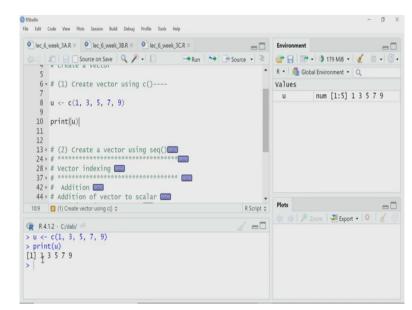
Data Analysis for Biologists Professor Biplab Bose Department of Biosciences and Bioengineering Mehta Family School of Data Science and Artificial Intelligence Indian Institute of Technology Guwahati Lecture 19 Linear algebra using R

Hello everyone, welcome back. In week 2, we have learned about linear algebra. And in this lecture, I will imply implement those concepts that we have learned in linear algebra course using R. So, in this lecture, we will start with vectors, we will do vector operations, then we will move into matrices and do matrix operations like multiplication of matrices, calculating the determinant of a matrix something like that.

And eventually, we will solve a system of linear equations using matrix algebra, linear algebra. So, let us start with vectors, I will create vectors and then perform the vector additions like vectors multiplication, and all other things that we have learned in linear algebra on vectors.

(Refer Slide Time: 01:23)



$u \leftarrow c(1, 3, 5, 7)$

print(u)

I will show you creation of vector using a two function one is c and other one is a sequence, both of these you have made earlier in the previous lectures. So, let us start with c function. I want to combine some data to create a vector. So, I have five data values 1, 3, 5, 7 and 9, I

want to combine them together using the c function so that I will create a new variable u as written here in the code and that u will be vector, 1-dimensional list.

So, if I execute u assign to c and those values then u is created in the environment you can see u has been created. Now, I can print u and you can see, it is saying u is 1, is a list of 1, 3, 5, 7, 9. So, we can call it as a vector, it is a 1-dimensional list so I can call it a vector. Now, apart from c as we have learned earlier, I can create a vector using the sequence function also.

(Refer Slide Time: 02:28)

O Kötudo File Edit Code Vew Plots Session Build Delsug Profile Tools Help		- 0 ×
lec_6_week_3AR ×	Environment	-0
🔅 🕼 🔚 🗋 Source on Save 🔍 🎢 🔋 📄 😁 Run 😁 🕞 Source 🔹	💣 🔒 🔛	• 🌗 179 MiB • 🖌 📄 • 🕑 •
8 u <- c(1, 3, 5, 7, 9)	R 🔹 🛑 Glo	bal Environment • Q
9	Values	
10 print(u) 11	u	num [1:5] 1 3 5 7 9
12	v	num [1:5] 12 20 28 3
13 * # (2) Create a vector using seq()		
14		
15 v <- seq(12, 46, 8) 16		
17 print(v)		
18		
19		
20 21		
77	Plots	
17:9 🚺 (2) Create a vector using seq() Create a vector using seq() R Script C		Zoom Zoom Export - 0
🙊 R 4.1.2 · C/dab/ 🔗 🚽 🚽 🖉		
> U <- C(1, 3, 3, 7, 9)		
> print(u) [1] 1 3 5 7 9		
> v <- seq(12, 46, 8)		
> print(v)		
[1] 12 20 28 36 44		
🟥 🔎 Type here to search 🛛 🛛 🖬 💿 🕕 🦻 📵 🍃 🔘	2 0	25°C ^ 10 41 4 BHG 18-11-2021

$v \leftarrow seq(12, 46, 8)$

print(v)

So, let us do that. So, what I am doing here in this, I want to create a vector that will start with 12. And it will go up to 46 with an increment of 8, that is what I do in sequence function, we give the first value the lowest value or the first value and then we give the upper limit and we give the increment, so that is what I have given here, 12 is the first value, 46 is the last value, 8 is the increment.

So, let me execute that and that will give me a v vector because I assign that to v. So, if I print v, I can see it is starting with 12, then we are adding 8, so it is becoming 20 and it is going up to 44 because if I add 44 plus 8, it will be 52. So, I cannot go beyond 46. So, it stops at 44. So, I have got two vectors. So, I have created these two vectors, you can read a data file and extract vectors also, we have done that earlier.

Now, once I have got the vector, one important thing that we have to understand is indexing of it. What is a vector? If you remember we have said that vectors are stacked numbers. So, now, if I have stacked numbers, 10 numbers, 3 numbers or 4 numbers, I should be able to label them. So, in R level start from 1 to or 2, 3, 4 something like that. So, I may ask that okay, I want a third number in my vector. So, that is a third index. So, I should be able to get a particular value of having a particular index number to that. So, let us try to do that.

(Refer Slide Time: 04:11)

Icc_6_week_3AR × I lcc_6_week_3B.R × I lcc_6_week_3C.R ×	6	invironment	-0
🗘 🕼 📄 Source on Save 🔍 🎢 🛯 🔂 🖶 Run 🗌			79 MiB 🔹 🖌 🥑 🖬
5 6, # (1) Create vector using c()	* R	🔹 💼 Global Enviro	onment • Q
13, # (2) Create a vector using seq()	V	alues	
24 • # **********************************		u num	[1:5] 1 3 5 7 9
28 + # Vector indexing 29		v num	[1:5] 12 20 28 3
<pre>30 # vactor_name[e]ement number] 31 32 # Get the third e]ement of u: 33 34 print(u(3)) 35 34:12 @ Vector indexing :</pre>	R Script ©		
(ℝ R.4.1.2 · C:/dab/	1 -0 -		
<pre>> u <- c(1, 3, 5, 7, 9) > print(u) (1] 1 3 5 7 9 > v <- seq(12, 46, 8) > print(v) (1] 12 20 28 36 44 > print(u[3]) (1] 5 T</pre>	P	Nots	

print(u[3])

So, how would you write it? We write the vector name, and then we put the index number or element number. For example, as I have shown in the script, if I have created a u vector, few seconds back if you look here in the console, u is 1, 3, 5, 7, 9. These five numbers stacked. I want to get the third number, what will be the third number, third number will be 5. The first number is 1, second number is 3, the third number is 5.

So, I want to get the third number. So, how I should write? I should write it u and in the square bracket I should write that index 3. So, u in square bracket 3 means it tells R that okay, I want the third element of this u vector, and then I am putting it under the print so that I can print it. Let us print that. So, it says 5, obviously, because the 5 is the third element or the third number in my vector u.

So, this is how you actually use indexing of vector, you may want a part of that vector and do some further calculations. Now, in the lecture on vector and vector operations in the last

week, we have learned about addition of vectors and multiplication of vector. So, let us start with addition. So, addition can be of two type, I may add two vectors, I may add a scalar to a vector also, the simplest one is always the addition of a scalar to a vector.

(Refer Slide Time: 05:43)

lec_6_week_3A.R × P lec_6_week_3B.R × P lec_6_week_3C.R ×	Enviro	nment	-
💷 🖉 📄 Source on Save 🔍 🎢 🛛 🔂 🔿 Run 😁	🕈 Source 🔹 🖉 🚰	9 1 1 1 1 1 1 1 1	179 MiB 🔹 🥑 🔳 🔹
44 - # Addition of vector to scalar	* R *	🜗 Global Envir	onment • Q
45 46 # Add 2 to vector u	Valu	es	
47 p <- u + 2	р	num	[1:5] 3 5 7 9 11
48 I	u	num	[1:5] 1 3 5 7 9
49 print(p) 50	v	num	[1:5] 12 20 28 3
64 - # # ********************************	R Script \$		
	/ -0		
R 4.1.2 · C:/dab/ 🗇			_
v <- seq(12, 46, 8)	+ Plots		
<pre>v <- seq(12, 46, 8) print(v)</pre>	+ Plots		-2 Export - 0
v <- seq(12, 46, 8) print(v) 1] 12 20 28 36 44	+ Plots		
<pre>v <- seq(12, 46, 8) print(v) 1] 12 20 28 36 44 print(u[3])</pre>	+ Plots		
v <- seq(12, 46, 8) print(v) 1] 12 20 28 36 44 print(u[3]) 1] 5	+ Plots		
<pre>R R412: c/dbb/ @ v <- seq(12, 46, 8) print(v) 1] 12 20 28 36 44 print(u[3]) 1] 5 p <- u + 2 print(p)</pre>	+ Plots		

e lec_b	week_3A.R × • lec_6_week_3B.R × • lec_6_week_3C.R ×		Environment					-
	🔊 🔒 🖸 Source on Save 🔍 🎢 🗧 📑 Run 😁 📑 So	urce 🔹 🔍	😭 🔒 📑	- 01	79 MiB •	1		•] @
	# Addition of vector to scalar		R 🔹 🛑 Glo	bal Enviro	onment •	Q		
45 46	and 2 so washing in		Values					
40	# Add 2 to vector u p <- u + 2	-	p	num	[1:5]	357	9	11
48			u		[1:5]			
49	print(p)		v		[1:5]			
50		٣						
47:1	Addition of vector to scalar c	R Script ‡						
R R4	1.2 · C:/dab/	/ =0						
	c(1, 3, 5, 7, 9)							
> prin								
	3 5 7 9 I seq(12, 46, 8)							
prin			Plots					
	20 28 36 44							-
	t(u[3])		(a a) p		- Expor	t • 6	,	L
[1] 5	u + 2							
> p <-	u + 2							
	t(n)							
> prin	5 7 9 11							

$p \leftarrow u + 2$

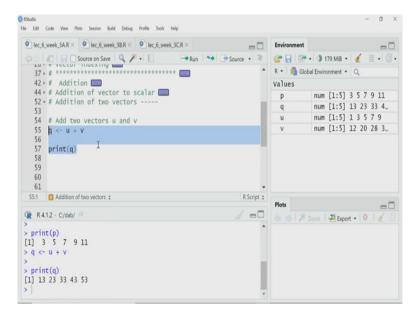
print(p)

So, suppose I want to add 2, 2 is a scalar value thing, to a vector u, to the vector u, u we have created few minutes back. So, how do I do a simple addition, I have to use the addition operator the plus sign here. So, u plus 2 will be summation of a vector u and the scalar 2, and it will give me a new vector, which I have assigned to p, and then I am printing this. So, let me execute both of them together.

So, what I have got, I have got a new vector 3, 5, 7, 9, 11, you can easily see my original vector u has 1, 3, 5, 7, 9. So, if I add 2 to them, the first element will become 3, the second

element will become 5, and so on. So, I have done scalar addition. Now, I want to do vector addition. I want to add two vectors. So, it should be similar, I only add the addition operator here.

(Refer Slide Time: 06:47)



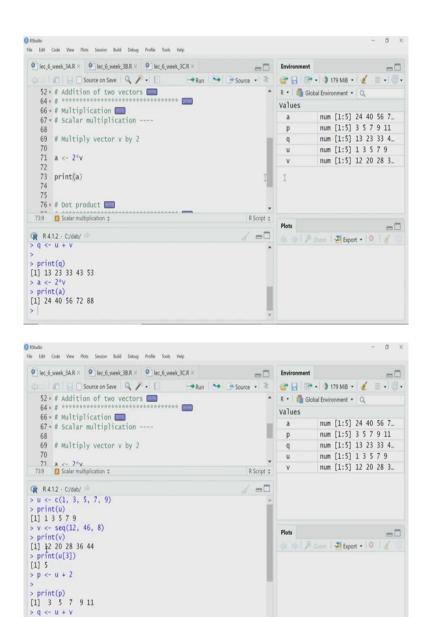
$q \leftarrow u + v$

print(q)

So, here what I am doing, I have created earlier two vectors, u and v and I want to join them, sum them together using the addition operator. And I will assign this whole thing to a new vector q. And then in the next line, I am asking to print the q, so that I can check what is that vector created by addition.

So, that is what we have got, we have got a new vector q with elements of the values 13, 23, 33, 43 and 53. So, we have learned addition, it is very simple to do. Now, we move into multiplication, again, just like addition, I can have a scalar multiplication, and I can have multiplication between vectors. So, start with the scalar one, the simplest one.

(Refer Slide Time: 07:34)



$a \leftarrow 2*v$

print(a)

So, for scalar multiplication, suppose I want to multiply the v vector with 2. So, what I will write, I will write 2, and then I will give the star, the star is the operator for multiplication if you remember, and then the vector I have written and I am assigning that because this multiplication will give me another vector. So, I am assigning that vector to a, new variable, and then in the next line, I am printing a so that I can check it.

So, let us execute this. So, I have done the multiplication, scalar multiplication and let me print it. My v if you see, let me expand the console v is 12, 20, 28 and so on. So, if I multiply by 2, I should be, it should become 24, 40 something like that and that is what has happened

here. You can see a is a vector with first element 24, second one is 40 and so on. So, I have performed scalar multiplication.

Now, I want to multiply two vectors. And when it comes to multiplication of two vectors, we talk, we are talking about here about the dot product. So, I want the dot product between two vectors. Now, you can do dot product by using the explicit definition how we have defined dot product and then you can use the base operator and base functions of R to do that.

(Refer Slide Time: 09:01)

