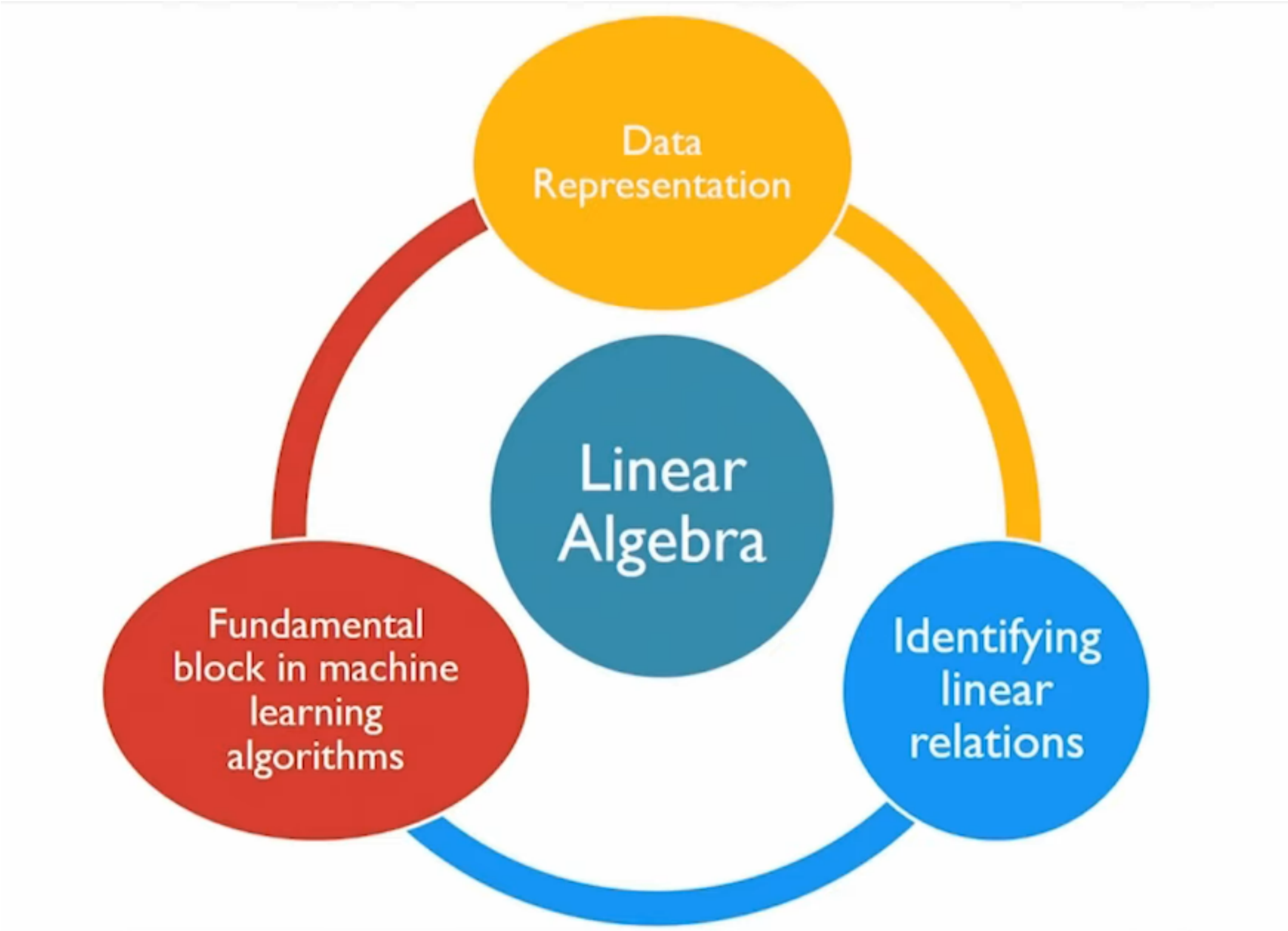
let's start with a question..

What linear algebra is useful for?

(When one talks about data science..)

1. **Data representation** becomes an important aspect.
   - Data is represented in a matrix form. *So, we talk about this representation and concepts*.
2. *(In the data science perspective..)* If the data contains several variables of interest, one might want to know **How many of these variables are really important?**, and **If there are any relationships, how can one uncover those?**
3. The ideas from linear algebra becomes fundamental concept in Machine Learning Algorithms.

these important questions are answered by the Linear Algebra.



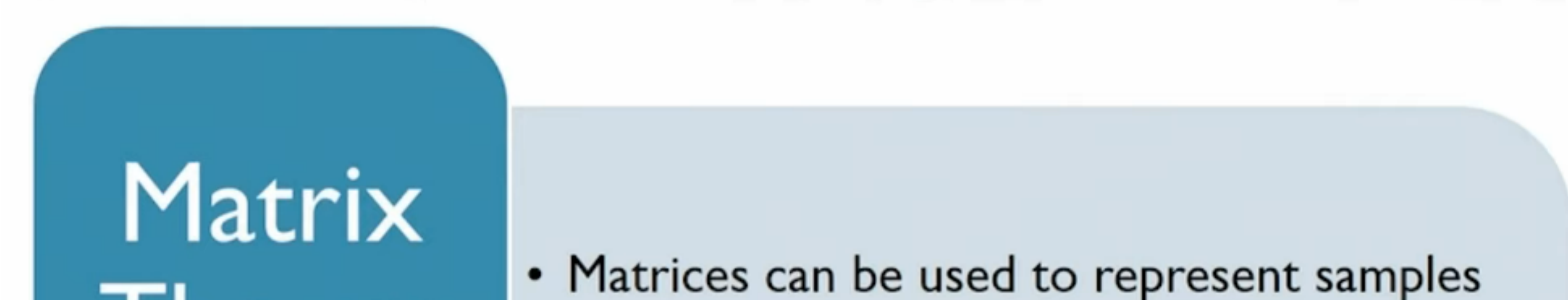
Each of the above is addressed in further notes..

Let's start with the matrices.. and summarize the points which are relevant to the data science perspective **What is a matrix?**

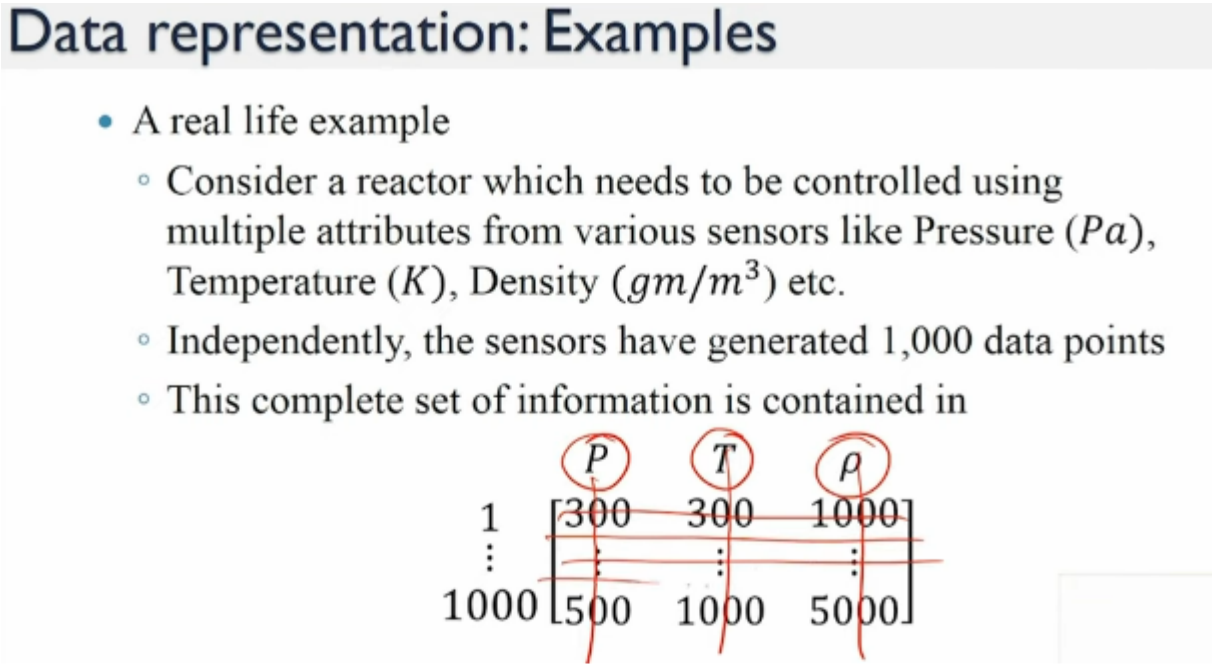
> A matrix is a form of organizing data into rows and columns.

There are many ways of representing the data, matrix provides a convinient way..

- If you are an engineer, and you deal with multiple variables, how do you store them so that you can use later?
- Matrices can also be used to store the coefficients of the equations

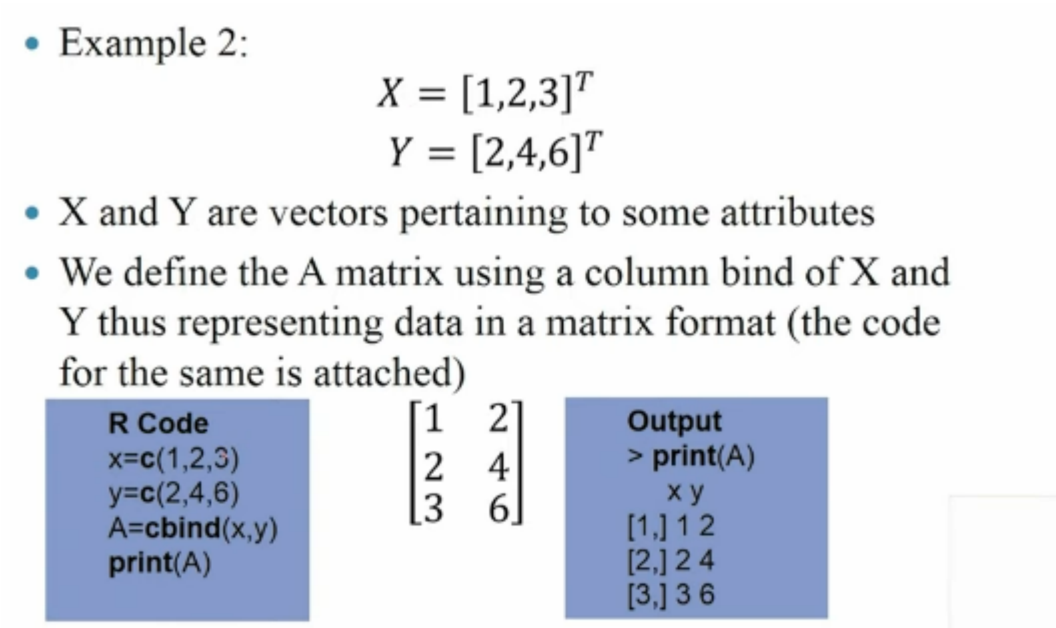


Data Representation: Examples

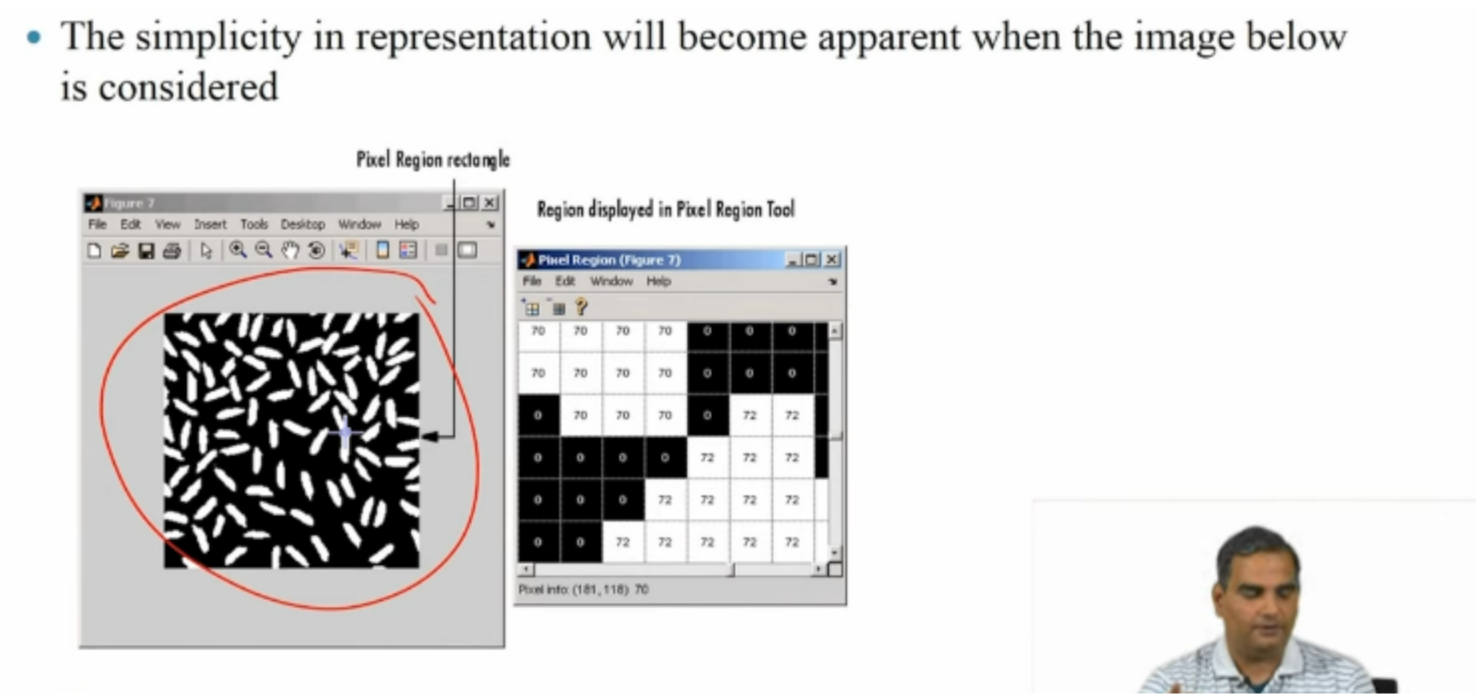


each row is a sample data.

A vector of attibutes..

