# Lecture 1: Data, Lists and Models

Information is described by **data**. While data entries are often not numerical values initially, they can always be encoded in a numerical manner so that we can look at **numerical data**. Here are some examples: stock data, weather data, networks, vocabularies, address books, encyclopedias, surfaces, books, the human genome, pictures, movies or music pieces. Is there any information which is not described by data?

While data describe complicated objects, they can be organized in **lists**, lists of lists, or lists of lists of lists etc. Stock market or weather data can be represented as finite sequences of numbers, pictures are arrays of numbers, a movie is an array of pictures. A book is a list of letters. A graph is given by a list of nodes and connections between these nodes. A movie is a list of pictures and each picture is a list of list of pixels and each pixel is a list of red, green and blue values. If you think about these examples, you realized that **vectors**, **matrices** ore higher dimensional arrays are helpful to organize the data. We will define these concepts later but a vector is just a finite list of things and a matrix is a list of lists, a **spreadsheet**. In the case of pictures, or music pieces the translation into an array is obvious. Is it always possible to encode data as sequences of numbers or arrays of numbers or lists of arrays of numbers etc? Even for complicated objects like networks, one can use lists. A network can be encoded with an array of numbers, where we put a 1 at the node $(i, j)$ if node $i$ and $j$ are connected and 0 otherwise. This example makes clear that we need a **mathematical language** to describe data. It looks like a difficult problem at first because data can appear in such different shapes and forms. It turns out that we can use vectors to describe data and use matrices to describe relations between these data. The fact that most popular databases are **relational databases** organized with tables vindicates this point of view. In this first lecture, we also want to see that linear algebra is a tool to organize and work with data. Even data manipulation can be described using linear algebra. Data of the same type can be added, scaled. We can mix for example two pictures to get a new picture. Already on a fundamental level, nature takes linear algebra seriously: while classical mechanics deals with differential equations which are in general nonlinear and complicated, quantum mechanics replaces this with a linear evolution of functions. Both in the classical and quantum world, we can describe the evolution of observables with linear laws.

Here are four important uses of linear algebra to describe data:

| Tool | Goal | Using | Example |
|---|---|---|---|
| Databases | Describe the data | Lists | Relational database, Adjacency matrix |
| Modeling | Model the data | Probability | Markov process, Filtering, Smoothing |
| Fitting | Reduce the data | Projections | Linear regression. |
| Computation | Manipulate the data | Algebra | Fourier theory. |

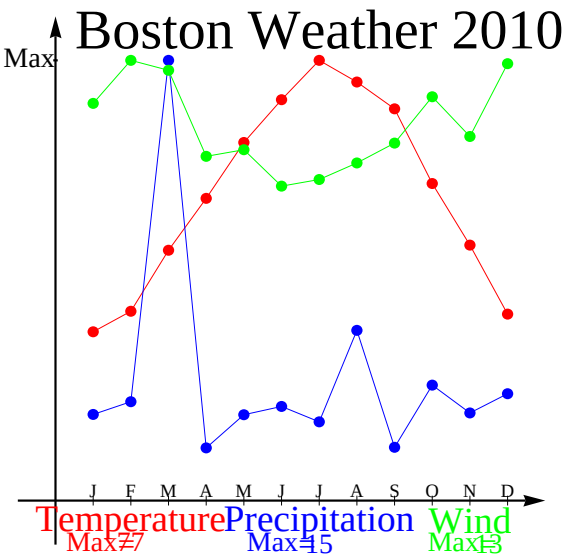
We have seen that a fundamental tool to organize data is the concept of a **list**. Mathematicians call this a **vector**. Since data can be added and scaled, data can be treated as vectors. We can also look at lists of lists. These are called **matrices**. Matrices are important because they allow to describe **relations** and **operations**. Given a matrix, we can access the data using coordinates. The entry $(3, 4)$ for example is the forth element in the third row. Having data organized in lists, one can manipulate them more easily. One can use **arithmetic** on entire lists or arrays. This is what spreadsheets do.

Observed quantities are functions defined on lists. One calls them also **random variables**. Because observables can be added or subtracted, they can be treated as vectors. If the data themselves are not numbers like the strings of a DNA, we can add and multiply numerical functions on these data. The function $X(x)$ for example could count the number of $A$ terms in a genome sequence $x$ with letters A,G,C,T which abbreviate Adenin, Guanin, Cytosin and Tymin. It is a fundamental and pretty modern insight that all mathematics can be described using algebras and operators. We have mentioned that data are often related and organized in relational form and that an array of data achieves this. Lets look at weather data accessed from http://www.nws.noaa.gov on January 4'th 2011, where one of the row coordinates is "time". The different data vectors are listed side by side and listed in form of a **matrix**.

| Month | Year | Temperature | Precipitation | Wind |
|---|---|---|---|---|
| 01 | 2010 | 29.6 | 2.91 | 12.0 |
| 02 | 2010 | 33.2 | 3.34 | 13.3 |
| 03 | 2010 | 43.9 | 14.87 | 13.0 |
| 04 | 2010 | 53.0 | 1.78 | 10.4 |
| 05 | 2010 | 62.8 | 2.90 | 10.6 |
| 06 | 2010 | 70.3 | 3.18 | 9.5 |
| 07 | 2010 | 77.2 | 2.66 | 9.7 |
| 08 | 2010 | 73.4 | 5.75 | 10.2 |
| 09 | 2010 | 68.7 | 1.80 | 10.8 |
| 10 | 2010 | 55.6 | 3.90 | 12.2 |
| 11 | 2010 | 44.8 | 2.96 | 11.0 |
| 12 | 2010 | 32.7 | 3.61 | 13.2 |