	Postal Packages			
lec_6_week_3A.R × lec_6_week_3B.R	X Plet Check for Package Updates		Environment	=
🔅 🔅 🗐 🔚 🗌 Source on Save 🔍	Version Control	, e - 🖻	😭 🔒 🐨	🔹 🌗 179 MiB 🔹 🕖 🗮 🔹 🤇
64 • # *****************	Set Set		R 🔹 💼 Glo	bal Environment * Q
66 - # Multiplication 🔤	Terminal		Values	
67 + # Scalar multiplication			a	num [1:5] 24 40 56 7.
76 - # Dot product 77	Addins Memory		p	num [1:5] 3 5 7 9 11
78 # Use dot() of the 'ge	ometry			num [1:5] 13 23 33 4.
79 # Install geometry lib	Reyound Shirticus Help Ha	t+Shift+K	q	num [1:5] 1 3 5 7 9
80	Edit Code Snippets		v	num [1:5] 12 20 28 3.
81 library(geometry)		rl+Shift+P	v	TUM [1.5] 12 20 20 5.
82	Project Options.			
83 # Calculate dot product	torua	-		
84 b <- dot(u, v)	Global Options			
78:21 Dot product \$		R Script ‡		
😨 R 4.1.2 · C:/dab/ 🗇		/ =0	Plots	-
Proverse and and a second seco				
[1] 5		-		Zoom 📲 Export 🔹 3 🥖
> p <- u + 2				
> print(p)				
[1] 3 5 7 9 11				
> q <- u + v				
×				
> print(q)				
[1] 13 23 33 43 53				
	bur Profile Tools Help	•		- Ø
e Edit Code View Plots Session Build Deb		•	Environment	
e Edit Code View Plots Session Build Det Olec_6_week_3A.R × Olec_6_week_3B.R	× • lec_6_week_3C.R ×	-0	Environment	-
Edit Code View Plots Session Build Det Dec_6_week_3AR × Dec_6_week_3BR	× ● lec_6_week_3C.R ×	▼ Source - ≷	er 🔒 🖙	- 🌖 179 MiB • 🖌 🖉 🚍 • 🛙
Idt Code Vew Pors Sension Build Det Iec_6_week_3AR Iec_6_week_3BR	× • lec_6_week_3C.R ×		er 🔒 🖙	-
Idt Code Vew Pots Session Build Det Idc.6,week,3AR × Idc.6,week,3AR × Idc.6,week,3AR × Idc.6,week,3AR × <td>× ● lec_6_week_3C.R ×</td> <td></td> <td>er 🔒 🖙</td> <td>- 🌖 179 Miß • 🖌 🖉 📄 • 🗍</td>	× ● lec_6_week_3C.R ×		er 🔒 🖙	- 🌖 179 Miß • 🖌 🖉 📄 • 🗍
<pre>b 1dk Code Vew Pots Seston build Deb lec_6_week_3AR × lec_6_week_3BR b lec_6_week_3AR × lec_6_week_3BR b lec_6</pre>	× ● lec_6_week_3C.R ×		🔒 🖙 R • 🛯 🐴 Glo	- 🌖 179 Miß • 🖌 🖉 📄 • 🗍
Idt Code Vew Pots Session Build Det Idc.6,week,3AR × Idc.6,week,3AR × Idc.6,week,3AR × Idc.6,week,3AR × <td>× ● lec.6.week.3CR × ≫ • ■ ⊕ Run → ●</td> <td></td> <td>🔒 🖙 R • 🛯 🐴 Glo</td> <td>• 179 MiB • 2 = • (</td>	× ● lec.6.week.3CR × ≫ • ■ ⊕ Run → ●		🔒 🖙 R • 🛯 🐴 Glo	• 179 MiB • 2 = • (
Idt Code Vew Pots Senson Build Det Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor Image: Comparison of the sensor <td>× ● lec.6.week.3CR × ≫ • ■ ⊕ Run → ●</td> <td></td> <td>R • Glo Values</td> <td>• • • 179 MiB • • • • • • • • • • • • • • • • • • •</td>	× ● lec.6.week.3CR × ≫ • ■ ⊕ Run → ●		R • Glo Values	• • • 179 MiB • • • • • • • • • • • • • • • • • • •
1 dk Code Vew Pots Sesson buld Det Bec.6.week.3AR × Pilec.6.week.3BR Bec.6.week.3AR × Pilec.6.week.3BR G4 + # G4 + # G6 + # Multiplication milliplication G7 + # Scalar multiplication G6 + # Dot product	O lec,6,week,3CR × P e e e e e e e e e e e e e e e e	Source - R	R • Glo Values	• • • 179 MiB • • • • • • • • • • • • • • • • • • •
1dt Code Vew Poto Senson Build Dete 0 lec.6.week.3AR × 0 lec.6.week.3AR O 0 lec.6.week.3AR × 0 lec.6.week.3AR 0 lec.6.week.3AR × 0 lec.6.week.3AR 0 def. # Source on Save Q 76 # Dot product 77 # # Use dot() of the 'be 79 # Install geometry lib 80 # No	O lec.6.week.3CR Paun	Source • 2 * Configuring Rep *	R • Glo Values) 179 MB •) 179 MB •
<pre>idit Code Vew Poto Session build Del lec.6.week.3AR</pre>	O lec,6,week,3CR × P e e e e e e e e e e e e e e e e	Source • 2 * Configuring Rep *	R • Glo Values) 179 MB • 2) 15) 179 MB • 2) 179 MB • 2<
1dt Code Vew Pots Senson Build Det 0 lec.6.week.3AR Image: Senson Source on Save Image: Sour	V O lec,6,week,3CR × P + Packages Install Packages Install from: Repository (CRAN) Packages (separate multiple with space reportery	Source • 2 * Configuring Rep *	R • Glo Values) 179 MB • 2) 15) 179 MB • 2) 179 MB • 2<
1 dit Code Vew Pots Seuton build Det 0 lec.6.week.3AR ×	V O lec,6,week,3CR × P + Packages Install Packages Install from: Repository (CRAN) Packages (separate multiple with space geometry	Source • 2 * Configuring Rep *	R • Glo Values) 179 MB • 2) 15) 179 MB • 2) 179 MB • 2<
1dt Code Vew Pots Senson Build Det 0 lec.6.week.3AR Image: Senson Source on Save Image: Sour		Source Source Configuring Rep ce or comma):	R • Glo Values) 179 MB • 2) 15) 179 MB • 2) 179 MB • 2<
<pre>idit Code View Plots Sensitiva huld Ded lec.6.week.3AR l</pre>	V O lec,6,week,3CR × P + Packages Install Packages Install from: Repository (CRAN) Packages (separate multiple with space geometry	Source Source Configuring Rep ce or comma):	R • Glo Values) 179 MB •) 179 MB •
idt Code Vew Pets Senton Buld Det ec.6.week.3AR × ec.6.week.3AR × ec.6.week.3AR × idt Code Vew Pets Senton Buld Det ec.6.week.3AR × idt Code Vew Pets Senton Buld Det idt Det Pets Senton Buld Det idt Det Pets Senton Buld Det idt Det Pets Senton Buld Det	V O lec.6, week. 3C.R × Provide the second secon	Source Source Configuring Rep ce or comma):	R • Glo Values	 179 MB • 2 170 mm [1:5] 24 40 56 7. 170 mum [1:5] 13 23 33 4. 171 mum [1:5] 13 27 9 171 mum [1:5] 13 27 9 172 10 28 3.
idit Code View Pots Senion Build Del ec.6.week.3AR × ec.6.week.3AR × ec.6.week.3AR × idit Code View Pots Senion Build Del ec.6.week.3AR × ec.6.week.3AR × idit Code View Pots Senion Build Del		Source Source Configuring Rep ce or comma):	R + Go Values	179 MB 179 MB 179 MB 179 MB 10 115] 24 40 56 7. num [1:5] 12 24 40 56 7. num [1:5] 13 23 33 4. num [1:5] 13 25 7 9 num [1:5] 13 27 3.
<pre>1dt Code Wee Pets Senton Build Ded lec.6.week.3AR Pets Senton Build Ded lec.6.week.3AR Sector Save Cetton lec.6.week.3AR Sector Save Cetton Source on Save Cetton G6+ # Multiplication G7+ # Scalar multiplication G7+ # Scalar multiplication G7+ # Use dot() of the 'Be T7 # Use dot() of the 'Be T7 # Install geometry lib 80 11 library(geometry) 82 83 # Calculate dot product 84 b <- dot(u, v) 7821 Dotproduct: R R4.12-C/dab/ Pets 11 5 </pre>	V O lec.6, week. 3C.R × Provide the second secon	Source Source Configuring Rep ce or comma):	R + Go Values	 179 MB • 2 170 mm [1:5] 24 40 56 7. 170 mum [1:5] 13 23 33 4. 171 mum [1:5] 13 27 9 171 mum [1:5] 13 27 9 172 10 28 3.
<pre>1 Idt Code Vew Pets Senson Build Ced lec.6.week.3AR Code Vew Pets Senson Build Ced lec.6.week.3AR Code Vew Pets Senson Build Ced lec.6.week.3AR Code Vew Pets Senson Build Ced Ced Vew Pets Senson Build Ced Vew Pets Senson Build Ced Ced Vew Pets Senson Build Ced Vew Pets Senson Build Ced Ced Vew Pets Senson Buil</pre>	V O lec.6, week. 3C.R × Provide the second secon	Source Source P Source P Configuring Rep Configuring Rep Rivin-library/4.1 [Defe	R + Gio Values	179 MB 179 MB 179 MB 179 MB 179 MB 171 17
<pre>idit Code Vew Plots Senson Build Del lec.6.week.3AR * Plec.6.week.3AR code Vew Plots Senson Build Del lec.6.week.3AR * Plec.6.week.3BR code version Save Code code V</pre>	V O lec.6, week. 3C.R × Provide the second secon	Source Source Configuring Rep Configuring Rep Revin-library/4.1 [Defe	R + Go Values	179 MB 179 MB 179 MB 179 MB 10 115] 24 40 56 7. num [1:5] 12 24 40 56 7. num [1:5] 13 23 33 4. num [1:5] 13 25 7 9 num [1:5] 13 27 3.
<pre>66 - # Multiplication 67 - # Scalar multiplication 76 - # Dot product 77 78 # Use dot() of the 'pe 79 # Install geometry lib 80 81 library(geometry) 82 83 # Calculate dot product 83 # Calculate dot product 84 b <- dot(u, v) 97 7821 Dotproduct : R R4.12 - Cydab/ @ (1] 5 > p <- u + 2 > print(p)</pre>	V O lec.6, week. 3C.R × Provide the second secon	Source Source P Source P Configuring Rep Configuring Rep Rivin-library/4.1 [Defe	R + Gio Values	179 MB 179 MB 179 MB 179 MB 10 115] 24 40 56 7. num [1:5] 12 24 40 56 7. num [1:5] 13 23 33 4. num [1:5] 13 25 7 9 num [1:5] 13 27 3.
<pre>idit Code Vew Plots Senson Build Del lec.6.week.3AR * Plec.6.week.3AR code Vew Plots Senson Build Del lec.6.week.3AR * Plec.6.week.3BR code version Save Code code V</pre>	V O lec.6, week. 3C.R × Provide the second secon	Source Source P Source P Configuring Rep Configuring Rep Rivin-library/4.1 [Defe	R + Gio Values	179 MB 179 MB 179 MB 179 MB 10 115] 24 40 56 7. num [1:5] 12 24 40 56 7. num [1:5] 13 23 33 4. num [1:5] 13 25 7 9 num [1:5] 13 27 3.
<pre>1 dit Code Vew Pets Seuton build Ded lec.6.week.3AR 2 lec.6.week.3AR 2</pre>	V O lec.6, week. 3C.R × Provide the second secon	Source Source P Source P Configuring Rep Configuring Rep Rivin-library/4.1 [Defe	R + Gio Values	179 MB 179 MB 179 MB 179 MB 10 115] 24 40 56 7. num [1:5] 12 24 40 56 7. num [1:5] 13 23 33 4. num [1:5] 13 25 7 9 num [1:5] 13 27 3.
<pre>1 dit Code Vee Ptot Senson build Ded lec.6.week.3AR 2 lec.6.week.3AR C C C C C C C C C C C C C C C C C C C</pre>	V O lec.6, week. 3C.R × Provide the second secon	Source Source P Source P Configuring Rep Configuring Rep Rivin-library/4.1 [Defe	R + Gio Values	179 MB 179 MB 179 MB 179 MB 179 MB 171 17

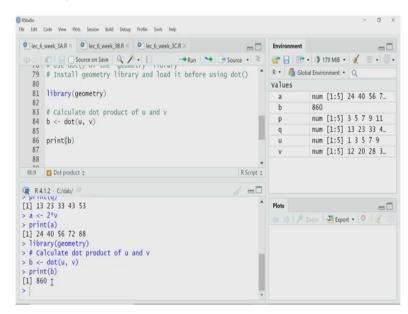
library(geometry)

But there is a library of R which is named as geometry, geometry library. And that library has a function called dot and that can do it in one line, the whole calculation of the dot product. So, just I have to give the two vectors as a argument to this dot function and it will give me the dot product. So, what I have done here, I have already installed this geometry library. If you do not know how to install, I will just show you briefly you go to this tool for this R Studio and then you click on this install package then it will ask what you want to install.

So, you can write here geometry and then you would simply click Install, I have already installed it so I will not do anything further. Now, once you have installed the library in R

does not mean that library is now accessible to you while you are working. You have to make that library accessible to the session current session of work. So, how we do that for any library, if you have to incorporate in the current session, you have to call that library.

(Refer Slide Time: 10:08)



$$b \leftarrow dot(u, v)$$

print(b)

So, we call that by using the function library, and then as the argument in the round bracket, you have to put the name of the library that you are calling and then you will use subsequently. So, I want to use the dot function, which is part of the geometry library, which I have installed earlier. Now, I want to use the dot, so I have to call the geometry library. So, I am calling geometry library here, in this line.

So, let me call that and the geometry library is now loaded in this session. So, now, what I will do, I will use this dot function to calculate dot product between u and v. So, u and v will be two arguments for this function dot. And if you remember, what should I get by the dot product? Do you remember? I should get a scalar. So, it is a scalar product. So, I am assigning that value to b.

So, let us perform that dot product. I have done the dot product calculation, now I want to print it, it is 860. So, I have shown you how to do simply the addition and multiplication in R for vectors. Now in vectors lecture, if you do not remember, go back and check we have also learned about magnitude or norm of a vector, we have learned what is called unit vectors,

then we have learned what is the angle between two vectors. So, now, I will implement those in R.

(Refer Slide Time: 11:41)

96 # Norm, Unit vector, and angle between vectors				-
96 + # Norm, Unit vector, and angle between vectors	🐏 🕒 Source 🔹 🧧 🚰	0.0	179 MiB 🔹 🥑	= • (
	s 📖 🔺 R 🔹	Global Envir	onment • Q	
98 • # Magnitude of a vector	valu	ies		
99 100 # Using norm()	a	num	[1:5] 24 40	56 7
101	b	860		
102 # Get the magnitude of vector u by norm() fur	nction p	num	[1:5] 3 5 7	9 11
103 x <- norm(u, type="2")	9		[1:5] 13 23	
104	u u		[1:5] 1 3 5	
105 print(x) 106	v		[1:5] 12 20	
100 # Explicitly by definition	* X		845232578665	
103:1 🔲 Magnitude of a vector \$	R Script ¢			
😰 R 4.1.2 · C:/dab/ 🔅	1 =0			
library(geometry)	* Plots			-
# Calculate dot product of u and v			-	
<pre>b <- dot(u, v)</pre>		📭 🖓 Zoom	- Export • 0	12
<pre>print(b)</pre>				
[1] 860				
<pre>x <- norm(u, type="2")</pre>				
> print(x) [1] 12.84523 I				

$x \leftarrow norm(u, type = "2")$

print(x)

So, the first thing that I will check is how to calculate the magnitude of a vector in R. So, what I will do, I will use norm function which is a base function present in the base R. So, these norm function will take two argument at least, that is one should be the vector for which you calculate the norm. And then you have to specify what type of norm. There are many types of norm, actually, the magnitude that we calculate for a vector is called the second norm, norm 2.

So, I have to specify that. So, what I am doing here in this line, I am saying that okay, you calculate norm, I am calling the norm function. And I am giving two inputs, two arguments, u is a vector for which I want to calculate the magnitude. And I am writing type is equal to 2. So, I am asking it to calculate the second norm, and it should assign that value, that will be a scalar value to x, and in the next line, I will print that value.

So, let me execute both of them together. So, it says the second norm or magnitude of my vector u is 12.84 something. So, I have calculated the norm. Now, I want to calculate unit vector. If you remember, unit vector is nothing but dividing a vector by its magnitude. So, you have taken a vector and you are normalizing with respect to its magnitude, so you will divide that vector by the second norm. So, that is what I will do simply.

(Refer Slide Time: 13:14)

e] lec_6_	week_3A.R × Olec_6_week_3B.R × Olec_6_week_3C.R ×	-0	Environment	-
	🔊 🔒 🖸 Source on Save 🔍 🎢 🔹 🚺 🔿 Run 😁 🔿 So	urce 🔹 🖹	💣 🔒 🐨	🔹 🌒 179 MiB 🔹 🕖 🚊 📲
	# Magnitude of a vector		R 🔹 🦺 Glo	bal Environment • Q
113 -	# Unit vector		Values	
	# Get the unit vector of u		a	num [1:5] 24 40 56 7
	i <- u/norm(u, type="2")		b	860
117			i	num [1:5] 0.0778 0.2
118	print(i)			num [1:5] 3 5 7 9 11
119			p	num [1:5] 13 23 33 4
120	# Check the magnitude of the unit vector i	q	num [1:5] 1 3 5 7 9	
121 122	<pre>print(norm(i, type="2"))</pre>		v	num [1:5] 12 20 28 3
	Unit vector \$	R Script ¢	x	12.8452325786651
110.9	Contractor \$	K asript 🗸	X	12.0432323700031
	1.2 · C:/dab/ 🗇	1 -0		
[1] 86			Plots	-
	norm(u, type="2")			
5	norm(of office a)		@ @ P	Zoom 🛛 📲 Export 🔹 🔯 🖉
> prin	t(x)			
	.84523			
	t the unit vector of u			
	u/norm(u, type="2")			
> prin				
[1] 9.	07784989 0.23354968 0.38924947 0.54494926 0.70064905			

$i \leftarrow u/norm(u, type = "2")$

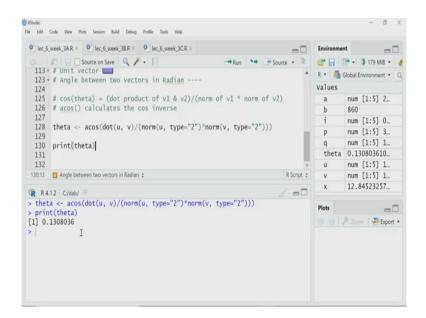
print(i)

print(norm(i, type = "2"))

So, that is what I am doing here in this line, I am taking that u vector, because I want to create a unit vector corresponding to u. So, I am taking that u vector, and then dividing it by its magnitude, just now I calculated the magnitude using the norm function. So, I am using the same norm function again.

And I will get a new vector, remember, unit vector is a vector, and I get a new vector that I assign to a variable called i. So, i will be my unit vector corresponding to u. So, I execute that, the i is created, but I want to see it. So, I write print i, and this is what I got. This is your unit vector corresponding to the vector u that we created earlier. So, just by simply dividing the norm, I got it.

(Refer Slide Time: 14:08)



theta \leftarrow acos(dot(u, v) / norm(u, type = "2")* norm(v, type = "2"))

print(theta)

Now, I will go to the last operations for vectors that you have learned in our earlier lecture is that angle between two vectors. So, let me explain what is the angle, how we define the angle between two vectors. Usually, we know the relationship that the angle, the angle is theta, angle between two vectors is theta, then the Cos of that, Cos theta is equal to dot product of these two vectors.

Suppose that two vectors are v1 and v2. So, you take a dot product of this vector and divide by the product of the magnitude of v1 and magnitude of v2. So, that is we know is the definition of Cos theta, when the theta is the angle between two vectors. So, I will use this formula to calculate Cos theta, and then I will inverse it, so that I will use Cos inverse to get the value of theta. So, how I am doing it, it is bit long, I will extend this.