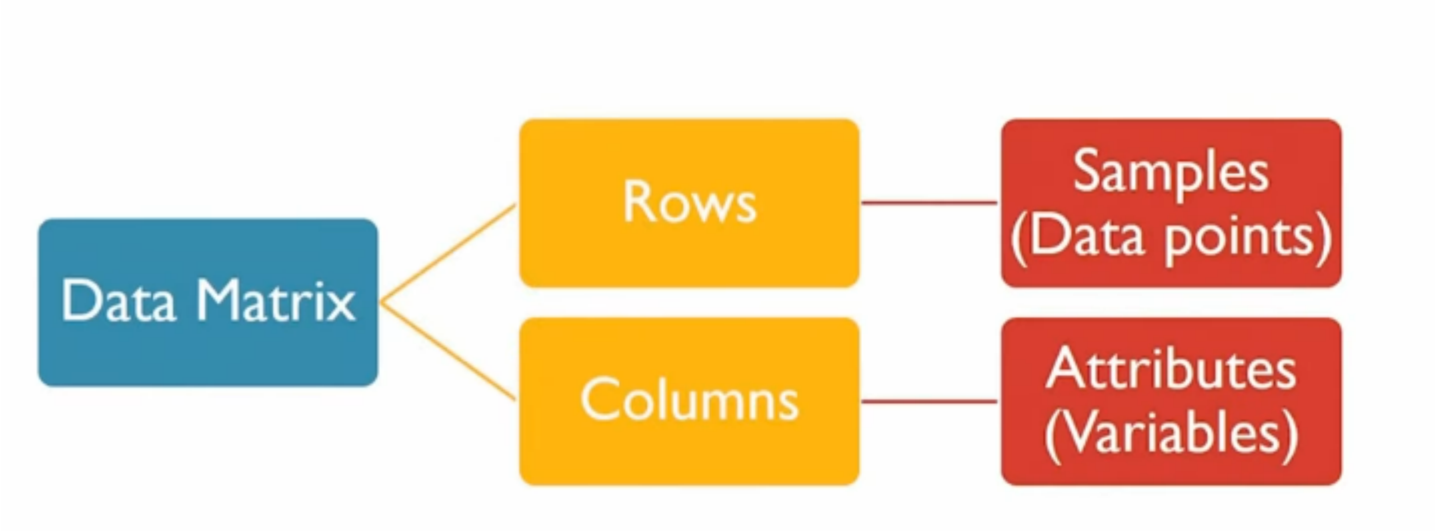


Storing images..



and answering, whether two images same or identify the sub-components in the image..... all of these can be performed through some form of matrix manipulation.

Data as matrix - Summary



now coming to the second aspect..

Identification of Idependent attributes

- One might be interested in knowing, if all the variables that **are there are imporant?** *(in other words..)*
- In all these variables, **how many of these are independent?**, so that, if contains (say) 1000's of variables and there are 4-5 independent variables, then one can store these variables and calculate the remaining as a function of these variables.

These lead to the following questions..

- Now that we can represent the data into a matrix format, we ask the following questions
  - Are all the attributes in the data matrix relevant/ important?
  - Is there any method which can identify if some attributes are related to the other attributes?
  - If yes, how do we identify the linear relationship?
  - Can we use this to reduce the size of the data matrix?

If could find those independent variables, can drop the remaining. --- This is an important aspect in Machine Learning.

Identification of independent attributes: Example

- Consider the ideal reactor example with multiple (say, 4) attributes like Pressure, Temperature, Density, Viscosity, etc. with 500 samples.
- Thus we have a 500 X 4 matrix such that
$$A = [P\ T\ D\ \eta]$$
- P, T, D and η are vectors of 500 samples from the pressure, temperature, density and viscosity sensors.
- How does one identify the number of independent attributes?

Identification of independent attributes: Example

- Domain knowledge

$$D \sim f(P,T)$$

- Thus, in some sense **D** is a function of **P** and **T**

How can one identify the independent attributes? - Rank of matrix

- Let us assume that we have many more samples than attributes for now
- Is there any approach which can be used to identify the number of linear relationships between the attributes purely using data?
- This is addressed by the concept of the **rank** of the matrix.
- **Rank** of a matrix refers to the number of linearly independent rows or columns of the matrix
- The rank of a matrix can be found using the rank command: rank(A)

Rank of matrix:

> Number of linearly independent rows or columns in a matrix. Denoted as `rank(A)`

See, the below matrix, its deliberately generated in such a way that, column-2 depends on column-1.

- Consider another example

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 0 \\ 3 & 6 & 0 \end{bmatrix}$$

- We observe that
  - (Col. 2)=2 x (Col. 1)
  - (Col. 3) is independent
- Thus, the rank of this matrix is 2

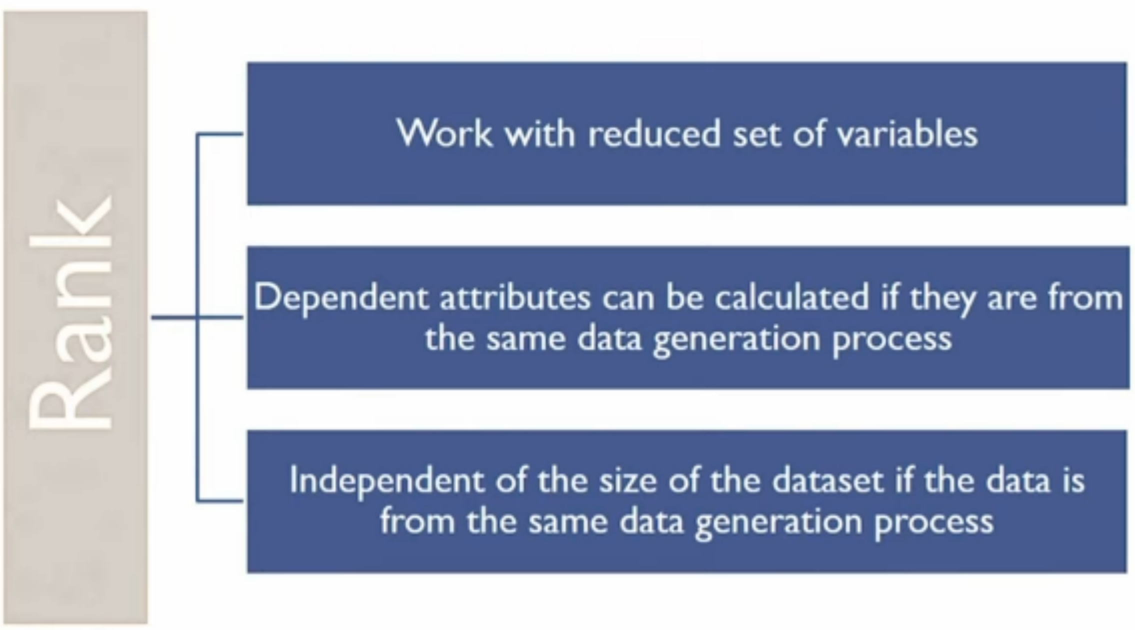
R Code

```
A=matrix(c(1,2,3,2,4,6,1,0,0),ncol=3,byrow=F)
library(pracma)
Rank(A)
```

Output

```
> Rank(A)
[1] 2
```

Rank: Advantages and summary



ok, now we come to know the **no. of indpendent variables, but how about the relationship with the dependent variables...??** *--this further motivates as ...*

Identification of Linear relationships among attributes - *via concept of **Null Space** and **Nullity***

NullSpace of a matrix

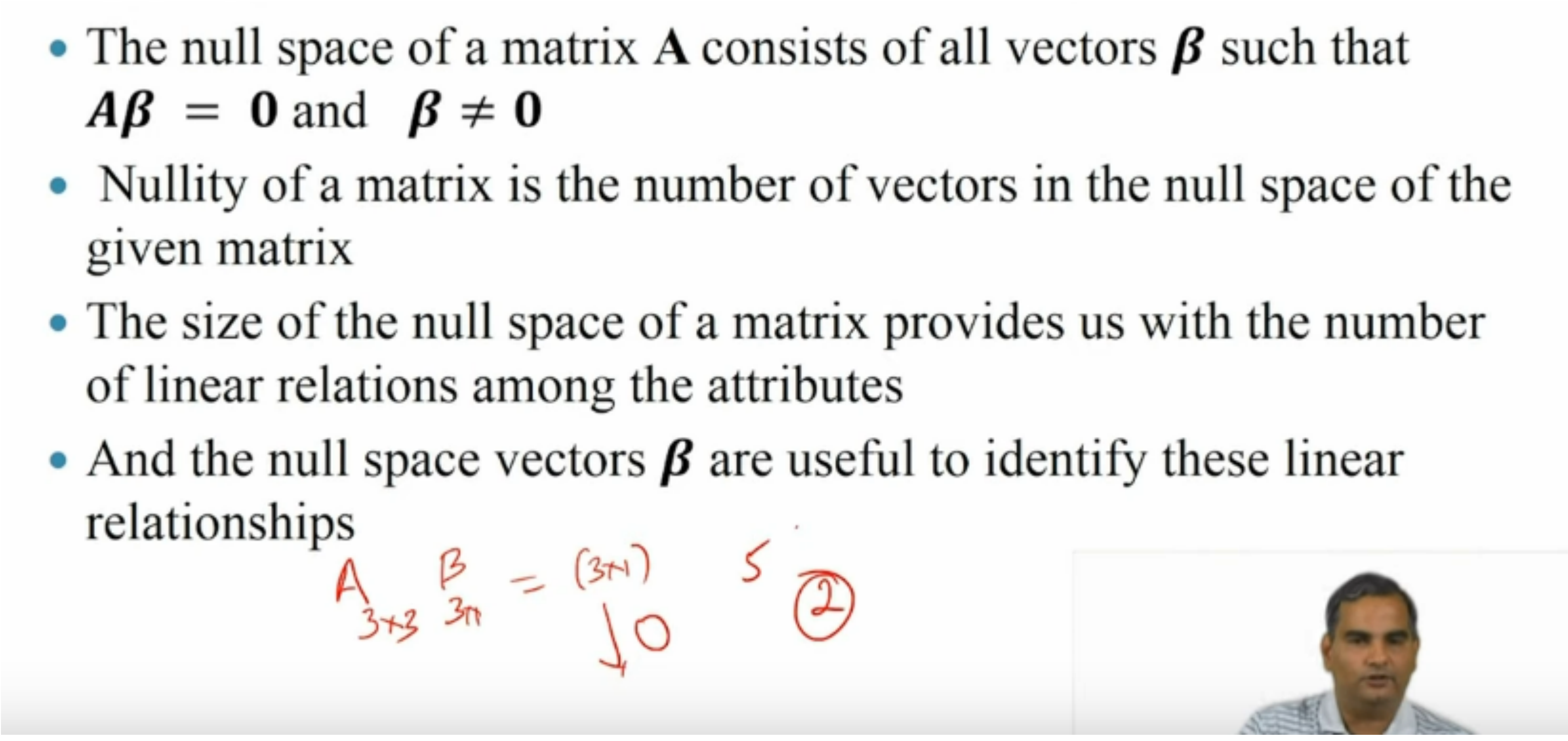
> The null space of a matrix A consists of all vectors β, such that $A\beta = 0$ and $\beta = 0$. Then β is called the null space of the matrix.
>
> > **What does $A\beta = 0$ mean?** *its addressed in the next section formally. (intuitively explained in the below slide picture)..*

Interestingly, **the size of the null space of a matrix provides us the *no. of linear relations* among the attributes.**

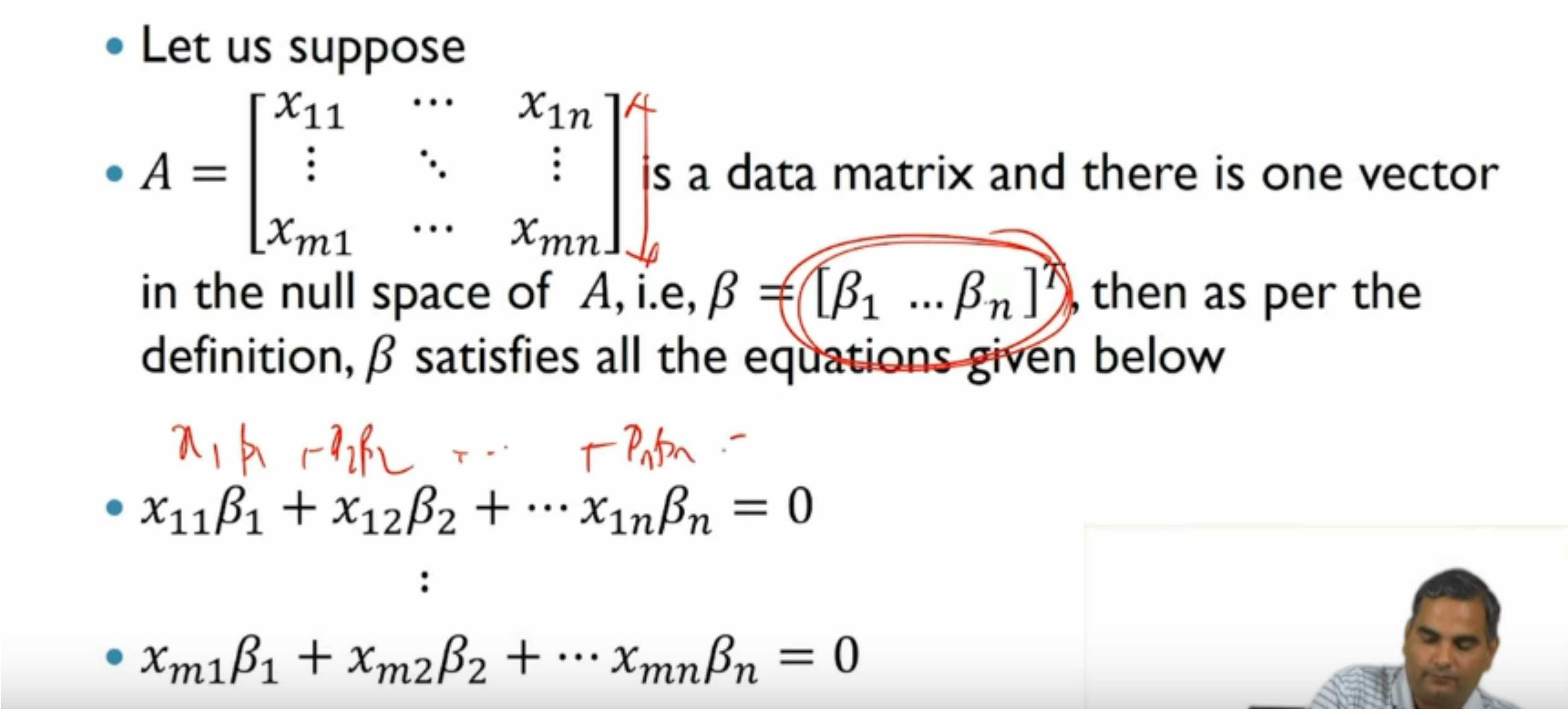
> i.e., if a matrix having dimensions 5, then if we got a null space matrix's size as `2`, then it implies that `5-2=3` are linearly independent.*(The other two depend on these, so got nullified).*

Nullity of a matrix

> Number of vectors in the null space of the given matrix -- *(from understanding [Please correct if not]..it tells **no. of dependent variables in the matrix**)*



