$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}_{m\times n}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}}_{x} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{m\times 1}}_{b}$$

# CX1104: **Linear Algebra for Computing**

Chap. No : **6.3.1**

Lecture : **Orthogonality**

Topic : **Gram–Schmidt for QR**

Concept : **Motivation and Review of Concepts**


Instructor: **A/P Chng Eng Siong**

TAs: **Zhang Su, Vishal Choudhari**



Rev: ces-22July2020

Rev: 29th June 2020

# Introducing A = QR

QR decomposition, given A (mxn) matrix, we can decompose it into the product of 2 matrixes,

$$A = QR$$

- Properties of Q:
  - C($Q$) == C($A$)
  - $Q^TQ = I$ , i.e $Q$ has orthonormal column (BUT not necessary square)
  - $QQ^T$ = projection matrix into col (A)

- $R$ is a square upper triangle matrix and Depending if
  - A has independent col, then R is invertible,
  - A has dependent col, then R is NOT-invertible.

Ref: 1) [https://en.wikipedia.org/wiki/QR_decomposition](https://en.wikipedia.org/wiki/QR_decomposition)
2) http://ee263.stanford.edu/lectures/qr.pdf

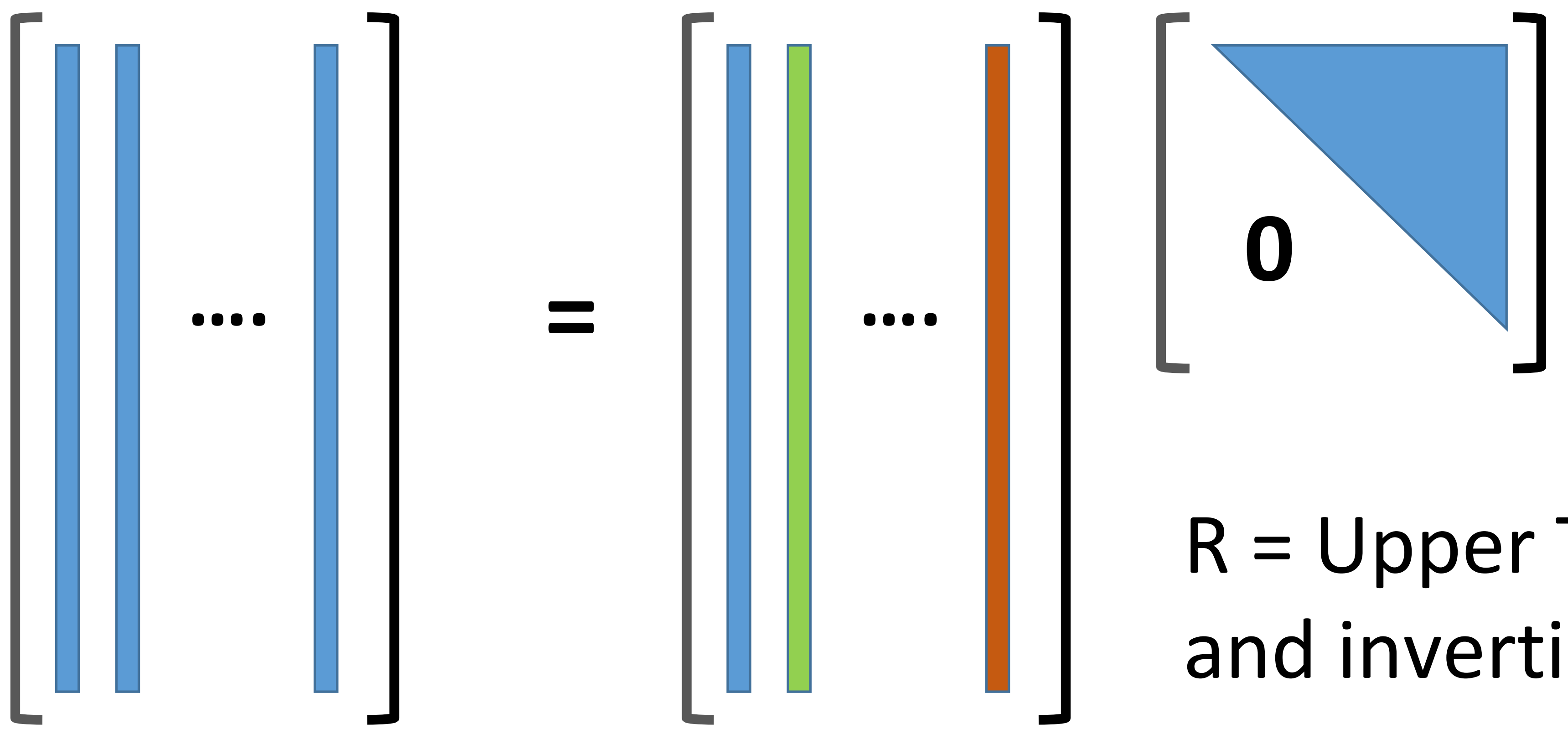
Warning: Theorem 12 is for A Having independent column.

THEOREM 12

**The QR Factorization**

If $A$ is an $m \times n$ matrix with linearly independent columns, then $A$ can be factored as $A = QR$, where $Q$ is an $m \times n$ matrix whose columns form an orthonormal basis for Col $A$ and $R$ is an $n \times n$ upper triangular invertible matrix with positive entries on its diagonal.

$$A = [\,\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_n\,] = [\,Q\mathbf{r}_1 \quad \cdots \quad Q\mathbf{r}_n\,] = QR$$




R = Upper Triangle and invertible

A $\in R^{m*n}$ Q = columns are orthonormal

# Motivation for QR

It has many applications, e.g,
solving least squares:

$$
\begin{aligned}
Ax &= y \\
QRx &= y \\
Q^T QRx &= Q^T y
\end{aligned}
$$

$$Rx = Q^T y$$

Is can be easily solved because $R$ is upper triangle
(If R is not invertible, it will be more involved, see least squares chapter)

There are at least 3 approaches to realise QR decomposition,
we will only introduce Gram-Schmidt orthogonalization to get Q first
https://www.math.ucla.edu/~yanovsky/Teaching/Math151B/handouts/GramSchmidt.pdf
https://towardsdatascience.com/can-qr-decomposition-be-actually-faster-schwarz-rutishauser-algorithm-a32c0cde8b9b

# What does having same column space mean? C(Q) == C(A)

When we can convert

$$Ax = b$$

To

$$QRx = b$$

Then finding the solution

$x$ is easier and more efficient especially when A is large sized

(e.g thousands of rows and columns)

The found solution $x$ will be the same.