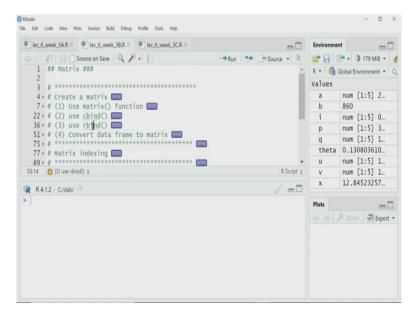
So, the first thing is I have to take that dot product of v1 and v2 two vectors. So, I am using two vectors u and v. So, here I have taken it dot of u and v, so I am using the dot function to calculate the dot product, then I have to divide that by norm of these two vectors. So, the first thing I am taking and in the denominator here, you see the division operator, and then I am dividing by norm of u and the second norm of v, they are multiplied by this multiplication operator.

So, this whole thing, this whole thing will give me Cos theta, but I want to know the angle in radians. So, I will use Cos inverse. So, that is why I use a Cos function. And that Cos

function will invert this and will give me the angle which I assigned to theta, theta is the angle in radian. And in the next line, I want to print that angle. So, it says the angle between these two vectors u and v is 0.1308 in radians.

So, it is very easy to do, once I have the dot function and norm function, I can easily do that. That is all for vectors, there are many other things that you can play around with vectors, but we will not go into those, I am sticking to what we have discussed in our linear algebra classes. So, now I will move to matrices and I will create first matrices, I will show how to create matrices. And then we will do matrix operations like multiplication, addition, then we have to learn eigen value, how to calculate eigen value, eigen vectors, determinant of a matrix and so on.

(Refer Slide Time: 16:52)



So, let us start with creation of a matrix. How can I create a matrix? Here, I will show four methods. The first one will use the matrix function, then the second one I will show using cbind function, the third with rbind, and eventually I will show how to convert a data frame, if you know the table format of data into matrix. So, let us start with first the matrix function, how the matrix function helps us? Matrix function will convert a vector into a matrix. So, I will give a vector as an input to this matrix function and it will convert it into a matrix. So, let me explain with an example.

(Refer Slide Time: 17:37)

e lec_o	week_3A.R × O lec_6_week_3B.R × O lec_6_week_3C.R ×	1	Environment					=[
	🔊 🔒 🖸 Source on Save 🔍 🎢 📲 🔂 🖼 🖓 Source 🔹 考		🚰 🔒 🔛	• 3 17	9 MiB 👻	1		• @
10		*	R • Glob	al Enviror	ment •	Q		
11	# create vector		Values					
12	d <- c(1:10)	t i	a	num	[1:5]	24 40	56	72
13 14	# create matrix using the vector		b	860	[210]	21 10	50	
15	<pre>m <- matrix(d, nrow = 2, ncol = 5, byrow = FALSE)</pre>		d		[1:10]	1 2	34	5
16	in a matrix(a) mon - e, near - s, ayran - mesey		i		[1:5]			
17	print(m)		p		[1:5]			
18			q		[1:5]			
19	# check the class		theta		080361			1.0
20	class(m) (1) Use matrix() function : R Script :		u		[1:5]			
			v		[1:5]			
R R4	.1.2 · C:/dab/ ∞ d =□		•		[1.0]	16 20	-	50
	eate vector		Plots					-
	c(1:10)					. 10		
d [1]	1 2 3 4 5 6 7 8 9 10 .		(a a) P :	200m •	Expor	t• @		
	1 2 3 4 3 6 7 8 9 10 I							

 $d \leftarrow c(1:10)$

 $m \leftarrow matrix(d, nrow = 2, ncol = 5, byrow = FALSE)$

print(m)

So, in the first line, here, I am creating a vector d using c function. And I am saying start with 1 end at 10. So, it will start at 1 and then take 2, 3 and 4 up to 10 and it will create a vector d. So, let me execute that. So, now it has created I can show you here this is d, it is written d and I have got those numbers. I can see also here in the console, if I write d I can see the vector.

Now, I want to convert these vector which is values from 1 to 10 into a matrix, that means I have to break it in part and arrange them so that this 1-dimensional thing will become a 2-dimensional thing. So, that means I have to tell the function, who will do this job, the matrix function will do this job, that means I have to tell matrix function that how many rows I want and how many columns I want.

So, that is what I have done here. I am calling the matrix function. And the arguments are within the round bracket. The first argument is d of this is the vector which is the input, and then I am telling how many rows and how many columns. Here, I have said the n row, that is number of rows that I want in the new matrix is 2 and the number of columns ncol is equal to 5, 2 into 5 is 10. And I have 10 element, 10 numbers in my vector d.

So, that means I want to take these 10 numbers which are arranged in 1-dimension, and I want to break them to create two rows and five columns. Now, you can do it in two way. So, you can start like this. You can arrange them the 1, 2, 3, 4, 5 and 6, 7, 8, 9, 10, something like that. So, that will be row wise, or you can start with 1, 2, 3, 4, 5, 6, something like that, that will be column wise. So, that is why the last argument here is by row and I have said FALSE. That means I do not want to row wise, I want column wise. Let me execute it. It will be more clear.

(Refer Slide Time: 19:47)

	week_3AR × • lec_6_week_3B.R × • lec_6_week_3C.R ×	Environment	-
	🔊 🔒 🖸 Source on Save 🔍 🎢 🔹 📑 Run 😁 📑 Source 🔹 🛎		• 🚺 179 MiB • 🥑 🔳 •
10 11	· · · · · · · · · · · · · · · · · · ·	R 🔹 💼 Glob	al Environment * Q
11	# create vector d <- c(1:10)	Data	
13	u (- c(1.10)	m jij	int [1:2, 1:5] 1 2
14	# create matrix using the vector	Values	
15	<pre>m <- matrix(d, nrow = 2, ncol = 5, byrow = FALSE)</pre>	a	num [1:5] 24 40 56 72.
16		b	860
	print(m)	d	int [1:10] 1 2 3 4 5 .
18	A shared when the state	i	num [1:5] 0.0778 0.23.
19	# check the class class(m)	p	num [1:5] 3 5 7 9 11
	🔁 (1) Use matrix() function o R Script o	q	num [1:5] 13 23 33 43.
		theta	0.130803610419603
	1.2 · C:/dab/ ∞		
	eate vector	Plots	_
d <-	c(1:10)		Zoom 📲 Export 🔹 🔍 🧹
	1 2 3 4 5 6 7 8 9 10	14 411	
	eate matrix using the vector		
m <-	<pre>matrix(d, nrow = 2, ncol = 5, byrow = FALSE)</pre>		

liec_0_wee	k_3A.R ×	9 lec_6_	week_3	BB.R ×	m ×	e lec.	6_week_3C.R ×		Environment	-
0010	I ⇒ Filter						Q,		🚰 🔒 🖙 ·	• 🌖 179 MiB • 🕖 📃 •
^ V1	¥ V2	÷ V3		V4 - *	V5				R 🔹 💼 Glob	al Environment • Q
1	1	3	5	7		9			Data	
2	2	4	6	8	,	0			m	int [1:2, 1:5] 1 2
			0	0					Values	
									a	num [1:5] 24 40 56 72
									b	860
									d	int [1:10] 1 2 3 4 5
									i	num [1:5] 0.0778 0.23
having the	and the state law	E tatal an							р	num [1:5] 3 5 7 9 11
howing 1 to	or 2 entries	, o total co	umns						q	num [1:5] 13 23 33 43
R 4.1.2	· C:/dab/							1 -0	theta	0.130803610419603
# crea									Plots	
> d <- c	(1:10)									-
> d									(m m) / p z	Coom 📲 Export 🔹 🔍 🥖
[1] 1 > # crea	2 3 4									
> # crea > m <- m						row =	FALSE)			
> View(m			- 1		, ,		(never			

 $d \leftarrow c(1:10)$

 $m \leftarrow matrix(d, nrow = 2, ncol = 5, byrow = FALSE)$

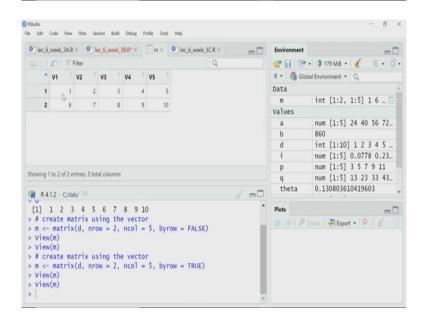
print(m)

class(m)

I have created m. You can go to this environment m double click it and you can see it here. See, originally I have 1 2 3 4 so on up to 10. Now, I have broken that in two row five column, but not row wise, I have said row wise is equal to FALSE, by row is FALSE. So, that means it is column row wise, that is why it is taken 1 2 3 4 5 6 7 8 9 10. So, that is how it has arranged.

(Refer Slide Time: 20:19)

	week_3AR × Olec_6_week_3B.R* × Olec_6_week_3CR ×	Environment	-		
	🔊 🔒 🖸 Source on Save 🔍 🎢 🔹 📑 Run 😁 📑 Source 🔹 🖻	er 🔒 🐨	• 🌖 179 MiB • 🥑 🔳 •		
10	*	R 🔹 💼 Glob	al Environment • Q		
11	# create vector	Data			
12 13	d <- c(1:10)	m	int [1:2, 1:5] 1 6		
14	# create matrix using the vector	Varues			
15	<pre>m <- matrix(d, nrow = 2, ncol = 5, byrow = TRUE)</pre>	a	num [1:5] 24 40 56 72		
16		b	860		
17	print(m)	d	int [1:10] 1 2 3 4 5		
18		i	num [1:5] 0.0778 0.23		
19	# check the class	p	num [1:5] 3 5 7 9 11		
20	class(m) (1) Use matrix() function a R Script a	q	num [1:5] 13 23 33 43		
R A	12 · C/dab/ 🗇 🥼 🗖	theta	0.130803610419603		
*	c(1:10)	Plots			
d [1]	1 2 3 4 5 6 7 8 9 10	🦛 📦 🏓 Zoom 😕 Export - 🔍 🥖			
	eate matrix using the vector				
	matrix(d, nrow = 2, ncol = 5, byrow = FALSE)				
View					
View	(m)				
	eate matrix using the vector				
-	<pre>matrix(d, nrow = 2, ncol = 5, byrow = TRUE)</pre>				



d← **c**(1 : 10)

 $m \leftarrow matrix(d, nrow = 2, ncol = 5, byrow = TRUE)$

print(m)

class(m)

If I have said it equal to this equal to TRUE, then if I execute it again. Now, let me see m. Now, you see it is row wise 1 2 3 4 up to 5 in the first row, and the rest of the data or numbers are in the second row. This is how we actually can convert a vector into a matrix.

(Refer Slide Time: 20:49)

17 print(m) 18 # check the class 20 class(m) 21 # check the class 22.# # (2) use cbind() # check the class 23.# # (4) Convert data frame to matrix # check the class 77.# Matrix indexing # check the class 9 (1)Use matrix(d, nrow = 2, ncol = 3, byrow = rAUE) 9 (1)Use matrix(d, nrow = 2, ncol = 5, byrow = TRUE) 9 # check the class 9 class(m)	ec_6_week_3AR × 🔍 lec_6_week_3B.R* × 🔍 lec_6_week_3C.R × 👝 🗖	Environment	-		
18 19 # check the class 19 # check the class 20 class(m) 21 int [1:2, 1:5] 1 6 22.# # (2) use cbind() 23.# # (4) Convert data frame to matrix 25.# # (4) Convert data frame to matrix 20 21 22.# (2) use cbind() 23.# # (4) Convert data frame to matrix 209 (1)Use matrix[unction : R R 412- C/dab/ witew(m) view(m) * Check the class	💷 🔒 🖸 Source on Save 🔍 🎢 📄 🖶 Run 😁 🕞 Source 🔹 🔍	🐨 🔒 🖙			
<pre>19 # check the class 20 class(m) 21 22 # # (2) use cbind()</pre>	17 print(m) *	R 🔹 💼 Glob	al Environment • Q		
20 class[m] 21 22 ≠ # (2) use cbind() 35 ≠ # (3) use rbind() 51 ≠ # (4) Convert data frame to matrix 75 ≠ # datrix indexing 00 ■ (1) Use matrix 0 function : R R412 • C/dab/ view(m) view(m) view(m) view(m) view(m) 20 − 2, ncol = 5, byrow = TRUE) view(m) view(m) 21 − 2, ncol = 5, byrow = TRUE) 22 + # (2) use cbind() 23 − 2, ncol = 5, byrow = TRUE) 24 − 2, ncol = 5, byrow = TRUE) 25 − 2, ncol = 5, byrow = TRUE) 26 − 2, ncol = 5, byrow = TRUE) 27 − 2, ncol = 5, byrow = TRUE		Data			
<pre>21 22. # (2) use cbind()</pre>		m	int [1:2, 1:5] 1 6		
22 ≠ # (2) use cbind() 36 ≠ (3) use rbind() 51 ≠ (4) Convert data frame to matrix 75 ≠ # Matrix indexing 209 © (1)Use matrin() function : R 84.12 • C/dab/ m ~ matrix(u, HrOm = 2, HEOT = 3, DyTOM = FALSE/ View(m) * create matrix using the vector m <- matrix(u, nrow = 2, ncol = 5, byrow = TRUE) View(m) * check the class		Values			
36 * # (3) use rbind() 51 * # (4) Convert data frame to matrix 75 * # *********************************		a	num [1:5] 24 40 56 72.		
75, # ************************************		b	860		
77, # Matrix indexing 00. ***********************************		d	int [1:10] 1 2 3 4 5 .		
00. 0 10 Use matrix function : P num [1:5] 3 5 7 9 11 209 10 Use matrix function : R Script : num [1:5] 13 23 33 43. R R412 - C/dab/ Image: Script : num [1:5] 13 23 33 43. Wew(m) Image: Script : Image: Script : View(m) Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : View(m) Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script : Image: Script :		i	num [1:5] 0.0778 0.23		
R R412 · C/dab/ @ R R412 · C/dab/ @ Wiew(m) View(m) View(m) x c- matrix(d, nrow = 2, ncol = 5, byrow = TRUE) View(m) x c- matrix(d, nrow = 2, ncol = 5, byrow = TRUE) View(m) x c- matrix(d, nrow = 2, ncol = 5, byrow = TRUE) x c- matrix(d	00. # 000000000000000000000000000000000	р	num [1:5] 3 5 7 9 11		
R R412: C(db/ @ View(m) View	20.9 🖪 (1) Use matrix() function 🗧 🛛 R Script 🗧	q	num [1:5] 13 23 33 43.		
<pre>view(m) view(m) # create matrix using the vector m <- matrix(d, nrow = 2, ncol = 5, byrow = TRUE) view(m) view(m) # check the class</pre>	R 4.1.2 · C/dab/ @	theta	0.130803610419603		
<pre>View(m) # create matrix using the vector m <- matrix(d, nrow = 2, ncol = 5, byrow = TRUE) View(m) # check the class</pre>		Plots			
<pre># create matrix using the vector m <- matrix(d, nrow = 2, ncol = 5, byrow = TRUE) View(m) View(m) # check the class</pre>					
view(m) view(m) # check the class			coom 😬 Export 🔹 🔍 🦉		
<pre>view(m) # check the class</pre>					
# check the class					

Now, let us see run one handy tool is to check that what we are doing. Are we really creating metrics or not is to call the class, class will tell me what data class is this variable belongs to. So, I have already created m. So, let me check what is this data class? And it says it is a matrix or array. That is what you want. You want a vector to convert into matrix or array. So, I have shown using matrix function.