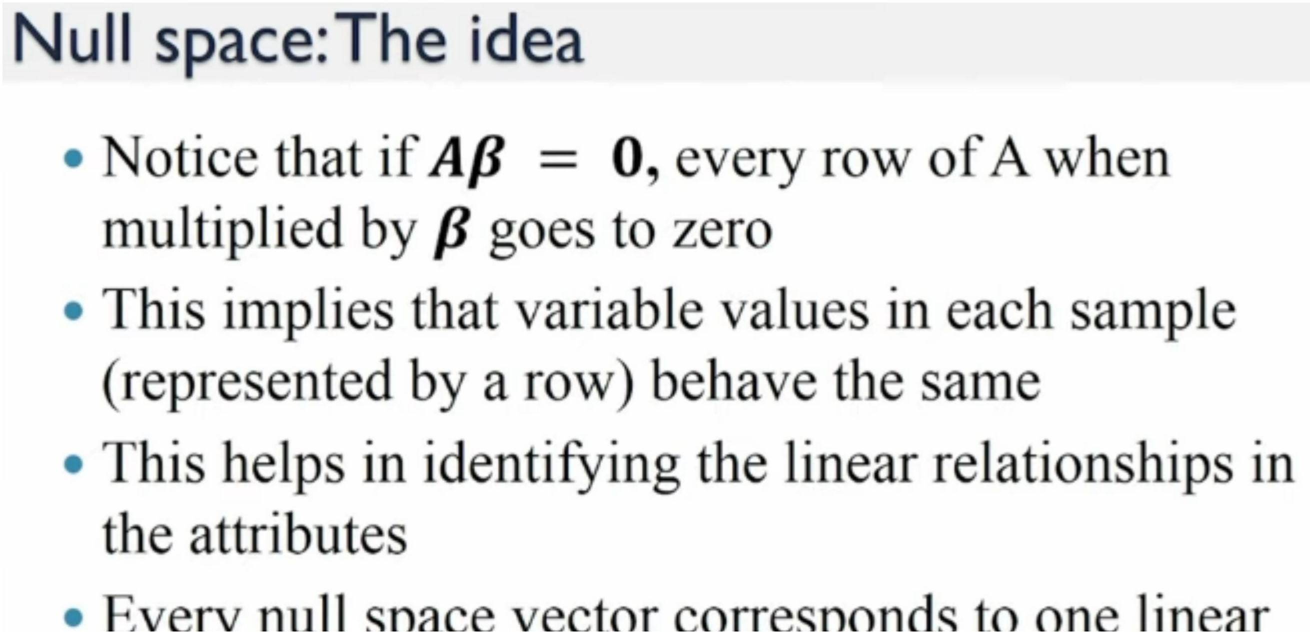
NullSpace: General Description



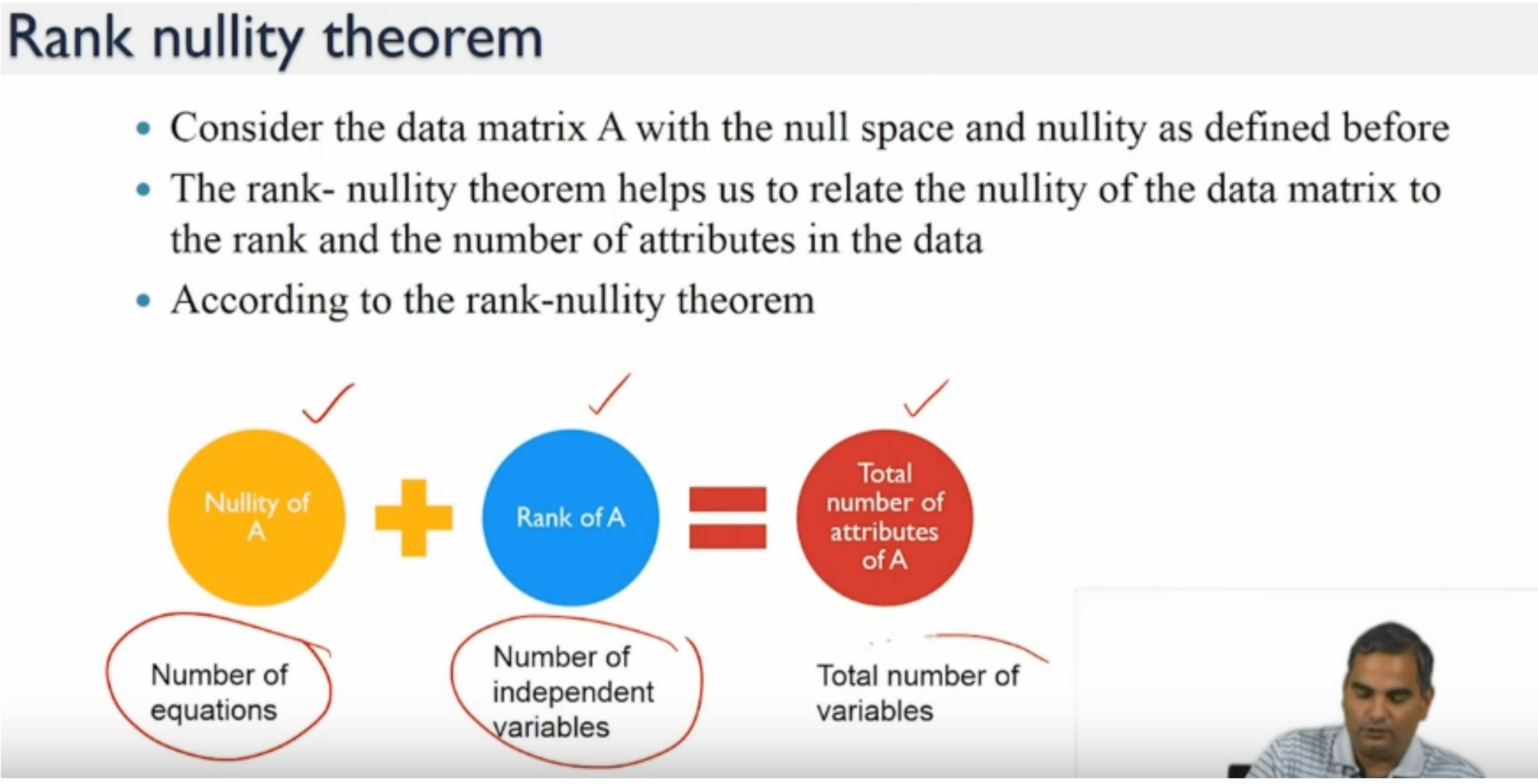
those equations denote the $A\beta = 0$.. hope not got clear..!! *(Referred from the dobut in above cell)*

- and the generic form is written above all the equations in the red-ink. It can be interpreted as..

  > Where you can take any sample and substitute the values of variables in that $x_1,\ x_2, ...x_n$, and this is to be satisfied, its a true relationship.



It can be demonstrated as..

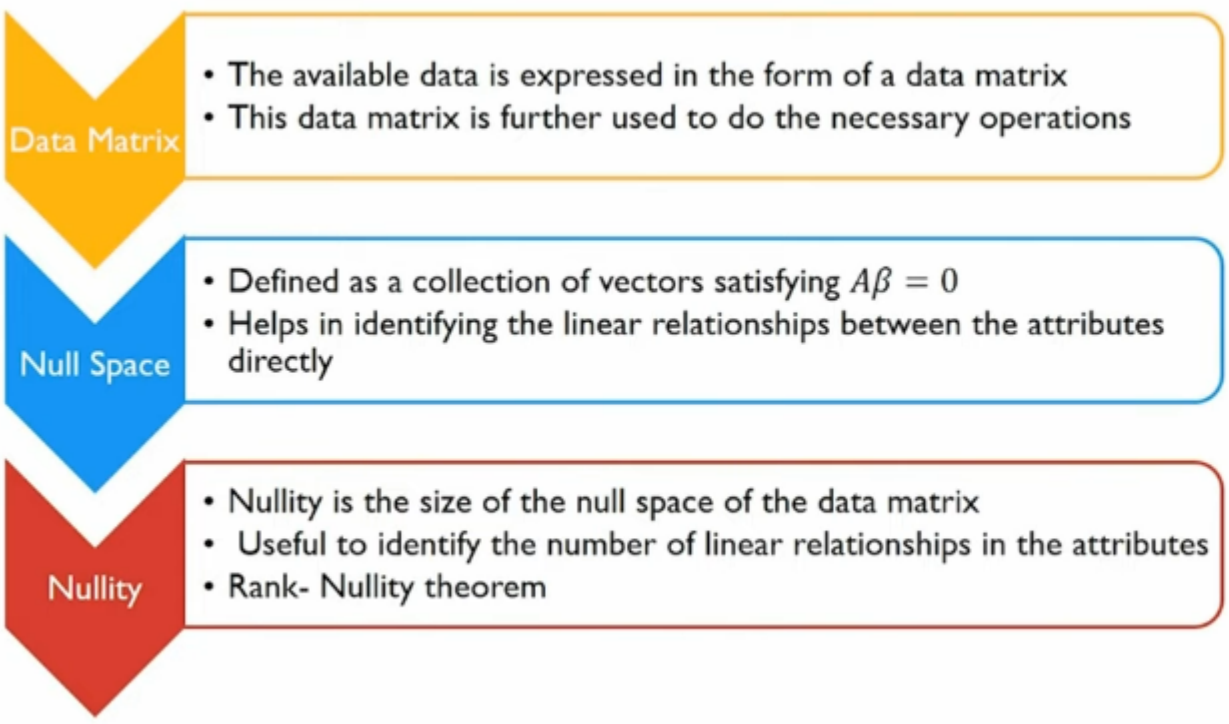


Why *"Nullity of A"* means *"Number of equations"*? Think..!!

> What does Nullity of A tells? *(from understanding, please correct if not)* it tells the no of dependent variables. So, how can we get those back after dropped, via independent variables? -- via equation right..??. So, if had n no. of dependent variables, then we need same n no. of equations to get those right..??

NOTE: -- its purely based on understanding, by grace. Correct if found to be incorrect.

Summary



Null space: An Example

- Consider the matrix A with attributes $\{x_1, x_2\}$

$$\begin{bmatrix}1 & 2\\3 & 4\\5 & 6\end{bmatrix}$$

Number of columns in A =2

Rank of A =2

Thus, nullity =0

- This implies that the null space of the matrix A does not contain any vectors
- Thus we can claim that all the attributes are linearly independent

Console output

```
> A
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

Console output

```
>print(columns)
[1] 2
> print(rank)
[1] 2
> print(nullity)
[1] 0
```

R Code

```
A=matrix(c(1,3,5,2,4,6),ncol=2, byrow=F)
columns=ncol(A)
library(pracma)
rank=Rank(A)
nullity=columns-rank
```

Linear Algebra

Null space: Another example

- Now consider A with attributes $\{x_1, x_2, x_3\}$ such that

$$\begin{bmatrix}1 & 2 & 0\\2 & 4 & 0\\3 & 6 & 1\end{bmatrix}$$

Number of columns in A =3

Rank of A =2

Thus, nullity =1

- Thus, we need to identify the vectors in the null space of A which is non-zero in this case

R Code

```
A=matrix(c(1,2,3,2,4,6,0,0,1),ncol=3, byrow=F)
columns=ncol(A)
library(pracma)
rank=Rank(A)
nullity=columns-rank
```

Console output

