Definition 1. Linear Code
An (n, k) *linear code* over a finite field F is a k-dimensional subspace V of the vector space

$$F^n = \underbrace{F \oplus F \oplus \cdots \oplus F}_{n \text{ copies}}$$

over F. The members of V are called the *code words*. The ratio k/n is called the *information rate* of the code. When F is $\mathbb{Z}_2$, the code is called binary.

Example 1. The Hamming (7,4) Code.
Assuming that our message consists of all possible 4-tuples of 0's and 1's (i.e., we wish to send a sequence of 0's and 1's of length 4). Encoding will be done by viewing these messages as four-dimensional vectors over the field $\mathbb{Z}_2$ and multiplying each of the 16 possible messages on the right by the matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The resulting seven-dimensional vectors are called *code words.* See Table 1.

Message	Encoder G	Code Word
0000	$\rightarrow$	0000000
0001	$\rightarrow$	0001111
0010	$\rightarrow$	0010110
0100	$\rightarrow$	0100101
1000	$\rightarrow$	1000011
1100	$\rightarrow$	1100110
1010	$\rightarrow$	1010101
1001	$\rightarrow$	1001100
0110	$\rightarrow$	0110011
0101	$\rightarrow$	0101010
0011	$\rightarrow$	0011001
1110	$\rightarrow$	1110000
1101	$\rightarrow$	1101001
1011	$\rightarrow$	1011010
0111	$\rightarrow$	0111100
1111	$\rightarrow$	1111111

Table 1

Definition 2. Hamming Distance, Hamming Weight
The *Hamming distance* between two vectors of a vector space is the number of components in which they differ. The *Hamming weight* of a vector is the umber of nonzero components of the vector.
We will use $d(u, v)$ to denote the Hamming distance between the vectors u and v and $\mathrm{wt}(u)$ for the Hamming weight of the vector u.

Theorem 1. *Properties of Hamming Distance and Hamming Weight*
For any vectors u, v and w of a linear code, $d(u, v) \leq d(u, w) + d(w, v)$ and $d(u, v) = wt(u - v)$.

Theorem 2. *Correcting Capability of a Linear Code*
If the Hamming weight of every nonzero code word in a linear code is at least $2t+1$, then the code can correct for any t or fewer errors. Furthermore, the same code can detect any $2t$ or fewer errors..