(Refer Slide Time: 21:14)

	Environment	-			
💷 🖉 🔚 🖸 Source on Save 🔍 🎢 🗧 📑 Run 🐤 📑 Source 🔹 😤	🚰 🔒 I 👺	🔹 🌖 179 MiB 🔹 🥖 🗮 🔹			
25 # Each vector is a column of the matrix A	R • 💼 Global Environment • Q Data				
20 27 # create two vectors					
28 d <- c(1:10)	m	int [1:10, 1:2] 1 2			
29 e <- c(21:30)	Values				
30	a	num [1:5] 24 40 56 72			
31 # bind the vectors to make matrix	b	860			
32 m <- cbind(d, e)	d	int [1:10] 1 2 3 4 5			
33 34 print(m)	e	int [1:10] 21 22 23 2			
35 *	i	num [1:5] 0.0778 0.23			
32:17 🔲 (2) use cbind() c R Script c	р	num [1:5] 3 5 7 9 11			
R R4.1.2 · C/dab/ 🗇 🚽 📼	q	num [1:5] 13 23 33 43			
m st macrix(u, mow = 2, mcor = 3, byrow = ikoc)					
View(m)	Plots	-			
View(m) # check the class	4.01 1	Zoom 🚈 Export 🔹 🔯 🛛 🥑			
+ check the class I class(m)					
1] "matrix" "array"					
d <- c(1:10)					
e <- c(21:30)					
# bind the vectors to make matrix					
m <- cbind(d, e)					

	veek_3A.R ×		=
	🗈 🕞 🗋 Source on Save 🔍 🎢 👘 👘 Source 🔹 👘	🚰 🔒 l 👺	• 🚺 179 MiB • 🕖 🔳 •
	# Each vector is a column of the matrix	R 🔹 🎒 Glob	pal Environment • Q
26 27	# create two vectors	Data	
	d <- c(1:10)	m	int [1:10, 1:2] 1 2
	e <- c(21:30)	Values	
34:9	(2) use cbind() a R Script a	a	num [1:5] 24 40 56 72
		b	860
R R 4.1.	2 · C/dab/ 🖗 🦪 🗖	d	int [1:10] 1 2 3 4 5
	c(21:30)	e	int [1:10] 21 22 23 2
	d the vectors to make matrix cbind(d, e)	i	num [1:5] 0.0778 0.23
print		р	num [1:5] 3 5 7 9 11
	d _T e	q	num [1:5] 13 23 33 43
	1-21		
	2 22	Plots	-
	3 23	(a a) (P)	Zoom 📲 Export 🔹 🙆 🛛 🥖
[4,]	4 24 5 25		
[5,]	6 26		
[5,] [6,]	6 26 7 27		
[5,] [6,] [7,]			

$d \leftarrow c(1:10)$

 $e \leftarrow c(21:30)$

 $m \leftarrow cbind(d, e)$

print(m)

Now, I will use cbind function, column bind, bind by column. What is that? So here, I want to combine vectors to create matrices. In the earlier case, I had one vector, break it down in matrices. Now, I have multiple vectors, I want to join them to create matrix. And in this case,

these original vectors, if you cbind, will be the columns of the new matrix. So, first, let me create two vectors to explain this.

So, I am creating two vectors, d and e here, in these two lines. It is a very simple, I am using the c function. So, I have created two vectors, d and e. Now, I want to combine them to create a matrix where these d and e will be columns. So, how I will do? I will call a cbind function. And the arguments are those vectors. So, the first vector will be d, the first column should be d, second column should be e.

So, if I execute it. So, I have created, let me print it. Here it goes. So, you see, the first column is d, second column is e, and vector d is there arranged as a column and e arranged as a column. So, I have created a new matrix m by binding column wise these two vectors. Now, I will use rbind, means I will bind vectors similarly, but row wise, the original vectors will become rows of the new matrix.

(Refer Slide Time: 22:56)

lec_6	week_3A.R × • lec_6_week_3B.R* × • lec_6_week_3C.R		Environment	-0
	🔊 🔒 🖸 Source on Save 🔍 🎢 🗧 👄 R	un 🐤 🕞 Source 🔹 🖻	🐨 🔒 😰	• 🌗 179 MiB • 🕖 🔳 • 🕝
39	# Each vector is a row of the matrix		R • 📕 Glob	pal Environment • Q
40			Data	
	# create two vectors		m.	int [1:2, 1:10] 1 2
42	d <- c(1:10)	values		
43	e <- c(21:30)		a	num [1:5] 24 40 56 72
44 45	# bind the vectors to make matrix		b	860
46	m <- rbind(d, e)		d	int [1:10] 1 2 3 4 5
47			e	int [1:10] 21 22 23 2_
48	print(m)		i	num [1:5] 0.0778 0.23
49 46:17	(3) use rbind() \$	R Script :	p	num [1:5] 3 5 7 9 11
40:17	(s) use roma() 😜	K Script 👙	q	num [1:5] 13 23 33 43
R R.4.	1.2 · C:/dab/ 🔅	1 =0	-	
[6,]	6 26		Plots	-0
[7,]	7 27		6 6 8	Zoom 📲 Export 🔹 🔕 🛛 🧹 🛛
8.1.1.1	8 28			
	9 29			
[10,]	c(1:10)			
	c(21:30)			
	nd the vectors to make matrix			
	rbind(d, e)			

lier,	_6_we	ek_3	A.R ×	0	lec_6_	week	3B.R*	×	m×	e le	c_6_w	eek_3C.	R×			Environment	-
	5		7 Filter									l	Q,			💣 🔒 🖙	• 🌗 179 MiB • 🖌 🚊 • 🛛
*	V1		V2	V	3	V4		V5 [±]	V6	° v	, c	V8		V9 [±]	V10 =	R 🔹 🚺 Glob	pal Environment * Q,
d		1	3	2	3		4	5		6	7		8	9	10	Data	
		21	2	5	23		24	25		26	27		28	29	30	m	int [1:2, 1:10] 1 2
		**						6.7								Values	
																a	num [1:5] 24 40 56 72.
																b	860
																d	int [1:10] 1 2 3 4 5 .
																e	int [1:10] 21 22 23 2.
																i	num [1:5] 0.0778 0.23.
howing	g 1 to	2 of	2 entrie	, 10 1	total c	olumn	5									р	num [1:5] 3 5 7 9 11
R	412		/dab/												-0	q	num [1:5] 13 23 33 43.
U 10		28														Plots	-
[8.]		29															Zoom 📲 Export • 0 🥖
																	2.dom 🚈 export • • •
[9,] 10,]			10)														
[9,] 10,] d <	- c		. 202														
e <	(- c	(21	:30)	tor	c +/		10	atriv									
[9,] 10,] d < # t	c- c c- c oind	(21 th			s to	ma	ke r	matrix									

 $d \leftarrow c(1:10)$

 $e \leftarrow c(21:30)$

 $m \leftarrow rbind(d, e)$

print(m)

So, to do that, I am creating again two vectors. And then I am joining them by calling the rbind function. Again, here, the arguments are d and e. So, d will be the first row, e will be the second row. So, I have created it. I can go to either print it or I can go and click on the environment pane on m, I should be able to see. Now, you see, d and e are not, no more columns, they are rows. So, the first row is d, vector originally 1 2 3 up to 10.

And the second row of the new matrix is the e vector starting with 21 up to 30. So, this is how using cbind and rbind I can create new matrices by combining vectors. Now, come to the last thing of creating metrics. I want to convert a data frame, means you have read a data in a table format from a CSV file or you have used inbuilt the data set whatever you are getting you have a data frame, a table. Now, you want to convert that into a matrix and so that you will use matrix operation on that.

(Refer Slide Time: 24:12)

lec_6_week_3A.R × O lec_6_week_3B.R* × O lec_6_week_3C.R ×		Environment	-	
💷 💭 🔂 Source on Save 🔍 🎢 📄 😁 Run 🐤 😁 Source 🖓	. 2	e	• 🕽 180 MiB • 🖌 📃 • 🕻	
49	R • Glob	al Environment · Q		
50	Data			
51 - # (4) Convert data frame to matrix 52	0 d	6 obs. of 2 variabl.		
52 53 # Get the BOD dataset of R	m	int [1:2, 1:10] 1 2		
54 d <- BOD	1	Values		
55		a	num [1:5] 24 40 56 72.	
56 # Check class		b	860	
57 class(d) 58		e	int [1:10] 21 22 23 2.	
59 # Convert to matrix	w	i	num [1:5] 0.0778 0.23.	
57:9 🚦 (4) Convert data frame to matrix 🗧 🛛 R S	Script ¢	р	num [1:5] 3 5 7 9 11	
A 112 A 114 A	-0	q	num [1:5] 13 23 33 43.	
R R4.12 · C/dab/ ∞ # bind the vectors to make matrix		Plots		
<pre># bind the vectors to make matrix m <- rbind(d, e)</pre>	î.		-	
View(m)		(a a) / 2 :	Zoom 🛛 📲 Export 🔹 🔯 🛛 🅖	
View(m)				
# Get the BOD dataset of R				
d <- BOD				
# Check class				
class(d)				

	week_SAR	lec_6_v	ek_38.R* × d × 🕐 lec_6_week_3C.R × 👘 🗖	Environment	=
	2 VA	lter	Q,	🚰 🔒 🐨	• 🌗 180 MiB • 🥑 🔳 •
*	Time	demand		R 🔹 💼 Glob	al Environment • Q,
1	1	8.3		Data	
2	2	10.3		0 d	6 obs. of 2 variabl
3	3	19.0		m	int [1:2, 1:10] 1 2
				Values	
4	4	16.0		a	num [1:5] 24 40 56 72
5	5	15.6		b	860
6	7	19.8		e	int [1:10] 21 22 23 2
howing 1	to 6 of 6 en	tries, 2 total coli	nos	i	num [1:5] 0.0778 0.23
ioning i	10 0 01 0 01	inna, a solar con		p q	num [1:5] 3 5 7 9 11 num [1:5] 13 23 33 43
R R.4.	1.2 · C:/dat		d =0	ч	num [1.5] 15 25 55 45
View	(m)			Plots	=
				0012	Zoom 🛛 📲 Export 🔹 🔯 🧹
View	t the R(OD dataset	of R		
# Ge					
# Ge d <-	BOD	is			
# Ge d <-	BOD eck clas	55			

 $d \leftarrow BOD$

class(d)

```
m \leftarrow data.matrix(d)
```

print(m)

So, for that what I will do, I use a function called data dot matrix. So, before doing that, I need some data in table format or a data frame format. So, what I will do I will use the BOD data set which is inbuilt in R. So, in your machine when you have installed R you have that, so I will call that data and assign it to a new variable d. So, you can see, let me check the class of these d what type of data is this, let me first read it, I have read the data.

Now, I want to check the class, the class it is saying d is a data frame. And you can double click here also to see here, it has time and demand, oxygen demand actually. So, you have this table. So, now I want to create a matrix out of this table. So, I want to convert these from data frame to matrix format.

(Refer Slide Time: 24:59)

lec 6 w	reek 3A.R ×	0 lec_6_week_38.R* × 0	lec 6 week 3C.R ×		Environment	-
			Run 🏞 🕒 Sour			•) 180 MiB • 🥑 🔳 • 🛛
52		source on save	S Er Kun I Er I Er sour	rce • ·		
		BOD dataset of R				al Environment • Q
	d <- BOD				Data	
55	# Check d	lace			0 d	6 obs. of 2 variabl
	class(d)	1455			m	num [1:6, 1:2] 1 2
58					Values	
		t to matrix			a	num [1:5] 24 40 56 72.
60 r 61	m <- data	a.matrix(d)			b	860
	print(m)				e i	int [1:10] 21 22 23 2. num [1:5] 0.0778 0.23.
<i>c</i> 2		data farma ta mateir A		T Covint A	p	num [1:5] 3 5 7 9 11
62:9	(4) Conven	t data frame to matrix 🗘		R Script \$	q	num [1:5] 13 23 33 43.
R A.1.	.2 · C:/dab/			/ =0	ч	
> # Get	the BOD	dataset of R			Plots	-
> d <-	BOD ck class				4	Zoom 📲 Export 🔹 🔕 🛛 🧹
	data.mat	rix(d)		v		
Studio		rix(d)	Tools Help	×		- σ
Studio e Edit Ce	ode View Pic				Environment	_
Studio F Edit Co P Edit Co	ode View Pic	rts Session Build Debug Profile Inter_6_week_38.R* ×				-
> 50udio e Edit Co @ lec_6_w	ode View Pic reek_3A.R ×	rts Session Build Debug Profile Inter_6_week_38.R* ×	m × • lec_6_week_3C.R ×	-0	🕋 🔒 🖙	_
> 50udio e Edit Co @ lec_6_w	ode View Pic reek_3A.R ×	ns Session Build Debug Profile	m × • lec_6_week_3C.R ×	-	🕋 🔒 🖙	- 🚺 180 MiB - 🖌 🗮 - 1
Situdio e Edit Co e lec_6_w	ode View Pic reek_3A.R ×	ns Session Build Debug Profile Image: Session Build Debug Image: Session Build Image: Session Build Image: Session Build Image: Session Build	m × • lec_6_week_3C.R ×	-0	🔐 🔒 🛛 😁 R 🔹 🚮 Glob	- 🚺 180 MiB - 🖌 🗮 - 1
Studio e Edit Co e lec_6_w 1 1 2	ode View Pk reek_3AR × Filter 1 2	ns Sesson Build Debug Profile Piec.6.week.38.R* × mand 8.3 10.3	m × • lec_6_week_3C.R ×		R • Glob Data 0 d m	
Studio e Edit Ce e Edit Co e Edit Ce e Edit Ce e Edit Ce e Edit Ce e Edit Ce e Edit Co e Edit Co Edit Co Ed	ode View Pk reek_3AR × Filter 1 2 3	ns Sesson Build Debug Profile Piec.6.week.38.R* × mand 8.3 10.3 19.0	m × • lec_6_week_3C.R ×	-	R • Glob Data O d M Values	 180 MB • 2 = • 0 al Environment • Q 6 obs. of 2 variabl num [1:6, 1:2] 1 2
Staudio e Edit Ce e Edit Co e Edit Co Edit Co Edit Co Edit Co Edit Co Edit Co Edit Co Edit Co Edit Co E	ode View Pk reek_3AR × Filter 1 2 3 4	nts Session Build Debug Profile Piec.6.week.38.R* × mand 8.3 10.3 19.0 16.0	m × • lec_6_week_3C.R ×		R · Glob Data O d M Values a	 180 MB • 2 180 MB • 2 al Environment • Q 6 obs. of 2 variabl. num [1:6, 1:2] 1 2 num [1:5] 24 40 56 72.
Studio e Edit Co • I • I 1 2 3 4 5	ode Vew Pk reek_3AR × Filter 1 2 3 4 5	ns Sesson Build Debug Profile Piec.6.week.38.R* × mand 8.3 10.3 19.0	m × • lec_6_week_3C.R ×		R • Glob Data O d m Values a b	 180 MB • 2 = • 0 180 MB • 2 = • 0 al Environment • Q 6 obs. of 2 variabl num [1:6, 1:2] 1 2 num [1:5] 24 40 56 72. 860
Staudio e Edit Ce e Edit Co e Edit Co Edit Co Edit Co Edit Co Edit Co Edit Co Edit Co Edit Co Edit Co E	ode View Pk reek_3AR × Filter 1 2 3 4	nts Session Build Debug Profile Piec.6.week.38.R* × mand 8.3 10.3 19.0 16.0	m × • lec_6_week_3C.R ×		R • Glob Data O d M Values a b b e	 180 MB • 2 180 MB • 2 al Environment • Q 6 obs. of 2 variabl num [1:6, 1:2] 1 2 num [1:5] 24 40 56 72. 860 int [1:10] 21 22 23 2.
Studio e fdit Co e fdit Co e fdit Co f e fdit Co f f f f f f f f f f f f f f f f f f f	ode View Pic reek.3AR × Time de 1 2 3 4 5 7	nts Session Build Debug Profile Profile Rec.6,week_38.R* × mmand 8.3 10.3 19.0 16.0 15.6 19.8	m × • lec_6_week_3C.R ×		R • Glob Data O d M Values a b b e i	 180 MB • 2 180 MB • 2
Studio e fdit Co e fdit Co e fdit Co f e fdit Co f f f f f f f f f f f f f f f f f f f	ode View Pic reek.3AR × Time de 1 2 3 4 5 7	nts Session Build Debug Profile Profile Rec.6.week.38.R* × mand 8.3 10.3 19.0 16.0 15.6	m × • lec_6_week_3C.R ×		R • Glob Data O d m Values a b b e i j p	 180 MB • 2 180 MB • 2 al Environment • Q 6 obs. of 2 variabl num [1:6, 1:2] 1 2 num [1:5] 24 40 56 72. 860 int [1:10] 21 22 23 2. num [1:5] 0.0778 0.23. num [1:5] 3 5 7 9 11
 Showing 1 t 	ode View Pic reek.3AR × Time de 1 2 3 4 5 7	ns Sesson Build Debug Profile Piec.6,week.38.R* × mand 8.3 10.3 19.0 16.0 15.6 19.8 s, 2 total columns	m × Pikc,6,week,3CR ×		R • Glob Data O d M Values a b b e i	 180 MB • 2 180 MB • 2
 Studio Edit Co Edit Co Iec_6_w I 	ode Vew Pic reek 3AR × Filter Time de 1 2 3 4 5 7 7 to 6 of 6 entrie 2 · C:/dab/ Ck class	ns Sesson Build Debug Profile Piec.6,week.38.R* × mand 8.3 10.3 19.0 16.0 15.6 19.8 s, 2 total columns	m × Pikc,6,week,3CR ×		R • Glob Data O d m Values a b b e i j p	 180 MB • 2 180 MB • 2 al Environment • Q 6 obs. of 2 variabl num [1:6, 1:2] 1 2 num [1:5] 24 40 56 72. 860 int [1:10] 21 22 23 2. num [1:5] 0.0778 0.23. num [1:5] 3 5 7 9 11

d ← BOD

class(d)

 $m \leftarrow data.matrix(d)$

print(m)

So, what I will do, I will call these data dot matrix function, the argument should be this d, d is the data frame that I have. And the whole thing I will assign to a new matrix called m. So, if I execute that, and now if I check in the environment pane, you can easily see m has been created here. And let me double click it you can see I have time and demand as the header and I have the whole data as in a matrix format. But how do I know it is matrix, it looks like a table?