Chapter 3. Vector Spaces and Subspaces

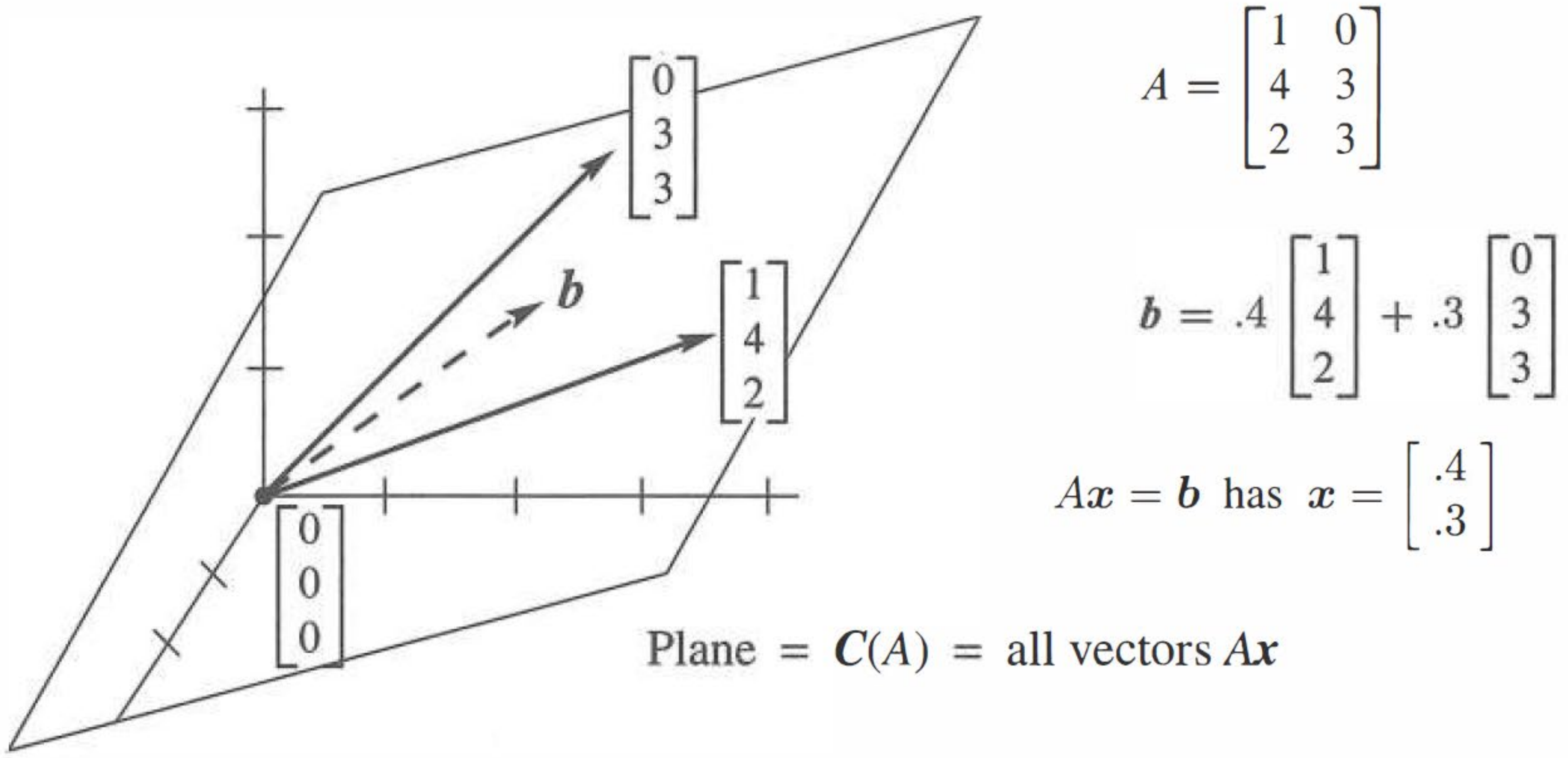



Figure 3.2: The column space $C(A)$ is a plane containing the two columns. $Ax = b$ is solvable when $b$ is on that plane. Then $b$ is a combination of the columns.

Interpretation: since C(Q) == C(A),

then Columns of $Q$ are orthonormal basis for range(A), since $Q^T Q = I$

# Revision: Projecting y onto an orthogonal vs orthonormal basis

**THEOREM 8** **The Orthogonal Decomposition Theorem**

Let $W$ be a subspace of $\mathbb{R}^n$. Then each $\mathbf{y}$ in $\mathbb{R}^n$ can be written uniquely in the form

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{z} \tag{1}$$

where $\hat{\mathbf{y}}$ is in $W$ and $\mathbf{z}$ is in $W^{\perp}$. In fact, if $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$ is any orthogonal basis of $W$, then

$$\hat{\mathbf{y}} = \frac{\mathbf{y}\cdot\mathbf{u}_1}{\mathbf{u}_1\cdot\mathbf{u}_1}\mathbf{u}_1 + \cdots + \frac{\mathbf{y}\cdot\mathbf{u}_p}{\mathbf{u}_p\cdot\mathbf{u}_p}\mathbf{u}_p \tag{2}$$

and $\mathbf{z} = \mathbf{y} - \hat{\mathbf{y}}$.

Lay5e pg 350

The vector $\hat{\mathbf{y}}$ in (1) is called the **orthogonal projection of y onto** $W$ and often is written as $\text{proj}_W\,\mathbf{y}$. See Figure 2. When $W$ is a one-dimensional subspace, the formula for $\hat{\mathbf{y}}$ matches the formula given in Section 6.2.

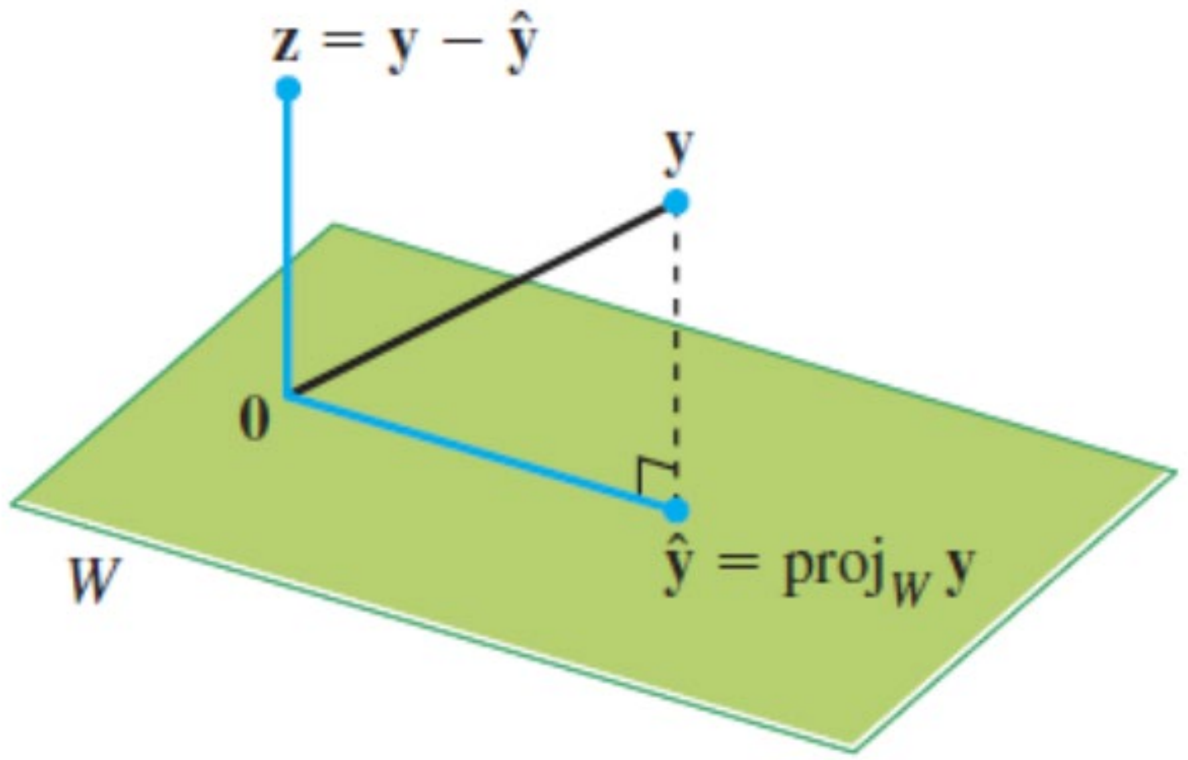



**FIGURE 2** The orthogonal projection of $\mathbf{y}$ onto $W$.

Lay5e pg 353

The final theorem in this section shows how formula (2) for $\text{proj}_W\,\mathbf{y}$ is simplified when the basis for $W$ is an orthonormal set.

**THEOREM 10**

If $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$ is an orthonormal basis for a subspace $W$ of $\mathbb{R}^n$, then

$$\text{proj}_W\,\mathbf{y} = (\mathbf{y}\cdot\mathbf{u}_1)\mathbf{u}_1 + (\mathbf{y}\cdot\mathbf{u}_2)\mathbf{u}_2 + \cdots + (\mathbf{y}\cdot\mathbf{u}_p)\mathbf{u}_p \tag{4}$$

If $U = [\,\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_p\,]$, then

$$\text{proj}_W\,\mathbf{y} = UU^T\mathbf{y} \quad \text{for all } \mathbf{y} \text{ in } \mathbb{R}^n \tag{5}$$

**PROOF** Formula (4) follows immediately from (2) in Theorem 8. Also, (4) shows that $\text{proj}_W\,\mathbf{y}$ is a linear combination of the columns of $U$ using the weights $\mathbf{y}\cdot\mathbf{u}_1, \mathbf{y}\cdot\mathbf{u}_2, \dots, \mathbf{y}\cdot\mathbf{u}_p$. The weights can be written as $\mathbf{u}_1^T\mathbf{y}, \mathbf{u}_2^T\mathbf{y}, \dots, \mathbf{u}_p^T\mathbf{y}$, showing that they are the entries in $U^T\mathbf{y}$ and justifying (5). ∎

# How to find Q from A?

## What does Gram-Schmidt Process Do?

It **orthogonalises** a set of vectors!