To illustrate how linear algebra enters, lets add up all the rows and divide by the number of rows. This is called the **average**. We can also look at the average squre distance to the mean, which is the variance. Its square root is called the **standard deviation**.

| Month | Year | Temperature | Precipitation | Wind |
|---|---|---|---|---|
| 6.5 | 2010 | 53.7667 | 4.13833 | 11.325 |

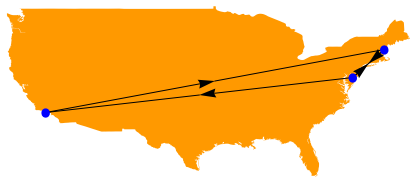


In the example data set, the average day was June 15th, 2010, the average temperature was 54 degrees Fahrenheit = 12 degrees Celsius, with an average of 4 inches of precipitation per month and an average wind speed of 11 miles per hour. The following figure visualizes the data. You see the unusually rainy March which had produced quite a bit of flooding. The rest is not so surprising. The temperatures are of course higher in summer than in winter and there is more wind in the cooler months. Since data often come with noise, one can simplify their model and reduce the amount of information to describe them. When looking at a particular precipitation data in Boston over a year, we have a lot of information which is not interesting. More interesting is the global trend, the deviation from this global trend and correlations within neighboring days. It is for example more likely to be rainy near a rainy day than near a sunny day. The data are not given by a random process like a dice. It we can identify the part of the data which are randomly chosen, how do we find the rule? This is a basic problem and can often be approached using linear algebra. The expectation will tell the most likely value of the data and the standard deviation tells us how noisy the data are. Already these notions have relations with geometry.
How do we model from a situation given only one data set? For example, given the DJI data, we would like to have a good model which predicts the nature of the process in the future. We can do this by looking for trends. Mathematically, this is done by data fitting and will be discussed extensively in this course. An other task is to model the process using a linear algebra model. Maybe it is the case that near a red pixel in a picture it is more likely to have a red pixel again or that after a gain in the DJI, we are more likely to gain more.

Lets illustrate how we can use lists of data to encode a traffic situation. Assume an airline services the towns of Boston, New York and Los Angeles. It flies from New York to Los Angeles, from Los Angeles to Boston and from Boston to New York as well as from New York to Boston. How can we can compute the number of round trips of any length $n$ in this network? Linea algebra helps: define the $3 \times 3$ **connection matrix** $A$ given below and compute the $n$'th power of the matrix. We will learn how to do that next week. In our case, there are 670976837021 different round trips of length 100 starting from Boston.

The matrix which encodes the situation is the following:

$$A = \begin{bmatrix} & BO & NY & LA \\ BO & 0 & 1 & 0 \\ NY & 1 & 0 & 1 \\ LA & 1 & 0 & 0 \end{bmatrix} .$$



To summarize, linear algebra enters in many different ways into data analysis. Lists and lists of lists are fundamental ways to represent data. They will be called vectors and matrices. Linear algebra is needed to find good models or to reduce data. Finally, even if we have a model, we want to do computations efficiently.

## Homework due February 2, 2011

**1** In the movie "Fermat's room", some mathematicians are trapped in a giant press and have to solve mathematical problems to stop the press from crushing them to death. In one of the math problems, they get the data stream of $169 = 13^2$ digits:
0000000000000001111111110001111111111100111111111100110
0010001100110001000110011111011111001111000111100011111111
1100000101010100000011010110000001111111000000000000000. Can you solve the riddle? Try to solve it yourself at first. If you need a hint, watch the clip
http://www.math.harvard.edu/ knill/mathmovies/text/fermat/riddle3.html

**2** This is problem 2 in Section 1.3 of the script, where 200 is replaced by 100: Flip a coin 100 times. Record the number of times, heads appeared in the first 10 experiments and call this $n_1$. Then call the number of times, heads appears in the next $N = 10$ experiments and call it $n_2$. This produces 10 inters $n_1, \ldots, n_{10}$. Find the **mean**

$$m = (n_1 + n_2 + \ldots + n_N)/N$$

of your data, then the **sample variance**

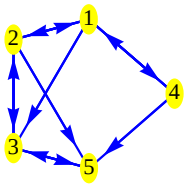
$$v = ((n_1 - m)^2 + (n_2 - m)^2 + \ldots + (n_N - m)^2)/(N - 1)$$

and finally the **sample standard deviation** $\sigma = \sqrt{v}$. Remark: there are statistical reasons that $(N - 1)$ is chosen and not $N$. It is called **Bessel's correction**.

If you use a computer, make a million experiments at least.

```
R:=Random[Integer,1]; M=100; data=Table[R,{M}];
m=Sum[data[[k]],{k,M}]/M;
sigma=Sqrt[Sum[(data[[k]]-m)^2,{k,M}]/(M-1)];
```

**3** a) Encode the following directed graph in a list of $5 \times 5$ integers 0 or 1. We call this a matrix. Use the rule that if there is an arrow from node $i$ to node $j$, then the matrix $A$ has an entry 1 in the $i$'th row and $j$'th column.




b) We often organize the data in a so called **binary tree**. Each vertex in the following tree can be described as a list of numbers $\{0, 1\}$. How is each point on the most outer circle determined by a list with 8 numbers? Assume you start at the origin and make steps to the right if 1 occurs in the list and left if 0 occurs. Assume you get the list $\{1, 1, 1, 1, 1, 1, 1, 1\}$. At which point do you end up? Which part of the outer circle corresponds to sequences which start with two 0?