(Refer Slide Time: 25:35)

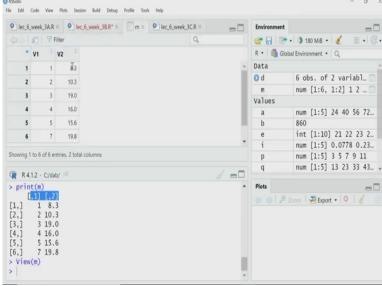
lec_6_w	reek_3A.R ×	lec_6_week_38	R* × 0 lec 6.	week_3C.R ×	-	Environm	ient	-
		ource on Save			•• • Source •		🕐 - 🌖 180 MiB -	
	class(d)			C.P. Mart 1	and a source		Global Environment •	
58	crass(u)					Data	diobal children	~
59	# Convert	to matrix				O d	6 obs. of 2	unninh]
	m <- data	.matrix(d)				m	num [1:6, 1	
61 62	print(m)					Values	num [1.0, 1	
63	princ(m)					a	num [1:5] 2	4 40 56 72
64	# check of	lass				b	860	4 40 50 72
	class(m)					e	int [1:10]	21 22 23 2
66 67	# romovo	column names				- i	num [1:5] 0	
		data frame to matri	(¢		R Scrip	to p	num [1:5] 3	
						q	num [1:5] 1	3 23 33 43
	.2 · C:/dab/				1 =			
View(Plots 		-
View(a) vert to m	atrix					🎤 Zoom 📲 Export	.01
	data.matr							
View(
View(
	ck class							
class								
	trix" "ar	"nav" -						
	urix di	Tay I						
•						Y		- 0
Studio Edit Ce	ode View Pic							- 0
itudio Edit Ce	reek_3A.R ×	ts Session Build						=
itudio Edit Co					3CR ×	er 🔒	😁 • 🜖 180 MiB •	
itudo Edit Co	reek_3A.R ×					er 🔒		
Studio Edit Ce	reek_3A.R ×	lec_6_week_38				er 🔒	Global Environment •	_ ر = ۱
Studio Edit Co lec_6_w	reek_3A.R × Filter Time der I	e lec_6_week_38					😁 • 🜖 180 MiB •	_ ر = ۱
Studio Edit Co Dec_6_w 1 1 2	reek_3A.R × Filter Fime den I 2	lec_6_week_38 mand 8.3 10.3				R • A	Global Environment •	ے ر
Studio Edit Co lec_6_w	reek_3AR × Filter Time den I 2 3	lec_6_week_38 mand				R • 1	Global Environment • 6 obs. of 2	ے ر
Studio Edit Co Dec_6_w 1 1 2	reek_3AR × Filter Time der 1 2 3 4	e lec,6,week,38 nand 6 8,3 10,3 19,0 16,0				R • Data O d m	Global Environment • 6 obs. of 2	<pre></pre>
Studio Edit Co lec_6_w	reek_3AR × Filter Time den I 2 3	lec_6_week_38 mand				R • 1 Data O d m Values	Global Environment • 6 obs. of 2 num [1:6, 1 num [1:5] 2 860	✓ ■ • □ Q variab1 ::2] 1 2 4 40 56 72
Studio Edit Ce Dec_6_w 1 1 1 2 3 4	reek_3AR × Filter Time der 1 2 3 4	e lec,6,week,38 nand 6 8,3 10,3 19,0 16,0				R • Data 0 d m Values a b e	Global Environment • Global Environment • num [1:6, 1 num [1:5] 2 860 int [1:10]	<pre></pre>
Studio Edit Co Edit Co Edita Edit Co Edit Co E	reek_3AR × Filter Time det 2 3 4 5 7	lec.6.week.38 mand 8.3 10.3 19.0 15.6 19.8				R • Data M Values a b	Global Environment • Global Environment • 6 obs. of 2 num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 0	<pre></pre>
Studio Edit Co Edit Co Edita Edit Co Edit Co E	reek_3AR × Filter Time det 2 3 4 5 7	e lec,6,week,38 mand 8.3 10.3 19.0 16.0 15.6				R • Data 0 d m Values a b e	Global Environment • Global Environment • num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 3	<pre>variab1 :2] 1 2 4 40 56 72 21 22 23 2 .0778 0.23 5 7 9 11</pre>
hudo fdt Ca P lec.6.w 1 1 1 2 3 4 5 6 howing 1 t	reek_3AR × Filter Time den 2 3 4 5 7 1 1 2 0 6 0 6 0 1 1 1 1 1 1 1 1 1 1 1 1 1	 lec, 6, week, 38 mand 8.3 10.3 19.0 16.0 15.6 19.8 s, 2 total columns 			Q	R · G Data O d m Values a b e i p q	Global Environment • Global Environment • 6 obs. of 2 num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 0	<pre>variab1 :2] 1 2 4 40 56 72 21 22 23 2 .0778 0.23 5 7 9 11</pre>
audo fdt Cc (cu) lec, 6, w 1 1 2 3 4 5 6 howing 1 t R R 4.1.	reek_3AR × Filter Time det 2 3 4 5 7 to 6 of 6 entrie 2 · C:/dab/	 lec, 6, week, 38 mand 8.3 10.3 19.0 16.0 15.6 19.8 s, 2 total columns 				R · C Data O d m Values a b e i p q	Global Environment • Global Environment • num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 3	<pre></pre>
20udio 6dt Co 2) lec.6,w 2) lec.6,w 2) lec.6,w 2 3 4 5 6 howing 1 t R R4.1. View((veek_JAR × Filter Time det 2 3 4 5 7 to 6 of 6 entrie 2 · C/dab/ d)	lec. 6, week 38 land			Q	 R • Data d d m Values a b e i p q Plots 	Global Environment • Global Environment • 6 obs. of 2 num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 0 num [1:5] 1	<pre>variabl i2] 1 2 4 40 56 72 21 22 23 2 .0778 0.23 5 7 9 11 3 23 33 43</pre>
Studio fdt Cc fdt Cc	reek_3AR × Filter Fine dec T 2 3 4 5 7 1 10 6 6 of 6 entrie 2 · C/dab/ d) vert to s	lec.6,week.38 mand a			Q	 R • Data d d m Values a b e i p q Plots 	Global Environment • Global Environment • num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 3	<pre>variabl i2] 1 2 4 40 56 72 21 22 23 2 .0778 0.23 5 7 9 11 3 23 33 43</pre>
9udo fdt Cr Plec.6.w 1 1 1 2 3 4 5 6 howing 1t R R4.1. View(() R R4.1. View() m <	veek_3AR × Filter Time dec 2 3 4 5 7 to 6 of 6 entrie 2 · C:/dab/ d) vert to a data.mate	lec.6,week.38 mand a			Q	 R • Data d d m Values a b e i p q Plots 	Global Environment • Global Environment • 6 obs. of 2 num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 0 num [1:5] 1	<pre>variabl i2] 1 2 4 40 56 72 21 22 23 2 .0778 0.23 5 7 9 11 3 23 33 43</pre>
Studio (dat Co (dat	reek_3AR × Filter Fine der 1 2 3 4 5 7 to 6 of 6 entrie 2 · C/dab/ d) vert to m data.matm	lec.6,week.38 mand a			Q	 R • Data d d m Values a b e i p q Plots 	Global Environment • Global Environment • 6 obs. of 2 num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 0 num [1:5] 1	<pre>variabl i2] 1 2 4 40 56 72 21 22 23 2 .0778 0.23 5 7 9 11 3 23 33 43</pre>
Studio Edit Co Studio Edit Co Image: Comparison of the comparis	reek_3AR × Filter Fine der 1 2 3 4 5 7 to 6 of 6 entrie 2 · C/dab/ d) vert to m data.matm	lec.6,week.38 mand a			Q	 R • Data d d m Values a b e i p q Plots 	Global Environment • Global Environment • 6 obs. of 2 num [1:6, 1 num [1:5] 2 860 int [1:10] num [1:5] 0 num [1:5] 1	<pre>variabl i2] 1 2 4 40 56 72 21 22 23 2 .0778 0.23 5 7 9 11 3 23 33 43</pre>

class(m)

So, I can check, I can go back and actually use that class option, class function to check the class of m now. So, let me check that and it is saying it is array is not a data frame. So, I have converted that data frame d, which is actually the BOD data into a matrix m. Now, this matrix has this header, column names. Suppose, you want to get rid of these column names. So, how can I do that?

(Refer Slide Time: 26:00)

lec_6	week_3A.R × • lec_6_week_38.R* × • lec_6_week_3C.R ×	Environment	-
	🔊 🔒 🖸 Source on Save 🔍 🎢 🗧 👘 Run 😁 📑 Source 🔹	💣 🔒 😰	• 🚺 180 MiB • 🛛 🎻 🛛 🗐 •
64	# Check class	R 🔹 🚺 Glob	oal Environment • Q
65	class(m)	Data	
66		0 d	6 obs. of 2 variabl
67	# remove column names	m	num [1:6, 1:2] 1 2
68 69	colnames(m) <- NULL	Values	non (arr) arej a e m
70	cornanes (iii) <- noce	a	num [1:5] 24 40 56 72
71	print(m)	b	860
72		e	int [1:10] 21 22 23 2
73 74		i	num [1:5] 0.0778 0.2
74	(4) Convert data frame to matrix \$ R Script \$	р	num [1:5] 3 5 7 9 11
		q	num [1:5] 13 23 33 43
•	1.2 · C/dab/ 🖗 🦪 🗖		
	ames(m) <- NULL	Plots	-
prin	1 [.2]	(0 10 P	Zoom 🛛 🚈 Export 🔹 🔯 🛛 🎻
I,] ¹	1 8.3 T		
2,]	2 10.3		
3,]	3 19.0		
4,] 5,]	4 16.0 5 15.6		
5.]	7 19.8		
	1 2010		



$columns(m) \leftarrow NULL$

print(m)

I can do this way, very simple. Column names of m, this is what I am writing here. That is what I am telling to R, column names of m, m is the argument to this function m. And I am assigning NULL, NULL mean nothing, not zero, NULL mean nothing. I am assigning NULL to that. So, column names, column names of m, when I am writing m in the argument, column names function will extract the names of the header columns of that matrix.

And then it as I am assigning NULL to them, all of them will become NULL. So, let me execute that and I will show you the effect. So, here I have executed. Now, let me, I can print

it. So, now, you see you do not have any header, only the column numbers are there or you can go to environment and you can see there is only v1, v2, because variable 1 variable 2 that is how it is saying, it does not have any columns, column names.

So, what I have done, I have learned till now, how to create matrices. Now, I should go for matrix operations, like addition, multiplication, but before that, I will go into what we have done for vector also, indexing issue. How to take a particular element of a matrix or a particular column or particular row of a matrix?

(Refer Slide Time: 27:25)

Robudio File Edit Code View Plots Session Build Debug Profile Tools Help		- 0 :
lec.6.week.3AR × lec.6.week.3B.R* × lec.6.week.3CR ×	Environment	-0
🗢 🗐 🔒 🕞 Source on Save 🔍 🎢 - 📄 🔿 Run 😁 📑 Source 🔹	· .	• 🕽 180 MiB • 🕖 🔳 • 🕲
77 - # Matrix indexing *	R = 📫 Glob	pal Environment • Q
78	Data	
<pre>79 # matrix[row number, column number] 80</pre>	0 d	6 obs. of 2 variabl
<pre>81 # Print the value or element in row = 3, column = 2</pre>	m	num [1:6, 1:2] 1 2
82	Values	
83 print(m[3, 2])	a	num [1:5] 24 40 56 72
84 85 # print the first column	b	860
83:15 Matrix indexing : R Script :	е	int [1:10] 21 22 23 2
	i	num [1:5] 0.0778 0.23
🥷 R 4.1.2 · C/dab/ 🖗 🥑 👘	р	num [1:5] 3 5 7 9 11
> print(m)	q	num [1:5] 13 23 33 43_
[,1] $[,2][1,]$ 1 8.3	Plots	-0
[2,] 2 10.3 [3,] 3 19.0	4.10 1 19	Zoom 🛛 🚈 Export 🔹 🔯 🖉
[3,] 3 19.0 [4,] 4 16.0		
[5,] 5 15.6		
[6,] 7 19.8		
> View(m)		
> print(m[3, 2])		
[1] 19 [2		



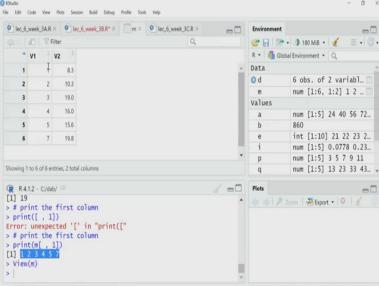
print(m[, 1])

So, if you look into matrix indexing, the system is like this, you have should have a matrix name. And then in the square bracket, you should have column number and row numbers. The first number is row number. And the second number after comma is the column number. This is similar to vector in vector, we are using vector name in square bracket, I have the index of the element number position. Here, in these cases, a 2-dimensional matrix.

So, that means I have two things I would say, mentioned, I have to mention the row number as well as the column number. So, the first number in that square bracket should be the row number, then comma and then the column number. So, now suppose I have already created this m matrix from BOD data set, and I want to take the data or the element of row 3, column 2. So, let me check, row 3 is this one, and column 2, column 2 is this column. So, row 3, column 2 should be 19. So, let us do it. I want to print it. So, what I am doing, in the print I am giving argument m square bracket 3, 2. That means 3 comma 2. That means 3 is the row number, 2 is the column number of whom, the m matrix and I want to print that. So, if I print. Yes, I get 19. Now, suppose I used to use the same indexing technique to not just get one value, but a whole column or whole row, I can also do that very easily.

(Refer Slide Time: 29:03)

lec_6_week_3A.R × Olec_6_week_3B.R* × Olec_6_week_3C.R ×	Environment	-
OI # Print the value or element in row = 5, column = 2		• 🚺 180 MiB • 🖌 🖉 • 🕅
82		pal Environment • Q
83 print(m[3, 2])	Data	
84 85 # print the first column	0 d	6 obs. of 2 variabl
86 print(m[, 1])	m	num [1:6, 1:2] 1 2
87	Values	
88	a	num [1:5] 24 40 56 72.
89 • # **********************************	b	860
91 # Transpose of a matrix 📖	e	int [1:10] 21 22 23 2.
99, # **********************************	i	num [1:5] 0.0778 0.23.
101:# Matrix addition 🔤	р	num [1:5] 3 5 7 9 11
B6:15 Matrix indexing c R Script c	q	num [1:5] 13 23 33 43.
R 4.1.2 · C/dab/ 🖗 🖉 🗖	Plots	-
print(m[3, 2])		Zoom 📲 Export • 0 🥖
1] 19		
# print the first column		
<pre>print([, 1]) rror: unexpected '[' in "print(["</pre>		
# print the first column		
print(m[, 1])		
1 1 2 3 4 5 7		

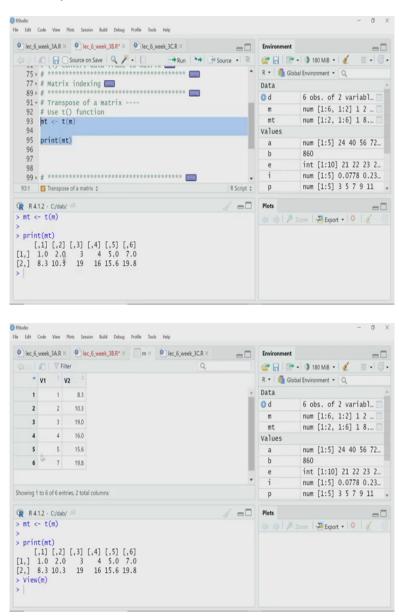


What I will do? So, suppose I want the first column, first column means I want these 1 2 3 4 5 up to 7. And I have to discard the second column. So, what I can say okay, I want the first column of m. So, m followed by the square bracket. And I do not put any more numbers because I know all, I want all the rows. So, I do not specify any row numbers. So, R will understand that I am asking it to give me all the rows data, and then after the comma, I am specifying the column number, the column number is 1.