```
> columns
[1] 3
> rank
[1] 2
> nullity
[1] 1
```

Null space: Further Example

$$A\boldsymbol{\beta} = 0$$

$$\begin{bmatrix}1 & 2 & 0\\2 & 4 & 0\\3 & 6 & 1\end{bmatrix}\begin{bmatrix}b_1\\b_2\\b_3\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

$A\beta = 0$

$c\beta$

$Ac\beta = 0$

$cA\beta = 0$

- Thus we obtain,

$$b_1 + 2b_2 = 0$$
$$b_3 = 0$$

$b_1 = -2b_2$

- The null vector is $\boldsymbol{B}=[b_1\ b_2\ b_3\ ]^T=[-2b_2\ b_2\ 0]^T=k[-2\ 1\ 0]^T$
- We see that we obtain a direct linear relationship between the attributes of A using null space and rank-nullity theorem

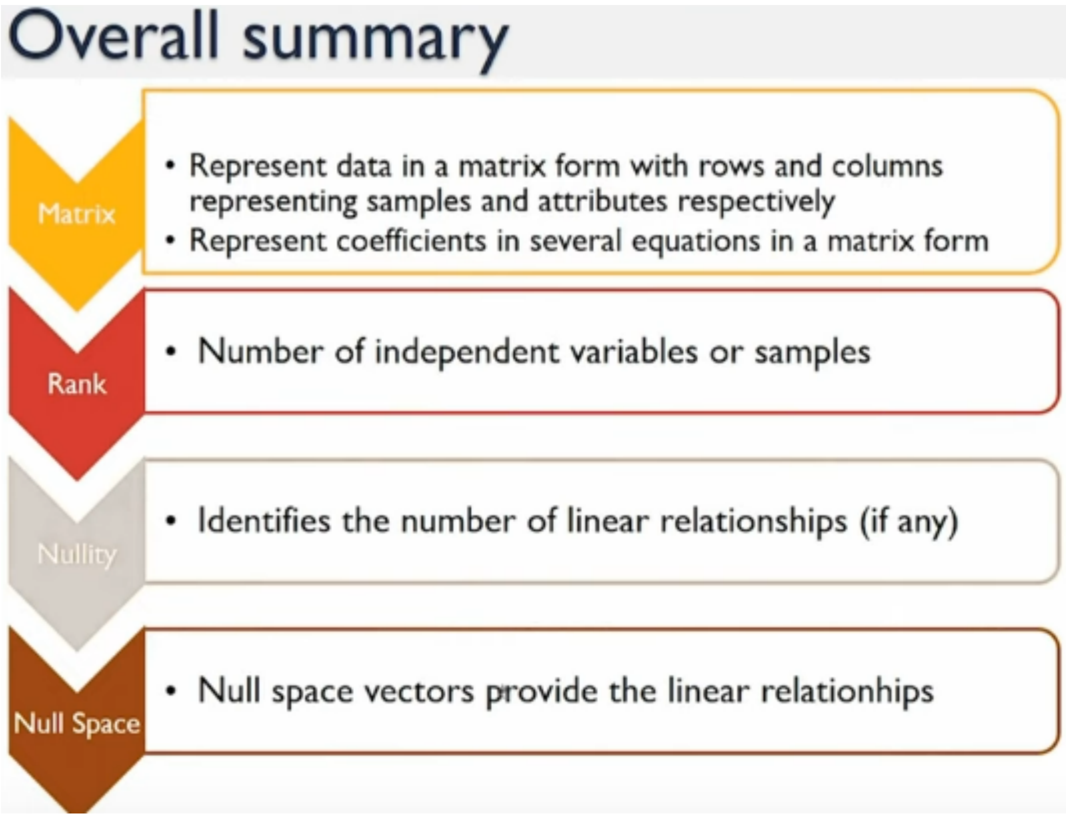
$-2x_1 + x_2 = 0$

- The same concept can be extended for bigger data set

-- Need workout on this.

- R1's clear..
- The 2nd row's equation was dropped, as its the multiple of R1.
- R3's, how about 3 and 6..??

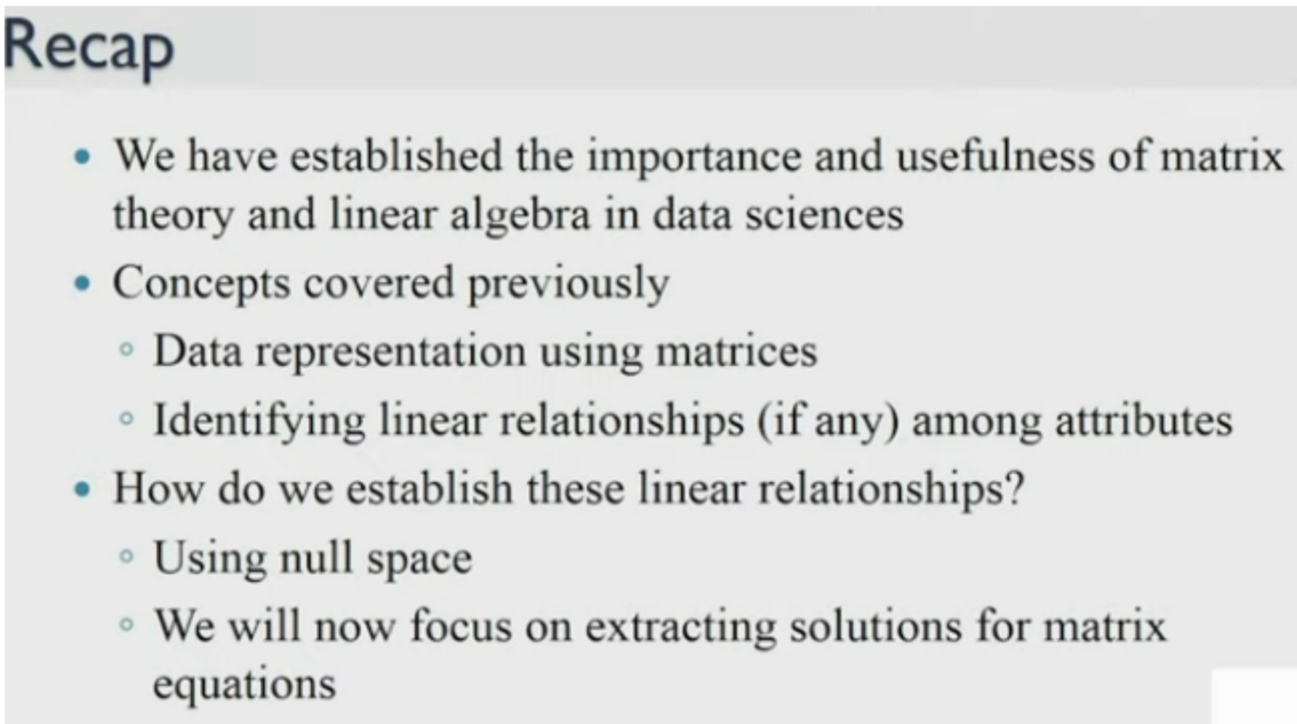
Overall Summary



End of Module-1

Module-2: **Solving Linear Equations**

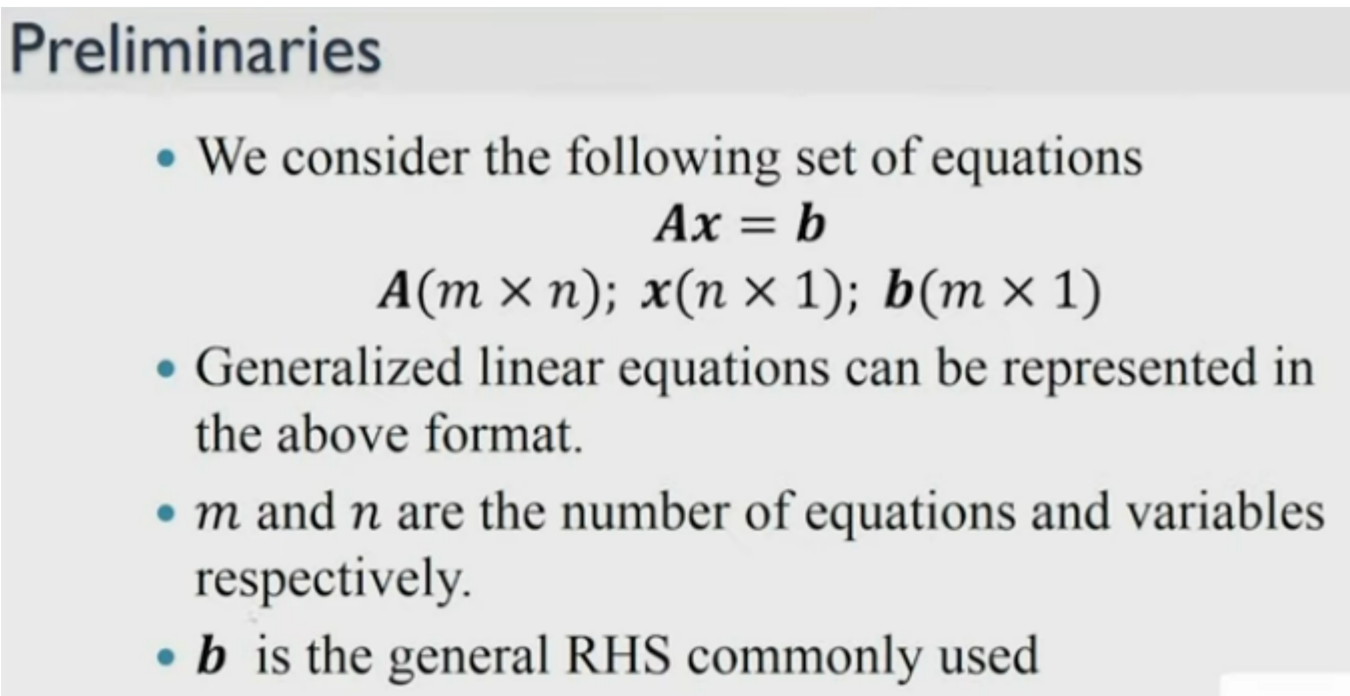
```
-- on 24th August, 2021 ~ Tuesday_4
```



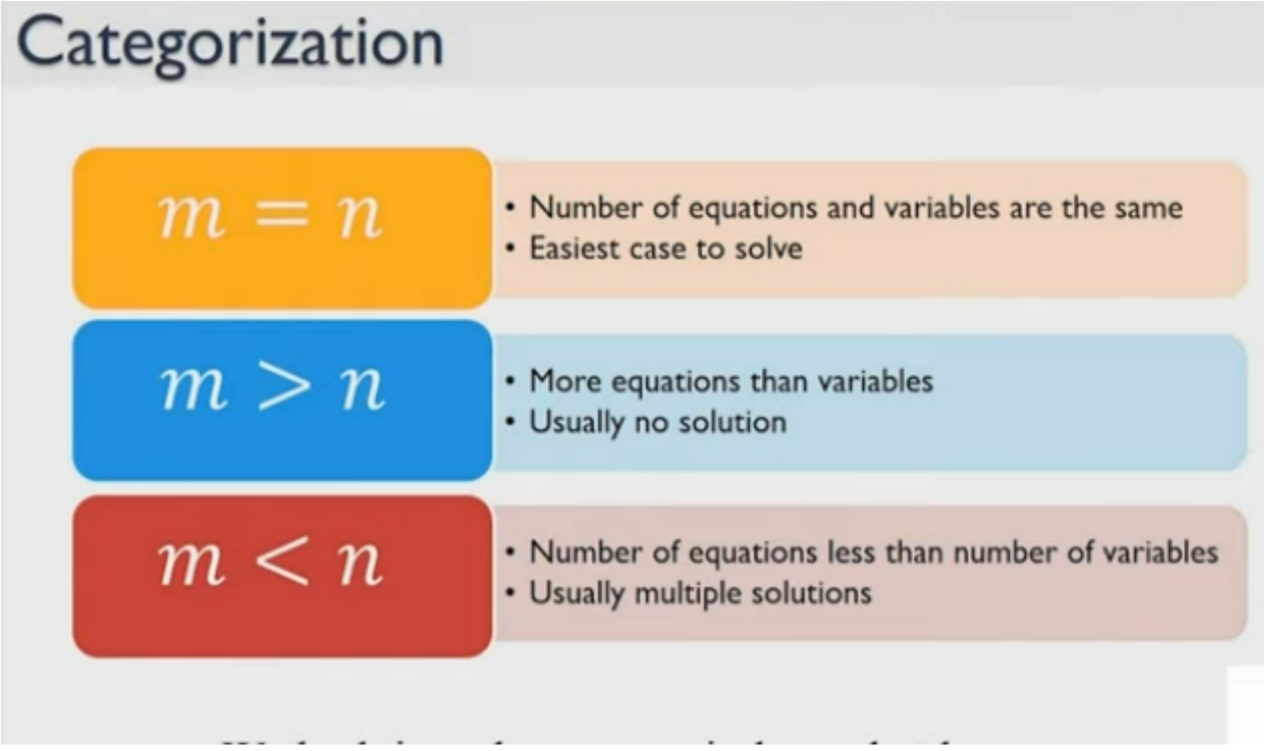
Recap..

Solving Matrix Equations

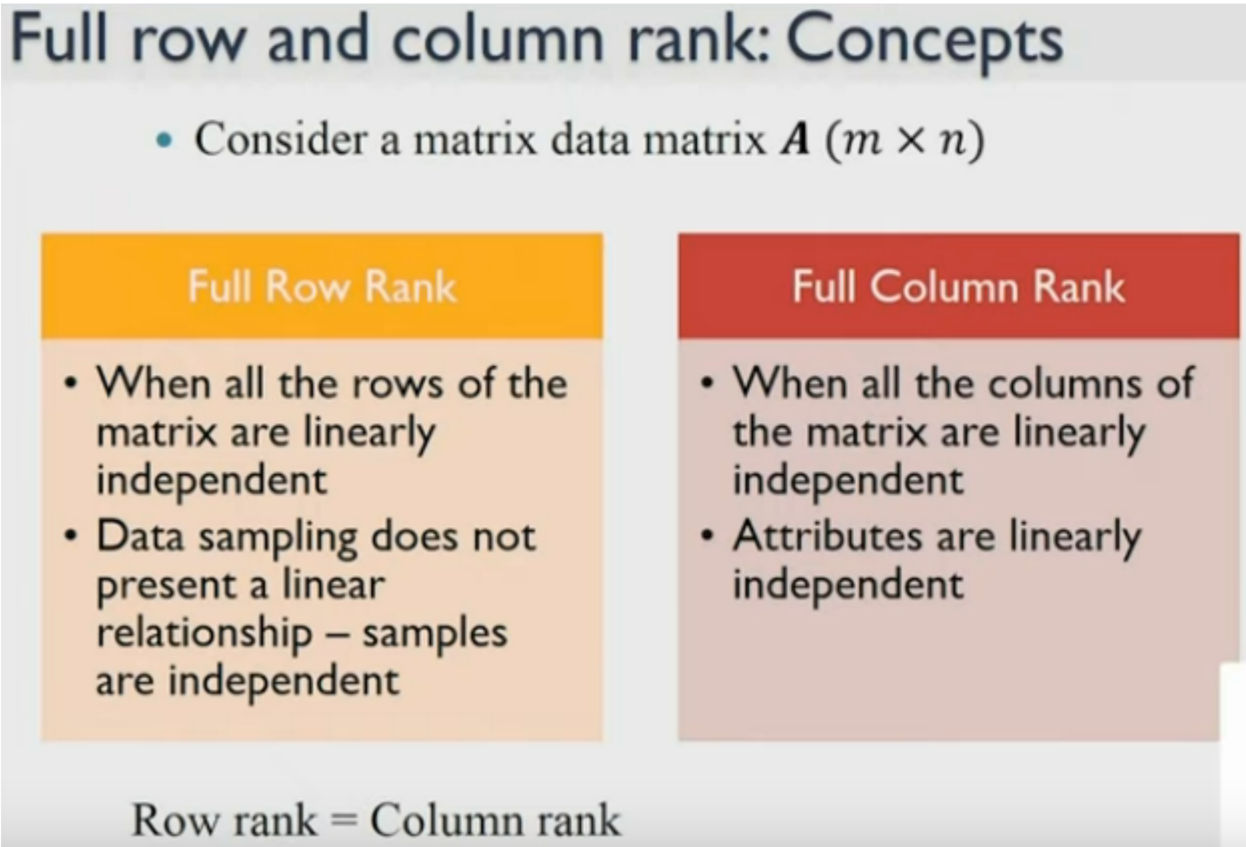
Recollect..!! from the M-I of Diploma Math, Smt. Gomati Madam has taught the subject..



Categorization..



A short note on Ranks.. *(as they are going to be used here..)*

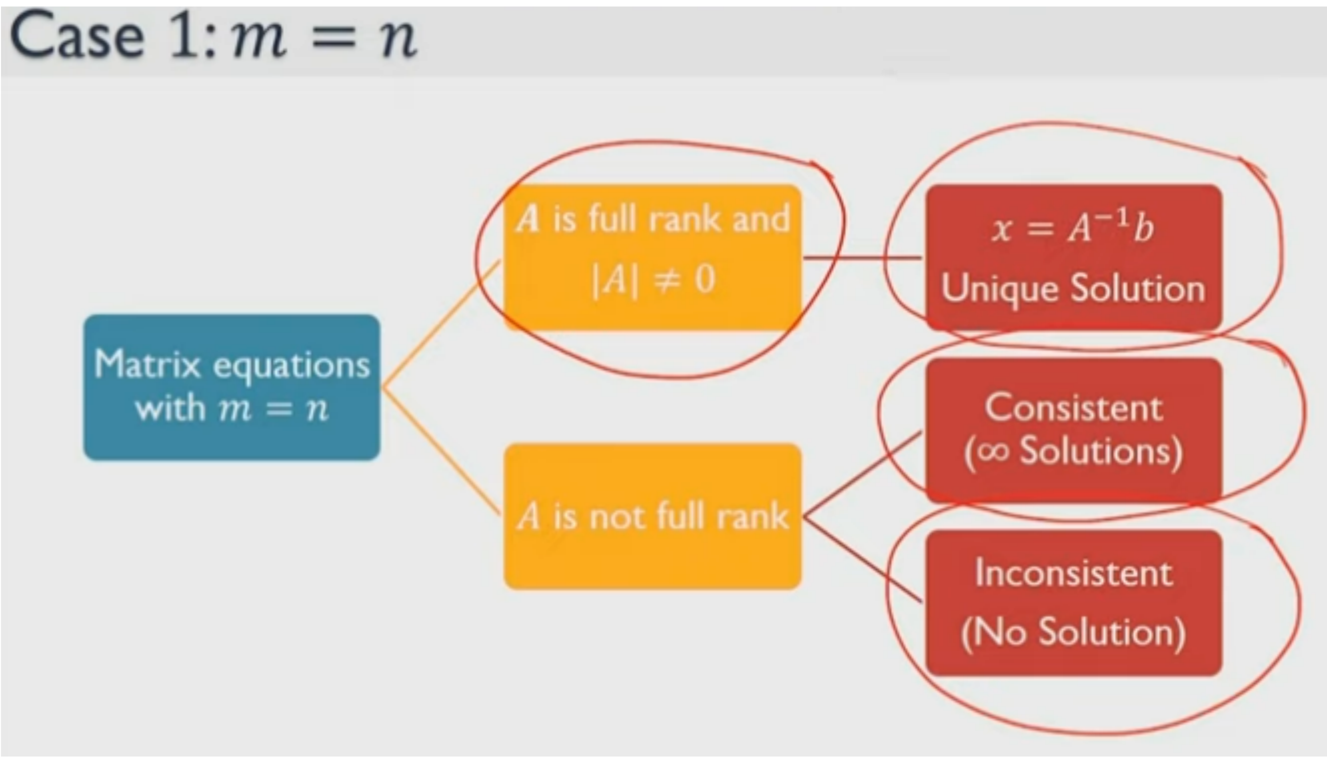


- Generally, if had a matrix A_{mxn}, then... maximum rank value can be atmost m if $m < n$, else n if $n < m$.-- i.e., the smallest value in dimensions of the matrix, is the maximum value, a rank can be.
- Generally, whatever may be the size of the matrix, **Row Rank = column Rank** -- *indicates that, if had certain no. of independent rows, then we have same no. of independent no, of cols.*

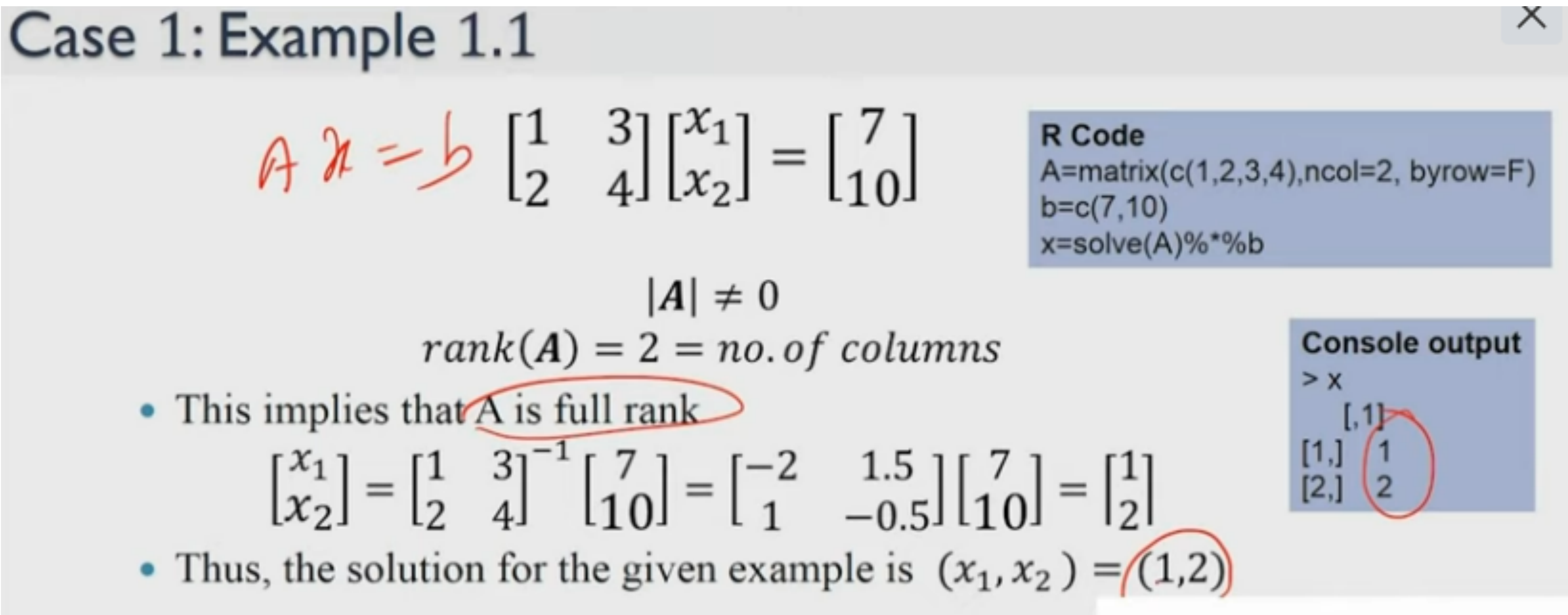
P2N: rank tells the no. of independent cols (or rows)

Case-1: `m==n`

- If A is full rank (Now the max rank can be either m or n, as m==n), then
  - We have all the rows or cols independent of each other *(in other words)*, we don't get any row or column such that [On LHS], its being linearly dependent on other. 00 then we have **Unique solution**, its $x = A^{-1}b$ -- How..?? as the equation is of the form $Ax = b$, and we need x, so by simply transposing A to R.H.S. ... we get that..
- The difficulty arises here.. think why the rank is less... on L.H.S. atleast one row or col, linearly depends on other. In this case, depending on what value we have on the R.H.S. we have 2 cases..
  - One is **Consistent** situation -- Will have many solutions..
  - Second is **Inconsistent** - no solution



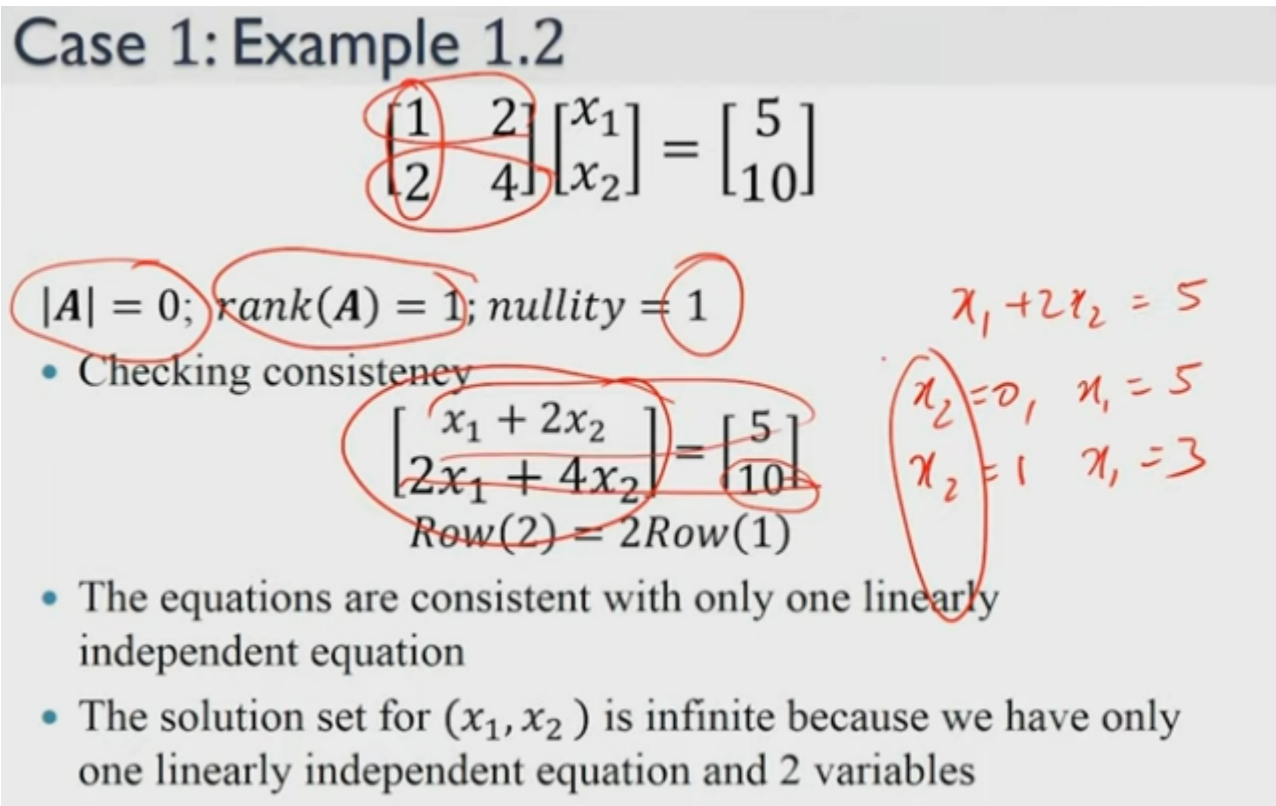
Example: With Full rank. -- demonstrating the *Unique solution*



--- P2N: No other solution, can satisfy the equation.-- as it is the unique solution.

Example: With < full rank... -- demonstarting the *Consistent solution*