- $A$ is a mxn matrix, **assume columns are** Not orthogonal
- $\boldsymbol{v_i} \in \boldsymbol{R^m}, i = 1..\boldsymbol{n}$

- The columns of Q are **Orthonormal** $\boldsymbol{u_i}$ $i = 1..n$
- **And spans the same space as A**

$A = \{v_1, v_2, v_3, \cdots, v_n\} \rightarrow$ Gram-Schmidt Process $\rightarrow Q = \{u_1, u_2, u_3, \cdots, u_n\}$

$$Span(A) = Span(Q)$$

**Note:**Q spans the same m-dimensional subspace of $R^m$ as that of $A$

Ref: http://www.seas.ucla.edu/~vandenbe/133A/lectures/qr.pdf Slide 6.7
See Matlab: https://www.mathworks.com/help/matlab/ref/qr.html
(economy QR factorization vs full QR decomposition)
More explanations of full vs economy qr : http://www.ece.northwestern.edu/local-apps/matlabhelp/techdoc/math_anal/mat_li23.html

# CX1104: **Linear Algebra for Computing**

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}_{m\times n}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}}_{x} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{m\times 1}}_{b}$$

Chap. No : **6.3.2**

Lecture : **Orthogonality**

Topic : **Gram–Schmidt Process**

Concept : **Gram-Schmidt Process for QR decomposition**

Instructor: **A/P Chng Eng Siong**

TAs: **Zhang Su, Vishal Choudhari**

Rev: 25 Sep 2021

Rev: 29th June 2020

# QR Factorisation revisited

The QR decomposition can be performed by Gram–Schmidt. Given a matrix A (mxn sized),

$$A = QR$$

$$A = [q_1 \quad q_2 \quad \cdots \quad q_n] \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1n} \\ 0 & R_{22} & \cdots & R_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{nn} \end{bmatrix}$$

**The $Q$ Factor (economy qr):**

- $Q$ is $m \times n$ with orthonormal columns and $Q^TQ = I$ dimension nxn
- If $A$ is square $(m = n)$, then Q is orthogonal, i.e, $Q^TQ = QQ^T = I$
- If A is tall (m>n), then $QQ^T \neq I$,
  The matrix $QQ^T$ is a projection matrix
  of dimension mxm, and it will project a vector $R^m$ onto
  the columns space of $A$. In other words,
  $QQ^Ty = \hat{y}$, where $\hat{y}$ is the least squares error approximation of $y$ in the column space of $A$ (see sec 7.1.4)

**The $R$ Factor:**

- $R$ is $n \times n$ upper triangular,
- If A has independent column, then R is invertible,
  else R is singular (not-invertible)

- Vectors $q_1, q_2, \dots, q_n$ are orthonormal m-dimensional vectors: $||q_i|| = 1 \quad and \quad q_i^Tq_j = 0 \; if \;\; i \neq j$

**NOTE:**

**$Q$ is obtained by performing GS Process on $A$**

# The Gram Schmidt Process

**What does Gram-Schmidt Process Do?**

It orthogonalises a set of vectors!

$W = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$
$\boldsymbol{x}_i \in R^m$

$\rightarrow$ **Gram-Schmidt Process** $\rightarrow$

$W' = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$
$\boldsymbol{v}_i \in R^m$ and
$\boldsymbol{v}_i . \boldsymbol{v}_j = 0$ when $i \neq j$

$$Span(W) = Span(W')$$

Typically, we are give a matrix $A$, and these $x_i$ are columns of $A$.

# The Gram Schmidt Process******

In basic Gram-Schmidt, we assume that $\{x_1, x_2, ..\}$ are independent columns,

**THEOREM 11**

**The Gram–Schmidt Process**

Given a basis $\{\mathbf{x}_1, \dots, \mathbf{x}_p\}$ for a nonzero subspace $W$ of $\mathbb{R}^n$, define

$$\mathbf{v}_1 = \mathbf{x}_1$$

$$\mathbf{v}_2 = \mathbf{x}_2 - \frac{\mathbf{x}_2 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1$$

$$\mathbf{v}_3 = \mathbf{x}_3 - \frac{\mathbf{x}_3 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1 - \frac{\mathbf{x}_3 \cdot \mathbf{v}_2}{\mathbf{v}_2 \cdot \mathbf{v}_2}\mathbf{v}_2$$

$$\vdots$$

$$\mathbf{v}_p = \mathbf{x}_p - \frac{\mathbf{x}_p \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1 - \frac{\mathbf{x}_p \cdot \mathbf{v}_2}{\mathbf{v}_2 \cdot \mathbf{v}_2}\mathbf{v}_2 - \cdots - \frac{\mathbf{x}_p \cdot \mathbf{v}_{p-1}}{\mathbf{v}_{p-1} \cdot \mathbf{v}_{p-1}}\mathbf{v}_{p-1}$$

Then $\{\mathbf{v}_1, \dots, \mathbf{v}_p\}$ is an orthogonal basis for $W$. In addition

$$\text{Span}\{\mathbf{v}_1, \dots, \mathbf{v}_k\} = \text{Span}\{\mathbf{x}_1, \dots, \mathbf{x}_k\} \quad \text{for } 1 \le k \le p \tag{1}$$

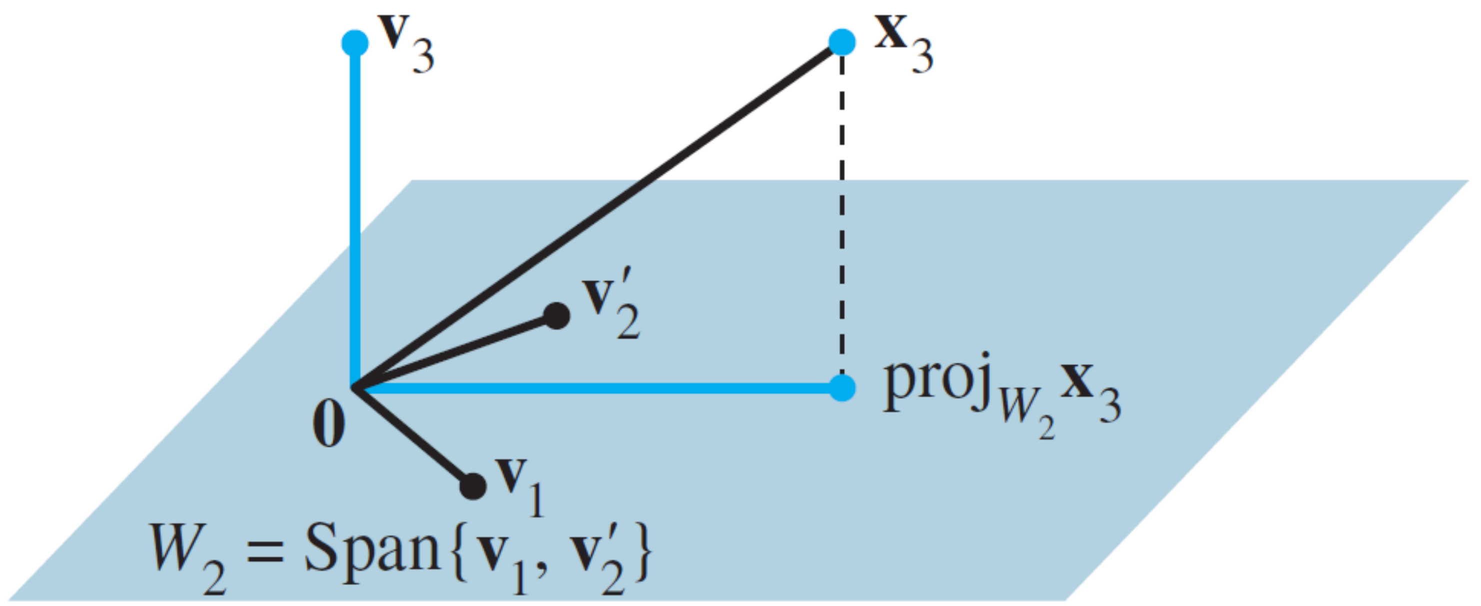
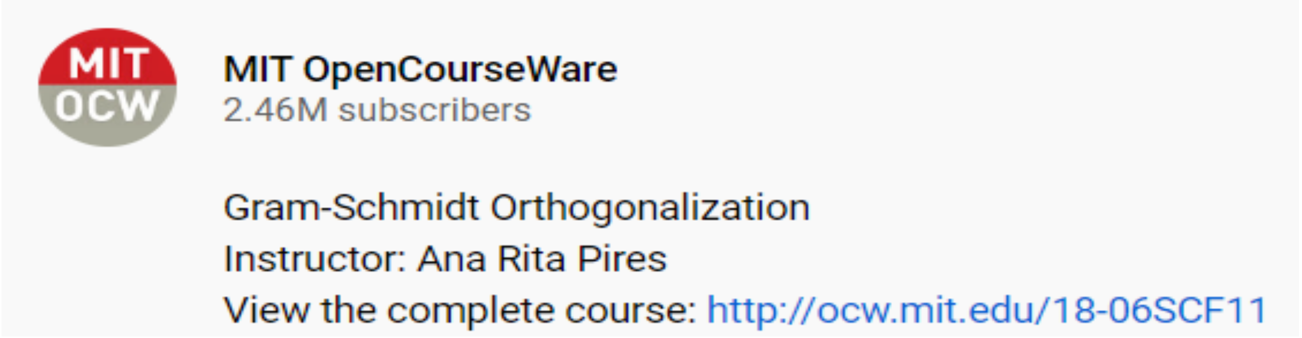
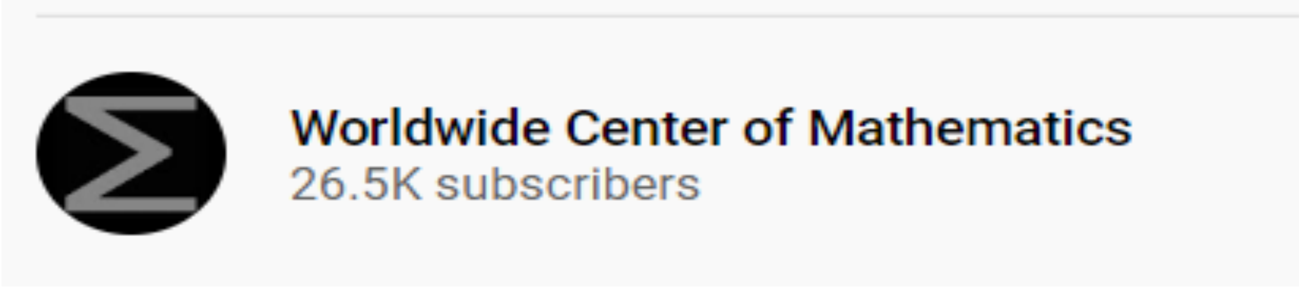