And I want to print that. So, let me print it. Somewhere, I have missed the m while correcting, so it has to be m. So, print m in the square bracket, no row number and then the comma and the column number. Now, it should work. I have got the first column 1 2 3 4 5 6 7, let us check in m. First column is 1 2 3 4 5 6 7. So, indexing is done. Now, we will go into

the next operation called transposing a matrix. If you remember, we can transpose. We can convert row to column, column to row, it can be done very easily in R.

(Refer Slide Time: 30:26)



$mt \leftarrow t(m)$

print(mt)

To do that, we have a function called t. And if you call that function and give the matrix as an argument, in this case, I will give m as an argument, I should be able to create the transpose of that. What I am doing in this line is that I am transposing m and creating and assigning that matrix to a new matrix called mt and I want to print it. So, let us do that. Now, you can see, the original m matrix is, how many rows we have, let us open, we have 6 rows and 2

columns. Now here, in case of mt, which is transpose, it has 2 rows and 6 columns. So, we have transposed m. Now, I will do matrix addition.

(Refer Slide Time: 31:14)

RStudio File Edit	Code View Plots Session Build Debug Profile Tools Help			- a
0 lec_6	week_3A.R × @_lec_6_week_3B.R* × @_lec_6_week_3C.R ×	-0	Environment	-
	a Gource on Save 🔍 🎽 - 📄 🛶 Run 😁 Source	- E	🕋 🔒 i 😁	• 🚺 180 MiB • 🛛 🎻 🛛 🗐 • 🖓
99 >	# Matrix addition	^	R 🔹 🎒 Glob	pal Environment * Q
101 .	· · · Matrix addition		Data	
103	# Create two matrices		0 d	6 obs. of 2 variabl
104	<pre>n <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE)</pre>		m	int [1:2, 1:2] 1 3
105			mt	num [1:2, 1:6] 1 8
106	$n \leftarrow matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALS$	SE)	n	int [1:2, 1:2] 11 1
107	A add two materians		Values	ine gare, and an an
108 109	<pre># Add two matrices sum.mn <- m + n</pre>		a	num [1:5] 24 40 56 72
110	50m, mit <~ m + 11		b	860
111	print(sum.mn)		e	int [1:10] 21 22 23 2
112		٣	-	num [1:5] 0.0778 0.23
104:1	Matrix addition	R Script \$	1	num [1:5] 0.0778 0.25
R R 4	.1.2 · C:/dab/ 🗇 🚽	-0	Plots	-0
> prin			6 6 9	Zoom 📲 Export 🔹 🔯 🧃 🥖
	,1] [,2] [,3] [,4] [,5] [,6]			
	1.0 2.0 3 4 5.0 7.0 8.3 10.3 19 16 15.6 19.8			
> View				
	matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE)			
>	and a set of the set o			
> n <-	<pre>matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)</pre>	- 1		
>		*		

	m × • lec_6_week_3C.R ×	Environme	ent 🖂
🗘 📄 🖉 Filter	Q,	er 🔒 🗌	🐨 📲 🔹 🚺 180 MiB 🔹 🖌 🍯 📲 🗐 🔘
* V1 V2 *		R + 💼	Global Environment • Q
1 1 2		Data	
2 3 4		O d	6 obs. of 2 variabl
		m	int [1:2, 1:2] 1 3
		mt	num [1:2, 1:6] 1 8
		n	int [1:2, 1:2] 11 1
		Values	
		a	num [1:5] 24 40 56 72.
		b	860
		e	int [1:10] 21 22 23 2.
howing 1 to 2 of 2 entries, 2 total columns		i	num [1:5] 0.0778 0.23.
R 4.1.2 · C:/dab/ 🔅		- Plots	-
<pre>[,1] [,2] [,3] [,4] [,5] [,6] [1,] 1.0 2.0 3 4 5.0 7.0 2,] 8.3 10.3 19 16 15.6 19.8 view(m) m <- matrix(c(1:4), nrow = 2, no n <- matrix(c(11:14), nrow = 2, View(m)</pre>	ol = 2, byrow = TRUE)	• (or (o))	🎢 Zoom 🖉 Export • 0 🧹

2 12 14 m int [1:2, 1:2] 1 3 mt num [1:2, 1:6] 1 8 n int [1:2, 1:2] 11 1 values a num [1:5] 24 40 56 72 b 860	e lec_6	week_3A.R	× e	lec_6	week_3B.R* × m × n ×	<pre>ec_6_week_3C >> </pre>	Environment	-1	
1 11 13 2 12 14 O d 6 obs. of 2 variabl m int [1:2, 1:2] 1 3 m int [1:2, 1:2] 1 3 m int [1:2, 1:2] 1 3 m int [1:2, 1:2] 1 1 with one of the state of the		0 8	Filter			Q,	👕 🔒 🖙 ·	📲 🗣 🚺 🖬 🖬 🖌 🖉 👘	
2 12 14 a 11 13 m int [1:2, 1:2] 1 3 m int [1:2, 1:2] 1 3 m int [1:2, 1:2] 1 3 m int [1:2, 1:2] 1 1 m int [1:2, 1:2] 1 1 wint [1:2, 1:2] 1 1 11.1 wint [1:2, 1:2] 1 1 11.1 values a a num [1:5] 24 40 56 72 b 860 e int [1:10] 21 22 23 2 i num [1:5] 0.0778 0.23. R R.4.12 · C/dab/ [1,1] 1.0 2.0 3 4 5.0 7.0 2 [2,1] 8.3 10.3 19 16 15.6 19.8 > view(m) 2 > m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE)	*	V1	V2				R 🔹 🎒 Global Environment 🔹 🔍		
2 12 14 m int [1:2, 1:2] 13 m int [1:2, 1:2] 13 mt num [1:2, 1:2] 13 n int [1:2, 1:2] 11 values a num [1:5] 24 40 56 72 b 860 e int [1:10] 21 22 23 2 i num [1:5] 0.0778 0.23. R R412 · C/dab/ (1,1 1.0 2.0 3 4 5.0 7.0 [2,] 8.3 10.3 19 16 15.6 19.8 view(m) r <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE) r <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)	1	11		13			Data		
m int [1:2, 1:2] 1 3 mt num [1:2, 1:2] 1 3 mt num [1:2, 1:2] 1 1 n int [1:2, 1:2] 1 1 values a num [1:2, 1:2] 1 1 a num [1:5] 24 40 56 72 b 860 e int [1:5] 24 40 56 72 b 860 e int [1:5] 0.0778 0.23 i num [1:5] 0.0778 0.23 R R 4.12 · C/dab/ /* Image: Com <	2	12		14			0 d	6 obs. of 2 variabl	
n int [1:2, 1:2] 11 1 Values a num [1:5] 24 40 56 72 b b 860 e int [1:10] 21 22 23 2 i num [1:5] 0.0778 0.23 R R412 · C/dab/ (1, 1 0. 2.0 3 4 5.0 7.0 (2, 1 8.3 10.3 19 16 15.6 19.8 view(m) m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE)							m	int [1:2, 1:2] 1 3	
Values a num [1:5] 24 40 56 72 b 860 e int [1:10] 21 22 23 2 i num [1:5] 0.0778 0.23 R R412 - C/dab/ @ 1,1 1.0 2.0 3 4 5.0 7.0 2,1 8.3 10.3 19 16 15.6 19.8 view(m) m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE)							mt	num [1:2, 1:6] 1 8	
a num [1:5] 24 40 56 72 b 860 e int [1:10] 21 22 23 2 i num [1] 1.0 2.0 3 4 5.0 7.0 [2] 3.3 10.3 19 16 15.6 19.8 View(m) m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE)								int [1:2, 1:2] 11 1	
b 860 e int [1:10] 21 22 23 2. i num [1:5] 0.0778 0.23. R R4.12 · C/dab/ ☆ I,] 1.0 2.0 3 4 5.0 7.0 2,] 8.3 10.3 19 16 15.6 19.8 View(m) m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE) n <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)							Values		
moving 1 to 2 of 2 entries, 2 total columns e int [1:10] 21 22 23 2						a			
howing 1 to 2 of 2 entries, 2 total columns R R412 · C/dab/ ← 1,] 1.0 2.0 3 4 5.0 7.0 2,] 8.3 10.3 19 16 15.6 19.8 · View(m) m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE) · n <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)						b	860		
R R.4.12 · C/dab/ ⇒ Image: Conject of the conject							-	int [1:10] 21 22 23 2	
1.] 1.0 2.0 3 4 5.0 7.0 2.] 8.3 10.3 19 16 15.6 19.8 ∨iew(m) m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE) • n <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)	howing 1	to 2 of 2 e	ntries, 2	total co	olumns		i	num [1:5] 0.0778 0.23	
1.] 1.0 2.0 3 4 5.0 7.0 2.] 8.3 10.3 19 16 15.6 19.8 View(m) m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE) • n <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)	R R 4	1.2 · C:/di	b/ 🔅			/ -0	Plots	-	
2,] 8.5 10.5 19 16 15.6 19.8 View(m) m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE) n <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)				3	4 5.0 7.0	*			
<pre>m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE) n <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)</pre>			3 1	.9	16 15.6 19.8			coom i A popur - i a i i a	
n <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)			1.12.	0		TRUE			
<pre>n <- matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)</pre>		matri	(((1:	4), 1	nrow = 2, $ncol = 2$, byrow	V = TRUE)			
		matrix	(c(11	:14)	. nrow = 2, ncol = 2, byr	ow = FALSE)			
> View(n)									

 $m \leftarrow matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE)$

 $n \leftarrow matrix(d(11:14), nrow = 2, ncol = 2, byrow = FALSE)$

 $\text{sum.mn} \leftarrow m + n$

print(sum.mn)

Matrix addition, to do that, I will create two new matrices with a small r so that we can easily perform some other analysis easily. So, I am creating a matrix 2 by 2 matrix, the m. So, what I am doing it, I am using the matrix function, that means I have to give a vector. So, the first argument here I am creating a vector using c function, and that vector will have values from 1 to 4. And I want to divide this vector into 2 by 2 matrix, and I want to make it row wise.

Whereas here, I want to create a new matrix n for which I am giving a vector which starts from 11 and ends at 14 and it is also 2 by 2 and I want by row equal to FALSE, just like that I have given, you could, I could have taken something else as an example. So, let me create these two matrices. I have created them, you can check here m and n, 2 by 2 matrices.

(Refer Slide Time: 32:21)

	week_3AR × 9 lec_6_week_3B.R* × 9 lec_6_week_3C.R ×	Environment	- 🕽 180 MiB - 🖌 🚍 - 🛛		
TAN	Create two matrices				
104	<pre>m <- matrix(c(1:4), nrow = 2, ncol = 2, byrow = TRUE)</pre>		R 🔹 🚮 Global Environment 🔹 🔍		
105 106	$n \leftarrow matrix(c(11:14), nrow = 2, ncol = 2, byrow = FALSE)$		Data		
100	$n \leftarrow matrix(c(11.14), nrow = 2, ncor = 2, byrow = ralse)$	0 d	6 obs. of 2 variabl		
108	# Add two matrices	m	int [1:2, 1:2] 1 3		
109	sum.mn <- m + n	mt	num [1:2, 1:6] 1 8		
110		n	int [1:2, 1:2] 11 1		
111	print(sum.mn)	sum.mn	int [1:2, 1:2] 12 1		
112		Values			
113		a	num [1:5] 24 40 56 72.		
114 115		b	860		
109:1	Matrix addition t R Script t	e	int [1:10] 21 22 23 2.		
R R 4.	1.2 · C/dab/ ∞	Plots	-		
View	(m) .		Zoom 🛛 🛺 Export 🔹 🔯 🖌 🧹		
View	(n)	A HIV	cooline i Manager and a la Manager a		
sum.	mn <- m + n				
	*(
	t(sum.mn) ,1] [,2]				
	42 15 15 15				

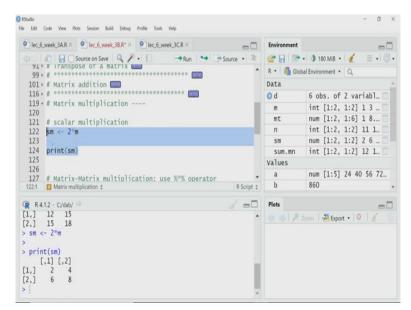


print(sum.mn)

Now, I will do all those vector operations on these two. So, I want to add, the first thing is that I want to add these two matrices, I will add m and n. So, just like addition of two vectors, I will use the plus operator and I should be able to add them. So, here what I am doing, I am adding m plus n, I am adding m and n matrices that those I made just now.

And I am assigning to a variable called sum dot mn. Obviously, that should also be a metrics. And I am printing it. So, let me do that, I have done it. So, I have added m and n. And I have got a new metric sum dot mn, which has these values 12 15 15 18. Now, as I have done the summation, obviously, you can easily do scalar addition to a matrix, I have not shown that, there is not an issue. Now, I will go and show you matrix multiplication.

(Refer Slide Time: 33:15)



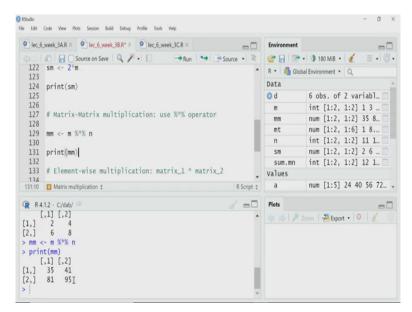
sm← 2*m

print(sm)

Just like vector multiplication matrix multiplication can be also of two types, at least two types. One is the scalar another one is a multiplication between two matrices that we have discussed in the lectures. So, first, I will multiply the m matrix that I created by 2, a scalar and then I will print it. So, I have multiplied it by 2 and then assign that to sm and I have printed sm here. And now I have got a new matrix 2 4 6 8.

Now, I want to do the complicated one, the matrix-matrix multiplication, two matrix has to multiply. And there is a rule to do that, you have to take the row and the column of the second one and do the multiplication. If you have forgotten, please go back and check the lecture on matrix multiplication to understand what I am saying, and then come back and look into this part of this video.

(Refer Slide Time: 34:07)



$mn \leftarrow m\%\%n$

print(mn)

So, if you want to perform matrix-matrix multiplication, the way we have taught in that lecture, then you have to use this particular operator, you first put the percentage sign, then the star and then the percentage sign. So, percentage star percentage, this is the operator for matrix-matrix multiplication. And what I want to do here, I want to do matrix-matrix multiplication between m and n, m and n we had created, and I assign that to mn, a new matrix.

And that is why I am using here that particular operator percentage star percentage, data here I execute it, and then I print it. So, I got 35 41 81 95. You can go back and check the calculation using the m and n matrix you will find you will get the same matrix. So, by multiplying these two matrices, I have got another matrix. Now, what will happen if you use the star symbol which is the multiplication operator in R, you can do that, but that will give a different result. What it will perform? It will perform element by element multiplication. What do I mean by that?

(Refer Slide Time: 35:29)

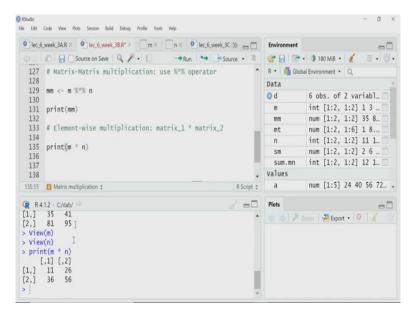
lec_6	week_3	AR ×	e lec	_6_week_38.R* ×	m×	e lec_6_w	ek_3C.R ×		Environment			-
	21	7 Filter					٩,		@ B .	• 🚺 180 MiB •	1	■ • (
	V1	V2							R • Glob	al Environment •	Q,	
1		1	2						Data			
2		3	4						0 d	6 obs. of	2 varia	ab1 🛛
		2	-						m	int [1:2,	1:2] 1	3
									mm	num [1:2,	1:2] 3	5 8
									mt	num [1:2,	1:6] 1	8
									n .6	int [1:2,		
									SM	num [1:2,		
									sum.mn	int [1:2,	1:2] 1	2 1
									Values			
howing 1	to 2 of	2 entrie	s, 2 tota	columns					a	num [1:5]	24 40	56 72.
R R4	1.2 · C	/dab/						1 -0	Plots			-
prin [1,]	81							, ,	(a 1) P 1	toom 🔏 Expor	- 0	1

lec_6_	week_3	A.R	×	lec	_6_week_38.R* ×	m× n	× O	lec_6_w	eek_3C >> e		Environmer	t				-1
	21	7F	ilter					Q,			e	- 0	180 MiB •	1	=	. 6
*	V1		V2								R - 💼 (lobal Envir	onment •	Q,		
1		11		13							Data					
2		12		14	9.						0 d	6 0	bs. of	2 var	iab1	
		16		17							m	int	[1:2,	1:2]	13	
											mm	num	[1:2,	1:2]	35 8	
											mt	num	[1:2,	1:6]	1 8.	
											n	int	[1:2,	1:2]	11 1	
											sm	num	[1:2,	1:2]	26	
											sum.mn	int	[1:2,	1:2]	12 1	
											Values					
howing 1	to 2 of	12 et	ntries, 2	tota	columns						a	num	[1:5]	24 40	56	72.
R R 4.	1.2 • 0	:/da	b/ 🗇						1	-0	Plots					_
2,]	6												-Z Expo	rt - 0	111	
	t(mm) ,1] 35 81 (m))]							ļ						

Let me open, m and n. This is m matrix; this is n matrix. So, if I use m star n then element by element multiplication will happen. That means, this 1 of m will get multiplied with 11 and that will remain in that position that value, 2 will get multiplied with 13. So, index numbers of these two matrices are matching, when I am taking the first row first column value and multiplying it with the first row first column value of the second matrix.