- Notice that, they are linearly dependent -- both column-wise and row-wise. -- its even said by *rank* and *determinant being=0*.
- We generally consider the LHS.



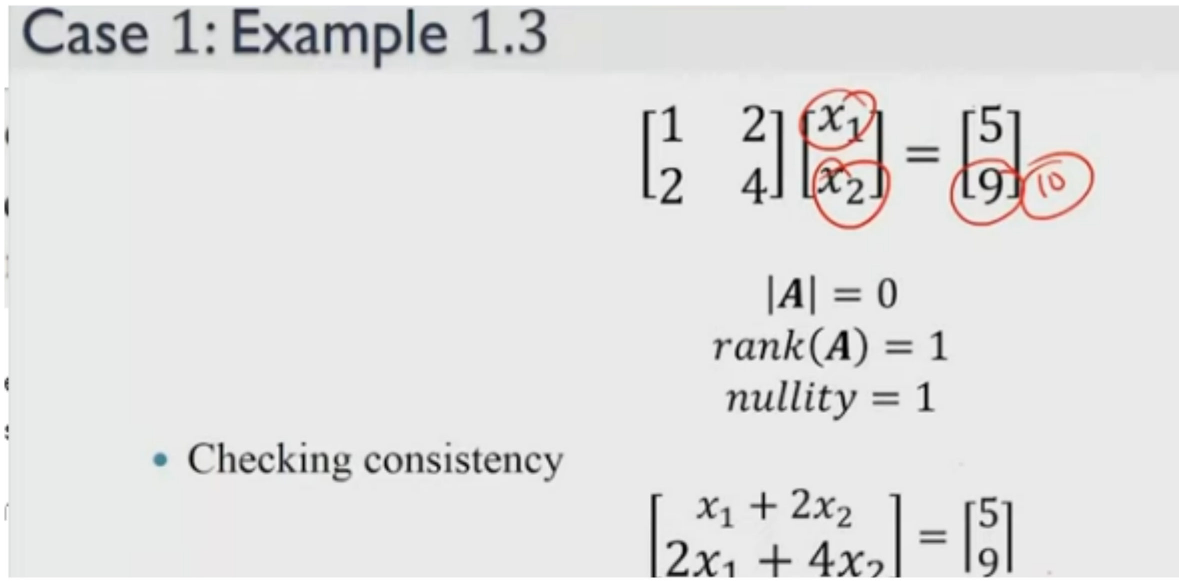
- Also notice that, even the R.H.S. is also linearly dependent. It means that, we can calculate the one and get the other. **NOTICE** that, we get the freedom to choose x_1 or x_2 (satisifying the left-over equation ofcourse..).

What is meant by having infinite no. of solutions?

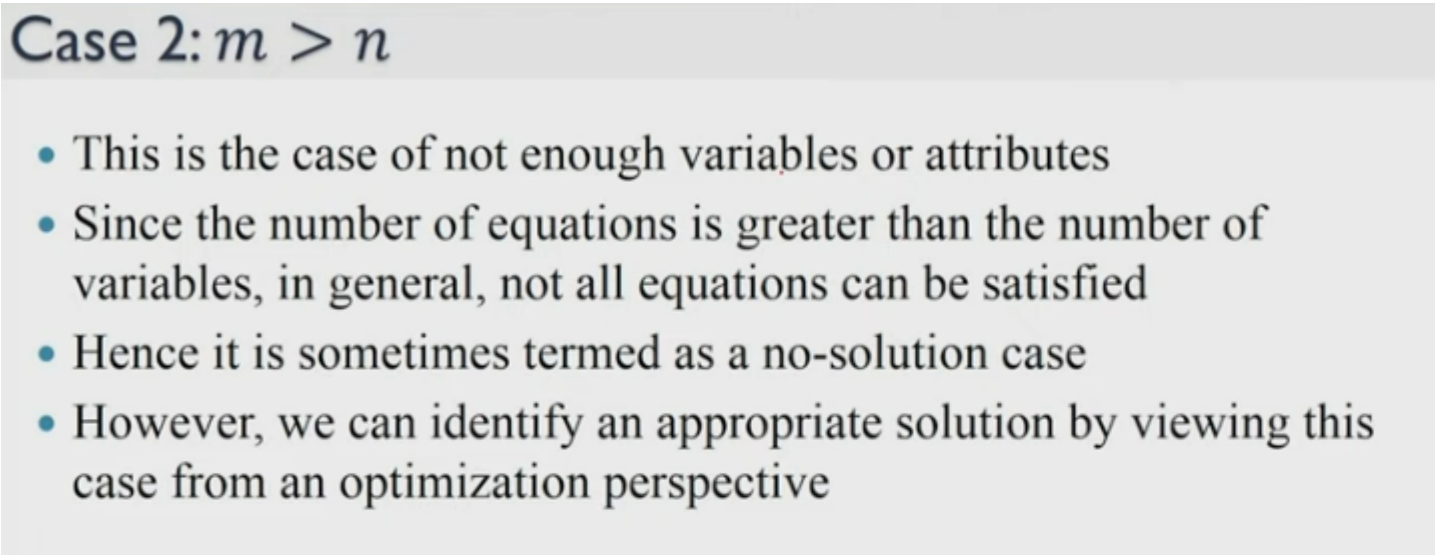
> See that, We have infinite choice of choosing the x_1 or x_2 (one keeping fixed) and all those infinite choices are valid for the equation. Hence, we get the infinite no. of solutions.

Example: with < full rank -- demonstrating the *Inconsistent solutions*.

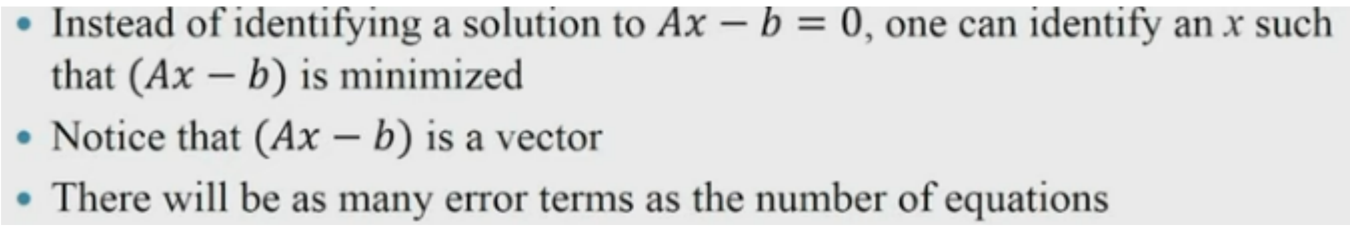
- Its same as above example, with a change that, RHS is not linearly dependent -- *Still the L.H.S. is linearly dependent*



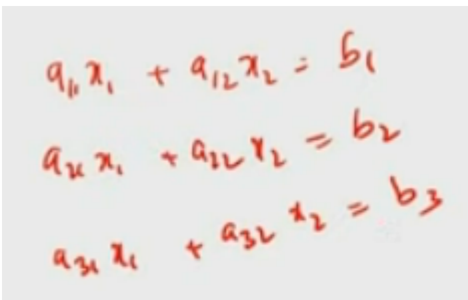
Case-2: $m > n$



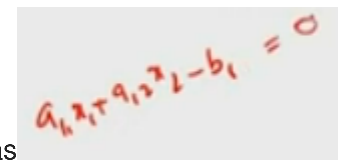
- Even though some times, its termed as no-solutions case -- from an optimization perspective can try..

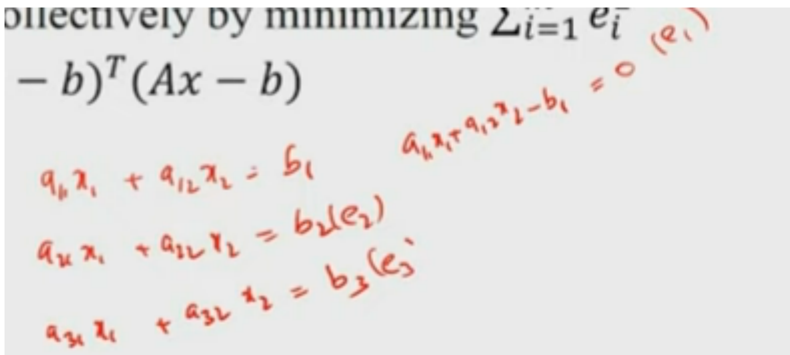


means..

- When had 

perfect solutions for all the three equations _(i., the values for which x_1, x_2, x_3 satisfies)_.

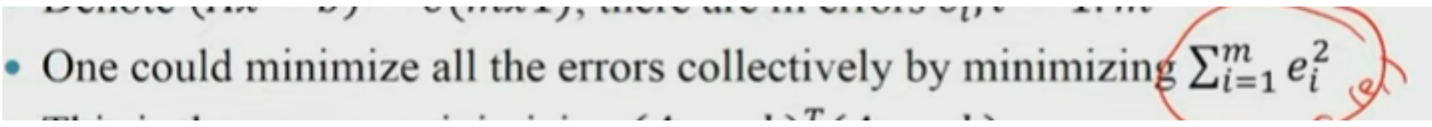
  Then can write as 

.
- But, whenever we don't have the perfect solution, let's call it error. And we allowed it, so we get 3 (or as many depending on equations) different errors right.. It's represented as 

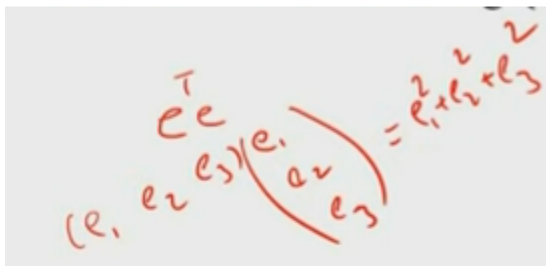
- in slide represented as

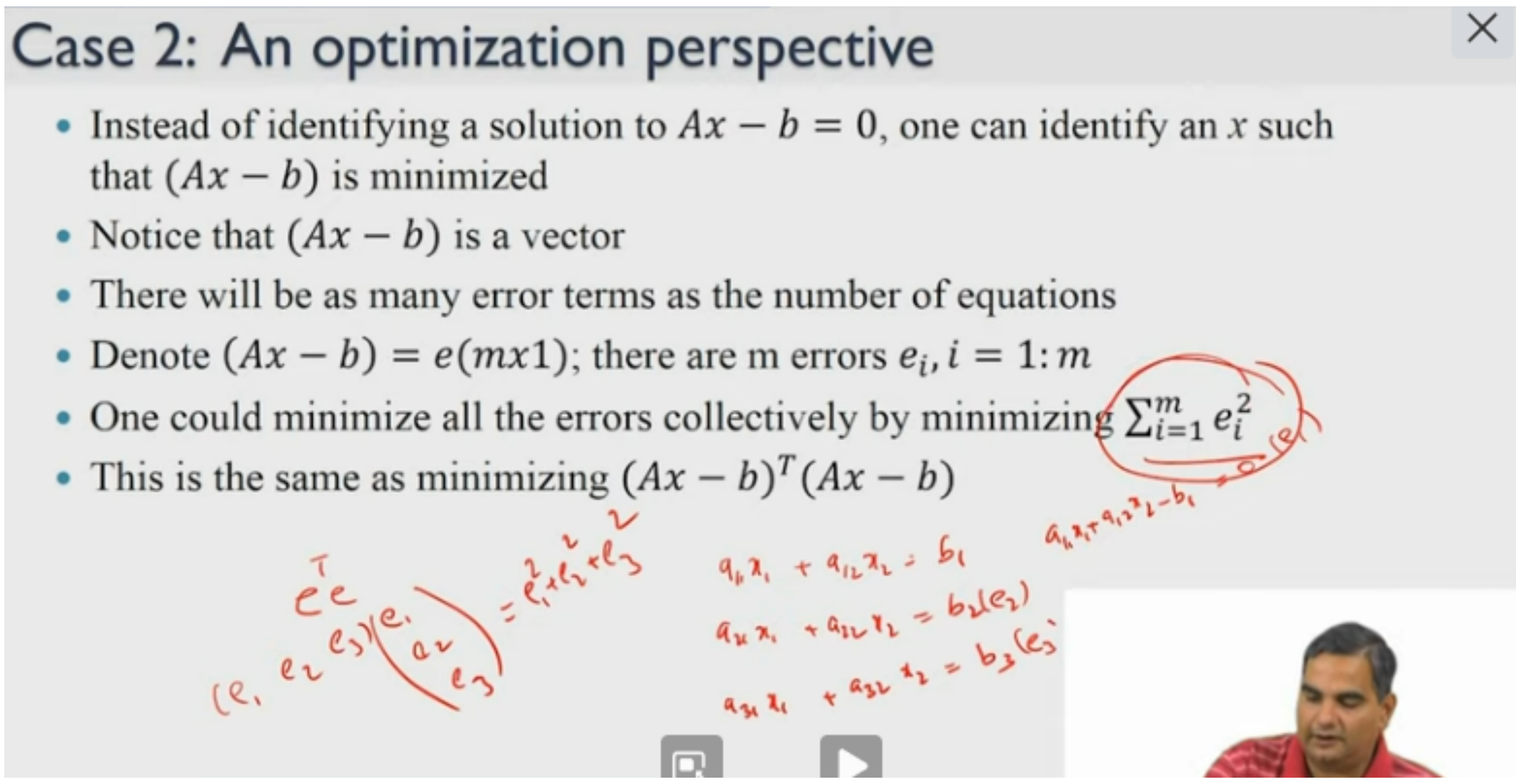
How do we minimize the errors collectively?

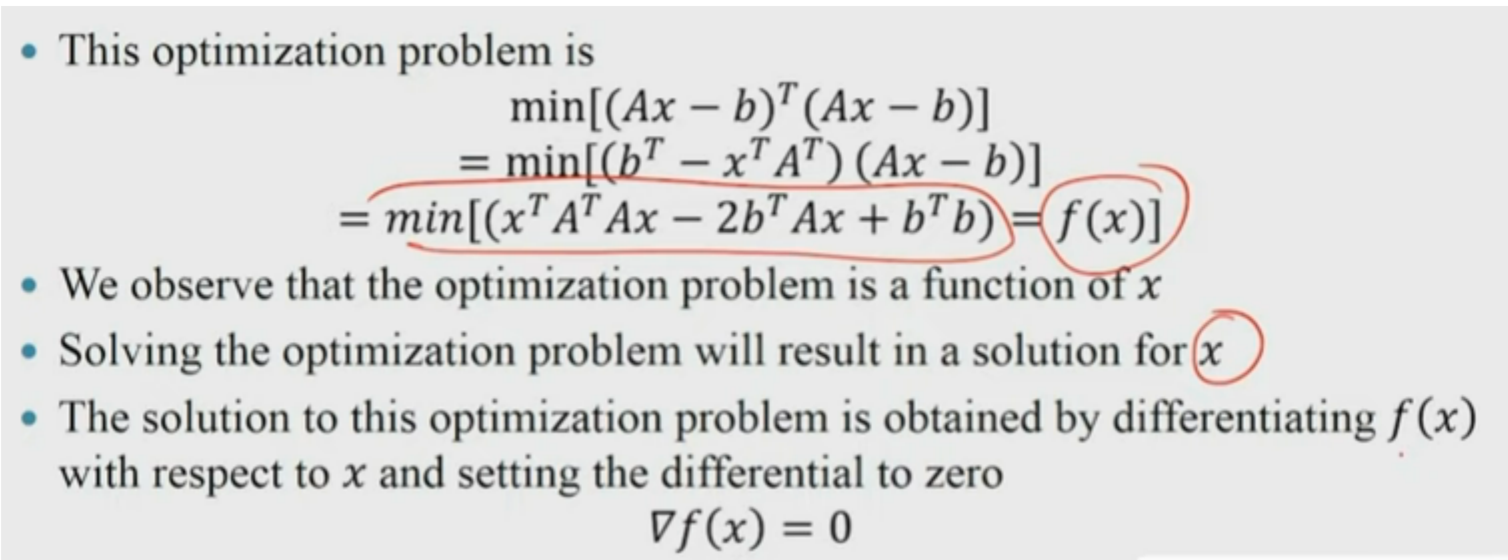
- One might say that **sum up all those**, but that won't be a good idea as, e_1 could be a vey big error in +ve direction, and e_2 could be a -ve direction which may sum up to 0, and e_3.. --- that's not a good solution at all.
- One way to do is, **Collectively minimize the error**



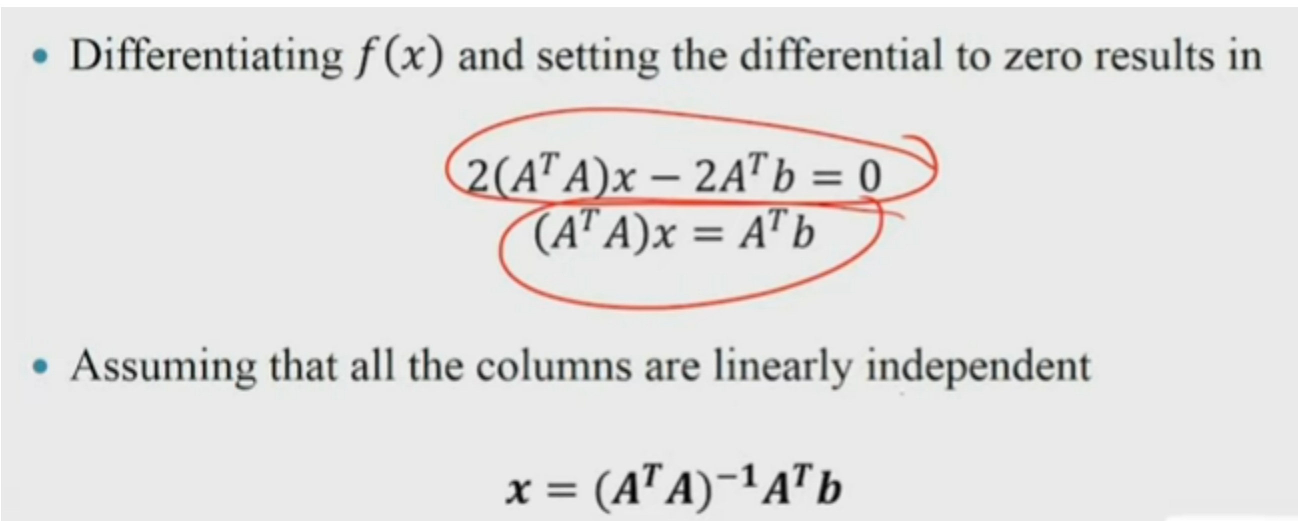
. i.e., Instead of going $e_1 + e_2 + e_3$, can go as $e_1^2 + e_2^2 + e_3^2$---- As long as these (e_1, e_2, e_3) are away from 0, the contribution to error is high. *(as squaring right..!!)* -- that ensures that, you do not go far away from 0. -- this is the **Least Square Solution**.

- Can also try as $|e_1| + |e_2| + |e_3|$ *(as $|$ always gives +ve)* -- that's also possible, but in general we consider this method.
- Notice that, one can write this($e_1^2 + e_2^2 + e_3^2$) also as $e^T e$ which is equivalent to 

To put it together.. 



-- The top-centered final equation is obtained *after some algebraic manipulation* -- that's what sir, said about it. You please work out on it.



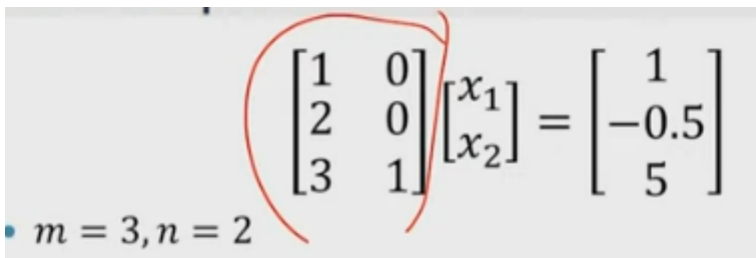