**FIGURE 2** The construction of $\mathbf{v}_3$ from $\mathbf{x}_3$ and $W_2$.

Watch these worked out examples:

1. GramSchmidt: https://www.youtube.com/watch?v=Aslf3KGq2UE
2. QR: https://www.youtube.com/watch?v=6DybLNNkWyE
3. MIT Gram Schmidt: https://www.youtube.com/watch?v=TRktLuAktBQ&t=17s




Lay, Linear Algebra and its Applications (4th Edition)

354 CHAPTER 6 Orthogonality and Least Squares

# Proof Theorem 11: Span of vectors generated by GS is same as original set of vectors (proof not tested)

**PROOF** For $1 \le k \le p$, let $W_k = \text{Span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$. Set $\mathbf{v}_1 = \mathbf{x}_1$, so that $\text{Span}\{\mathbf{v}_1\} = \text{Span}\{\mathbf{x}_1\}$. Suppose, for some $k < p$, we have constructed $\mathbf{v}_1, \ldots, \mathbf{v}_k$ so that $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ is an orthogonal basis for $W_k$. Define

$$\mathbf{v}_{k+1} = \mathbf{x}_{k+1} - \text{proj}_{W_k} \mathbf{x}_{k+1} \tag{2}$$

By the Orthogonal Decomposition Theorem, $\mathbf{v}_{k+1}$ is orthogonal to $W_k$. Note that $\text{proj}_{W_k} \mathbf{x}_{k+1}$ is in $W_k$ and hence also in $W_{k+1}$. Since $\mathbf{x}_{k+1}$ is in $W_{k+1}$, so is $\mathbf{v}_{k+1}$ (because $W_{k+1}$ is a subspace and is closed under subtraction). Furthermore, $\mathbf{v}_{k+1} \neq \mathbf{0}$ because $\mathbf{x}_{k+1}$ is not in $W_k = \text{Span}\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$. Hence $\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\}$ is an orthogonal set of nonzero vectors in the $(k+1)$-dimensional space $W_{k+1}$. By the Basis Theorem in Section 4.5, this set is an orthogonal basis for $W_{k+1}$. Hence $W_{k+1} = \text{Span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\}$. When $k + 1 = p$, the process stops. ■

Theorem 11 shows that any nonzero subspace $W$ of $\mathbb{R}^n$ has an orthogonal basis, because an ordinary basis $\{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ is always available (by Theorem 11 in Section 4.5), and the Gram–Schmidt process depends only on the existence of orthogonal projections onto subspaces of $W$ that already have orthogonal bases.

# Example: (must know how to compute)

**EXAMPLE 2** Let $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$, $\mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$, and $\mathbf{x}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$. Then $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ is clearly linearly independent and thus is a basis for a subspace $W$ of $\mathbb{R}^4$. Construct an orthogonal basis for $W$.

**SOLUTION**

***Step 1.*** Let $\mathbf{v}_1 = \mathbf{x}_1$ and $W_1 = \text{Span}\{\mathbf{x}_1\} = \text{Span}\{\mathbf{v}_1\}$.

***Step 2.*** Let $\mathbf{v}_2$ be the vector produced by subtracting from $\mathbf{x}_2$ its projection onto the subspace $W_1$. That is, let

$$\begin{aligned} \mathbf{v}_2 &= \mathbf{x}_2 - \text{proj}_{W_1} \mathbf{x}_2 \\ &= \mathbf{x}_2 - \frac{\mathbf{x}_2 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1 \qquad \text{Since } \mathbf{v}_1 = \mathbf{x}_1 \\ &= \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \frac{3}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -3/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \end{aligned}$$

As in Example 1, $\mathbf{v}_2$ is the component of $\mathbf{x}_2$ orthogonal to $\mathbf{x}_1$, and $\{\mathbf{v}_1, \mathbf{v}_2\}$ is an orthogonal basis for the subspace $W_2$ spanned by $\mathbf{x}_1$ and $\mathbf{x}_2$.

Note: $v_2' = v_2 * 4$ to get rid of denominator in $v_2$

***Step 3.*** Let $\mathbf{v}_3$ be the vector produced by subtracting from $\mathbf{x}_3$ its projection onto the subspace $W_2$. Use the orthogonal basis $\{\mathbf{v}_1, \mathbf{v}_2'\}$ to compute this projection onto $W_2$:

$$\text{proj}_{W_2} \mathbf{x}_3 = \overbrace{\frac{\mathbf{x}_3 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1}^{\substack{\text{Projection of} \\ \mathbf{x}_3 \text{ onto } \mathbf{v}_1}} + \overbrace{\frac{\mathbf{x}_3 \cdot \mathbf{v}_2'}{\mathbf{v}_2' \cdot \mathbf{v}_2'}\mathbf{v}_2'}^{\substack{\text{Projection of} \\ \mathbf{x}_3 \text{ onto } \mathbf{v}_2'}} = \frac{2}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \frac{2}{12}\begin{bmatrix} -3 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2/3 \\ 2/3 \\ 2/3 \end{bmatrix}$$

Then $\mathbf{v}_3$ is the component of $\mathbf{x}_3$ orthogonal to $W_2$, namely,

$$\mathbf{v}_3 = \mathbf{x}_3 - \text{proj}_{W_2} \mathbf{x}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 2/3 \\ 2/3 \\ 2/3 \end{bmatrix} = \begin{bmatrix} 0 \\ -2/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

See Slide 3 of Chapter 6.2.5 for explanation.

Lay, Linear Algebra and its Applications (4th Edition)

# Example:

$$\mathbf{v}_3 = \mathbf{x}_3 - \text{proj}_{W_2}\,\mathbf{x}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 2/3 \\ 2/3 \\ 2/3 \end{bmatrix} = \begin{bmatrix} 0 \\ -2/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

See Fig. 2 for a diagram of this construction. Observe that $\mathbf{v}_3$ is in $W$, because $\mathbf{x}_3$ and $\text{proj}_{W_2}\mathbf{x}_3$ are both in $W$. Thus $\{\mathbf{v}_1, \mathbf{v}_2', \mathbf{v}_3\}$ is an orthogonal set of nonzero vectors and hence a linearly independent set in $W$. Note that $W$ is three-dimensional since it was defined by a basis of three vectors. Hence, by the Basis Theorem in Section 4.5, $\{\mathbf{v}_1, \mathbf{v}_2', \mathbf{v}_3\}$ is an orthogonal basis for $W$. ■