The second row second column value of the first matrix will be multiplied with second row second column value of the second matrix. So, this will eliminate wise I will multiply, I will show you how you, what will happen if I do multiplication like that.

(Refer Slide Time: 36:18)



print(m * n)

So, here I am asking to print just using the star symbol, multiplication symbol. So, there should be element wise multiplication between m and n. Here, you can see 11 and this one is 26, it is this one is 11 because 1 into 11 is 11, then this one is 13. And the corresponding value here in the m is 2, 13 into 2 is 26.

So, if you want to do element wise multiplication, then only you use the star symbol, usually in the matrix multiplication requirement that we have in this data analysis course, we do not need that we require the original multiplication that we have learned in the linear algebra class. In that case, remember, you have to use these percentage star percentage operator to tell R that you want matrix-matrix multiplication, not element wise multiplication between the matrices.

(Refer Slide Time: 37:18)

lec_6_week_3A.R × 0 lec_6_week_3B.R* × m× n× 0 lec_6_week_3C >>	Environment	-
Source on Save Q 🖉 - 📄 👄 Run 😁 🖙 Source - ≷	e	- 🕒 180 MiB - 🖌 📃 - 🔇
110 • #	R • Glo	bal Environment • Q
119 + # Matrix multiplication 📼	Data	
140 - # Matrix determinant	0 d	6 obs. of 2 variabl
141		
142 # Use det() function	m	int [1:2, 1:2] 1 3
143	mm	num [1:2, 1:2] 35 8
144 print(det(m))	mt	num [1:2, 1:6] 1 8
145	n	int [1:2, 1:2] 11 1
146	sm	num [1:2, 1:2] 2 6
147	sum.mn	int [1:2, 1:2] 12 1
148 - # **********************************	Values	
1449 * # Eldenvalues and elden vectors and elden	a	num [1:5] 24 40 56 72
R R 4.1.2 · C/dab/ 🖗 🚽 🚽	Plots	_
View(m)		Zoom - Zeport - 0 /
View(n)		Zoom 1 2 Export - 1 - 11 2
print(m * n)		
[,1] [,2]		
1,] 11 26		
2,] 36 56 print(det(m))		
princ(dec(m))		

print(det(m))

So, we have learned the matrix multiplication. Now, I will go to determinant and eigen values. How can I calculate the determinant of a matrix? In R it is very easy, it has a determinant function, det function, which will take the matrix as an argument and it will output the determinant of that. So, I am printing it here in one line. So, I am saying print date of m and if m is the matrix we already know. So, if I do that, I get minus 2. So, that is the determinant of m matrix, that is the this one, this matrix, you can manually do and check that.

(Refer Slide Time: 37:57)

lec_6_week_3A.R × Olec_6_week_3B.R* × m × n × Olec_6_week	k3C » =	Environment	_
💷 🖉 📄 Source on Save 🔍 🎢 🛛 📑 Run 😁 🖪	• Source • 🔍	🚰 🔒 i 👺	• 🌗 180 MiB • 🗹 🔳 •
147	*	R 🔹 💼 Gloi	pal Environment 🝷 🔍
148 - #		a	num [1:5] 24 40 56 72
149 * # Eigenvalues and eigen vectors 150		b	860
151 # Use eigen() function that returns both		e	int [1:10] 21 22 23 2
152 ev <- eigen(m)		i	num [1:5] 0.0778 0.23
153		р	num [1:5] 3 5 7 9 11
154 # Get the eigenvalues 155 values <- evSvalues		q	num [1:5] 13 23 33 43
155 values <- evovalues		theta	0.130803610419603
157 print(values)		u	num [1:5] 1 3 5 7 9
158	¥	V	num [1:5] 12 20 28 36
57:14 🚺 Eigenvalues and eigen vectors 🗧	R Script 🗘	values	num [1:2] 5.372 -0.37
R 4.1.2 · C:/dab/ 🖗	1 -0	Plots	_
print(det(m))		000	Zoom 📲 Export 🔹 🔯 🛛 🥑
1] -2			
<pre># Use eigen() function that returns both ev <- eigen(m)</pre>			
# Get the eigenvalues			
values <- ev\$values			
print(values)			

 $ev \leftarrow eigen(m)$

values $\leftarrow ev$ \$values

print(values)

Now, I have done the determinant. Now, I will go to what we call eigen values and eigen vector, two very important quantities related to square matrices. So, luckily, R has an inbuilt function to calculate both eigen values and eigen vectors. That function is called eigen. It will take the matrix as an argument and it will return both eigen values and eigen vectors. Let me first ask it to calculate the eigen values and eigen vector and I will assign that whole data to this variable ev.

This ev will store, this variable will store both eigen values and eigen vector. Subsequently, I will separately collect the eigen vector and eigen value and print them. So, the first thing is I run the eigen function. So, it has calculated both the eigen values and eigen vectors for m and has assigned the whole data to ev. Now, I want to collect the eigen values only.

So, I am writing ev dollar sign eigen values. So, value is an object of the output comes from the eigen function, and that values object has all the eigen values in it. And I want to fetch that from ev. So, I am using this dollar symbol. So, I am saying give me the values that you are storing in value as values object in ev and I will assign those thing to values variables, those are the eigen values.

So, I do that and then I print them and the eigen values are 5.37 and minus 0.37 and so on. So, now, I can extract the eigen vectors in the same way. So, the output of an eigen function has an object called vectors. And I have stored the data in ev. So, I want ev to give me the vectors object.

(Refer Slide Time: 39:58)

	week_3AR × • lec_6_week_3B.R* × m × n × • lec_6_week_3C. >> =	Environment	-
111	🔊 🔒 🖸 Source on Save 🔍 🎢 🔹 📑 Run 🐤 📑 Source 🔹 🛎		• 🚺 173 MiB • 🛛 🎻 🔳 • 🖉
154	# Get the eigenvalues	R 🔹 🚺 Glob	oal Environment 🝷 🔍
155	values <- ev\$values	vectors	num [1:2, 1:2] -0.4
156		Values	
157	print(values)	a	num [1:5] 24 40 56 72_
158 159	# Get the eigenvectors	b	860
160	vectors <- evSvectors	e	int [1:10] 21 22 23 2.
161		i	num [1:5] 0.0778 0.23.
162	print(vectors)	p	num [1:5] 3 5 7 9 11
163		q	num [1:5] 13 23 33 43.
164		theta	0,130803610419603
165	• • • • • • • • • • • • • • • • • • •	u	num [1:5] 1 3 5 7 9
62:15	Eigenvalues and eigen vectors \$ R Script \$		** ** ** ** ** **
R R 4	1.2 · C/dab/ 🖈 🥑 🗖	Plots	-
	t(values)	6 6 2	Zoom 📲 Export 🔹 🔯 🛛 🥖
	.3722813 -0.3722813		
	t the eigenvectors ors <- ev\$vectors		
	t(vectors)		
Bi ili	π[,1] [,2]		
1.1 -	0.4159736 -0.8245648		
2,] -	0.9093767 0.5657675		
1			

values $\leftarrow ev$ \$values

print(values)

vectors \leftarrow ev\$vectors

print(vectors)

So, I am using ev, then the dollar sign and then the vectors. And if I run these and I will assign that to this new variable vectors, done that, I want to print the vectors. Now, you can see both of vectors. It is a two dimensional matrix, it is a two dimensional matrix, so it will have two eigen values two eigen vectors, check that it has two eigen values. So, correspondingly there should be two eigen vectors. The first eigen value is 5.37.

So, it is Victor is here, the first column minus 0.41, minus 0.9. So, this is the eigen vector corresponding to the first eigen value. And next one is the second eigen value here minus 0.37. So, the next column in the vectors is the second eigen vector corresponding to the second eigen value. That is all for matrices.

We have learned the how to create vectors how to create matrices and their operations, addition, multiplication, indexing, all these issues. One important thing that we have learned in linear algebra class in this course, is that, we can use linear algebra to solve a system of linear equations, I will end this lecture with that topic. So, let me recapitulate with how we solve a system of linear equation using linear algebra.

(Refer Slide Time: 41:21)

<pre>Plec.6.week.3AR × Plec.6.week.3AR × Plec.6.</pre>		 173 MB • 2 173 MB • 2 al Environment • Q num [1:2, 1:2] -0.4 num [1:5] 24 40 56 72. 860 int [1:10] 21 22 23 2. num [1:5] 0.0778 0.23. num [1:5] 3 5 7 9 11
<pre>1 2 # Solve a homogeneous system of linear equations 3 4 # Solve : x + y = 6; -3x + y = 2 5 6 # writing the system of equation in matrix notation: 7 # A = 1 h 8 # -3 1 9 10 # b = 6 11 # 2</pre>	R • Glob vectors Values a b e i j p	al Environment • Q num [1:2, 1:2] -0.4[num [1:5] 24 40 56 72, 860 int [1:10] 21 22 23 2, num [1:5] 0.0778 0.23.
<pre>2 # solve a homogeneous system of linear equations 3 4 # Solve : x + y = 6; -3x + y = 2 5 6 # Writing the system of equation in matrix notation: 7 # A = 1 # 9 10 # b = 6 11 # 2</pre>	vectors Values a b e i i p	num [1:2, 1:2] -0.4 num [1:5] 24 40 56 72. 860 int [1:10] 21 22 23 2. num [1:5] 0.0778 0.23.
3 4 # Solve : $x + y = 6$; $-3x + y = 2$ 5 6 # Writing the system of equation in matrix notation: 7 # A = 1 $\frac{1}{11}$ 9 10 # b = 6 11 # 2	Values a b e i p	num [1:5] 24 40 56 72. 860 int [1:10] 21 22 23 2. num [1:5] 0.0778 0.23.
5 6 # Writing the system of equation in matrix notation: 7 # A = 1 $\frac{1}{12}$ 8 # -3 1 9 10 # b = 6 11 # 2	a b e i p	860 int [1:10] 21 22 23 2 num [1:5] 0.0778 0.23
6 # Writing the system of equation in matrix notation: 7 # A = 1 $\frac{1}{12}$ 8 # -3 1 9 10 # b = 6 11 # 2	b e i p	860 int [1:10] 21 22 23 2 num [1:5] 0.0778 0.23
7 # $A = 1$ 8 # -3 1 9 10 # b = 6 11 # 2	e i p	int [1:10] 21 22 23 2 num [1:5] 0.0778 0.23
8 # -3 1 9 10 # b = 6 11 # 2	i p	num [1:5] 0.0778 0.23
10 # b = 6 11 # 2	р	
11 # 2		
		num [1:5] 13 23 33 43.
14	theta	0.130803610419603
12 (u	num [1:5] 1 3 5 7 9
7:10 (Top Level) c R Script c	u	11.11 [1.1] 1 3 3 7 9
R 4.1.2 · C:/dab/ 🖗 / 🗖	Plots	_
print(values)		com 🛺 Export • 0 🥖
1] 5.3722813 -0.3722813		
[,1] [,2]		
1,] -0.4159736 -0.8245648		
2,] -0.9093767 0.5657675		
×		
udio		- 0
Edit Code View Plots Session Build Debug Profile Tools Help		
lec_6_week_3AR × • lec_6_week_3BR* × • lec_6_week_3CR ×		Environment
	Source + 2	🚰 🔒 🜁 • 🌖 173 MiB •
	winner	
12	*	R • 💼 Global Environment •
12 13 # X = x		
12 13 # X = x 14 # y	A	R • di Global Environment • vecto num [1:2,] Values
12 # X = x 13 # X = x 14 # y 15 16 # So, AX = b		vecto… num [1:2, …] Values
Le X = x 14 # Y 15 # So, AX = b 17	Ĺ	vecto… num [1:2, …] Values
12 13 # X = x 14 # y 15 16 # So, AX = b 17 18 # The solution to this system of equation is X = inverse_of_A	Ĺ	vecto num [1:2, [Values a num [1:5] 24
12 13 # X = x 14 # y 15 16 # So, AX = b 17 18 # The solution to this system of equation is X = inverse_of_A 19	Ĺ	vecto num [1:2, Values a num [1:5] 24 b 860
12 3 # X = x 14 # y 15 16 # So, AX = b 17 18 # The solution to this system of equation is X = inverse_of_A 19 20 # Solve() function performs this work in R	Ĺ	vecto… num [1:2, …] Values a num [1:5] 24… b 860 e int [1:10] 2…
12 13 # X = x 14 # y 15 16 # So, AX = b 17 20 # Solve() function to this system of equation is X = inverse_of_A 19 20 # Solve() function performs this work in R 21 # It takes A and b as arguments 22 I	Ĺ	vecto num [1:2, values a num [1:5] 24 b 860 e int [1:10] 2 i num [1:5] 0
12 # X = x 14 # y 15 16 # So, AX = b 17 18 # The solution to this system of equation is X = inverse_of_A 19 20 # Solve() function performs this work in R 21 # It takes A and b as arguments 22 I I Z2 I I	Ĺ	vectonum [1:2, Values a num [1:5] 24 b 860 e int [1:10] 2 i num [1:5] 0 p num [1:5] 3
12 13 # X = x 14 # y 15 16 # So, AX = b 17 18 # The solution to this system of equation is X = inverse_of_A 19 20 # Solve() function performs this work in R 21 # It takes A and b as arguments 22 I	Ĺ	vecto num [1:2,

Suppose, you have told me that I have to solve these two equation x plus y equal to 6, minus 3x plus y equal to 2, I have to solve these two. That means, I have to get the value of x and y that we satisfy both this linear equation. So, how can I do that? I have to create some matrix and vectors. I have to create a coefficient matrix usually we call it A. So, in this case, the coefficient matrix will be 1, 1, minus 3, 1. 1, 1 will be the first row, minus 3, 1 will be the second row.

[1] 5.3722813 -0.3722813 > # Get the eigenvectors > vectors <- evSvectors > print(vectors) [,1] [,2] [1,] -0.4159736 -0.8245648 [2,] -0.9093767 0.5657675

Why? Because I have 1 into x plus 1 into y in this first equation. So, I have 1, 1. In the second equation, I have minus 3 into x plus 1 into y. So, that is why I have minus 3 and 1. So, this is the coefficient matrix. Then everything on the right hand side of the equal to sign should be in another vector. So, I have 6 and 2. So, that will be vector b and I will have should have 6 and 2. And the solution should be another vector that is a capital X here and it should have the unknown x and y.

And I have to calculate this capital X vector. In other words, I have to calculate this small x and small y. How do we do that? We know this whole system of equation can be represented as A into X, A is the coefficient matrix, X is the unknown vector, and b is the right hand side vector. So, if I write A into X equal to b, then I can use linear algebra, and I can get a solution of these, if I do something like that the solution will be X, the vector, result vector will be equal to inverse of A, because I am taking A on the right hand side is equal to sign, inverse of A into b.

So, that means, I have to do inversion of A. Now, this is what you do to solve a system of linear equation by linear algebra. Now, I will not do manually inversion and all these things to solve this problem, I will call a solve function write a function which will solve this problem for me. So, R has that function, it is called solve. So, it should take two argument, A the coefficient matrix and the B vector and then it will spit out this x the solved values, solutions.

