Lecture 1 – Working with Data

Prof. Michael T. Heath

Department of Computer Science
University of Illinois at Urbana-Champaign
heath@illinois.edu

January 18, 2016

Objectives

Understand the what, when, who, why, where, and how of data!

- What are data?
- How are data represented?
- How can we manipulate and understand data?
- How can we make decisions based on data?

Example: ID Card or Driver's License

What kinds of data does it contain?

- name — character string
- id number — number (or is it?)
- date of birth — numbers
- height and weight — numbers
- eye color — character string
- photo — two-dimensional array of pixels

The data values for a given item (in this case a person) form a one-dimensional array or list, (which in this case is heterogeneous)

A one-dimensional array of *numbers* is called a *vector*

Aggregating Data

A collection of data is often organized as a two-dimensional array with data values in rows and columns, as in a spreadsheet

A class roster, for example, typically has a row for each student and columns for student name, id number, email address, homework and exam scores, etc.

In this example, the data are heterogeneous (numbers, strings, etc.)

A two-dimensional array of *numbers* is called a *matrix*

We will make extensive use of vectors and matrices in this course

Summarizing Data

To make large data sets more comprehensible, we often seek to summarize them by only a few values, such as an average

But not all such operations make sense on all data types

For which of the following data can we compute a meaningful class average?

- *exam score*
- *height*
- *name*
- *id number*
- *eye color*
- *age*

Later we will learn more sophisticated ways to approximate high-dimensional data using fewer dimensions

Representing Data

In a digital computer, data of all types are represented by bits, each of whose value is either 0 or 1

- bits $\rightarrow$ digits $\rightarrow$ numbers $\rightarrow$ vectors, matrices
- bits $\rightarrow$ characters $\rightarrow$ strings $\rightarrow$ text
- bits $\rightarrow$ pixels $\rightarrow$ images $\rightarrow$ video

A bit string can have different meanings, depending on context

01111010 represents 122 as a decimal integer, z as an ASCII character, and dark blue as an RGB color

We will focus primarily on numeric data, but analyzing other types of data may still require numeric computation (e.g., Google PageRank)

As its title suggests, *numerical methods* for such computations are the main subject of this course

Operating on Data

At the most fundamental level, digital computers have a very limited repertoire of operations they can perform, including

- arithmetic ($+$, $-$, $\times$, $\div$)
- logic (and, or, exor, not)
- comparison ($<$, $>$, $=$)

These operations are performed bit by bit, so more complicated operations must be built up from these basic operations

Most programming languages operate on groups of bits called *words* (integers, floating-point numbers, characters, etc.) and larger data structures (e.g., arrays, strings) composed of them

Python

In this class our main tool for working with data will be Python, a programming language that is well suited for numerical computing but also works with other types of data as well

If you do not already know Python, you can learn it incrementally via online demos, examples, and tutorials provided on the class website

There are many additional textbooks and online tutorials that may help you learn both the basics and more sophisticated aspects of Python

In addition to the usual capabilities for basic programming, Python also provides powerful capabilities for common numerical computations (via numpy and scipy), the use of which will make programming easier and your programs much more efficient