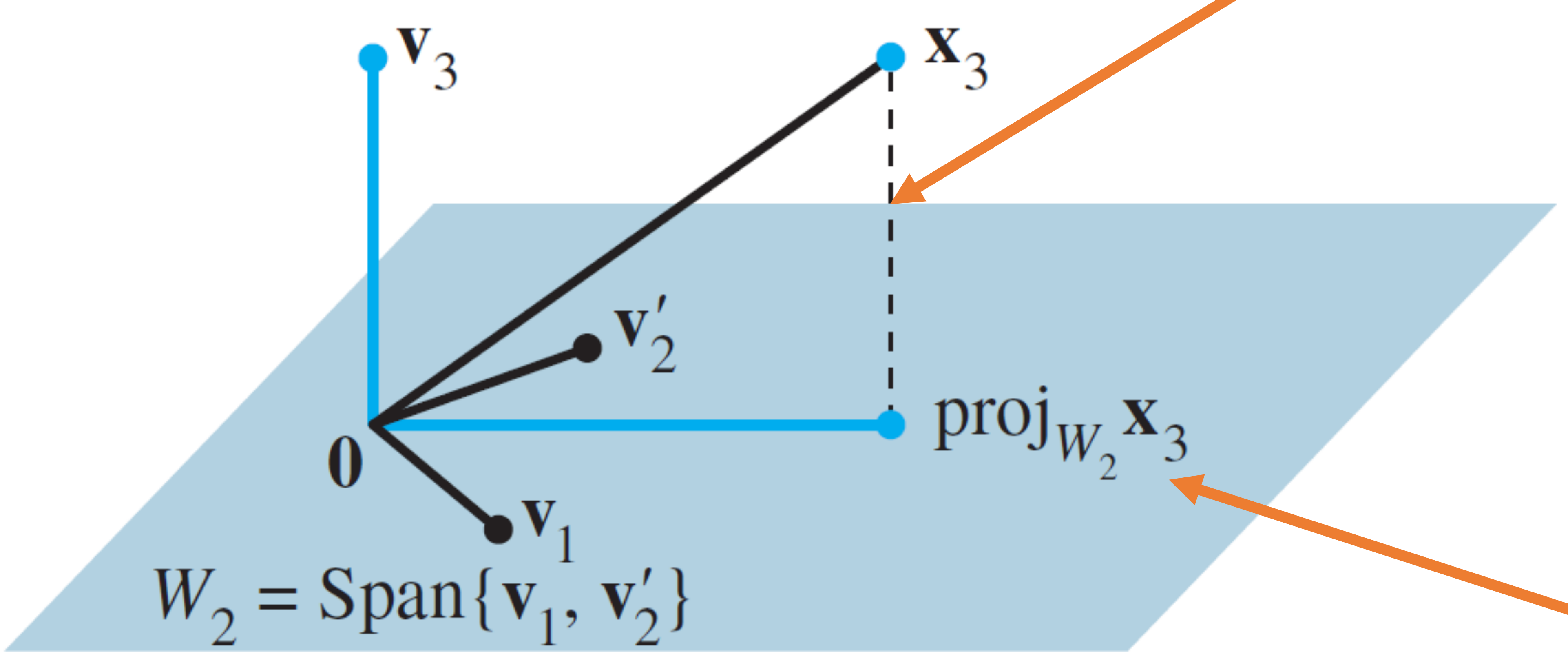



**FIGURE 2** The construction of $\mathbf{v}_3$ from $\mathbf{x}_3$ and $W_2$.

$$\text{proj}_{W_2}\,\mathbf{x}_3 = \underbrace{\frac{\mathbf{x}_3 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1}_{\text{Projection of } \mathbf{x}_3 \text{ onto } \mathbf{v}_1} + \underbrace{\frac{\mathbf{x}_3 \cdot \mathbf{v}_2'}{\mathbf{v}_2' \cdot \mathbf{v}_2'}\mathbf{v}_2'}_{\text{Projection of } \mathbf{x}_3 \text{ onto } \mathbf{v}_2'} = \frac{2}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \frac{2}{12}\begin{bmatrix} -3 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2/3 \\ 2/3 \\ 2/3 \end{bmatrix}$$

**The Basis Theorem**

Let $V$ be a $p$-dimensional vector space, $p \geq 1$. Any linearly independent set of exactly $p$ elements in $V$ is automatically a basis for $V$. Any set of exactly $p$ elements that spans $V$ is automatically a basis for $V$.

Lay, Linear Algebra and its Applications (4th Edition)

# Orthogonal basis vs Orthonormal basis

The columns of Q
are orthogonal,
as well as orthonormal!
$Q^T Q = I$

Orthonormal == orthogonal +
(length of vector ==1)

## Orthonormal Bases

An orthonormal basis is constructed easily from an orthogonal basis $\{\mathbf{v}_1, \dots, \mathbf{v}_p\}$: simply normalize (i.e., "scale") all the $\mathbf{v}_k$. When working problems by hand, this is easier than normalizing each $\mathbf{v}_k$ as soon as it is found (because it avoids unnecessary writing of square roots).

**EXAMPLE 3** Example 1 constructed the orthogonal basis

$$\mathbf{v}_1 = \begin{bmatrix} 3 \\ 6 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

An orthonormal basis is

$$\mathbf{u}_1 = \frac{1}{\|\mathbf{v}_1\|}\mathbf{v}_1 = \frac{1}{\sqrt{45}}\begin{bmatrix} 3 \\ 6 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \\ 0 \end{bmatrix}$$

$$\mathbf{u}_2 = \frac{1}{\|\mathbf{v}_2\|}\mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \blacksquare$$

# The entries of R matrix during GS

Consider A has independent col (mxn matrix)

$$A = [\, x_1, x_2, x_3, \dots, x_n]$$

Gram–Schmidt Process

$$\mathrm{Proj}_v x = \frac{v.x}{v.v}\, v$$

$$v_1 = x_1$$

$$v_2 = x_2 - \mathrm{Proj}_{v_1} x_2$$

$$v_3 = x_3 - \mathrm{Proj}_{v_1} x_3 - \mathrm{Proj}_{v_2} x_3$$

$$v_i = x_i - \sum_{j=1}^{i-1} \mathrm{Proj}_{v_j} x_i$$

Ortho-normalization

$$u_1 = \frac{v_1}{||v_1||}$$

$$u_2 = \frac{v_2}{||v_2||}$$

$$u_i = \frac{v_i}{||v_i||}$$

We can now express the $x_i$ over our newly computed orthonormal basis: $e_j$

$$x_1 = u_1.x_1\ u_1$$

$$x_2 = u_1.x_2\ u_1 + u_2.x_2\ u_2$$

$$x_3 = u_1.x_3\ u_1 + u_2.x_3\ u_2 + u_3.x_3\ u_3$$

$$\vdots$$

$$x_n = \sum_{j=1}^{n} u_j . x_n\ u_j$$

This can be written in matrix form:

$$A = QR$$

where:

$$Q = [\, u_1, u_2, u_3, \dots, u_n]$$

and

$$R = \begin{pmatrix} u_1.x_1 & u_1.x_2 & u_1.x_3 & \cdots \\ 0 & u_2.x_2 & u_2.x_3 & \cdots \\ 0 & 0 & u_3.x_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

https://www.math.ucla.edu/~yanovsky/Teaching/Math151B/handouts/GramSchmidt.pdf

# Example: Gram–Schmidt on a 3x3 matrix

**Example** [ edit ]

Consider the decomposition of

$$A = \begin{pmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{pmatrix}.$$

Recall that an orthonormal matrix $Q$ has the property

$$Q^\mathsf{T} Q = I.$$

Then, we can calculate $Q$ by means of Gram–Schmidt as follows:

$$U = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{pmatrix} = \begin{pmatrix} 12 & -69 & -58/5 \\ 6 & 158 & 6/5 \\ -4 & 30 & -33 \end{pmatrix};$$

$$Q = \begin{pmatrix} \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} & \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} & \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \end{pmatrix} = \begin{pmatrix} 6/7 & -69/175 & -58/175 \\ 3/7 & 158/175 & 6/175 \\ -2/7 & 6/35 & -33/35 \end{pmatrix}$$

Ref: https://en.wikipedia.org/wiki/QR_decomposition#Using_the_Gram–Schmidt_process

Thus, we have

$$Q^\mathsf{T} A = Q^\mathsf{T} Q\, R = R;$$

$$R = Q^\mathsf{T} A = \begin{pmatrix} 14 & 21 & -14 \\ 0 & 175 & -70 \\ 0 & 0 & 35 \end{pmatrix}$$

Hence,

$$A = \begin{pmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{pmatrix} =$$

$$\begin{pmatrix} 6/7 & -69/175 & -58/175 \\ 3/7 & 158/175 & 6/175 \\ -2/7 & 6/35 & -33/35 \end{pmatrix} \times \begin{pmatrix} 14 & 21 & -14 \\ 0 & 175 & -70 \\ 0 & 0 & 35 \end{pmatrix}$$