(Refer Slide Time: 43:49)

lec_6_week_3A.R × Olec_6_week_3B.R* × Olec_6_week_3C.R ×		Environment	•
🖂 🖉 📄 Source on Save 🔍 🎢 📲	🔿 Run 🐤 🕞 Source 🔹 🗷	🕋 🔒 🗐	🕶 📲 🌒 173 MiB 🔹
19 20 # solut() furthis suffere this much is a	A	R • 🛑 G	lobal Environment •
20 # Solve() function performs this work in R 21 # It takes A and b as arguments		Data	
22 # It takes A and b as arguments		A	num [1:2,
23 - # Create A and b			6 obs. of
24		Oev	
<pre>25 A <- rbind(c(1,1), c(-3,1))</pre>			List of 2 Q
26		Π.	int [1:2,
27 b <- c(6, 2)		mm	num [1:2, 🗌
28		mt	num [1:2, 🗌
29 # solve the system of equations		n	int [1:2, 🗌
36 - # **********************************	*	sm	num [1:2, 🗌
25:28 🔲 Create A and b 🛊	R Script \$	sum.mn	int [1:2,
R 4.1.2 · C:/dab/ 🔗	/ -0	Plots	-
1] 5.3722813 -0.3722813			Zoom Zoom Export
# Get the eigenvectors			200m " Export
vectors <- ev\$vectors			
print(vectors)			
[,1] [,2]			
1,] -0.4159736 -0.8245648 2,] -0.9093767 0.5657675			
A <- rbind(c(1,1), c(-3,1))			
A <- (D)((((1,1), ((-),1)))			

e] lec_6_w	eek_3A.R	× Q	lec_6_week_3B.I	R* X 🕘 lec_6_week_3C.R X	A×		-	Environme	nt — [
4014	0 7	Filter				Q,		er 🔒 🛙	🐡 🔹 🌖 173 MiB 🔹
^ \	/ 1 *	V2						R • 💧	Global Environment •
1	1		1					Data	
2	-3		1					A	num [1:2,
•	-7							0 d	6 obs. of 🕅
								0 ev	List of 2 Q
								m	int [1:2,
								mm	num [1:2,
								mt	num [1:2,
								n	int [1:2,
								sm	num [1:2,
howing 1 to	o 2 of 2 e	ntries, 21	otal columns					sum.mr	n int [1:2, … 🕅
R 4.1.	2 · C:/da	ib/ 🖈					-0	Plots	-
# Get									De Zoom
vector			ctors						
> print		1]	[,2]						
1.1 -0			.8245648						
			.5657675						
A <-) View()		c(1,1), c(-3,1))					
	4.3								

$A \leftarrow rbind(c(1, 1), c(-3, 1))$

b← c(6, 2)

So, let me start with the creation of A and b, A is this matrix 1, 1, minus 3, 1. So, I am using rbind function. First, I am creating the vector 1, 1 here. And then I am creating the second vector minus 3, 1, and I am joining them, I am stacking them row wise, I am binding them row wise using rbind. So, I should get A, I have got A, I can double click and check. I have got 1, 1, minus 3, 1, I will use this A.

(Refer Slide Time: 44:21)

Run Run Source - Run Source - Run Run Source - Run Run </th <th>] lec_6_week_3A.R × 9] lec_6_week_3B.R* × 9] lec_6_week_3C.R ×</th> <th>Enviror</th> <th>ment</th> <th></th> <th>-</th>] lec_6_week_3A.R × 9] lec_6_week_3B.R* × 9] lec_6_week_3C.R ×	Enviror	ment		-
19 20 # Solve() function performs this work in R 1 # It takes A and b as arguments m int [1:2, 22 + # Create A and b mt num [1:2, 23 + # Create A and b mt num [1:2, 24 sm num [1:2, 25 A <- rbind(c(1,1), c(-3,1)) sm num [1:2, 26 y # solve the system of equations 27.13 Create A and b : R R4.12 - C/dab/	💷 💭 🔄 Source on Save 🔍 🎢 🛛 📄 🖼 😁 Source 🔹	2 💣 🖥		🕽 173 Mi	8 - I
20 # Solve() function performs this work in R 21 # It takes A and b as arguments 22 23 * # Create A and b 24 25 A <- rbind(c(1,1), c(-3,1)) 26 27 b <- c(6, 2) 28 R R412 · C/dab/ @ [,1] [,2] 1,] -0.4159736 -0.8245648 2,] -0.9093767 0.5657675 A <- rbind(c(1,1), c(-3,1)) View(A) View(A) View(A)		* R *	Global I	Environmen	nt • 0
<pre>21 # It takes A and b as arguments 22 23 # Create A and b 24 24 25 # Create A and b 24 25 A <- rbind(c(1,1), c(-3,1)) 26 29 # solve the system of equations 29 # solve the system of equations 27.13 Create A and b:</pre>			int	[1:2.	
22 23 - # Create A and b 24 25 A <- rbind(c(1,1), c(-3,1)) 26 27 b <- c(6, 2)] 28 29 - # solve the system of equations ■ 29 - # solve the system of equations ■ 21 create A and b : R Script: R R412 - Cydab/ (,1] [,2] 1, -0.4159736 - 0.82455648 2, 1 -0.9093767 0.5657675 A <- rbind(c(1,1), c(-3,1)) View(A) b <- c(6, 2)					
23 - # Create A and b n int [1:2, 24 24 sm num [1:2, 24 sm num [1:2, 25 A <- rbind(c(1,1), c(-3,1)) sm num [1:2, 26 sm num [1:2, 27 b <- c(6, 2) 28 29. # solve the system of equations Values 35 29. # solve the system of equations Values 35 27.13 21.1					
24 24 25 A <~ rbind(c(1,1), c(-3,1)) 26 27 27 b <~ c(6, 2) 28 29 ≠ # solve the system of equations 25. a < c(6, 2) vecto. num [1:2, 28 28. (2, 1) 27.13 Create A and b: R A12 · C/dab/ 21,] -0.4159786 -0.8245648 21,] -0.9993767 0.5657675 A <~ rbind(c(1,1), c(-3,1)) View(A) b <<< c(6, 2)	23 - # Create A and b				
20 a num [1:2,] 26 27 b c (6, 2) vecto num [1:2,] 28 29 # solve the system of equations a num [1:2,] vecto num [1:2,] 28 27.13 Create A and b: R Script: b num [1:2] 6 2 28 R4.12 · C/dab/ @ Image: Create A and b: R Script: D num [1:2] 6 2 29. # solve the system of equations Image: Create A and b: R Script: D num [1:2] 6 2 28 R4.12 · C/dab/ @ Image: Create A and b: Image: Create A and b: D num [1:2] 6 2 29. # solve the system of equations Image: Create A and b: Image: Create A and b: D D D D D D D D D D	24				
27 b < < c(6, 2) 28 29 ≠ # solve the system of equations 29 ≠ # solve the system of equations a num [1:2], 27.13 Create A and b: (1,1) [,2] 1, -0.4159736 - 0.8245648 2, -0.9093767 0.5657675 A <- rbind(c(1,1), c(-3,1)) View(A) b <- c(6, 2)	25 A <- rbind(c(1,1), c(-3,1))				
28 values 29 # solve the system of equations a num [1:5] 24 27.13 Create A and b : R Script : (1,1) [,2] 1, -0.4159736 - 0.8245648 > 2, 1 -0.9093767 0.5657675 > A <- rbind(c(1,1), c(-3,1)) > View(A) >					
29 # # solve the system of equations a num [1:5] 24 28 # Atl2 · Cydab/ R Script: a num [1:5] 24 28 R 412 · Cydab/ Image: Constant A and b : a num [1:2] 6 2 29 # Joint A and b : Image: Constant A and b : a num [1:2] 6 2 29 # Joint A and b : Image: Constant A and b : a num [1:2] 6 2 20 # Joint A and b : Image: Constant A and b : a num [1:2] 6 2 20 # Joint A and b : Image: Constant A and b : a num [1:2] 6 2 21 - 0.4159736 - 0.8245648 Image: Constant A and b : Image: Constant A and b : Image: Constant A and b : 21 - 0.9093767 - 0.5657675 A <- rbind(c(1,1), c(-3,1)) Image: Constant A and b : Image: Constant A and b : 22 - minimum Constant A and b : Image: Constant A and b : Image: Constant A and b : Image: Constant A and b : 23 - 0.9093767 - 0.5657675 A <- rbind(c(1,1), c(-3,1)) Image: Constant A and b : Image: Constant A and b : 24 - 0.0000 - Image: Constant A and b : 25 - 0.6(6, 2) Image: Constant A and b : Image: Constant A and b : Image:		vec	to num	[1:2,	🔲
2C.ja a Indim [1:3] 24 27.13 Create A and b : R Script : b num [1:2] 6 2R A12 · C/dab/ @ [1,1] [,2] [,1] [,2] 1,] -0.4159786 - 0.8245648 [] [] [] [] 2,] -0.9093767 0.5657675 [] [] [] [] A <- rbind(c(1,1), c(-3,1)) View(A) [] [] [] View(A) b - [] []		Value	5		
R R412 - C/dab/ @ Image: R Script : b num [1:2] 6 2 R R412 - C/dab/ @ Image: R Script : b num [1:2] 6 2 I, I [,2] [,2] Image: R Script : b 1, I -0.4159736 - 0.8245648 Image: R Script : Plots Image: R Script : 2, I -0.9093767 0.5657675 A <- rbind(c(1,1), c(-3,1)) Image: R Script : Image: R Script : Image: R Script : View(A) b <- c(6, 2) Image: R Script : Image: R Script : Image: R Script : Image: R Script :		- a	num	[1:5]	24
[,1] [,2] 1,] -0.4159736 -0.8245648 2,] -0.9093767 0.5657675 A <- rbind(c(1,1), c(-3,1)) View(A) b <- c(6, 2)		ript ‡ b			
[,1] [,2] 1,] -0.4159736 -0.8245648 2,] -0.9093767 0.5657675 A <- rbind(c(1,1), c(-3,1)) View(A) b <- c(6, 2)	R 4.1.2 · C/dab/ 🖗	Plots			-
1,] -0.4159736 -0.8245648 2,] -0.9093767 0.5657675 A <- rbind(c(1,1), c(-3,1)) View(A) b <- c(6, 2)	[,1] [,2]				
A <- rbind(c(1,1), c(-3,1)) View(A) b <- c(6, 2)				m 198	xport •
View(A) b <- c(6, 2)	2,] -0.9093767 0.5657675				
b <- c(6, 2)					
	b <- c(6, 2) b				
	1] 6 2 I				

 $A \leftarrow rbind(c(1, 1), c(-3, 1))$

 $b \leftarrow c(6, 2)$

I want to keep the b vector, b vector should be 6, 2, as I have said here. So, that is what I am doing here, I am using c function to create b. So, b has been created. Let me print and see b, yes, b is 6, 2. So, now I have got a and b. So, now I can call solve function. So, I will do that.

(Refer Slide Time: 44:42)

lec_6_week_3A.R × lec_6_week_3B.R* × lec_6_week_3C.R ×	-0	Environmen	•
🗘 🖉 📄 🖸 Source on Save 🔍 🎢 📲	🖙 Run 🐤 🔿 Source 🔹 🗷	* 8 1	🏞 • 🌖 173 MiB •
26		R • 💼 0	Slobal Environment •
27 b <- c(6, 2)		<	int [1:2,
28			num [1:2,
29 - # solve the system of equations		Values	num [1:2,
30			
31 sol <- solve(A, b)		a	num [1:5] 24
32 33 print(sol)		b	num [1:2] 6 2
33 print(sol) 34		e	int [1:10] 2
35 • # **************************		i	num [1:5] 0
36 - # ****************************	II	р	num [1:5] 3
37 + # Matrix inverse	· ·	q	num [1:5] 13
33:11 🔯 solve the system of equations C	R Script \$	sol	num [1:2] 1 5
R 4.1.2 · C:/dab/	1 =0	Plots	-
A <- rbind(c(1,1), c(-3,1))	*		Zoom
View(A)			and a subort
b <- c(6, 2)			
b 1] 6 2			
sol <- solve(A, b)			
print(sol)			
1] 1 5			

 $solv \leftarrow solve(A. b)$

print(sol)

Here, I am calling solve function of r, I am giving A and b as argument. So, the solve function will solve and spit out the vector x, which has the solution for x and y. And I am assigning that to a new variable called solution, sol. I have done that. Now, I want to print a solution and the solution is 1 and 5. So, x equal to 1, y equal to 5, you can put these two value on those two equation, and you will see the solution will satisfy both this equation.

So, this is the way in one single line, I can solve a system of linear equations. Now, when I was discussing matrices, I had discussed determinant, transpose, eigen values, how to do those things in R. But in the lectures of linear algebra, we have also studied how to invert a matrix, inversion of a matrix.

(Refer Slide Time: 45:36)

● lec_6_v	week_3A.R × elec_6_week_38.R* × elec_6_week_3C.R ×	-0	Environme	nt 👝
	🔊 🗧 🖸 Source on Save 🔍 🎢 📲	🔫 Run 🛛 😁 Source 🔹 🔍	er 🔒 🛛	🏞 🔹 🌖 173 MiB 🔹
14 #	#		* R * 💼	Global Environment •
15			sum m	n int [1:2, [
	♯ So, AX = b			. num [1:2, [
17	and a standard as able and a standard by	. And the second se	Values	. num [1:2;
18 # 19	# The solution to this system of equation is	X = Inverse_or_A	1.	num [1,5] 24
	# Solve() function performs this work in R		a	num [1:5] 24.
	# It takes A and b as arguments		b	num [1:2] 6 2
22	a concorrana o ao arganeneo		e	int [1:10] 2.
23 - #	# Create A and b		i	num [1:5] 0
24			р	num [1:5] 3
25 A	A <- rbind(c(1,1), c(-3,1))		• q	num [1:5] 13.
18:63	(Top Level) \$	R Script	sol	num [1:2] 1 5
R 4.1	1.2 · C:/dab/ 🖗	1 -0	Plots	-
	rbind(c(1,1), c(-3,1))		·	🥬 Zoom 🛛 🛺 Export
> View(
	c(6, 2)			
> b [1] 6 2	2			
	<- solve(A, b)			
> print				
[1] 1 5				
>				

A \leftarrow rbind(c(1, 1), c(-3, 1))

And as I said, here, to solve a linear equation, you have to do the inversion of a matrix. But today's lecture, I never discuss our inversion of a matrix, which we do very frequently in data analysis. So, how to do that? Actually, the solve function itself can do that. That is a good thing. So, how can I use the solve function to do that?

(Refer Slide Time: 45:58)

lec_6_week_3A.R × O lec_6_week_3B.R* × O lec_6_week_3C.R ×	- Environ	ment (
💷 🗐 📄 🖸 Source on Save 🔍 🎢 🛛 📑	Run 😁 🕞 Source 🔹 😅 🔒	🕐 • 🌖 173 MiB •
35 - # **********************************	* R * 1	Global Environment •
36 • # **********************************		mn int [1:2,
37 - # Matrix inverse		
38		:o num [1:2,
39 # Use solve() function	Values	
40	a	num [1:5] 24
<pre>41 # solve() needs two arguments: A and b</pre>	b	num [1:2] 6 2
42 # If missing, b is considered by default as Identit	ty matrix e	int [1:10] 2
<pre>43 # & solve() returns the inverse of A only. 44</pre>	i	num [1:5] 0
44 45 print(solve(A))	p	num [1:5] 3
46	- q	num [1:5] 13
45:18 🚺 Matrix inverse 🛊	R Script © sol	num [1:2] 1 5
R 4.1.2 · C:/dab/ 🖗	Plots	-
1] 6 2	A	Doom Zoom
sol <- solve(A, b)		r ye zoom i 🚈 Export
print(sol)		
1] 1 5		
<pre>print(solve(A)) [,1] [,2]</pre>		
1,] 0.25 -0.25		
2,] 0,75 0.25		

print(solve(A))

It is very easy, the solve function has been written in such a way that, see I have said I have to give two arguments, A is the coefficient matrix, b as the right hand side vector. Now, if you do not provide b, this function by default will consider b as the identity matrix. And if you remember that, if I multiply any matrix with an identity matrix, then I get the same matrix. So, in this case, it will multiply identity matrix with the inverse of A, so it will give me inverse of A.

So, if you want to get the inversion of a matrix, so what you simply do, you call the solve function, but only provide the matrix do not provide the b vector. So, that is what I am doing here. I am calling solve, only providing A not providing any b vector. So, it will only report me the inversion of A. So, if I run it and print it, this is the inversion of A, A inverse.

So, A inverse is 0.25, minus 0.25, and so on. That is all for this lecture. We have learned how to use R for all those things that we have learned in linear algebra course in this course, in the last week. Thank you for joining me today. Keep on learning. See you in the next video.