones of the trumpet is more prominent, giving rise the more sharper sounding instrument than that of the piano. Since the overtones and the fundamental tone overlap, the two instruments are, however, playing the same note.

To rescale the axis, we use that the sampling rate is 44100 samples per second. Scaling with the ratio of the sampling rate and the sample size, we obtain the units of frequency (1/s), see Figure 3. We find the index at the maximum of the spectra, and find the fundamental frequency, to be 525.231 Hz, so very likely the instruments played the fifth C (C5), an octave above the middle C.

## 3 Newman 7.4

PART A: We load the dow-file and plot the data as seen in Figure 4.

PART B: We calculate the coefficients of the discrete Fourier transform of the data using the numpy function `np.fft.rfft`.

PART C: We now set the first 10 percent of the elements of the coefficients to zero.

PARCT D: We carry out the inverse Fourier transform. See Figure 5

PARCT E: Same as before, but now with an even stronger low pass, so it's only the first 2 percent of the coefficients that are nonzero. See Figure 6.

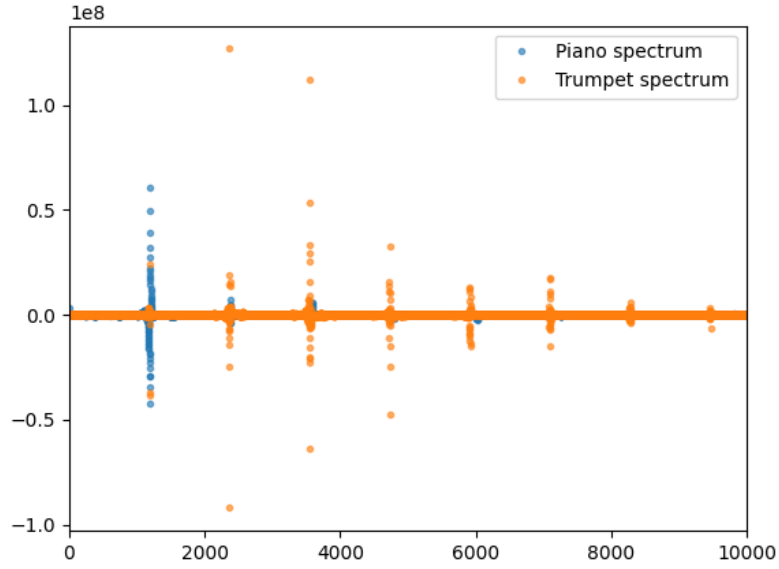


Figure 2: Here we see the side-by-side comparison of the spectrum of the two instruments. One notices the sharper overtones from the trumpet.

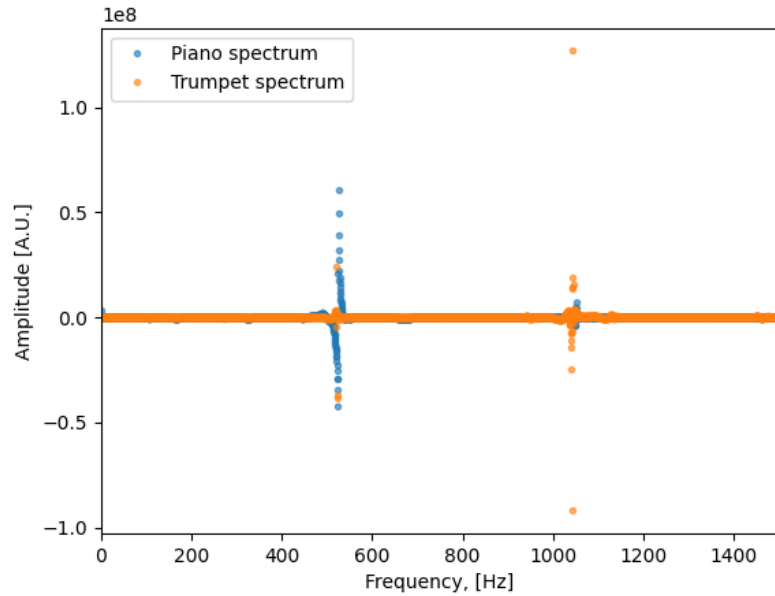


Figure 3: The fundamental tone and first overtone of the piano and trumpet with scaled axis in units of Hz.