$Q$: Orthogonal Matrix

$R$: Upper Triangular Matrix

# Warning: modify QR when A has dependent column! (X)

From: pg 4-6 http://ee263.stanford.edu/lectures/qr.pdf

Note: here the columns of Q are denoted as $q_i$

## $QR$ decomposition

written in matrix form: $A = QR$, where $A \in \mathbb{R}^{m\times n}$, $Q \in \mathbb{R}^{m\times n}$, $R \in \mathbb{R}^{n\times n}$:

$$\underbrace{\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}}_{R}$$

- in basic G-S we assume $a_1, \ldots, a_n \in \mathbb{R}^m$ are independent
- if $a_1, \ldots, a_n$ are dependent, we find $\tilde{q}_j = 0$ for some $j$, which means $a_j$ is linearly dependent on $a_1, \ldots, a_{j-1}$
- modified algorithm: when we encounter $\tilde{q}_j = 0$, skip to next vector $a_{j+1}$ and continue:

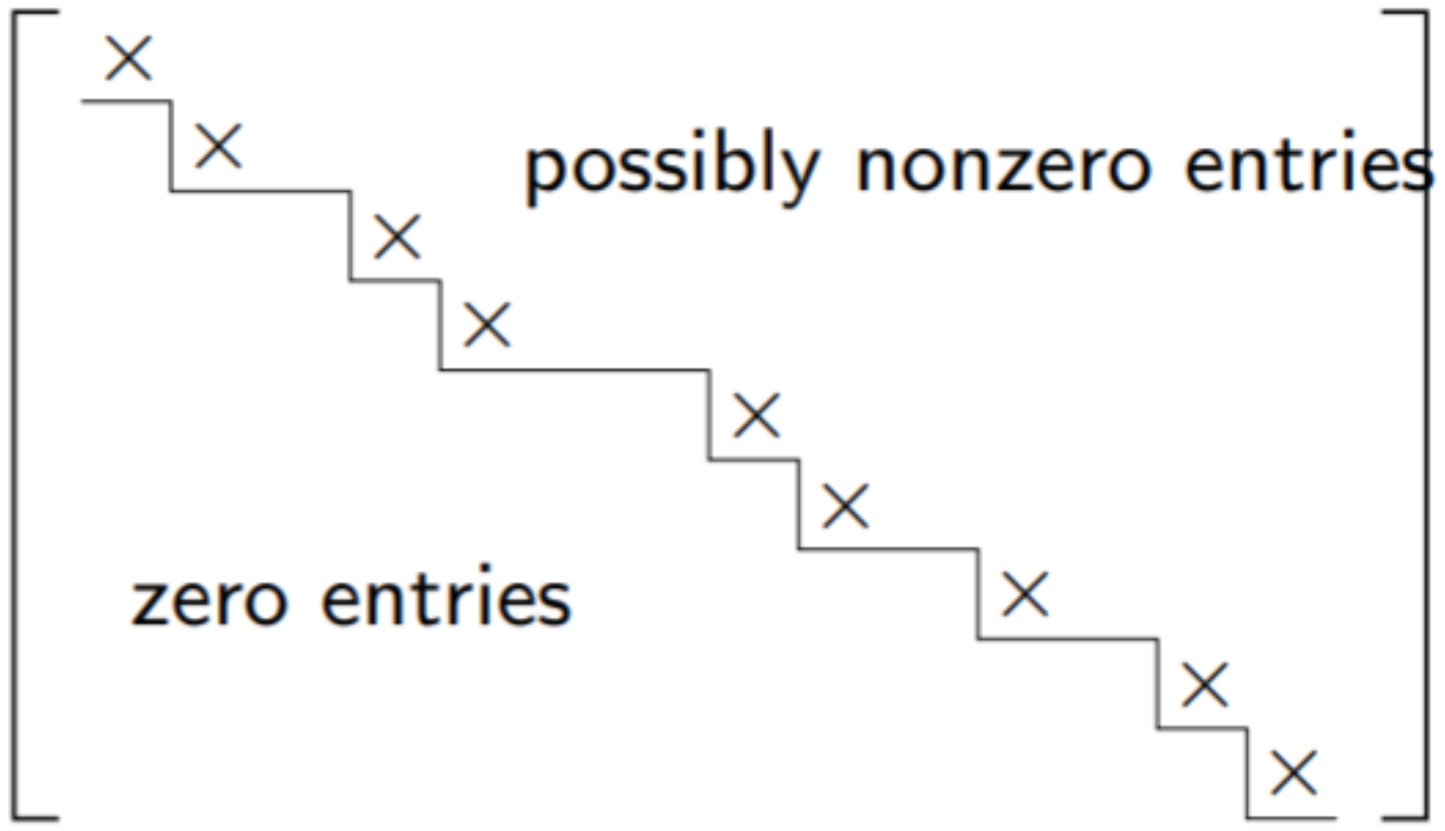
$r = 0$
**for** $i = 1, \ldots, n$
    $\tilde{a} = a_i - \sum_{j=1}^{r} q_j q_j^T a_i$
    **if** $\tilde{a} \neq 0$
        $r = r + 1$
        $q_r = \tilde{a}/\|\tilde{a}\|$

# Warning: modify QR when A has dependent column!(X)

on exit,

- $q_1, \ldots, q_r$ is an orthonormal basis for **range**$(A)$ (hence $r =$ **Rank**$(A)$)
- each $a_i$ is linear combination of previously generated $q_j$'s

in matrix notation we have $A = QR$ with $Q^{\mathsf{T}}Q = I$ and $R \in \mathbb{R}^{r \times n}$ in *upper staircase form*




'corner' entries (shown as $\times$) are nonzero

From: pg 4-6 http://ee263.stanford.edu/lectures/qr.pdf

# How to get full QR decomposition when A is tall and skinny? (X)

**'Full' $QR$ factorization**

with $A = Q_1 R_1$ the $QR$ factorization as above, write

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

where $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ is orthogonal, *i.e.*, columns of $Q_2 \in \mathbb{R}^{m \times (m-r)}$ are orthonormal, orthogonal to $Q_1$

to find $Q_2$:

- find any matrix $\tilde{A}$ s.t. $\begin{bmatrix} A & \tilde{A} \end{bmatrix}$ has rank $m$ (*e.g.*, $\tilde{A} = I$)
- apply general Gram-Schmidt to $\begin{bmatrix} A & \tilde{A} \end{bmatrix}$
- $Q_1$ are orthonormal vectors obtained from columns of $A$
- $Q_2$ are orthonormal vectors obtained from extra columns ($\tilde{A}$)

*i.e.*, any set of orthonormal vectors can be *extended* to an orthonormal basis for $\mathbb{R}^m$

# look ahead: $QQ^T$ relationship to Least Squares

Ref: relating $QQ^T$ to least squares solution (see Sec 7.1.4)
and Boyd's lecture:
https://see.stanford.edu/materials/lsoeldsee263/05-ls.pdf Pg 5-8

## Least-squares via $QR$ factorization

- $A \in \mathbf{R}^{m \times n}$ skinny, full rank
- factor as $A = QR$ with $Q^TQ = I_n$, $R \in \mathbf{R}^{n \times n}$ upper triangular, invertible
- pseudo-inverse is

