Inner product of two vectors $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$ (real numbers; vectors of length n.)

then $u \cdot v = (u,v) = u^T v = \sum_{k=1}^{n} u_k v_k$

with complex vectors put a complex conjugate on one of them....
Julia has complex numbers built in.

Outer product of two vectors $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$ (vectors can be different length)

$$u \otimes v = uv^T = \begin{bmatrix} u_1v_1 & u_1v_2 & \cdots & u_1v_n \\ \vdots & & & \\ u_mv_1 & & & u_mv_n \end{bmatrix} \in \mathbb{R}^{m \times n}$$

$m \times n$ matrix with real entries

$$u = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad v = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 3 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 3 & 4 & 5 \\ 6 & 8 & 10 \end{bmatrix} = uv^T$$

Matrix-Matrix product $A \in \mathbb{R}^{m \times n}$ $B \in \mathbb{R}^{n \times p}$

$$A = \begin{bmatrix} - & u_1^T & - \\ - & u_2^T & - \\ & \vdots & \\ - & u_m^T & - \end{bmatrix} \in \mathbb{R}^{m \times n} \quad u_i \in \mathbb{R}^n \qquad B = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_p \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times p} \quad v_i \in \mathbb{R}^n$$

(rows; columns)

Definition from linear Algebra class (using inner products)

$$AB = \begin{bmatrix} - & u_1^T & - \\ - & u_2^T & - \\ & \vdots & \\ - & u_m^T & - \end{bmatrix} \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_p \\ | & | & & | \end{bmatrix} = \begin{bmatrix} u_1^Tv_1 & u_1^Tv_2 & \cdots & u_1^Tv_p \\ \vdots & & & \vdots \\ u_m^Tv_1 & - & - & u_m^Tv_p \end{bmatrix}$$

$$AB = \begin{bmatrix} u_1 \cdot v_1 & u_1 \cdot v_2 & \cdots & u_1 \cdot v_p \\ \vdots & & & \vdots \\ u_m \cdot v_1 & - & - & u_m \cdot v_p \end{bmatrix}$$

Matrix-matrix product using outer products.

$$A = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{m \times n} \qquad a_i \in \mathbb{R}^m$$

$$B = \begin{bmatrix} - & b_1^T & - \\ - & b_2^T & - \\ & \vdots & \\ - & b_n^T & - \end{bmatrix} \in \mathbb{R}^{n \times p} \qquad b_i \in \mathbb{R}^p$$

$$AB = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} - & b_1^T & - \\ - & b_2^T & - \\ & \vdots & \\ - & b_n^T & - \end{bmatrix} = \sum_{k=1}^{n} a_k b_k^T$$

Idea is use the outer product to understand the LU matrix factorization...

Recall Gram-Schmidt algorithm. It creates a triangular matrix U by performing row operations. The multipliers in the elimination steps go into the matrix L. Then $A = LU$.

Consider two matrices

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 8 & 1 & 0 \\ 9 & 10 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 0 & 0 & 7 \end{bmatrix}$$

$\det L = 1 \cdot 1 \cdot 1 = 1$ $\qquad$ $\det U = 2 \cdot 5 \cdot 7 = 70$

Multiply LU using outer products

$$l_1 = \begin{bmatrix} 1 \\ 8 \\ 9 \end{bmatrix} \quad l_2 = \begin{bmatrix} 0 \\ 1 \\ 10 \end{bmatrix} \quad l_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$u_1 = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} \quad u_2 = \begin{bmatrix} 0 \\ 5 \\ 6 \end{bmatrix} \quad u_3 = \begin{bmatrix} 0 \\ 0 \\ 7 \end{bmatrix}$$

column vectors

$$LU = l_1 u_1^T + l_2 u_2^T + l_3 u_3^T$$

```
julia> L=[1 0 0; 8 1 0; 9 10 1]
3×3 Matrix{Int64}:
 1   0  0
 8   1  0
 9  10  1

julia> U=[2 3 4; 0 5 6; 0 0 7]
3×3 Matrix{Int64}:
 2  3  4
 0  5  6
 0  0  7
```

Notation for the j^{th} column

$L[:, j]$

Notation for the i^{th} row

$U[i, :]$ as a column vector

```
julia> L[:,1]*U[1,:]'
3×3 Matrix{Int64}:
  2   3   4
 16  24  32
 18  27  36

julia> L[:,2]*U[2,:]'
3×3 Matrix{Int64}:
 0   0   0
 0   5   6
 0  50  60

julia> L[:,3]*U[3,:]'
3×3 Matrix{Int64}:
 0  0  0
 0  0  0
 0  0  7
```

first row same

```
julia> L[:,1]
3-element Vector{Int64}:
 1
 8
 9

julia> U[1,:]
3-element Vector{Int64}:
 2
 3
 4
```

column vector

column vector

```
julia> L*U
3×3 Matrix{Int64}:
 2   3    4
 16  29   38
 18  77  103
```

neither inner or outer products but some optimized way in between for speed depending on hardware...

same...

```
julia> sum(L[:,k]*U[k,:]' for k=1:3)
3×3 Matrix{Int64}:
  2   3    4
 16  29   38
 18  77  103
```

$$LU = \sum_{k=1}^{3} l_k u_k^T =$$

Given $A = \begin{bmatrix} 2 & 3 & 4 \\ 16 & 29 & 38 \\ 18 & 77 & 103 \end{bmatrix}$ a way of finding LU different than Gaussian elimination...

Ae_1 where e_i are the standard basis vectors.

first column of A.

$$\begin{bmatrix} 2 & 3 & 4 \\ 16 & 29 & 38 \\ 18 & 77 & 103 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 16 \\ 18 \end{bmatrix} = A[:,1]$$

$e_1^T A = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 16 & 29 & 38 \\ 18 & 77 & 103 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} = A[1,:]^T$

first row ...

here the rows come out as rows

Now suppose $A = LU$. Then

$$e_1^T A = e_1^T LU = e_1^T \sum_{k=1}^{3} l_k u_k^T = \sum_{k=1}^{3} e_1^T l_k u_k^T = u_1^T$$

first row of A

$(e_1 \cdot l_k)$

first row of U.

$$L = \begin{bmatrix} 1 & 0 & 0 \\ ? & 1 & 0 \\ ? & ? & 1 \end{bmatrix}$$

l_1 l_2 l_3

Friday is lab... Next week finish figuring out what LU is from outer product representation of matrix-matrix product...