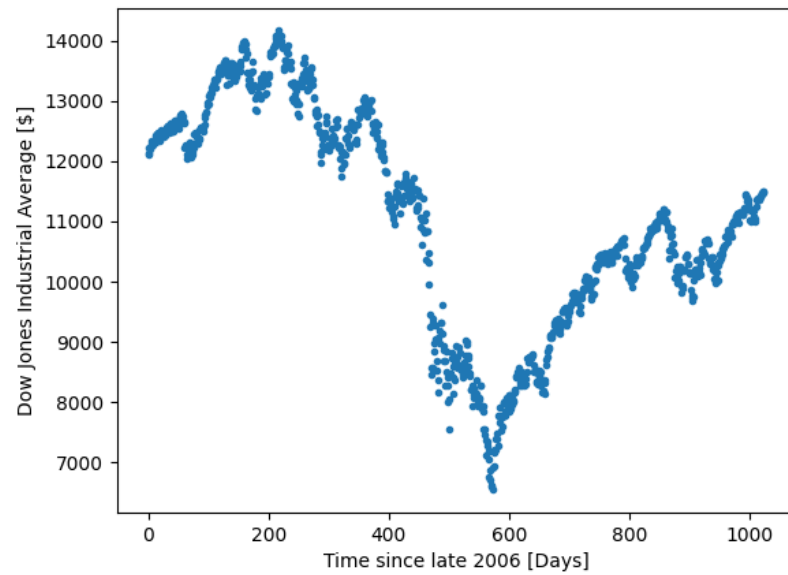


Figure 4: The Dow Jones Average from 2006 to 2010.

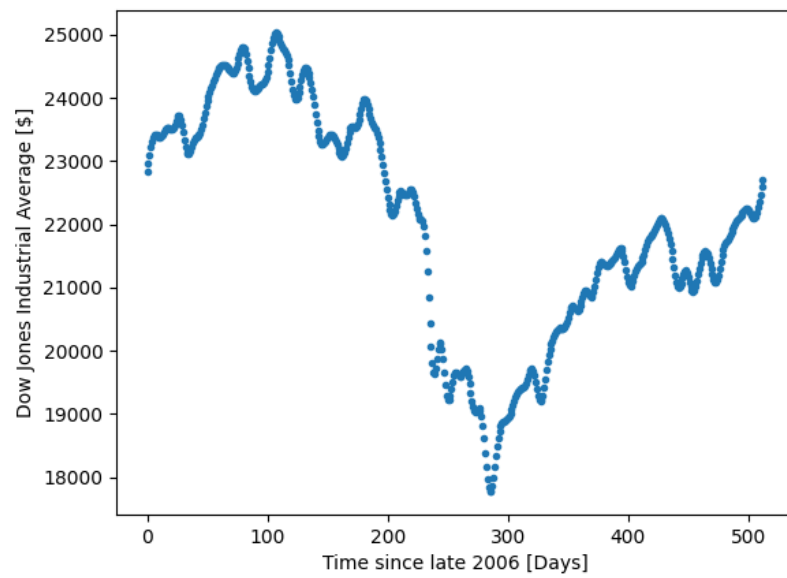


Figure 5: The Dow Jones Average from 2006 to 2010, but now with a low pass filter, by only letting the first 10 percent of the Fourier coefficients be nonzero. This is effectively removing the high oscillating effects on the graph.

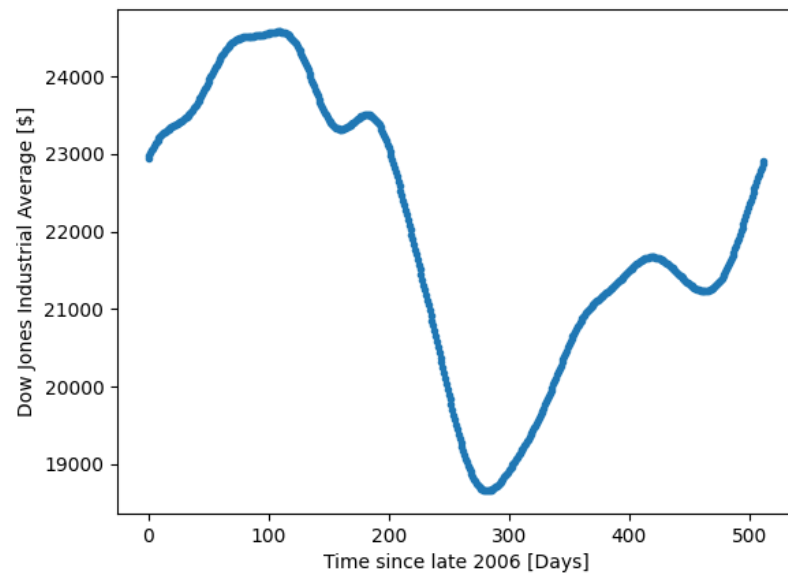


Figure 6: The Dow Jones Average from 2006 to 2010, but now with a stronger low pass filter, by only letting the lowest 2 percent of the Fourier coefficients be nonzero.