$$(A^TA)^{-1}A^T = (R^TQ^TQR)^{-1}R^TQ^T = R^{-1}Q^T$$

so $x_{\rm ls} = R^{-1}Q^Ty$

- projection on $\mathcal{R}(A)$ given by matrix

$$A(A^TA)^{-1}A^T = AR^{-1}Q^T = QQ^T$$

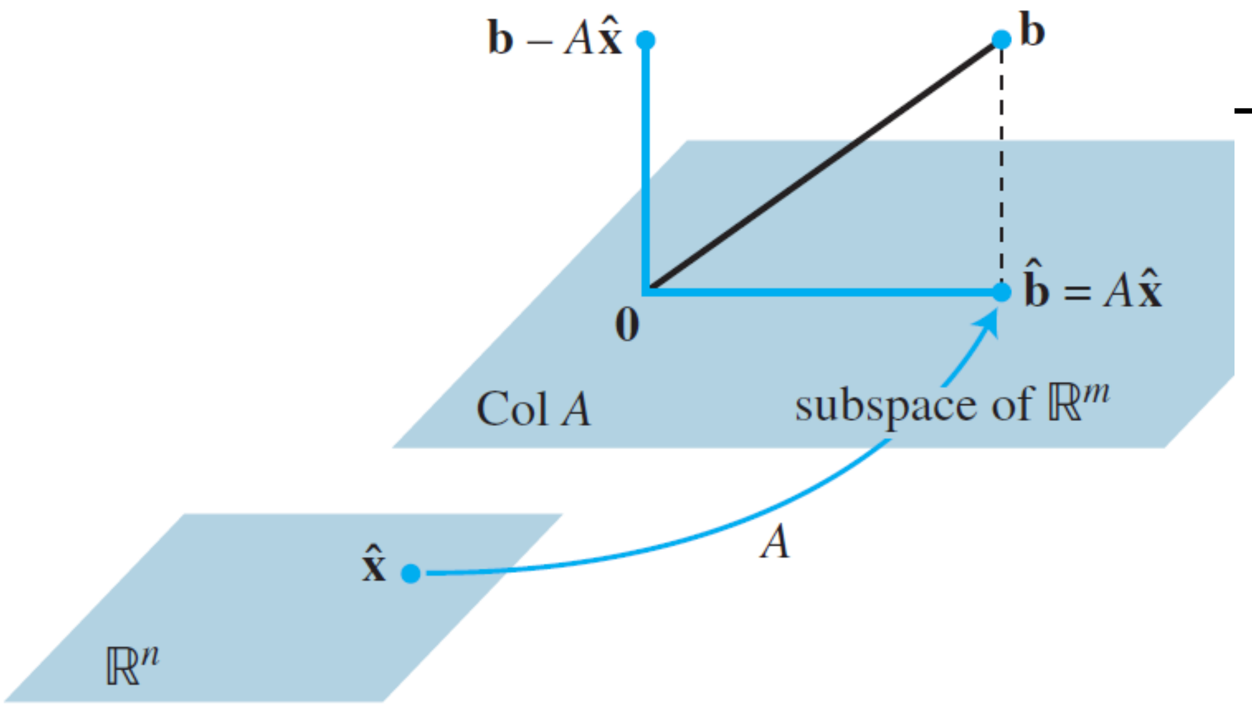



**FIGURE 2** The least-squares solution $\hat{\mathbf{x}}$ is in $\mathbb{R}^n$.

THEOREM 14

Let $A$ be an $m \times n$ matrix. The following statements are logically equivalent:

a. The equation $A\mathbf{x} = \mathbf{b}$ has a unique least-squares solution for each $\mathbf{b}$ in $\mathbb{R}^m$.

b. The columns of $A$ are linearly indpendent.

c. The matrix $A^TA$ is invertible.

When these statements are true, the least-squares solution $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}} = (A^TA)^{-1}A^T\mathbf{b} \tag{4}$$

Therefore if $\hat{\mathrm{x}} = (\mathrm{A}^TA)^{-1}A^Tb$, then
$\hat{b} = A\hat{x}$, means
$\hat{b} = A(\mathrm{A}^TA)^{-1}A^T\ b$

And this matrix $A(A^TA)^{-1}A^T$ is a projection matrix, projecting $b$ into the column space of A.

See 7.1.3 (pg 4) : projection matrix of least squares

# Appendix: Proof that $UU^T$ is a projection matrix (X)

2. (a) Let $\mathbf{u} \in \mathbb{R}^n$ be of unit length. Argue that the matrix $\mathbf{u}\mathbf{u}^T$ represents the projection on span$\{\mathbf{u}\}$.

Let $\mathbf{y} \in \mathbb{R}^n$. Observe then that

$$\begin{aligned}
\text{proj}_{\mathbf{u}}(\mathbf{y}) &= \left(\frac{\mathbf{u}.\mathbf{y}}{\mathbf{u}.\mathbf{u}}\right)\mathbf{u} \\
&= (\mathbf{u}.\mathbf{y})\mathbf{u} \qquad (\because \mathbf{u}.\mathbf{u} = 1) \\
&= \mathbf{u}(\mathbf{u}.\mathbf{y}) \qquad (\because \mathbf{u}.\mathbf{y} \in \mathbb{R}) \\
&= \mathbf{u}(\mathbf{u}^T\mathbf{y}) \\
&= (\mathbf{u}\mathbf{u}^T)\mathbf{y}. \qquad (\because \text{associativity of matrix multiplication})
\end{aligned}$$

This shows that $\text{proj}_{\mathbf{u}} = \mathbf{u}\mathbf{u}^T$.

(b) Let $W$ be a subspace of $\mathbb{R}^n$ and let $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\}$ be an orthonormal basis for $W$. Argue that the matrix

$$P = \mathbf{u}_1\mathbf{u}_1^T + \mathbf{u}_2\mathbf{u}_2^T + \ldots + \mathbf{u}_p\mathbf{u}_p^T$$

is precisely the projection onto $W$.

In fact, if $U$ is the matrix $(\mathbf{u}_1 \mid \mathbf{u}_2 \mid \ldots \mid \mathbf{u}_p)$, then we can also write this projection as $UU^T$.

Let $\mathbf{y} \in \mathbb{R}^n$. Replicating the steps in (a), we have

$$\begin{aligned}
\text{proj}_W(\mathbf{y}) &= \left(\frac{\mathbf{u}_1.\mathbf{y}}{\mathbf{u}_1.\mathbf{u}_1}\right)\mathbf{u}_1 + \left(\frac{\mathbf{u}_2.\mathbf{y}}{\mathbf{u}_2.\mathbf{u}_2}\right)\mathbf{u}_2 + \ldots + \left(\frac{\mathbf{u}_p.\mathbf{y}}{\mathbf{u}_p.\mathbf{u}_p}\right)\mathbf{u}_p \\
&= (\mathbf{u}_1.\mathbf{y})\,\mathbf{u}_1 + (\mathbf{u}_2.\mathbf{y})\,\mathbf{u}_2 + \ldots + (\mathbf{u}_p.\mathbf{y})\,\mathbf{u}_p \\
&= \mathbf{u}_1\left(\mathbf{u}_1^T\mathbf{y}\right) + \mathbf{u}_2\left(\mathbf{u}_2^T\mathbf{y}\right) + \ldots + \mathbf{u}_p\left(\mathbf{u}_p^T\mathbf{y}\right) \\
&= (\mathbf{u}_1\mathbf{u}_1^T)\mathbf{y} + (\mathbf{u}_2\mathbf{u}_2^T)\mathbf{y} + \ldots + (\mathbf{u}_p\mathbf{u}_p^T)\mathbf{y} \\
&= (\mathbf{u}_1\mathbf{u}_1^T + \mathbf{u}_2\mathbf{u}_2^T + \ldots + \mathbf{u}_p\mathbf{u}_p^T)\mathbf{y} \\
&= P\mathbf{y}.
\end{aligned}$$

This proves that $\text{proj}_W = P$.

Ref: https://math.berkeley.edu/~qadeer/teaching/F15Math54/Worksheet%204%20Solutions.pdf

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}_{m\times n}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}}_{x} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{m\times 1}}_{b}$$

# CX1104: **Linear Algebra for Computing**

Chap. No : **6.3.3**

Lecture : **Orthogonality**

Topic : **Gram–Schmidt Process**

Concept : **Using Matlab to get QR and 4 cases of A**

Instructor: **A/P Chng Eng Siong**

TAs: **Zhang Su, Vishal Choudhari**

Last updated: 19 Sep 2021

Rev: 29th June 2020

# Using Matlab for QR

We will consider only the two common cases:

i) A is a square matrix
ii) A has dimension mxn (m>n)

Using Matlab, there are 2 options, complete (full) vs economy decomposition.

Ref: 1) [https://en.wikipedia.org/wiki/QR_decomposition](https://en.wikipedia.org/wiki/QR_decomposition)
2) http://ee263.stanford.edu/lectures/qr.pdf

## Matlab

```
qr     Orthogonal-triangular decomposition.
   [Q,R] = qr(A), where A is m-by-n, produces an m-by-n upper triangular
   matrix R and an m-by-m unitary matrix Q so that A = Q*R.

   [Q,R] = qr(A,0) produces the "economy size" decomposition.
   If m>n, only the first n columns of Q and the first n rows of R are
   computed. If m<=n, this is the same as [Q,R] = qr(A).
```

## numpy.linalg.qr

`linalg.qr(a, mode='reduced')`

Compute the qr factorization of a matrix.

Factor the matrix *a* as *qr*, where *q* is orthonormal and *r* is upper-triangular.

Parameters: **a** : *array_like, shape (M, N)*
Matrix to be factored.

**mode** : *{'reduced', 'complete', 'r', 'raw'}, optional*
If K = min(M, N), then
- 'reduced' : returns q, r with dimensions (M, K), (K, N) (default)
- 'complete' : returns q, r with dimensions (M, M), (M, N)

Note: option for numpy 'reduced' == matlab 'economy' when performing QR

# QR using Matlab and the 4 cases

We explore QR of A for the following 4 cases

1) Square A matrix (size 3x3)
   - Ex1 ) where A has 3 independent col
   - Ex2 ) where A has 2 independent col, and 1 dependent col

2) Tall A matrix (size 3x2)
   - Ex3 ) where A has 2 independent col
   - Ex4 ) where A col are dependent (col2 == 2xcol1)

Study the effect decomposition has on Q and R, as well
as selection of Matlab economy vs complete (full) QR decomposition.