-- the final solution x, might not satisfy for all the values of equations, it ensures that, error is collectively minimized.

Concludes the lecture, in the next lecture..

- An ex. to illustrate how these are satisfied or not satisfied. Next case-3 where m>n.
- Next, how can these can be combined into one elegant solution.

Module-3

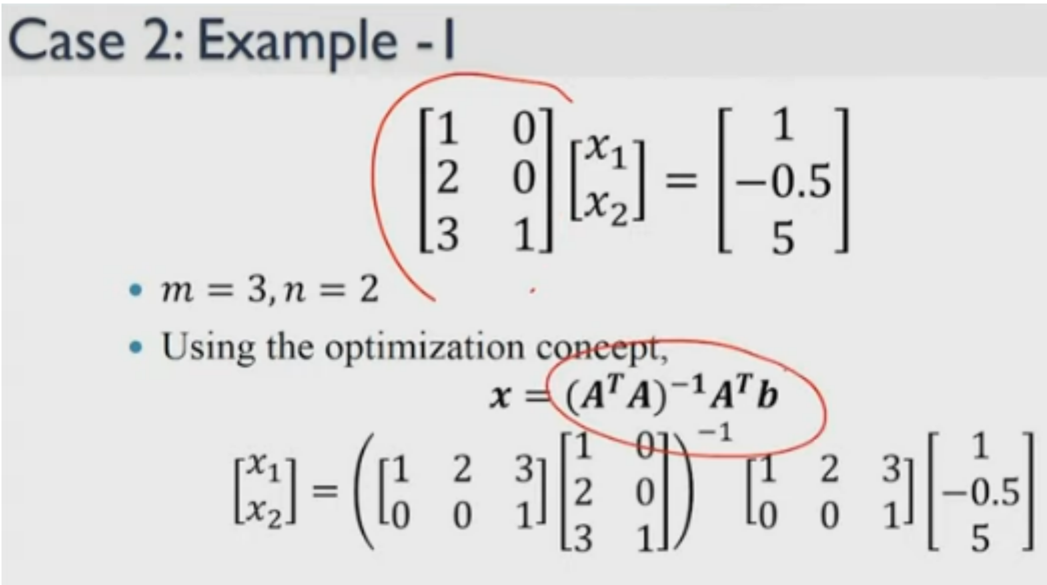
--- on 2nd September, 2021 ~ Thursday (Ekadashi)

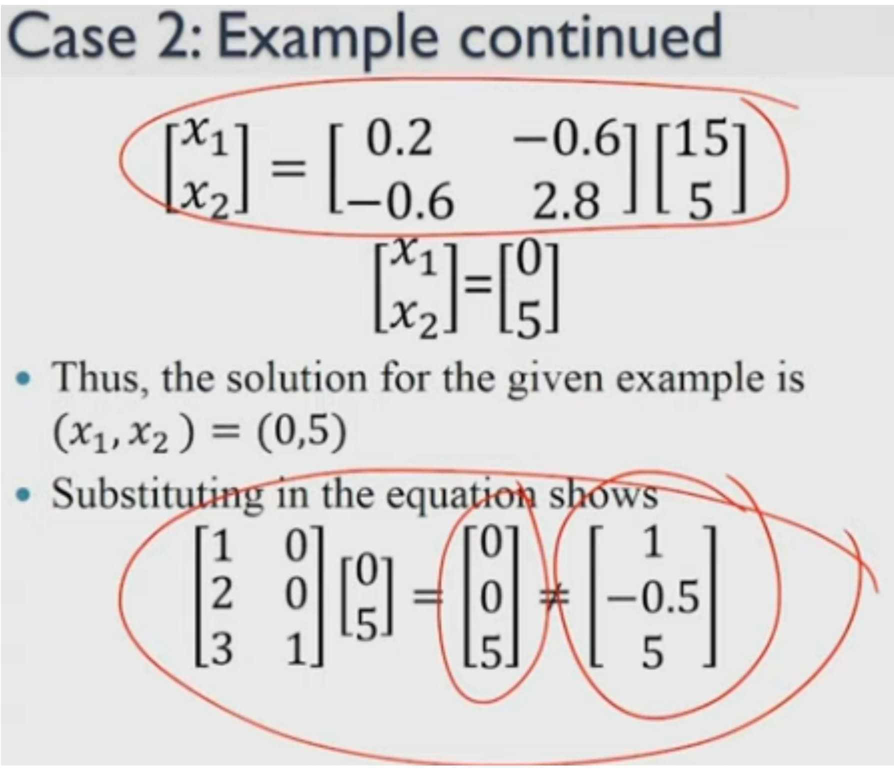


- You can observe that the first 2 equations are inconsistent. i.e., $1x_1 = 1$.... `#1` and $2x_2 = -0.5$...... `#2` , think here.. by what value can you satisfy these two combinely? -- that isn't possible, hence its incosistent.
- Coming to the $3x_1 + x_2 = 5$........ `#x3` , by fixing x_1, can get x_2.

On the whole, this is not solvable.

Then, how do we solve these..?? --- via **Optimization concept** : $x = (A^TA)^{-1}A^Tb$ .... as discussed in previous lecture *(Refer to the prev- lecture notes)*





Workout part...

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \\ 5 \end{bmatrix}$$

$3\times2 \quad 2\times2$

$$A \quad x \quad = b$$

Optimization concept $x = (A^T \cdot A)^{-1} A^T \cdot b$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \left(\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -0.5 \\ 5 \end{bmatrix}$$

$2\times3 \quad 3\times2 \qquad 2\times3 \quad 3\times1$

$$\begin{bmatrix} \overset{1}{1\times1} + \overset{4}{2\times2} + \overset{9}{3\times3} = 14 & 3 \\ 3 & 1 \end{bmatrix} \qquad \begin{bmatrix} \overset{1}{1\times1} + \overset{-1}{2\times(-0.5)} + \overset{15}{3\cdot5} \\ 1\times5 \end{bmatrix}$$

$$\begin{bmatrix} 14 & 3 \\ 3 & 1 \end{bmatrix} \qquad \begin{bmatrix} 15 \\ 5 \end{bmatrix}$$

$$A^{-1} = \frac{\text{adj}A}{|A|}$$

$|A| = 14\cdot1 - 9 = 5$

adjA for $14_{(1,1)} = 1\cdot(-1)^{1+1=2} +$

$3_{[1,2]} = 3\cdot(-1)^{1+2=3} -$

$3_{[2,1]} = 3\,(-1)^{2+1=3} -$

$1_{[2,2]} = 14(-1)^{2+2=4} +$

$$\begin{bmatrix} +1 & -3 \\ -3 & +14 \end{bmatrix}$$

$$\frac{\text{adj}A}{|A|} = \frac{\begin{bmatrix} 1 & -3 \\ -3 & 14 \end{bmatrix}}{5} \Rightarrow \begin{bmatrix} 1/5 & -3/5 \\ -3/5 & 14/5 \end{bmatrix} \Rightarrow$$

[x₁] ... −0.6] [15] [1 0][0] [1 0] [1]

Case 2: Example

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

- $m = 3, n = 2$
- Using the optimization concept,

$$\boldsymbol{x} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \left(\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

Case 2: Example continued

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.2 & -0.6 \\ -0.6 & 2.8 \end{bmatrix} \begin{bmatrix} 20 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

- Thus, the solution for the given example is $(x_1, x_2) = (1,2)$
- Substituting in the equation shows

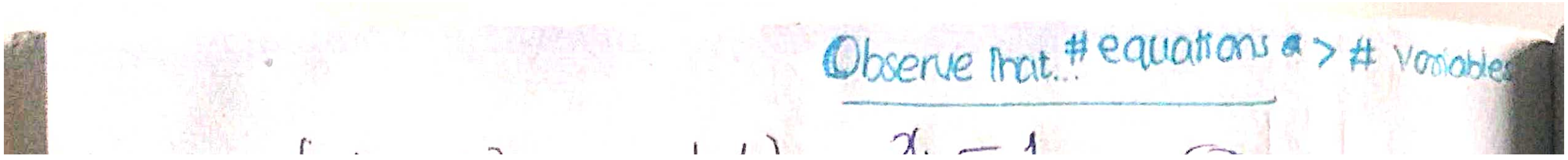
$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

R Code

