The nature of data unit:

- Objects and vectors (**today**)
- K nearest neighbors classification (Friday)
- (Start linear regression next week)

Today:

- Wrap-up from lab and recent examples
- From object to vectors
- Tidy data
- Increasing and decreasing attributes
- The curse of dimensionality

While not all data is numerical, it is often useful to consider data in a numerical format. In this book, we assume that data has already been appropriately converted into a numerical representation suitable for reading into a computer program. Therefore, we think of data as vectors. As another illustration of how subtle words are, there are (at least) three different ways to think about vectors: a vector as an array of numbers (a computer science view), a vector as an arrow with a direction and magnitude (a physics view), and a vector as an object that obeys addition and scaling (a mathematical view).

Deisenroth et al, Mathematics for Machine Learning, pg 4

Address	ZIP	Price	Taxes	Year	Sq ft	Flood plain
214 Hickory	27148	$417K	$12K	1987	2120	N
17 W Riverview	30921	$295	$7K	1927	3015	Y
55 Park NE	81066	$528K	$22K	2012	3525	N
2s912 Hopper	67811	$238	$4k	1971	1800	N

```
public class House {
    private String streetAddr;
    private int zip;
    private int price;
    private int taxes;
    private long yearBuilt;
    private int sqFt;
    private boolean floodPlain;
}
```

Address	ZIP	Price	Taxes	Year	Sq ft	Flood plain
214 Hickory	27148	$417K	$12K	1987	2120	N
17 W Riverview	30921	$295	$7K	1927	3015	Y
55 Park NE	81066	$528K	$22K	2012	3525	N
2s912 Hopper	67811	$238	$4k	1971	1800	N

```
char* street_addr[100];
int zip[100];
int price[100];
int taxes[100];
time_t year_build[100];
int sq_ft[100];
int flood_plain[100];
```

From Hadley Wickham, "Tidy Data", *Journal of Statistical Software*, Aug 2014, 59:10.

> *A dataset is a collection of values, usually either numbers (if quantitative) or strings (if qualitative). Values are organized in two ways. Every value belongs to a variable and an observation. A variable contains vall values that measure the same underlying attribute (like height, temperature, duration) across units. An observation contains all values measured on the same unit (like a person, a day, or a race) across attributes.* *Wickham, pg 3*
>
> *Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns, and tables are matched up with observations, variables, and types. In tidy data,*
>
> 1. *Each variable forms a column.*
> 2. *Each observation forms a row.*
> 3. *Each type of observational unit forms a table.*
>
> *Wickham, pg 4*

The curse of dimensionality is a very strong name, so you can probably guess that it is a bit of a problem. The essence of the curse is the realization that as the number of dimensions increases, the volume of the unit hypersphere does not increase with it. The unit hypersphere is the region we get if we start at the origin and draw all the points that are distance 1 away from the origin. In 2 dimensions we get a circle of radius 1 around (0,0), and in 3D we get a sphere around (0,0,0). The graph shows clearly that as the number of dimensions tends to infinity, so the volume of the hypersphere tends to zero.

Marsland, Machine Learning, An Algorithmic Perspective, pg 17–19

The origin of the problem [the curse of dimensionality] is illustrated in that if we divide a region of a space into regular cells, then the number of such cells grows exponentially with the dimensionality of the space. The problem with an exponentially large number of cells is that we would need an exponentially large quantity of training data in order to ensure that the cells are not empty.

Bishop, Pattern Recognition and Machine Learning, pg 35.

We are so used to living in three dimensions that our intuition fails us when we try to imagine a high-dimensional space. Even a basic 4D hypercube is incredibly hard to picture in our mind, let alone a 200-dimensional ellipsoid bent in a 1000-dimensional space.

It turns out that many things behave very differently in high-dimensional space. For example, if you pick a random point in a unit square, it will have only about a 0.4% *chance of being located less than 0.001 from a border (in other words, it is very unlikely that a random point will be "extreme" along any dimension.) But in a 10000-dimensional unit hypercube, this probability is greater than* 99.999999%. *Most points in a high-dimensional hypercube are very close to the border.*

In theory, one solution to the curse of dimensionality could be to increase the size of the training set to reach a sufficient density of training instances. Unfortunately, in practice, the number of training instances required to reach a given density grows exponentially with the number of dimensions. With just 100 features, you would need more training instances than atoms in the observable universe in order for the training instances to be within 0.1 of each other on average, assuming they were spread out uniformly across all dimensions.

Geron, Hand-On Machine Learning, pg 208–209

Coming up:

Take "objects and vectors" quiz (due class time Friday)

Do numpy assignment (due end-of-day Friday)
Read about data and models from Chapters 1 and 8 in the textbook (see Schoology for details) (due end-of-day next week Friday)

Read and respond to Boston Housing Dataset article (see Schoology for details) (due next week Tuesday)