# Introducing $A = QR$ for Square A

## Example 1: A is 3x3 square and has independent column

### Square matrix [ edit ]

Any real square matrix $A$ may be decomposed as

$$A = QR,$$

where $Q$ is an orthogonal matrix (its columns are orthogonal unit vectors meaning $Q^\mathsf{T} = Q^{-1}$) and $R$ is an upper triangular matrix (also called right triangular matrix). If $A$ is invertible, then the factorization is unique if we require the diagonal elements of $R$ to be positive.

If instead $A$ is a complex square matrix, then there is a decomposition $A = QR$ where $Q$ is a unitary matrix (so $Q^* = Q^{-1}$).

If $A$ has $n$ linearly independent columns, then the first $n$ columns of $Q$ form an orthonormal basis for the column space of $A$. More generally, the first $k$ columns of $Q$ form an orthonormal basis for the span of the first $k$ columns of $A$ for any $1 \le k \le n$.[1] The fact that any column $k$ of $A$ only depends on the first $k$ columns of $Q$ is responsible for the triangular form of $R$.[1]

```
>> A = [1 2 3; 1 2 1; 1 1 3]

A =

     1     2     3
     1     2     1
     1     1     3

>> rank(A)

ans =

     3
```

```
>> [Q,R] = qr(A)

Q =

   -0.5774   -0.4082   -0.7071
   -0.5774   -0.4082    0.7071
   -0.5774    0.8165   -0.0000


R =

   -1.7321   -2.8868   -4.0415
         0   -0.8165    0.8165
         0         0   -1.4142
```

```
>> rank(R)

ans =

     3
```

```
>> Q*R

ans =

    1.0000    2.0000    3.0000
    1.0000    2.0000    1.0000
    1.0000    1.0000    3.0000
```

```
>> Q'*Q

ans =

    1.0000    0.0000    0.0000
    0.0000    1.0000   -0.0000
    0.0000   -0.0000    1.0000

>> Q*Q'

ans =

    1.0000   -0.0000   -0.0000
   -0.0000    1.0000   -0.0000
   -0.0000   -0.0000    1.0000
```

Matlab Notation:

$$Q' == Q^T$$

When A is square and has independent column, factorizing A to QR, then
Q'*Q == Q*Q' == identity matrix
And R which is square is invertible!

# Square A with dependent col

Example 2: A is 3x3 square and
A has 1 dependent column

```
>> A = [1 2 3; 1 2 3; 1 1 2]

A =

     1     2     3
     1     2     3
     1     1     2

>> rank(A)

ans =

     2
```

```
>> [Q,R] = qr(A)

Q =

   -0.5774   -0.4082   -0.7071
   -0.5774   -0.4082    0.7071
   -0.5774    0.8165   -0.0000


R =

   -1.7321   -2.8868   -4.6188
         0   -0.8165   -0.8165
         0         0   -0.0000
```

```
>> Q*R

ans =

    1.0000    2.0000    3.0000
    1.0000    2.0000    3.0000
    1.0000    1.0000    2.0000
```

```
>> rank(R)

ans =

     2
```

```
>> Q'*Q

ans =

    1.0000    0.0000    0.0000
    0.0000    1.0000   -0.0000
    0.0000   -0.0000    1.0000

>> Q*Q'

ans =

    1.0000   -0.0000   -0.0000
   -0.0000    1.0000   -0.0000
   -0.0000   -0.0000    1.0000
```

When A is square and has dependent column,
It still can be decomposed into QR, and
Q’*Q == Q*Q’ == identity matrix
BUT now, R which is square is NOT invertible!
R has rank 2 bcos A has 2 independent col.

Ref: https://en.wikipedia.org/wiki/QR_decomposition

# Introducing $A = QR$ for Tall A $(m>n)$

**Rectangular matrix** [ edit ]

More generally, we can factor a complex $m \times n$ matrix $A$, with $m \geq n$, as the product of an $m \times m$ unitary matrix $Q$ and an $m \times n$ upper triangular matrix $R$. As the bottom $(m-n)$ rows of an $m \times n$ upper triangular matrix consist entirely of zeroes, it is often useful to partition $R$, or both $R$ and $Q$:

$$A = QR = Q\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1,$$

where $R_1$ is an $n \times n$ upper triangular matrix, 0 is an $(m - n) \times n$ zero matrix, $Q_1$ is $m \times n$, $Q_2$ is $m \times (m - n)$, and $Q_1$ and $Q_2$ both have orthogonal columns.

Golub & Van Loan (1996, §5.2) call $Q_1R_1$ the *thin QR factorization* of $A$; Trefethen and Bau call this the *reduced QR factorization*.[1] If $A$ is of full rank $n$ and we require that the diagonal elements of $R_1$ are positive then $R_1$ and $Q_1$ are unique, but in general $Q_2$ is not. $R_1$ is then equal to the upper triangular factor of the Cholesky decomposition of $A^* A$ (= $A^\mathrm{T}A$ if $A$ is real).

Note:

col($Q_1$)== col(A),

while

col($Q_2$) = orthogonal complement of col(A)

see example 3 and 4 (in next pages)

# Tall A

$$A = QR = Q\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = [Q_1 \quad Q_2]\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1,$$

## Example 3: A is tall (3x2) and A has 2 independent column

```
A =

     1     2
     1     2
     1     1

>> [Q,R] = qr(A)

Q =

   -0.5774   -0.4082   -0.7071
   -0.5774   -0.4082    0.7071
   -0.5774    0.8165   -0.0000


R =

   -1.7321   -2.8868
         0   -0.8165
         0         0
```

```
>> Q'*Q

ans =

    1.0000    0.0000    0.0000
    0.0000    1.0000   -0.0000
    0.0000   -0.0000    1.0000

>> Q*Q'

ans =

    1.0000   -0.0000   -0.0000
   -0.0000    1.0000   -0.0000
   -0.0000   -0.0000    1.0000
```

[Q,R] = qr(A) (complete decomposition)
- Complete Q has size == 3x3 (and it an orthogonal matrix even though A is 3x2.
- R is 3x2 (row 1 and 2 of R are non-zero bcos A has 2 independent col)

$Q_1$ is the first 2 columns of Q (bcos A has 2 independent col)
$\text{col}(Q_1) == \text{col(A)}$
$Q_2$ is the last column of $Q$

```
A =

     1     2
     1     2
     1     1

>> [Q,R] = qr(A,0)

Q =

   -0.5774   -0.4082
   -0.5774   -0.4082
   -0.5774    0.8165


R =

   -1.7321   -2.8868
         0   -0.8165
```

```
>> Q'*Q

ans =

    1.0000    0.0000
    0.0000    1.0000

>> Q*Q'

ans =

    0.5000    0.5000   -0.0000
    0.5000    0.5000    0.0000
   -0.0000    0.0000    1.0000
```

[Q,R] = qr(A,0) (using economy decomposition)
reduced Q has size == 3x2.
Note that Q'*Q = I (size (2x2))
While Q*Q' = P
a projection matrix size 3x3.

A projection matrix: see Strang lecture 16
https://www.youtube.com/watch?v=osh80YCg_GM

# Tall A

$$A = QR = Q\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = [Q_1 \quad Q_2]\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1,$$

## Example 4: A is tall (3x2) and A has dependent column

```
A =

     1     2
     1     2
     1     2

>> [Q,R] = qr(A)

Q =

   -0.5774    0.8165   -0.0000
   -0.5774   -0.4082   -0.7071
   -0.5774   -0.4082    0.7071


R =

   -1.7321   -3.4641
         0   -0.0000
         0         0
```

```
>> Q*R

ans =

    1.0000    2.0000
    1.0000    2.0000
    1.0000    2.0000
```

[Q,R] = qr(A) (complete decomposition)
Complete Q has size == 3x3 even though A is 3x2
And R is 3x2 (Only row 1 of R is non-zero, bcos there is ONLY 1 independent col in A)

$Q_1$ is ONLY the first column of Q (bcos A has ONLY 1 independent col), $col(Q_1) == col(A)$
$Q_2$ is the last 2 columns of $Q$

```
A =

     1     2
     1     2
     1     2

>> [Q,R] = qr(A,0)

Q =

   -0.5774    0.8165
   -0.5774   -0.4082
   -0.5774   -0.4082


R =

   -1.7321   -3.4641
         0   -0.0000
```

```
>> Q*R

ans =

    1.0000    2.0000
    1.0000    2.0000
    1.0000    2.0000
```

[Q,R] = qr(A,0) (economy decomposition)
reduced Q has size == 3x2