```
A=matrix(c(1,2,3,0,0,1),ncol=2, byrow=F)
b=matrix(c(1,2,5),ncol=2, byrow=F)
x=inv(t(A)%*%A)%*%t(A)%*%b
x
```

Console output

```
> x=inv(t(A)%*%A)%*%t(A)%*%b
> x
     [,1]
[1,]    1
[2,]    2
>
```



In [5]:

```
# Equations..
A = matrix(c(1, 2, 3, 0, 0, 1), ncol=2, byrow=F)
b = matrix(c(1, 2, 5), ncol=1, byrow=F)
A
b
```

1	0
2	0
3	1

1
2
5

In [10]:

```
install.packages("pracma")
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' ... done
```

In [11]:

```
# Solving the equations, via least square method..
x = inv( t(A) %*% A ) %*% t(A) %*% b
x
```

```
Error in inv(t(A) %*% A): could not find function "inv"
Traceback:
```

`case-3` : Where no. equations(m) < no. of variables(n)

---- i.e., rows(m) < cols(n)

- This case addresses the problem of more attributes or variables than equations
- Since the number of attributes is greater than the number of equations, one can obtain multiple solutions for the attributes
- This is termed as an infinite-solution case

How to think this or understand this..??

> Consider 2 equations *(m=2)* with 3 variables *(n=3)*, this can be thought as...
>
> ```
> * **You could choose any value for $x_3$, plug in the 2 equations,, take them to R.H.S.*(as now
> they became constants)* and solve for rest $x_1 $ and $x_2$.**
> * Here **one can choose infinite range of values for $x_3$**, Hence *Inifinite solutions*..
> ---------------------making a clear sense right.. think for a while..
> ```

Now...if had infinite solutions....

- How does one choose a single solution from the set of infinite possible solutions?

> There is no way to distinguish between these infinite possible solutions, we need to have a metric -- with which we can pickup a solution.

Case 3: $m < n$

- This case addresses the problem of more attributes or variables than equations
- Since the number of attributes is greater than the number of equations, one can obtain multiple solutions for the attributes
- This is termed as an infinite-solution case
- How does one choose a single solution from the set of infinite possible solutions?

An optimization perspective.. *(like the previous case...)*

- Pose the following optimization problem

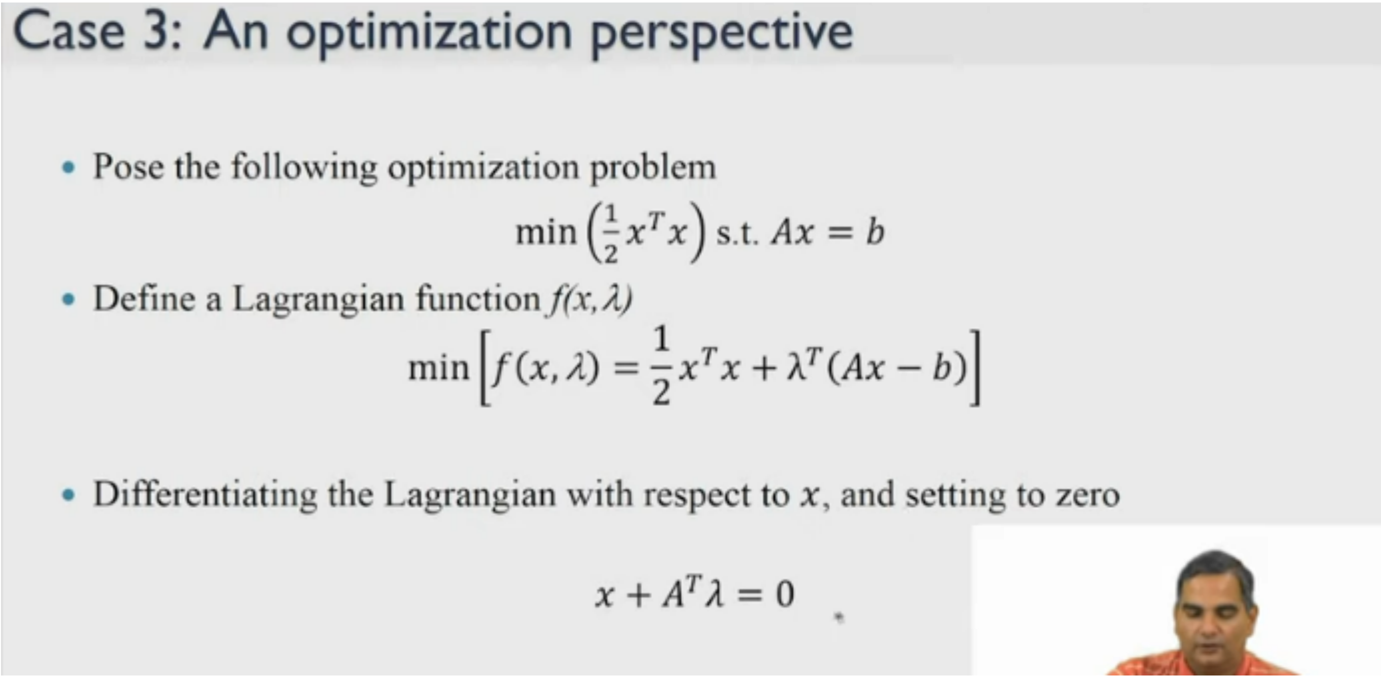
$$\min\left(\frac{1}{2}x^T x\right) \text{ s.t. } Ax = b$$

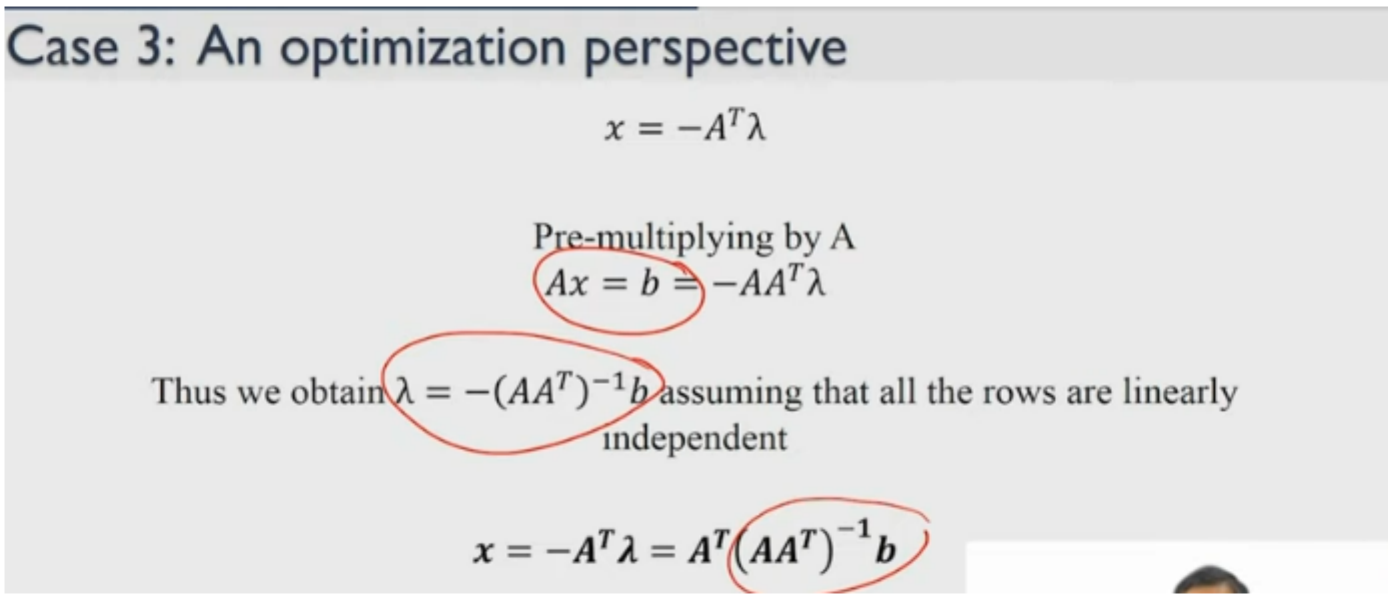
- Notice the constraint: $Ax = b$ ***(s.t.:** subject-to)*. On an whole tells that: **Need to minimize $\frac{1}{2}x^T x$ subject to the constraint $Ax = b$.**
  - In other words.... Whatever solution we get for x, has to satisfy the $Ax = b$. This is not the problem, as we have infinite no. of solutions. and what the $min\left(\frac{1}{2}x^T x\right)$ doing is... **How do I pick that one solution which minimizes the $x^T x$**

    > $1/2$ is to make sure that, solution comes out in a nice form.

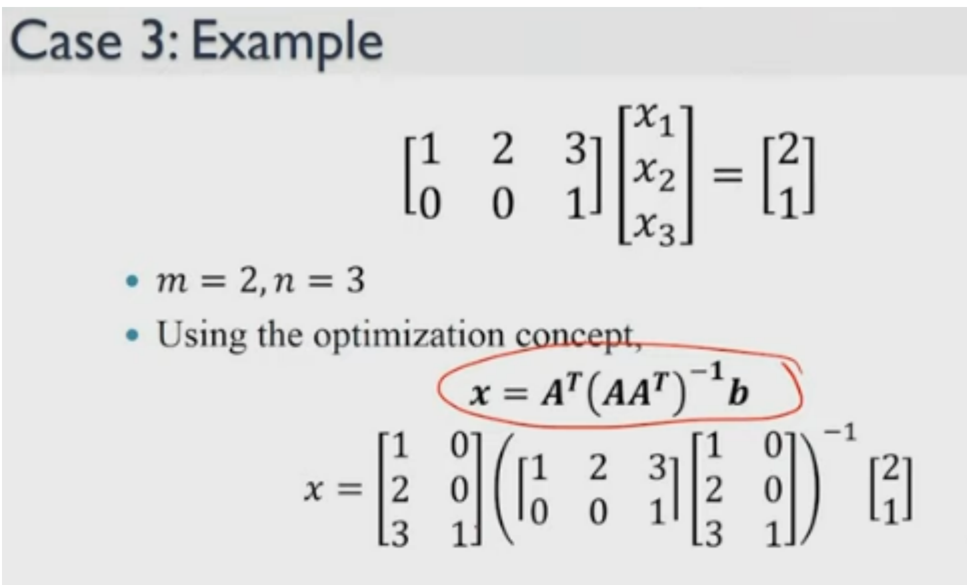
Engineering Perspective:, Say you have many parameters for manufacturing, and you need only of the smallest cost/size...

Unlike previous one, where there is no constraint, so its called :**Unconstrained Optimization**, here as there exists a constraint, its **Constrained Optimization**.





Example..



Case 3: Example

$$x = \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 14 & 3 \\ 3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$x = \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} -0.2 \\ 1.6 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -0.2 \\ -0.4 \\ 1 \end{bmatrix}$$

- The solution for the given example is $(x_1, x_2, x_3) = (-0.2, -0.4, 1)$

R Code

```
A=matrix(c(1,0,2,0,3,1),ncol=3)
b=c(2,1)
library(MASS)
x = t(A)%*%inv(A%*%t(A)) %*%b
x
```

Console output

```
A=matrix(c(1,0,2,0,3,1),ncol=3, byrow=F)
b=c(2,1)
x = t(A)%*%inv(A%*%t(A)) %*%b
x
```

Check...

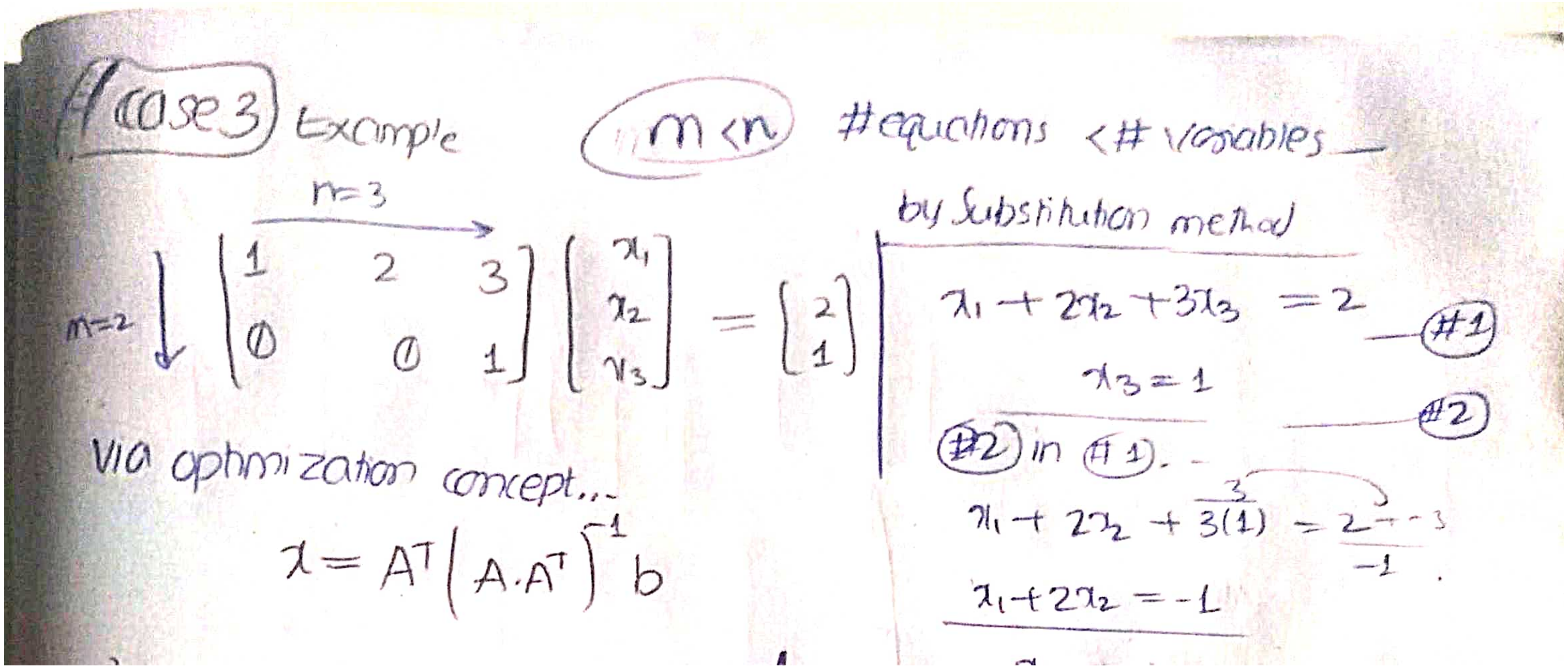
Case 3: Example

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

- The solution for the given example is $(x_1, x_2, x_3) = (-0.2, -0.4, 1)$
- Verify this is a solution that satisfies the original equation
- This also turns out to be minimum norm solution

-- Minimum Norm Solution

Workout...

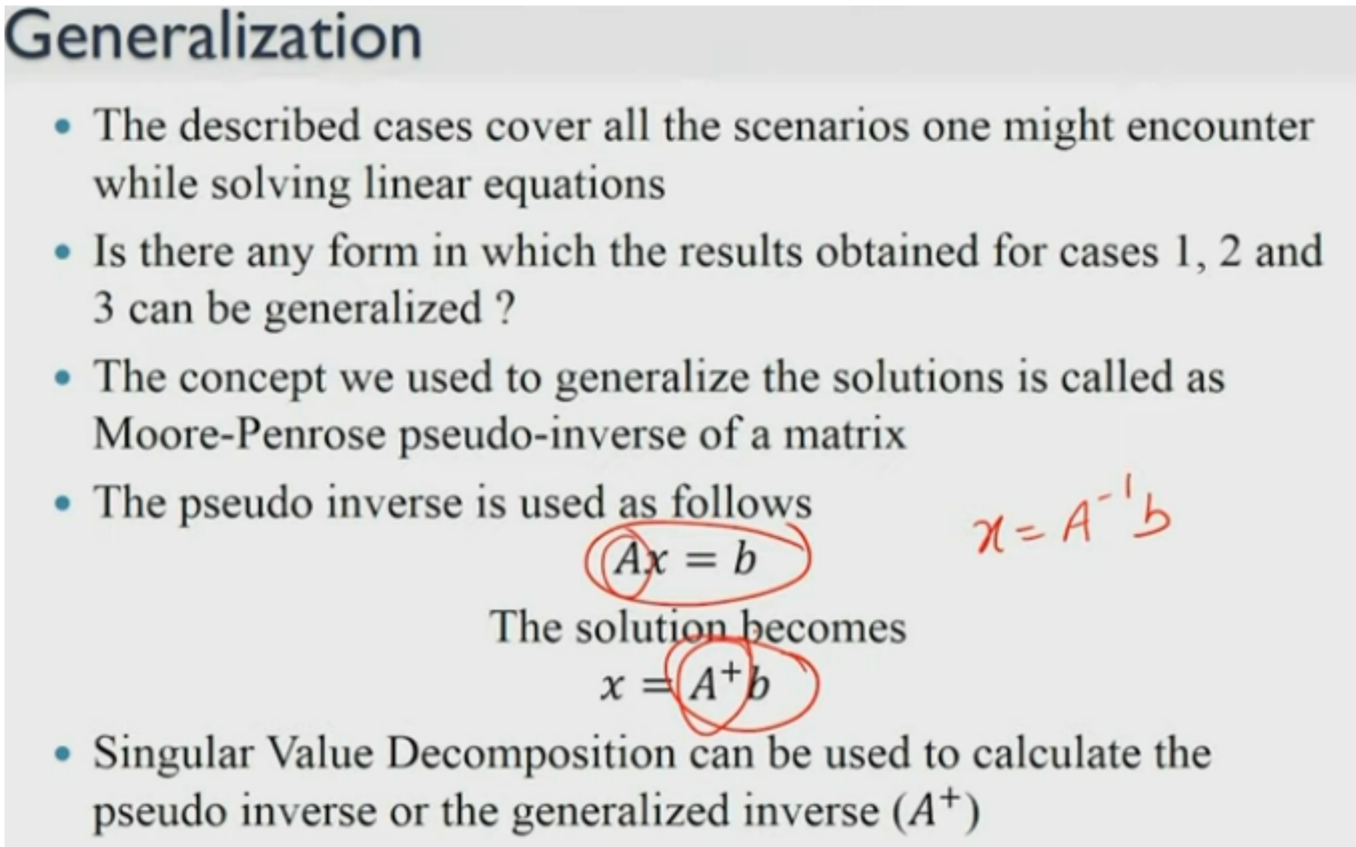


In [15]:

```
A = matrix(c(01, 0, 2, 0, 3, 1), ncol=3, byrow=F)
b = c(2, 1)
x = t(A)  %*% inv(A  %*% t(A))  %*% b
```

```
Error in inv(A %*% t(A)): could not find function "inv"
Traceback:
```

Till now, we came across, different methods for different case right.., **Is there any generalized method, with which one can solve all the cases** *(whether square [m==n] or rectangle [m!=n: $m < n$ or $m > n$])* --- Yes.. its **Moore-Penrose pseudo-inverse matrix**



-- here $x = A^{-1}b$ is the solution which we were used to do in previous cases, now this way...

- There are many ways to compute pseudo-inverse, SVD-Singular Value Decomposition is one of the method.

In [20]:

```
A = matrix( c(1, 2, 3, 0, 0, 1), ncol=2, byrow=F)
b = matrix(c(1, 2, 5), ncol=1, byrow=F)
A
b
```

1	0
2	0
3	1

1
2
5

In [21]:

```
# Calculate the pseudo-inverse of the matrix..  which is of more variables than equations ... m<n case..
library(MASS)
x = ginv(A) %*% b   # Here 'g' in "ginv" stands for """Generalization"""
x
```

1

In [27]:
```
A = matrix( c(1, 0, 2, 0, 3, 1), ncol=3, byrow=F)
b = c(2, 1)
A
b
```

1	2	3
0	0	1

1. 2
2. 1

In [28]:
```
# Calculate the pseudo-inverse of the matrix..  which is of more equations than variables ...m>n case..
library(MASS)
x = ginv(A) %*% b
x
```

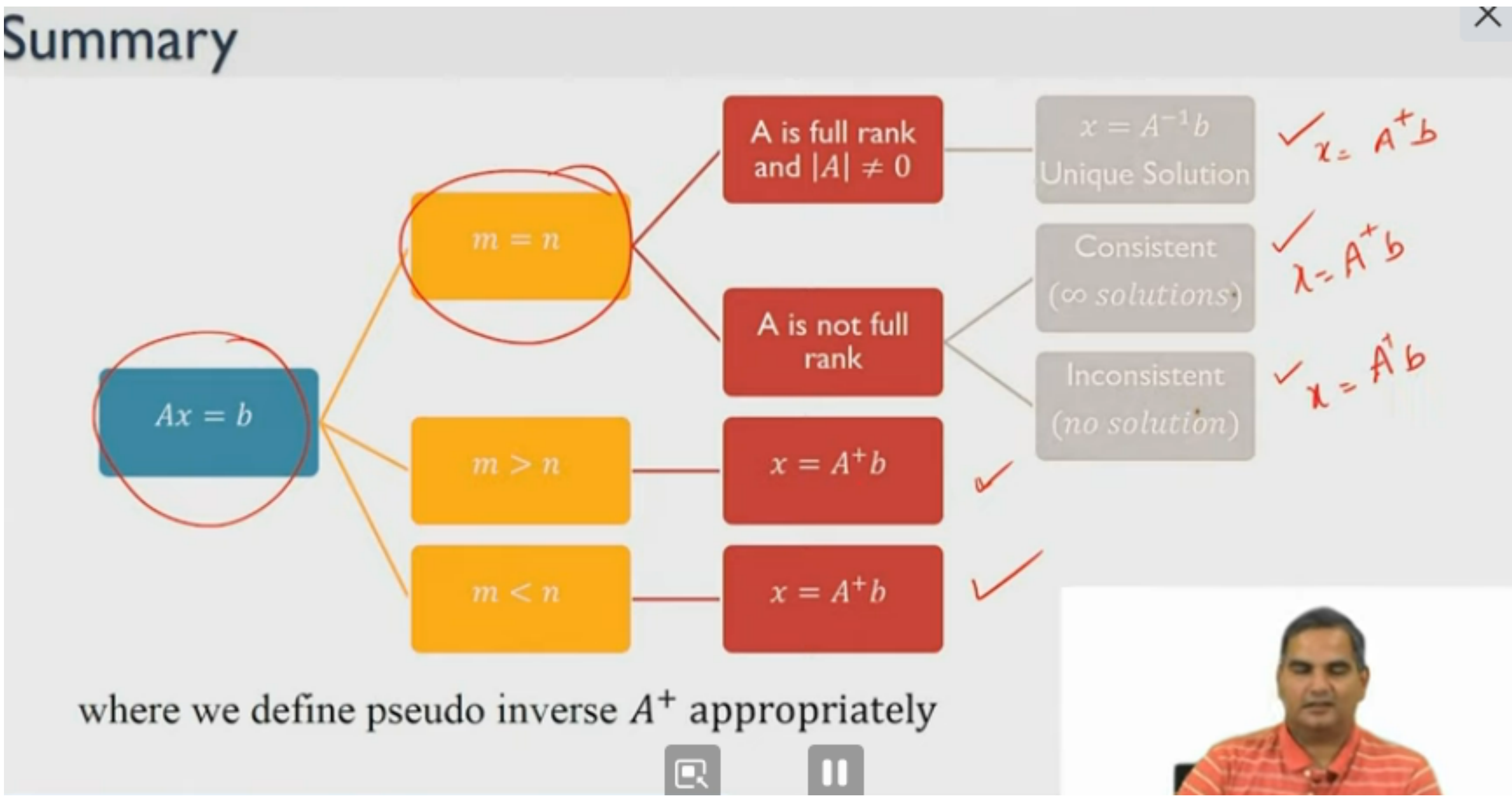
-0.2
-0.4
1.0

The values of x can be calculated via these... **How to interpret those....??**

That's what done in the lecture..

- For the case of $m > n$, its the solution: **Least Square Solution***(which minimizes the error collectively)* i.e., minimizes $e_1^2 + e_2^2 + e_3^2 \ldots \ldots$
- For the case of $m < n$, --- Minimum norm solution, this is the solution that is closest to the originm when had infinite no. of solutions.

Summary



So, without worrying about the square or rectangular system and not worrying about, whether the columns are dependent or independent, we can use generalized solution as one unify concept to find the solution to all the cases.

Coming up next...

- Geometrical View of these..